



UNIVERSIDAD DE CUENCA



FACULTAD DE INGENIERÍA

MAESTRIA EN GERENCIA DE SISTEMAS DE INFORMACIÓN

ANÁLISIS Y DISEÑO DE UN DATAMART Y LA CONSTRUCCION DE UN
PROTOTIPO DE SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS PARA LA
EMPRESA IMPOCOMJAHER CIA. LTDA.

TESIS PREVIO A LA OBTENCIÓN DEL GRADO DE
MAGISTER EN GERENCIA DE SISTEMAS DE INFORMACIÓN

AUTOR: Ing. Giovanni Chasifán Chicaiza.

DIRECTOR: Ing. José Xavier Mejía Freire. M.Sc.

Junio de 2014

CUENCA - ECUADOR



RESUMEN

Este trabajo está enfocado en el diseño, desarrollo, implementación de un *datamart* y la creación del prototipo para los departamentos de ventas y cobranzas de *Impocomjaher*. El tema se divide en varias etapas, en primer lugar realizamos una revisión de la infraestructura técnica con la que cuenta la empresa, a continuación realizamos un análisis de requerimientos con los gerentes de ventas y de cobranzas para saber exactamente cuáles son sus necesidades en cuanto a la información para la toma de decisiones, la tercera etapa es determinar los indicadores de los dos departamentos estableciendo la correspondencia con los datos transaccionales para determinar cada una de las tablas y los campos que van a servir para la creación del *datamart* y el nivel de granularidad que van a tener dichos indicadores. La cuarta etapa es determinar el modelo lógico del *datamart* estableciendo las tablas de dimensiones y de hechos de los indicadores establecidos y la unión que van a tener entre estos. La quinta etapa es la creación de los procesos de carga inicial tanto de las tablas de dimensiones como de hechos y los procesos de actualización de los indicadores al igual que su periodicidad de actualización (ETL), a continuación realizamos la construcción del *datamart* y un prototipo de visualización de la información para poder consultar los indicadores creados y por último realizamos los manuales operacionales del *datamart*.

PALABRAS CLAVES:

CONSTRUCCIÓN PROTOTIPO, DATAMART, SOLUCIÓN INTELIGENCIA DE NEGOCIOS, IMPOCOMJAHER, INFORMACIÓN ESTRATÉGICA VENTAS Y COBRANZAS, ANÁLISIS Y DISEÑO DATAMART, PROCESOS ETL.



ABSTRACT

This work is focused on design, development and implementation of a *datamart* and the creation of a prototype for the sales and collections departments of *Impocomjaher*. This topic has many steps, First a review of the infrastructure, the second task is a requirements analysis with the managers of sales and collections departments to know which are their specific information needs for decision making, the third step is determining indicators of the departments and the mapping with transactional data to choose the tables and fields for the *datamart* creation and the granularity level of the indicators. The fourth step is building the logic model to establish the dimensions and facts tables and the relationship with the indicators. The fifth step is the creation of the initial loading processes to fill the dimensions and facts tables and also the indicators updating processes and its frequency (ETL). Finally we build the *datamart* and a prototype of visualization of the information to review the indicators, also we make the operational manuals of the *datamart*.

KEYWORDS:

PROTOTYPE, CONSTRUCTION, DATAMART, BUSINESS INTELLIGENCE SOLUTION, IMPOCOMJAHER, SALES AND COLLECTION STRATEGIC INFORMATION, ANALYSIS AND DESIGN OF A DATAMART, ETL PROCESSES.



Índice del Contenido.

RESUMEN	2
ABSTRACT	3
ÍNDICE DE TABLAS	10
ÍNDICE DE ILUSTRACIONES.....	11
AGRADECIMIENTO	15
DEDICATORIA	16
INTRODUCCIÓN	17
CAPÍTULO 1.	18
1.1 Metodología hefesto.....	18
1.2 Presentación del Problema.	20
1.3 Justificación.	20
1.4 Alcance.	21
1.5 Objetivo General.	22
1.6 Objetivos Específicos.....	22
1.7 Estructura de la tesis.	22
CAPÍTULO 2.	24
ANÁLISIS DE LA INFRAESTRUCTURA.....	24
2.1 Requerimientos.	24
2.2 Revisión de hardware.....	24
2.3 Revisión de los Sistemas Operativos.	24
2.4 Informe de Evaluación de la infraestructura técnica.	24
2.4.1 Aplicaciones (Sistema Operativo).	25
2.4.2 Equipos Informáticos (Hardware).	25
2.4.3 Redes de comunicación.....	25
2.4.4 Equipamiento Auxiliar.	25
2.5 Diagnóstico de la infraestructura técnica.	27
CAPÍTULO 3.	28
ANÁLISIS DE REQUERIMIENTOS.	28
3.1 Entrevista.	28
3.1.1 Guía de Entrevista.	28
3.2 Identificación de indicadores y perspectivas de análisis.....	29
3.3 Modelo Conceptual.....	31



CAPÍTULO 4.	41
ANÁLISIS DE LOS OLTP.....	41
4.1 Determinación de indicadores.	41
4.2 Establecer correspondencia con los requerimientos.....	43
4.3 Nivel de granularidad.	46
CAPÍTULO 5.	61
MODELO LÓGICO DE LA ESTRUCTURA DEL DATAMART.	61
5.1 Tipo del modelo lógico del datamart.	61
5.1.1 Tablas de dimensiones.	61
5.1.2 Tablas de hechos.	64
5.1.2.1 Cumplimiento de ventas.....	64
5.1.2.2 Rentabilidad.	64
5.1.2.3 Morosidad de la cartera.	64
5.1.2.4 Flujo corriente.	64
5.1.2.5 Mes presente puntual.	64
5.1.2.6 Mes presente total.....	64
5.1.2.7 Primeras cuotas.....	64
5.1.3 Uniones.	66
5.1.3.1 Cumplimiento de ventas.....	66
5.1.3.2 Rentabilidad.	67
5.1.3.3 Morosidad de la cartera.	68
5.1.3.4 Flujo corriente.	69
5.1.3.5 Mes presente puntual.	70
5.1.3.6 Mes presente total.....	71
5.1.3.7 Primeras cuotas.....	72
CAPÍTULO 6.	73
INTEGRACIÓN DE DATOS.....	73
6.1 Carga inicial.	73
6.1.1 Tabla de dimensión dm_periodo_balances.	73
6.1.2 Tabla de dimensión dm_anio_mes_ventas.....	73
6.1.3 Tabla de dimensión dm_plazos_ventas.	73
6.1.4 Tabla de dimensión dm_lineas.....	74
6.1.5 Tabla de dimensión dm_marcas.	74
6.1.6 Tabla de dimensión dm_productos.....	74



6.1.7	Tabla de dimensión dm_forma_pago.....	74
6.1.8	Tabla de dimensión dm_tipo_cliente.....	74
6.1.9	Tabla de dimensión dm_politicas.....	74
6.1.10	Tabla de dimensión dm_calificacion_del_cliente.....	75
6.1.11	Tabla de dimensión dm_es_empleado.....	75
6.1.12	Tabla de dimensión dm_fechas_vencimientos.....	75
6.1.13	Tabla de dimensión dm_fechas_ventas.....	75
6.1.14	Tabla de dimensión ad_provincias.....	75
6.1.15	Tabla de dimensión ad_cantones.....	75
6.1.16	Tabla de dimensión ad_agencias_x_cantones.....	75
6.1.17	Tabla de dimensión ad_cantones_x_provincias.....	76
6.1.18	Tabla de dimensión ad_provincias_x_regiones.....	76
6.1.19	Tabla de dimensión ad_regiones_x_empresas.....	76
6.1.20	Tabla de dimensión ad_naciones.....	76
6.1.21	Tabla de dimensión ad_empresas.....	76
6.1.22	Tabla de dimensión tg_agencia.....	76
6.1.23	Tabla de hechos dm_indice_cumplimiento_ventas.....	76
6.1.24	Tabla de hechos dm_indice_rentabilidad.....	77
6.1.25	Tabla de hechos cc_indice_de_morosidad.....	77
6.1.26	Tabla de hechos cc_flujo_corriente.....	77
6.1.27	Tabla de hechos cc_produc_mes_presente_puntual.....	77
6.1.28	Tabla de hechos cc_produc_mes_presente_total.....	77
6.1.29	Tabla de hechos cc_primeras_cuotas.....	77
6.2	Actualización de datos.....	78
6.2.1	Tablas de dimensión.....	78
6.2.1.1	Indicador Cumplimiento de Ventas.....	78
6.2.1.2	Indicador de rentabilidad.....	78
6.2.1.3	Indicador Morosidad de la cartera.....	79
6.2.1.4	Indicador Flujo Corriente.....	79
6.2.1.5	Indicador Mes Presente Puntual.....	79
6.2.1.6	Indicador Mes Presente Total.....	80
6.2.1.7	Indicador Primeras Cuotas.....	80
CAPÍTULO 7.	81
CONSTRUCCIÓN DEL PROTOTIPO DE HERRAMIENTA BI.	81



7.1 Instalación de apex y Construcción del prototipo de visualización de la información.	81
7.1.1 Instalación y configuración de apex.....	81
7.1.2 Construcción de la página principal del prototipo de visualización de la información.	83
CONCLUSIONES	86
RECOMENDACIONES	87
GLOSARIO	88
BIBLIOGRAFÍA.....	91
ANEXOS	93
Anexo A1.	94
Aplicaciones (Sistema Operativo).	94
Anexo A2.	95
Equipos Informáticos (Hardware).	95
Anexo A3.	97
Redes de comunicación.....	97
Anexo A4.	98
Equipamiento Auxiliar.	98
Anexo A5.	99
Diccionario de Datos de los OLTP.....	99
Anexo A6.	122
Modelo Lógico de la estructura del datamart – Tabla de Dimensiones.	122
Anexo A7.	165
Modelo Lógico de la estructura del datamart – Tabla de Hechos.	165
Anexo A8.	172
Script carga inicial tabla dimensionamiento dm_periodos_balances.....	172
Anexo A9.	174
Script carga inicial tabla dimensionamiento dm_anio_mes_ventas.....	174
Anexo A10.	176
Script carga inicial tabla dimensionamiento dm_plazos_ventas.	176
Anexo A11.	178
Script carga inicial tabla dimensionamiento dm_lineas.....	178
Anexo A12.	180
Script carga inicial tabla dimensionamiento dm_marcas.	180
Anexo A13.	182



Script carga inicial tabla dimensionamiento dm_productos.....	182
Anexo A14.	184
Script carga inicial tabla dimensionamiento dm_forma_pago.	184
Anexo A15.	186
Script carga inicial tabla dimensionamiento dm_tipo_cliente.....	186
Anexo A16.	188
Script carga inicial tabla dimensionamiento dm_politicas.....	188
Anexo A17.	190
Script carga inicial tabla dimensionamiento dm_calificacion_del_cliente.	190
Anexo A18.	192
Script carga inicial tabla dimensionamiento dm_es_empleado.	192
Anexo A19.	194
Script carga inicial tabla dimensionamiento dm_fechas_vencimientos.....	194
Anexo A20.	196
Script carga inicial tabla dimensionamiento dm_fechas_ventas.....	196
Anexo A21.	198
Script carga inicial tabla dimensionamiento ad_empresas.....	198
Anexo A22.	200
Script carga inicial tabla dimensionamiento tg_agencia.....	200
Anexo A23.	202
Script carga inicial tabla dm_indice_cumplimiento_ventas.....	202
Anexo A24.	210
Script carga inicial tabla dm_indice_rentabilidad.	210
Anexo A25.	221
Script carga inicial tabla cc_indice_de_morosidad.	221
Anexo A26.	235
Script carga inicial tabla cc_flujo_corriente.	235
Anexo A27.	249
Script carga inicial tabla cc_produc_mes_presente_puntual.	249
Anexo A28.	263
Script carga inicial tabla cc_produc_mes_presente_total.	263
Anexo A29.	275
Script carga inicial tabla cc_primeras_cuotas.	275
Anexo A30.	288



Script actualización de tablas de dimensión.....	288
Anexo A31.....	301
Script actualización índice de cumplimiento de ventas.....	301
Anexo A32.....	324
Script actualización índice de rentabilidad.....	324
Anexo A33.....	349
Script actualización índice de morosidad.....	349
Anexo A34.....	375
Script actualización índice flujo corriente.....	375
Anexo A35.....	402
Script actualización índice mes presente puntual.....	402
Anexo A36.....	430
Script actualización índice mes presente total.....	430
Anexo A37.....	455
Script actualización índice primeras cuotas.....	455
Anexo A38.....	484
MANUALES DE OPERACIÓN DE LA HERRAMIENTA.....	484
8.1 Ingreso al sistema.....	484
8.2 Ventas.....	485
8.2.1 Cumplimiento de Ventas.....	485
8.2.2 Rentabilidad.....	489
8.3 Cobranzas.....	495
8.3.1 Morosidad de la cartera.....	496
8.3.2 Flujo Corriente.....	500
8.3.3 Mes Presente Puntual.....	506
8.3.4 Mes Presente Total.....	512
8.3.5 Primeras Cuotas.....	518



ÍNDICE DE TABLAS

Tabla 3.1 Granularidad Cumplimiento de ventas.	47
Tabla 3.2 Granularidad Rentabilidad.	48
Tabla 3.3 Granularidad Morosidad de la cartera.	49
Tabla 3.4 Granularidad Flujo corriente.	50
Tabla 3.5 Granularidad Mes presente puntual.	51
Tabla 3.6 Granularidad Mes presente total.	52
Tabla 3.7 Granularidad Primeras cuotas.	53
Tabla 3.8 Ad_naciones.....	99
Tabla 3.9 Ad_empresas	100
Tabla 3.10 Ad_parametros_generales	101
Tabla 3.11 Ad_regiones_x_empresas.....	102
Tabla 3.12 Tg_agencia.....	103
Tabla 3.13 Ad_provincias	104
Tabla 3.14 Ad_provincias_x_regiones	104
Tabla 3.15 Ad_cantones	105
Tabla 3.16 Ad_cantones_x_provincias.....	105
Tabla 3.17 Ad_agencias_x_cantones	106
Tabla 3.18 Ve_objetivos_x_agencias.....	106
Tabla 3.19 Tg_modelo_item.....	107
Tabla 3.20 Marca	107
Tabla 3.21 Producto	108
Tabla 3.22 Comprobante.....	109
Tabla 3.23 Sta_rentabilidad_temporal.....	111
Tabla 3.24 St_politica_credito	111
Tabla 3.25 Cc_balances_mensuales	112
Tabla 3.26 Forma_pago_comprobante	117
Tabla 3.27 St_cancelacion_d	117
Tabla 3.28 Movimiento	118
Tabla 3.29 Cc_siembra_x_vencimiento	119
Tabla 3.30 Forma_pago	120
Tabla 3.31 Empresa	120
Tabla 3.32 Ad_agencias.....	121



ÍNDICE DE ILUSTRACIONES

Figura 2.1 Modelo conceptual - Ventas.....	32
Figura 2.2 Modelo conceptual – Cumplimiento de ventas.....	33
Figura 2.3 Modelo conceptual – Rentabilidad.....	34
Figura 2.4 Modelo conceptual - Cobranzas.....	35
Figura 2.5 Modelo conceptual – Morosidad de la cartera.....	36
Figura 2.6 Modelo conceptual – Flujo corriente.....	37
Figura 2.7 Modelo conceptual – Mes presente puntual.....	38
Figura 2.8 Modelo conceptual – Mes presente total.....	39
Figura 2.9 Modelo conceptual – Primeras cuotas.....	40
Figura 3.10 Modelo conceptual ampliado – Cumplimiento de ventas.....	54
Figura 3.11 Modelo conceptual ampliado – Rentabilidad.....	55
Figura 3.12 Modelo conceptual ampliado – Morosidad de la cartera.....	56
Figura 3.13 Modelo conceptual ampliado – Flujo corriente.....	57
Figura 3.14 Modelo conceptual ampliado – Mes presente puntual.....	58
Figura 3.15 Modelo conceptual ampliado – Mes presente total.....	59
Figura 3.16 Modelo conceptual ampliado – Primeras cuotas.....	60
Figura 4.67 Uniones – Cumplimiento de ventas.....	66
Figura 4.68 Uniones – Rentabilidad.....	67
Figura 4.69 Uniones – Morosidad de la cartera.....	68
Figura 4.70 Uniones – Flujo corriente.....	69
Figura 4.71 Uniones – Mes presente puntual.....	70
Figura 4.72 Uniones – Mes presente total.....	71
Figura 4.73 Uniones – Primeras cuotas.....	72
Figura 4.17 Tablas de dimensiones - Cumplimiento de ventas - Ubicación geográfica.....	122
Figura 4.18 Tablas de dimensiones - Cumplimiento de ventas - Tiempo.....	123
Figura 4.19 Tablas de dimensiones – Rentabilidad - Ubicación geográfica... ..	124
Figura 4.20 Tablas de dimensiones – Rentabilidad - Producto.....	125
Figura 4.21 Tablas de dimensiones – Rentabilidad - Plazo.....	126
Figura 4.22 Tablas de dimensiones – Rentabilidad - Tiempo.....	127
Figura 4.23 Tablas de dimensiones – Rentabilidad – Forma pago.....	128
Figura 4.24 Tablas de dimensiones – Rentabilidad – Línea.....	129
Figura 4.25 Tablas de dimensiones – Rentabilidad – Marca.....	130
Figura 4.26 Tablas de dimensiones – Rentabilidad – Tipo de cliente.....	131
Figura 4.27 Tablas de dimensiones – Morosidad de la cartera – Ubicación geográfica.....	132
Figura 4.28 Tablas de dimensiones - Morosidad de la cartera - Tiempo.....	133
Figura 4.29 Tablas de dimensiones – Morosidad de la cartera - Línea.....	134
Figura 4.30 Tablas de dimensiones – Morosidad de la cartera - Marca.....	135
Figura 4.31 Tablas de dimensiones – Morosidad de la cartera - Política.....	136



Figura 4.32 Tablas de dimensiones – Flujo corriente – Ubicación geográfica.	137
Figura 4.33 Tablas de dimensiones - Flujo corriente - Tiempo.	138
Figura 4.34 Tablas de dimensiones – Flujo corriente - Línea.	139
Figura 4.35 Tablas de dimensiones – Flujo corriente - Marca.	140
Figura 4.36 Tablas de dimensiones – Flujo corriente - Política.	141
Figura 4.37 Tablas de dimensiones – Flujo corriente – Calificación cliente. ..	142
Figura 4.38 Tablas de dimensiones – Flujo corriente – Es empleado.	143
Figura 4.39 Tablas de dimensiones – Mes presente puntual – Ubicación geográfica.	144
Figura 4.40 Tablas de dimensiones - Mes presente puntual - Tiempo.	145
Figura 4.41 Tablas de dimensiones – Mes presente puntual - Línea.	146
Figura 4.42 Tablas de dimensiones – Mes presente puntual - Marca.	147
Figura 4.43 Tablas de dimensiones – Mes presente puntual - Política.	148
Figura 4.44 Tablas de dimensiones – Mes presente puntual – Calificación cliente.	149
Figura 4.45 Tablas de dimensiones – Mes presente puntual – Es empleado.	150
Figura 4.46 Tablas de dimensiones – Mes presente total – Ubicación geográfica.	151
Figura 4.47 Tablas de dimensiones - Mes presente total - Tiempo.	152
Figura 4.48 Tablas de dimensiones – Mes presente total - Línea.	153
Figura 4.49 Tablas de dimensiones – Mes presente total - Marca.	154
Figura 4.50 Tablas de dimensiones – Mes presente total - Política.	155
Figura 4.51 Tablas de dimensiones – Mes presente total – Calificación cliente.	156
Figura 4.52 Tablas de dimensiones – Mes presente total – Es empleado.	157
Figura 4.53 Tablas de dimensiones – Primeras cuotas – Ubicación geográfica.	158
Figura 4.54 Tablas de dimensiones - Primeras cuotas - Tiempo.	159
Figura 4.55 Tablas de dimensiones – Primeras cuotas - Línea.	160
Figura 4.56 Tablas de dimensiones – Primeras cuotas - Marca.	161
Figura 4.57 Tablas de dimensiones – Primeras cuotas - Política.	162
Figura 4.58 Tablas de dimensiones – Primeras cuotas – Calificación cliente.	163
Figura 4.59 Tablas de dimensiones – Primeras cuotas – Es empleado.	164
Figura 4.60 Tablas de hechos – Cumplimiento de ventas.	165
Figura 4.61 Tablas de hechos – Rentabilidad.	166
Figura 4.62 Tablas de hechos – Morosidad de la cartera.	167
Figura 4.63 Tablas de hechos – Flujo corriente.	168
Figura 4.64 Tablas de hechos – Mes presente puntual.	169
Figura 4.65 Tablas de hechos – Mes presente total.	170
Figura 4.66 Tablas de hechos – Primeras cuotas.	171



RECONOCIMIENTO DE LOS DERECHOS DE AUTOR DE LA UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA
Fundada en 1867

Yo, Giovanny Chasifán Chicaiza, autor de la tesis "Análisis y diseño de un datamart y la construcción de un prototipo de solución de inteligencia de negocios para la empresa Impocomjaher Cía. Ltda.", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Magister en Gerencia de Sistemas de Información. El uso que la Universidad de Cuenca hiciera de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, 11 de Noviembre de 2013

Ing. Giovanny Chasifán Chicaiza
0102632528



DECLARACION DE RESPONSABILIDAD



UNIVERSIDAD DE CUENCA
Fundada en 1867

Yo, Giovanni Chasifán Chicaiza, autor de la tesis “Análisis y diseño de un datamart y la construcción de un prototipo de solución de inteligencia de negocios para la empresa Impocomjaher Cía. Ltda.”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 11 de Noviembre de 2013

Ing. Giovanni Chasifán Chicaiza.
0102632528



AGRADECIMIENTO

Agradezco a Dios y la Virgen María, especialmente a mi madre y hermana Mariana por todo el apoyo que me entregaron durante su paso por este mundo, también agradezco a mi hija Gabriela por su paciencia y comprensión durante estos años, a mis hermanos Willian y Wilson por su apoyo.

Ing. Giovanni Chasifán Chicaiza



DEDICATORIA

Dedico este trabajo a la memoria de mi madre y hermana, Dios y la Virgen quiso tenerlas en su gloria a mi hermana desde 1992 y a mi madre desde Enero de 2013. Durante sus existencias siempre estuvieron a mi lado apoyándome en los momentos más duros y difíciles de mi vida, entregándome todo su amor, comprensión, cariño, ternura y bondad, luchando siempre a mi lado por cada nueva meta que me ponía en mi vida, en este momento no pueden estar físicamente a mi lado en la última meta que nos propusimos años atrás, pero estoy seguro que espiritualmente están feliz porque logre concluir. Madre y hermana, algún día nos volveremos a ver, siempre y cuando haga bien las cosas en la tierra como ustedes lo hicieron durante su paso por este mundo.

Dedicado a nuestra madre de parte de tu hija
Gabriela:

“Siempre has sido nuestro ángel, dándonos amor, paz y felicidad, guiándonos por el camino del bien y de la fe.

Por eso te entregamos a Dios a la virgencita y al niño Jesús, porque te mereces el descanso, la paz y la felicidad eterna.

Siempre estarás en nuestra mente y corazón
Madrecita adorada.”

Ing. Giovanni Chasifán Chicaiza



INTRODUCCIÓN

Impocomjaher Cía. Ltda., fue fundada en Cuenca en 1989 como un almacén de electrodomésticos local, a partir de 1996 forma una cadena nacional con 33 almacenes en 15 ciudades del país, actualmente tiene más de 70 agencias en todo el país convirtiéndose en una de las cadenas más importantes del Ecuador. *Impocomjaher* brinda a sus clientes atención personalizada, garantía de calidad en todos sus productos, servicio de mantenimiento técnico especializado, crédito inmediato y en condiciones favorables. Durante estos 23 años de actividad comercial, seria y responsable; ha contribuido al desarrollo empresarial del país, atendiendo eficientemente las necesidades de los hogares ecuatorianos y es fuente de empleo directo para más 700 familias ecuatorianas.

Impocomjaher se encuentra en la búsqueda de implementar estrategias que le permita responder a las cambiantes condiciones del mercado.

Con la era de la información y sus ventajas contamos con posibilidades que acortan las distancias y aumentan los beneficios para las empresas que tienen acceso a los datos, pero esto ha llevado a que tengan problemas con el manejo de grandes volúmenes de información. En el mundo empresarial se diferencian entre ellas por la información que manejan, el conocimiento del mercado y la toma de decisiones, por eso es necesario saber cómo y dónde se organiza la información.

En vista de esto van surgiendo conceptos y filosofías de trabajo, como el *datamart*. Un *datamart*, permite la generación de la información necesaria para generar estadísticas que ayuden a los usuarios estratégicos de *Impocomjaher*, analizar el desempeño y los factores que ayuden a resolver problemas de la empresa.

El objetivo principal de esta tesis es el “Análisis, diseño e implementación de un *datamart* para los departamentos de Ventas y Cobranzas para la empresa *Impocomjaher*, alineado a sus objetivos estratégicos, y que sirva de fuente de información consistente, única, uniforme, confiable; y además brindar oportunamente la información necesaria para la toma de decisiones”. Esta tesis no se va a centrar en el análisis o evaluación del *business intelligence*, razón por la cual solo se va a crear un ejemplo o prototipo para la visualización del *datamart* implementado.

Para el desarrollo de esta tesis es necesario el uso de herramientas metodológicas que permitan analizar, diseñar e implementar un *datamart*, para ello se usará como referencia la metodología hefeito, adaptándola a la realidad de la empresa con la finalidad de obtener un resultado acorde a las necesidades reales de la organización.



CAPÍTULO 1.

En este capítulo describiremos la metodología utilizada para el desarrollo de esta tesis, el problema a resolver, justificaremos por qué realizar esta tesis, identificaremos el alcance que vamos a tener, los objetivos que perseguimos y cómo va a estar estructurado el resto de capítulos.

1.1 Metodología hefesto.

Para la creación del datamart que es la parte central de esta tesis hemos utilizamos la metodología HEFESTO, esta metodología fue creada por el ing. Ricardo Dario Bernabeu, su primera versión fue lanzada en noviembre de 2007, está creada bajo los términos de licencia de documentación libre de GNU (GFDL – GNU Free Documentation License) esto permite proteger su libre difusión, utilización y actualización.

La metodología HEFESTO transmite la comprensión de cada uno de los pasos que se realizará para la creación de un datawarehouse, de tal forma que sepa exactamente que está realizando y porque razón se lo hace. Esta metodología se ha basado en una investigación comparando metodologías existentes y aportando experiencias del autor y de todos aquellos quienes han utilizado esta metodología en los procesos de la creación del datawarehouse.

Hefesto no ingresa en fases amplias de recolección de requerimientos ni en fases de desarrollo muy extensos que conllevan a tiempos demasiados largos en la presentación del producto final.

El principal objetivo de la metodología hefesto es entender y comprender cada uno de los pasos que se van realizando en el transcurso de la creación del datawarehouse, de tal forma que se sepa exactamente lo que se está realizando y por qué.

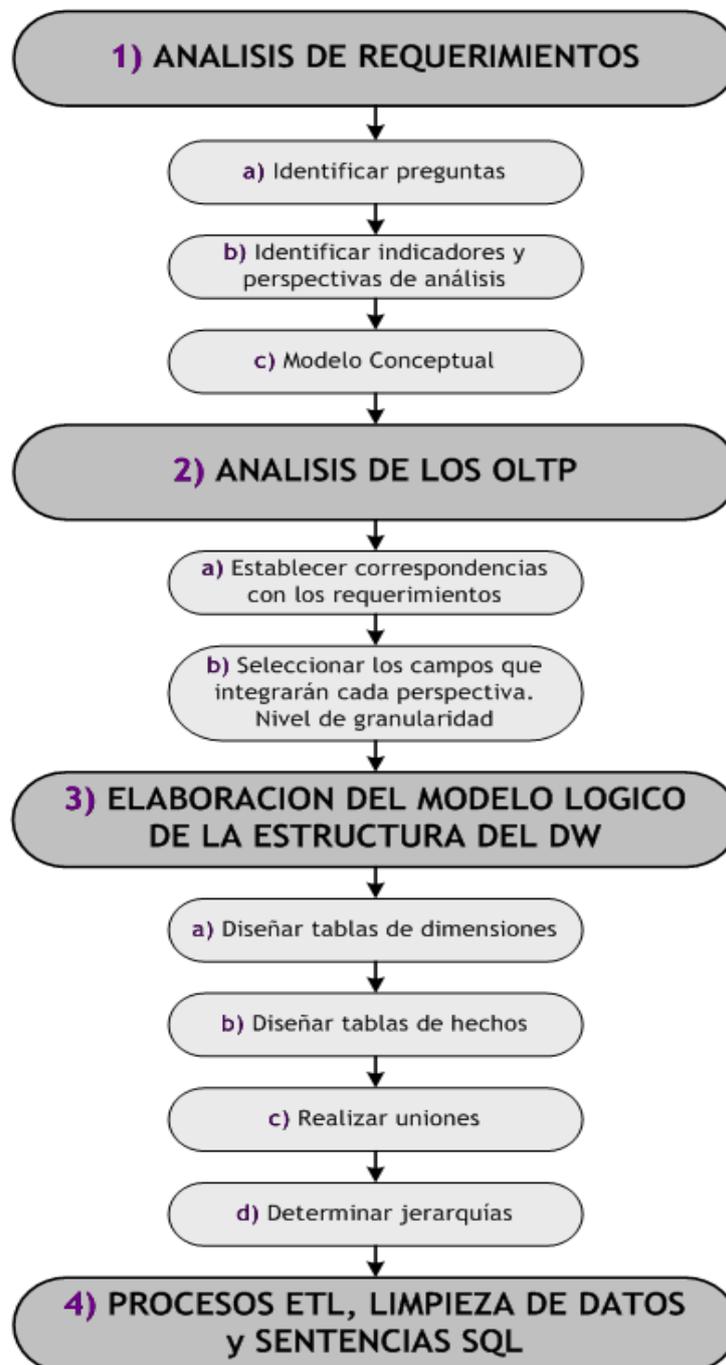
La metodología hefesto comienza con un análisis de requerimiento en donde se obtiene las necesidades de los usuarios para luego identificar los indicadores con las respectivas perspectivas de análisis para a través de esto construir el modelo conceptual del datawarehouse.

El siguiente paso es realizar el análisis de los oltp para saber de dónde se va a obtener la información para construir los indicadores, estableciendo la correspondencia con las fuentes de datos y seleccionando los campos de estudio de cada perspectiva.

A continuación se construirá el modelo lógico del datawarehouse estableciendo el tipo de esquema que se va a utilizar para luego construir las tablas de dimensiones, de hecho y la unión entre ellas.

Finalmente se construirán los procesos ETL en donde se definirán los procesos para la carga inicial del datawarehouse y las actualizaciones respectivas de cada indicador.

La metodología hefesto se puede resumir mediante el siguiente gráfico:



Fuente: Metodología Hefesto.
Elaboración: El autor.



1.2 Presentación del Problema.

En vista del crecimiento vertiginoso empresarial durante estos años *Impocomjaher* necesita de información adecuada que le ayuden a tomar decisiones en cuanto a las ventas y cobranzas, por lo cual hemos visto la carencia en aspectos tales como:

- A. La falta de capacidad de respuesta a los cambios continuos del mercado por no tener la información necesaria.
- B. Determinados procesos son ejecutados fuera de tiempo y no cuando se necesita realmente.
- C. Falta de flexibilidad en la generación de información para análisis paralelos, comparativos o cruce de información.
- D. No contar con información oportuna y en línea para la toma de decisiones.
- E. Falta de indicadores de apoyo en la gestión de ventas.
- F. Falta de indicadores de información estratégica para la toma de decisiones en la cobranza.

Estos problemas han ocasionado que los ejecutivos de la empresa no tengan información suficiente para tomar decisiones y satisfacer las necesidades de los clientes actuales y nuevos de la empresa *Impocomjaher*.

1.3 Justificación.

Impocomjaher al carecer de un apoyo tecnológico para la obtención de la información necesaria, directa y oportuna para la toma de decisiones a nivel ejecutivo, ha creído conveniente apoyar con el desarrollo de esta tesis para que ésta sirva de base para el comienzo de un gran proyecto integrador para la empresa.

Impocomjaher habrá logrado contar con un prototipo de *datamart* que le permitirá tener una información proveniente de los sistemas operacionales, de una manera organizada, integrada y adecuada, además de tener una fuente única y confiable para la toma de decisiones.

El *datamart* podrá ser actualizado diariamente, semanalmente, mensualmente o anualmente de acuerdo a las necesidades de los gerentes departamentales.

La creación de un prototipo de *datamart* para los departamentos de ventas y cobranzas de la empresa *Impocomjaher* permitirá tener un mayor conocimiento y valuación de sus procesos para satisfacer las necesidades de la ciudadanía a nivel nacional; y, que los clientes actuales y nuevos se sientan satisfechos y seguros con los servicios que presta la empresa *Impocomjaher*.



1.4 Alcance.

La empresa *Impocomjaher* cuenta con las áreas Financiero, Ventas, Crédito, Cobranzas, Importaciones, Recursos Humanos, Publicidad, Tecnología, Logística y Bodegas; para el presente estudio tomaremos los departamentos de Ventas y Cobranzas con la finalidad de realizar el análisis, diseño de un *datamart* y la construcción de un prototipo de solución de Inteligencia de Negocios de tal forma que cumpla con los objetivos señalados anteriormente y que el mismo sirva como una herramienta tecnológica necesaria para apoyar y soportar el proceso de toma de decisiones enmarcado en los objetivos estratégicos y empresariales del negocio. El giro del negocio de *Impocomjaher* son las ventas y la recuperación de la cartera, razón por la cual las gerencias necesitan la información rápida y oportuna y es por esto que se escoge estos dos departamentos. Se partirá de la bibliografía existente del *datawarehouse* en cuanto a las características, estructura y arquitectura, luego se aplicará esta teoría para realizar el análisis, diseño y elaboración del *datamart* y la construcción de un prototipo de una solución de Inteligencia de Negocio para la empresa *Impocomjaher*. La información se tomará de los sistemas transaccionales que posee la empresa *Impocomjaher* desde el año 2005 hasta la fecha, además el departamento de ventas se centrará en los indicadores de:

- Cumplimiento de ventas.
- Rentabilidad de la empresa.

Para el departamento de cobranzas se enfocará en los indicadores de:

- Morosidad de la cartera, retraso de pago que tienen los clientes a sus obligaciones pactadas a una fecha determinada.
- Flujo corriente, se refiere a la recuperación de la cartera de los créditos que se encuentren al día y cuya cuota se vence en el mes que se evalúa.
- Mes presente puntual, es la recuperación de la cartera de los créditos que se encuentren al día y cuya cuota se vence en el mes que se evalúa discriminando las primeras cuotas.
- Mes presente total, es la recuperación de la cartera de todos los créditos cuya cuota se vence en el mes que se evalúa.
- Primeras cuotas, corresponde a la recuperación de la cartera de las primeras cuotas cuya fecha de vencimiento se encuentre en el mes que se evalúa.



Para el almacenamiento y extracción de la información se utilizará la base de datos ORACLE y para la creación del prototipo de la solución de inteligencia de negocio se utilizará la herramienta APEX de oracle. Es importante indicar que se utilizará la herramienta APEX para la visualización inicial de la información del datamart, debido a que esta tesis se centrará en el análisis y desarrollo de un datamart y no en la solución de business intelligence.

1.5 Objetivo General.

Diseñar un *datamart* para los departamentos de Ventas y Cobranzas para la empresa *Impocomjajer* alineado a sus objetivos estratégicos que sirva de fuente de información consistente, única, uniforme y confiable y además brindar oportunamente la información necesaria para la toma de decisiones mediante el desarrollo de un prototipo de solución de inteligencia de negocio que utilice el *datamart* creado.

1.6 Objetivos Específicos.

- A. Analizar la situación técnica y las necesidades de información para la toma de decisiones de los departamentos de Ventas y Cobranzas de *Impocomjajer*.
- B. Identificar los indicadores más relevantes para los departamentos de Ventas y Cobranzas que apoyen a la toma de decisiones.
- C. Diseñar una estructura de datos eficiente y confiable para la adquisición y consolidación de información (*datamart*).
- D. Crear un prototipo de solución de inteligencia de negocio para la toma de decisiones en las áreas de Ventas y Cobranzas, basándose en información integrada y global del negocio plasmado en el *datamart* diseñado.
- E. Contar con una fuente de información completa y confiable para ayudar en el proceso de toma de decisiones.

1.7 Estructura de la tesis.

Los capítulos restantes de esta tesis están organizados en 7 capítulos de acuerdo al siguiente esquema:

Capítulo 2. Análisis de la infraestructura. Se realizará una revisión de la infraestructura (hardware y los sistemas operativos) que posee la empresa.

Capítulo 3. Análisis de requerimientos. En este capítulo realizaremos la entrevista a los gerentes de ventas y cobranzas, identificaremos los indicadores y las perspectivas de análisis en base a las entrevistas y estableceremos el modelo conceptual de los departamentos de ventas y cobranzas.



Capítulo 4. Análisis de los OLTP. En este capítulo se indicará las fórmulas de cálculo para obtener los indicadores, se establecerá la correspondencia entre el modelo conceptual y las fuentes de datos, luego se definirán los campos que va a contener cada perspectiva, a continuación se desarrollará el modelo conceptual ampliado colocando bajo cada perspectiva los campos elegidos y bajo cada indicador su respectiva fórmula de cálculo.

Capítulo 5. Modelo lógico de la estructura del datamart. Una vez definido en el capítulo anterior el modelo lógico con los atributos que va a tener cada perspectiva de análisis y las fórmulas de cálculo, en este capítulo procederemos a crear el modelo lógico de la estructura del datamart, tomando el modelo conceptual ampliado como base. Definiremos primero el tipo de modelo que utilizaremos, luego diseñaremos las tablas de dimensión, las tablas de hecho y al final realizaremos la unión entre ellas.

Capítulo 6. Integración de datos. En este capítulo realizaremos los procesos ETL para extraer los datos de los sistemas OLTP, para lo cual crearemos los procesos PISql mediante los cuales extraeremos la información. Al momento de generar los ETL se considerará todas las condiciones y restricciones para almacenar la información en el *datamart*

Capítulo 7. Construcción del prototipo de herramienta de B.I. En este capítulo se realizará la construcción del prototipo de herramienta de BI, que consiste en un visualizador de la información almacenada en el *datamart* creado para *Impocomjaher*. La aplicación utilizada para la creación de la herramienta de visualización será apex de Oracle, describiremos en forma general los pasos que se realizaron para la instalación y configuración de apex, también vamos a describir la creación de la página principal con sus opciones para tener una idea de cómo se utilizó la herramienta apex.

Anexos. En esta sección se encuentra todos los gráficos, tablas, script y los manuales de operación del *datamart*.



CAPÍTULO 2.

ANÁLISIS DE LA INFRAESTRUCTURA.

En este capítulo se realiza un levantamiento de la información de la infraestructura informática que posee la empresa con la finalidad de identificar las fuentes posibles de información requerida para el *datamart*, los posibles servidores donde se puede ubicar en producción el proyecto así como el servidor donde se va a desarrollar.

2.1 Requerimientos.

Para la creación del *datamart* y la configuración de apex 4.2.2 vamos a necesitar una base de datos 11gR2 y este debe estar instalado en un servidor con las siguientes características mínimas de hardware:

- Memoria RAM mayor o igual a 16Mb.
- Swap 16 Gb.
- Espacio disponible en disco 1.3Gb.
- Sistema operativo Red Hat Enterprise Linux 5 update 2 o Red Hat Enterprise Linux 5 update 5 de 64 bits.
- kernel 2.6.18 o superior.

2.2 Revisión de hardware.

Para la revisión del hardware que dispone *Impocomjager* se solicitó la colaboración del administrador de redes y hardware a nivel nacional, para lo cual se utilizó un formulario (ver Anexo A2.) en donde se describió cada uno de los servidores que posee la empresa con cobertura a nivel nacional y que se encuentran localizados en la ciudad de Cuenca.

2.3 Revisión de los Sistemas Operativos.

De la misma forma para la revisión de los sistemas operativos se coordinó con el administrador de redes y hardware a nivel nacional de *Impocomjager* utilizando un formulario (ver Anexo A1.) para inventariar los sistemas operativos que utiliza los diferentes servidores de la empresa con cobertura a nivel nacional y que se encuentran localizados en la ciudad Cuenca.

2.4 Informe de Evaluación de la infraestructura técnica.

A continuación describiremos los sistemas operativos, servidores y equipos auxiliares que posee la empresa:



2.4.1 Aplicaciones (Sistema Operativo).

Impocomjajer cuenta con dos tipos de sistemas operativos en los servidores, Windows y RedHat, el sistema operativo redhat es utilizado para la base de datos y Windows es utilizado para terminal server. La descripción detallada de los sistemas operativos puede consultarse en Anexo A1.

2.4.2 Equipos Informáticos (Hardware).

Existen 12 servidores que son utilizados para:

- 1 servidor almacena las transacciones de la empresa.
- 1 servidor utilizado para pruebas.
- 1 servidor para la administración del EVA.
- 8 terminal server para la conexión de los usuarios.
- 1 servidor de aplicaciones.

El detalle de los servidores puede ser consultado en el Anexo A2.

2.4.3 Redes de comunicación.

Impocomjajer cuenta con la siguiente infraestructura de red.

1. Red Lan usando protocolo TCP/IP (intranet).
2. Switch 3COM de 24 puertos, 10/100 con 2 puertos 10/100/1000 de Fibra Óptica.
3. Router Cisco 1750 con 2 interfaces Eth 10/1000.

EL diagrama completo de la infraestructura de red se encuentra en el Anexo A3.

2.4.4 Equipamiento Auxiliar.

1. Generador de Energía.

Marca: Power Generation
Modelo: ES12D6
RATED BHP: 22
RATED kw : 12
RATED KVA: 12
FULL LOAD CUR (1.0pf): 55 AMP
RATED VOLTAGE: 220V
Frequently : 60 Hz.
RATED RPM: 1800
BATTERY VOLTAGE: 12 V DC



GENSET MAX. WEIGHT: 527 KG.

2. Ups Servidor base de datos. Sostiene todos los servidores de la sala de servidores.

Ups powercom.
Banco de Baterías de 12 Kva.

3. Aire acondicionado en la sala de servidores.

El diagrama de los equipos auxiliares puede consultarse en el Anexo A4.



2.5 Diagnóstico de la infraestructura técnica.

El servidor que tiene mejores prestaciones de rendimiento y cumple con los requerimientos mínimos indicados en la sección 2.1 es el servidor de base de datos de producción. En este servidor crearemos el *datamart* para *Impocomjaher*.

A continuación describiremos las características del servidor seleccionado:

Servidor	Espacio Disponible en Disco	Procesador	Sistema Operativo	Versión Sistema Operativo	Arquitectura
Base de datos Producción JAHER	1200Gb	2 CPU Six Core Intel Xeon 2933MHz (equivalente 24 microprocesadores)	RedHat Enterprise	5.5	x64



CAPÍTULO 3.

ANÁLISIS DE REQUERIMIENTOS.

En este capítulo realizaremos la entrevista a los gerentes de ventas y cobranzas, en base a las entrevistas identificaremos los indicadores y las variables o perspectivas de análisis que más le interesa conocer y estableceremos el modelo conceptual de los departamentos de ventas y cobranzas.

3.1 Entrevista.

En esta etapa se realizó una entrevista en forma individual con los gerentes departamentales nacionales de Ventas y Cobranzas.

A continuación describiremos el proceso realizado para obtener la información. Realizamos varias reuniones de trabajo con cada gerente departamental, comenzamos la reunión con una explicación de lo que pretendemos obtener con este proyecto, luego realizamos una lluvia de ideas de las necesidades de información para mejorar su toma de decisiones, a continuación los gerentes departamentales seleccionaron los indicadores más importantes según su criterio, definimos lo que debería contener cada indicador (perspectivas luego llamado dimensión) y finalmente con toda la información recopilada en las reuniones creamos un cuestionario donde se plasmó las necesidades de cada departamento para dejar constancia de sus necesidades.

3.1.1 Guía de Entrevista.

Material utilizado: grabadora, papel, esfero, pizarrón y cuestionario.

3.1.2 Formato del Cuestionario.

Las respuestas entregadas en este cuestionario nos servirán para conocer de mejor manera la información que usted necesita para alcanzar los objetivos planteados en el departamento.

Nombre:

Departamento:

Cargo:

- 1. ¿Cuáles son los indicadores más importantes que usted considera se deben desarrollar?**
- 2. Realice una descripción de cada uno de los indicadores.**
- 3. ¿Cuál es la frecuencia de la medición o análisis de los indicadores?**
- 4. ¿Cuáles son las posibles perspectivas de análisis de cada indicador?**



3.2 Identificación de indicadores y perspectivas de análisis.

Una vez definido en la sección 3.1 los indicadores y sus perspectivas de análisis (más adelante llamado dimensiones), daremos una introducción a los conceptos de indicador y perspectiva utilizados en esta tesis.

Existen muchas definiciones de indicador pero queremos compartir la siguiente definición proporcionada por un catedrático del tecnológico de monterrey de inteligencia de negocios.

Indicador: un indicador es como una cinta métrica que sirve para medir en cualquier momento histórico sin importar si el resultado de la medición es bueno o malo, es decir es una herramienta de medición de objetivos, solo mide no opina.

Perspectiva: en la etapa del modelo conceptual de la metodología hefeito utiliza las palabras perspectiva de análisis, se refiere a los objetos mediante los cuales se quiere examinar los indicadores, también se le conoce como las variables de análisis o temas de interés que permiten estudiar a la información desde diferentes puntos de vista, en la etapa del modelo lógico del datamart las perspectivas se convierten en dimensiones.

En base a la entrevista y cuestionario realizado a los gerentes departamentales de Ventas y Cobranzas en la sección 3.1, se ha establecido los siguientes indicadores y perspectivas de análisis por ser de alta prioridad para ellos.

3.2.1 Departamento de Ventas:

Para el departamento de ventas se definieron los siguientes indicadores y perspectivas de análisis:

Departamento de Ventas	
Indicador	Perspectiva de análisis
Cumplimiento de ventas	Ubicación geográfica
	Tiempo
Rentabilidad	Ubicación geográfica
	Producto
	Forma pago
	Plazo
	Tiempo
	Línea
	Marca
Tipo cliente	



3.2.2 Departamento de Cobranzas:

Para el departamento de cobranzas se definieron los siguientes indicadores y perspectivas:

Departamento de Cobranzas	
Indicador	Perspectiva de análisis
Morosidad de la cartera	Ubicación geográfica
	Línea
	Marca
	Política de venta
	Tiempo
Flujo Corriente	Ubicación geográfica
	Política de venta
	Línea
	Marca
	Tiempo
	Calificación del Cliente
Mes presente puntual	Ubicación geográfica
	Política de venta
	Línea
	Marca
	Tiempo
	Calificación del Cliente
Mes presente total	Ubicación geográfica
	Política de venta
	Línea
	Marca
	Tiempo
	Calificación del Cliente
Primeras cuotas	Ubicación geográfica
	Política de venta
	Línea
	Marca
	Tiempo
	Calificación del Cliente
	Es empleado



3.3 Modelo Conceptual.

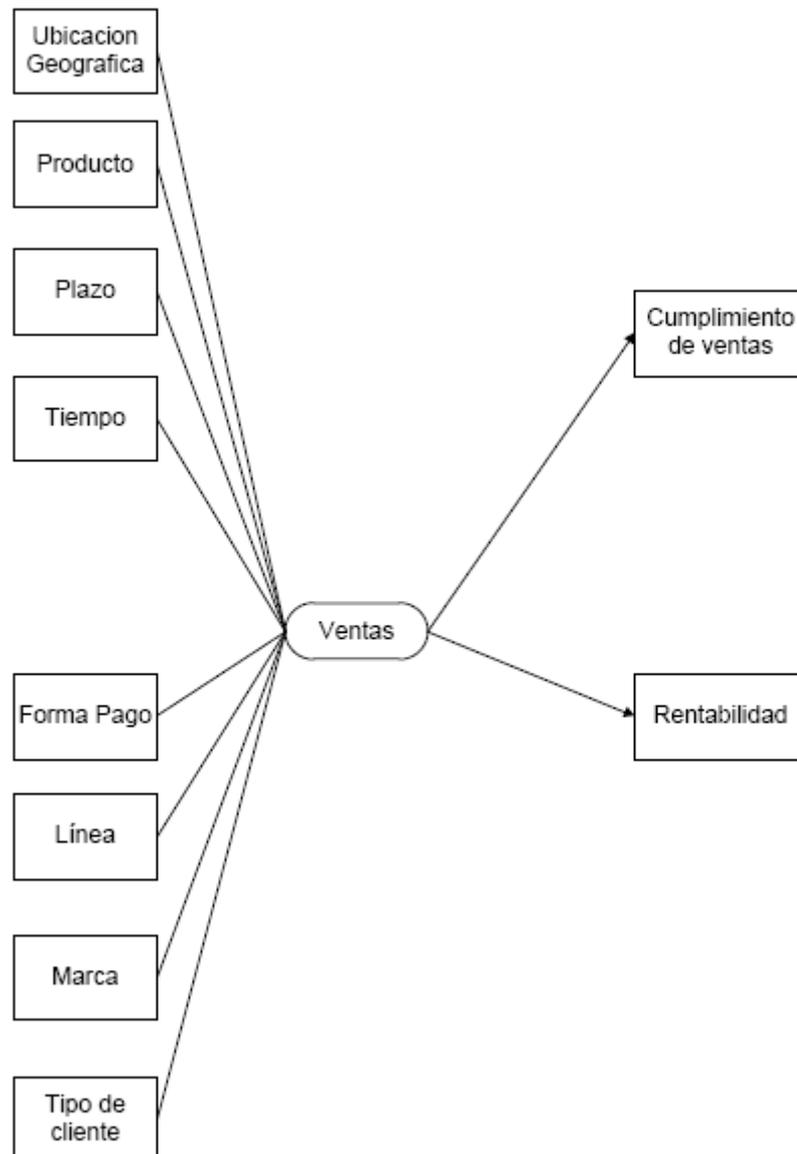
En la sección 3.2 hemos identificado los indicadores y las perspectivas de análisis de los departamentos de ventas y cobranzas, en esta sección realizaremos el modelo conceptual del datamart. Es importante indicar la simbología utilizada en el modelo conceptual utilizada por el método hefesto, en el lado izquierdo representado por un rectángulo se colocan las perspectivas seleccionadas, que serán unidas a un óvalo central que representa la relación que existe entre ellas. La relación, constituye el proceso o área de estudio elegida, de dicha relación y entrelazadas con flechas se desprenden los indicadores y se ubican a la derecha del esquema (rectángulo).

A continuación vamos a desarrollo el modelo conceptual de los indicadores de los departamentos de ventas y cobranzas.

3.3.1 Ventas.

En la Figura 2.1 graficamos el modelo conceptual del departamento de ventas de forma global. Esta figura muestra todas las perspectivas de análisis que va tener el departamento de ventas y los indicadores a ser desarrollados.

Figura 2.1 Modelo conceptual - Ventas.

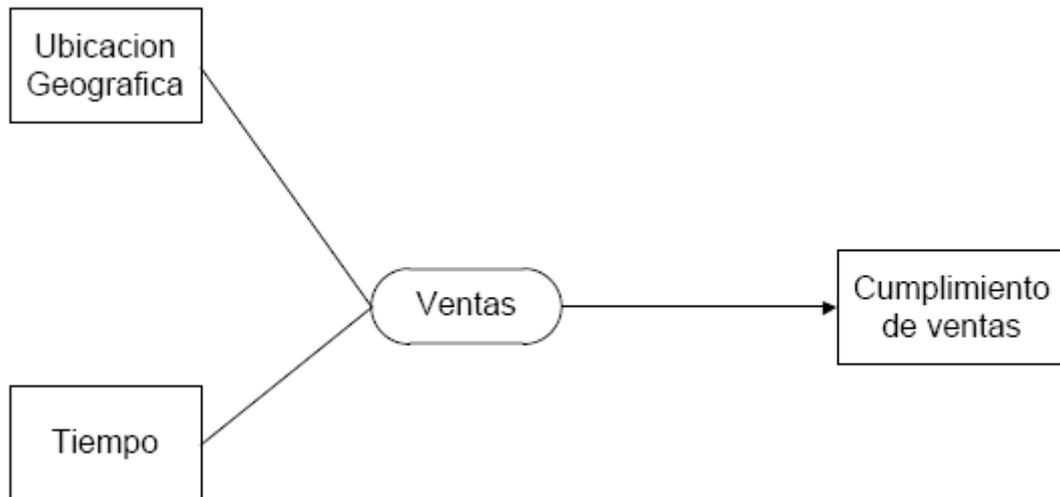


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

3.3.1.1 Ventas – Cumplimiento de ventas.

El cumplimiento de ventas corresponde al objetivo planteado versus las ventas netas realizadas. En la Figura 2.2 graficamos el indicador de cumplimiento de ventas y las dos perspectivas de análisis que va a tener dicho indicador.

Figura 2.2 Modelo conceptual – Cumplimiento de ventas.

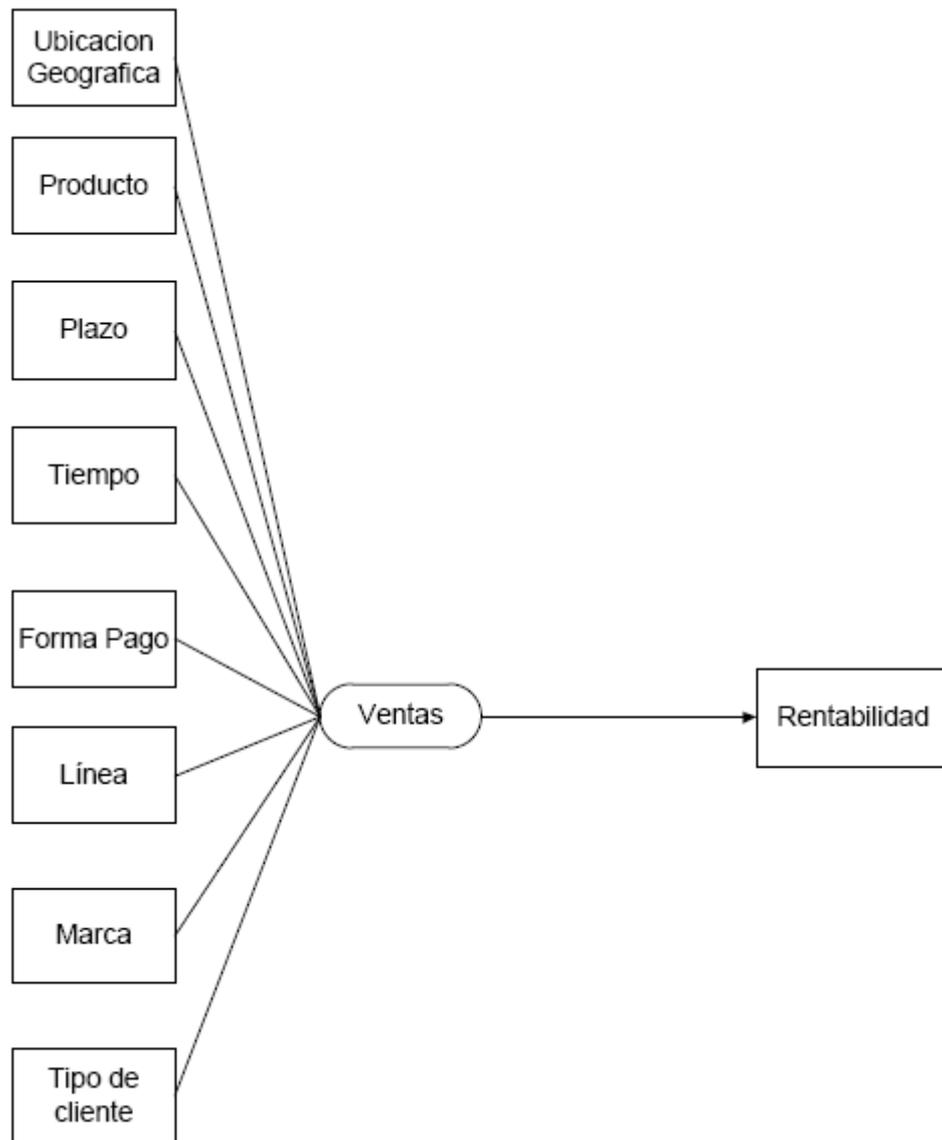


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

3.3.1.2 Ventas – Rentabilidad.

En la Figura 2.3 graficamos el indicador de rentabilidad y las siete perspectivas de análisis que va a tener el indicador.

Figura 2.3 Modelo conceptual – Rentabilidad.



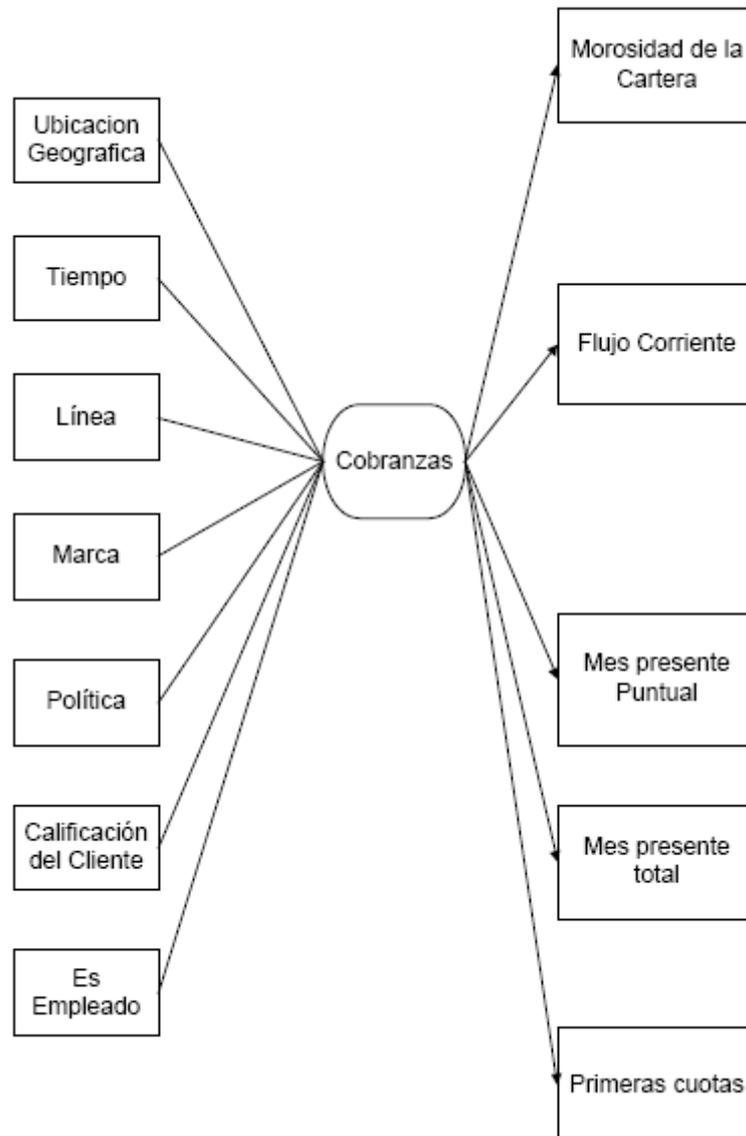
Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.

3.3.2 Cobranzas.

En la Figura 2.4 graficamos el modelo conceptual del departamento de cobranzas de forma global. Esta figura muestra todas las perspectivas de análisis que va a tener el departamento de cobranzas y los indicadores a ser desarrollados.

Figura 2.4 Modelo conceptual - Cobranzas.

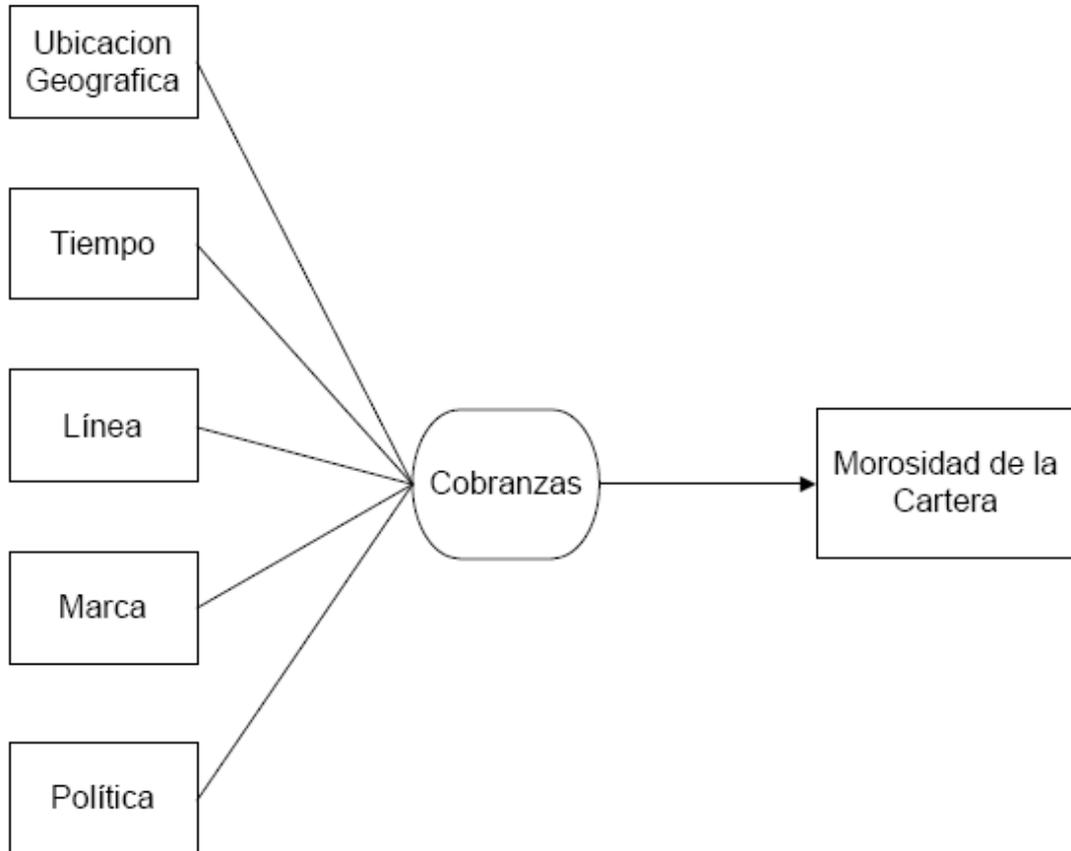


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

3.3.2.1 Cobranzas – Morosidad de la Cartera.

La morosidad de la cartera es el retraso de pago que tienen los clientes a sus obligaciones pactadas a una fecha determinada, en la Figura 2.5 graficamos el indicador de morosidad de la cartera y las cinco perspectivas de análisis que va a tener el indicador.

Figura 2.5 Modelo conceptual – Morosidad de la cartera.



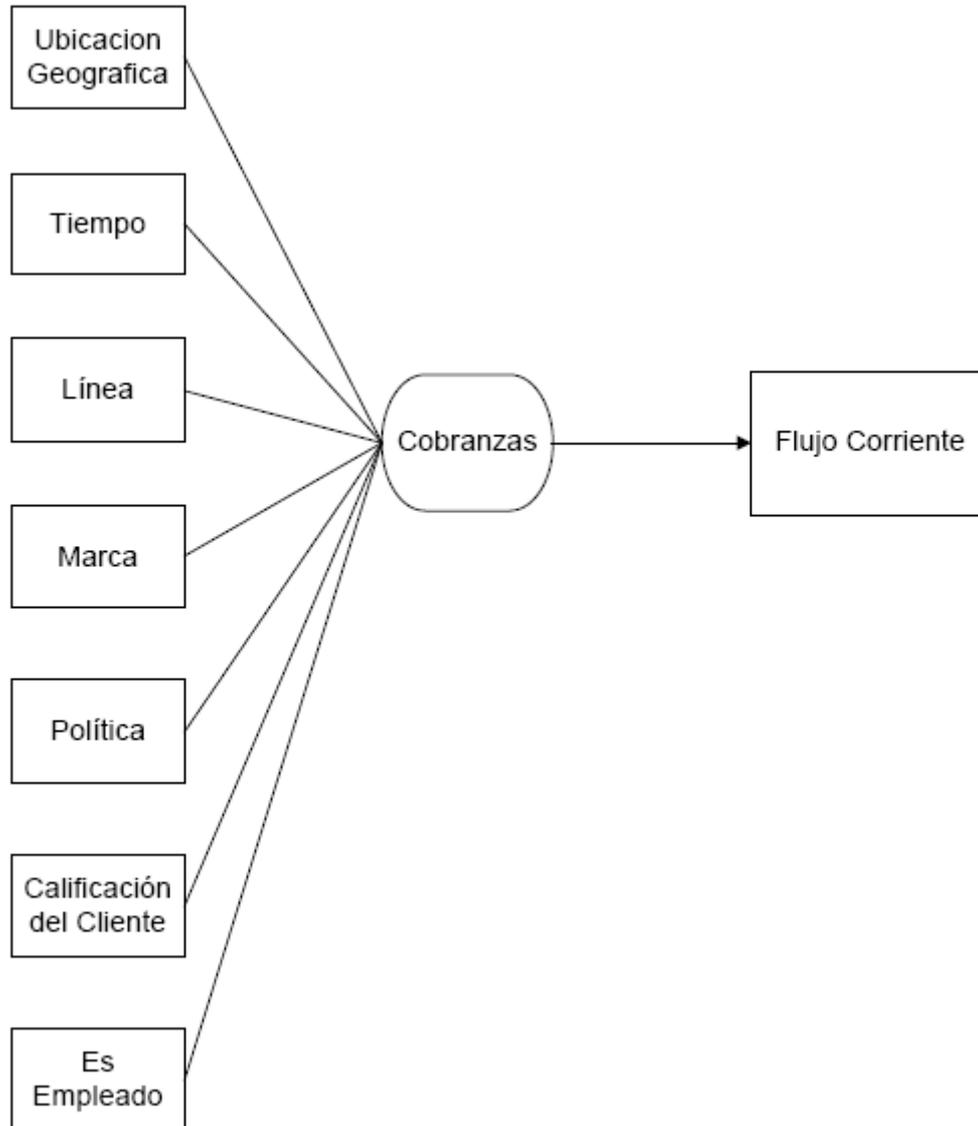
Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.

3.3.2.2 Cobranzas – Flujo corriente.

Flujo corriente, se refiere a la recuperación de la cartera de los créditos que se encuentren al día y cuya cuota se vence en el mes que se evalúa. En la Figura 2.6 graficamos el indicador de morosidad de la cartera y las siete perspectivas de análisis que va a tener el indicador.

Figura 2.6 Modelo conceptual – Flujo corriente.

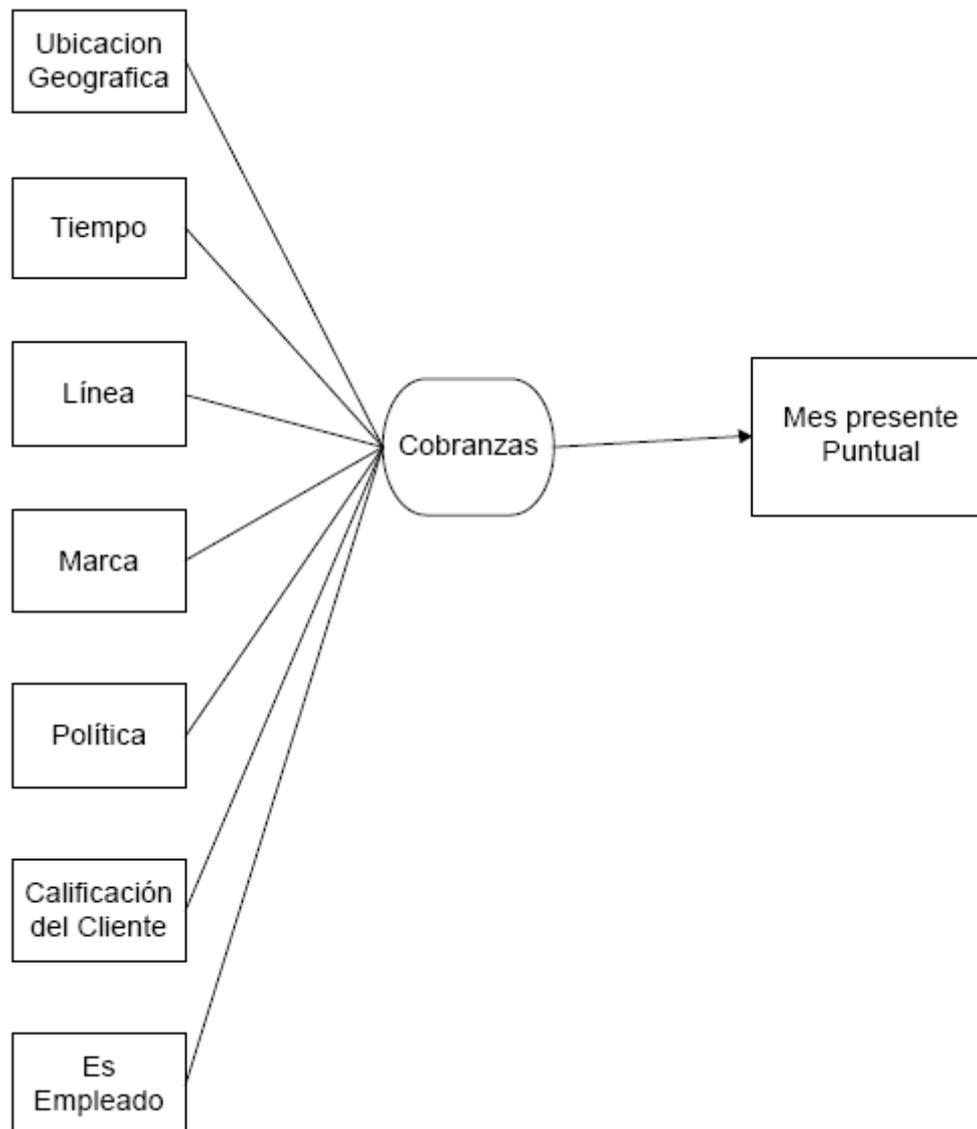


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

3.3.2.3 Cobranzas – Mes presente puntual.

Mes presente puntual, es la recuperación de la cartera de los créditos que se encuentren al día y cuya cuota se vence en el mes que se evalúa discriminando las primeras cuotas. En la Figura 2.7 graficamos el indicador del mes presente puntual y las siete perspectivas de análisis que va a tener el indicador.

Figura 2.7 Modelo conceptual – Mes presente puntual.



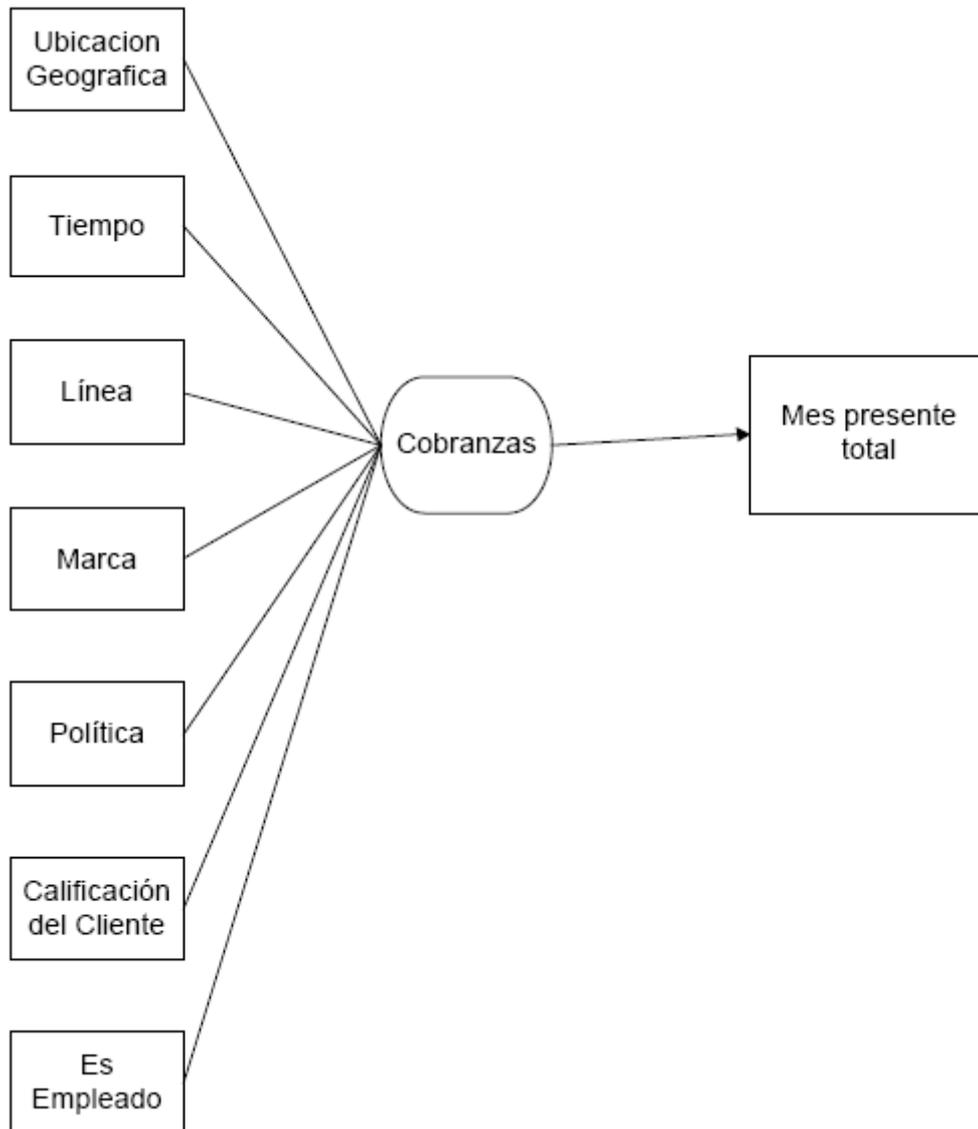
Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.

3.3.2.4 Cobranzas – Mes presente total.

Mes presente total, es la recuperación de la cartera de todos los créditos cuya cuota se vence en el mes que se evalúa. En la Figura 2.8 graficamos el indicador del mes presente total y las siete perspectivas de análisis que va a tener el indicador.

Figura 2.8 Modelo conceptual – Mes presente total.

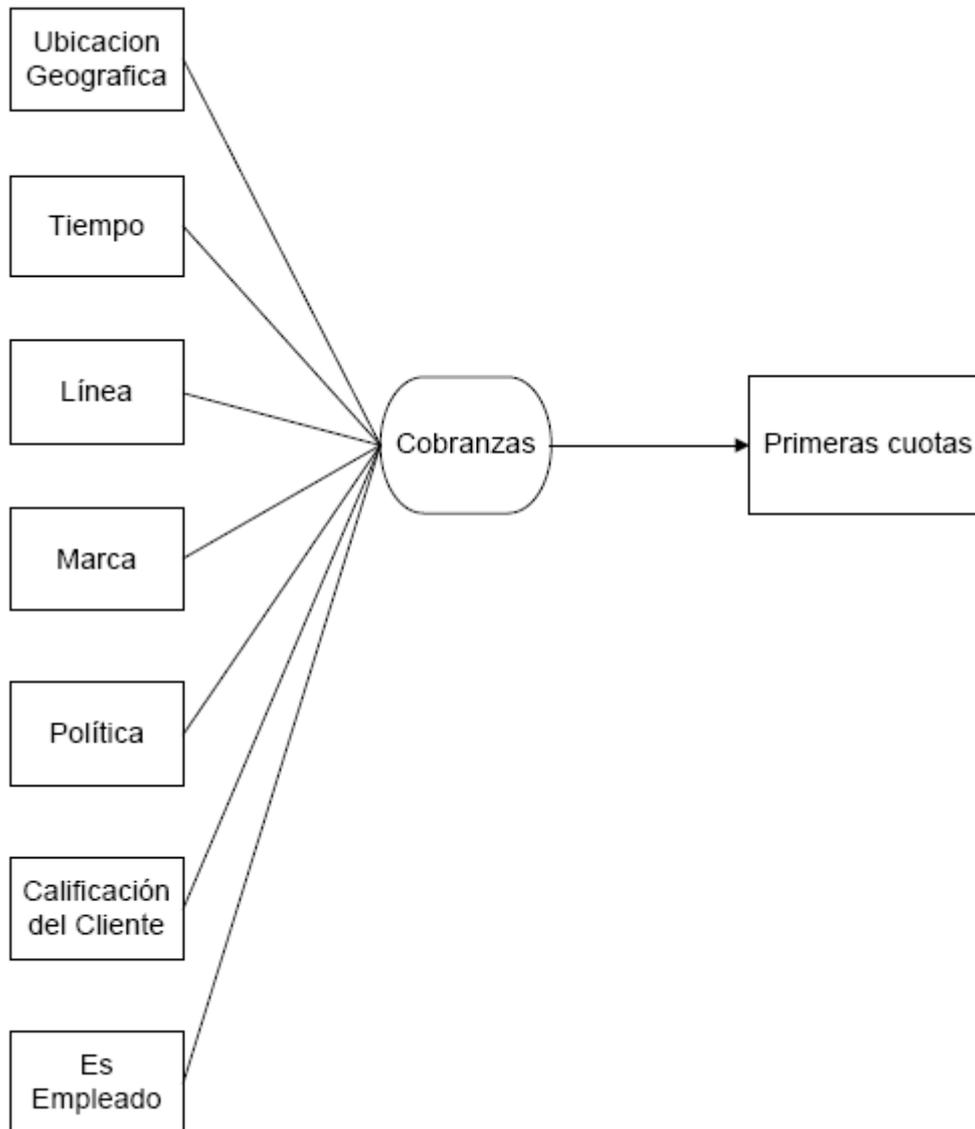


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

3.3.2.5 Cobranzas – Primeras cuotas.

Primeras cuotas, corresponde a la recuperación de la cartera de las primeras cuotas cuya fecha de vencimiento se encuentre en el mes que se evalúa. En la Figura 2.9 graficamos el indicador primeras cuotas y las siete perspectivas de análisis que va a tener el indicador.

Figura 2.9 Modelo conceptual – Primeras cuotas.



Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



CAPÍTULO 4.

ANÁLISIS DE LOS OLTP.

En este capítulo se indicará las fórmulas de cálculo para obtener los indicadores, se establecerá la correspondencia entre el modelo conceptual y las fuentes de datos, luego se definirán los campos que va a contener cada perspectiva, a continuación se desarrollará el modelo conceptual ampliado colocando bajo cada perspectiva los campos elegidos y bajo cada indicador su respectiva fórmula de cálculo.

4.1 Determinación de indicadores.

Una vez realizado el modelo conceptual desarrollado en el capítulo 3, a continuación indicaremos las fórmulas de cálculo que se utilizarán para obtener cada indicador:

4.1.1 Departamento de Ventas:

4.1.1.1 Cumplimiento de ventas, de acuerdo al modelo conceptual de la Figura 2.2 para obtener el indicador de cumplimiento de ventas la fórmula de cálculo es el siguiente:

$$((\text{Total_ventas})/\text{objetivo de ventas})*100$$

4.1.1.2 Rentabilidad, de acuerdo al modelo conceptual de la Figura 2.3 para obtener el indicador de rentabilidad se debe utilizar la siguiente fórmula de cálculo:

$$((\text{Ventasnetas} / \text{Costosnetos})-1)*100$$

4.1.2 Departamento de Cobranzas:

4.1.2.1 Morosidad de la cartera, de acuerdo al modelo conceptual de la Figura 2.5 para obtener el indicador de morosidad de la cartera se utilizará la siguiente fórmula:

$$(\text{CarteraVencida}/\text{TotalCarteraAlCierre}) *100.$$



4.1.2.2 Flujo Corriente, para obtener el indicador flujo corriente identificado en la Figura 2.6 del modelo conceptual, se utilizará la siguiente fórmula:

(Cobro efectivo del mes presente puntual/Cartera mes presente puntual)*100.

4.1.2.3 Mes presente puntual, para obtener el indicador mes presente puntual identificado en la Figura 2.7 del modelo conceptual, se utilizará la siguiente fórmula:

(Cobro efectivo del mes presente puntual/Cartera mes presente puntual)*100

4.1.2.4 Mes presente total, para obtener el indicador mes presente total identificado en la Figura 2.8 del modelo conceptual, se utilizará la siguiente fórmula:

(Cobro efectivo del mes presente puntual/Cartera mes presente puntual)*100

4.1.2.5 Primeras cuotas, para obtener el indicador primeras cuotas identificado en la Figura 2.9 del modelo conceptual, se utilizará la siguiente fórmula:

(Cobro efectivo de primeras cuotas del mes de cierre/Cartera primera cuotas del mes de cierre)*100



4.2 Establecer correspondencia con los requerimientos.

Después de haber definido las fórmulas de cálculo de cada indicador desarrollado en el sección 4.1, en esta sección estableceremos las correspondencias entre el modelo conceptual y las fuentes de datos, para conseguir este objetivo examinaremos los OLTP que contengan la información necesaria para poder establecer dicha correspondencia.

Se ha examinado los OLTP de *Impocomjajer* obteniendo las tablas transaccionales que se necesitan para establecer la correspondencia con el modelo conceptual. El diccionario de datos de las tablas transaccionales necesarias se describe en el Anexo A5.

A continuación describiremos cada correspondencia entre la perspectiva de análisis del modelo conceptual con las tablas transaccionales de *Impocomjajer*.

Correspondencia entre la Perspectiva de Análisis y las Tablas Transaccionales		
Perspectiva de Análisis	Tabla transaccional	Indicador
Ubicación Geográfica	AD_PARAMETROS_GENERALES	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
	AD_REGIONES_X_EMPRESAS	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
	AD_PROVINCIAS	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
	AD_CANTONES	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente



		Mes presente puntual Mes presente total Primeras cuotas
	TG_AGENCIA	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
	AD_AGENCIAS_X_CANTONES	Cumplimiento de ventas Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
Línea	TG_MODELO_ITEM	Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
Marca	MARCA	Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
Política	ST_POLITICA_CREDITO	Rentabilidad Morosidad de cartera Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
Calificación del Cliente	CC_BALANCES_MENSUALES	Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
Es empleado	CC_BALANCES_MENSUALES	Flujo corriente Mes presente puntual Mes presente total Primeras cuotas



Cumplimiento de Ventas	MOVIMIENTO	Cumplimiento de ventas
	VE_OBJETIVOS_X_AGENCIAS	
Producto	PRODUCTO	Rentabilidad
Tiempo	FORMA_PAGO_COMPROBANTE	Mes presente puntual Mes presente total Primeras cuotas
	COMPROBANTE	Rentabilidad
Forma Pago	COMPROBANTE	Rentabilidad
Tipo Cliente	TG_MODELO_ITEM	Rentabilidad
Rentabilidad	STA_RENTABILIDAD_TEMPORAL	Rentabilidad
Morosidad de la Cartera	CC_BALANCES_MENSUALES	Morosidad de la cartera
	CC_SIEMBRA_X_VENCIMIENTO	Rentabilidad
Valor Indicador	FORMA_PAGO_COMPROBANTE	Flujo corriente Mes presente puntual Mes presente total Primeras cuotas
	ST_CANCELACION_D	Flujo corriente Mes presente puntual Mes presente total Primeras cuotas



4.3 Nivel de granularidad.

Una vez establecido las relaciones entre las perspectivas del modelo conceptual con los OLTP, analizaremos y seleccionaremos los campos que contendrán cada perspectiva y será a través de estos que se filtrarán y manipularán los indicadores.

Para conseguir el nivel de granularidad de cada perspectiva del modelo conceptual se realizó varias jornadas de trabajo con los gerentes de ventas y cobranzas, se les expuso los datos existentes que cubran sus necesidades, explicándoles el significado de los campos y los valores que almacenan, seleccionando al final los campos que cubren sus necesidades para los indicadores seleccionados.

A continuación presentaremos el nivel de granularidad que va a tener cada perspectiva de análisis del modelo conceptual especificando la tabla de donde se va a obtener y los campos que se van a considerar y también se especificará las tablas con las que se pueden realizar unión para obtener información adicional.



4.3.1 Indicador Cumplimiento Ventas:

El indicador del cumplimiento de ventas tiene las perspectivas ubicación geográfica y tiempo, se van a obtener la información de las tablas transaccionales ad_agencias_x_cantones y movimiento y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas y ad_naciones para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.1.

Tabla 3.1 Granularidad Cumplimiento de ventas.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Tiempo	movimiento	anio_mes_venta		
Tiempo	movimiento	dia_venta		



4.3.2 Indicador Rentabilidad:

El indicador de rentabilidad tienen las perspectivas ubicación geográfica, línea, marca, producto, forma_pago, plazo, tipo de cliente y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones, cc_balances_mensuales, sta_rentabilidad_temporal y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca, producto y comprobantes para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.2.

Tabla 3.2 Granularidad Rentabilidad.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Producto	sta_rentabilidad_temporal	cod_producto, empresa	producto	nombre
Forma_pago		empresa, tipo_comprobante, cod_comprobante	comprobantes	forma_pago
Plazo		pagos		
Tipo de cliente		empresa, cod_modelo, cod_item	tg_modelo_item	nombre
Tiempo		fecha		
Tiempo	cc_balances_mensuales	periodo_cierre		



4.3.3 Indicador Morosidad de la cartera:

El indicador de morosidad de la cartera tienen las perspectivas ubicación geográfica, línea, marca, política y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones y cc_balances_mensuales y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca y st_politica crédito para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.3.

Tabla 3.3 Granularidad Morosidad de la cartera.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Política		codigo_politica, codigo_empresa	st_politica_credito	nombre
Tiempo		anio_mes_venta		
Tiempo		periodo_cierre		



4.3.4 Indicador Flujo Corriente:

El indicador flujo corriente tienen las perspectivas ubicación geográfica, línea, marca, política, calificación del cliente, es empleado y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones, cc_balances_mensuales y forma_pago_comprobante y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca, st_politica crédito y st_cat_cliente para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.4.

Tabla 3.4 Granularidad Flujo corriente.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Política		codigo_politica, codigo_empresa	st_politica_credito	nombre
Calificación del cliente		cod_cat_cliente	st_cat_cliente	nombre
Es empleado		es_empleado		
Tiempo	forma_pago_comprobante	fecha_vencimiento		
Tiempo	cc_balances_mensuales	periodo_cierre		



4.3.5 Indicador Mes presente puntual:

El indicador mes presente puntual tienen las perspectivas ubicación geográfica, línea, marca, política, calificación del cliente, es empleado y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones, cc_balances_mensuales y forma_pago_comprobante y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca, st_politica crédito y st_cat_cliente para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.5.

Tabla 3.5 Granularidad Mes presente puntual.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Política		codigo_politica, codigo_empresa	st_politica_credito	nombre
Calificación del cliente		cod_cat_cliente	st_cat_cliente	nombre
Es empleado		es_empleado		
Tiempo	forma_pago_comprobante	fecha_vencimiento		
Tiempo	cc_balances_mensuales	periodo_cierre		



4.3.6 Indicador Mes presente total:

El indicador mes presente total tienen las perspectivas ubicación geográfica, línea, marca, política, calificación del cliente, es empleado y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones, cc_balances_mensuales y forma_pago_comprobante y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca, st_politica crédito y st_cat_cliente para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.6.

Tabla 3.6 Granularidad Mes presente total.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla union OLTP	Campo Tabla union OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Política		codigo_politica, codigo_empresa	st_politica_credito	nombre
Calificación del cliente		cod_cat_cliente	st_cat_cliente	nombre
Es empleado		es_empleado		
Tiempo	forma_pago_comprobante	fecha_vencimiento		
Tiempo	cc_balances_mensuales	periodo_cierre		



4.3.7 Indicador Primeras cuotas:

El indicador primeras cuotas tienen las perspectivas ubicación geográfica, línea, marca, política, calificación del cliente, es empleado y tiempo, se va a obtener la información de las tablas transaccionales ad_agencias_x_cantones, cc_balances_mensuales y forma_pago_comprobante y se relacionan con las tablas transaccionales tg_agencia, ad_cantones, ad_provincias, ad_regiones_x_empresas, ad_naciones, tg_modelo_item, marca, st_politica crédito y st_cat_cliente para obtener los nombres y descripciones. A continuación describimos el nivel de granularidad en la Tabla 3.7.

Tabla 3.7 Granularidad Primeras cuotas.

Perspectiva	Tabla OLTP	Campo Tabla OLTP	Tabla unión OLTP	Campo Tabla unión OLTP
Ubicación geográfica	ad_agencias_x_cantones	codigo_agencia, codigo_canton, codigo_provincia, codigo_region, codigo_nacion, codigo_empresa	tg_agencia	nombre
			ad_cantones	descripción
			ad_provincias	descripción
			ad_regiones_x_empresas	descripción
			ad_naciones	descripción
Línea	cc_balances_mensuales	cod_modelo_cat_linea, cod_item_cat_linea	tg_modelo_item	nombre
Marca		codigo_marca, codigo_empresa	marca	nombre
Política		codigo_politica, codigo_empresa	st_politica_credito	nombre
Calificación del cliente		cod_cat_cliente	st_cat_cliente	nombre
Es empleado		es_empleado		
Tiempo	forma_pago_comprobante	fecha_vencimiento		
Tiempo	cc_balances_mensuales	periodo_cierre		



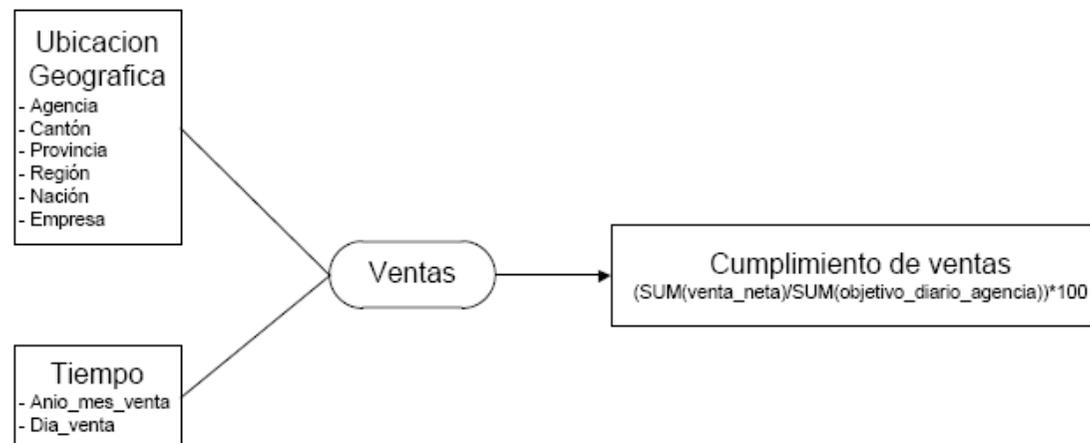
4.4 Modelo conceptual ampliado.

Después de haber definido en la sección 4.3 el nivel de granularidad de los indicadores vamos a ampliar el modelo conceptual colocando bajo cada perspectiva los campos elegidos y bajo cada indicador su respectiva fórmula de cálculo.

4.4.1 Ventas – Cumplimiento de ventas.

En la Figura 3.10 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador cumplimiento de ventas.

Figura 3.10 Modelo conceptual ampliado – Cumplimiento de ventas.



Fuente: Impocomjaher Cía. Ltda.

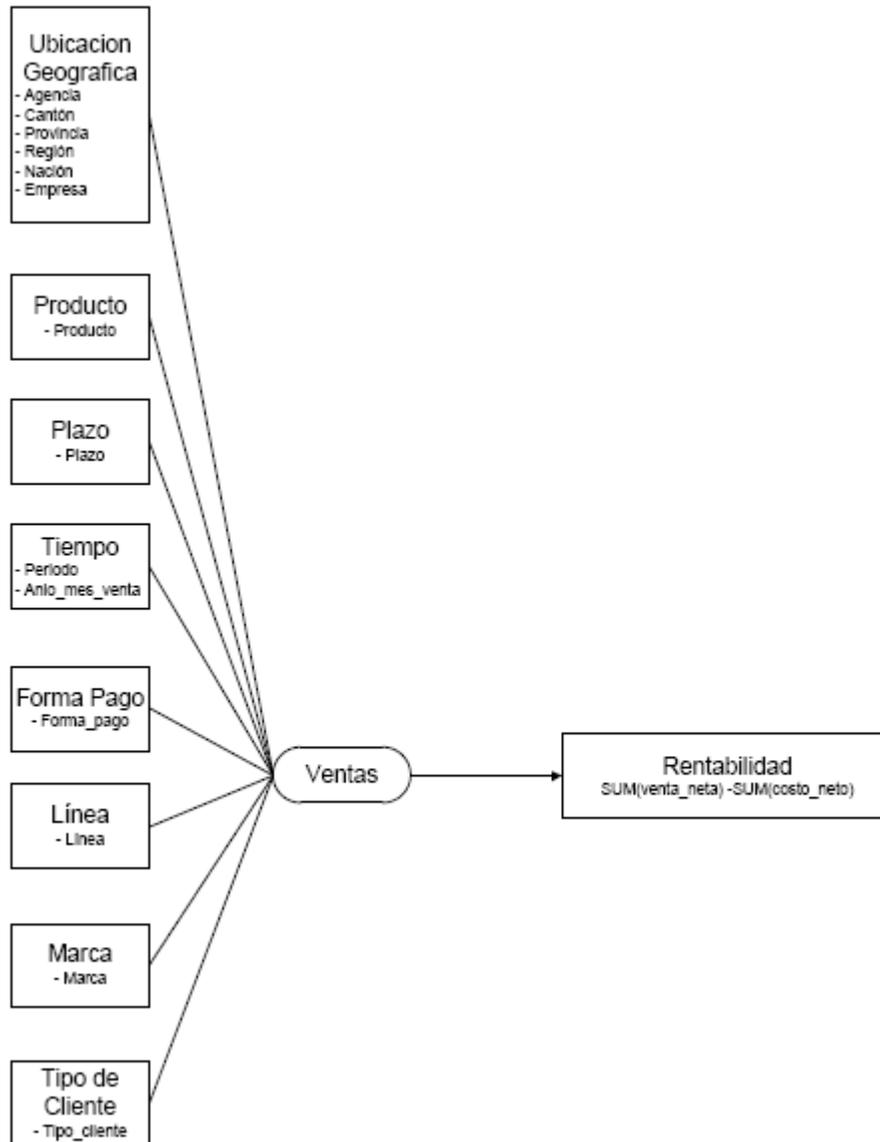
Elaboración: El autor.



4.4.2 Ventas – Rentabilidad.

En la Figura 3.11 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador rentabilidad.

Figura 3.11 Modelo conceptual ampliado – Rentabilidad.





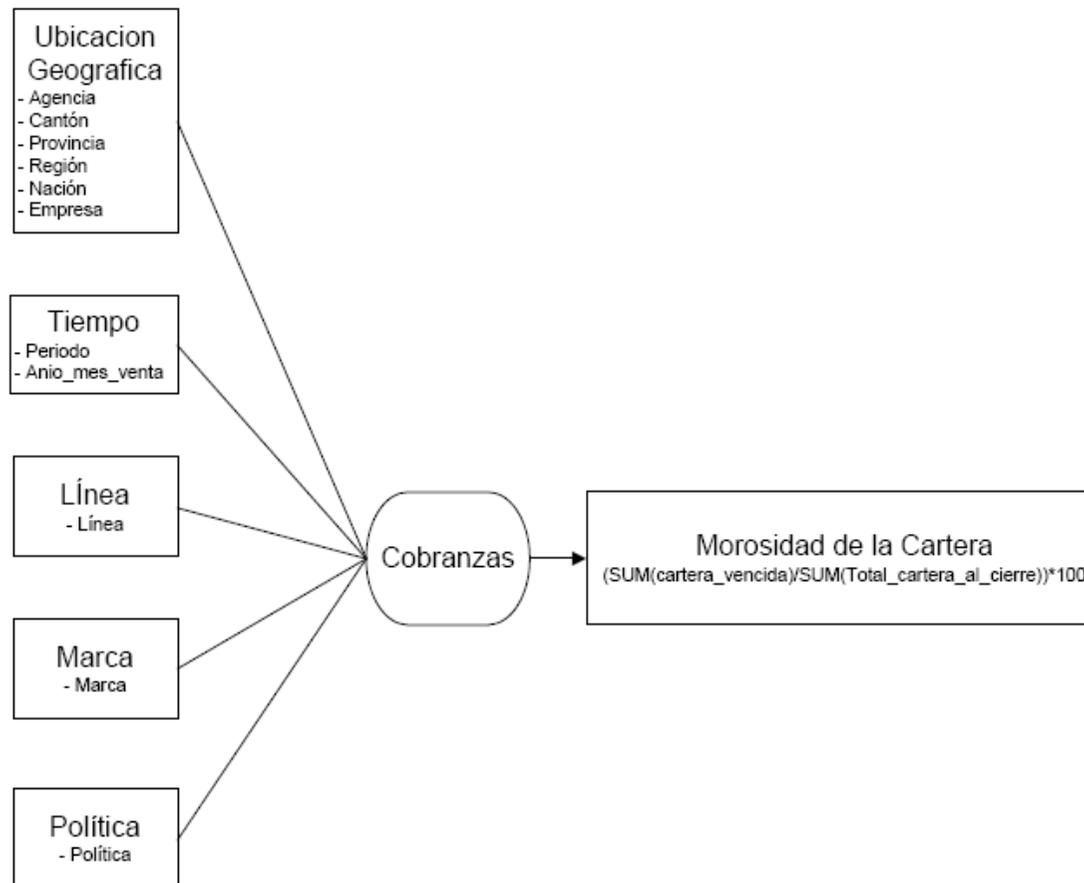
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



4.4.3 Cobranzas – Morosidad de la cartera.

En la Figura 3.12 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador morosidad de la cartera.

Figura 3.12 Modelo conceptual ampliado – Morosidad de la cartera.



Fuente: Impocomjaher Cía. Ltda.

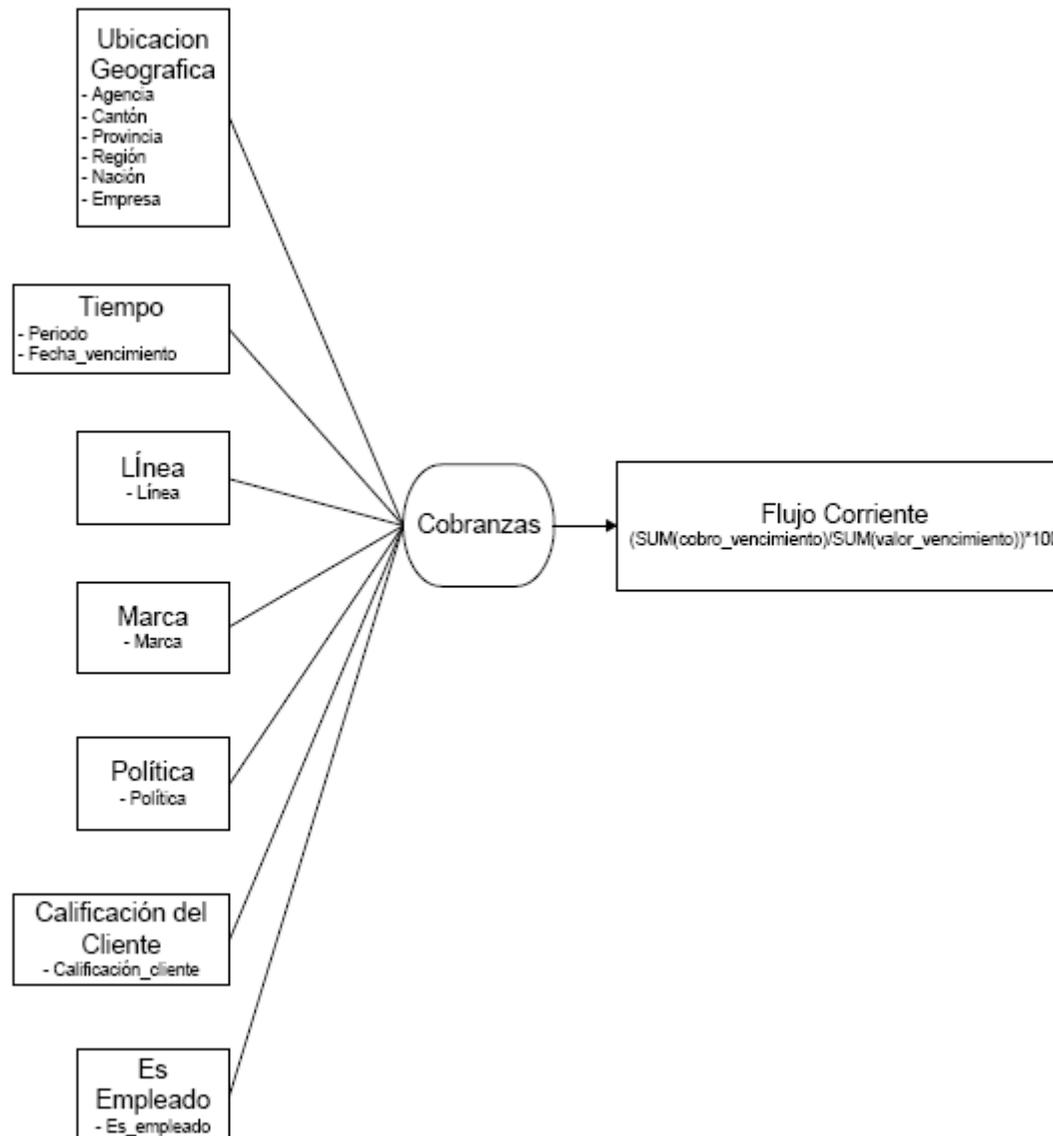
Elaboración: El autor.



4.4.4 Cobranzas – Flujo corriente.

En la Figura 3.13 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador flujo corriente.

Figura 3.13 Modelo conceptual ampliado – Flujo corriente.





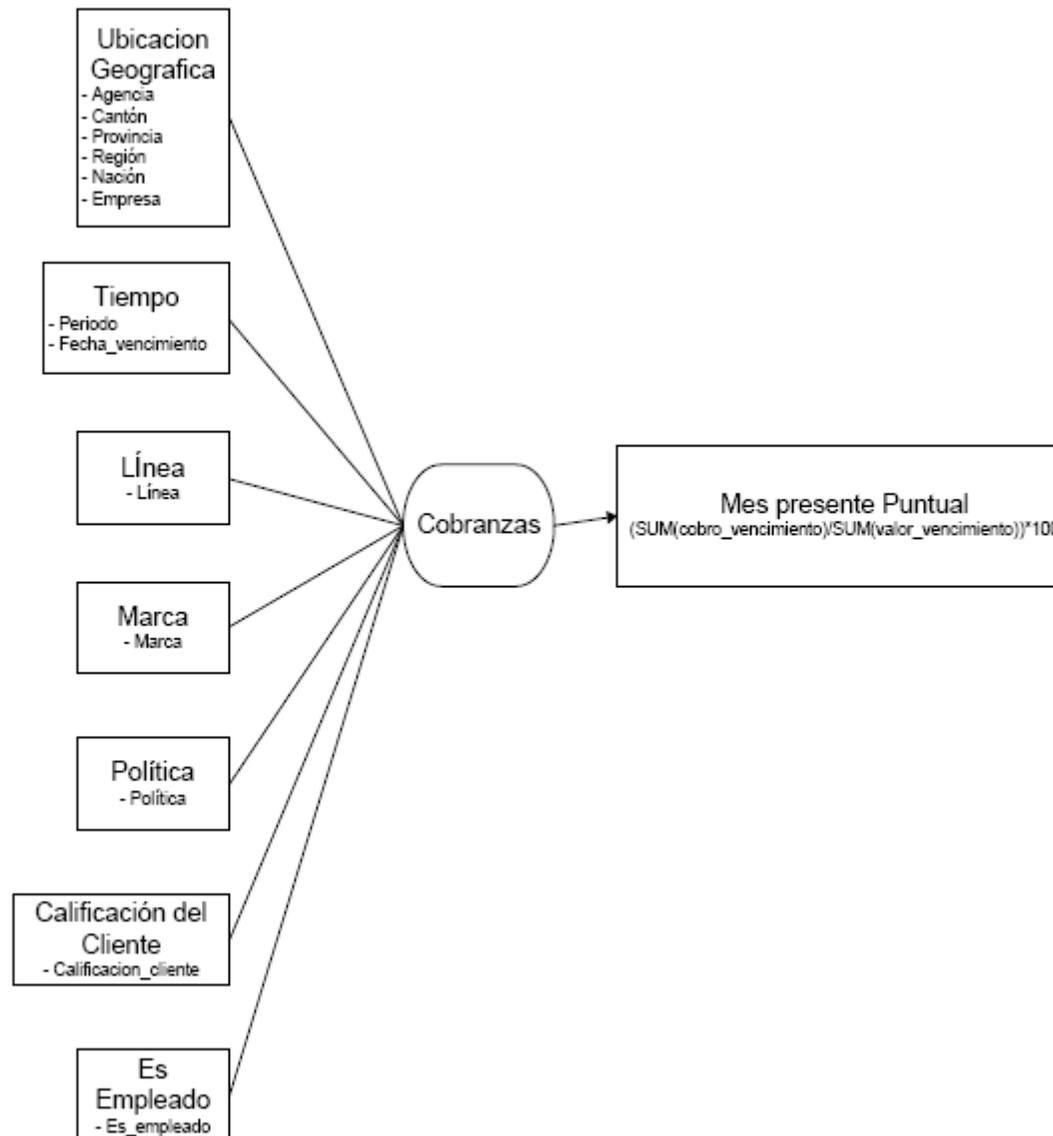
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



4.4.5 Cobranzas – Mes presente puntual.

En la Figura 3.14 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador mes presente puntual.

Figura 3.14 Modelo conceptual ampliado – Mes presente puntual.





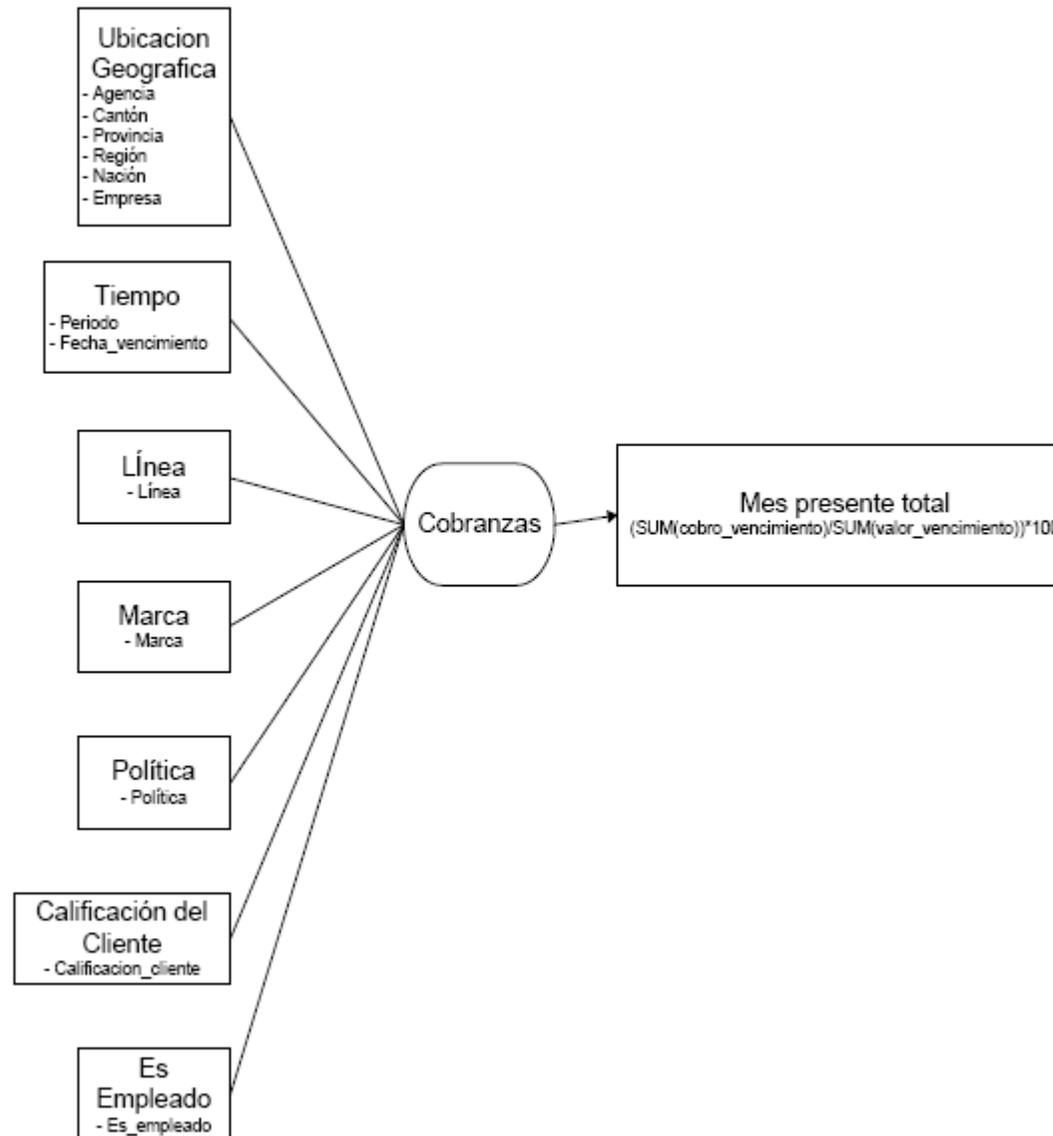
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



4.4.6 Cobranzas – Mes presente total.

En la Figura 3.15 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador mes presente total.

Figura 3.15 Modelo conceptual ampliado – Mes presente total.





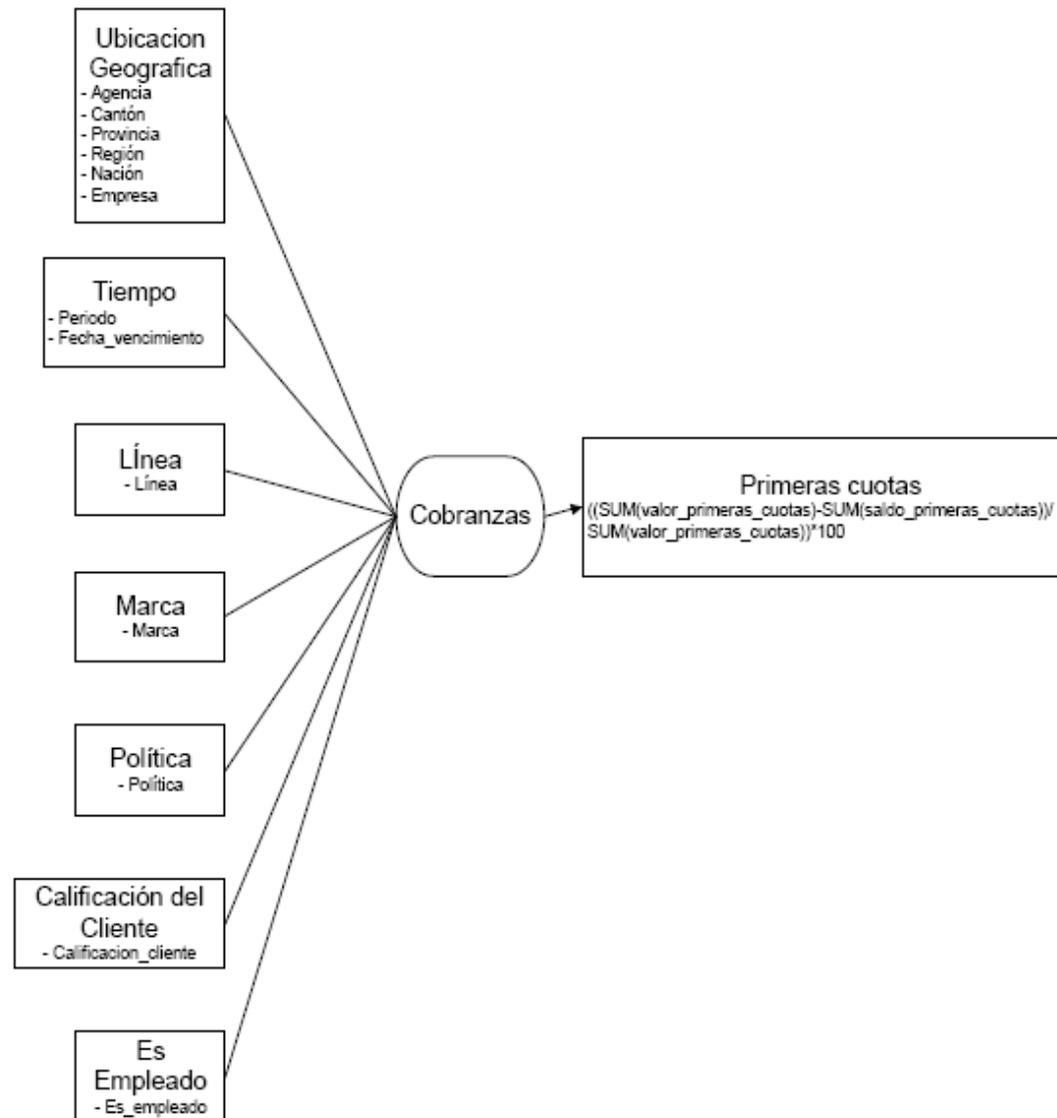
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



4.4.7 Cobranzas – Primeras cuotas.

En la Figura 3.16 especificamos los atributos que va a tener cada perspectiva y la fórmula de cálculo para el indicador primeras cuotas.

Figura 3.16 Modelo conceptual ampliado – Primeras cuotas.





Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

En este capítulo hemos creado las tablas de dimensiones, las tablas hechos y la unión entre ellas de acuerdo al modelo lógico establecido en el capítulo 4. En el siguiente capítulo vamos a crear los procesos de extracción y carga de la información para el *datamart*.



CAPÍTULO 5.

MODELO LÓGICO DE LA ESTRUCTURA DEL DATAMART.

Un vez definido en el capítulo anterior el modelo lógico con los atributos que va a tener cada perspectiva de análisis y las fórmulas de cálculo, en este capítulo procederemos a crear el modelo lógico de la estructura del datamart, tomando el modelo conceptual ampliado como base realizado en la sección 4.4 del capítulo 4. En primer lugar definiremos el tipo de modelo que utilizaremos, luego diseñaremos las tablas de dimensión, las tablas de hecho y al final realizaremos la unión entre ellas.

5.1 Tipo del modelo lógico del datamart.

Para el desarrollo del *datamart* para *Impocomjaher* se ha seleccionado el esquema constelación, porque está conformado por un grupo de estrellas y tablas de dimensiones con jerarquías.

5.1.1 Tablas de dimensiones.

Para crear las tablas de dimensiones se tomara cada perspectiva con sus respectivos atributos definido en la sección 4.4 del capítulo 4, luego se le otorgará un nombre que identifique de mejor manera a la dimensión, a continuación se le ubicarán los nombres de los campos de acuerdo a los atributos de la perspectiva, pueden ser los mismos nombres o se pueden cambiar si no son lo suficientemente intuitivos y finalmente se especificará la clave principal de la tabla de dimensión. Este proceso se realizará con todas y cada una de las perspectivas de análisis.

El resultado del modelo lógico de las tablas de dimensiones se encuentra con mayor detalle en el Anexo A6.

Hemos creado una tabla resumen que se encuentra estructurado de la siguiente manera:

Departamento: corresponde al departamento analizado.



Indicador: nombre del indicador a ser creado.

Ver figura: hace referencia al gráfico que puede ser consultado con mayor detalle como resultado de este proceso.

Perspectiva de análisis: se refiere a los objetos mediante los cuales se quiere examinar los indicadores. El concepto de perspectiva de análisis se encuentra en el capítulo 3 sección 3.2.

Tabla Dimensión: corresponde a las tablas de dimensiones que van a ser creadas.



A continuación presentamos la tabla resumen:

Tablas de Dimensiones				
Departamento	Indicador	Ver Figura	Perspectiva de análisis	Tabla Dimensión
Ventas	Cumplimiento de Ventas	Anexo A6. Figura 4.24	Ubicación Geográfica	ad_agencias_x_cantones
	Rentabilidad	Anexo A6. Figura 4.26		ad_cantones_x_provincia
Cobranzas	Morosidad de la Cartera	Anexo A6. Figura 4.34		ad_provincias_x_regiones
	Flujo Corriente	Anexo A6. Figura 4.39		ad_regiones_x_empresas
	Mes Presente Puntual	Anexo A6. Figura 4.46		ad_provincias
	Mes Presente Total	Anexo A6. Figura 4.53		ad_naciones
	Primeras Cuotas	Anexo A6. Figura 4.60		ad_empresas
Ventas	Cumplimiento de Ventas	Anexo A6. Figura 4.25	Tiempo	dm_fechas_ventas
Ventas	Rentabilidad	Anexo A6. Figura 4.27	Producto	dm_productos
Ventas	Rentabilidad	Anexo A6. Figura 4.28	Plazo	dm_plazos_ventas
Ventas	Rentabilidad	Anexo A6. Figura 4.29	Tiempo	dm_anio_mes_ventas
Cartera	Morosidad de la Cartera	Anexo A6. Figura 4.35		
Ventas	Rentabilidad	Anexo A6. Figura 4.29	Tiempo	dm_periodos_balances
Cartera	Morosidad de la Cartera	Anexo A6. Figura 4.35		
	Flujo Corriente	Anexo A6. Figura 4.40		
	Mes Presente Puntual	Anexo A6. Figura 4.47		
	Mes Presente Total	Anexo A6. Figura 4.54		
	Primeras Cuotas	Anexo A6. Figura 4.61		
Ventas	Rentabilidad	Anexo A6. Figura 4.30	Forma Pago	dm_forma_pago



Ventas	Rentabilidad	Anexo A6. Figura 4.31		
Cobranzas	Morosidad de la Cartera	Anexo A6. Figura 4.36	Linea	dm_lineas
	Flujo Corriente	Anexo A6. Figura 4.41		
	Mes Presente Puntual	Anexo A6. Figura 4.48		
	Mes Presente Total	Anexo A6. Figura 4.55		
	Primeras Cuotas	Anexo A6. Figura 4.62		
Ventas	Rentabilidad	Anexo A6. Figura 4.32		
Cobranzas	Morosidad de la Cartera	Anexo A6. Figura 4.37	Marca	dm_marcas
	Flujo Corriente	Anexo A6. Figura 4.42		
	Mes Presente Puntual	Anexo A6. Figura 4.49		
	Mes Presente Total	Anexo A6. Figura 4.56		
	Primeras Cuotas	Anexo A6. Figura 4.63		
Ventas	Rentabilidad	Anexo A6. Figura 4.33	Tipo de Cliente	dm_tipo_cliente
Cobranzas	Morosidad de la Cartera	Anexo A6. Figura 4.38	Politica	dm_politicas
	Flujo Corriente	Anexo A6. Figura 4.43		
	Mes Presente Puntual	Anexo A6. Figura 4.50		
	Mes Presente Total	Anexo A6. Figura 4.57		
	Primeras Cuotas	Anexo A6. Figura 4.64		
Cobranzas	Flujo Corriente	Anexo A6. Figura 4.40	Tiempo	dm_fechas_vencimientos
	Mes Presente Puntual	Anexo A6. Figura 4.47		
	Mes Presente Total	Anexo A6. Figura 4.54		



	Primeras Cuotas	Anexo A6. Figura 4.61		
Cobranzas	Flujo Corriente	Anexo A6. Figura 4.44	Calificacion del Cliente	dm_calificacion_del_cliente
	Mes Presente Puntual	Anexo A6. Figura 4.51		
	Mes Presente Total	Anexo A6. Figura 4.58		
	Primeras Cuotas	Anexo A6. Figura 4.65		
Cobranzas	Flujo Corriente	Anexo A6. Figura 4.45	Es_Empleado	dm_es_empleado
	Mes Presente Puntual	Anexo A6. Figura 4.52		
	Mes Presente Total	Anexo A6. Figura 4.59		
	Primeras Cuotas	Anexo A6. Figura 4.66		



5.1.2 Tablas de hechos.

Después de haber definido las dimensiones en la sección 5.1.1 del capítulo 5 se procederá a definir las tablas de hechos a través de los cuales se construirá los indicadores, se comenzará asignando un nombre, este nombre debe representar la información que se analiza o el área de investigación, luego se definirá la clave primaria que va a estar compuesto por las llaves primarias de las tablas de dimensiones que se relaciona y finalmente se crearán los campos tantos como se hayan definido en el modelo conceptual ampliado.

5.1.2.1 Cumplimiento de ventas.

El cumplimiento de ventas definido en el modelo conceptual Figura 3.10 lo llamaremos `dm_indice_cumplimiento_ventas`, va a estar relacionado con las dimensiones definido en la Figura 4.24 y Figura 4.25.

5.1.2.2 Rentabilidad.

La rentabilidad definido en el modelo conceptual Figura 3.11 lo llamaremos `dm_indice_rentabilidad`, este va a estar relacionado con las dimensiones definido desde la



Figura 4.26 hasta la Figura 4.33.

5.1.2.3 Morosidad de la cartera.

La morosidad de la cartera definido en el modelo conceptual Figura 3.12 lo llamaremos `dm_indice_de_morosidad`, este va a estar relacionado con las dimensiones definido desde la Figura 4.34 hasta la Figura 4.38.

5.1.2.4 Flujo corriente.

El flujo corriente definido en el modelo conceptual Figura 3.13 lo llamaremos `cc_flujo_corriente`, este va a estar relacionado con las dimensiones definido desde la Figura 4.39 hasta la Figura 4.45.

5.1.2.5 Mes presente puntual.

El mes presente puntual definido en el modelo conceptual Figura 3.14 lo llamaremos `cc_produc_mes_presente_puntual`, este va a estar relacionado con las dimensiones definido desde la Figura 4.46 hasta la Figura 4.52.

5.1.2.6 Mes presente total.

El mes presente total definido en el modelo conceptual Figura 3.15 lo llamaremos `cc_produc_mes_presente_total`, este va a estar relacionado con las dimensiones definido desde la Figura 4.53 hasta la Figura 4.59.

5.1.2.7 Primeras cuotas.

Las primeras cuotas definido en el modelo conceptual Figura 3.16 lo llamaremos `cc_primeras_cuotas`, este va a estar relacionado con las dimensiones definido desde la Figura 4.60 hasta la Figura 4.66.

Presentamos a continuación la tabla resumen de la definición de las tablas de hechos.

Tablas de Hechos			
Departamento	Indicador	Tabla de Hecho	Ver Figura
Ventas	Cumplimiento de Ventas	<code>dm_indice_cumplimiento_ventas</code>	Anexo A7. Figura 4.67
	Rentabilidad	<code>dm_indice_rentabilidad</code>	Anexo A7. Figura 4.68
Cobranzas	Morosidad de la Cartera	<code>dm_indice_de_morosidad</code>	Anexo A7. Figura 4.69
	Flujo Corriente	<code>cc_flujo_corriente</code>	Anexo A7. Figura 4.70
	Mes Presente Puntual	<code>cc_produc_mes_presente_puntual</code>	Anexo A7. Figura 4.71
	Mes Presente Total	<code>cc_produc_mes_presente_total</code>	Anexo A7. Figura 4.72
	Primeras Cuotas	<code>cc_primeras_cuotas</code>	Anexo A7. Figura 4.73

La definición de las tablas de hechos representadas en forma gráfica pueden ser consultadas en el Anexo A7.





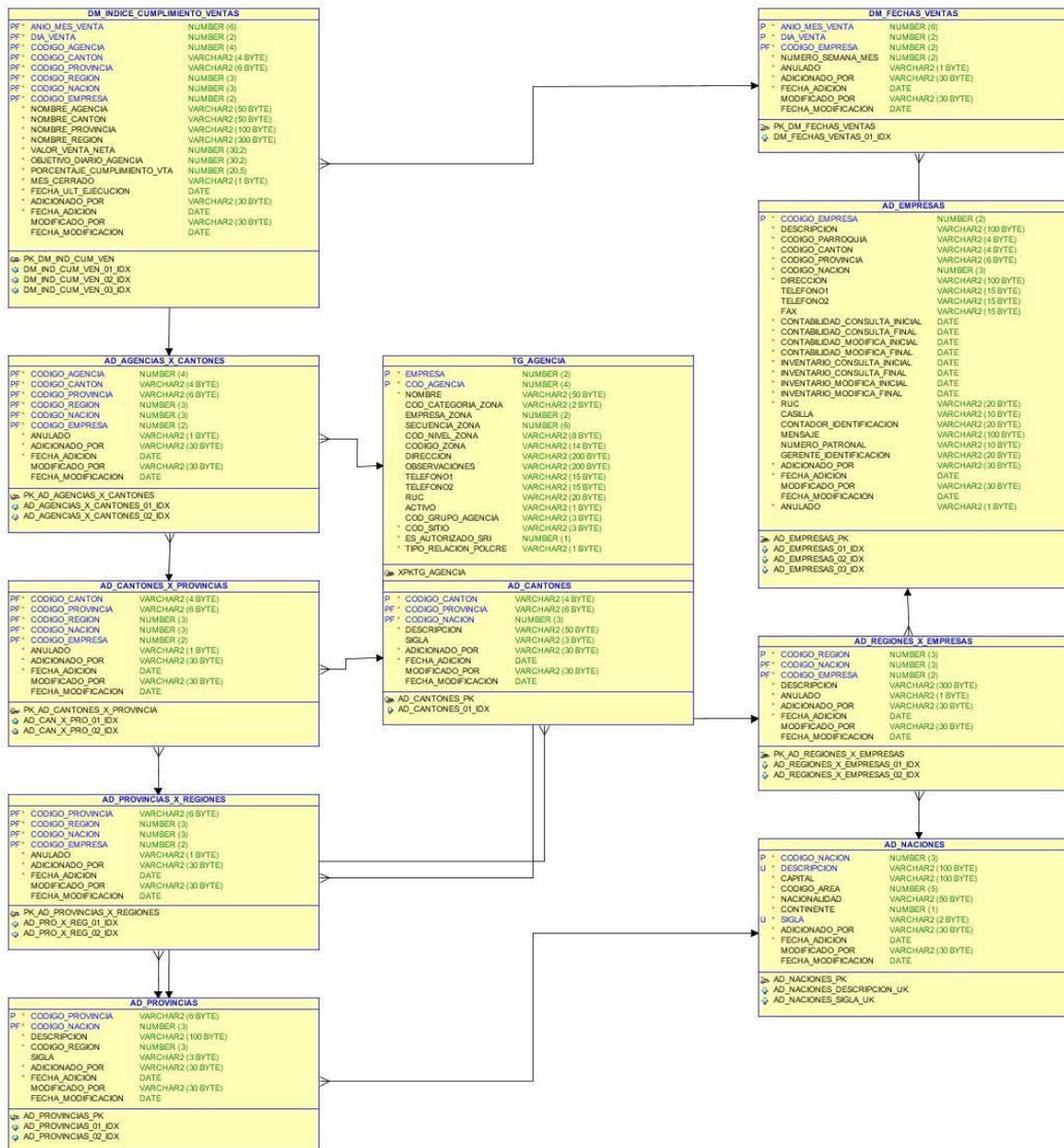
5.1.3 Uniones.

A continuación realizaremos las uniones entre las tablas de dimensiones y las tablas de hechos de acuerdo a sus correspondencias.

5.1.3.1 Cumplimiento de ventas.

En la Figura 4.17 presentamos el resultado de la unión del indicador del cumplimiento de ventas con sus dimensiones.

Figura 4.17 Uniones – Cumplimiento de ventas.



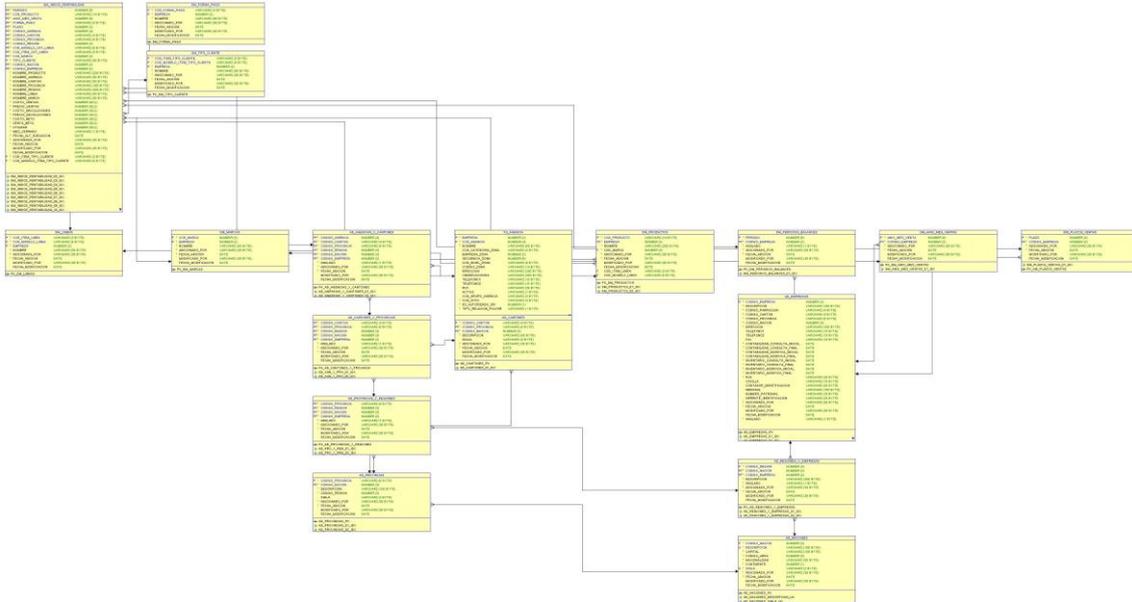
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



5.1.3.2 Rentabilidad.

En la Figura 4.18 presentamos el resultado de la unión del indicador de rentabilidad con sus dimensiones.

Figura 4.18 Uniones – Rentabilidad.



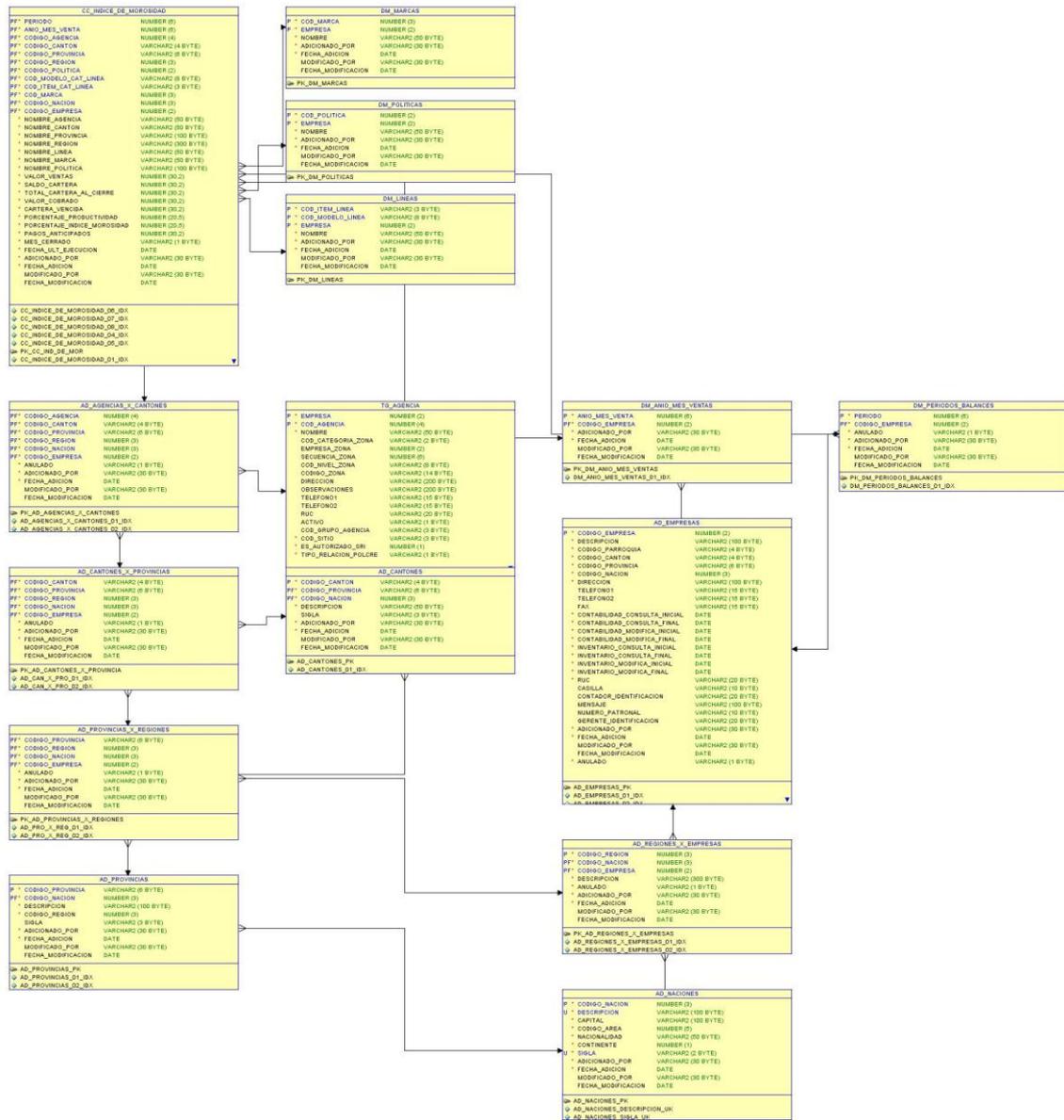
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



5.1.3.3 Morosidad de la cartera.

En la Figura 4.19 presentamos el resultado de la unión del indicador de morosidad de la cartera con sus dimensiones.

Figura 4.19 Uniones – Morosidad de la cartera.

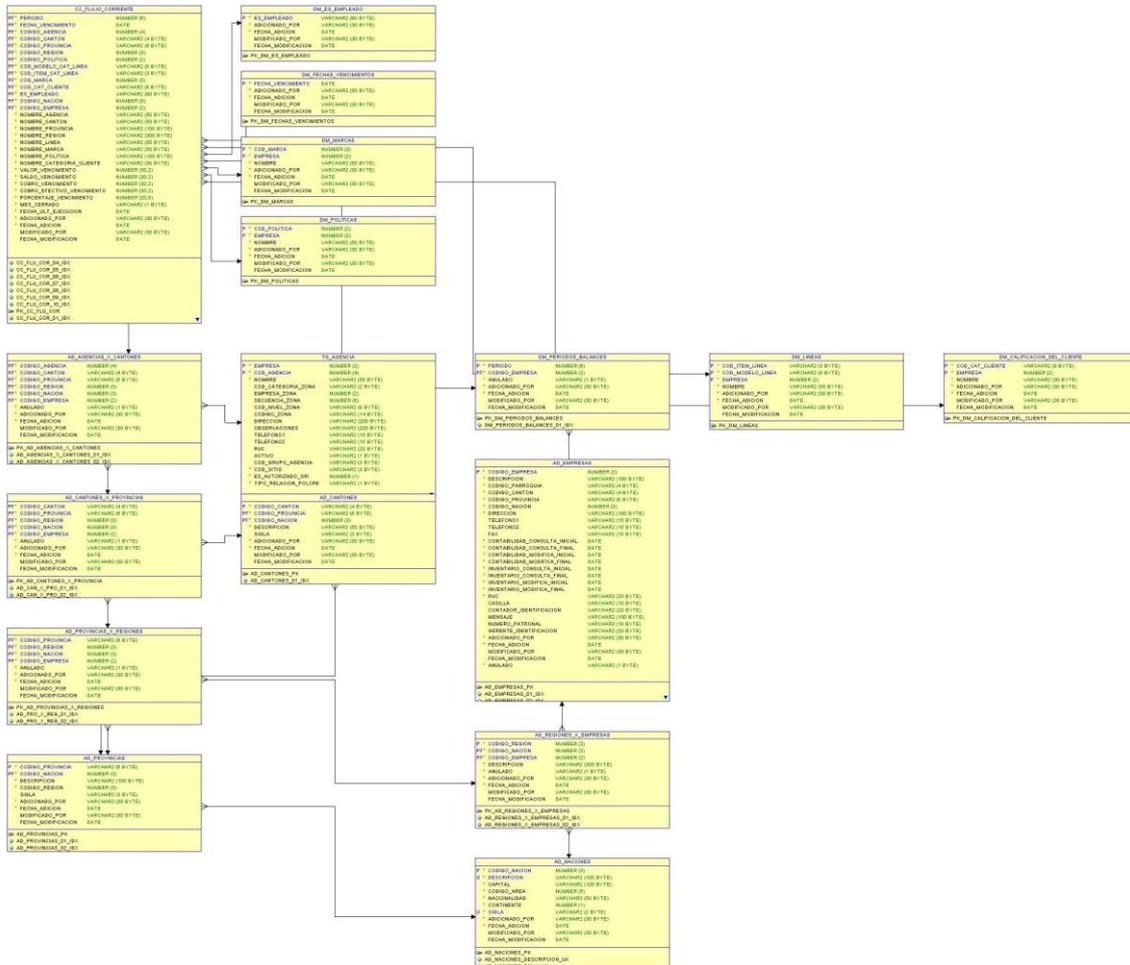


Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.

5.1.3.4 Flujo corriente.

En la Figura 4.20 presentamos el resultado de la unión del indicador de rentabilidad con sus dimensiones.

Figura 4.20 Uniones – Flujo corriente.

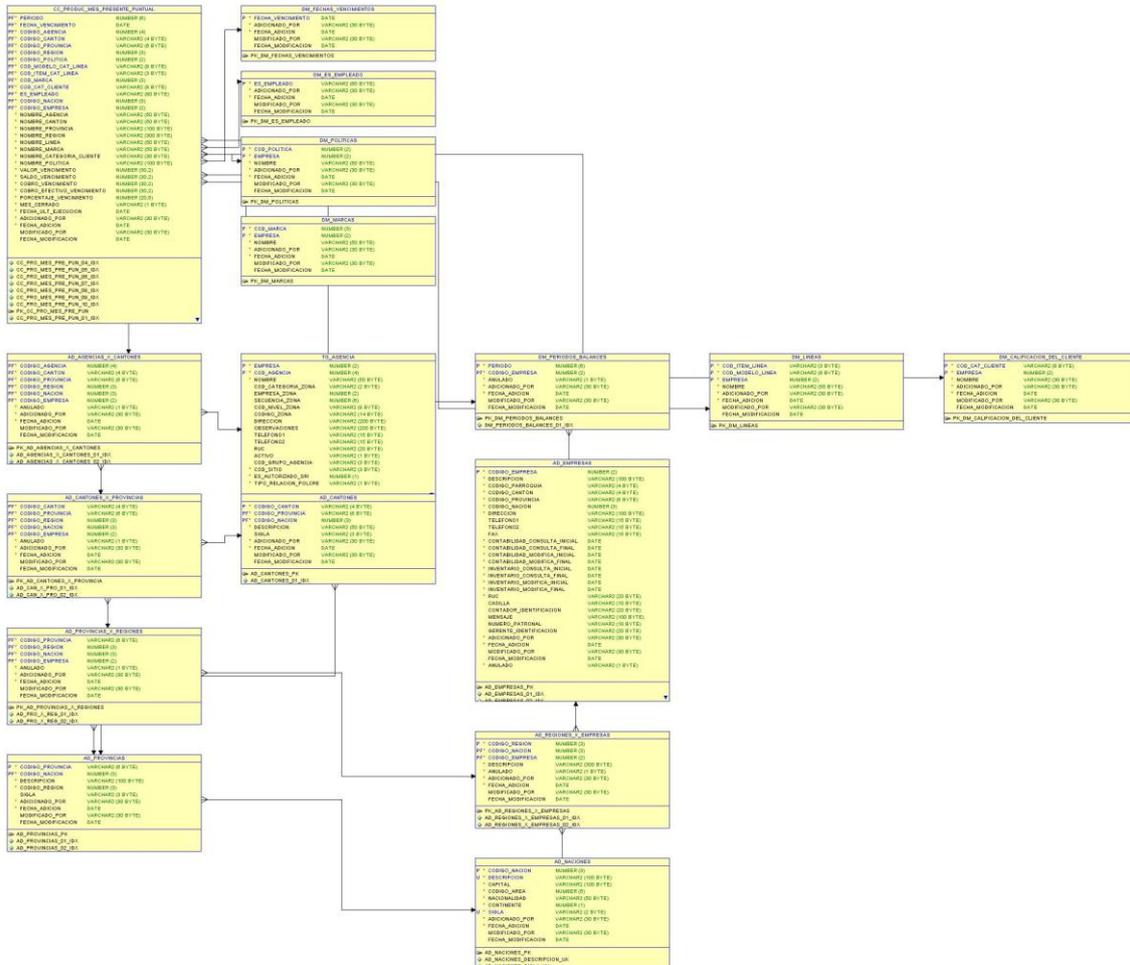


Fuente: Impocomjajer Cía. Ltda.
Elaboración: El autor.

5.1.3.5 Mes presente puntual.

En la Figura 4.21 presentamos el resultado de la unión del indicador del mes presente puntual con sus dimensiones.

Figura 4.21 Uniones – Mes presente puntual.



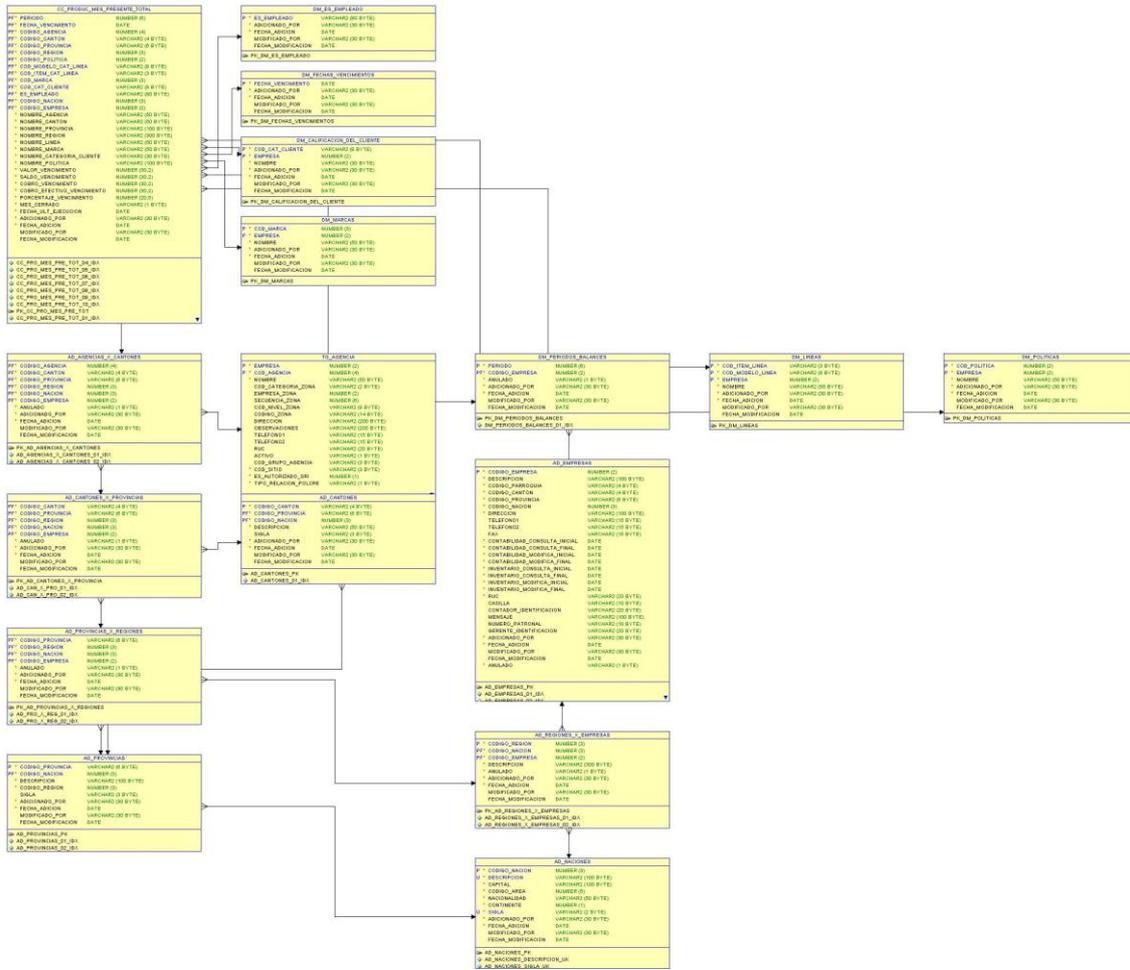
Fuente: Impocomjher Cía. Ltda.
Elaboración: El autor.



5.1.3.6 Mes presente total.

En la Figura 4.22 presentamos el resultado de la unión del indicador del mes presente total con sus dimensiones.

Figura 4.22 Uniones – Mes presente total.



Fuente: Impocomjher Cía. Ltda.
Elaboración: El autor.



CAPÍTULO 6.

INTEGRACIÓN DE DATOS.

En este capítulo realizaremos los procesos ETL para extraer los datos de los sistemas OLTP, para lo cual crearemos los procesos PISql¹ mediante los cuales extraeremos la información. Al momento de generar los ETL se considerará todas las condiciones y restricciones para almacenar la información en el *datamart*.

6.1 Carga inicial.

Para realizar la carga inicial del *datamart* se ha creado procesos ETL para cada una de las tablas de hecho y las tablas de dimensiones.

6.1.1 Tabla de dimensión dm_periodo_balances.

Para la carga de tabla de dimensiones DM_PERIODO_BALANCES, se ha tomado como fuente la tabla CC_BALANCES_MENSUALES (descrito en la Tabla 3.25) del OLTP agrupado por períodos. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A8.

6.1.2 Tabla de dimensión dm_anio_mes_ventas.

Para la carga de tabla de dimensiones DM_ANIO_MES_VENTAS, se ha tomado como fuente la tabla COMPROBANTE (descrito en la Tabla 3.22) del OLTP agrupado por año mes de venta y solo los tipos de comprobantes A0, NV, DB y D0. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A9.

6.1.3 Tabla de dimensión dm_plazos_ventas.

Para la carga de tabla de dimensiones DM_PLAZOS_VENTAS, se ha tomado como fuente la tabla COMPROBANTE (descrito en la Tabla 3.22) del OLTP agrupado por número de pago y empresa y solo los tipos de comprobantes A0, NV, DB y D0. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A10.

¹ PL/Sql (Procedural Language/Structured Query Language). Es un lenguaje de programación incrustado en Oracle. El PL/SQL es un lenguaje que permite definir procedimientos utilizando sentencias SQL.



6.1.4 Tabla de dimensión dm_lineas.

Para la carga de tabla de dimensiones DM_LINEAS, se ha tomado como fuente todas las líneas de la empresa que se encuentran en la tabla TG_MODELO_ITEM (descrito en la Tabla 3.19) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A11.

6.1.5 Tabla de dimensión dm_marcas.

Para la carga de tabla de dimensiones DM_MARCAS, se ha tomado como fuente la tabla MARCA (descrito en la Tabla 3.20) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A12.

6.1.6 Tabla de dimensión dm_productos.

Para la carga de tabla de dimensiones DM_PRODUCTOS, se ha tomado como fuente la tabla PRODUCTO (descrito en la Tabla 3.21) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A13.

6.1.7 Tabla de dimensión dm_forma_pago.

Para la carga de tabla de dimensiones DM_FORMA_PAGO, se ha tomado como fuente la tabla FORMA_PAGO (descrito en la Tabla 3.30) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A14.

6.1.8 Tabla de dimensión dm_tipo_cliente.

Para la carga de tabla de dimensiones DM_TIPO_CLIENTE, se ha tomado como fuente la tabla TG_MODELO_ITEM (descrito en la Tabla 3.19) del OLTP filtra los registros cuya campo COD_MODELO sea igual a CLI1. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A15.

6.1.9 Tabla de dimensión dm_politicas.

Para la carga de tabla de dimensiones DM_POLITICAS, se ha tomado como fuente la tabla ST_POLITICA_CREDITO (descrito en la Tabla 3.24) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A16.



6.1.10 Tabla de dimensión dm_calificacion_del_cliente.

Para la carga de tabla de dimensiones DM_CALIFICACION_DEL_CLIENTE, se ha tomado como fuente la tabla ST_CAT_CLIENTE del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A17.

6.1.11 Tabla de dimensión dm_es_employado.

Para la carga de tabla de dimensiones DM_ES_EMPLEADO, se ha tomado como fuente la tabla ST_EMPLEADOS del OLTP agrupados por nombre de la empresa. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A18.

6.1.12 Tabla de dimensión dm_fechas_vencimientos.

Para la carga de tabla de dimensiones DM_FECHAS_VENCIMIENTOS, se ha tomado como fuente la tabla FORMA_PAGO_COMPROBANTE del OLTP agrupados por la fecha de vencimiento. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A19.

6.1.13 Tabla de dimensión dm_fechas_ventas.

Para la carga de tabla de dimensiones DM_FECHAS_VENTAS, se ha tomado como fuente la tabla COMPROBANTE del OLTP agrupados por año mes de venta, día de venta y código de la empresa. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A20.

6.1.14 Tabla de dimensión ad_provincias.

Para la carga de tabla de dimensiones AD_PROVINCIAS será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_PROVINCIAS.

6.1.15 Tabla de dimensión ad_cantones.

Para la carga de tabla de dimensiones AD_CANTONES, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_CANTONES.

6.1.16 Tabla de dimensión ad_agencias_x_cantones.

Para la carga de tabla de dimensiones AD_AGENCIAS_X_CANTONES, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_AGENCIAS_X_CANTONES, esta creación lo realizará el área de tecnología porque debe ubicarse en que cantón del país se ubica la nueva agencia.



6.1.17 Tabla de dimensión ad_cantones_x_provincias.

Para la carga de tabla de dimensiones AD_CANTONES_X_PROVINCIAS, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_CANTONES_X_PROVINCIAS, esta creación lo realizará el área de tecnología porque debe ubicarse en que provincias del país se ubica el nuevo cantón.

6.1.18 Tabla de dimensión ad_provincias_x_regiones.

Para la carga de tabla de dimensiones AD_PROVINCIAS_X_REGIONES, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_PROVINCIAS_X_REGIONES, esta creación lo realizará el área de tecnología porque debe ubicarse en qué región del país se ubica la nueva provincia.

6.1.19 Tabla de dimensión ad_regiones_x_empresas.

Para la carga de tabla de dimensiones AD_REGIONES_X_EMPRESAS, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_REGIONES_X_EMPRESAS, esta creación lo realizará el área de tecnología porque debe ubicarse en qué empresa se ubica la nueva región.

6.1.20 Tabla de dimensión ad_naciones.

Para la carga de tabla de dimensiones AD_NACIONES, esto será realizado por el área de tecnología, se lo realizará cada vez que se necesite crear un nuevo registro en la tabla AD_NACIONES.

6.1.21 Tabla de dimensión ad_empresas.

Para la carga de tabla de dimensiones AD_EMPRESAS, se ha tomado como fuente la tabla EMPRESA (descrito en la Tabla 3.31) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A21.

6.1.22 Tabla de dimensión tg_agencia.

Para la carga de tabla de dimensiones TG_AGENCIA, se ha tomado como fuente la tabla AD_AGENCIAS (descrito en Tabla 3.32) del OLTP. Para automatizar el proceso de carga se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A22.

6.1.23 Tabla de hechos dm_indice_cumplimiento_ventas.

Para automatizar el proceso de la carga inicial de la tabla DM_INDICE_CUMPLIMIENTO_VENTAS, se ha implementado un script. La



descripción de las acciones que contempla este script puede ser consultado en el Anexo A23.

6.1.24 Tabla de hechos dm_indice_rentabilidad.

Para automatizar el proceso de la carga inicial de la tabla DM_INDICE_RENTABILIDAD, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A24.

6.1.25 Tabla de hechos cc_indice_de_morosidad.

Para automatizar el proceso de la carga inicial de la tabla CC_INDICE_DE_MOROSIDAD, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A25.

6.1.26 Tabla de hechos cc_flujo_corriente.

Para automatizar el proceso de la carga inicial de la tabla CC_FLUJO_CORRIENTE, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A26.

6.1.27 Tabla de hechos cc_produc_mes_presente_puntual.

Para automatizar el proceso de la carga inicial de la tabla CC_PRODUC_MES_PRESENTE_PUNTUAL, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A27.

6.1.28 Tabla de hechos cc_produc_mes_presente_total.

Para automatizar el proceso de la carga inicial de la tabla CC_PRODUC_MES_PRESENTE_TOTAL, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A28.

6.1.29 Tabla de hechos cc_primeras_cuotas.

Para automatizar el proceso de la carga inicial de la tabla CC_PRIMERAS_CUOTAS, se ha implementado un script. La descripción de las acciones que contempla este script puede ser consultado en el Anexo A29.



6.2 Actualización de datos.

Para realizar la actualización de datos del *datamart* se ha creado procesos ETL para cada uno de los indicadores y las tablas de dimensiones.

6.2.1 Tablas de dimensión.

Todas las tablas de dimensiones se actualizarán en el procedimiento almacenado llamado PK_DM_PROCESOS_TB_DIMENSIONES y este procedimiento almacenado se ejecutará en un job de la base de datos al cierre de cada mes. La descripción de la acción que contempla el procedimiento almacenado puede ser consultada en el Anexo A30.

6.2.1.1 Indicador Cumplimiento de Ventas.

El indicador cumplimiento de ventas se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado PK_CC_PROCESO_IND_CUMPLIM_VTA. P_CARGAR_IND_CUMVTA_CIERRE_MES y el segundo proceso actualizará diariamente el cumplimiento de ventas solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado PK_CC_PROCESO_IND_CUMPLIM_VTA.

P_ACTUALIZACION_DIA_IND_CUMVTA. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultado en el Anexo A31.

6.2.1.2 Indicador de rentabilidad.

El indicador de rentabilidad se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado PK_CC_PROCESO_IND_RENTABILIDAD. P_CARGAR_IND_REN_CIERRE_MES y el segundo proceso actualizará diariamente el índice de rentabilidad solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado PK_CC_PROCESO_IND_RENTABILIDAD.

P_ACTUALIZACION_DIARIA_IND_REN. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A32.



6.2.1.3 Indicador Morosidad de la cartera.

El indicador morosidad de cartera se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado `PK_CC_PROCESO_INDICE_MOROSIDAD`. `P_CARGAR_IND_MOR_CIERRE_MES` y el segundo proceso actualizará diariamente el índice de morosidad solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado `PK_CC_PROCESO_INDICE_MOROSIDAD`. `P_ACTUALIZACION_DIARIA_IND_MOR`. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A33.

6.2.1.4 Indicador Flujo Corriente.

El indicador flujo corriente se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado `PK_CC_PROCESO_FLUJO_CORRIENTE`. `P_CARGAR_FLUJO_CORRIENTE` y el segundo proceso actualizará diariamente el índice del flujo corriente solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado `PK_CC_PROCESO_FLUJO_CORRIENTE`. `P_ACTUALIZACION_FLU_COR`. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A34.

6.2.1.5 Indicador Mes Presente Puntual.

El indicador mes presente puntual se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado `PK_CC_PROCESO_MES_PRESENTE_PUN`. `P_CARGAR_MES_PRESENTE_PUNTUAL` y el segundo proceso actualizará diariamente el índice del mes presente puntual solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado `PK_CC_PROCESO_MES_PRESENTE_PUN`. `P_ACTUALIZACION_DIA_MES_PRE_PU`. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A35.



6.2.1.6 Indicador Mes Presente Total.

El indicador mes presente total se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado `PK_CC_PROCESO_MES_PRESENTE_TOT`. `P_CARGAR_MES_PRESENTE_TOTAL` y el segundo proceso actualizará diariamente el índice del mes presente total solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado `PK_CC_PROCESO_MES_PRESENTE_TOT`. `P_ACTUALIZACION_DIA_MES_PRE_TO`. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A36.

6.2.1.7 Indicador Primeras Cuotas.

El indicador primeras cuotas se actualizará en dos instancias, el primero se actualizará al cierre de cada mes mediante el procedimiento almacenado llamado `PK_CC_PROCESO_PRIMERAS_CUOTAS`. `P_CARGAR_PRI_CUO_CIERRE_MES` y el segundo proceso actualizará diariamente el índice de primeras cuotas solo del período vigente, esto se lo realizará a través del procedimiento almacenado llamado `PK_CC_PROCESO_PRIMERAS_CUOTAS`. `P_ACTUALIZACION_DIARIA_PRI_CUO`. La descripción de las acciones que contempla el procedimiento almacenado puede ser consultada en el Anexo A37.

En este capítulo hemos creado los procesos para la extracción y carga de la información; se han creado procedimientos almacenados para la carga inicial de las tablas de dimensiones y las tablas de hechos y posteriormente se crearon procedimientos almacenados para actualizar diariamente y mensualmente las tablas de dimensiones e indicadores, en el siguiente capítulo vamos a construir el prototipo de visualización del *datamart*.



CAPÍTULO 7.

CONSTRUCCIÓN DEL PROTOTIPO DE HERRAMIENTA BI.

En este capítulo se realizó la construcción del prototipo de herramienta de BI, que consiste en un visualizador de la información almacenada en el *datamart* creado para *Impocomjaher*. La aplicación utilizada para la creación de la herramienta de visualización será apex de Oracle.

7.1 Instalación de apex y Construcción del prototipo de visualización de la información.

Como se indicó en la sección 1.3 “esta tesis se centrará en el análisis y desarrollo de un *datamart* y no en la solución de business intelligence”, por esta razón describiremos en forma general los pasos que se realizaron para la instalación y configuración de apex² al igual que la creación del prototipo. A continuación indicaremos las actividades realizadas:

7.1.1 Instalación y configuración de apex.

Para la instalación de apex comenzamos creando el tablespace³ donde se va a crear el usuario apex y el tablespace donde se van a crear los archivos del usuario apex, los tablespace fueron creados con el usuario SYS de la siguiente forma:

```
SQL> CREATE SMALLFILE TABLESPACE “APEX_TBS_040200” DATAFILE  
‘+DATOS/apex_tbs_040200.dbf’ SIZE 512M REUSE LOGGING EXTENT  
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

```
SQL> CREATE SMALLFILE TABLESPACE “APEX_TBS_FILES” DATAFILE  
‘+DATOS/apex_tbs_files.dbf’ SIZE 512M REUSE LOGGING EXTENT  
MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

A continuación creamos el directorio donde vamos a instalar apex, esto lo realizamos mediante el siguiente comando:

```
$ mkdir apex422
```

En este directorio descomprimimos el archivo zip del instalador de apex, una vez descomprimido el archivo nos vamos al directorio apex, ingresamos al

² Apex: Oracle Application Express. es una herramienta de desarrollo rápido de aplicaciones que se ejecuta con una base de datos Oracle.

³ Tablespace, es una unidad lógica de almacenamiento dentro de una base de datos Oracle.



sqlplus con el usuario SYS de la base de datos para instalar apex, para ello ejecutamos los siguientes comandos:

```
$ cd apex422/apex
```

```
$ sqlplus sys/**** as sysdba
```

```
SQL> @apexins.sql APEX_TBS_040200 APEX_TBS_FILES TEMP /i/
```

APEX_TBS_040200 es el nombre del tablespace para el usuario de la aplicación apex (oracle application express).

APEX_TBS_FILES es el nombre del tablespace para los archivos del usuario apex (oracle application express).

TEMP corresponde al nombre del tablespace temporal.

/i/ corresponde al directorio virtual de imágenes de apex.

Una vez instalado apex cambiamos la contraseña del usuario ADMIN y definimos el puerto, esto lo hacemos mediante el siguiente comando:

```
SQL> @apxconf.sql
```

A continuación desbloqueamos las siguientes cuentas:

```
SQL> ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

```
SQL> ALTER USER ANONYMOUS ACCOUNT UNLOCK;
```

A continuación especificamos el directorio donde se descomprimió el instalador de apex, esto se realiza mediante el siguiente comando:

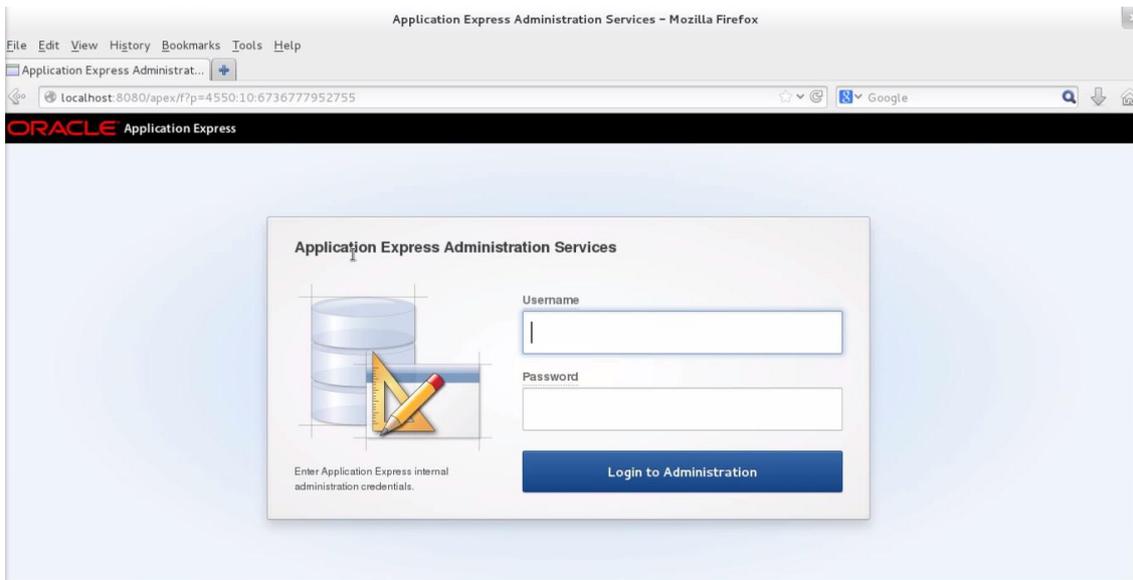
```
SQL> @apex_epg_config.sql /u01/...../apex422
```



Para verificar la instalación de apex abrimos el navegador y digitamos la siguiente dirección:

http://localhost:8080/apex/apex_admin

A continuación visualizamos la siguiente pantalla:

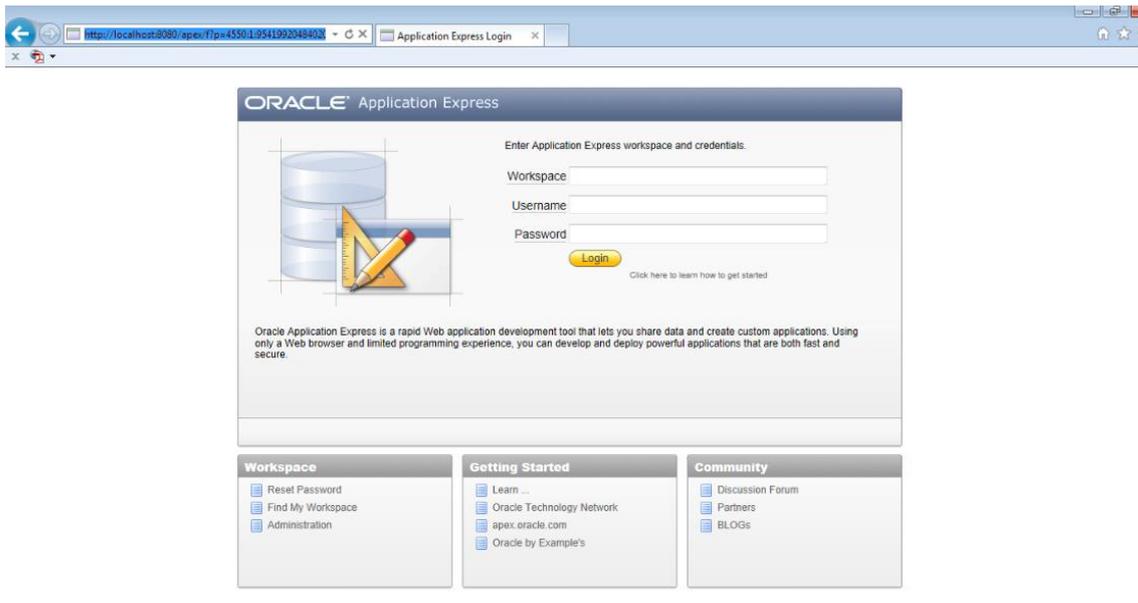


En esta sección hemos narrado como instalamos y configuramos apex, en la siguiente sección vamos a describir la forma como se construyó la página principal del aplicativo.

7.1.2 Construcción de la página principal del prototipo de visualización de la información.

Una vez instalado y configurado apex en la sección 7.1.1 procedemos a construir la página principal del prototipo. Volvemos a recalcar como lo indicamos en la sección 1.3 “esta tesis se centrará en el análisis y desarrollo de un *datamart* y no en la solución de business intelligence”, por esta razón en esta sección solo vamos a describir la creación de la página principal con sus opciones para tener una idea de cómo se utilizó la herramienta.

Para crear la página principal del aplicativo con las opciones, ingresamos a apex mediante la siguiente dirección <http://localhost:8080/apex>, a continuación visualizamos la siguiente pantalla:



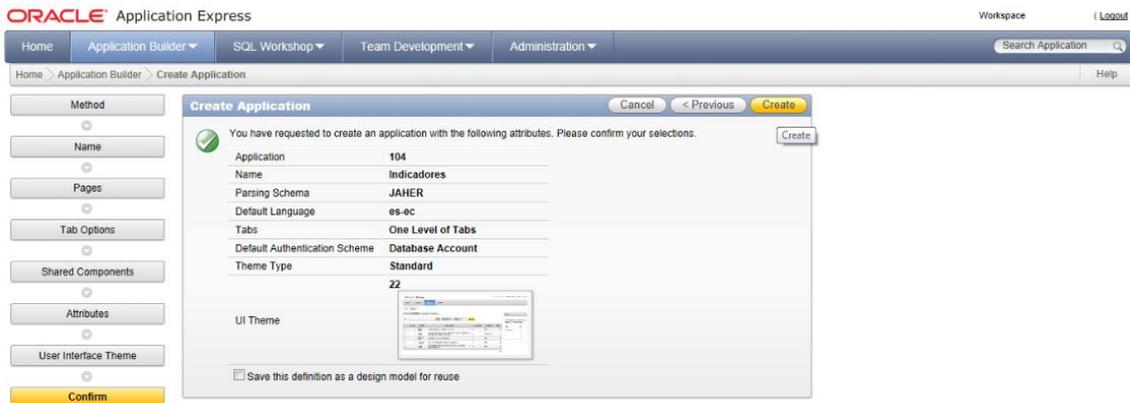
Una vez digitado los datos solicitados, ingresamos al apex dando un clic en el botón login, a continuación presentará la pantalla:



Ahora vamos a crear la aplicación dando clic en el botón Application Builder, luego damos un clic en el botón  para crear la aplicación, nos presentará la pantalla siguiente:



En esta pantalla seleccionamos la opción Database porque va a ser autenticado a nivel de base de datos, damos clic en el botón **Next >** y a continuación ingresamos el nombre que le vamos a dar a nuestro aplicativo, a continuación definimos las paginas (opciones) que va a tener nuestro aplicativo, luego definimos los atributos como el lenguaje que va a ser español Ecuador, el formato de la fecha que va a ser año, mes y día (yyyy-mm-dd), el formato de la fecha y hora que va a ser año, mes, día horas, minutos y segundos (yyyy-mm-dd hh24:mi:ss) a continuación seleccionamos la interface 22 que va a ser nuestro template para el aplicativo y al final nos solicita confirmar todo lo definido, presentamos la pantalla resumen.



En esta sección hemos descrito la forma como creamos la página principal de nuestro aplicativo.

En este capítulo hemos descrito la instalación, configuración de apex y la construcción de la página principal del prototipo de visualización de la información. En el siguiente capítulo describiremos los manuales de operación del prototipo de herramienta de visualización de la información del *datamart*.



CONCLUSIONES

1. El acceso a la información para la toma de decisiones es un factor crítico en toda organización, razón por la cual debe obtenerse de manera oportuna y eficiente, además debe existir una correcta integración y calidad de la información que permita contar con una fuente fiable para la toma de decisiones.
2. El datamart producto de este trabajo es construido como apoyo estratégico y gerencial para conseguir los objetivos de Impocomjahaer. Este cubre las necesidades actuales de información estratégica de la empresa en el ámbito de las Ventas y Cobranzas.
3. La construcción del datamart permite contar con un repositorio de información único y disponible para el análisis de los indicadores definidos.
4. El levantamiento de la información realizado en la etapa inicial permitió diagnosticar las necesidades más importantes de información estratégica de los departamentos de ventas y cobranzas.
5. Hay que tener especial atención en los procesos ETL (Extract Transform and Load), ya que deben estar correctamente definidos, para garantizar la calidad y validez de la información del datamart.
6. La metodología hefesto utilizada para la construcción del datamart permitió contar con las mejores prácticas de la industria en el ámbito de la Inteligencia de Negocios para el desarrollo de esta tesis.
7. La construcción del prototipo permite disponer de una solución de Inteligencia de negocios para explotar la información del datamart.



RECOMENDACIONES

1. La presente tesis ha permitido construir el primer datamart para Impocomjaher, por lo que se recomienda continuar definiendo más indicadores de acuerdo a las necesidades de información gerencial y estratégica en las diferentes áreas de la empresa.
2. Sociabilizar los indicadores definidos en este proyecto a todos los departamentos en la empresa que se considere necesario y concientizar la importancia de los mismos a la organización.
3. Realizar evaluaciones constantes de la información obtenida con la finalidad de hacer optimizaciones a los indicadores, incorporar nuevos y a su vez validar la Calidad de Información.
4. Hacer revisiones de la metodología de manera periódica con la finalidad de optimizarla.
5. Continuar evolucionando el prototipo de solución de BI en función a las necesidades de la organización.
6. Evaluar la implementación de soluciones más completas de Inteligencia de Negocio que permitan explotar de mejor forma la información de los *datamarts* definidos.



GLOSARIO

BI: Son las siglas en inglés de Business Intelligence, la Inteligencia de negocio es el uso de los datos de una empresa para facilitar la toma de decisiones mediante la comprensión del funcionamiento actual y la anticipación de acciones para dar una dirección operativa optima a la empresa.

Datamart: Un *Datamart* es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica.

Datawarehouse: Un Datawarehouse es una colección de datos orientada al negocio, integrada, variante en el tiempo y no volátil para el soporte del proceso de toma de decisiones de la gerencia.

ETL: son las siglas en inglés de Extraer, Transformar y Cargar (Extract, Transform and Load). Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos, limpiarlos y cargarlos en otra base de datos, datamart o Datawarehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

EVA: El software HP StorageWorks Command View EVA es un conjunto de aplicaciones diseñado para simplificar el aprovisionamiento de cabinas de discos y la gestión de toda la familia HP StorageWorks EVA de los productos de cabina de almacenamiento. Puede crear Vdisks de manera sencilla y aumentar o reducir los LUN sin que haya un periodo de inactividad.

Hardware: Conjunto de los componentes que integran la parte material de una computadora.

Indicador: Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos en la ejecución de un proyecto, programa o actividad, es decir son herramientas que ayudan en su utilización a medir el nivel de cumplimiento, control y funcionamiento efectivo de una actividad.

LUN: En almacenamiento, una logical unit number o LUN es una dirección para una unidad de disco duro y por extensión, el disco en sí mismo, un LUN no es normalmente un disco entero sino una partición virtual (o *volumen*) dentro de un conjunto RAID.

Modelo conceptual: Descripción de alto nivel de la estructura de la base de datos, en la cual la información es representada a través de objetos, relaciones y atributos.



Modelo lógico: Representación de una estructura de datos, que puede procesarse y almacenarse en algún sistema de gestión de base de datos (SGBD).

OLTP: Siglas en inglés On Line Transaction Processing, sistemas de procesamiento de transacciones en línea, o sistemas transaccionales, en los cuales residen las operaciones del día a día de cada negocio y que son la fuente prioritaria de datos para cada datamart o Datawarehouse corporativo.

Perspectiva: Punto de vista desde el cual se considera o se analiza un asunto, es decir las perspectivas se refieren a los objetos mediante los cuales se quiere examinar los indicadores, con el fin de responder a las preguntas planteadas.

PL/Sql: (Procedural Language/Structured Query Language). Es un lenguaje de programación incrustado en Oracle. El PL/SQL es un lenguaje que permite definir procedimientos utilizando sentencias SQL.

RAID: siglas en inglés Redundant Array of Independent Disks, conjunto redundante de discos independientes, hace referencia a un sistema de almacenamiento de datos que usa múltiples unidades de almacenamiento de datos (discos duros) entre los que se distribuyen o replican los datos.

Sistema Operativo: Un sistema operativo (SO, frecuentemente OS, del inglés *Operating System*) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes y anteriores.

Switch: Un conmutador o *switch* es un dispositivo digital lógico de interconexión de redes de computadoras. Su función es interconectar dos o más segmentos de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red.

SGBD: Los SGBD (Sistema de Gestión de Base de Datos) son un tipo de software muy específico, dedicados a servir de interfaz entre la base de datos, los usuarios y las aplicaciones que lo utilizan. Se compone de lenguajes de definición, manipulación, consulta y seguridad de datos.

Tablas de hechos: Tabla de datos central que contiene cada una de las métricas que se desea medir dentro de un área específica del negocio.



Tablas de Dimensiones: Se refiere a la entidad que se encarga de agrupar, calificar o catalogar cada uno de los hechos contenidos dentro de un datamart.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN).

Apex: Oracle Application Express. Es una herramienta de desarrollo rápido de aplicaciones que se ejecuta con una base de datos oracle.

Tablespace: es una unidad lógica de almacenamiento dentro de una base de datos Oracle.



BIBLIOGRAFÍA

- [1] **William H. Inmon**, "Building the Data Warehouse". Cuarta Edición. Wiley Publishing, Inc. Indianapolis. 2005.
- [2] **Robert Stackowiak, Joseph Rayman, Rick Greenwald**, "ORACLE DATA WAREHOUSING and Business Intelligence Solutions". Wiley Publishing, Inc. Indianapolis. 2007.
- [3] **Ralph Kimball, Margy Ross, Warren Thorntwaite, Joy Mundy, Bob Becker**, "THE DATA WAREHOUSE LIFECYCLE TOOLKIT", Practical Techniques for Building Data Warehouse and Business Intelligence Systems. Segunda Edición. Wiley Publishing, Inc. Indianapolis. 2008.
- [4] **Ralph Kimball, Joe Caserta**, "THE DATA WAREHOUSE ETL TOOLKIT ", Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley Publishing, Inc. Indianapolis. 2004.
- [5] **Lilian Hobbs, Susan Hillson, Shilpa Lawande, Pete Smith**, "ORACLE DATABASE 10g DATA WAREHOUSING". Elsevier Digital Press. 2005.
- [6] **Gavin Powell**, "ORACLE DATA WAREHOUSE TUNNING FOR 10g". Elsevier Digital Press. 2005.
- [7] **Carolyn K. Hamm, Ph.D.**, "ORACLE DATA MINING, Mining Gold From Your Data Warehouse". Primera Edición. Rampant TechPress. September 2007.
- [8] **Bon K. Sy, Arjun K. Gupta**. "INFORMATION-STATISTICAL DATA MINING, Warehouse Integration with Examples of Oracle Basics", Kluwer Academic Publishers. 2004.
- [9] **Marlene Theriault, Rachel Carmichael, James Viscusi**. "ORACLE 9i DBA 101, Learn the Essentials of Oracle Database Administration". McGraw-Hill. 2002.
- [10] **Bob Bryla, Kevin Loney**. "ORACLE DATABASE 11g DBA HANDBOOK, Administer a Scalable, Secure Oracle Enterprise Database". McGraw-Hill. 2008.
- [11] **Ing. Bernabeu Ricardo Dario**. "DATAWAREHOUSING: Investigación y Sistematización de Conceptos. HEFESTO: Metodología para la Construcción de un Data Warehouse.". Disponible. <http://www.businessintelligence.info/docs/hefesto-v2.pdf>.



- [12] **Arie Geller, Matthew Lyon.** “Oracle Application Express 3.2 The Essentials and More”. Packt Publishing. 2010.
- [13] **Doug Gault, Karen Cannell, Patrick Cimolini, Martin Giffy D’Souza, and Timothy St. Hilaire.** “Beginning Oracle Application Express”. Apress. 2011.
- [14] **Jhon Scott, Dietmar Aust, Martin D’Souza, Doug Gault, Dimitri Gielis, Roel Hartman, Michael Hichwa, Sharon Kennedy, Denes Kubicek, Raj Mattamal, Dan McGhan, Francis Mignault, and Anton Nielsen.** “Expert Oracle Application Express”. Apress. 2011.
- [15] **Marcel van der Plas, Michel van Zoest.** “Oracle APEX 4.0 Cookbook”. Packt Publishing. 2010.
- [16] **Martin Giffy D’Souza.** “Expert Oracle Application Express Plugins”. Packt 2011.
- [17] **Edmund Zehoo.** “Oracle Application Express 4 Recipes”. Apress. 2011.
- [18] Apuntes de clases.



ANEXOS



Anexo A1.

Aplicaciones (Sistema Operativo).

Impocomjaher Cía. Ltda.
Inventario Sistema Operativo Servidores Base de Datos y Application Server

Servidor	Localización	Ubicación	Sistema Operativo	Versión Sistema Operativo	Arquitectura
Base de datos Producción JAHER	Cuenca	Sala de Servidores	Redhat Enterprise	5.5	x64
Oracle Application Server JAHER	Cuenca	Sala de Servidores	Redhat Enterprise	5.5	x64
Administrador EVA JAJER	Cuenca	Sala de Servidores	Windows Server 2008	R2 Standar Sp1	x64
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Standar edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Standar edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Standar edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Standar edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Enterprise edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2000	Advanced edition Sp2	x86
Terminal Server JAHER	Cuenca	Sala de Servidores	Windows Server 2003	Enterprise edition Sp2	x86
Base de datos Desarrollo	Cuenca	Sala de Servidores	Redhat Enterprise	5.5	X64



Anexo A2.

Equipos Informáticos (Hardware).

Impocomjaher Cía. Ltda.

Inventario Servidores Base de Datos y Terminal Server

Servidor	Modelo	Procesador	RAM	HHDD	Revoluciones HHDD	Arquitectura
Base de datos Producción JAHER	HP ProLiant BL460c G6	2 CPU Six Core Intel Xeon 2933MHz (equivalente 24 microprocesadores)	64 Gb	2 Discos Internos de 146Gb cada uno	15k	x64
Oracle Application Server JAHER	HP ProLiant BL460c G6	2 CPU Six Core Intel Xeon 2933MHz (equivalente 24 microprocesadores)	64 Gb	2 Discos Internos de 146Gb cada uno	15k	x64
Administrador EVA JAHER	ProLiant BL280c G6	1 CPU Quad Core Intel Xeon 2000MHz (equivalente 4 microprocesadores)	4 Gb	2 Discos Internos de 120Gb cada uno		x64
Storage JAHER	Hp Storage Works HSV300			1 Cuchilla de 14 Discos de 300 Gb cada uno	15k	
TapeBlade	HP StorageWorks TapeBlade MSL2024			22 Cintas de Respaldo 1.2 Tb c/u		
Terminal Server JAHER	HP ProLiant BL280c G6	1 CPU Quad Core Intel Xeon 2000MHz (equivalente 4 microprocesadores)	4 Gb	2 Discos Internos de 120Gb cada uno		x86
Terminal Server JAHER	HP ProLiant ML 370 G5	Intel Xeon 1,86MHz (equivalente a 4 microprocesadores)	4 Gb	4 Discos Internos de 72Gb cada uno	10k	x86
Terminal Server JAHER	HP ProLiant DL160 G6	Intel Xeon 1,86MHz (equivalente a 4 microprocesadores)	4 Gb	2 Discos Internos de 250Gb cada uno	7.2 k	x86
Terminal Server JAHER	HP ProLiant DL160 G6	Intel Xeon 1,86MHz (equivalente a 4 microprocesadores)	4 Gb	discos internos		x86
Terminal Server JAHER	HP ProLiant ML350 G6	Intel Xeon 2.27MHz (equivalente a 8 microprocesadores)	18Gb	2 Discos Internos de 300Gb cada uno	10k	x86
Terminal Server JAHER	HP ProLiant ML110 G2		2 Gb			x86
Terminal Server JAHER	HP ProLiantDL360 G7	Intel Xeon 2.4MHz (equivalente a 6 procesadores, 12 procesadores lógicos)	18Gb	2 Discos Internos de 300Gb cada uno	10k	x86
Terminal Server JAHER	HP ProLiant ML110 G3	Pentium D 3.9Ghz (equivalente a 2	2			x86

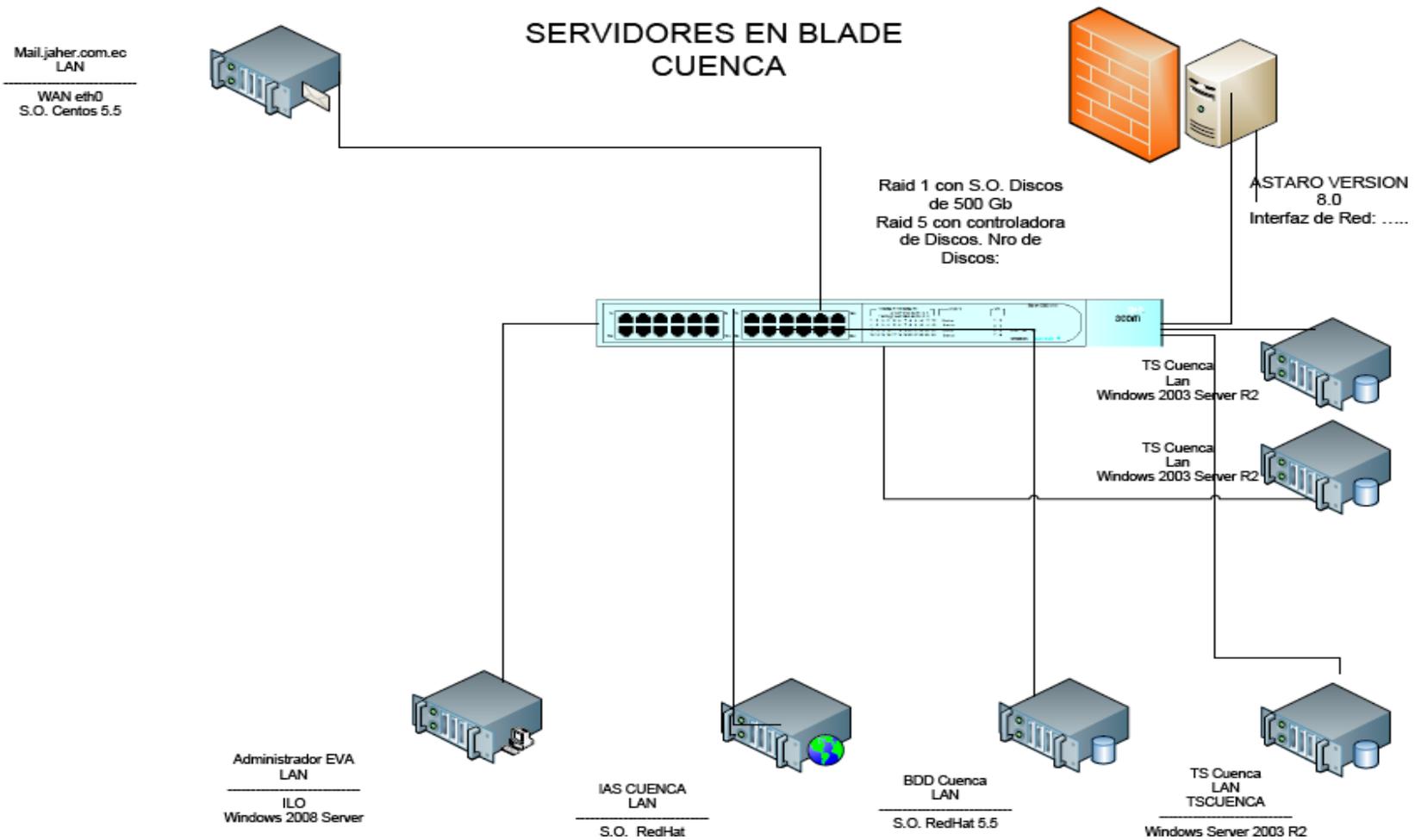


		microprocesadores)	Gb		
Base de datos Desarrollo	Hp ProLiant DL380 G6	Intel Xeon 2.40 GHz	6 Gb	3 Discos de 146 Gb Serial SCSI 4 Discos de 300 Gb SAS	X64



Anexo A3.

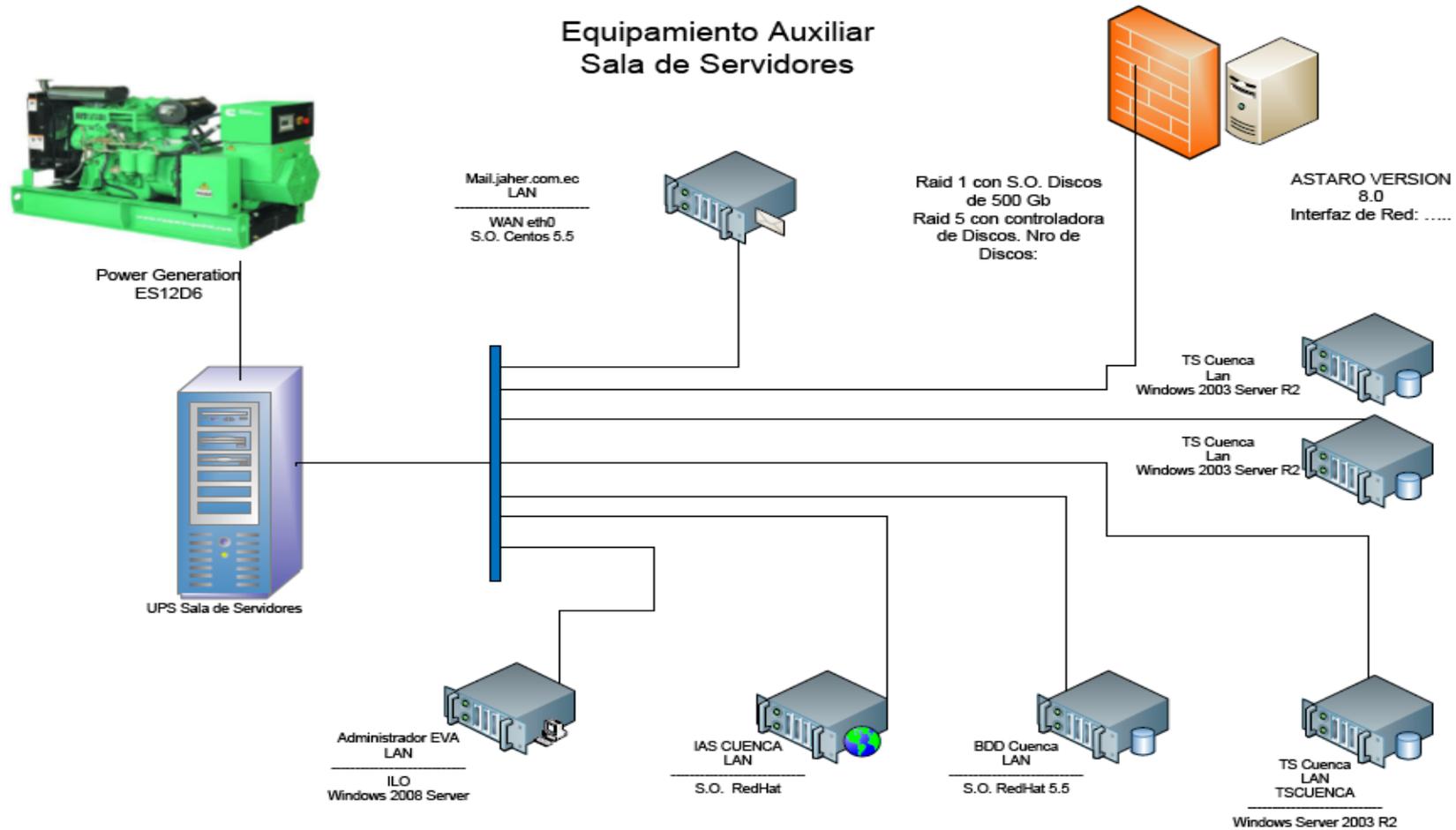
Redes de comunicación.





Anexo A4.

Equipamiento Auxiliar.





Anexo A5.

Diccionario de Datos de los OLTP.

Tabla 3.8 Ad_naciones

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_NACION	NUMBER(3) not null	Código de la nación
DESCRIPCION	VARCHAR2(100) not null	Nombre de la nación
CAPITAL	VARCHAR2(100) not null	Capital de la nación
CODIGO_AREA	NUMBER(5) not null	Código de area
NACIONALIDAD	VARCHAR2(50) not null	Nacionalidad
CONTINENTE	NUMBER(1) not null	Continente
SIGLA	VARCHAR2(2) not null	Sigla
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro



Tabla 3.9 Ad_empresas

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
DESCRIPCION	VARCHAR2(100) not null	Nombre de la empresa
CODIGO_PARROQUIA	VARCHAR2(4) not null	Código de la parroquia
CODIGO_CANTON	VARCHAR2(4) not null	Código del cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_NACION	NUMBER(3) not null	Código de la nación
DIRECCION	VARCHAR2(100) not null	Dirección de la empresa
TELEFONO1	VARCHAR2(15)	Teléfono 1 de la empresa
TELEFONO2	VARCHAR2(15)	Teléfono 2 de la empresa
FAX	VARCHAR2(15)	Fax de la empresa
RUC	VARCHAR2(20) not null	Número del RUC
CASILLA	VARCHAR2(10)	Número de casilla
CONTADOR_IDENTIFICACION	VARCHAR2(20)	Identificación del contador
NUMERO_PATRONAL	VARCHAR2(10)	Número patronal
GERENTE_IDENTIFICACION	VARCHAR2(20)	Identificación del gerente
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE,	Fecha que modifica el registro
ANULADO	VARCHAR2(1) not null	Registro Anulado Si o No



Tabla 3.10 Ad_parametros_generales

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
CODIGO_MONEDA	NUMBER(2) not null	Código de la moneda del país
PORCENTAJE_IVA	NUMBER(7,4) not null	Porcentaje del IVA
TASA_INTERES_DEUDOR	NUMBER(7,4) not null	Tasa de interés deudora
TASA_INTERES_ACREEDOR	NUMBER(7,4) not null	Tasa de interés acreedora
PORCENTAJE_DESCUENTO	NUMBER(7,4)	Porcentaje de descuento
PORCENTAJE_ENTRADA	NUMBER(7,4)	Porcentaje de entrada
PORCENTAJE_IESS_PATRONAL	NUMBER(7,4)	Porcentaje IESS patronal
PORCENTAJE_IESS_CESANTIA	NUMBER(7,4)	Porcentaje IESS cesantia
PORCENTAJE_IESS_SECAP	NUMBER(7,4)	Porcentaje IESS secap
PORCENTAJE_IESS_IECE	NUMBER(7,4)	Porcentaje IECE
PORCENTAJE_REFINANCIACION	NUMBER(7,4)	Porcentaje de refinanciamiento
VALOR_ACCIONES	NUMBER(12)	Valor de acciones
FACTOR_VENTA	NUMBER(14,2)	Factor de Venta
NUMERO_DECIMALES	NUMBER(4) default 0 not null	Número de decimales
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE,	Fecha que modifica el registro
PATH_REPORTES_SERVIDOR	VARCHAR2(300)	Ubicación de los reportes en el servidor
PATH_REPORTES_CLIENTE	VARCHAR2(300)	Ubicación de los reportes en el cliente
PATH_LOGOS_EMPRESAS	VARCHAR2(300)	Ubicación de logos de la empresa
CIERRE_CARTERA_DIAS	NUMBER(2) not null	Número de días para el cierre del balance desde el último día del mes
COMPROMISO_PAGO	VARCHAR2(10) not null	Código del compromiso de pago
NOMBRE_ARCHIVO_COBRO_CELULAR	VARCHAR2(30)	Nombre del archivo del cobro en celulares
PATH_LEER_ARCHIVOS_USB	VARCHAR2(200)	Ubicación para leer los archivos desde el celular
PATH_GRABAR_ARCHIVO_USB	VARCHAR2(200)	Ubicación para grabar los archivos al celular
PATH_ARCHIVOS_LOCALES_CLIENTE	VARCHAR2(200)	Ubicación de los archivos locales del cliente
ID_SERVIDOR	VARCHAR2(3) not null	ID del servidor
CASA	VARCHAR2(10)	Código de la gestión domiciliaria
TRABAJO	VARCHAR2(10)	Código de la gestión del trabajo
DIR_NO_VALIDA	VARCHAR2(10)	Código de la dirección no valida
DIAS_PLAZO_C_P	NUMBER(2)	Número de días máximo para el compromiso de pago



INTENTOS_COMP_PAGO	NUMBER(2)	Número de intentos para almacenar compromiso de pago
MAX_FECHA_COMP_PAGO	NUMBER(2)	Días máximo para el compromiso de pago
CODIGO_SEGMENTS_CARTERA	NUMBER(3)	Código segmento de cartera que se considera para el reporte de cobros del contact center
SALDO_MINIMO_ENVIAR_RECAUDADOR	NUMBER(18,2)	Saldo mínimo para enviar a la recaudadora externa
CODIGO_CARGO_COBRADOR	NUMBER(6)	Código del cargo del cobrador
CODIGO_CARGO_OPERADOR_TELEF	NUMBER(6)	Código del cargo del operador telefónico
CODIGO_CARGO_VERIFICADOR	NUMBER(6)	Código del cargo del verificador
CODIGO_GESTION_CLI_NO_UBICADO	VARCHAR2(10)	Código de tipo de gestión de clientes no ubicados
TRUNK_DOMICILIO	VARCHAR2(500)	Trunk para los teléfonos del domicilio convencional
TRUNK_CELULAR	VARCHAR2(500)	Trunk para los teléfonos celulares
COMPROMISO_PAGO_CALL_CENTER	VARCHAR2(10) not null	Código del tipo de gestión para el compromiso de pago del contact center
DIAS_MORA_ENV_RECAUD_EXTER	NUMBER(4) not null	Días de mora desde el cuál debe considerar para enviar a las recaudadoras externas
SALDO_X_CUOTA_ENV_RECAUD_EXTER	NUMBER(18,2) not null	Saldo de cada cuota debe ser mayor para ser enviado a las recaudadores externas
CODIGO_ENVIAR_A_URD	VARCHAR2(10)	Código del tipo de gestión para enviar a la URD
NUMERO_CAMPANIA_URD	NUMBER(10)	Número de campaña actual de la URD
CODIGO_NACION	NUMBER(3)	Código de la nación
COD_EST_CUO_CLI_NO_UBI	VARCHAR2(3)	Código del estado de cuota para enviar a la URD
URD_CAMPANIA_CLIE_REPETIDOS	NUMBER(1)	Número de veces que se repite el cliente en las campañas de la URD

Tabla 3.11 Ad_regiones_x_empresas

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_REGION	NUMBER(3) not null	Código de la región
CODIGO_NACION	NUMBER(3) not null	Código de la nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
DESCRIPCION	VARCHAR2(300) not null	Nombre de la región
ANULADO	VARCHAR2(1) not null	Registro anulado Si o No
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro



Tabla 3.12 Tg_agencia

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
COD_AGENCIA	NUMBER(4) not null	Código de la agencia
NOMBRE	VARCHAR2(50) not null	Nombre de la agencia
COD_CATEGORIA_ZONA	VARCHAR2(2)	Código de la categoría de la zona geográfica de la agencia
EMPRESA_ZONA	NUMBER(2)	Código de empresa de la zona geográfica de la agencia
SECUENCIA_ZONA	NUMBER(6)	Código de la secuencia de la zona geográfica de la agencia
COD_NIVEL_ZONA	VARCHAR2(8)	Código del nivel de la zona geográfica de la agencia
CODIGO_ZONA	VARCHAR2(14)	Código de la zona geográfica de la agencia
DIRECCION	VARCHAR2(200)	Dirección de la agencia
OBSERVACIONES	VARCHAR2(200)	Observaciones
TELEFONO1	VARCHAR2(15)	Teléfono 1 de la agencia
TELEFONO2	VARCHAR2(15)	Teléfono 2 de la agencia
RUC	VARCHAR2(20)	Ruc de la agencia
ACTIVO	VARCHAR2(1)	Activo Si o No
COD_GRUPO_AGENCIA	VARCHAR2(3)	Código del grupo de la agencia
COD_SITIO	VARCHAR2(3) not null	Código del sitio de la agencia
ES_AUTORIZADO_SRI	NUMBER(1) default 0 not null	Es autorizado por el SRI
TIPO_RELACION_POLCRE	VARCHAR2(1) default 'N' not null	Tipo de relación de las políticas de crédito



Tabla 3.13 Ad_provincias

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_NACION	NUMBER(3) not null	Código de la nación
DESCRIPCION	VARCHAR2(100) not null	Nombre de la provincia
CODIGO_REGION	NUMBER(3) not null	Código de la región
SIGLA	VARCHAR2(3)	Siglas de la provincia
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro

Tabla 3.14 Ad_provincias_x_regiones

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_REGION	NUMBER(3) not null	Código de la región
CODIGO_NACION	NUMBER(3) not null	Código de la nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
ANULADO	VARCHAR2(1) not null	Registro anulado Si o No
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro



Tabla 3.15 Ad_cantones

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_CANTON	VARCHAR2(4) not null	Código del cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_NACION	NUMBER(3) not null	Código de la nación
DESCRIPCION	VARCHAR2(50) not null	Nombre del cantón
SIGLA	VARCHAR2(3)	Siglas del canton
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro

Tabla 3.16 Ad_cantones_x_provincias

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_CANTON	VARCHAR2(4) not null	Código del cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_REGION	NUMBER(3) not null	Código de la región
CODIGO_NACION	NUMBER(3) not null	Código de la nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
ANULADO	VARCHAR2(1) not null	Registro anulado Si o No
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro



Tabla 3.17 Ad_agencias_x_cantones

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_AGENCIA	NUMBER(4) not null	Código de la agencia
CODIGO_CANTON	VARCHAR2(4) not null	Código del cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_REGION	NUMBER(3) not null	Código de la región
CODIGO_NACION	NUMBER(3) not null	Código de la nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
ANULADO	VARCHAR2(1) not null	Registro anulado Si o No
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro

Tabla 3.18 Ve_objetivos_x_agencias

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
PERIODO	NUMBER(6) not null	Período
CODIGO_AGENCIA	NUMBER(4) not null	Código de la agencia
CODIGO_CANTON	VARCHAR2(4) not null	Código del cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de la provincia
CODIGO_REGION	NUMBER(3) not null	Código de la región
CODIGO_NACION	NUMBER(3) not null	Código de la nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
VALOR_OBJETIVO_MENSUAL	NUMBER(30,2) not null	Valor del objetivo mensual
VALOR_OBJETIVO_DIARIO	NUMBER(30,2) not null	Valor del objetivo diario
ANULADO	VARCHAR2(1) not null	Registro anulado Si o No
ADICIONADO_POR	VARCHAR2(30) default USER not null	Código del usuario que inserta el registro
FECHA_ADICION	DATE default SYSDATE not null	Fecha inserta el registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE	Fecha que modifica el registro



Tabla 3.19 Tg_modelo_item

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
COD_MODELO	VARCHAR2(8) not null	Código del modelo PRO2=línea CL11=Tipo de cliente
COD_ITEM	VARCHAR2(3) not null	Código de la línea o tipo del cliente
NOMBRE	VARCHAR2(50) not null	Nombre de la línea o del tipo del cliente
OBSERVACIONES	VARCHAR2(2000)	Observación

Tabla 3.20 Marca

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
COD_MARCA	NUMBER(3) not null	Código de la marca
EMPRESA	NUMBER(2) not null	Código de la empresa
NOMBRE	VARCHAR2(50) not null	Nombre de la marca
DESCUENTO_PROMOCION	VARCHAR2(1)	Tiene descuento Si o No



Tabla 3.21 Producto

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
COD_PRODUCTO	VARCHAR2(14) not null	Código del producto
COD_MARCA	NUMBER(3) not null	Código de la marca
COD_UNIDAD	VARCHAR2(8) not null	Código de la unidad
COD_ALTERNO	VARCHAR2(14)	Código alterno del producto
NOMBRE	VARCHAR2(200) not null	Nombre del producto
COD_BARRA	VARCHAR2(13)	Código de barra
USERIDC	VARCHAR2(3) not null	Iniciales del usuario que crea el registro
PROMEDIO	NUMBER(12,2) not null	Promedio
IVA	VARCHAR2(1) default 'S' not null	Calcula el IVA Si o No
REFERENCIA	VARCHAR2(200)	Código de referencia
COSTO	NUMBER(12,2)	Costo del producto
ACTIVO	VARCHAR2(1) not null	El producto se encuentra activo Si o No
COD_UNIDAD_R	VARCHAR2(8)	Código de la Unidad de la referencia
ES_FABRICADO	VARCHAR2(1) not null	El producto es fabricado por la empresa Si o No
COD_MODELO_CAT	VARCHAR2(8)	Código modelo de la línea
COD_ITEM_CAT	VARCHAR2(3)	Código de la línea
COD_UNIDAD_F	VARCHAR2(8)	Código de unidad de fabricación
CANTIDAD	NUMBER(14,2)	Cantidad
SERIE	VARCHAR2(1)	El producto tiene serie Si o No
PRECIO	NUMBER(14,2)	Precio del producto



Tabla 3.22 Comprobante

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
TIPO_COMPROBANTE	VARCHAR2(2) not null	Tipo de comprobante
COD_COMPROBANTE	VARCHAR2(9) not null	Número de comprobante
COD_TIPO_PERSONA	VARCHAR2(3) not null	Código tipo de persona
COD_PERSONA	VARCHAR2(14) not null	Código de la persona
FECHA	DATE not null	Fecha del comprobante
PEDIDO	VARCHAR2(13)	Número de pedido
IVA	NUMBER(14,2) not null	Valor del IVA
VALOR	NUMBER(14,2) default 0 not null	Valor del comprobante
FINANCIAMIENTO	NUMBER(14,2) not null	Valor del financiamiento
OTROS	NUMBER(14,2) not null	Otros valores
DESCUENTO	NUMBER(14,2) not null	Valor de descuento
TIPO_IVA	VARCHAR2(1)	Tipo de IVA
COD_LIQUIDACION	VARCHAR2(9) not null	Código de la liquidación
USERIDC	VARCHAR2(3) not null	Código del usuario
FACTURA_MANUAL	VARCHAR2(9) not null	Factura Manual
ANULADO	VARCHAR2(1)	Comprobante anulado Si o No
NOMBRE_PERSONA	VARCHAR2(40)	Nombre del Cliente
TRANSPORTADOR	VARCHAR2(30)	Código del transportador
PLACA	VARCHAR2(7)	Placa
OBSERVACIONES	VARCHAR2(2000)	Observaciones
ENTRADA	NUMBER(14,2)	Valor de la entrada
COD_AGENTE	VARCHAR2(14)	Código del vendedor
CANCELADA	VARCHAR2(1)	Cancelada Si o No
SALDO	NUMBER(14,2)	Saldo del comprobante
COD_AGENCIA	NUMBER(4) not null	Código de la agencia
FORMA_PAGO	VARCHAR2(3)	forma de venta
COTIZACION	NUMBER(14,2)	Cotización
TRANSFERENCIA	VARCHAR2(1)	Transferencia Si o No
CODIGO_CLIENTE	VARCHAR2(14)	Código del cliente
TIPO_COMPROBANTE_PR	VARCHAR2(3)	Tipo de proforma
COD_COMPROBANTE_PR	VARCHAR2(9)	Código de la proforma
COD_TIPO_PERSONA_GAR	VARCHAR2(3)	Código tipo de garante



COD_PERSONA_GAR	VARCHAR2(14)	Código de la persona del garante
NUMERO_PAGOS	NUMBER(3) not null	Plazo
CUOTAS_GRATIS	NUMBER(4,2) not null	Número de cuotas gratis
DIAS_ATRAZO	NUMBER(6,2)	Días de atraso
DESCUENTO_PROMOCION	NUMBER(14,2)	Descuento de promoción
COD_TARJETA	VARCHAR2(3)	Código de la tarjeta de crédito
NUM_TARJETA	VARCHAR2(30)	Número de la tarjeta de crédito
NUM_RECAP	VARCHAR2(15)	Número de recap
NUM_VOUCHER	VARCHAR2(15)	Número de Voucher
NUM_AUTORIZACION	VARCHAR2(15)	Número de autorización
COD_POLITICA	NUMBER(2)	Código de política
TIPO_COMPROBANTE_PEDIDO	VARCHAR2(2)	Tipo comprobante pedido
COD_COMPROBANTE_PEDIDO	VARCHAR2(9)	Código de pedido
REBATE	NUMBER(14,2)	Valor del Rebate
BASE_IMPONIBLE	NUMBER(14,2)	Valor base imponible
BASE_EXCENTA	NUMBER(14,2)	Base excenta
COD_CAJA	NUMBER(3)	Código de caja
FECHA_VENCIMIENTO1	DATE,	Fecha del primer vencimiento
AUD_FECHA	DATE,	Fecha de creación del registro
AUD_USUARIO	VARCHAR2(30)	Código del usuario que crea el registro
AUD_TERMINAL	VARCHAR2(50)	Código del terminal
COD_TIPO_PERSONA_APROB	VARCHAR2(3)	Código tipo persona que aprueba
COD_PERSONA_APROB	VARCHAR2(14)	Código persona que aprueba
COD_TIPO_PERSONA_VERIF	VARCHAR2(3)	Código tipo persona que verifica
COD_PERSONA_VERIF	VARCHAR2(14)	Código persona que verifica



Tabla 3.23 Sta_rentabilidad_temporal

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
COD_AGENCIA	NUMBER(4) not null	Código de la agencia
FECHA	DATE not null	Fecha de la transacción
NUMERO_SERIE	VARCHAR2(30)	Número de serie
COD_PRODUCTO	VARCHAR2(14) not null	Código del producto
PAGOS	NUMBER(3) not null	Plazo
TOTAL_LINEA	NUMBER(14,2)	Valor total de la línea
IVA	NUMBER(14,2)	Valor del IVA
COD_PERSONA	VARCHAR2(14) not null	Código de la persona
DEB_CRED	NUMBER(1) not null	Débito o Crédito
COSTO_REEXPRESADO	NUMBER,	Costo reexpresado
USUARIO	VARCHAR2(20)	Código del usuario
TIPO_COMPROBANTE	VARCHAR2(2)	Tipo de comprobante
COD_COMPROBANTE	VARCHAR2(9)	Código de comprobante
COD_ESTADO_PRODUCTO	VARCHAR2(2)	Código de estado del producto

Tabla 3.24 St_politica_credito

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
COD_POLITICA	NUMBER(2) not null	Código de la política
EMPRESA	NUMBER(2) not null	Código de la empresa
NOMBRE	VARCHAR2(50) not null	Nombre de la política
OBSERVACIONES	VARCHAR2(500)	Observación



Tabla 3.25 Cc_balances_mensuales

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
COD_COMPROBANTE	VARCHAR2(9) not null	Código del comprobante
TIPO_COMPROBANTE	VARCHAR2(2) not null	Tipo de comprobante
PERIODO_CIERRE	NUMBER(6) not null	Período de cierre
CARTERA	VARCHAR2(3) not null	Tipo de cartera
CODIGO_AGENCIA	NUMBER(4) not null	Código de agencia
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
CODIGO_POLITICA	NUMBER(2) not null	Código de la política
NOMBRE_POLITICA	VARCHAR2(100) not null	Nombre de la política
CODIGO_ESTADO_CUOTA	VARCHAR2(3)	Código del estado de la cuota
CODIGO_PERSONA	VARCHAR2(14) not null	Código de la persona
CODIGO_TIPO_PERSONA	VARCHAR2(3) not null	código del tipo de la persona
ZONA_GEOGRAFICAH	VARCHAR2(15)	Código de la zona geográfica
NOMBRE_PARROQUIA	VARCHAR2(50)	Nombre de la parroquia
COD_MODELO	VARCHAR2(8)	Código de modelo de la zona
COD_ITEM	VARCHAR2(3)	Código del item de la zona
NOMBRE_ZONA	VARCHAR2(50)	Nombre de la zona
CUOTAS_GRATIS	NUMBER(4,2) not null	Número de cuotas gratis
FACTURA_MANUAL	VARCHAR2(9) not null	Factura manual
NUMERO_PAGOS	NUMBER(3) not null	Plazo
COD_AGENTE	VARCHAR2(14)	Código de vendedor
NOMBRE_VENDEDOR	VARCHAR2(255)	Nombre del vendedor
FECHA_VENCIMIENTO	DATE not null	Fecha del primer vencimiento
DIRECCION_CALLEH	VARCHAR2(200)	Dirección del cliente
NUMERO_CASA	VARCHAR2(20)	Número de casa del cliente
CALLE_TRANSVERSAL	VARCHAR2(100)	Calle transversal del cliente
TELEFONO1H	VARCHAR2(15)	teléfono1 del cliente
TELEFONO2H	VARCHAR2(15)	teléfono2 del cliente
COD_CAT_CLIENTE	VARCHAR2(8)	Calificación del cliente
COD_PRODUCTO	VARCHAR2(14)	Código del producto
NOMBRE_PRODUCTO	VARCHAR2(200)	Nombre del producto
COD_MODELO_CAT	VARCHAR2(8)	Código del modelo de la línea
COD_ITEM_CAT	VARCHAR2(3)	Código del item de la línea
NOMBRE_LINEA	VARCHAR2(50)	Nombre de la línea



ES_EMPLEADO	VARCHAR2(100) not null	Nombre de la empresa si es empleado
SALDO_ANTICIPO	NUMBER(14,2) not null	Saldo de anticipos
FECHA_INICIO_MES	DATE not null	Fecha de inicio del mes del balance
FECHA_FIN_MES	DATE not null	Fecha final del mes del balance
VALOR_MES_ACTUAL_VENCIDO	NUMBER(14,2) not null	Valor del mes actual vencido
VALOR_X_VENCER_29_MESES	NUMBER(14,2) not null	Valor por vencer 29 meses
VALOR_X_VENCER_28_MESES	NUMBER(14,2) not null	Valor por vencer 28 meses
VALOR_X_VENCER_27_MESES	NUMBER(14,2) not null	Valor por vencer 27 meses
VALOR_X_VENCER_26_MESES	NUMBER(14,2) not null	Valor por vencer 26 meses
VALOR_X_VENCER_25_MESES	NUMBER(14,2) not null	Valor por vencer 25 meses
VALOR_X_VENCER_24_MESES	NUMBER(14,2) not null	Valor por vencer 24 meses
VALOR_X_VENCER_23_MESES	NUMBER(14,2) not null	Valor por vencer 23 meses
VALOR_X_VENCER_22_MESES	NUMBER(14,2) not null	Valor por vencer 22 meses
VALOR_X_VENCER_21_MESES	NUMBER(14,2) not null	Valor por vencer 21 meses
VALOR_X_VENCER_20_MESES	NUMBER(14,2) not null	Valor por vencer 20 meses
VALOR_X_VENCER_19_MESES	NUMBER(14,2) not null	Valor por vencer 19 meses
VALOR_X_VENCER_18_MESES	NUMBER(14,2) not null	Valor por vencer 18 meses
VALOR_X_VENCER_17_MESES	NUMBER(14,2) not null	Valor por vencer 17 meses
VALOR_X_VENCER_16_MESES	NUMBER(14,2) not null	Valor por vencer 16 meses
VALOR_X_VENCER_15_MESES	NUMBER(14,2) not null	Valor por vencer 15 meses
VALOR_X_VENCER_14_MESES	NUMBER(14,2) not null	Valor por vencer 14 meses
VALOR_X_VENCER_13_MESES	NUMBER(14,2) not null	Valor por vencer 13 meses
VALOR_X_VENCER_12_MESES	NUMBER(14,2) not null	Valor por vencer 12 meses
VALOR_X_VENCER_11_MESES	NUMBER(14,2) not null	Valor por vencer 11 meses
VALOR_X_VENCER_10_MESES	NUMBER(14,2) not null	Valor por vencer 10 meses
VALOR_X_VENCER_09_MESES	NUMBER(14,2) not null	Valor por vencer 9 meses
VALOR_X_VENCER_08_MESES	NUMBER(14,2) not null	Valor por vencer 8 meses
VALOR_X_VENCER_07_MESES	NUMBER(14,2) not null	Valor por vencer 7 meses
VALOR_X_VENCER_06_MESES	NUMBER(14,2) not null	Valor por vencer 6 meses
VALOR_X_VENCER_05_MESES	NUMBER(14,2) not null	Valor por vencer 5 meses
VALOR_X_VENCER_04_MESES	NUMBER(14,2) not null	Valor por vencer 4 meses
VALOR_X_VENCER_03_MESES	NUMBER(14,2) not null	Valor por vencer 3 meses
VALOR_X_VENCER_02_MESES	NUMBER(14,2) not null	Valor por vencer 2 meses
VALOR_X_VENCER_01_MES	NUMBER(14,2) not null	Valor por vencer 1 mes
VALOR_VENCIDO_01_MES	NUMBER(14,2) not null	Valor vencido 1 mes
VALOR_VENCIDO_02_MESES	NUMBER(14,2) not null	Valor vencido 2 meses



VALOR_VENCIDO_03_MESES	NUMBER(14,2) not null	Valor vencido 3 meses
VALOR_VENCIDO_04_MESES	NUMBER(14,2) not null	Valor vencido 4 meses
VALOR_VENCIDO_05_MESES	NUMBER(14,2) not null	Valor vencido 5 meses
VALOR_VENCIDO_06_MESES	NUMBER(14,2) not null	Valor vencido 6 meses
VALOR_VENCIDO_07_MESES	NUMBER(14,2) not null	Valor vencido 7 meses
VALOR_VENCIDO_08_MESES	NUMBER(14,2) not null	Valor vencido 8 meses
VALOR_VENCIDO_09_MESES	NUMBER(14,2) not null	Valor vencido 9 meses
VALOR_VENCIDO_10_MESES	NUMBER(14,2) not null	Valor vencido 10 meses
VALOR_VENCIDO_11_MESES	NUMBER(14,2) not null	Valor vencido 11 meses
VALOR_VENCIDO_12_MESES	NUMBER(14,2) not null	Valor vencido 12 meses
VALOR_VENCIDO_MAS_12_MESES	NUMBER(14,2) not null	Valor vencido más de 12 meses
INTERES_MORA_GENERADO	NUMBER(14,2) not null	Valor interés mora generado
INTERES_MORA_ABONADO	NUMBER(14,2) not null	Valor interés mora abonado
GASTO_COBRANZA_GENERADO	NUMBER(14,2) not null	Valor gasto cobranza generado
GASTO_COBRANZA_ABONADO	NUMBER(14,2) not null	Valor gasto cobranza abonado
DIAS_ATRASO_ADELANTO	NUMBER(8) not null	Días de atraso o adelanto
CODIGO_SEGMENTO_CARTERA	NUMBER(3)	Código de segmento de cartera
NOMBRE_SEGMENTO_CARTERA	VARCHAR2(100)	Nombre del segmento de cartera
ADICIONADO_POR	VARCHAR2(30) default user not null	Código del usuario que crea el registro
FECHA_ADICION	DATE default sysdate not null	Fecha de la creación del registro
MODIFICADO_POR	VARCHAR2(30)	Código del usuario que modifica el registro
FECHA_MODIFICACION	DATE,	Fecha de la modificación del registro
NEGOCIADO_CON	VARCHAR2(255)	Código de la entidad financiera
NUMERO_NEGOCIACION	NUMBER(9)	Número de negociación
VALOR_CUOTA	NUMBER(14,2) not null	Valor de la cuota
VERIFICACION_CAMPO_POR	VARCHAR2(105)	Código del usuario que realizó la verificación de campo
VERIFICACION_TELEFONICA_POR	VARCHAR2(105)	Código del usuario que realizó la verificación telefónica
CREDITO_APROBADO_POR	VARCHAR2(105)	Código del usuario que aprobó el crédito
CODIGO_CALL_CENTER	NUMBER(3)	Código del call center
NOMBRE_CALL_CENTER	VARCHAR2(100)	Nombre del call center
ANIO_VENTA	NUMBER(4)	Año de venta
MES_VENTA	NUMBER(2)	mes de venta
DIA_VENTA	NUMBER(2)	día de venta
DIA_VENCIMIENTO	NUMBER(2)	Día del primer vencimiento
ULTIMA_GESTION	VARCHAR2(10)	Código de la última gestión de cobranza
FECHA_ULTIMA_GESTION	DATE,	Fecha de la última gestión de cobranza



CUPO_APROBADO	NUMBER(14,2)	Cupo del cliente
CREDIJAHER	VARCHAR2(1)	Tiene credijaher Si o No
VALOR_ENTRADA	NUMBER(14,2)	Valor de la entrada
VALOR_ADICIONAL	NUMBER(14,2)	Valor adicional
COD_GARANTE	VARCHAR2(14)	Código del garante
FECHA_NACIMIENTO	DATE,	Fecha de nacimiento del cliente
VALOR_INGRESOS	NUMBER(14,2)	Valor de ingresos del cliente
COBRADOR_ASIGNADO	VARCHAR2(100)	Cobrador asignado
FECHA_APLICACION_FONDO	DATE,	Fecha en la que se aplicó el fondo de garantía
MONTO_CREDITO	NUMBER(14,2)	Monto del crédito
FECHA_ULTIMO_PAGO	DATE,	Fecha del último pago
NUMERO_CUOTAS_CANCELADAS	NUMBER(3)	Número de cuotas canceladas
NUMERO_CUOTAS_MORA	NUMBER(3)	Número de cuotas en mora
OPERADOR_ASIGNADO	VARCHAR2(100)	Operador del Call Center asignado
PRESUPUESTO	NUMBER(14,2)	Valor del presupuesto
NRO_CREDIJAHER	VARCHAR2(21)	Número tarjeta de credijaher
TOTAL_VENCIDO	NUMBER(18,2)	Total vencido
TOTAL_X_VENCER	NUMBER(18,2)	Total por vencer
SALDO_DEL_CREDITO	NUMBER(18,2)	Saldo del crédito
CODIGO_OCUPACION_LABORAL	VARCHAR2(3)	Código de la ocupación laboral
OCUPACION_LABORAL	VARCHAR2(100)	Nombre de la ocupación laboral
CALIFICACION_CENTRAL_RIESGO	VARCHAR2(1)	calificación de la central de riesgo
CODIGO_TIPO_VIVIENDA	VARCHAR2(3)	Código tipo de vivienda
TIPO_VIVIENDA	VARCHAR2(100)	Nombre del tipo de vivienda
TOTAL_CREDITOS_VIGENTES	NUMBER(3) default 0 not null	Total de créditos vigentes
CODIGO_MARCA	NUMBER(3)	Código de marca
NOMBRE_MARCA	VARCHAR2(50)	Nombre de la marca
NOMBRE_CATEGORIA_CLIENTE	VARCHAR2(30)	Nombre de la calificación del cliente
NOMBRE_CLIENTE	VARCHAR2(255)	Nombres del cliente
FECHA_ULT_COBRO	DATE,	Fecha del último cobro
NOMBRE_AGENCIA	VARCHAR2(50)	Nombre de la agencia
NOMBRE_ULTIMA_GESTION	VARCHAR2(100)	Nombre de la última gestión
COD_SEGMEN_CARTERA_BAL_ANT	NUMBER(3)	Código del segmento de cartera del balance del mes anterior
NOMBRE_SEGMEN_CARTE_BAL_ANT	VARCHAR2(100)	Nombre del segmento de cartera del balance del mes anterior
FECHA_ULT_ACTUALIZ_SAL_ACT_HOY	DATE,	Fecha de la última actualización del saldo actual
SALDO_ACTUAL_HOY	NUMBER(18,2) default 0 not null	Saldo actual a hoy



DIAS_MINIMO_COBRO	NUMBER(10,2) default 0 not null	Días mínimo de cobro
DIAS_MAXIMO_COBRO	NUMBER(10,2) default 0 not null	Días máximo de cobro
DIAS_PROMEDIO_COBRO	NUMBER(10,2) default 0 not null	Días de promedio de cobro
DIAS_MINIMO_ENTRE_COBROS	NUMBER(10,2) default 0 not null	Días mínimo entre cobros
DIAS_MAXIMO_ENTRE_COBROS	NUMBER(10,2) default 0 not null	Días máximo entre cobros
DIAS_PROMEDIO_ENTRE_COBROS	NUMBER(10,2) default 0 not null	Días promedio entre cobros
TOTAL_GESTIONES_CALL_CENTER	NUMBER(10) default 0 not null	Total gestiones del call center
TOTAL_GESTIONES_POSITIVAS_CALL	NUMBER(10) default 0 not null	Total gestiones positivas del call center
TOTAL_GESTIONES_COBRAD_VERIF	NUMBER(10) default 0 not null	Total gestiones de cobradores verificados
VALOR_CANCELADO_DESDE	NUMBER(10,2) default 0 not null	Valor cancelado desde
TOTAL_CUOTAS_CANCELAD_DESDE	NUMBER(5) default 0 not null	Total cuotas cancelados desde
INTANGIBLE_GARANTIA_EXTENDIDA	NUMBER(10,2) default 0 not null	Intangible garantía extendida
INTANGIBLE_LINEA_BLANCA	NUMBER(10,2) default 0 not null	Intangible línea blanca
INTANGIBLE_ARROBA	NUMBER(10,2) default 0 not null	Intangible arroba
SUMATORIA_DIAS_ATRASO	NUMBER(18) default 0 not null	Sumatoria días atrasado
PROMOCION_CUOTAS_GRATIS	VARCHAR2(1)	Tiene la promoción de las cuotas gratis Si o No
COBRADOR_ASIGNADO_MES_SIG	VARCHAR2(100)	Cobrador asignado mes siguiente del balance
OPERADOR_ASIGNADO_MES_SIG	VARCHAR2(100)	Operador asignado mes siguiente del balance
SALDO_ACTUAL_HOY_PRIMER_VENCIM	NUMBER(18,2) default 0 not null	Saldo actual a hoy del primer vencimiento
SCORE	NUMBER(6)	Score
SEGMENTACION	VARCHAR2(20)	Segmentación



Tabla 3.26 Forma_pago_comprobante

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
TIPO_COMPROBANTE	VARCHAR2(2) not null	Tipo de comprobante
COD_COMPROBANTE	VARCHAR2(9) not null	Código del comprobante
COD_FORMA_PAGO	VARCHAR2(3) not null	Código forma de venta
SECUENCIA	NUMBER(4) not null	Número de vencimiento
VENCIMIENTO	DATE not null	Fecha de vencimiento
VALOR	NUMBER(14,2) not null	Valor de la cuota
CONCEPTO	VARCHAR2(40)	Concepto
USERIDC	VARCHAR2(3) not null	Código del usuario
DOCUMENTO	VARCHAR2(10) not null	Código del documento de liquidación
SALDO	NUMBER(14,2) not null	Saldo de la cuota
ESTADO	VARCHAR2(1) not null	Estado de la cuota
DEBITO_CREDITO	NUMBER(1) not null	Débito o crédito
COD_TIPO_PERSONA	VARCHAR2(3)	Código tipo de persona
COD_PERSONA	VARCHAR2(14)	Código de persona
NUMERO_DOCUMENTO	VARCHAR2(20)	Número de documento
DIAS_ATRAZO	NUMBER(6,2)	Días de atraso
COD_ESTADO_CUOTA	VARCHAR2(3)	Código estado de la cuota

Tabla 3.27 St_cancelacion_d

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
SECUENCIA	NUMBER(4) not null	Secuencia del registro del comprobante
TIPO_COMPROBANTE	VARCHAR2(2) not null	Tipo de comprobante de cancelación
COD_COMPROBANTE	VARCHAR2(9) not null	Código de comprobante de cancelación
TIPO_COMPROBANTE_FP	VARCHAR2(2)	Tipo de comprobante de la factura
COD_COMPROBANTE_FP	VARCHAR2(9)	Código de comprobante de la factura
COD_FORMA_PAGO_FP	VARCHAR2(3)	Código de la forma de venta de la factura
SECUENCIA_FP	NUMBER(4)	Número de vencimiento de la factura
COD_TIPO_CANCELACION	VARCHAR2(2) not null	Código tipo de cancelación



COD_DIVISA	VARCHAR2(20) not null	Código de divisa
FECHA	DATE not null	Fecha del cobro
DEBITO_CREDITO	NUMBER(1) not null	Débito o Crédito
VALOR	NUMBER(14,2) not null	Valor del cobro
DOCUMENTO	VARCHAR2(20)	Código del documento de liquidación

Tabla 3.28 Movimiento

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
TIPO_COMPROBANTE	VARCHAR2(2) not null	Tipo de comprobante
COD_COMPROBANTE	VARCHAR2(9) not null	codigo del comprobante
SECUENCIA	NUMBER(4) not null	secuencia de cada comprobante
COD_PRODUCTO	VARCHAR2(14) not null	código del producto
CANTIDAD	NUMBER(14,2) not null	cantidad
DEBITO_CREDITO	NUMBER(1) not null	débito o crédito
PRECIO	NUMBER(14,2) not null	precio
DESCUENTO	NUMBER(14,2) not null	descuento
COSTO	NUMBER(17,6)	costo
BODEGA	NUMBER(4) not null	codigo de bodega
IVA	NUMBER(14,2) not null	valor del iva
FECHA	DATE not null	fecha de la transacción
FACTURA_MANUAL	VARCHAR2(9) not null	numero de factura manual
COD_UNIDAD	VARCHAR2(8) not null	codigo de unidad
ANULADO	VARCHAR2(1) not null	Estado de la transacción
TOTAL_LINEA	NUMBER(14,2) not null	Total de la línea
PORCE_DESCUENTO	NUMBER(5,2)	Porcentaje de descuento
ES_SERIE	NUMBER(1)	Tiene serie sí o no
COD_ESTADO_PRODUCTO	VARCHAR2(2) not null	Código estado del producto
COD_TIPO_INVENTARIO	NUMBER(2) not null	Código Tipo Inventario
COD_PROMOCION	NUMBER(10)	Código de Promoción
COD_TIPO_CANAL	NUMBER(4)	Código Tipo Canal
DESCUENTO_REGALO	NUMBER(14,2)	Descuento Regalo



Tabla 3.29 Cc_siembra_x_vencimiento

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
ANIO_MES_VENCIMIENTO	NUMBER(6) not null	Año y Mes del Vencimiento
ANIO_MES_VENTA	NUMBER(6) not null	Año y Mes de la Venta
CODIGO_AGENCIA	NUMBER(4) not null	Código de agencia
CODIGO_CANTON	VARCHAR2(4) not null	Código del Cantón
CODIGO_PROVINCIA	VARCHAR2(6) not null	Código de Provincia
CODIGO_REGION	NUMBER(3) not null	Código de la Región
CODIGO_POLITICA	NUMBER(2) not null	Código de la Política
COD_MODELO_CAT_LINEA	VARCHAR2(8) not null	Código de la línea
COD_ITEM_CAT_LINEA	VARCHAR2(3) not null	Código de la línea
COD_MARCA	NUMBER(3) not null	Código de Marca
CODIGO_NACION	NUMBER(3) not null	Código de Nación
CODIGO_EMPRESA	NUMBER(2) not null	Código de Empresa
NOMBRE_AGENCIA	VARCHAR2(50) not null	Nombre de la Agencia
NOMBRE_CANTON	VARCHAR2(50) not null	Nombre del Cantón
NOMBRE_PROVINCIA	VARCHAR2(100) not null	Nombre de la Provincia
NOMBRE_REGION	VARCHAR2(300) not null	Nombre de la Región
NOMBRE_LINEA	VARCHAR2(50) not null	Nombre de la Línea
NOMBRE_MARCA	VARCHAR2(50) not null	Nombre de la Marca
NOMBRE_POLITICA	VARCHAR2(100) not null	Nombre de la Política
VALOR_VENCIMIENTO	NUMBER(30,2) not null	Valor del vencimiento
FECHA_ULT_EJECUCION	DATE not null	Fecha de última ejecución
ADICIONADO_POR	VARCHAR2(30) not null	Quién adicionó el registro
FECHA_ADICION	DATE not null	Fecha en la que fue adicionado el registro
MODIFICADO_POR	VARCHAR2(30)	Quién modificó el registro
FECHA_MODIFICACION	DATE	Fecha de modificación del registro



Tabla 3.30 Forma_pago

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
COD_FORMA_PAGO	VARCHAR2(3) not null	Código de la forma de pago
NOMBRE	VARCHAR2(30) not null	Nombre de la forma de pago
CANCELA	VARCHAR2(1)	Código del Cantón
EFFECTIVO	VARCHAR2(1)	Define el tipo de pago (Efectivo, Cheque)

Tabla 3.31 Empresa

Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
EMPRESA	NUMBER(2) not null	Código de la empresa
NOMBRE	VARCHAR2(40) not null	Nombre de la empresa
PAIS	VARCHAR2(15) not null	Nombre del País donde radica la empresa
CIUDAD	VARCHAR2(20) not null	Nombre de la ciudad donde radica la empresa
DIRECCION	VARCHAR2(40) not null	Dirección donde radica la empresa.
TELEFONO1	NUMBER(8)	Teléfono de la empresa
TELEFONO2	NUMBER(8)	Teléfono 2 de la empresa
FAX	NUMBER(8)	Número del fax de la empresa
RUC	VARCHAR2(13) not null	Número del R.U.C. de la empresa.
MONEDA	VARCHAR2(20) not null	Nombre de la moneda que se utiliza
SIGNO_MONEDA	VARCHAR2(3) not null	Signo de la moneda
CASILLA	VARCHAR2(10)	Número de Casilla de la empresa
CONTADOR	VARCHAR2(40) not null	Nombres del Contador
NUMERO_PATRONAL	VARCHAR2(10)	Número Patronal
PROVINCIA	VARCHAR2(20) not null	Nombre de la provincia donde radica la empresa
CANTON	VARCHAR2(20) not null	Nombre del cantón donde radica la empresa
PARROQUIA	VARCHAR2(20)	Nombre de la parroquia donde radica la empresa



Tabla 3.32 Ad_agencias

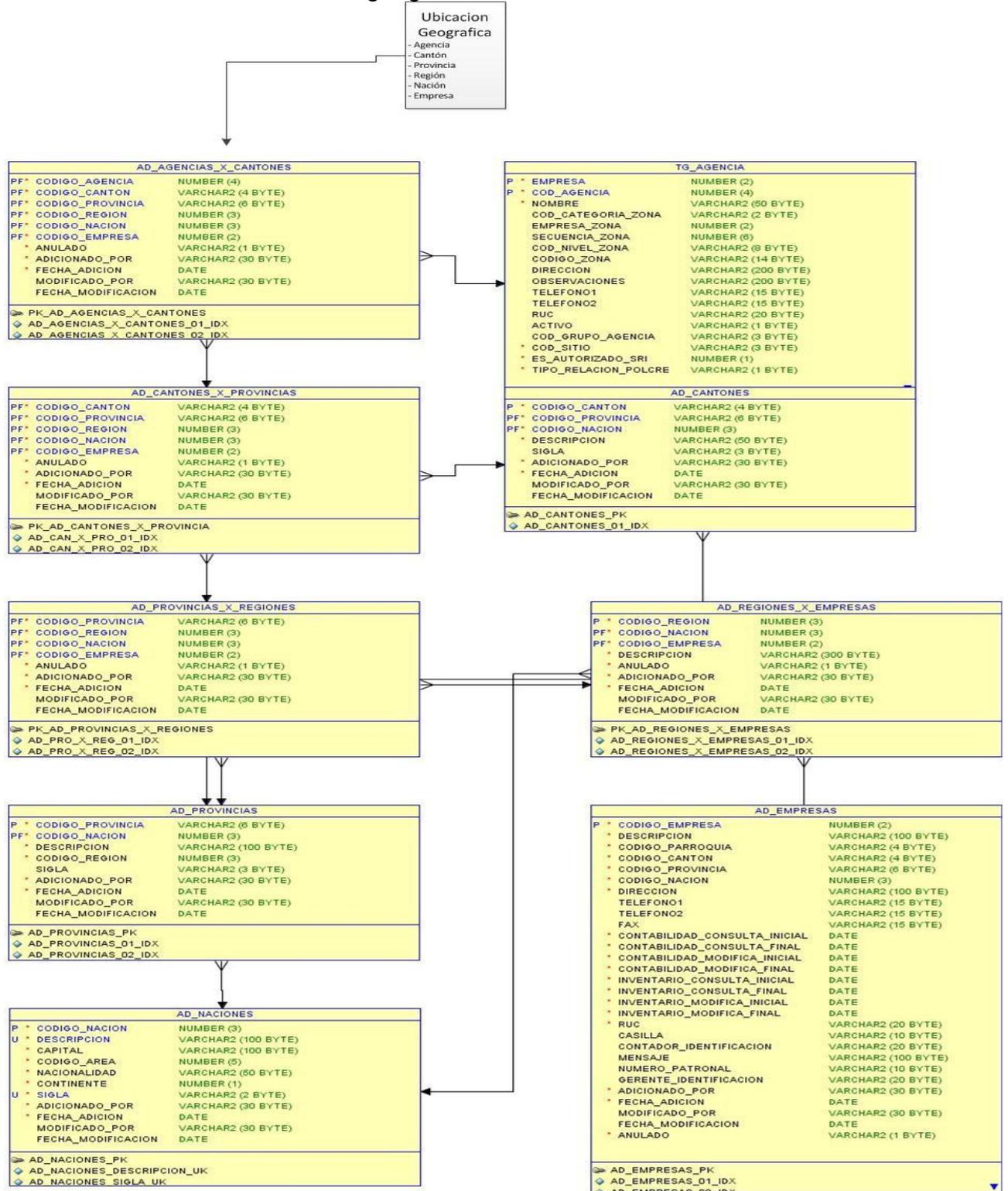
Nombre del Campo	Tipo y Tamaño del Campo	Descripción del campo
CODIGO_AGENCIA	NUMBER(4) not null	Código de la Agencia
CODIGO_EMPRESA	NUMBER(2) not null	Código de la empresa
DESCRIPCION	VARCHAR2(100) not null	Nombre de la Agencia
DIRECCION	VARCHAR2(100) not null	Dirección de la Agencia
TELEFONO1	VARCHAR2(15)	Teléfono 1 de la Agencia
TELEFONO2	VARCHAR2(15)	Teléfono 2 de la Agencia
OBSERVACIONES	VARCHAR2(200)	Código de la Política
EMAIL	VARCHAR2(100)	Correo electrónico de la agencia



Anexo A6.

Modelo Lógico de la estructura del datamart – Tabla de Dimensiones.

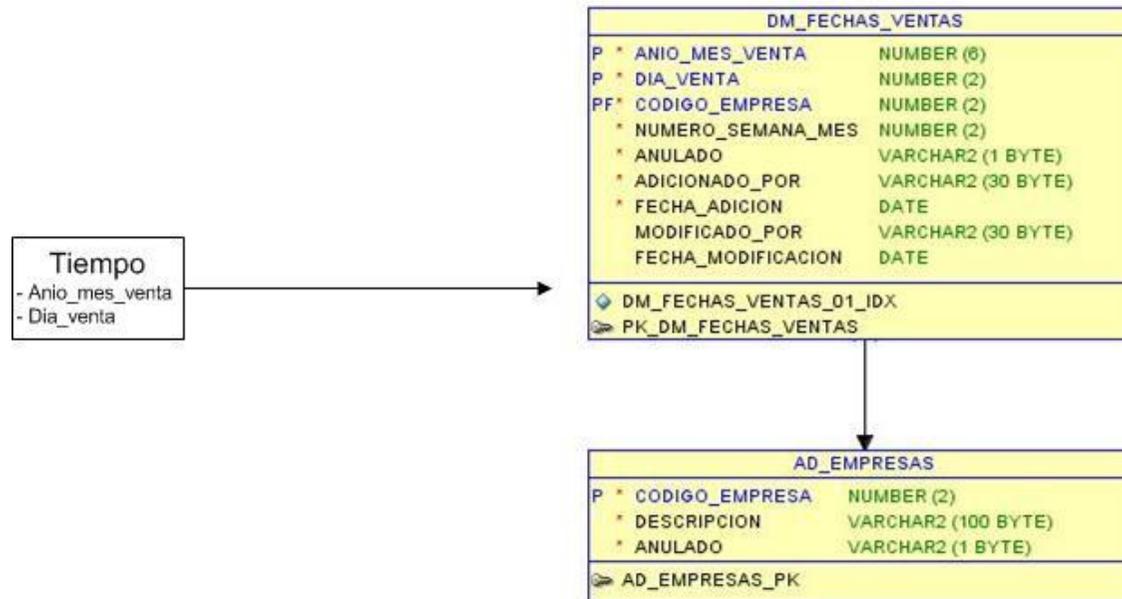
Figura 4.24 Tablas de dimensiones - Cumplimiento de ventas - Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



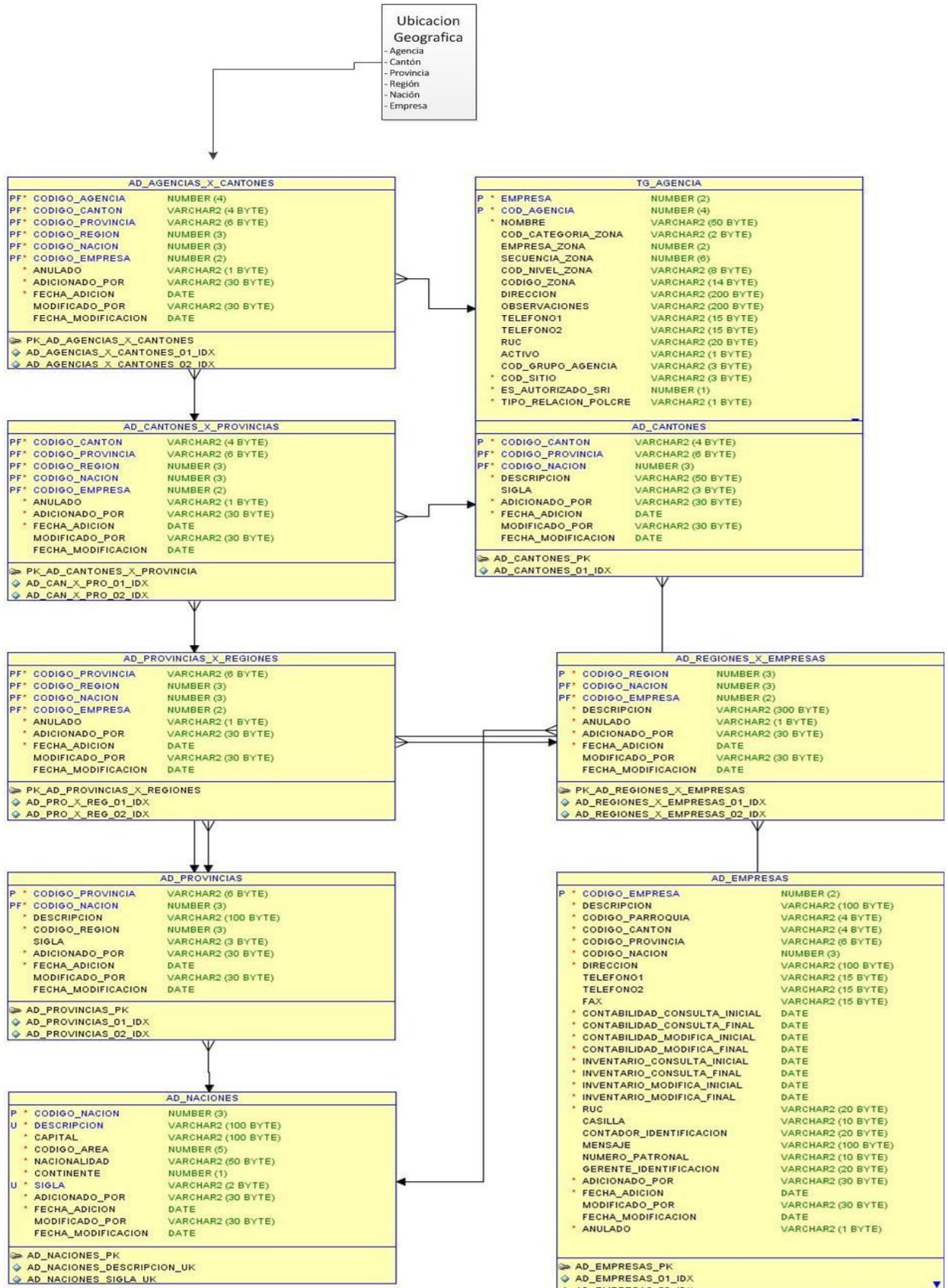
Figura 4.25 Tablas de dimensiones - Cumplimiento de ventas - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



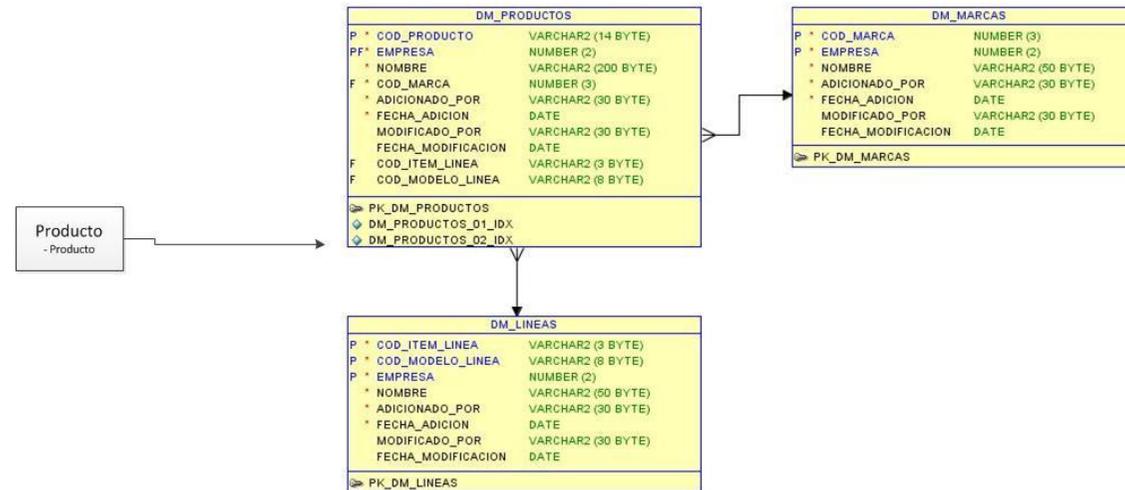
Figura 4.26 Tablas de dimensiones – Rentabilidad - Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.27 Tablas de dimensiones – Rentabilidad - Producto.

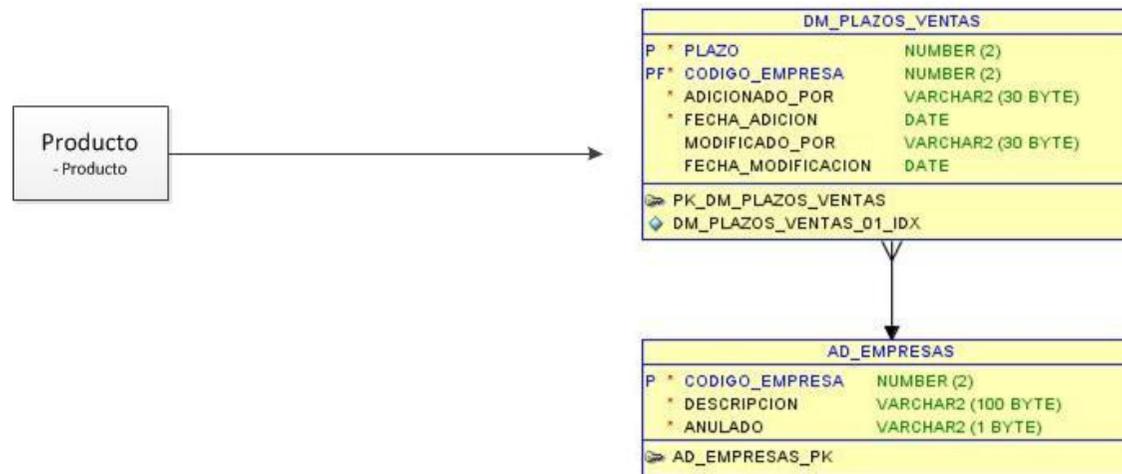


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



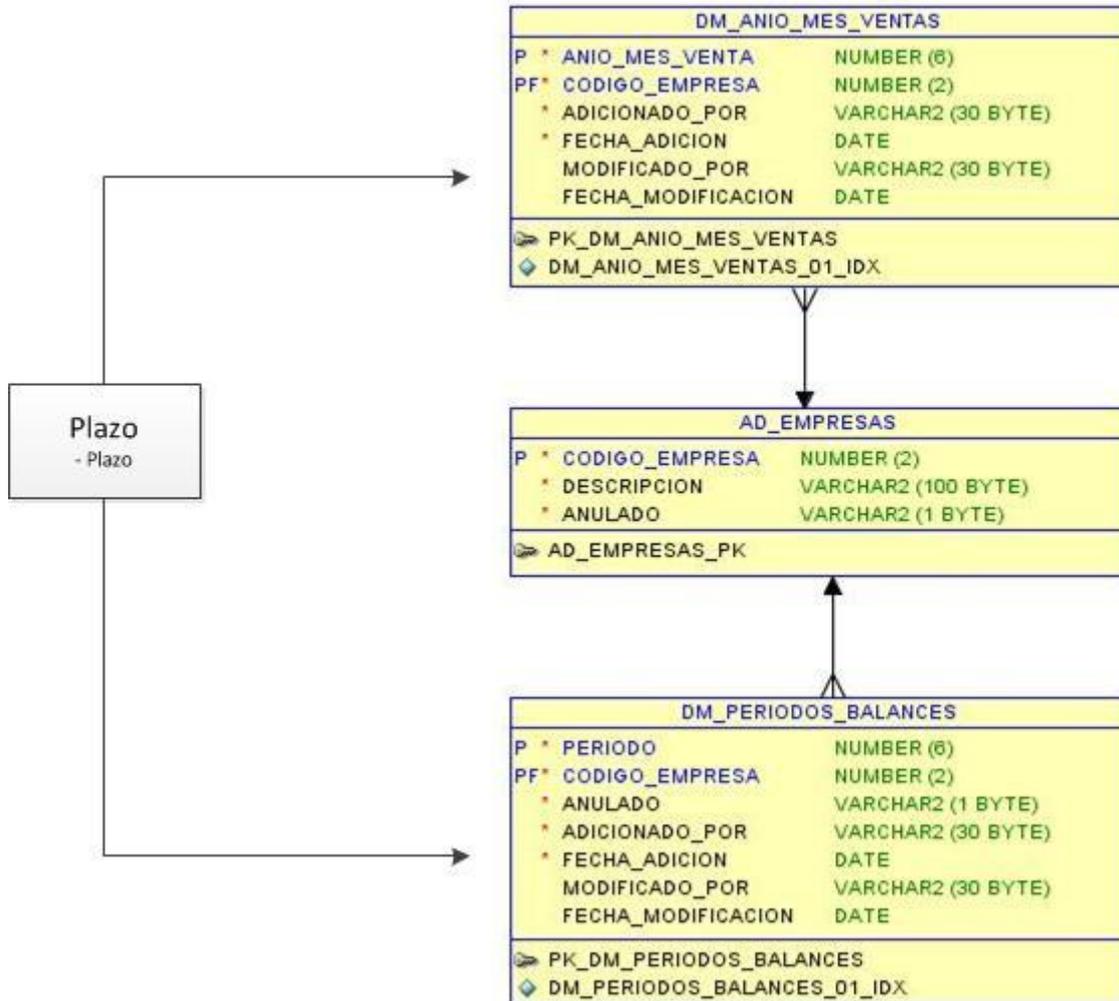
Figura 4.28 Tablas de dimensiones – Rentabilidad - Plazo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



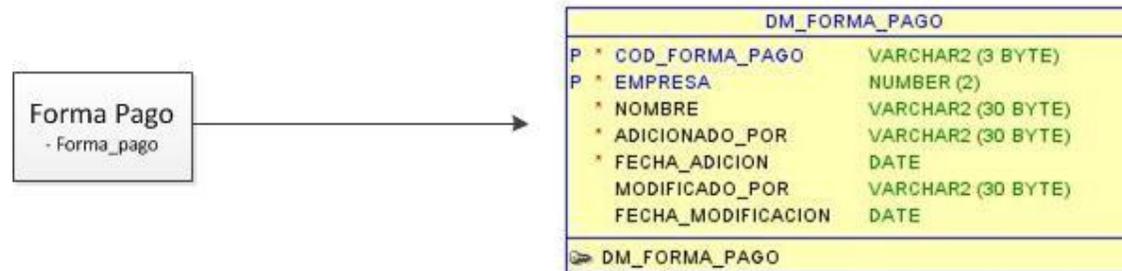
Figura 4.29 Tablas de dimensiones – Rentabilidad - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



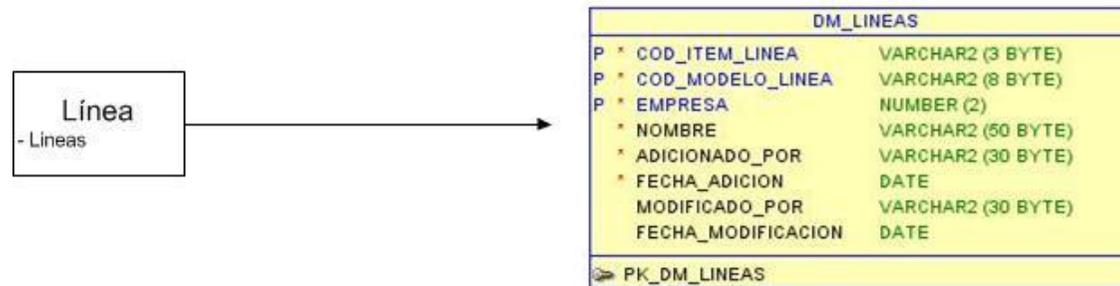
Figura 4.30 Tablas de dimensiones – Rentabilidad – Forma pago.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



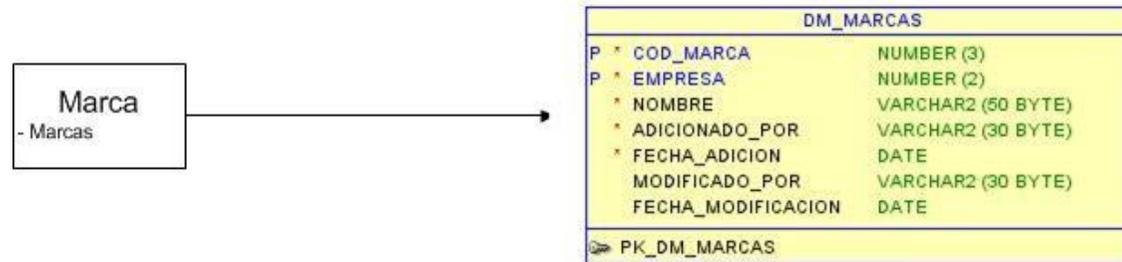
Figura 4.31 Tablas de dimensiones – Rentabilidad – Línea.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.32 Tablas de dimensiones – Rentabilidad – Marca.

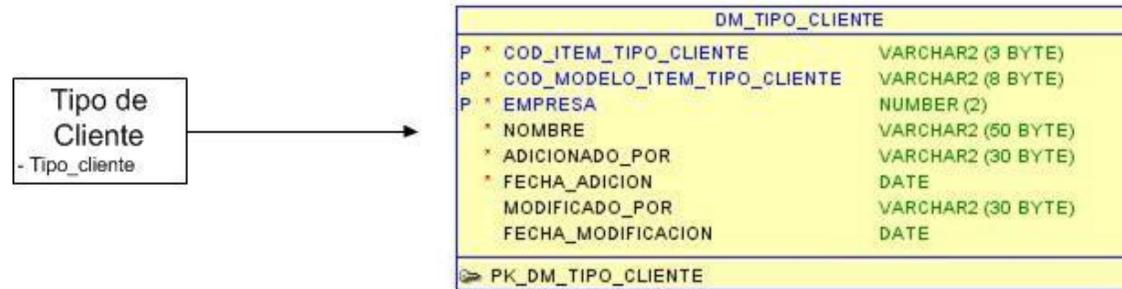


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.33 Tablas de dimensiones – Rentabilidad – Tipo de cliente.



Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



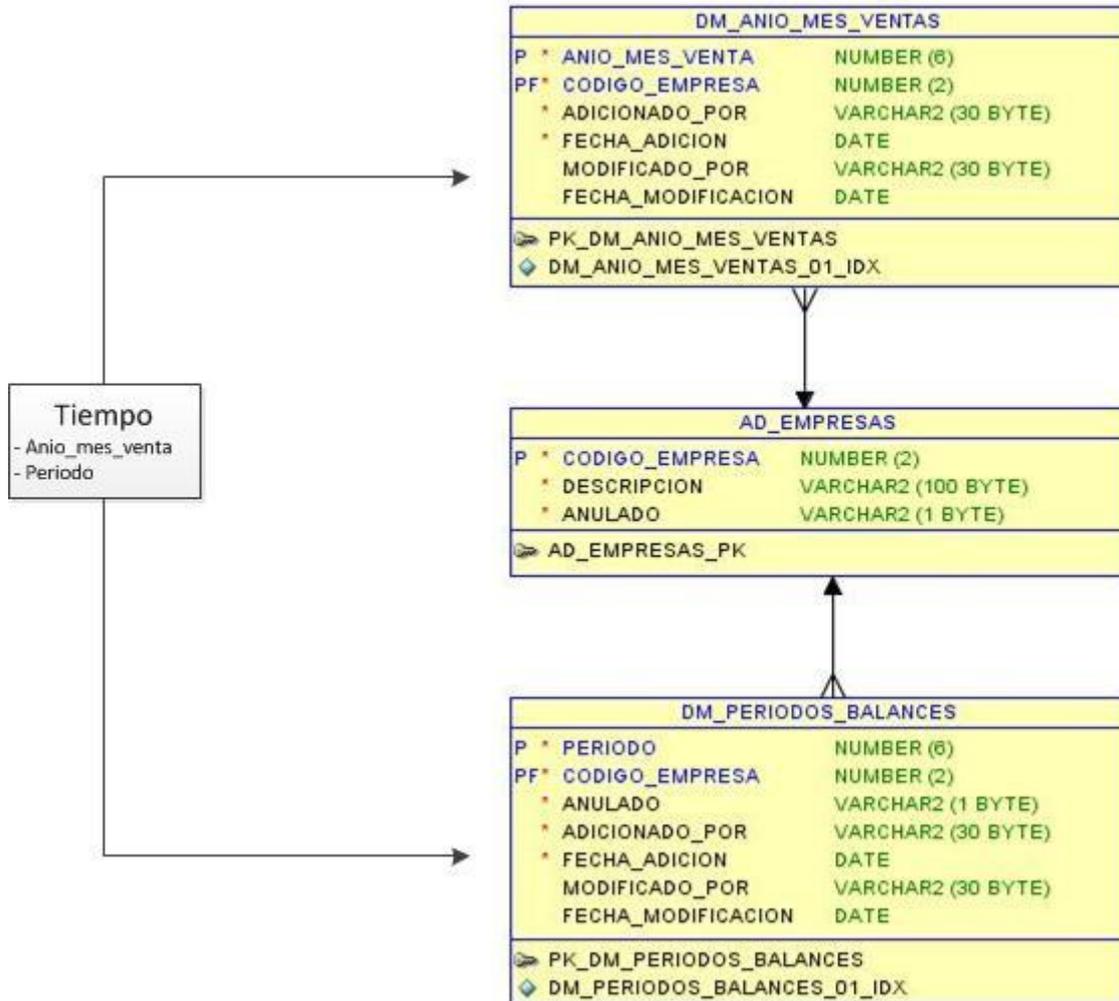
Figura 4.34 Tablas de dimensiones – Morosidad de la cartera – Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



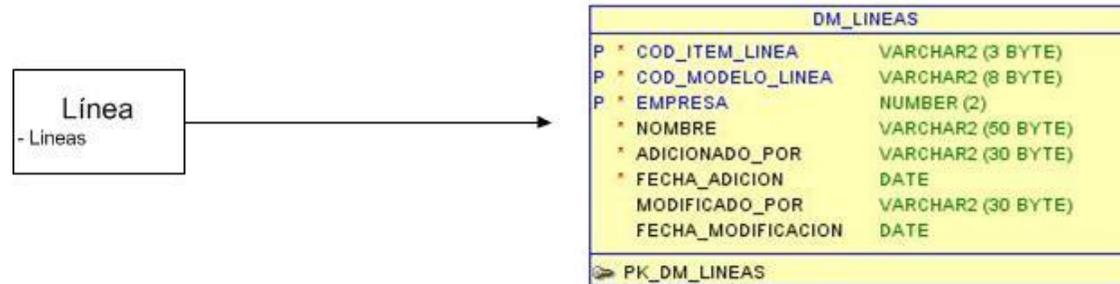
Figura 4.35 Tablas de dimensiones - Morosidad de la cartera - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



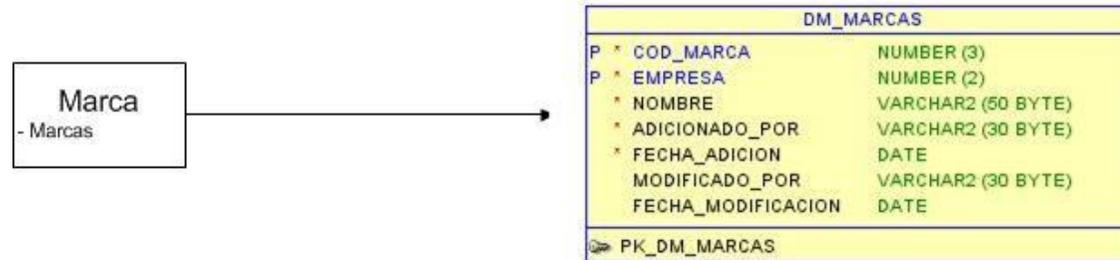
Figura 4.36 Tablas de dimensiones – Morosidad de la cartera - Línea.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



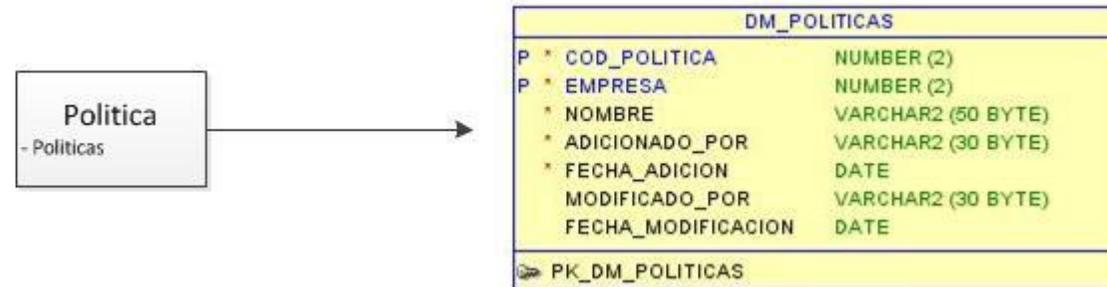
Figura 4.37 Tablas de dimensiones – Morosidad de la cartera - Marca.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.38 Tablas de dimensiones – Morosidad de la cartera - Política.



Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



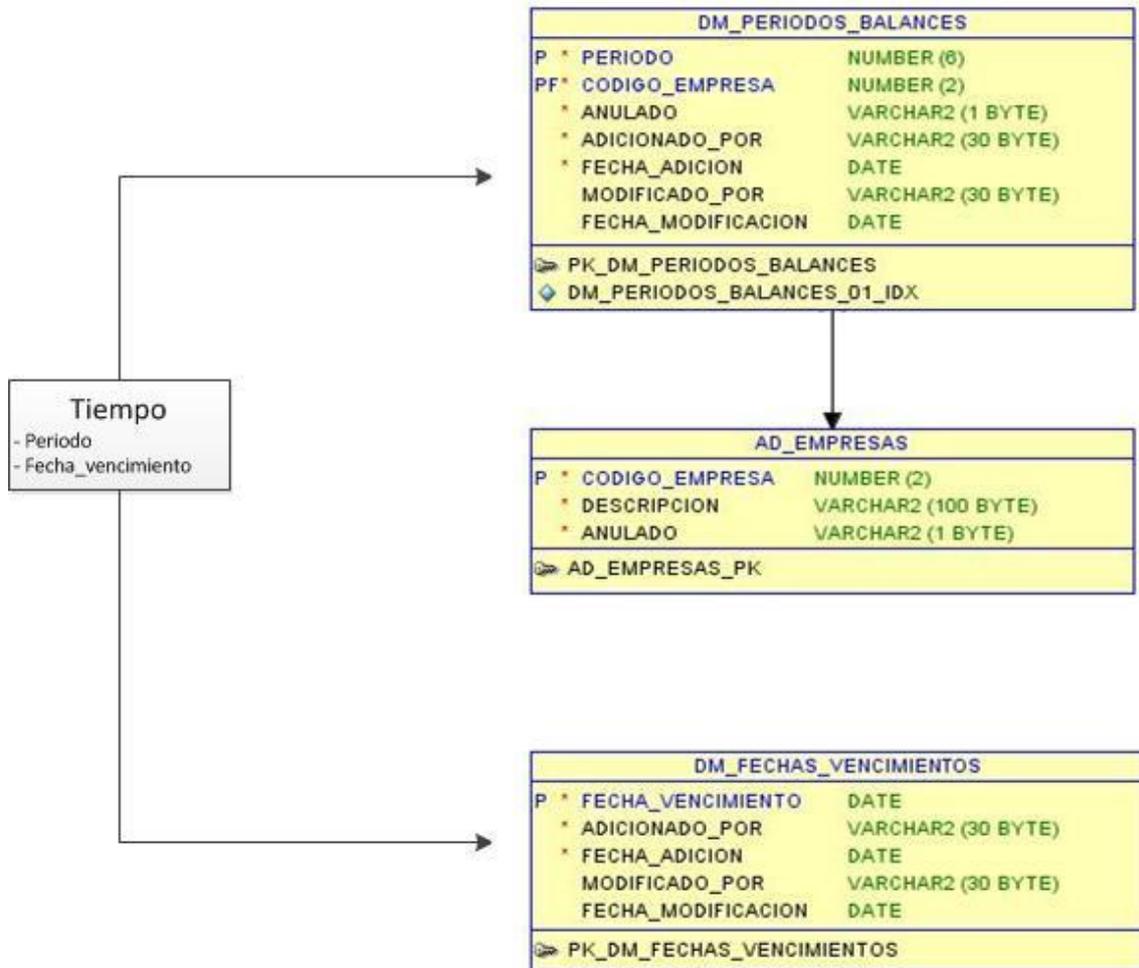
Figura 4.39 Tablas de dimensiones – Flujo corriente – Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



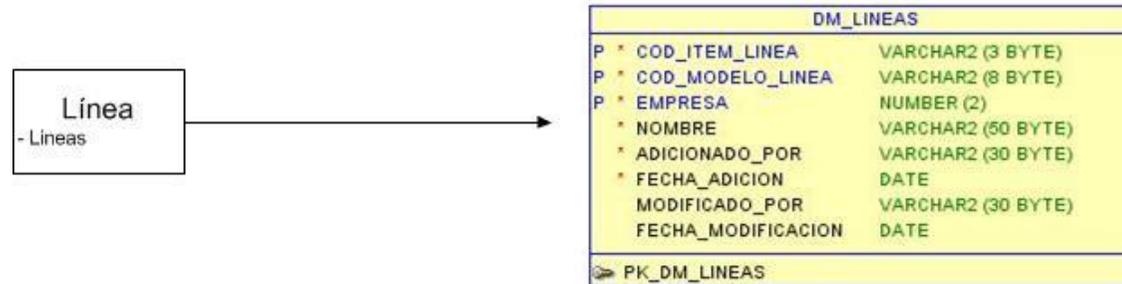
Figura 4.40 Tablas de dimensiones - Flujo corriente - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



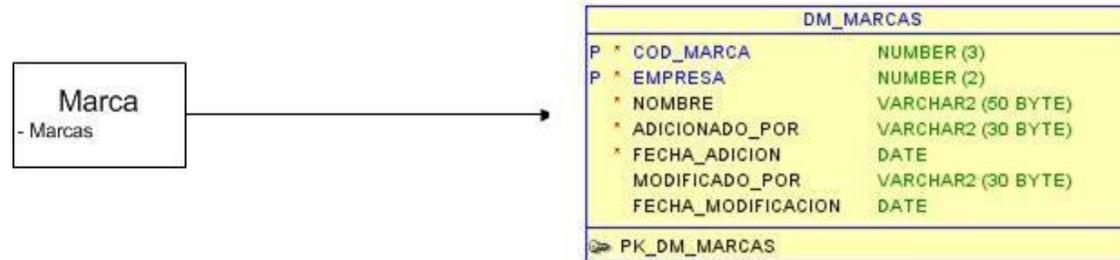
Figura 4.41 Tablas de dimensiones – Flujo corriente - Línea.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



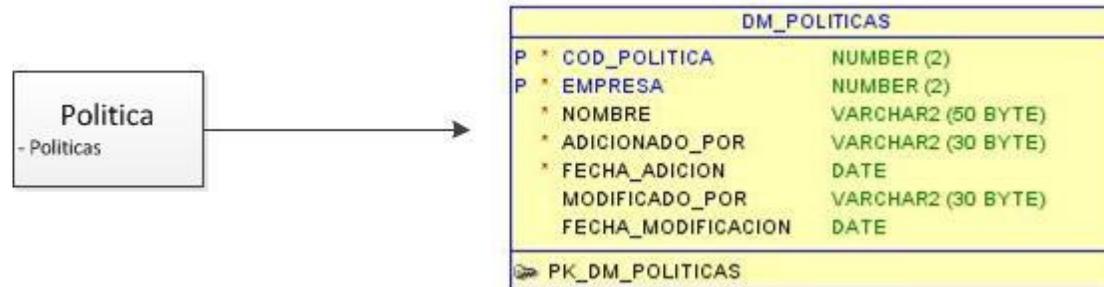
Figura 4.42 Tablas de dimensiones – Flujo corriente - Marca.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.43 Tablas de dimensiones – Flujo corriente - Política.

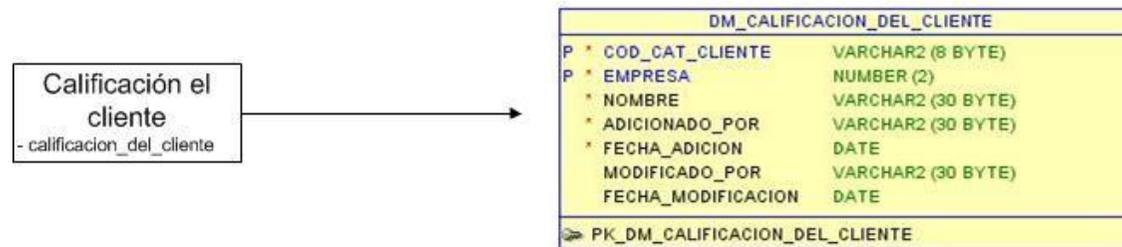


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.44 Tablas de dimensiones – Flujo corriente – Calificación cliente.

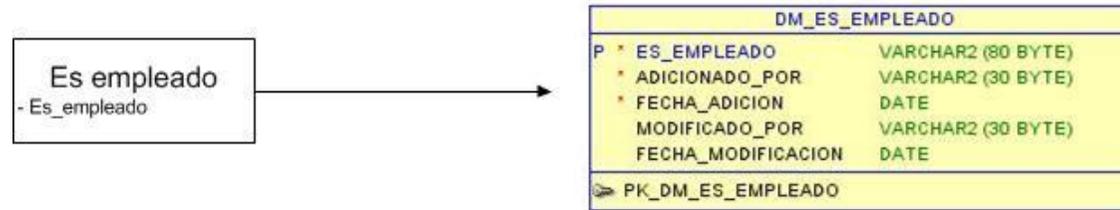


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



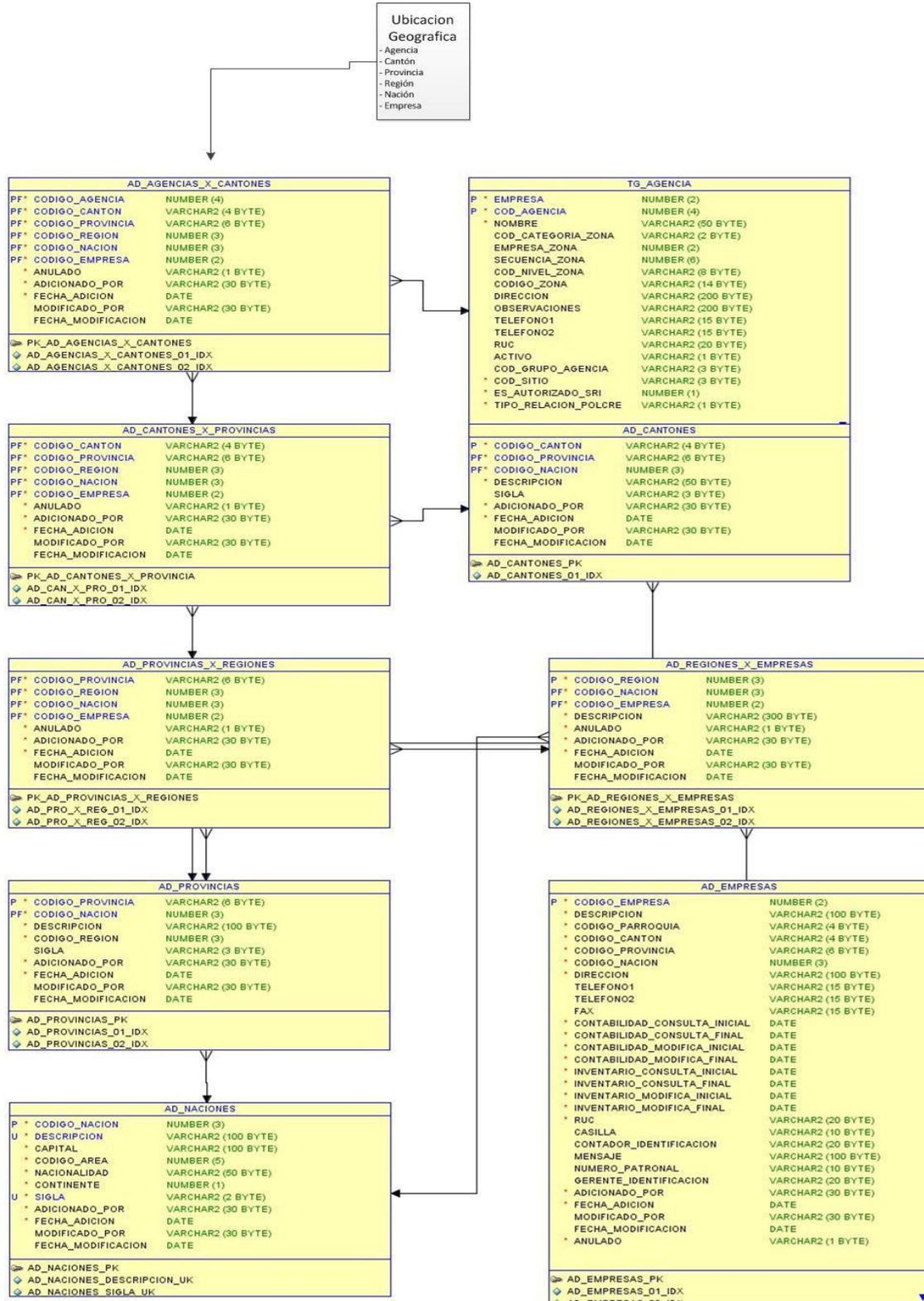
Figura 4.45 Tablas de dimensiones – Flujo corriente – Es empleado.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



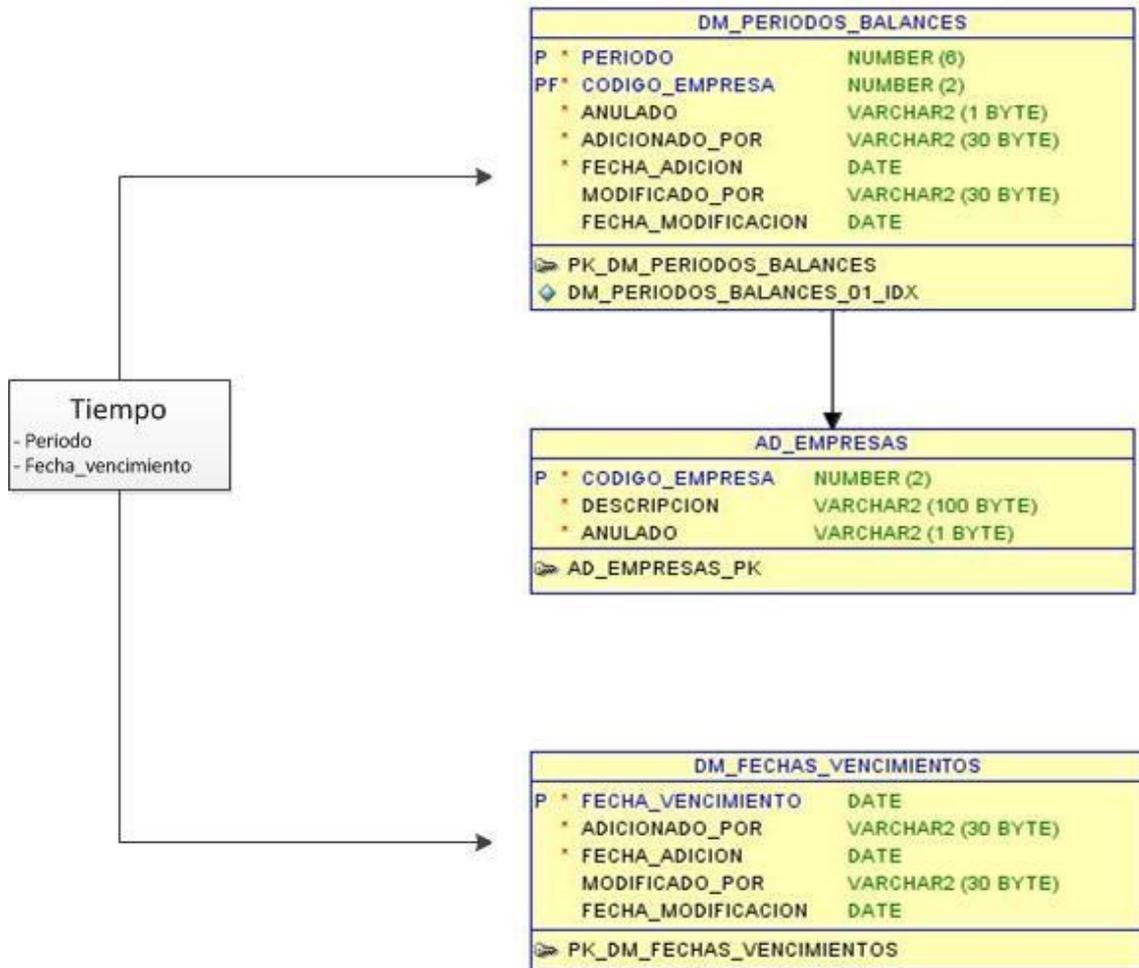
Figura 4.46 Tablas de dimensiones – Mes presente puntual – Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



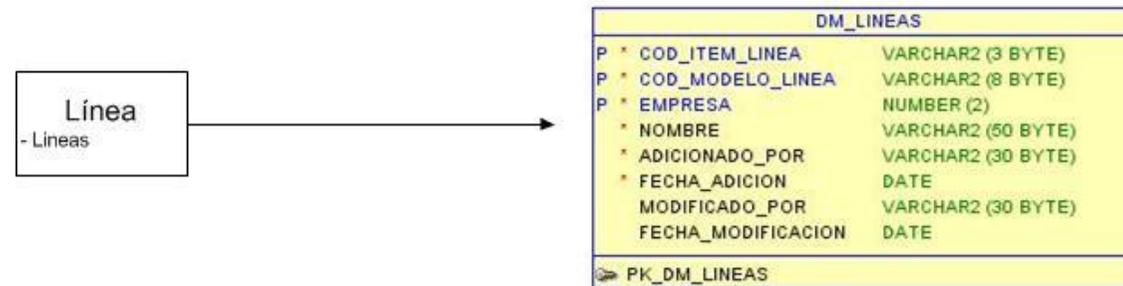
Figura 4.47 Tablas de dimensiones - Mes presente puntual - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.48 Tablas de dimensiones – Mes presente puntual - Línea.

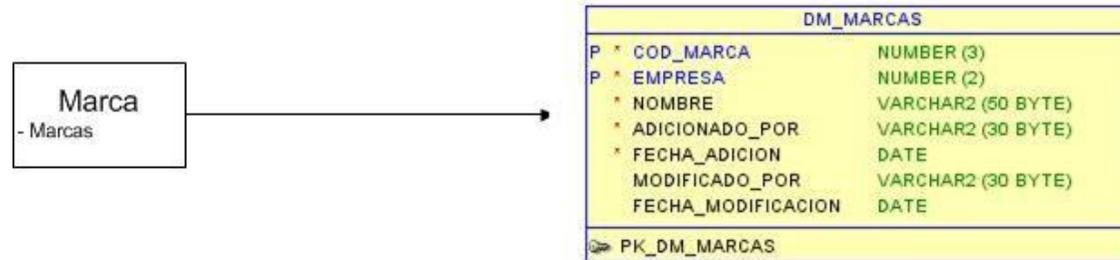


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



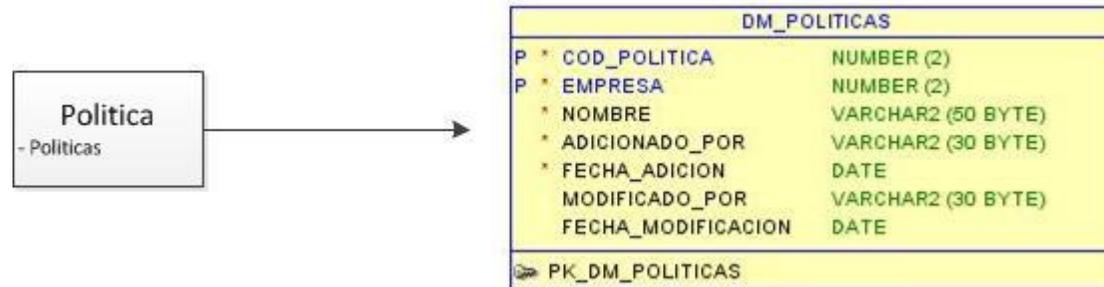
Figura 4.49 Tablas de dimensiones – Mes presente puntual - Marca.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.50 Tablas de dimensiones – Mes presente puntual - Política.

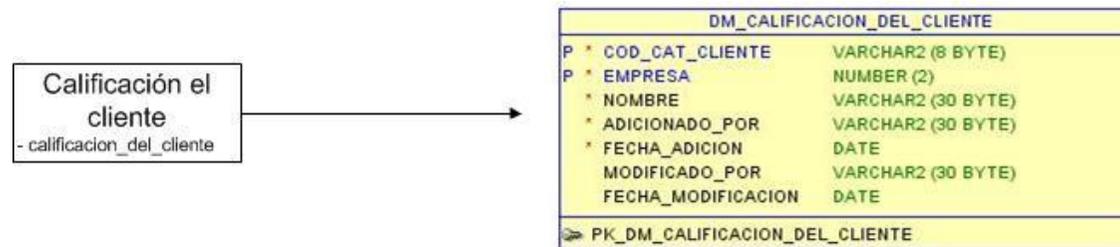


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.51 Tablas de dimensiones – Mes presente puntual – Calificación cliente.

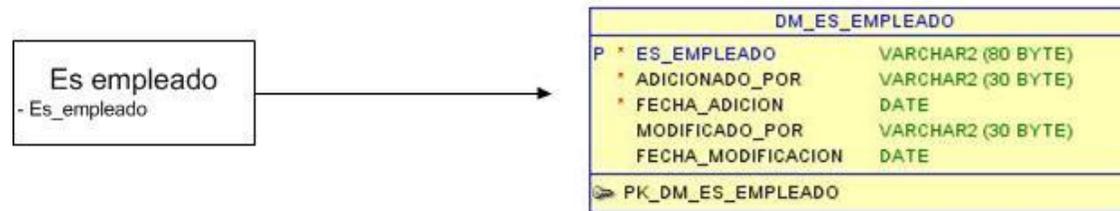


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.52 Tablas de dimensiones – Mes presente puntual – Es empleado.

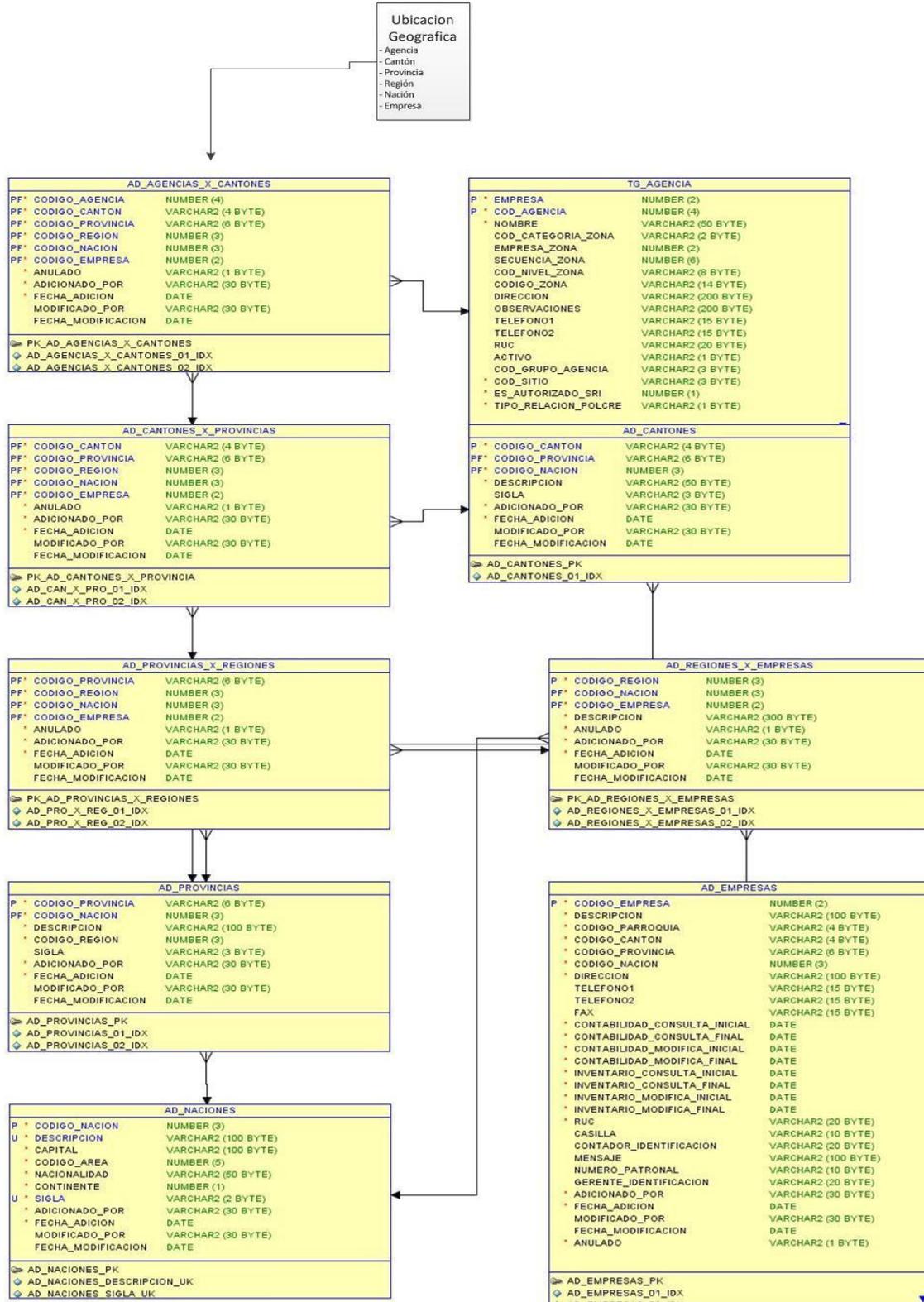


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



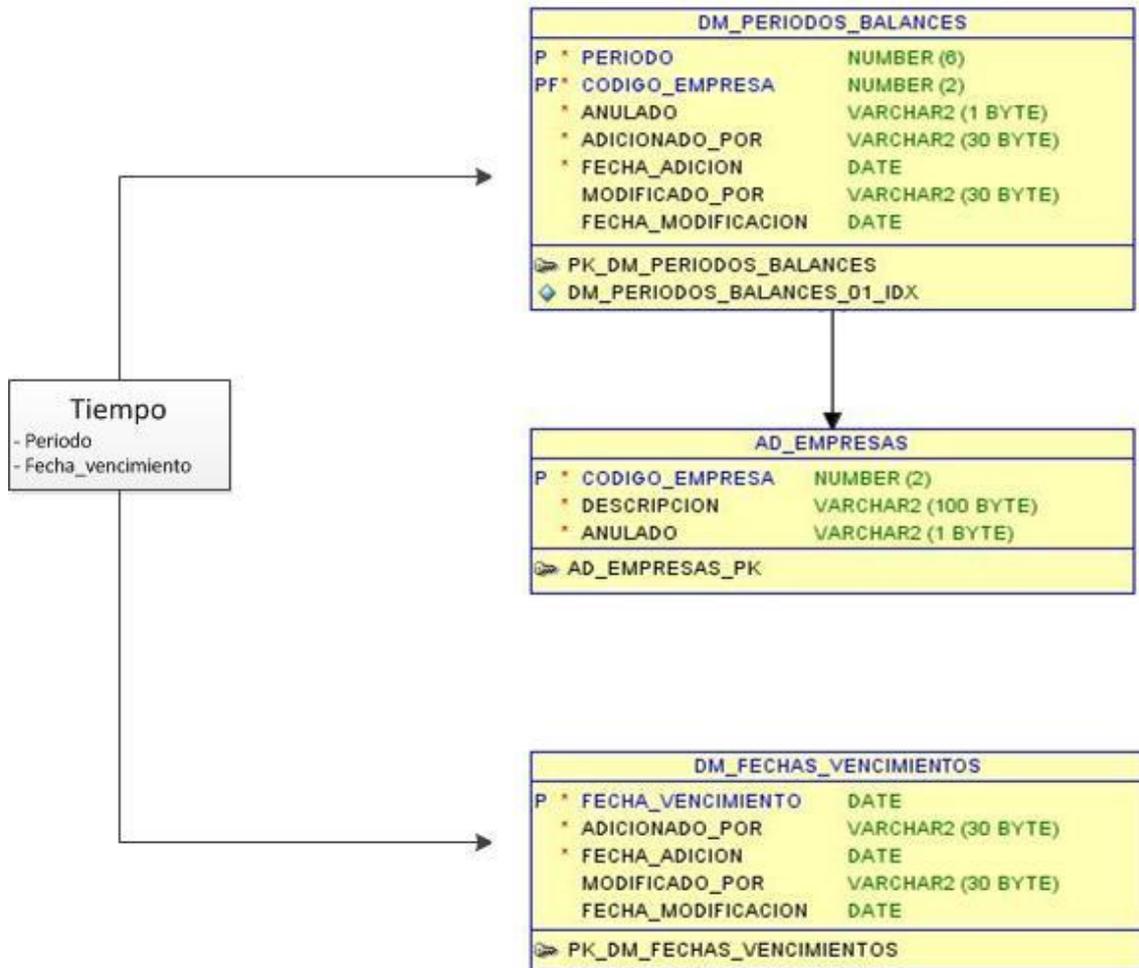
Figura 4.53 Tablas de dimensiones – Mes presente total – Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



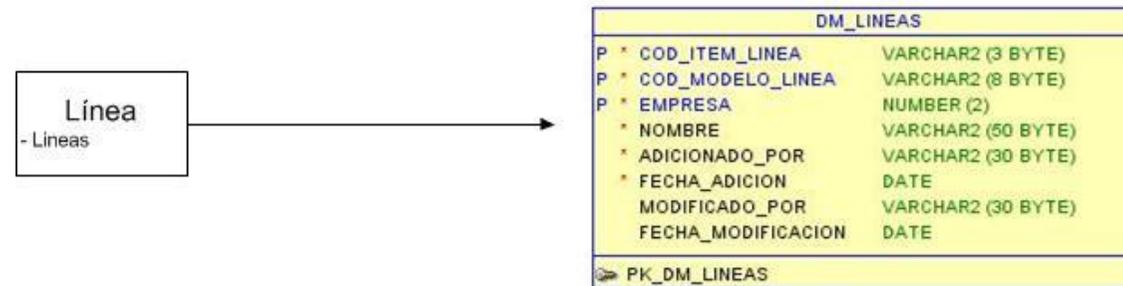
Figura 4.54 Tablas de dimensiones - Mes presente total - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



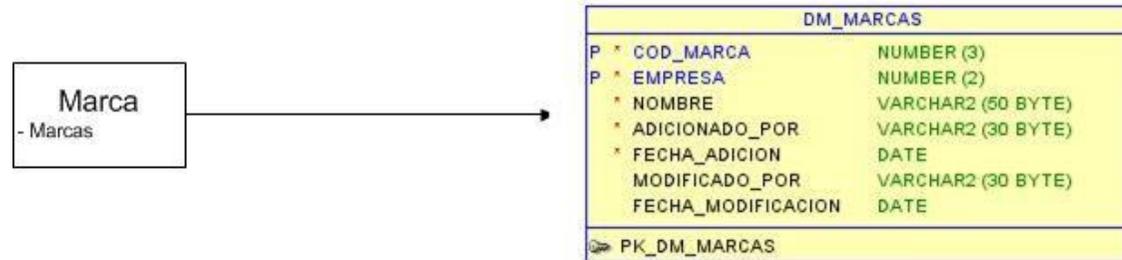
Figura 4.55 Tablas de dimensiones – Mes presente total - Línea.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



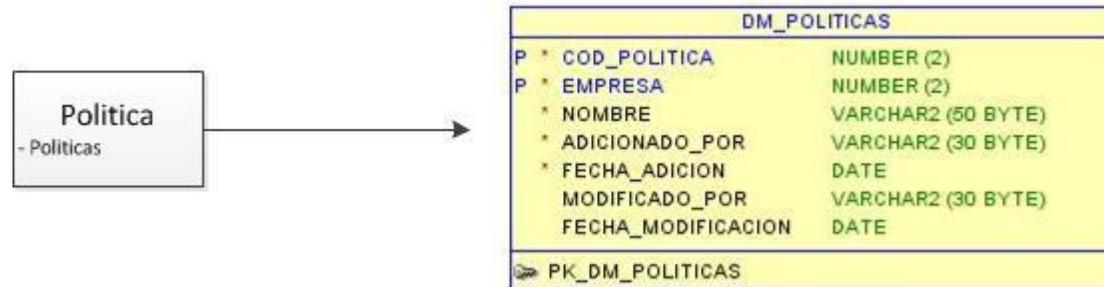
Figura 4.56 Tablas de dimensiones – Mes presente total - Marca.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.57 Tablas de dimensiones – Mes presente total - Política.

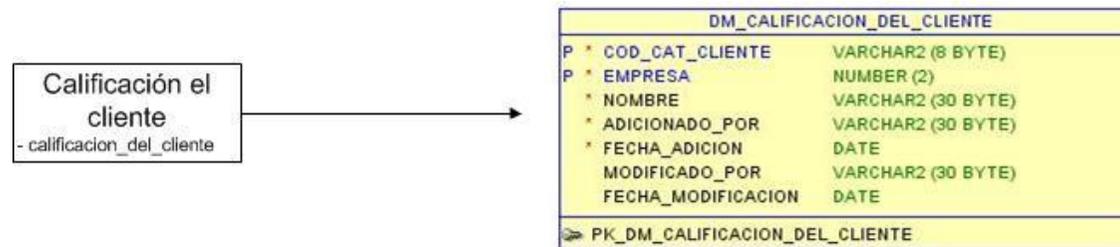


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.58 Tablas de dimensiones – Mes presente total – Calificación cliente.

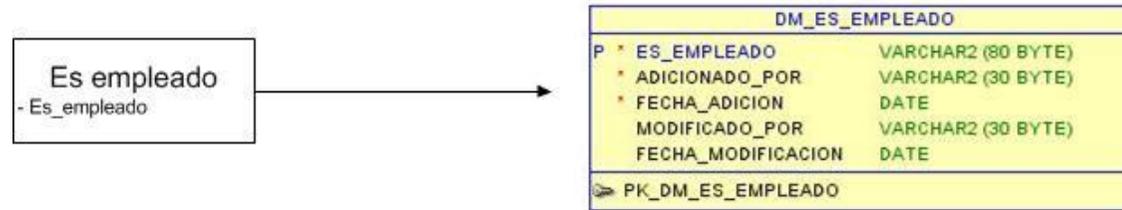


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.59 Tablas de dimensiones – Mes presente total – Es empleado.



Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



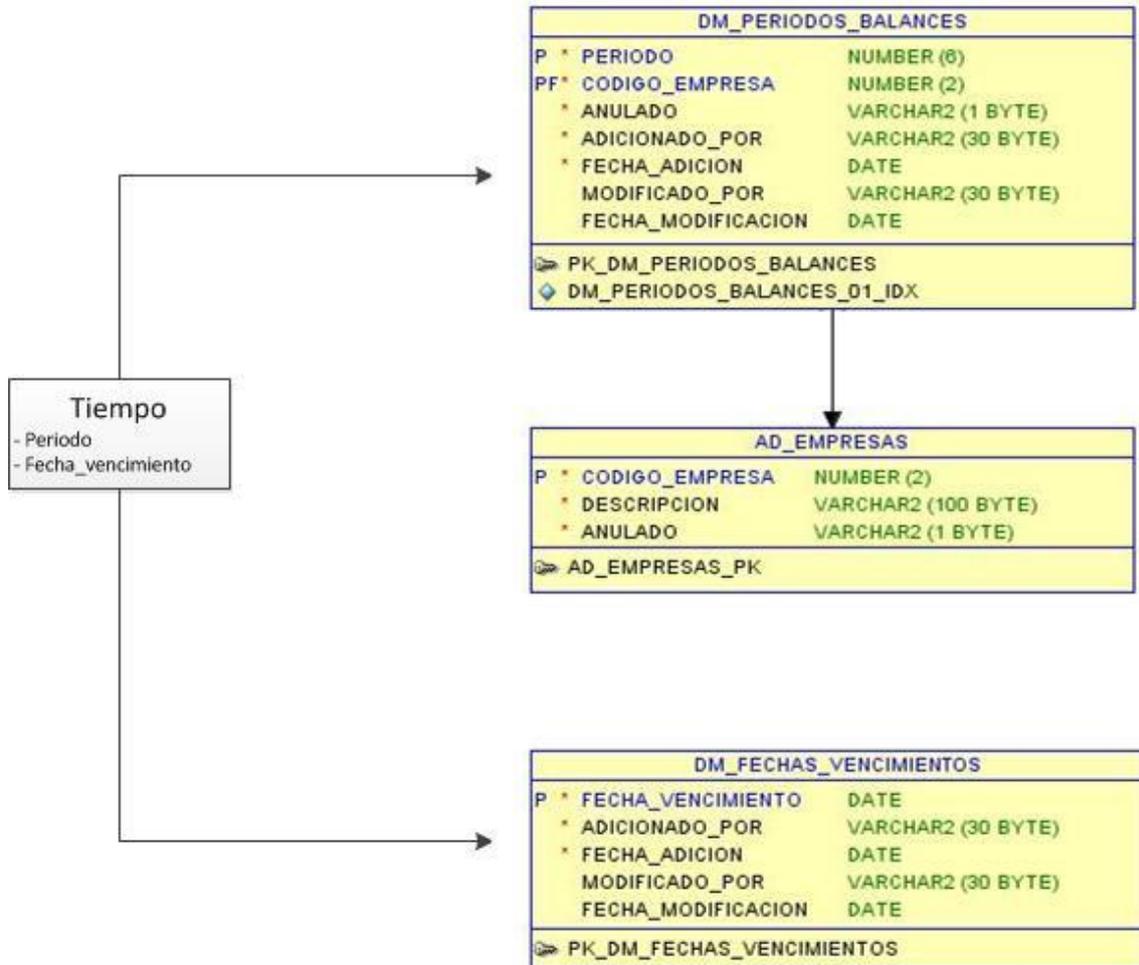
Figura 4.60 Tablas de dimensiones – Primeras cuotas – Ubicación geográfica.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



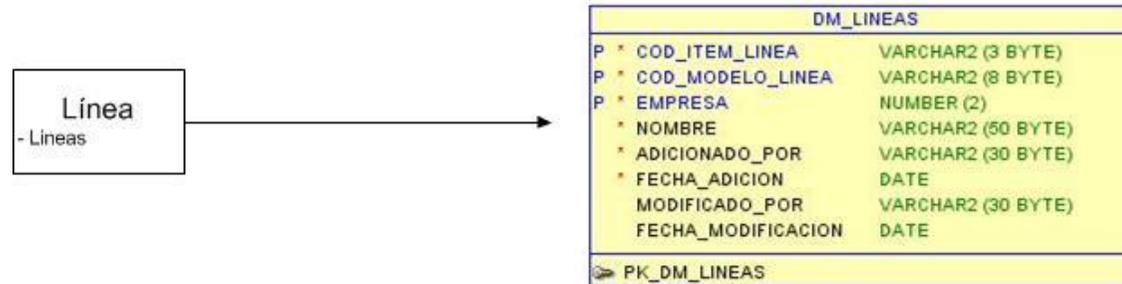
Figura 4.61 Tablas de dimensiones - Primeras cuotas - Tiempo.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



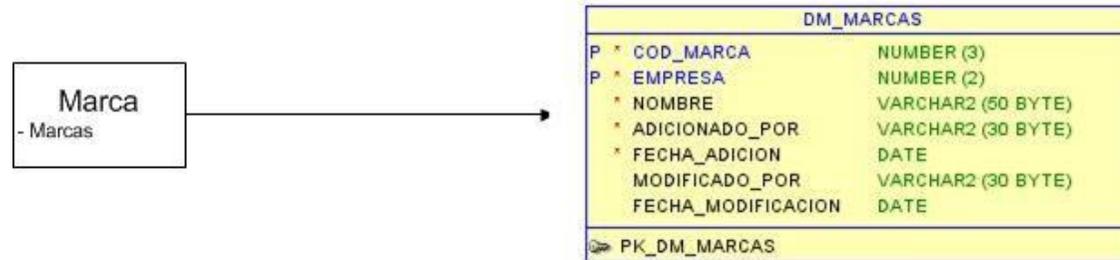
Figura 4.62 Tablas de dimensiones – Primeras cuotas - Línea.



Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.63 Tablas de dimensiones – Primeras cuotas - Marca.

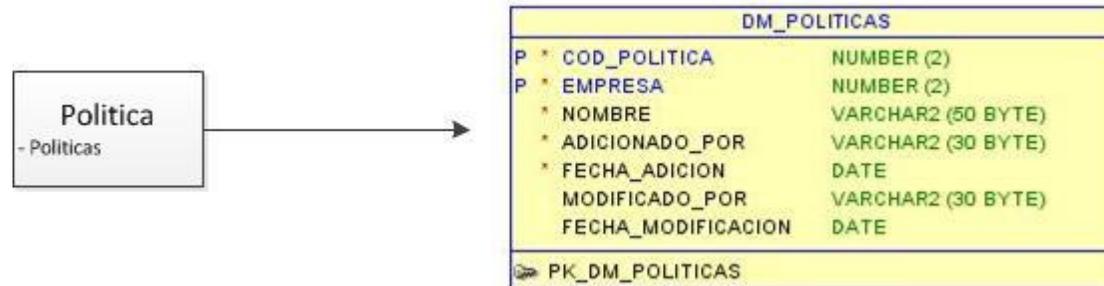


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.64 Tablas de dimensiones – Primeras cuotas - Política.

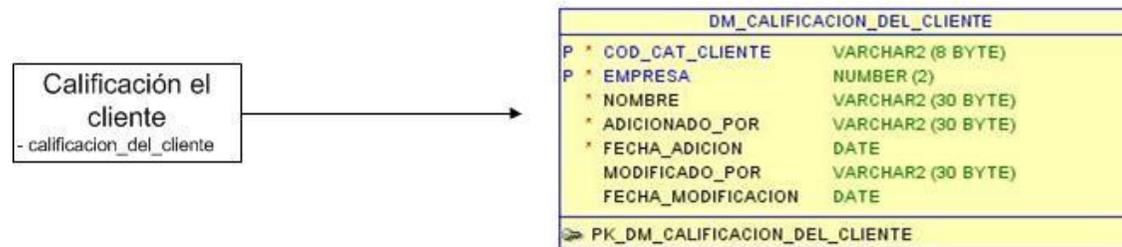


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.65 Tablas de dimensiones – Primeras cuotas – Calificación cliente.

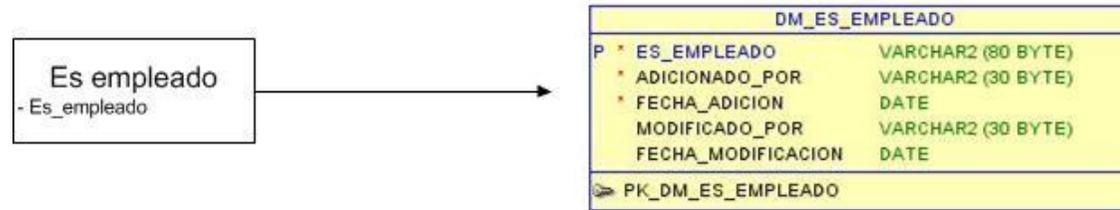


Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.66 Tablas de dimensiones – Primeras cuotas – Es empleado.



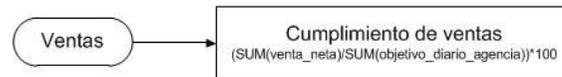
Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Anexo A7.

Modelo Lógico de la estructura del datamart – Tabla de Hechos.

Figura 4.67 Tablas de hechos – Cumplimiento de ventas.



DM_INDICE_CUMPLIMIENTO_VENTAS	
P * ANIO_MES_VENTA	NUMBER (6)
P * DIA_VENTA	NUMBER (2)
P * CODIGO_AGENCIA	NUMBER (4)
P * CODIGO_CANTON	VARCHAR2 (4 BYTE)
P * CODIGO_PROVINCIA	VARCHAR2 (6 BYTE)
P * CODIGO_REGION	NUMBER (3)
P * CODIGO_NACION	NUMBER (3)
P * CODIGO_EMPRESA	NUMBER (2)
* NOMBRE_AGENCIA	VARCHAR2 (50 BYTE)
* NOMBRE_CANTON	VARCHAR2 (50 BYTE)
* NOMBRE_PROVINCIA	VARCHAR2 (100 BYTE)
* NOMBRE_REGION	VARCHAR2 (300 BYTE)
* VALOR_VENTA_NETA	NUMBER (30,2)
* OBJETIVO_DIARIO_AGENCIA	NUMBER (30,2)
* PORCENTAJE_CUMPLIMIENTO_VTA	NUMBER (20,5)
* MES_CERRADO	VARCHAR2 (1 BYTE)
* FECHA_ULT_EJECUCION	DATE
* ADICIONADO_POR	VARCHAR2 (30 BYTE)
* FECHA_ADICION	DATE
MODIFICADO_POR	VARCHAR2 (30 BYTE)
FECHA_MODIFICACION	DATE

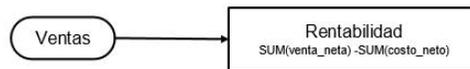
PK_DM_IND_CUM_VEN
DM_IND_CUM_VEN_01_IDX
DM_IND_CUM_VEN_02_IDX

Fuente: Impocomjaher Cía. Ltda.

Elaboración: El autor.



Figura 4.68 Tablas de hechos – Rentabilidad.



JAHER_DM_INDICE_RENTABILIDAD		
P	PERIODO	NUMBER (9)
P	COD_PRODUCTO	VARCHAR2 (14 BYTE)
P	ANIO_MES_VENTA	NUMBER (6)
P	FORMA_PAGO	VARCHAR2 (3 BYTE)
P	PLAZO	NUMBER (3)
P	CODIGO_AGENCIA	NUMBER (4)
P	CODIGO_CANTON	VARCHAR2 (4 BYTE)
P	CODIGO_PROVINCIA	VARCHAR2 (8 BYTE)
P	CODIGO_REGION	NUMBER (3)
P	COD_MODELO_CAT_LINEA	VARCHAR2 (8 BYTE)
P	COD_ITEM_CAT_LINEA	VARCHAR2 (3 BYTE)
P	COD_MARCA	NUMBER (3)
P	TIPO_CLIENTE	VARCHAR2 (50 BYTE)
P	CODIGO_NACION	NUMBER (3)
P	CODIGO_EMPRESA	NUMBER (2)
*	NOMBRE_PRODUCTO	VARCHAR2 (200 BYTE)
*	NOMBRE_AGENCIA	VARCHAR2 (50 BYTE)
*	NOMBRE_CANTON	VARCHAR2 (50 BYTE)
*	NOMBRE_PROVINCIA	VARCHAR2 (100 BYTE)
*	NOMBRE_REGION	VARCHAR2 (300 BYTE)
*	NOMBRE_LINEA	VARCHAR2 (50 BYTE)
*	NOMBRE_MARCA	VARCHAR2 (50 BYTE)
*	COSTO_VENTAS	NUMBER (30,2)
*	PRECIO_VENTAS	NUMBER (30,2)
*	COSTO_DEVOLUCIONES	NUMBER (30,2)
*	PRECIO_DEVOLUCIONES	NUMBER (30,2)
*	COSTO_NETO	NUMBER (30,2)
*	VENTA_NETA	NUMBER (30,2)
*	UTILIDAD	NUMBER (30,2)
*	MES_CERRADO	VARCHAR2 (1 BYTE)
*	FECHA_ULT_EJECUCION	DATE
*	ADICIONADO_POR	VARCHAR2 (30 BYTE)
*	FECHA_ADICION	DATE
*	MODIFICADO_POR	VARCHAR2 (30 BYTE)
*	FECHA_MODIFICACION	DATE
*	COD_ITEM_TIPO_CLIENTE	VARCHAR2 (3 BYTE)
*	COD_MODELO_ITEM_TIPO_CLIENTE	VARCHAR2 (8 BYTE)
PK_DM_INDICE_RENTABILIDAD (PERIODO, COD_PRODUCTO, ANIO_MES_VENTA, FORMA_PAGO, PLAZO, CODIGO_AGENCIA, CODIGO_CANTON, CODIGO_PROVINCIA, CODIGO_REGION)		
DM_INDICE_RENTABILIDAD_D1_IDX (CODIGO_AGENCIA, CODIGO_CANTON, CODIGO_PROVINCIA, CODIGO_REGION)		
DM_INDICE_RENTABILIDAD_D2_IDX (COD_ITEM_CAT_LINEA, COD_MODELO_CAT_LINEA, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D3_IDX (PERIODO, MES_CERRADO, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D4_IDX (COD_PRODUCTO, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D5_IDX (PERIODO, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D7_IDX (ANIO_MES_VENTA, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D8_IDX (PLAZO, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D9_IDX (FORMA_PAGO, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D10_IDX (COD_ITEM_TIPO_CLIENTE, COD_MODELO_ITEM_TIPO_CLIENTE, CODIGO_EMPRESA)		
DM_INDICE_RENTABILIDAD_D11_IDX (COD_MARCA, CODIGO_EMPRESA)		
PK_DM_INDICE_RENTABILIDAD (PERIODO, COD_PRODUCTO, ANIO_MES_VENTA, FORMA_PAGO, PLAZO, CODIGO_AGENCIA, CODIGO_CANTON, CODIGO_PROVINCIA, CODIGO_REGION)		

Fuente: Impocomjager Cía. Ltda.
Elaboración: El autor.



Figura 4.69 Tablas de hechos – Morosidad de la cartera.

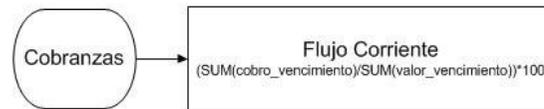


CC_INDICE_DE_MOROSIDAD	
P * PERIODO	NUMBER (6)
P * ANIO_MES_VENTA	NUMBER (6)
P * CODIGO_AGENCIA	NUMBER (4)
P * CODIGO_CANTON	VARCHAR2 (4 BYTE)
P * CODIGO_PROVINCIA	VARCHAR2 (6 BYTE)
P * CODIGO_REGION	NUMBER (3)
P * CODIGO_POLITICA	NUMBER (2)
P * COD_MODELO_CAT_LINEA	VARCHAR2 (8 BYTE)
P * COD_ITEM_CAT_LINEA	VARCHAR2 (3 BYTE)
P * COD_MARCA	NUMBER (3)
P * CODIGO_NACION	NUMBER (3)
P * CODIGO_EMPRESA	NUMBER (2)
* NOMBRE_AGENCIA	VARCHAR2 (50 BYTE)
* NOMBRE_CANTON	VARCHAR2 (50 BYTE)
* NOMBRE_PROVINCIA	VARCHAR2 (100 BYTE)
* NOMBRE_REGION	VARCHAR2 (300 BYTE)
* NOMBRE_LINEA	VARCHAR2 (50 BYTE)
* NOMBRE_MARCA	VARCHAR2 (50 BYTE)
* NOMBRE_POLITICA	VARCHAR2 (100 BYTE)
* VALOR_VENTAS	NUMBER (30,2)
* SALDO_CARTERA	NUMBER (30,2)
* TOTAL_CARTERA_AL_CIERRE	NUMBER (30,2)
* VALOR_COBRADO	NUMBER (30,2)
* CARTERA_VENCIDA	NUMBER (30,2)
* PORCENTAJE_PRODUCTIVIDAD	NUMBER (20,5)
* PORCENTAJE_INDICE_MOROSIDAD	NUMBER (20,5)
* PAGOS_ANTICIPADOS	NUMBER (30,2)
* MES_CERRADO	VARCHAR2 (1 BYTE)
* FECHA_ULT_EJECUCION	DATE
* ADICIONADO_POR	VARCHAR2 (30 BYTE)
* FECHA_ADICION	DATE
MODIFICADO_POR	VARCHAR2 (30 BYTE)
FECHA_MODIFICACION	DATE
CC_INDICE_DE_MOROSIDAD_06_IDX	
CC_INDICE_DE_MOROSIDAD_07_IDX	
CC_INDICE_DE_MOROSIDAD_08_IDX	
CC_INDICE_DE_MOROSIDAD_04_IDX	
CC_INDICE_DE_MOROSIDAD_05_IDX	
PK_CC_IND_DE_MOR	
CC_INDICE_DE_MOROSIDAD_01_IDX	

Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Figura 4.70 Tablas de hechos – Flujo corriente.

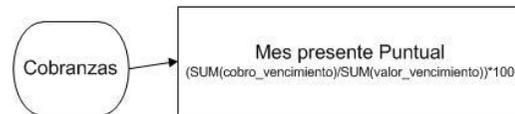


CC_FLUJO_CORRIENTE	
P *	PERIODO NUMBER (6)
P *	FECHA_VENCIMIENTO DATE
P *	CODIGO_AGENCIA NUMBER (4)
P *	CODIGO_CANTON VARCHAR2 (4 BYTE)
P *	CODIGO_PROVINCIA VARCHAR2 (6 BYTE)
P *	CODIGO_REGION NUMBER (3)
P *	CODIGO_POLITICA NUMBER (2)
P *	COD_MODELO_CAT_LINEA VARCHAR2 (8 BYTE)
P *	COD_ITEM_CAT_LINEA VARCHAR2 (3 BYTE)
P *	COD_MARCA NUMBER (3)
P *	COD_CAT_CLIENTE VARCHAR2 (8 BYTE)
P *	ES_EMPLEADO VARCHAR2 (100 BYTE)
P *	CODIGO_NACION NUMBER (3)
P *	CODIGO_EMPRESA NUMBER (2)
*	NOMBRE_AGENCIA VARCHAR2 (50 BYTE)
*	NOMBRE_CANTON VARCHAR2 (50 BYTE)
*	NOMBRE_PROVINCIA VARCHAR2 (100 BYTE)
*	NOMBRE_REGION VARCHAR2 (300 BYTE)
*	NOMBRE_LINEA VARCHAR2 (50 BYTE)
*	NOMBRE_MARCA VARCHAR2 (50 BYTE)
*	NOMBRE_POLITICA VARCHAR2 (100 BYTE)
*	NOMBRE_CATEGORIA_CLIENTE VARCHAR2 (30 BYTE)
*	VALOR_VENCIMIENTO NUMBER (30,2)
*	SALDO_VENCIMIENTO NUMBER (30,2)
*	COBRO_VENCIMIENTO NUMBER (30,2)
*	COBRO_EFECTIVO_VENCIMIENTO NUMBER (30,2)
*	PORCENTAJE_VENCIMIENTO NUMBER (20,5)
*	MES_CERRADO VARCHAR2 (1 BYTE)
*	FECHA_ULT_EJECUCION DATE
*	ADICIONADO_POR VARCHAR2 (30 BYTE)
*	FECHA_ADICION DATE
*	MODIFICADO_POR VARCHAR2 (30 BYTE)
*	FECHA_MODIFICACION DATE
◆	CC_FLU_COR_04_IDX
◆	CC_FLU_COR_05_IDX
◆	CC_FLU_COR_06_IDX
◆	CC_FLU_COR_07_IDX
◆	CC_FLU_COR_08_IDX
◆	CC_FLU_COR_09_IDX
◆	CC_FLU_COR_10_IDX
◆	PK_CC_FLU_COR
◆	CC_FLU_COR_01_IDX

Fuente: Impocomjaher Cía. Ltda.
 Elaboración: El autor.



Figura 4.71 Tablas de hechos – Mes presente puntual.

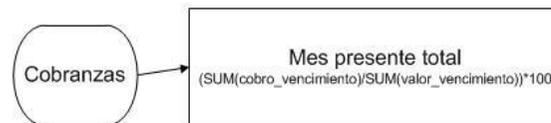


CC_PRODUC_MES_PRESENTE_PUNTUAL	
P *	PERIODO NUMBER (6)
P *	FECHA_VENCIMIENTO DATE
P *	CODIGO_AGENCIA NUMBER (4)
P *	CODIGO_CANTON VARCHAR2 (4 BYTE)
P *	CODIGO_PROVINCIA VARCHAR2 (6 BYTE)
P *	CODIGO_REGION NUMBER (3)
P *	CODIGO_POLITICA NUMBER (2)
P *	COD_MODELO_CAT_LINEA VARCHAR2 (8 BYTE)
P *	COD_ITEM_CAT_LINEA VARCHAR2 (3 BYTE)
P *	COD_MARCA NUMBER (3)
P *	COD_CAT_CLIENTE VARCHAR2 (8 BYTE)
P *	ES_EMPLEADO VARCHAR2 (100 BYTE)
P *	CODIGO_NACION NUMBER (3)
P *	CODIGO_EMPRESA NUMBER (2)
*	NOMBRE_AGENCIA VARCHAR2 (50 BYTE)
*	NOMBRE_CANTON VARCHAR2 (50 BYTE)
*	NOMBRE_PROVINCIA VARCHAR2 (100 BYTE)
*	NOMBRE_REGION VARCHAR2 (300 BYTE)
*	NOMBRE_LINEA VARCHAR2 (50 BYTE)
*	NOMBRE_MARCA VARCHAR2 (50 BYTE)
*	NOMBRE_CATEGORIA_CLIENTE VARCHAR2 (30 BYTE)
*	NOMBRE_POLITICA VARCHAR2 (100 BYTE)
*	VALOR_VENCIMIENTO NUMBER (30,2)
*	SALDO_VENCIMIENTO NUMBER (30,2)
*	COBRO_VENCIMIENTO NUMBER (30,2)
*	COBRO_EFECTIVO_VENCIMIENTO NUMBER (30,2)
*	PORCENTAJE_VENCIMIENTO NUMBER (20,5)
*	MES_CERRADO VARCHAR2 (1 BYTE)
*	FECHA_ULT_EJECUCION DATE
*	ADICIONADO_POR VARCHAR2 (30 BYTE)
*	FECHA_ADICION DATE
*	MODIFICADO_POR VARCHAR2 (30 BYTE)
*	FECHA_MODIFICACION DATE
◆	CC_PRO_MES_PRE_PUN_04_IDX
◆	CC_PRO_MES_PRE_PUN_05_IDX
◆	CC_PRO_MES_PRE_PUN_06_IDX
◆	CC_PRO_MES_PRE_PUN_07_IDX
◆	CC_PRO_MES_PRE_PUN_08_IDX
◆	CC_PRO_MES_PRE_PUN_09_IDX
◆	CC_PRO_MES_PRE_PUN_10_IDX
◆	PK_CC_PRO_MES_PRE_PUN
◆	CC_PRO_MES_PRE_PUN_01_IDX

Fuente: Impocomjaher Cía. Ltda.
 Elaboración: El autor.



Figura 4.72 Tablas de hechos – Mes presente total.

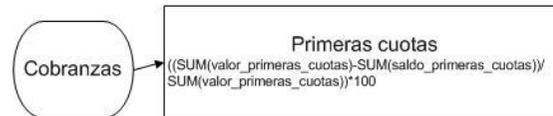


CC_PRODUC_MES_PRESENTE_TOTAL	
P *	PERIODO NUMBER (6)
P *	FECHA_VENCIMIENTO DATE
P *	CODIGO_AGENCIA NUMBER (4)
P *	CODIGO_CANTON VARCHAR2 (4 BYTE)
P *	CODIGO_PROVINCIA VARCHAR2 (6 BYTE)
P *	CODIGO_REGION NUMBER (3)
P *	CODIGO_POLITICA NUMBER (2)
P *	COD_MODELO_CAT_LINEA VARCHAR2 (8 BYTE)
P *	COD_ITEM_CAT_LINEA VARCHAR2 (3 BYTE)
P *	COD_MARCA NUMBER (3)
P *	COD_CAT_CLIENTE VARCHAR2 (8 BYTE)
P *	ES_EMPLEADO VARCHAR2 (100 BYTE)
P *	CODIGO_NACION NUMBER (3)
P *	CODIGO_EMPRESA NUMBER (2)
*	NOMBRE_AGENCIA VARCHAR2 (50 BYTE)
*	NOMBRE_CANTON VARCHAR2 (50 BYTE)
*	NOMBRE_PROVINCIA VARCHAR2 (100 BYTE)
*	NOMBRE_REGION VARCHAR2 (300 BYTE)
*	NOMBRE_LINEA VARCHAR2 (50 BYTE)
*	NOMBRE_MARCA VARCHAR2 (50 BYTE)
*	NOMBRE_CATEGORIA_CLIENTE VARCHAR2 (30 BYTE)
*	NOMBRE_POLITICA VARCHAR2 (100 BYTE)
*	VALOR_VENCIMIENTO NUMBER (30,2)
*	SALDO_VENCIMIENTO NUMBER (30,2)
*	COBRO_VENCIMIENTO NUMBER (30,2)
*	COBRO_EFECTIVO_VENCIMIENTO NUMBER (30,2)
*	PORCENTAJE_VENCIMIENTO NUMBER (20,5)
*	MES_CERRADO VARCHAR2 (1 BYTE)
*	FECHA_ULT_EJECUCION DATE
*	ADICIONADO_POR VARCHAR2 (30 BYTE)
*	FECHA_ADICION DATE
*	MODIFICADO_POR VARCHAR2 (30 BYTE)
*	FECHA_MODIFICACION DATE
PK_CC_PRO_MES_PRE_TOT	
CC_PRO_MES_PRE_TOT_01_IDX	

Fuente: Impocomjaher Cía. Ltda.
 Elaboración: El autor.



Figura 4.73 Tablas de hechos – Primeras cuotas.



CC_PRIMERAS_CUOTAS	
P * PERIODO	NUMBER (6)
P * FECHA_VENCIMIENTO_PRI_CUO	DATE
P * CODIGO_AGENCIA	NUMBER (4)
P * CODIGO_CANTON	VARCHAR2 (4 BYTE)
P * CODIGO_PROVINCIA	VARCHAR2 (6 BYTE)
P * CODIGO_REGION	NUMBER (3)
P * CODIGO_POLITICA	NUMBER (2)
P * COD_MODELO_CAT_LINEA	VARCHAR2 (8 BYTE)
P * COD_ITEM_CAT_LINEA	VARCHAR2 (3 BYTE)
P * COD_MARCA	NUMBER (3)
P * COD_CAT_CLIENTE	VARCHAR2 (8 BYTE)
P * ES_EMPLEADO	VARCHAR2 (100 BYTE)
P * CODIGO_NACION	NUMBER (3)
P * CODIGO_EMPRESA	NUMBER (2)
* NOMBRE_AGENCIA	VARCHAR2 (50 BYTE)
* NOMBRE_CANTON	VARCHAR2 (50 BYTE)
* NOMBRE_PROVINCIA	VARCHAR2 (100 BYTE)
* NOMBRE_REGION	VARCHAR2 (300 BYTE)
* NOMBRE_LINEA	VARCHAR2 (50 BYTE)
* NOMBRE_MARCA	VARCHAR2 (50 BYTE)
* NOMBRE_CATEGORIA_CLIENTE	VARCHAR2 (30 BYTE)
* NOMBRE_POLITICA	VARCHAR2 (100 BYTE)
* VALOR_PRIMERAS_CUOTAS	NUMBER (30,2)
* SALDO_PRIMERAS_CUOTAS	NUMBER (30,2)
* COBRO_PRIMERAS_CUOTAS	NUMBER (30,2)
* COBRO_EFECTIVO_PRI_CUO	NUMBER (30,2)
* PORCENTAJE_PRIMERAS_CUOTAS	NUMBER (20,6)
* MES_CERRADO	VARCHAR2 (1 BYTE)
* FECHA_ULT_EJECUCION	DATE
* ADICIONADO_POR	VARCHAR2 (30 BYTE)
* FECHA_ADICION	DATE
* MODIFICADO_POR	VARCHAR2 (30 BYTE)
* FECHA_MODIFICACION	DATE
◆ CC_PRIMERAS_CUOTAS_06_IDX	
◆ CC_PRIMERAS_CUOTAS_07_IDX	
◆ CC_PRIMERAS_CUOTAS_08_IDX	
◆ CC_PRIMERAS_CUOTAS_09_IDX	
◆ CC_PRIMERAS_CUOTAS_10_IDX	
◆ CC_PRIMERAS_CUOTAS_04_IDX	
◆ CC_PRIMERAS_CUOTAS_03_IDX	
◆ CC_PRIMERAS_CUOTAS_05_IDX	
PK_CC_PRI_CUO	

Fuente: Impocomjaher Cía. Ltda.
Elaboración: El autor.



Anexo A8.

Script carga inicial tabla dimensionamiento dm_periodos_balances.

```
DECLARE
  --
  --
  ****
  **
  --
  -- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
  DE DIMENSIONES
  --                      DE UN PERIODO DE BALANCE CERRADO
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  --
  -- Variables Locales
  --
  Lv_CodigoError        VARCHAR2(3000);
  Lv_mensaje            VARCHAR2(3000);
  --
  -- Obtiene los periodos cerrados de los balances
  --
  CURSOR C_periodos_balances IS
    SELECT periodo_cierre, codigo_empresa
    FROM   cc_balances_mensuales balmen
    WHERE  NOT EXISTS (SELECT 'X'
                      FROM dm_periodos_balances dmperbal
                      WHERE dmperbal.periodo      = balmen.periodo_cierre
                      AND   dmperbal.codigo_empresa = balmen.codigo_empresa
                      )
    GROUP BY codigo_empresa, periodo_cierre
    ORDER BY codigo_empresa, periodo_cierre;
BEGIN
  Lv_mensaje := NULL;
  --
```



```
-- Obtiene los periodos cerrados de los balances
--
FOR C IN C_periodos_balances LOOP
  Lv_mensaje := 'Tabla DM_PERIODOS_BALANCES. Periodo='||C.periodo_cierre||' Empr='||C.codigo_empresa;
  INSERT INTO dm_periodos_balances (PERIODO
                                   CODIGO_EMPRESA
                                   )
    VALUES (C.periodo_cierre,
            C.codigo_empresa
            );
END LOOP;-- FOR C IN C_periodos_balances LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
                    null,--Pv_archivo          IN      VARCHAR2,
                    2,--Pn_codigo_empresa      IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
  VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A9.

Script carga inicial tabla dimensionamiento dm_anio_mes_ventas.

```
DECLARE
  --
  --
  ****
  **
  --
  -- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
  --                      ANIO Y MES DE VENTA
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  --
  -- Variables Locales
  --
  Lv_CodigoError        VARCHAR2(3000);
  Lv_mensaje            VARCHAR2(3000);
  --
  -- Obtiene las fechas de las ventas
  --
  CURSOR C_comprobante IS
  SELECT to_number(to_char(com.fecha,'yyyymm')) anio_mes_venta, com.empresa codigo_empresa
  FROM   comprobante com
  WHERE  com.tipo_comprobante in ('A0','NV','D0','DB')
  AND    NOT EXISTS (SELECT 'X'
                     FROM   dm_anio_mes_ventas dmanimesven
                     WHERE  dmanimesven.anio_mes_venta = to_number(to_char(com.fecha,'yyyymm'))
                     AND    dmanimesven.codigo_empresa = com.empresa
                     )
  GROUP BY to_number(to_char(com.fecha,'yyyymm')), com.empresa
  ORDER BY to_number(to_char(com.fecha,'yyyymm')), com.empresa;
BEGIN
  Lv_mensaje := NULL;
  --
```



```
-- Obtiene las fechas de las ventas
--
FOR C IN C_comprobante LOOP
  Lv_mensaje := 'Tabla DM_ANIO_MES_VENTAS. AnioMesVentao='||C.anio_mes_venta||' Empr='||C.codigo_empresa;
  INSERT INTO dm_anio_mes_ventas (ANIO_MES_VENTA
                                CODIGO_EMPRESA
                                )
    VALUES (C.anio_mes_venta,
            C.codigo_empresa
            );
END LOOP;-- FOR C IN C_comprobante LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio          IN      VARCHAR2,
                    null,--Pv_archivo              IN      VARCHAR2,
                    2,--Pn_codigo_empresa          IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto          IN
  VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje          IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError          IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A10.

Script carga inicial tabla dimensionamiento dm_plazos_ventas.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      PLAZOS DE VENTA
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene el plazo de las ventas
--
CURSOR C_comprobante_x_plazo IS
SELECT com.numero_pagos, com.empresa codigo_empresa
FROM comprobante com
WHERE com.tipo_comprobante in ('A0','NV','D0','DB')
AND NOT EXISTS (SELECT 'X'
FROM dm_plazos_ventas dmplaven
WHERE dmplaven.plazo = com.numero_pagos
AND dmplaven.codigo_empresa = com.empresa
)
GROUP BY com.numero_pagos, empresa
ORDER BY empresa, com.numero_pagos;
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene el plazo de las ventas
```



```
--
FOR C IN C_comprobante_x_plazo LOOP
  Lv_mensaje := 'Tabla DM_PLAZOS_VENTAS. Plazo='||C.numero_pagos||' Empr='||C.codigo_empresa;
  INSERT INTO dm_plazos_ventas (PLAZO
                                CODIGO_EMPRESA
                                )
  VALUES (C.numero_pagos,
           C.codigo_empresa
           );
END LOOP;-- FOR C IN C_comprobante_x_plazo LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio          IN      VARCHAR2,
                    null,--Pv_archivo              IN      VARCHAR2,
                    2,--Pn_codigo_empresa          IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto          IN
VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje          IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError          IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A11.

Script carga inicial tabla dimensionamiento dm_lineas.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      LINEAS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las Lineas de todas las empresas
--
CURSOR C_tg_modelo_item IS
SELECT *
FROM   tg_modelo_item modite
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_lineas dmlin
                   WHERE  dmlin.empresa      = modite.empresa
                   AND    dmlin.cod_modelo_linea = modite.cod_modelo
                   AND    dmlin.cod_item_linea  = modite.cod_item
                   );
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene las Lineas de todas las empresas
--
```



```
FOR C IN C_tg_modelo_item LOOP
  Lv_mensaje := 'Tabla DM_LINEA. Empr='||C.empresa||' Modelo_Linea='||C.cod_modelo||'
Item_Linea='||C.cod_item;
  INSERT INTO dm_lineas (EMPRESA
                        COD_MODELO_LINEA
                        COD_ITEM_LINEA
                        NOMBRE
                        )
  VALUES (C.empresa,
          C.cod_modelo,
          C.cod_item,
          C.nombre
          );
END LOOP;-- FOR C IN C_tg_modelo_item LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
                    null,--Pv_archivo         IN      VARCHAR2,
                    2,--Pn_codigo_empresa     IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A12.

Script carga inicial tabla dimensionamiento dm_marcas.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      MARCAS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las Marcas de todas las empresas
--
CURSOR C_marca IS
  SELECT *
  FROM   marca mar
  WHERE  NOT EXISTS (SELECT 'X'
                    FROM   dm_marcas dmmar
                    WHERE  dmmar.cod_marca = mar.cod_marca
                    AND    dmmar.empresa  = mar.empresa
                    );
BEGIN
--
-- Obtiene las Marcas de todas las empresas
--
FOR C IN C_marca LOOP
  Lv_mensaje := 'Tabla DM_MARCAS. Marca='||C.cod_marca||' Empr='||C.empresa;
  INSERT INTO dm_marcas (COD_MARCA
```



```

                EMPRESA
                NOMBRE
            )
VALUES (C.cod_marca,
        C.empresa,
        C.nombre
        );
END LOOP; -- FOR C IN C_marca LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
    --
    -- Envia correo de Error a DBA
    --
    Lv_CodigoError := null;
    P_ENVIAR_MAILS_DBA (null, --Pv_directorio          IN      VARCHAR2,
                       null, --Pv_archivo             IN      VARCHAR2,
                       2, --Pn_codigo_empresa         IN      NUMBER,
                       'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto          IN
    VARCHAR2,
                       'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                       Lv_mensaje||chr(10)||chr(10)||
                       'Error: '||SQLERRM, --Pv_mensaje          IN      VARCHAR2,
                       Lv_CodigoError--Pv_CodigoError          IN OUT VARCHAR2
                       );
    raise_application_error (-20003,SQLERRM);
END;
```



Anexo A13.

Script carga inicial tabla dimensionamiento dm_productos.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      PRODUCTOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene los Productos de todas las empresas
--
CURSOR C_productos IS
SELECT *
FROM   producto pro
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_productos dmpro
                   WHERE  dmpro.cod_producto = pro.cod_producto
                   AND    dmpro.empresa     = pro.empresa
                   );
BEGIN
--
-- Obtiene los Productos de todas las empresas
--
FOR C IN C_productos LOOP
INSERT INTO dm_productos (COD_PRODUCTO
```



```

        EMPRESA
        NOMBRE
        COD_MARCA
    )
VALUES (C.cod_producto,
        C.empresa,
        C.nombre,
        C.cod_marca
    );
END LOOP; -- FOR C IN C_productos LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
    --
    -- Envia correo de Error a DBA
    --
    Lv_CodigoError := null;
    P_ENVIAR_MAILS_DBA (null, --Pv_directorio          IN      VARCHAR2,
                       null, --Pv_archivo            IN      VARCHAR2,
                       2, --Pn_codigo_empresa        IN      NUMBER,
                       'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto          IN
    VARCHAR2,
                       'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                       Lv_mensaje||chr(10)||chr(10)||
                       'Error: '||SQLERRM, --Pv_mensaje          IN      VARCHAR2,
                       Lv_CodigoError --Pv_CodigoError          IN OUT VARCHAR2
    );
    raise_application_error (-20003,SQLERRM);
END;
```



Anexo A14.

Script carga inicial tabla dimensionamiento dm_forma_pago.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      FORMAS DE VENTA
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las Formas de Ventas de todas las empresas
--
CURSOR C_forma_pago IS
SELECT *
FROM   forma_pago forpag
WHERE  NOT EXISTS (SELECT 'X'
                  FROM   dm_forma_pago dmforpag
                  WHERE  dmforpag.cod_forma_pago = forpag.cod_forma_pago
                  AND    dmforpag.empresa      = forpag.empresa
                  );
BEGIN
--
-- Obtiene las Formas de Ventas de todas las empresas
--
FOR C IN C_forma_pago LOOP
INSERT INTO dm_forma_pago (COD_FORMA_PAGO,
                          EMPRESA,
```



```

                                NOMBRE
                                )
VALUES (C.cod_forma_pago,
        C.empresa,
        C.nombre
        );
END LOOP; -- FOR C IN C_forma_pago LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
--
-- Envia correo de Error a DBA
--
Lv_CodigoError := null;
P_ENVIAR_MAILS_DBA (null, --Pv_directorio      IN      VARCHAR2,
                    null, --Pv_archivo        IN      VARCHAR2,
                    2, --Pn_codigo_empresa    IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto      IN
VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
Lv_mensaje||chr(10)||chr(10)||
'Error: '||SQLERRM, --Pv_mensaje              IN      VARCHAR2,
Lv_CodigoError--Pv_CodigoError                IN OUT VARCHAR2
                    );
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A15.

Script carga inicial tabla dimensionamiento dm_tipo_cliente.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      TIPO DE CLIENTE
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene los Tipos de Clientes de todas las empresas
--
CURSOR C_tg_modelo_item_tip_cli IS
SELECT *
FROM   tg_modelo_item modite
WHERE  modite.cod_modelo = 'CLI1'
AND    NOT EXISTS (SELECT 'X'
                   FROM   dm_tipo_cliente dmtipcli
                   WHERE  dmtipcli.cod_item_tipo_cliente = modite.cod_item
                   AND    dmtipcli.cod_modelo_item_tipo_cliente = modite.cod_modelo
                   AND    dmtipcli.empresa = modite.empresa
                   );
BEGIN
--
-- Obtiene los Tipos de Clientes de todas las empresas
--
```



```
FOR C IN C_tg_modelo_item_tip_cli LOOP
  INSERT INTO dm_tipo_cliente (COD_ITEM_TIPO_CLIENTE
                              COD_MODELO_ITEM_TIPO_CLIENTE
                              EMPRESA
                              NOMBRE
                              )
  VALUES (C.cod_item,
          C.cod_modelo,
          C.empresa,
          C.nombre
          );
END LOOP;-- FOR C IN C_tg_modelo_item_tip_cli LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
                    null,--Pv_archivo         IN      VARCHAR2,
                    2,--Pn_codigo_empresa     IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
  VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A16.

Script carga inicial tabla dimensionamiento dm_politicas.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      POLITICAS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las Politicas de todas las empresas
--
CURSOR C_politica_credito IS
SELECT *
FROM   st_politica_credito polcre
WHERE  NOT EXISTS (SELECT 'X'
                  FROM   dm_politicas dmpol
                  WHERE  dmpol.cod_politica = polcre.cod_politica
                  AND    dmpol.empresa     = polcre.empresa
                  );
BEGIN
--
-- Obtiene las Politicas de todas las empresas
--
FOR C IN C_politica_credito LOOP
INSERT INTO dm_politicas (COD_POLITICA
```



```

                EMPRESA
                NOMBRE
            )
VALUES (C.cod_politica,
        C.empresa,
        C.nombre
    );
END LOOP; -- FOR C IN C_politica_credito LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
    --
    -- Envia correo de Error a DBA
    --
    Lv_CodigoError := null;
    P_ENVIAR_MAILS_DBA (null, --Pv_directorio          IN    VARCHAR2,
                       null, --Pv_archivo             IN    VARCHAR2,
                       2, --Pn_codigo_empresa         IN    NUMBER,
                       'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto          IN
    VARCHAR2,
                       'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                       Lv_mensaje||chr(10)||chr(10)||
                       'Error: '||SQLERRM, --Pv_mensaje          IN    VARCHAR2,
                       Lv_CodigoError--Pv_CodigoError          IN OUT VARCHAR2
    );
    raise_application_error (-20003,SQLERRM);
END;
```



Anexo A17.

Script carga inicial tabla dimensionamiento dm_calificacion_del_cliente.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      CALIFICACION DEL CLIENTE
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene el codigo de las calificaciones de los clientes de todas las empresas
--
CURSOR C_st_cat_cliente IS
SELECT *
FROM   st_cat_cliente catcli
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_calificacion_del_cliente dmcalcli
                   WHERE  dmcalcli.cod_cat_cliente = catcli.cod_cat_cliente
                   AND    dmcalcli.empresa        = catcli.empresa
                   );
BEGIN
--
-- Obtiene el codigo de las calificaciones de los clientes de todas las empresas
--
FOR C IN C_st_cat_cliente LOOP
INSERT INTO dm_calificacion_del_cliente (COD_CAT_CLIENTE,
```



```

                                EMPRESA,
                                NOMBRE
                                )

VALUES (C.cod_cat_cliente,
        C.empresa,
        C.nombre
        );

END LOOP;-- FOR C IN C_st_cat_cliente LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
--
-- Envia correo de Error a DBA
--
Lv_CodigoError := null;
P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN   VARCHAR2,
                    null,--Pv_archivo        IN   VARCHAR2,
                    2,--Pn_codigo_empresa    IN   NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje      IN   VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
                    );
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A18.

Script carga inicial tabla dimensionamiento dm_es_empleado.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      ES EMPLEADO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las empresas donde trabajan los empleados de todas las empresas
--
CURSOR C_st_empleados IS
SELECT stemp.nombre_empresa
FROM   st_empleados stemp
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_es_empleado dmesemp
                   WHERE  dmesemp.es_empleado = stemp.nombre_empresa
                  )
GROUP BY stemp.nombre_empresa;
BEGIN
--
-- Obtiene las empresas donde trabajan los empleados de todas las empresas
--
FOR C IN C_st_empleados LOOP
INSERT INTO dm_es_empleado (ES_EMPLEADO
```



```
VALUES (C.nombre_empresa
)
);
END LOOP;-- FOR C IN C_st_empleados LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
--
-- Envia correo de Error a DBA
--
Lv_CodigoError := null;
P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
null,--Pv_archivo      IN      VARCHAR2,
2,--Pn_codigo_empresa  IN      NUMBER,
'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
VARCHAR2,
'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
Lv_mensaje||chr(10)||chr(10)||
'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
);
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A19.

Script carga inicial tabla dimensionamiento dm_fechas_vencimientos.

```
DECLARE
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      FECHAS DE VENCIMIENTOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Lv_mensaje            VARCHAR2(3000);
--
-- Obtiene las Fechas de Vencimientos de todas las cuotas de todas las empresas
--
CURSOR C_fechas_vencimiento IS
SELECT trunc(vencimiento) vencimiento
FROM   forma_pago_comprobante forpagcom
WHERE  forpagcom.vencimiento > to_date('2006-01-01','yyyy-mm-dd')
AND    NOT EXISTS (SELECT 'X'
                   FROM   dm_fechas_vencimientos dmfecven
                   WHERE  dmfecven.fecha_vencimiento = trunc(forpagcom.vencimiento)
                   )
GROUP BY trunc(vencimiento)
ORDER BY trunc(vencimiento);
BEGIN
--
-- Obtiene las Fechas de Vencimientos de todas las cuotas de todas las empresas
--
```



```
FOR C IN C_fechas_vencimiento LOOP
  INSERT INTO dm_fechas_vencimientos (FECHA_VENCIMIENTO
                                     )
    VALUES (C.vencimiento
            );
END LOOP;-- FOR C IN C_fechas_vencimiento LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
                    null,--Pv_archivo         IN      VARCHAR2,
                    2,--Pn_codigo_empresa     IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
                    Lv_mensaje||chr(10)||chr(10)||
                    'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError--Pv_CodigoError     IN OUT VARCHAR2
                    );
  raise_application_error (-20003,SQLERRM);
END;
```



Anexo A20.

Script carga inicial tabla dimensionamiento dm_fechas_ventas.

```
DECLARE
--
-- Obtiene las fechas de las ventas en Anio_Mes Venta y Dia de Venta
--
CURSOR C_comprobante_aaaamm_dia IS
SELECT to_number(to_char(com.fecha,'yyyymm')) anio_mes_venta, to_number(to_char(com.fecha,'dd')) dia_venta,
com.empresa codigo_empresa
FROM   comprobante com
WHERE  com.tipo_comprobante in ('A0','NV','D0','DB')
AND    NOT EXISTS (SELECT 'X'
                   FROM   dm_fechas_ventas dmfecven
                   WHERE  dmfecven.anio_mes_venta = to_number(to_char(com.fecha,'yyyymm'))
                   AND    dmfecven.dia_venta      = to_number(to_char(com.fecha,'dd'))
                   AND    dmfecven.codigo_empresa = com.empresa
                  )
GROUP BY to_number(to_char(com.fecha,'yyyymm')), to_number(to_char(com.fecha,'dd')), com.empresa
ORDER BY to_number(to_char(com.fecha,'yyyymm')), to_number(to_char(com.fecha,'dd')), com.empresa;
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene las fechas de las ventas en Anio_Mes Venta y Dia de Venta
--
FOR C IN C_comprobante_aaaamm_dia LOOP
Lv_mensaje := 'Tabla DM_FECHAS_VENTAS. AnioMesVenta='||C.anio_mes_venta||' DiaVta='||C.dia_venta||'
Empr='||C.codigo_empresa;
INSERT INTO dm_fechas_ventas (ANIO_MES_VENTA
                             DIA_VENTA,
                             CODIGO_EMPRESA
                            )
VALUES (C.anio_mes_venta,
        C.dia_venta,
        C.codigo_empresa
```



```
);
END LOOP;-- FOR C IN C_comprobante_aaaamm_dia LOOP
--
--
--
commit;
EXCEPTION
WHEN others THEN
--
-- Envia correo de Error a DBA
--
Lv_CodigoError := null;
P_ENVIAR_MAILS_DBA (null,--Pv_directorio      IN      VARCHAR2,
null,--Pv_archivo      IN      VARCHAR2,
2,--Pn_codigo_empresa  IN      NUMBER,
'Error. No pudo cargar las tablas de Dimensionamiento.',--Pv_asunto      IN
VARCHAR2,
'Error. No pudo cargar las tablas de Dimensionamiento. '||chr(10)||chr(10)||
Lv_mensaje||chr(10)||chr(10)||
'Error: '||SQLERRM,--Pv_mensaje      IN      VARCHAR2,
Lv_CodigoError--Pv_CodigoError      IN OUT VARCHAR2
);
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A21.

Script carga inicial tabla dimensionamiento ad_empresas.

```
DECLARE
--
-- Obtiene las empresas
--
CURSOR C_empresa IS
SELECT emp.empresa, emp.nombre
FROM   empresa emp
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   ad_empresas ademp
                   WHERE  ademp.codigo_empresa = emp.empresa
                   );
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene las empresas
--
FOR C IN C_empresa LOOP
Lv_mensaje := 'Tabla AD_EMPRESAS. Empr='||C.codigo_empresa;
INSERT INTO ad_empresas (CODIGO_EMPRESA
                        , DESCRIPCION
                        , ANULADO
                        )
VALUES (C.codigo_empresa,
       C.nombre,
       'N'
       );
END LOOP;-- FOR C IN C_empresa LOOP
--
--
--
commit;
EXCEPTION
```



```
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null, --Pv_directorio      IN      VARCHAR2,
                    null, --Pv_archivo         IN      VARCHAR2,
                    2, --Pn_codigo_empresa    IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto      IN
  VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. ' || chr(10) || chr(10) ||
                    Lv_mensaje || chr(10) || chr(10) ||
                    'Error: ' || SQLERRM, --Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError --Pv_CodigoError      IN OUT VARCHAR2
                    );
  raise_application_error (-20003, SQLERRM);
END;
```



Anexo A22.

Script carga inicial tabla dimensionamiento tg_agencia.

```
DECLARE
--
-- Obtiene las Agencias
--
CURSOR C_agencias IS
SELECT age.codigo_agencia, age.codigo_empresa, age.descripcion
FROM   ad_agencias age
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   tg_agencia tgage
                   WHERE  tgage.empresa   = age.codigo_empresa
                   AND    tgage.cod_agencia = age.codigo_agencia
                   );
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene las Agencias
--
FOR C IN C_agencias LOOP
Lv_mensaje := 'Tabla TG_AGENCIA. Agen='||C.codigo_agencia||' Empr='||C.codigo_empresa;
INSERT INTO tg_agencia (COD_AGENCIA
                       EMPRESA,
                       NOMBRE
                       )
VALUES (C.codigo_agencia,
       C.codigo_empresa,
       C.descripcion
       );
END LOOP;-- FOR C IN C_agencias LOOP
--
--
--
commit;
```



```
EXCEPTION
  WHEN others THEN
    --
    -- Envia correo de Error a DBA
    --
    Lv_CodigoError := null;
    P_ENVIAR_MAILS_DBA (null, --Pv_directorio      IN      VARCHAR2,
                       null, --Pv_archivo        IN      VARCHAR2,
                       2, --Pn_codigo_empresa    IN      NUMBER,
                       'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto      IN
    VARCHAR2,
                       'Error. No pudo cargar las tablas de Dimensionamiento. ' || chr(10) || chr(10) ||
                       Lv_mensaje || chr(10) || chr(10) ||
                       'Error: ' || SQLERRM, --Pv_mensaje      IN      VARCHAR2,
                       Lv_CodigoError --Pv_CodigoError      IN OUT VARCHAR2
                       );
    raise_application_error (-20003, SQLERRM);
END;
```



Anexo A23.

Script carga inicial tabla dm_indice_cumplimiento_ventas.

```
DECLARE
--
-- Variables Locales
--
Ld_fecha_inicial          DATE;
Ld_fecha_final           DATE;
Ln_periodo_balance       cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_codigo_nacion         ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton         ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region         ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia      ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Ln_contador              NUMBER;
Ln_anio_inicial          NUMBER(4);
Ln_anio_final            NUMBER(4);
Ln_anio                  NUMBER(4);
Ln_mes                   NUMBER(2);
Ln_dia                   NUMBER(2);
Ln_dia_final             NUMBER(2);
Ld_fecha_venta           DATE;
Ln_numero_semana_mes    NUMBER(2);
Lv_CodigoError           VARCHAR2(3000);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM   cc_tipos_carteras tipcar
WHERE  tipcar.estado = 'N'
AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
```



```
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel que
utilizaba Franklin Bernal
--
CURSOR C_ventas_detalladas IS
SELECT /*+ first_rows */
    e.cod_agencia,
    e.nombre Agencia,
    to_char(a.fecha,'yyyymm') Fecha,
    to_char(a.fecha,'yyyy-mm-dd') Fecha_dia,
    to_char(a.fecha,'dd') dia_venta,
    d.nombre Linea,
    g.nombre Marca,
    f.nombre Vendedor,
    a.numero_pagos,
    a.forma_pago,
    k.nombre Tipo_cliente,
    j.cod_cat_cliente Categoria_cliente,
    decode(a.tipo_comprobante,'D0',(b.total_linea-b.iva) * -1,(b.total_linea-b.iva))Valor
FROM
    (
        SELECT ven.cod_comprobante, ven.empresa, ven.tipo_comprobante,
        ven.fecha,ven.numero_pagos,ven.forma_pago,ven.cod_agencia,
        ven.cod_agente,ven.cod_persona,secuen_certificado,observaciones
        FROM stock.comprobante VEN
        WHERE ven.tipo_comprobante in ('A0','NV')
        AND ven.anulado='N'
        AND ven.fecha between to_date(to_char(Ld_fecha_inicial,'yyyy-mm-dd')||' 00:00:00','yyyy-mm-dd hh24:mi:ss')
        AND to_date(to_char(Ld_fecha_final,'yyyy-mm-dd')||' 23:59:59','yyyy-mm-dd hh24:mi:ss')
        AND ven.empresa=Ln_codigo_empresa
        UNION ALL
        SELECT dev.cod_comprobante,dev.empresa, dev.tipo_comprobante,
        dev.fecha,dev.numero_pagos,dev.forma_pago,ven_dev.cod_agencia,
        dev.cod_agente,dev.cod_persona,dev.secuen_certificado,dev.observaciones
        from
        STOCK.COMPROBANTE DEV,
        STOCK.ST_COMPROBANTE_COMPROBANTE X,
```



```
STOCK.COMPROBANTE VEN_DEV,  
STA_PARAMETROS_VISTA_ELEC par  
WHERE dev.cod_comprobante=x.cod_comprobante_s  
AND dev.tipo_comprobante=x.tipo_comprobante_s  
AND dev.empresa=x.empresa_s  
AND ven_dev.cod_comprobante=x.cod_comprobante  
AND ven_dev.tipo_comprobante=x.tipo_comprobante  
AND ven_dev.empresa=x.empresa  
AND dev.tipo_comprobante in ('D0')  
AND dev.anulado='N'  
AND dev.empresa=Ln_codigo_empresa  
AND dev.fecha between to_date(to_char(Ld_fecha_inicial,'yyyy-mm-dd')||' 00:00:00','yyyy-mm-dd hh24:mi:ss')  
AND to_date(to_char(Ld_fecha_final,'yyyy-mm-dd')||' 23:59:59','yyyy-mm-dd hh24:mi:ss')  
 )a  
 ,stock.movimiento b  
 ,stock.producto c  
 ,computo.tg_modelo_item d  
 ,computo.tg_agencia e  
 ,stock.persona f  
 ,stock.marca g  
 ,stock.cliente h  
 ,stock.cliente_hor i  
 ,stock.st_cliente_cupo j  
 ,computo.tg_modelo_item k  
 ,stock.st_comprobante_comprobante l  
 ,stock.comprobante m --devolucion  
 ,stock.forma_pago_comprobante n  
 ,stock.st_tipo_devolucion_anticipo o  
 ,stock.st_credijaher p  
 ,stock.st_credijaher_origen q  
 WHERE b.cod_comprobante=a.cod_comprobante  
 AND b.tipo_comprobante=a.tipo_comprobante  
 AND b.empresa=a.empresa  
 AND c.cod_producto=b.cod_producto  
 AND c.empresa=b.empresa  
 AND d.cod_modelo(+)=c.cod_modelo_cat  
 AND d.cod_item(+)=c.cod_item_cat
```



```
AND d.empresa(+) = c.empresa
AND e.cod_agencia = a.cod_agencia
AND e.empresa = a.empresa
AND f.cod_persona = a.cod_agente
AND f.empresa = a.empresa
AND f.cod_tipo_persona = 'VEN'
AND g.cod_marca = c.cod_marca
AND g.empresa = c.empresa
AND h.cod_cliente = a.cod_persona
AND h.empresa = a.empresa
AND i.cod_clienteh = h.cod_cliente
AND i.empresa = h.empresa
AND j.empresa(+) = h.empresa
AND j.cod_cliente(+) = h.cod_cliente
AND k.empresa = i.empresa
AND k.cod_modelo = 'CL11'
AND k.cod_item = i.cod_tipo_clienteh
--Devoluciones
AND l.empresa(+) = a.empresa
AND l.tipo_comprobante_s(+) = a.tipo_comprobante
AND l.cod_comprobante_s(+) = a.cod_comprobante
AND m.empresa(+) = l.empresa
AND m.tipo_comprobante(+) = l.tipo_comprobante
AND m.cod_comprobante(+) = l.cod_comprobante
AND a.empresa = n.empresa(+)
AND a.tipo_comprobante = n.tipo_comprobante(+)
AND a.cod_comprobante = n.cod_comprobante(+)
AND n.cod_forma_pago(+) = 'ADI'
AND o.empresa(+) = a.empresa
AND o.cod_tipo_devolucion(+) = a.secuen_certificado
AND p.cod_cliente(+) = a.cod_persona
AND q.cod_origen(+) = p.cod_origen;
BEGIN
Ln_contador := 0;
Ld_fecha_inicial := to_date('2009-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss');
Ld_fecha_final := to_date('2011-12-31 23:59:59', 'yyyy-mm-dd hh24:mi:ss');
--
```



```
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  Ln_anio_inicial := to_number(to_char(Ld_fecha_inicial,'yyyy'));
  Ln_anio_final   := to_number(to_char(Ld_fecha_final,'yyyy'));
  FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP
    FOR Ln_mes IN 1..12 LOOP
      Ln_dia_final := to_number(to_char(last_day(to_date(Ln_anio||lpad(Ln_mes,2,'0')||'01','yyyy-mm-
dd')), 'dd'));
      FOR Ln_dia IN 1..Ln_dia_final LOOP
        --
        -- Inserta los dias
        --
        BEGIN
          INSERT INTO ve_fechas_ventas (ANIO_MES_VENTA
                                     DIA_VENTA
                                     CODIGO_EMPRESA
                                     )
          VALUES (to_number(Ln_anio||lpad(Ln_mes,2,'0')), --ANIO_MES_VENTA          NOT NULL
                 Ln_dia, --DIA_VENTA          NOT NULL NUMBER(2)
                 J.codigo_empresa--CODIGO_EMPRESA          NOT NULL NUMBER(2)
                 );
        EXCEPTION
          WHEN others THEN
            NULL;
        END;
      END LOOP; -- FOR Ln_dia IN 1..Ln_dia_final LOOP
    END LOOP; -- FOR Ln_mes IN 1..12 LOOP
  END LOOP; -- FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  --
```



```
--
*****
**
--
-- Funcion      : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--              de la tabla AD_PARAMETROS_GENERALES
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                    );

IF Ln_codigo_nacion IS NULL THEN
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel
que utilizaba Franklin Bernal
--
FOR C IN C_ventas_detalladas LOOP
--
--
*****
**
--
-- Procedimiento : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--              de la tabla AD_AGENCIAS_X_CANTONES
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
```



```
J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
Lv_codigo_canton,--Pv_codigo_canton       IN OUT VARCHAR2,
Ln_codigo_region,--Pn_codigo_region       IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD AGENCIAS X CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region       IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa     IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Actualiza las ventas netas del Cumplimiento de ventas
--
UPDATE dm_indice_cumplimiento_ventas indcumven
SET indcumven.valor_venta_neta = indcumven.valor_venta_neta + C.valor
WHERE indcumven.anio_mes_venta = to_number(C.fecha)
AND indcumven.dia_venta = to_number(C.dia_venta)
AND indcumven.codigo_agencia = C.cod_agencia
AND indcumven.codigo_canton = Lv_codigo_canton
AND indcumven.codigo_provincia = Lv_codigo_provincia
AND indcumven.codigo_region = Ln_codigo_region
```



```
AND indcumven.codigo_nacion = Ln_codigo_nacion
AND indcumven.codigo_empresa = J.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO dm_indice_cumplimiento_ventas (ANIO_MES_VENTA
                                           DIA_VENTA
                                           CODIGO_AGENCIA
                                           CODIGO_EMPRESA
                                           VALOR_VENTA_NETA
                                           MES_CERRADO
                                           )
  VALUES (to_number(C.fecha), --PERIODO NOT NULL NUMBER(6)
          to_number(C.dia_venta), --DIA_VENTA NOT NULL NUMBER(2)
          C.cod_agencia, --CODIGO_AGENCIA NOT NULL NUMBER(4)
          J.codigo_empresa, --CODIGO_EMPRESA NOT NULL NUMBER(2)
          C.valor, --VALOR_VENTA_NETA NOT NULL NUMBER(30,2)
          'C' --MES_CERRADO NOT NULL VARCHAR2(1)
          );
END IF; -- IF Sql%Notfound THEN
--
--
--
Ln_contador := Ln_contador + 1;
IF Ln_contador > 2000 THEN
  commit;
  Ln_contador := 0;
END IF; -- IF Ln_contador > 2000 THEN
END LOOP; -- FOR C IN C_ventas_detalladas LOOP
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
--
--
commit;
END;
```



Anexo A24.

Script carga inicial tabla dm_indice_rentabilidad.

```
--  
--  
*****  
**  
--  
-- Procedimiento      : Carga el Indice de rentabilidad Para el Nuevo Mes  
--                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes  
--                   : Inserta en la Tabla DM INDICE RENTABILIDAD  
-- Autor              : Ing. Giovanni Chasifan Chicaiza.  
-- Fecha creacion     : Viernes 1 de Abril de 2011.  
DECLARE  
--  
-- Variables Locales  
--  
Lv_CodigoError        VARCHAR2(3000);  
Ln_codigo_empresa     ad_empresas.codigo_empresa%TYPE;  
Ln_periodo_balance     cc_periodos_cerrados.periodo_cierre%TYPE;  
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;  
Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;  
Ln_contador            NUMBER(30);  
Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;  
Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;  
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;  
Lv_cod_modelo_cat_linea producto.cod_modelo_cat%TYPE;  
Lv_cod_item_cat_linea  producto.cod_item_cat%TYPE;  
Ln_cod_marca            cc_balances_mensuales.codigo_marca%TYPE;  
Lv_cartera              cc_periodos_cerrados.cartera%TYPE;  
Lv_cod_producto        sta_rentabilidad_temporal.cod_producto%TYPE;  
Lv_tipo_cliente         tg_modelo_item.nombre%TYPE;  
Lv_cod_modelo_cliente  tg_modelo_item.cod_modelo%TYPE;  
Lv_cod_item_cliente    tg_modelo_item.cod_item%TYPE;  
--
```



```
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM cc_tipos_carteras tipcar
WHERE tipcar.estado = 'N'
AND tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
SELECT periodo_cierre, fecha_inicial, fecha_final
FROM cc_periodos_cerrados percer
WHERE codigo_empresa = Ln_codigo_empresa
AND cartera = Lv_cartera
AND periodo_cierre > 200512
AND NOT EXISTS (SELECT 'X'
FROM dm_indice_rentabilidad indren
WHERE indren.periodo = percer.periodo_cierre
AND indren.codigo_empresa = percer.codigo_empresa
)
GROUP BY periodo_cierre, fecha_inicial, fecha_final
ORDER BY 1;
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
CURSOR C_rentabilidad IS
select a.empresa,
a.cod_agencia,
e.nombre agencia,
to_number(to_char(a.fecha,'yyyymm')) fecha,
a.cod_producto,
trim(b.nombre) nombre,
d.nombre linea,
c.nombre marca,
a.pagos num_pagos,
```



```
g.nombre tipo_cliente,  
g.cod_modelo,  
g.cod_item,  
  
round(sum(decode(a.deb_cred,2,decode(a.costo_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costo_reexpresado),0)),2) costo_ventas,  
round(sum(decode(a.deb_cred,2,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1),0)),2) precio_ventas,  
  
round(sum(decode(a.deb_cred,1,decode(a.costo_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costo_reexpresado)*-1,0)),2) costo_devoluciones,  
round(sum(decode(a.deb_cred,1,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1)*-1,0)),2) precio_devoluciones,  
i.forma_pago  
from sta_rentabilidad_temporal a,  
producto b,  
marca c,  
tg_modelo_item d,  
tg_agencia e,  
cliente_hor f,  
sta_parametros_vista_elec h,  
tg_modelo_item g,  
comprobante i  
where a.empresa = h.empresa  
and h.activo='S'  
and a.usuario=user  
and a.cod_persona!='0190331024001'  
and b.empresa=a.empresa  
and b.cod_producto=a.cod_producto  
and c.empresa=b.empresa  
and c.cod_marca=b.cod_marca  
and d.empresa=b.empresa  
and d.cod_modelo='PRO2'  
and d.cod_item=b.cod_item_cat  
and e.empresa=a.empresa
```



```
and e.cod_agencia=a.cod_agencia
and f.empresah=a.empresa
and f.cod_clienteh=a.cod_persona
and g.empresa=f.empresah
and g.cod_modelo='CL11'
and g.cod_item=f.cod_tipo_clienteh
and d.nombre<>'Electropolis S.A.'
and i.empresa=a.empresa
and i.tipo_comprobante=a.tipo_comprobante
and i.cod_comprobante=a.cod_comprobante
group by a.empresa,
a.cod_agencia,
to_number(to_char(a.fecha,'yyyymm')),
a.cod_producto ,
b.nombre ,
c.nombre ,
a.pagos,
d.nombre ,
e.nombre ,
i.forma_pago,
g.nombre,
g.cod_modelo,
g.cod_item
order by fecha asc;
BEGIN
--
-- Inserta Tablas Maestras del DM_INDICE_RENTABILIDAD
--
--
--
*****
**
--
-- Procedimiento : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS
TABLAS DE DIMENSIONES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Miercoles 4 de Julio de 2012.
```



```
PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS_NUEVOS;
--
-- ***** INSERTA TODAS LAS VENTAS EN EL PERIODO CERRADO
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  Ln_codigo_empresa := J.codigo_empresa;
  Lv_cartera := J.cartera;
  --
  --
  *****
  **
  --
  -- Funcion : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
  -- de la tabla AD PARAMETROS GENERALES
  -- Autor : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación : Martes 19 de Octubre de 2010.
  Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa IN
NUMBER
);
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
FOR I IN C_periodos_cerrados LOOP
  Ln_periodo_balance := I.periodo_cierre;
  Ld_fecha_inicial := I.fecha_inicial;
  Ld_fecha_final := I.fecha_final;
  --
  --
```



```
--
Ln_contador := 0;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta_rentabilidad_temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
COMMIT;
--
-- ***** Obtiene a Cargar los Valores para el Calculo de la Rentabilidad
--
ps_rentabilidad (Ln_codigo_empresa, --pcod_empresa          number,
USER, --p_user          varchar2,
Ld_fecha_inicial, --p_fecha_inicial          date,
Ld_fecha_final --p_fecha_final          date
);
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
FOR C IN C_rentabilidad LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.empresa, --Pn_codigo_empresa IN NUMBER,
C.cod_producto, --Pv_cod_producto IN VARCHAR2,
Lv_cod_modelo_cat_linea, --Pv_cod_modelo_cat IN OUT VARCHAR2,
```



```

                                Lv_cod_item_cat_linea,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                Lv_CodigoError
                                );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
--
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DE LA MARCA DE UN PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 15 de Diciembre de 2009.
PK_OBTENER_PRODUCTO.P_COD_MARCA (C.empresa,--Pn_codigo_empresa IN      NUMBER,
                                C.cod_producto,--Pv_cod_producto IN  VARCHAR2,
                                Ln_cod_marca,--Pn_codigo_marca  IN OUT NUMBER,
                                Lv_CodigoError
                                );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
--
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia IN      NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion IN      NUMBER,
                                                C.empresa,--Pn_codigo_empresa IN      NUMBER,
                                                Lv_codigo_canton,--Pv_codigo_canton IN OUT VARCHAR2,
```



```
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
--
-- de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
C.empresa,--Pn_codigo_empresa      IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
UPDATE dm_indice_rentabilidad indren
SET   indren.costos_ventas = indren.costos_ventas + C.costos_ventas,
      indren.precio_ventas = indren.precio_ventas + C.precio_ventas,
      indren.costos_devoluciones = indren.costos_devoluciones + C.costos_devoluciones,
      indren.precio_devoluciones = indren.precio_devoluciones + C.precio_devoluciones
WHERE indren.periodo      = Ln_periodo_balance
AND   indren.cod_producto = C.cod_producto
AND   indren.anio_mes_venta = C.fecha
AND   indren.forma_pago    = C.forma_pago
```



```
AND indren.plazo = C.num_pagos
AND indren.codigo_agencia = C.cod_agencia
AND indren.codigo_canton = Lv_codigo_canton
AND indren.codigo_provincia = Lv_codigo_provincia
AND indren.codigo_region = Ln_codigo_region
AND indren.cod_modelo_cat_linea = Lv_cod_modelo_cat_linea
AND indren.cod_item_cat_linea = Lv_cod_item_cat_linea
AND indren.cod_marca = Ln_cod_marca
AND indren.tipo_cliente = C.tipo_cliente
AND indren.codigo_nacion = Ln_codigo_nacion
AND indren.codigo_empresa = C.empresa;
--
-- Verifica si Existe Registro
--
IF SQL%NOTFOUND THEN
Lv_cod_producto := C.cod_producto;
Lv_tipo_cliente := C.tipo_cliente;
Lv_cod_modelo_cliente := C.cod_modelo;
Lv_cod_item_cliente := C.cod_item;
INSERT INTO dm_indice_rentabilidad (PERIODO
COD_PRODUCTO
ANIO_MES_VENTA
FORMA_PAGO
PLAZO
CODIGO_AGENCIA
TIPO_CLIENTE
COD_ITEM_TIPO_CLIENTE
COD_MODELO_ITEM_TIPO_CLIENTE
CODIGO_NACION
CODIGO_EMPRESA
COSTO_VENTAS
PRECIO_VENTAS
COSTO_DEVOLUCIONES
PRECIO_DEVOLUCIONES
MES_CERRADO
)
VALUES (Ln_periodo_balance, --PERIODO NOT NULL NUMBER(6)
```



```

C.cod_producto,--COD_PRODUCTO                NOT NULL VARCHAR2(14)
C.fecha,--ANIO_MES_VENTA                     NOT NULL NUMBER(6)
C.forma_pago,--FORMA_PAGO                    NOT NULL VARCHAR2(3)
C.num_pagos,--PLAZO                          NOT NULL NUMBER(3)
C.cod_agencia,--CODIGO_AGENCIA               NOT NULL NUMBER(4)
C.tipo_cliente,--TIPO_CLIENTE               NOT NULL VARCHAR2(50)
C.cod_item,--COD_ITEM_TIPO_CLIENTE          NOT NULL VARCHAR2(3)
C.cod_modelo,--COD_MODELO_ITEM_TIPO_CLIENTE NOT NULL VARCHAR2(8)
Ln_codigo_nacion,--CODIGO_NACION            NOT NULL NUMBER(3)
C.empresa,--CODIGO_EMPRESA                  NOT NULL NUMBER(2)
C.costos_ventas,--COSTO_VENTAS              NOT NULL NUMBER(30,2)
C.precio_ventas,--PRECIO_VENTAS             NOT NULL NUMBER(30,2)
C.costos_devoluciones,--COSTO_DEVOLUCIONES  NOT NULL NUMBER(30,2)
C.precio_devoluciones,--PRECIO_DEVOLUCIONES NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                             NOT NULL VARCHAR2(1)
);
END IF;-- IF SQL%NOTFOUND THEN
END LOOP;-- FOR C IN C_rentabilidad LOOP
--
--
--
COMMIT;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta_rentabilidad_temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
--
COMMIT;
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
EXCEPTION
WHEN OTHERS THEN
```



```
        raise_application_error (-20003, 'Ln_perodo_balance='||Ln_perodo_balance||'  
C.cod_producto='||Lv_cod_producto||  
        ' Lv_tipo_cliente='||Lv_tipo_cliente||  
        ' Lv_cod_modelo_cliente='||Lv_cod_modelo_cliente||  
        ' Lv_cod_item_cliente='||Lv_cod_item_cliente||' '||  
        SQLERRM);  
  
END;
```



Anexo A25.

Script carga inicial tabla cc_indice_de_morosidad.

```
--  
--  
*****  
**  
--  
-- Procedimiento      : Carga el Indice de Morosidad Para el Nuevo Mes  
--                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes  
--                   : Inserta en la Tabla CC INDICE DE MOROSIDAD  
-- Autor              : Ing. Giovanni Chasifan Chicaiza.  
-- Fecha creacion     : Miercoles 1 de Diciembre de 2010.  
DECLARE  
--  
-- Variables Locales  
--  
Lv_CodigoError          VARCHAR2(3000);  
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;  
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;  
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;  
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;  
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;  
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;  
Ln_tenia_que_cobrar     NUMBER(30,2);  
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;  
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;  
Lv_nombre_producto      producto.nombre%TYPE;  
Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;  
Lv_cod_item_cat         producto.cod_item_cat%TYPE;  
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;  
Ln_siguiete_periodo_trabajo NUMBER(6);  
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;  
Ln_contador             NUMBER(30);
```



```
Ld_fecha_inicial_mes_sig          cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig            cc_periodos_cerrados.fecha_final%TYPE;
Lv_cartera                        cc_periodos_cerrados.cartera%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2; -- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
  SELECT periodo_cierre, fecha_inicial, fecha_final
  FROM   cc_periodos_cerrados percer
  WHERE  codigo_empresa = Ln_codigo_empresa
  AND    carterera          = Lv_cartera
  AND NOT EXISTS (SELECT 'X'
                  FROM   cc_indice_de_morosidad indmor
                  WHERE  indmor.periodo          = percer.periodo_cierre
                  AND    indmor.codigo_empresa = percer.codigo_empresa
                  )
  GROUP BY periodo_cierre, fecha_inicial, fecha_final
  ORDER BY 1;
--
-- Obtiene todos los Ventas realizadas
--
CURSOR C_siembra_x_vencimiento IS
  SELECT /*+ first_rows */ sum(siexven.valor_vencimiento) valor_ventas,
        siexven.anio_mes_venta,
        siexven.codigo_agencia,
        siexven.codigo_canton,
        siexven.codigo_provincia,
        siexven.codigo_region,
```



```
        siexven.codigo_politica,  
        siexven.cod_modelo_cat_linea,  
        siexven.cod_item_cat_linea,  
        siexven.cod_marca,  
        siexven.codigo_nacion,  
        siexven.codigo_empresa  
FROM    cc_siembra_x_vencimiento siexven  
WHERE   siexven.codigo_empresa = Ln_codigo_empresa  
GROUP BY siexven.anio_mes_venta,  
        siexven.codigo_agencia,  
        siexven.codigo_canton,  
        siexven.codigo_provincia,  
        siexven.codigo_region,  
        siexven.codigo_politica,  
        siexven.cod_modelo_cat_linea,  
        siexven.cod_item_cat_linea,  
        siexven.cod_marca,  
        siexven.codigo_nacion,  
        siexven.codigo_empresa;  
  
--  
  
-- Obtiene los creditos de un determinado periodo  
--  
CURSOR C_balances IS  
SELECT /*+ first_rows */ *  
FROM    cc_balances_mensuales balmen  
WHERE   balmen.periodo_cierre = Ln_periodo_balance  
AND     balmen.codigo_empresa = Ln_codigo_empresa  
ORDER BY balmen.anio_venta, balmen.mes_venta;  
BEGIN  
  
--  
-- ***** INSERTA TODAS LAS VENTAS EN EL PERIODO CERRADO  
--  
  
--  
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto  
--  
FOR J IN C_tipos_carteras LOOP  
    Ln_codigo_empresa := J.codigo_empresa;
```



```
Lv_cartera := J.cartera;
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
FOR I IN C_periodos_cerrados LOOP
  Ln_periodo_balance := I.periodo_cierre;
  Ld_fecha_inicial := I.fecha_inicial;
  Ld_fecha_final := I.fecha_final;
  --
  --
  --
  Ln_contador := 0;
--
-- ***** Obtiene todos los Ventas realizadas
--
FOR C IN C_siembra_x_vencimiento LOOP
  --
  --
  *****
  **
  --
  -- Función : OBTIENE EL TOTAL DE VENCIMIENTOS (SIEMBRA) A UN DETERMINADO ANIO Y MES DE
VENCIMIENTO
  -- en la tabla CC_SIEMBRA_X_VENCIMIENTO
  -- Autor : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación : Martes 14 de Diciembre de 2010.
  Ln_tenia_que_cobrar := PK_OBTENER_CC_SIE_X_VEN.F_TENIA_QUE_COBRAR (Ln_periodo_balance, --
Pn_anio_mes_vencimiento IN NUMBER,--ANIO_MES_VENCIMIENTO NOT NULL NUMBER(6)
C.anio_mes_venta,--Pn_anio_mes_venta
IN NUMBER,--ANIO_MES_VENTA NOT NULL NUMBER(6)
C.codigo_agencia,--Pn_codigo_agencia
IN NUMBER,--CODIGO AGENCIA NOT NULL NUMBER(4)
C.codigo_canton,--Pv_codigo_canton
IN VARCHAR2,--CODIGO_CANTON NOT NULL VARCHAR2(4)
C.codigo_provincia,--
Pv_codigo_provincia IN VARCHAR2,--CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
```



```
IN NUMBER,--CODIGO_REGION NOT NULL NUMBER(3)
IN NUMBER,--CODIGO_POLITICA NOT NULL NUMBER(2)
Pv_cod_modelo_cat_linea IN VARCHAR2,--COD_MODELO_CAT_LINEA
Pv_cod_item_cat_linea IN VARCHAR2,--COD_ITEM_CAT_LINEA
IN NUMBER,--COD MARCA NOT NULL NUMBER(3)
IN NUMBER,--CODIGO_NACION NOT NULL NUMBER(3)
IN NUMBER--CODIGO_EMPRESA NOT NULL NUMBER(2)

--
-- Inserta el Indice de Morosidad
--
INSERT INTO cc_indice_de_morosidad (PERIODO
ANIO_MES_VENTA
CODIGO_AGENCIA
CODIGO_CANTON
CODIGO_PROVINCIA
CODIGO_REGION
CODIGO_POLITICA
COD_MODELO_CAT_LINEA
COD_ITEM_CAT_LINEA
COD_MARCA
CODIGO_NACION
CODIGO_EMPRESA
VALOR_VENTAS
SALDO_CARTERA
TOTAL_CARTERA_AL_CIERRE
CARTERA_VENCIDA
MES_CERRADO
)
VALUES (Ln_periodo_balance,--PERIODO
```

```
C.codigo_region,--Pn_codigo_region
C.codigo_politica,--Pn_codigo_politica
C.cod_modelo_cat_linea,--
NOT NULL VARCHAR2(8)
C.cod_item_cat_linea,--
NOT NULL VARCHAR2(3)
C.cod_marca,--Pn_cod_marca
C.codigo_nacion,--Pn_codigo_nacion
C.codigo_empresa--Pn_codigo_empresa
);
```



```

C.anio_mes_venta,--ANIO_MES_VENTA                NOT NULL NUMBER(6)
C.codigo_agencia,--CODIGO_AGENCIA                 NOT NULL NUMBER(4)
C.codigo_canton,--CODIGO_CANTON                   NOT NULL VARCHAR2(4)
C.codigo_provincia,--CODIGO_PROVINCIA              NOT NULL VARCHAR2(6)
C.codigo_region,--CODIGO_REGION                   NOT NULL NUMBER(3)
C.Codigo_Politica,--CODIGO_POLITICA               NOT NULL NUMBER(2)
C.cod_modelo_cat_linea,--COD_MODELO_CAT_LINEA      NOT NULL VARCHAR2(8)
C.cod_item_cat_linea,--COD_ITEM_CAT_LINEA         NOT NULL VARCHAR2(3)
C.cod_marca,--COD_MARCA                           NOT NULL NUMBER(3)
C.codigo_nacion,--CODIGO_NACION                   NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                 NOT NULL NUMBER(2)
C.valor_ventas,--VALOR_VENTAS                     NOT NULL NUMBER(30,2)
0,--SALDO_CARTERA                                NOT NULL NUMBER(30,2)
Ln_tenia_que_cobrar,--TOTAL_CARTERA_AL_CIERRE     NOT NULL NUMBER(30,2)
0,--CARTERA_VENCIDA                               NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                                  NOT NULL VARCHAR2(1)
);

--
--
--
Ln_contador := Ln_contador + 1;
IF Ln_contador > 2000 THEN
  COMMIT;
  Ln_contador := 0;
END IF;-- IF Ln_contador > 2000 THEN
END LOOP;-- FOR C_siembra_x_vencimiento LOOP
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
```



```
--  
--  
*****  
**  
  
--  
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA  
--                  : de la tabla AD_PARAMETROS_GENERALES  
-- Autor            : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación  : Martes 19 de Octubre de 2010.  
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN  
NUMBER  
                                );  
  
IF Ln_codigo_nacion IS NULL THEN  
    Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.  
Empresa='||to_char(J.codigo_empresa);  
    raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Ln_codigo_nacion IS NULL THEN  
Ln_codigo_empresa := J.codigo_empresa;  
Lv_cartera        := J.cartera;  
--  
-- Obtiene los Periodos de los Balances Por Empresa y Cartera  
--  
FOR I IN C_periodos_cerrados LOOP  
    Ln_periodo_balance := I.periodo_cierre;  
    Ld_fecha_inicial   := I.fecha_inicial;  
    Ld_fecha_final     := I.fecha_final;  
    --  
    -- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance  
    --  
    Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
    Ld_fecha_final_mes_sig  := add_months(trunc(Ld_fecha_final),1);  
    --  
    --  
    --  
    -- Obtiene los creditos de un determinado periodo  
    --
```



```
FOR C IN C_balances LOOP
  Ln_tenia_que_cobrar := 0;
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
  --                    de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                  Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                  J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                  Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,
                                                  Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,
                                                  Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
  --                    de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Viernes 26 de Noviembre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                              Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,
                                              Ln_codigo_nacion,--Pn_codigo_nacion     IN      NUMBER,
                                              J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                              Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                              Lv_CodigoError
```



```
);  
IF Lv_CodigoError IS NOT NULL THEN  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la  
Factura  
--  
Lv_cod_modelo_cat := C.cod_modelo_cat;  
Lv_cod_item_cat := C.cod_item_cat;  
--  
-- Verifica SI existe dato en el codigo de la Linea  
--  
IF Lv_cod_modelo_cat IS NULL THEN  
--  
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante  
--  
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,  
C.tipo_comprobante,  
C.cod_comprobante);  
  
IF Lv_cod_producto = ' ' THEN  
  Lv_cod_producto := NULL;  
END IF;-- IF :NEW.cod_producto = ' ' THEN  
--  
--  
--  
IF Lv_cod_producto IS NOT NULL THEN  
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO  
--                    : de la tabla STOCK.PRODUCTO  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Domingo 7 de Enero de 2007.  
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
```



```

Lv_cod_producto,--Pv_cod_producto  IN  VARCHAR2,
Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
  --                    : de la tabla STOCK.PRODUCTO
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Domingo 7 de Enero de 2007.
  PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN  NUMBER,
                                             Lv_cod_producto,--Pv_cod_producto  IN  VARCHAR2,
                                             Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                             Lv_cod_item_cat,--Pv_cod_item_cat  IN OUT VARCHAR2,
                                             Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
    Lv_CodigoError := null;
  ELSE-- IF Lv_CodigoError IS NOT NULL THEN
    --
    --
    ****
    **
    --
    -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
    --                    : de la tabla TG MODELO_ITEM
    -- Autor              : Ing. Giovanni Chasifán Chicaiza.
    -- Fecha creación    : Domingo 7 de Enero de 2007.
    PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN  NUMBER,
```



```

Lv_cod_modelo_cat, --Pv_cod_modelo      IN      VARCHAR2,
Lv_cod_item_cat, --Pv_cod_item          IN      VARCHAR2,
Lv_nombre_linea, --Pv_nombre            IN OUT VARCHAR2,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;

END IF; -- IF Lv_CodigoError IS NOT NULL THEN
END IF; -- IF Lv_CodigoError IS NOT NULL THEN
ELSE -- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
END IF; -- IF Lv_cod_producto IS NOT NULL THEN
END IF; -- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF; -- IF C.codigo_marca IS NULL THEN
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_indice_de_morosidad indmor
SET     indmor.saldo_cartera      = indmor.saldo_cartera + C.Saldo_Del_Credito,
        indmor.valor_cobrado     = indmor.valor_ventas - indmor.saldo_cartera,
        indmor.cartera_vencida   = indmor.cartera_vencida + C.Total_Vencido
WHERE   indmor.periodo           = Ln_periodo_balance
AND     indmor.anio_mes_venta    = C.anio_venta || LPAD(C.mes_venta,2,0)
AND     indmor.codigo_agencia    = C.codigo_agencia
AND     indmor.codigo_canton     = Lv_codigo_canton
AND     indmor.codigo_provincia  = Lv_codigo_provincia
AND     indmor.codigo_region     = Ln_codigo_region
AND     indmor.codigo_politica   = C.codigo_politica
AND     indmor.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND     indmor.cod_item_cat_linea = Lv_cod_item_cat
```



```
AND      indmor.cod_marca          = Ln_codigo_marca
AND      indmor.codigo_nacion     = Ln_codigo_nacion
AND      indmor.codigo_empresa   = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  raise_application_error (-20003,'No Existe el Registro.'||chr(10)||
    ' Compr='||C.cod_comprobante||
    ' TipCompr='||C.tipo_comprobante||
    ' Periodo='||Ln_periodo_balance||
    ' AnioMesVta='||C.anio_venta||LPAD(C.mes_venta,2,0)||
    ' Agencia='||C.codigo_agencia||
    ' Canton='||Lv_codigo_canton||
    ' Provin='||Lv_codigo_provincia||
    ' Region='||Ln_codigo_region||
    ' Polit='||C.codigo_politica||
    ' ModelLinea='||Lv_cod_modelo_cat||
    ' ItemLinea='||Lv_cod_item_cat||
    ' Marca='||Ln_codigo_marca||
    ' Nacion='||Ln_codigo_nacion||
    ' Empresa='||C.codigo_empresa||chr(10)||
    ' en la Tabla CC_INDICE_DE_MOROSIDAD.
PK_CC_PROCESO_INDICE_MOROSIDAD.P_CARGAR_IND_MOR_CIERRE_MES.'||chr(10)||
    SQLERRM
  );
  END IF;-- IF Sql%Notfound THEN
  END LOOP;-- FOR C IN C_balances LOOP
  END LOOP;-- FOR I IN C_periodos_cerrados LOOP
  END LOOP;-- FOR J IN C_tipos_carteras LOOP
  --
  --
  --
  COMMIT;
  --
  -- ***** CARGA LOS DATOS PARA EL SIGUIENTE MES DE TRABAJO
  -- ***** PARA PODER EVALUAR DIARIAMENTE COMO ESTA EL INDICE DE MOROSIDAD
```



```
--
--
-- Obtiene los Tipos de Cartera
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_empresa := J.codigo_empresa;
  Lv_cartera        := J.cartera;
  --
  -- Obtiene los Periodos de los Balances Por Empresa y Cartera
  --
  FOR I IN C_periodos_cerrados LOOP
    Ln_periodo_balance := I.periodo_cierre;
    Ld_fecha_inicial   := I.fecha_inicial;
    Ld_fecha_final     := I.fecha_final;
    --
    -- Obtiene el Anio y Mes del siguiente periodo de Trabajo
    --
    Ln_siguiente_periodo_trabajo := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
  --
  -- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
  -- Como datos definitos del cierre de mes
  --
  INSERT INTO cc_indice_de_morosidad
  SELECT
    Ln_siguiente_periodo_trabajo, -- PERIODO
    ANIO_MES_VENTA
    CODIGO_AGENCIA
    CODIGO_CANTON
    CODIGO_PROVINCIA
    CODIGO_REGION
    CODIGO_POLITICA
    COD_MODELO_CAT_LINEA
    COD_ITEM_CAT_LINEA
    COD_MARCA
    CODIGO_NACION
    CODIGO_EMPRESA
    NOMBRE_AGENCIA
```



```
NOMBRE_CANTON /
NOMBRE_PROVINCIA /
NOMBRE_REGION /
NOMBRE_LINEA /
NOMBRE_MARCA /
NOMBRE_POLITICA /
VALOR_VENTAS /
SALDO_CARTERA /
TOTAL_CARTERA_AL_CIERRE /
VALOR_COBRADO /
CARTERA_VENCIDA /
0,--PORCENTAJE_PRODUCTIVIDAD /
0,--PORCENTAJE_INDICE_MOROSIDAD /
0,--PAGOS_ANTICIPADOS /
'A',--MES_CERRADO /
FECHA_ULT_EJECUCION /
ADICIONADO_POR /
FECHA_ADICION /
MODIFICADO_POR /
FECHA_MODIFICACION /
FROM cc_indice_de_morosidad indmor
WHERE indmor.periodo = Ln_periodo_balance
AND indmor.mes_cerrado = 'C'--A=MES ABIERTO, C=MES CERRADO
AND indmor.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A26.

Script carga inicial tabla cc_flujo_corriente.

```
--  
--  
*****  
**  
--  
-- Procedimiento      : Carga los VENCIMIENTOS QUE SE VENDE EN EL MES DEL BALANCE  
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.  
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes  
--                    : Inserta en la Tabla CC_FLUJO_CORRIENTE  
-- Autor              : Ing. Giovanni Chasifan Chicaiza.  
-- Fecha creacion     : Lunes 13 de Diciembre de 2010.  
DECLARE  
--  
-- Variables Locales  
--  
Lv_CodigoError          VARCHAR2(3000);  
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;  
Ln_periodo_balance_ant  cc_periodos_cerrados.periodo_cierre%TYPE;  
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;  
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;  
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;  
Ln_Cobro_efectivo       NUMBER(30,2);  
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);  
Ln_descuento_mes_sig    NUMBER(30,2);  
Ln_descuento            NUMBER(30,2);  
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;  
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;  
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;  
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;  
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
```



```
Lv_cod_producto          cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion     NUMBER(30,2);
Lv_cartera              cc_periodos_cerrados.cartera%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
  SELECT periodo_cierre, fecha_inicial, fecha_final
  FROM   cc_periodos_cerrados percer
  WHERE  codigo_empresa = Ln_codigo_empresa
  AND    carterã         = Lv_carterã
  AND    periodo_cierre > 200512
  AND    NOT EXISTS (SELECT 'X'
                    FROM   cc_flujo_corriente flucor
                    WHERE  flucor.periodo = percer.periodo_cierre
                    AND    flucor.codigo_empresa = percer.codigo_empresa
                    )
  GROUP BY periodo_cierre, fecha_inicial, fecha_final
  ORDER BY 1;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
```



```
SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
FROM   cc_balances_mensuales balmen,
       forma_pago_comprobante forpagcom
WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
AND    balmen.total_vencido = 0
AND    balmen.codigo_empresa = Ln_codigo_empresa
AND    forpagcom.cod_comprobante = balmen.cod_comprobante
AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND    (forpagcom.cod_forma_pago = 'CRE'
        OR forpagcom.cod_forma_pago = 'NDB'
        OR forpagcom.cod_forma_pago = 'TCR'
        OR forpagcom.cod_forma_pago = 'BAN'
        OR forpagcom.cod_forma_pago = 'ADI')
AND    forpagcom.vencimiento >= Ld_fecha_inicial
AND    forpagcom.vencimiento <= Ld_fecha_final
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                       J.cartera,--Pv_cartera                      IN      VARCHAR2,
                                                       Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
                                                       Lv_CodigoError);
```



```
IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- ***** Borrre las Transacciones ABIertas porque los datos seran cargados mas adelante
--           Como datos definitos del cierre de mes
--
DELETE cc_flujo_corriente flucor
WHERE flucor.periodo      = Ln_periodo_balance
AND   flucor.mes cerrado  = 'A'--A=MES ABIERTO, C=MES CERRADO
AND   flucor.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  Ln_codigo_empresa := J.codigo_empresa;
  Lv_cartera       := J.cartera;
  --
  --
  *****
  **
  --
  -- Funcion           : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
  --                   de la tabla AD_PARAMETROS_GENERALES
  -- Autor            : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación  : Martes 19 de Octubre de 2010.
  Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa           IN
NUMBER
                                     );
  IF Ln_codigo_nacion IS NULL THEN
```



```
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.  
Empresa='||to_char(J.codigo_empresa);  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Ln_codigo_nacion IS NULL THEN  
--  
-- Obtiene los Periodos de los Balances Por Empresa y Cartera  
--  
FOR I IN C_periodos_cerrados LOOP  
  Ln_periodo_balance := I.periodo_cierre;  
  Ld_fecha_inicial   := I.fecha_inicial;  
  Ld_fecha_final     := I.fecha_final;  
  --  
  -- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance  
  --  
  Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);  
  Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));  
  --  
  -- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance  
  --  
  Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
  Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);  
  --  
  -- Obtiene los creditos de un determinado periodo  
  --  
  FOR C IN C_balances LOOP  
    --  
    --  
    *****  
    **  
    --  
    -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y  
EMPRESA  
    --                      de la tabla AD_AGENCIAS_X_CANTONES  
    -- Autor              : Ing. Giovanni Chasifán Chicaiza.  
    -- Fecha creación     : Jueves 21 de Octubre de 2010.  
    PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia   IN      NUMBER,  
                                                    Ln_codigo_nacion,--Pn_codigo_nacion   IN      NUMBER,
```



```
J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
Lv_codigo_canton,--Pv_codigo_canton      IN OUT VARCHAR2,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
--
-- de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanny Chasifán Chicaiza.
-- Fecha creación     : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia  IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion   IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
```



```
--  
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante  
--  
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,  
                                                    C.tipo_comprobante,  
                                                    C.cod_comprobante);  
  
IF Lv_cod_producto = ' ' THEN  
  Lv_cod_producto := NULL;  
END IF;-- IF :NEW.cod_producto = ' ' THEN  
--  
--  
--  
IF Lv_cod_producto IS NOT NULL THEN  
  --  
  --  
  *****  
  **  
  
  --  
  -- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO  
  --                    : de la tabla STOCK.PRODUCTO  
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.  
  -- Fecha creación    : Domingo 7 de Enero de 2007.  
  PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,  
                                         Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,  
                                         Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,  
                                         Lv_CodigoError);  
  
  IF Lv_CodigoError IS NOT NULL THEN  
    Lv_cod_modelo_cat := '0';  
    Lv_cod_item_cat := '0';  
    Lv_CodigoError := null;  
  ELSE-- IF Lv_CodigoError IS NOT NULL THEN  
    --  
    --  
    *****  
    **  
  
    --  
    -- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
```



```
--                                de la tabla STOCK.PRODUCTO
-- Autor                          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación                 : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento                : Obtiene EL NOMBRE DEL MODELO ITEM
  --                               de la tabla TG_MODELO_ITEM
  -- Autor                          : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación                 : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item         IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                   Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
```



```
--  
-- Verifica si el codigo de la Marca es NULL  
--  
Ln_codigo_marca := C.codigo_marca;  
IF C.codigo_marca IS NULL THEN  
    Ln_codigo_marca := 0;  
END IF;-- IF C.codigo_marca IS NULL THEN  
  
--  
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado  
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a  
nivel de Oracle  
--  
--  
--  
*****  
**  
  
--  
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO  
Y RANGO DE FECHAS  
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION  
-- Autor            : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación  : Jueves 3 de Junio de 2010.  
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --  
Pv_cod_comprobante      IN      VARCHAR2,  
  
C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,  
  
C.secuencia, --  
Pn_numero_vencimiento  IN      NUMBER,  
  
C.Codigo_Empresa, --  
Pn_empresa             IN      NUMBER,  
  
to_date('2000-01-  
01','yyyy-mm-dd'), --Ld_fecha_inicial, --Pd_fecha_inicial      IN      DATE,  
  
Ld_fecha_final --  
Pd_fecha_final         IN      DATE  
  
);  
--
```



```
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln Cobro efectivo mes sig := PK OBTENER ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante    IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa,-- Pn_empresa                IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final        IN      DATE
C.secuencia,--
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función           : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Viernes 6 de Agosto de 2010.
```



```
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
                                                                    C.Tipo_Comprobante, --
Pv_tipo_comprobante     IN      VARCHAR2,
                                                                    C.secuencia, --
Pn_numero_vencimiento  IN      NUMBER,
                                                                    C.Codigo_Empresa, --Pn_empresa
IN      NUMBER,
                                                                    to_date('2000-01-01','yyyy-mm-
dd'),--Ld fecha inicial,--Pd fecha inicial      IN      DATE,
                                                                    Ld_fecha_final--Pd_fecha_final
IN      DATE
                                                                    );
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
                                                                    C.Tipo_Comprobante, --
Pv_tipo_comprobante     IN      VARCHAR2,
                                                                    C.secuencia, --
Pn_numero_vencimiento  IN      NUMBER,
                                                                    C.Codigo_Empresa, --
Pn_empresa              IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final          IN      DATE
                                                                    );
--
```



```
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, --
Pv tipo comprobante IN VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa, --
Pn_empresa IN NUMBER,
to_date('2000-01-
01','yyyy-mm-dd'), --Pd_fecha_inicial IN DATE,
Id_fecha_final --
Pd_fecha_final IN DATE
);
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_flujo_corriente flucor
SET flucor.valor_vencimiento = flucor.valor_vencimiento + (C.valor - Ln_valor_devolucion),
flucor.saldo_vencimiento = flucor.saldo_vencimiento + C.saldo +
Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig,
flucor.cobro_efectivo_vencimiento = flucor.cobro_efectivo_vencimiento + Ln_Cobro_efectivo
WHERE flucor.periodo = Ln_periodo_balance
AND flucor.fecha_vencimiento = C.vencimiento
```



```
AND flucor.codigo_agencia = C.codigo_agencia
AND flucor.codigo_canton = Lv_codigo_canton
AND flucor.codigo_provincia = Lv_codigo_provincia
AND flucor.codigo_region = Ln_codigo_region
AND flucor.codigo_politica = C.Codigo_Politica
AND flucor.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND flucor.cod_item_cat_linea = Lv_cod_item_cat
AND flucor.cod_marca = Ln_codigo_marca
AND flucor.cod_cat_cliente = C.cod_cat_cliente
AND flucor.es empleado = C.es empleado
AND flucor.codigo_nacion = Ln_codigo_nacion
AND flucor.codigo_empresa = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_flujo_corriente (PERIODO
                                FECHA_VENCIMIENTO
                                CODIGO_AGENCIA
                                CODIGO_CANTON
                                CODIGO_PROVINCIA
                                CODIGO_REGION
                                CODIGO_POLITICA
                                COD_MODELO_CAT_LINEA
                                COD_ITEM_CAT_LINEA
                                COD_MARCA
                                COD_CAT_CLIENTE
                                ES_EMPLEADO
                                CODIGO_NACION
                                CODIGO_EMPRESA
                                valor_vencimiento
                                saldo_vencimiento
                                cobro_efectivo_vencimiento
                                MES_CERRADO
                                )
  VALUES (Ln_periodo_balance, --PERIODO
          C.vencimiento, --fecha_vencimiento
          NOT NULL NUMBER(6)
          NOT NULL DATE
```



```
C.codigo_agencia,--CODIGO_AGENCIA NOT NULL NUMBER(4)
Lv_codigo_canton,--CODIGO_CANTON NOT NULL VARCHAR2(4)
Lv_codigo_provincia,--CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
Ln_codigo_region,--CODIGO_REGION NOT NULL NUMBER(3)
C.Codigo_Politica,--CODIGO_POLITICA NOT NULL NUMBER(2)
Lv_cod_modelo_cat,--COD_MODELO_CAT_LINEA NOT NULL VARCHAR2(8)
Lv_cod_item_cat,--COD_ITEM_CAT_LINEA NOT NULL VARCHAR2(3)
Ln_codigo_marca,--COD_MARCA NOT NULL NUMBER(3)
C.cod_cat_cliente,--COD_CAT_CLIENTE NOT NULL VARCHAR2(8)
C.es empleado,--ES EMPLEADO NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA NOT NULL NUMBER(2)
(C.valor - Ln_valor_devolucion),--C.valor,--valor_vencimiento NOT
NULL NUMBER(30,2)
(C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--saldo_vencimiento
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo,--cobro_efectivo NOT NULL NUMBER(30,2)
'C'--MES_CERRADO NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
raise_application_error (-20003,SQLERRM);
END;
```



Anexo A27.

Script carga inicial tabla cc_produc_mes_presente_puntual.

```
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_PRODUC_MES_PRESENTE_PUNTUAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
DECLARE
--
-- Variables Locales
--
Lv_CodigoError        VARCHAR2(3000);
Ln_periodo_balance    cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial      cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final        cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo     NUMBER(30,2);
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig  NUMBER(30,2);
Ln_descuento          NUMBER(30,2);
Ln_codigo_nacion      ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton      ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region      ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia   ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa     ad_empresas.codigo_empresa%TYPE;
```



```
Lv_cod_producto          cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat          producto.cod_item_cat%TYPE;
Lv_nombre_linea          tg_modelo_item.nombre%TYPE;
Ln_codigo_marca          cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion      NUMBER(30,2);
Lv_cartera              cc_periodos_cerrados.cartera%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
  SELECT periodo_cierre, fecha_inicial, fecha_final
  FROM   cc_periodos_cerrados percer
  WHERE  codigo_empresa = Ln_codigo_empresa
  AND    carterā        = Lv_carterā
  AND    periodo_cierre > 200512
  AND    NOT EXISTS (SELECT 'X'
                    FROM   cc_produc_mes_presente_puntual mesprepun
                    WHERE  mesprepun.periodo          = percer.periodo_cierre
                    AND    mesprepun.codigo_empresa = percer.codigo_empresa
                    )
  GROUP BY periodo_cierre, fecha_inicial, fecha_final
  ORDER BY 1;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
```



```
SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
FROM   cc_balances_mensuales balmen,
       forma_pago_comprobante forpagcom
WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
AND    balmen.total_vencido  = 0
AND    balmen.codigo_empresa = Ln_codigo_empresa
AND    forpagcom.cod_comprobante = balmen.cod_comprobante
AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND    forpagcom.secuencia    <> 1
AND    forpagcom.empresa      = balmen.codigo_empresa
AND    (forpagcom.cod_forma_pago = 'CRE'
       OR forpagcom.cod_forma_pago = 'NDB'
       OR forpagcom.cod_forma_pago = 'TCR'
       OR forpagcom.cod_forma_pago = 'BAN'
       OR forpagcom.cod_forma_pago = 'ADI')
AND    forpagcom.vencimiento  >= Ld_fecha_inicial
AND    forpagcom.vencimiento  <= Ld_fecha_final
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  Ln_codigo_empresa := J.codigo_empresa;
  Lv_cartera       := J.cartera;
--
--
*****
**
--
-- Funcion           : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                   : de la tabla AD_PARAMETROS_GENERALES
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
```



```
);  
  
IF Ln_codigo_nacion IS NULL THEN  
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.  
Empresa='||to_char(J.codigo_empresa);  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Ln_codigo_nacion IS NULL THEN  
--  
-- Obtiene los Periodos de los Balances Por Empresa y Cartera  
--  
FOR I IN C periodos cerrados LOOP  
  Ln_periodo_balance := I.periodo_cierre;  
  Ld_fecha_inicial   := I.fecha_inicial;  
  Ld_fecha_final     := I.fecha_final;  
  --  
  --  
  --  
  -- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance  
  --  
  Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);  
  Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));  
  --  
  -- Obtiene la Fecha Inicial y Final del Siguiente Mes del Balance  
  --  
  Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
  Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);  
  --  
  -- Obtiene los creditos de un determinado periodo  
  --  
  FOR C IN C_balances LOOP  
    --  
    --  
*****  
**  
    --  
    -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y  
EMPRESA
```



```
--          de la tabla AD AGENCIAS X CANTONES
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                Lv_codigo_canton,--Pv_codigo_canton    IN OUT VARCHAR2,
                                                Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
EMPRESA
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
--
--          de la tabla AD_AGENCIAS_X_CANTONES
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                                Lv_CodigoError
                                                );

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
```



```
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                    C.tipo_comprobante,
                                                    C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
  Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
```



```
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
*****
**

--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                    Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                    Lv_cod_item_cat,--Pv_cod_item         IN      VARCHAR2,
                                    Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                    Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
```



```
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
Pn_empresa IN NUMBER,
01,'yyyy-mm-dd'), --Ld_fecha_inicial, --Pd_fecha_inicial IN DATE,
```

```
C.secuencia, --
C.Codigo_Empresa, --
to_date('2000-01-
```



```
Pd_fecha_final          IN          DATE                                Ld_fecha_final--
                                                                    );
--
--
*****
**
--
-- Función                : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                        : de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor                  : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación        : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante          IN          VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante        IN          VARCHAR2,
Pn_numero_vencimiento IN          NUMBER,
- Pn_empresa           IN          NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN          DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final          IN          DATE
                                                                    );
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
--
```



```
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,  
VENCIMIENTO Y RANGO DE FECHAS  
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION  
-- Autor            : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación   : Viernes 6 de Agosto de 2010.  
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --  
Pv_cod_comprobante      IN      VARCHAR2,                                     C.Tipo_Comprobante, --  
Pv_tipo_comprobante     IN      VARCHAR2,                                     C.secuencia, --  
Pn_numero_vencimiento   IN      NUMBER,                                       C.Codigo_Empresa, --Pn_empresa  
IN      NUMBER,                                       to_date('2000-01-01','yyyy-mm-  
dd'), --Ld_fecha_inicial, --Pd_fecha_inicial      IN      DATE,                                       Ld_fecha_final --Pd_fecha_final  
IN      DATE                                       );  
--  
--  
*****  
**  
--  
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,  
VENCIMIENTO Y RANGO DE FECHAS  
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION  
-- Autor            : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación   : Viernes 6 de Agosto de 2010.  
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --  
Pv_cod_comprobante      IN      VARCHAR2,                                     C.Tipo_Comprobante, --  
Pv tipo comprobante     IN      VARCHAR2,                                     C.secuencia, --  
Pn_numero_vencimiento   IN      NUMBER,                                       C.Codigo_Empresa, --  
Pn_empresa              IN      NUMBER,
```



```
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,

Ld_fecha_final_mes_sig--Pd_fecha_final          IN      DATE

);

--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante  IN      VARCHAR2,
--                  C.Tipo_Comprobante, --
Pv_tipo_comprobante IN      VARCHAR2,
--                  C.secuencia, --
Pn_numero_vencimiento IN    NUMBER,
--                  C.Codigo_Empresa, --
Pn_empresa          IN      NUMBER,
--                  to_date('2000-01-
01','yyyy-mm-dd'),--Pd_fecha_inicial      IN      DATE,
--                  Ld_fecha_final--
Pd_fecha_final      IN      DATE
--                  );
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_produc_mes_presente_puntual promesprepun
```



```
SET      promesprepun.valor_vencimiento      = promesprepun.valor_vencimiento + (C.valor -
Ln_valor_devolucion),
        promesprepun.saldo_vencimiento      = promesprepun.saldo_vencimiento + C.saldo +
Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig,
        promesprepun.cobro_efectivo_vencimiento = promesprepun.cobro_efectivo_vencimiento +
Ln_Cobro_efectivo - Ln_descuento
WHERE    promesprepun.periodo                = Ln_periodo_balance
AND      promesprepun.fecha_vencimiento      = C.vencimiento
AND      promesprepun.codigo_agencia         = C.codigo_agencia
AND      promesprepun.codigo_canton          = Lv_codigo_canton
AND      promesprepun.codigo_provincia       = Lv_codigo_provincia
AND      promesprepun.codigo_region          = Ln_codigo_region
AND      promesprepun.codigo_politica        = C.codigo_politica
AND      promesprepun.cod_modelo_cat_linea    = Lv_cod_modelo_cat
AND      promesprepun.cod_item_cat_linea     = Lv_cod_item_cat
AND      promesprepun.cod_marca              = Ln_codigo_marca
AND      promesprepun.cod_cat_cliente        = C.cod_cat_cliente
AND      promesprepun.es_empleado            = C.es_empleado
AND      promesprepun.codigo_nacion          = Ln_codigo_nacion
AND      promesprepun.codigo_empresa        = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
INSERT INTO cc_produc_mes_presente_puntual (PERIODO
FECHA_VENCIMIENTO
CODIGO_AGENCIA
CODIGO_CANTON
CODIGO_PROVINCIA
CODIGO_REGION
CODIGO_POLITICA
COD_MODELO_CAT_LINEA
COD_ITEM_CAT_LINEA
COD_MARCA
COD_CAT_CLIENTE
ES_EMPLEADO
CODIGO_NACION
,
```



```

                                CODIGO_EMPRESA
                                valor_vencimiento
                                saldo_vencimiento
                                cobro_efectivo_vencimiento
                                MES_CERRADO
                                )
VALUES (Ln_periodo_balance, --PERIODO                                NOT NULL NUMBER(6)
       C.vencimiento, --fecha_vencimiento                            NOT NULL DATE
       C.codigo_agencia, --CODIGO_AGENCIA                            NOT NULL NUMBER(4)
       Lv_codigo_canton, --CODIGO_CANTON                             NOT NULL VARCHAR2(4)
       Lv_codigo_provincia, --CODIGO_PROVINCIA                       NOT NULL VARCHAR2(6)
       Ln_codigo_region, --CODIGO_REGION                             NOT NULL NUMBER(3)
       C.codigo_politica, --CODIGO_POLITICA                          NOT NULL NUMBER(2)
       Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA                     NOT NULL VARCHAR2(8)
       Lv_cod_item_cat, --COD_ITEM_CAT_LINEA                         NOT NULL VARCHAR2(3)
       Ln_codigo_marca, --COD_MARCA                                  NOT NULL NUMBER(3)
       C.cod_cat_cliente, --COD_CAT_CLIENTE                          NOT NULL VARCHAR2(8)
       C.es_empleado, --ES_EMPLEADO                                  NOT NULL VARCHAR2(100)
       Ln_codigo_nacion, --CODIGO_NACION                             NOT NULL NUMBER(3)
       C.codigo_empresa, --CODIGO_EMPRESA                            NOT NULL NUMBER(2)
       (C.valor - Ln_valor_devolucion), --C.valor, --valor_vencimiento NOT NULL NUMBER(30,2)
       (C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig), --C.saldo, --saldo_vencimiento
       NOT NULL NUMBER(30,2)
       (Ln_Cobro_efectivo - Ln_descuento), --cobro_efectivo        NOT NULL NUMBER(30,2)
       'C' --MES_CERRADO                                           NOT NULL VARCHAR2(1)
       );
END IF; -- IF Sql%Notfound THEN
END LOOP; -- FOR C IN C_balances LOOP
END LOOP; -- FOR I IN C_periodos_cerrados LOOP
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
```



```
raise_application_error(-20003,SQLERRM);  
END;
```



Anexo A28.

Script carga inicial tabla cc_produc_mes_presente_total.

```
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                    : SIN VALIDAR SI EL CREDITO TIENE CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_PRODUC_MES_PRESENTE_TOTAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
DECLARE
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant  cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo       NUMBER(30,2);
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig    NUMBER(30,2);
Ln_descuento            NUMBER(30,2);
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
```



```
Lv_cod_producto          cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
Lv_cartera              cc_periodos_cerrados.cartera%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
  SELECT periodo_cierre, fecha_inicial, fecha_final
  FROM   cc_periodos_cerrados percer
  WHERE  codigo_empresa = Ln_codigo_empresa
  AND    cartera        = Lv_cartera
  AND    periodo_cierre > 200512
  AND    NOT EXISTS (SELECT 'X'
                    FROM   cc_produc_mes_presente_total mespretot
                    WHERE  mespretot.periodo          = percer.periodo_cierre
                    AND    mespretot.codigo_empresa = percer.codigo_empresa
                    )
  GROUP BY periodo_cierre, fecha_inicial, fecha_final
  ORDER BY 1;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
```



```
FROM cc_balances_mensuales balmen,
     forma_pago_comprobante forpagcom
WHERE balmen.periodo_cierre = Ln_periodo_balance_ant
AND balmen.codigo_empresa = Ln_codigo_empresa
AND forpagcom.cod_comprobante = balmen.cod_comprobante
AND forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND forpagcom.empresa = balmen.codigo_empresa
AND (forpagcom.cod_forma_pago = 'CRE'
     OR forpagcom.cod_forma_pago = 'NDB'
     OR forpagcom.cod_forma_pago = 'TCR'
     OR forpagcom.cod_forma_pago = 'BAN'
     OR forpagcom.cod_forma_pago = 'ADI')
AND forpagcom.vencimiento >= Ld_fecha_inicial
AND forpagcom.vencimiento <= Ld_fecha_final
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
Ln_codigo_nacion := null;
Ln_codigo_empresa := J.codigo_empresa;
Lv_cartera := J.cartera;
--
--
*****
**
--
-- Funcion : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
-- de la tabla AD_PARAMETROS_GENERALES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa IN
NUMBER
);
IF Ln_codigo_nacion IS NULL THEN
```



```
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.  
Empresa='||to_char(J.codigo_empresa);  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Ln_codigo_nacion IS NULL THEN  
--  
-- Obtiene los Periodos de los Balances Por Empresa y Cartera  
--  
FOR I IN C_periodos_cerrados LOOP  
  Ln_periodo_balance := I.periodo_cierre;  
  Ld_fecha_inicial   := I.fecha_inicial;  
  Ld_fecha_final     := I.fecha_final;  
  --  
  -- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance  
  --  
  Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);  
  Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));  
  --  
  -- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance  
  --  
  Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
  Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);  
  --  
  -- Obtiene los creditos de un determinado periodo  
  --  
  FOR C IN C_balances LOOP  
    --  
    --  
*****  
**  
    --  
    -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y  
EMPRESA  
    --                      de la tabla AD_AGENCIAS_X_CANTONES  
    -- Autor              : Ing. Giovanni Chasifán Chicaiza.  
    -- Fecha creación     : Jueves 21 de Octubre de 2010.  
    PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia   IN      NUMBER,  
                                                    Ln_codigo_nacion,--Pn_codigo_nacion   IN      NUMBER,
```



```
J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
Lv_codigo_canton,--Pv_codigo_canton      IN OUT VARCHAR2,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
--
-- de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia  IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion     IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
```



```
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                    C.tipo_comprobante,
                                                    C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
  Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
  --                    : de la tabla STOCK.PRODUCTO
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Domingo 7 de Enero de 2007.
  PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                         Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                         Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
                                         Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
    Lv_CodigoError := null;
  ELSE-- IF Lv_CodigoError IS NOT NULL THEN
    --
    --
    ****
    **
    --
    -- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
```



```
--                                de la tabla STOCK.PRODUCTO
-- Autor                          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación                 : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento                : Obtiene EL NOMBRE DEL MODELO ITEM
  --                               de la tabla TG_MODELO_ITEM
  -- Autor                          : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación                 : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item          IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre             IN OUT VARCHAR2,
                                   Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
```



```
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante,--
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante IN VARCHAR2,
C.secuencia,--
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa,-- Pn_empresa IN NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial IN DATE,
Ld_fecha_final,--
Pd_fecha_final IN DATE
);
--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
```



```
--          de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante      IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa,-- Pn_empresa                IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final      IN      DATE
);
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--          de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante,--
Pv_cod_comprobante      IN      VARCHAR2,
Pv_tipo_comprobante      IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
IN      NUMBER,
Pd_fecha_inicial      IN      DATE,
C.Tipo_Comprobante,--
C.secuencia,--
C.Codigo_Empresa,--Pn_empresa
Ld_fecha_inicial,--
```



```

                                Ld_fecha_final--Pd_fecha_final
IN      DATE
                                );
                                --
                                --
*****
**
                                --
                                -- Función      : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
                                --                de la tabla ST_CANCELACION_D y ST_CANCELACION
                                -- Autor          : Ing. Giovanni Chasifán Chicaiza.
                                -- Fecha creación : Viernes 6 de Agosto de 2010.
                                Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
                                C.Tipo_Comprobante, --
Pv_tipo_comprobante     IN      VARCHAR2,
                                C.secuencia, --
Pn_numero_vencimiento  IN      NUMBER,
                                C.Codigo_Empresa, --
Pn_empresa              IN      NUMBER,
                                Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
                                Ld_fecha_final_mes_sig--Pd_fecha_final          IN      DATE
                                );
                                --
                                -- Actualiza el Indice de Morosidad
                                --
                                UPDATE cc_produc_mes_presente_total promespretot
                                SET      promespretot.valor_vencimiento      = promespretot.valor_vencimiento + C.valor,
                                       promespretot.saldo_vencimiento        = promespretot.saldo_vencimiento + C.saldo +
Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig,
                                       promespretot.cobro_efectivo_vencimiento = promespretot.cobro_efectivo_vencimiento +
Ln_Cobro_efectivo
                                WHERE  promespretot.periodo                    = Ln_periodo_balance
                                AND    promespretot.fecha_vencimiento        = C.vencimiento
```



```
AND promespretot.codigo_agencia = C.codigo_agencia
AND promespretot.codigo_canton = Lv_codigo_canton
AND promespretot.codigo_provincia = Lv_codigo_provincia
AND promespretot.codigo_region = Ln_codigo_region
AND promespretot.codigo_politica = C.codigo_politica
AND promespretot.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND promespretot.cod_item_cat_linea = Lv_cod_item_cat
AND promespretot.cod_marca = Ln_codigo_marca
AND promespretot.cod_cat_cliente = C.cod_cat_cliente
AND promespretot.es_empleado = C.es_empleado
AND promespretot.codigo_nacion = Ln_codigo_nacion
AND promespretot.codigo_empresa = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_produc_mes_presente_total (PERIODO
    FECHA_VENCIMIENTO
    CODIGO_AGENCIA
    CODIGO_CANTON
    CODIGO_PROVINCIA
    CODIGO_REGION
    CODIGO_POLITICA
    COD_MODELO_CAT_LINEA
    COD_ITEM_CAT_LINEA
    COD_MARCA
    COD_CAT_CLIENTE
    ES_EMPLEADO
    CODIGO_NACION
    CODIGO_EMPRESA
    valor_vencimiento
    saldo_vencimiento
    cobro_efectivo_vencimiento
    MES_CERRADO
  )
  VALUES (Ln_periodo_balance, --PERIODO
    C.vencimiento, --fecha_vencimiento
    NOT NULL NUMBER(6)
    NOT NULL DATE
```



```

C.codigo_agencia,--CODIGO_AGENCIA                                NOT NULL NUMBER(4)
Lv_codigo_canton,--CODIGO_CANTON                                NOT NULL VARCHAR2(4)
Lv_codigo_provincia,--CODIGO_PROVINCIA                          NOT NULL VARCHAR2(6)
Ln_codigo_region,--CODIGO_REGION                                NOT NULL NUMBER(3)
C.codigo_politica,--CODIGO_POLITICA                            NOT NULL NUMBER(2)
Lv_cod_modelo_cat,--COD_MODELO_CAT_LINEA                        NOT NULL VARCHAR2(8)
Lv_cod_item_cat,--COD_ITEM_CAT_LINEA                           NOT NULL VARCHAR2(3)
Ln_codigo_marca,--COD_MARCA                                    NOT NULL NUMBER(3)
C.cod_cat_cliente,--COD_CAT_CLIENTE                             NOT NULL VARCHAR2(8)
C.es empleado,--ES EMPLEADO                                     NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                                NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                              NOT NULL NUMBER(2)
C.valor,--valor_vencimiento                                     NOT NULL NUMBER(30,2)
(C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--saldo_vencimiento
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo,--cobro_efectivo                             NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                                               NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
raise_application_error(-20003,SQLERRM);
END;
```



Anexo A29.

Script carga inicial tabla cc_primeras_cuotas.

```
--  
--  
*****  
**  
--  
-- Procedimiento      : Carga las PRIMERAS CUOTAS, DONDE SECUENCIA = 1 Y LA FECHA DE VENCIMIENTO SEA DEL MES DEL  
CIERRE DEL BALANCE  
--  
--                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes  
--                   : Inserta en la Tabla CC_PRIMERAS_CUOTAS  
-- Autor              : Ing. Giovanni Chasifan Chicaiza.  
-- Fecha creacion    : Miercoles 1 de Diciembre de 2010.  
DECLARE  
--  
-- Variables Locales  
--  
Lv_CodigoError          VARCHAR2(3000);  
Ln_periodo_balance     cc_periodos_cerrados.periodo_cierre%TYPE;  
Ln_periodo_balance_ant cc_periodos_cerrados.periodo_cierre%TYPE;  
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;  
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;  
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;  
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;  
Ln_Cobro_efectivo_pri_cuo NUMBER(30,2);  
Ln_Cob_efe_pri_cuo_mes_sig NUMBER(30,2);  
Ln_descuento_mes_sig    NUMBER(30,2);  
Ln_descuento            NUMBER(30,2);  
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;  
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;  
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;  
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;  
Ln_codigo_empresa      ad_empresas.codigo_empresa%TYPE;
```



```
Lv_cod_producto          cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion     NUMBER(30,2);
Lv_cartera              cc_periodos_cerrados.cartera%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
CURSOR C_periodos_cerrados IS
  SELECT periodo_cierre, fecha_inicial, fecha_final
  FROM   cc_periodos_cerrados percer
  WHERE  codigo_empresa = Ln_codigo_empresa
  AND    carterã         = Lv_carterã
  AND    periodo_cierre > 200512
  AND    NOT EXISTS (SELECT 'X'
                    FROM   cc_primeras_cuotas pricuo
                    WHERE  pricuo.periodo          = percer.periodo_cierre
                    AND    pricuo.codigo_empresa = percer.codigo_empresa
                    )
  GROUP BY periodo_cierre, fecha_inicial, fecha_final
  ORDER BY 1;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
```



```
SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
FROM   cc_balances_mensuales balmen,
       forma_pago_comprobante forpagcom
WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
AND    balmen.codigo_empresa = Ln_codigo_empresa
AND    forpagcom.cod_comprobante = balmen.cod_comprobante
AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND    forpagcom.empresa = balmen.codigo_empresa
AND    forpagcom.secuencia = 1
AND    (forpagcom.cod_forma_pago = 'CRE'
        OR forpagcom.cod_forma_pago = 'NDB'
        OR forpagcom.cod_forma_pago = 'TCR'
        OR forpagcom.cod_forma_pago = 'BAN'
        OR forpagcom.cod_forma_pago = 'ADI')
AND    forpagcom.vencimiento >= Id_fecha_inicial
AND    forpagcom.vencimiento <= Id_fecha_final
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  Ln_codigo_empresa := J.codigo_empresa;
  Lv_cartera := J.cartera;
--
--
*****
**
--
-- Funcion : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
-- de la tabla AD PARAMETROS GENERALES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
  Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa IN
NUMBER
);
```



```
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
-- Obtiene los Periodos de los Balances Por Empresa y Cartera
--
FOR I IN C_periodos_cerrados LOOP
  Ln periodo balance := I.periodo cierre;
  Ld_fecha_inicial   := I.fecha_inicial;
  Ld_fecha_final     := I.fecha_final;
  --
  -- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance
  --
  Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);
  Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));
  --
  -- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance
  --
  Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
  Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
  --
  -- Obtiene los creditos de un determinado periodo
  --
  FOR C IN C_balances LOOP
    --
    --
    *****
    **
    --
    -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
    --
    --                      de la tabla AD_AGENCIAS_X_CANTONES
    -- Autor                : Ing. Giovanni Chasifán Chicaiza.
    -- Fecha creación      : Jueves 21 de Octubre de 2010.
    PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia   IN      NUMBER,
```



```
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa     IN      NUMBER,
Lv_codigo_canton,--Pv_codigo_canton      IN OUT VARCHAR2,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y
EMPRESA
--
-- de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia  IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion     IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa     IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
```



```
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                    C.tipo_comprobante,
                                                    C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
  Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
-- de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
```



```
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
  --                    : de la tabla TG_MODELO_ITEM
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación     : Domingo 7 de Enero de 2007.
  PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                      Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                      Lv_cod_item_cat,--Pv_cod_item         IN      VARCHAR2,
                                      Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                      Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
```



```
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_pri_cuo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa,-- Pn_empresa IN NUMBER,
Pd_fecha_final IN DATE,
C.secuencia,--
to_date('2000-01-
01','yyyy-mm-dd'),--Ld_fecha_inicial,--Pd_fecha_inicial IN DATE,
Ld_fecha_final--
);
--
```



```
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln Cob efe pri cuo mes sig := PK OBTENER ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante     IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa,-- Pn_empresa                 IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final        IN      DATE
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Viernes 6 de Agosto de 2010.
```



```
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, --
Pv_tipo_comprobante IN VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa, --Pn_empresa
IN NUMBER,
to_date('2000-01-01','yyyy-mm-
dd'), --Ld fecha inicial, --Pd fecha inicial IN DATE,
Ld_fecha_final --Pd_fecha_final
IN DATE
);
--
--
*****
**
--
-- Función : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, --
Pv_tipo_comprobante IN VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa, --
Pn_empresa IN NUMBER,
Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial IN DATE,
Ld_fecha_final_mes_sig --Pd_fecha_final IN DATE
);
--
```



```
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2, C.Tipo_Comprobante, --
Pv_tipo_comprobante IN VARCHAR2, C.secuencia, --
Pn_numero_vencimiento IN NUMBER, C.Codigo_Empresa, --
Pn_empresa IN NUMBER, to_date('2000-01-
01','yyyy-mm-dd'), --Pd_fecha_inicial IN DATE, Id_fecha_final--
Pd_fecha_final IN DATE );
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_primeras_cuotas pricuo
SET pricuo.valor_primeras_cuotas = pricuo.valor_primeras_cuotas + (C.valor -
Ln_valor_devolucion),
pricuo.saldo_primeras_cuotas = pricuo.saldo_primeras_cuotas + (C.saldo +
Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig),
pricuo.cobro_efectivo_pri_cuo = pricuo.cobro_efectivo_pri_cuo + Ln_Cobro_efectivo_pri_cuo
WHERE pricuo.periodo = Ln_periodo_balance
```



```
AND pricuo.fecha_vencimiento_pri_cuo = C.vencimiento
AND pricuo.codigo_agencia = C.codigo_agencia
AND pricuo.codigo_canton = Lv_codigo_canton
AND pricuo.codigo_provincia = Lv_codigo_provincia
AND pricuo.codigo_region = Ln_codigo_region
AND pricuo.codigo_politica = C.codigo_politica
AND pricuo.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND pricuo.cod_item_cat_linea = Lv_cod_item_cat
AND pricuo.cod_marca = Ln_codigo_marca
AND pricuo.cod_cat_cliente = C.cod_cat_cliente
AND pricuo.es_empleado = C.es_empleado
AND pricuo.codigo_nacion = Ln_codigo_nacion
AND pricuo.codigo_empresa = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_primeras_cuotas (PERIODO
                                FECHA_VENCIMIENTO_PRI_CUO
                                CODIGO_AGENCIA
                                CODIGO_CANTON
                                CODIGO_PROVINCIA
                                CODIGO_REGION
                                CODIGO_POLITICA
                                COD_MODELO_CAT_LINEA
                                COD_ITEM_CAT_LINEA
                                COD_MARCA
                                COD_CAT_CLIENTE
                                ES_EMPLEADO
                                CODIGO_NACION
                                CODIGO_EMPRESA
                                VALOR_PRIMERAS_CUOTAS
                                SALDO_PRIMERAS_CUOTAS
                                COBRO_EFECTIVO_PRI_CUO
                                MES_CERRADO
                                )
  VALUES (Ln_periodo_balance, --PERIODO NOT NULL NUMBER(6)
```



```
C.vencimiento,--FECHA_VENCIMIENTO_PRI_CUO                NOT NULL DATE
C.codigo_agencia,--CODIGO_AGENCIA                        NOT NULL NUMBER(4)
Lv_codigo_canton,--CODIGO_CANTON                        NOT NULL VARCHAR2(4)
Lv_codigo_provincia,--CODIGO_PROVINCIA                  NOT NULL VARCHAR2(6)
Ln_codigo_region,--CODIGO_REGION                        NOT NULL NUMBER(3)
C.codigo_politica,--CODIGO_POLITICA                     NOT NULL NUMBER(2)
Lv_cod_modelo_cat,--COD_MODELO_CAT_LINEA                NOT NULL VARCHAR2(8)
Lv_cod_item_cat,--COD_ITEM_CAT_LINEA                   NOT NULL VARCHAR2(3)
Ln_codigo_marca,--COD_MARCA                             NOT NULL NUMBER(3)
C.cod_cat_cliente,--COD_CAT_CLIENTE                     NOT NULL VARCHAR2(8)
C.es_empleado,--ES_EMPLEADO                             NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                       NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                       NOT NULL NUMBER(2)
(C.valor - Ln_valor_devolucion),--C.valor,--VALOR_PRIMERAS_CUOTAS
NOT NULL NUMBER(30,2)
(C.saldo + Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--SALDO_PRIMERAS_CUOTAS
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo_pri_cuo,--COBRO_EFECTIVO_PRI_CUO     NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                                       NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR I IN C_periodos_cerrados LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
raise_application_error(-20003,SQLERRM);
END;
```



Anexo A30.

Script actualización de tablas de dimensión.

```
CREATE OR REPLACE PACKAGE BODY PK_DM_PROCESOS_TB_DIMENSIONES IS
--
--
*****
**
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS TABLAS
DE DIMENSIONES
--                      DE UN PERIODO DE BALANCE CERRADO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Miercoles 4 de Julio de 2010.
PROCEDURE P_INSERTAR_REGISTROS_NUEVOS
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Lv_mensaje              VARCHAR2(3000);
--
-- Obtiene los periodos cerrados de los balances
--
CURSOR C_periodos_balances IS
SELECT periodo_cierre, codigo_empresa
FROM   cc_balances_mensuales balmen
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_periodos_balances dmperbal
                   WHERE  dmperbal.periodo = balmen.periodo_cierre
                   AND    dmperbal.codigo_empresa = balmen.codigo_empresa
                   )
GROUP BY codigo_empresa, periodo_cierre
ORDER BY codigo_empresa, periodo_cierre;
--
```



```
-- Obtiene los Tipos de Cartera para insertar el periodo corriente, a pesar que no esta cerrado el balance
--
CURSOR C_tipos_carteras IS
SELECT *
FROM cc_tipos_carteras tipcar
WHERE tipcar.estado = 'N'
AND tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene las fechas de las ventas
--
CURSOR C_comprobante IS
SELECT to_number(to_char(com.fecha,'yyyymm')) anio_mes_venta, com.empresa codigo_empresa
FROM comprobante com
WHERE com.tipo_comprobante in ('A0','NV','D0','DB')
AND NOT EXISTS (SELECT 'X'
FROM dm_anio_mes_ventas dmanimesven
WHERE dmanimesven.anio_mes_venta = to_number(to_char(com.fecha,'yyyymm'))
AND dmanimesven.codigo_empresa = com.empresa
)
GROUP BY to_number(to_char(com.fecha,'yyyymm')), com.empresa
ORDER BY to_number(to_char(com.fecha,'yyyymm')), com.empresa;
--
-- Obtiene el plazo de las ventas
--
CURSOR C_comprobante_x_plazo IS
SELECT com.numero_pagos, com.empresa codigo_empresa
FROM comprobante com
WHERE com.tipo_comprobante in ('A0','NV','D0','DB')
AND NOT EXISTS (SELECT 'X'
FROM dm_plazos_ventas dmplaven
WHERE dmplaven.plazo = com.numero_pagos
AND dmplaven.codigo_empresa = com.empresa
)
GROUP BY com.numero_pagos, com.empresa
ORDER BY com.numero_pagos, com.empresa;
--
```



```
-- Obtiene las Lineas de todas las empresas
--
CURSOR C_tg_modelo_item IS
  SELECT *
  FROM   tg_modelo_item modite
  WHERE  NOT EXISTS (SELECT 'X'
                    FROM   dm_lineas dmlin
                    WHERE  dmlin.empresa      = modite.empresa
                    AND    dmlin.cod_modelo_linea = modite.cod_modelo
                    AND    dmlin.cod_item_linea  = modite.cod_item
                    );
--
-- Obtiene las Marcas de todas las empresas
--
CURSOR C_marca IS
  SELECT *
  FROM   marca mar
  WHERE  NOT EXISTS (SELECT 'X'
                    FROM   dm_marcas dmmar
                    WHERE  dmmar.cod_marca = mar.cod_marca
                    AND    dmmar.empresa  = mar.empresa
                    );
--
-- Obtiene los Productos de todas las empresas
--
CURSOR C_productos IS
  SELECT *
  FROM   producto pro
  WHERE  NOT EXISTS (SELECT 'X'
                    FROM   dm_productos dmpro
                    WHERE  dmpro.cod_producto = pro.cod_producto
                    AND    dmpro.empresa      = pro.empresa
                    );
--
-- Obtiene las Formas de Ventas de todas las empresas
--
CURSOR C_forma_pago IS
```



```
SELECT *
FROM   forma_pago forpag
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_forma_pago dmforpag
                   WHERE  dmforpag.cod_forma_pago = forpag.cod_forma_pago
                   AND    dmforpag.empresa      = forpag.empresa
                   );

--
-- Obtiene los Tipos de Clientes de todas las empresas
--
CURSOR C_tg_modelo_item_tip_cli IS
SELECT *
FROM   tg_modelo_item modite
WHERE  modite.cod_modelo = 'CLI1'
AND    NOT EXISTS (SELECT 'X'
                   FROM   dm_tipo_cliente dmtipcli
                   WHERE  dmtipcli.cod_item_tipo_cliente = modite.cod_item
                   AND    dmtipcli.cod_modelo_item_tipo_cliente = modite.cod_modelo
                   AND    dmtipcli.empresa = modite.empresa
                   );

--
-- Obtiene las Politicas de todas las empresas
--
CURSOR C_politica_credito IS
SELECT *
FROM   st_politica_credito polcre
WHERE  NOT EXISTS (SELECT 'X'
                   FROM   dm_politicas dmpol
                   WHERE  dmpol.cod_politica = polcre.cod_politica
                   AND    dmpol.empresa = polcre.empresa
                   );

--
-- Obtiene el codigo de las calificaciones de los clientes de todas las empresas
--
CURSOR C_st_cat_cliente IS
SELECT *
FROM   st_cat_cliente catcli
```



```
WHERE NOT EXISTS (SELECT 'X'
                   FROM   dm_calificacion_del_cliente dmcalcli
                   WHERE   dmcalcli.cod_cat_cliente = catcli.cod_cat_cliente
                   AND     dmcalcli.empresa      = catcli.empresa
                   );
--
-- Obtiene las empresas donde trabajan los empleados de todas las empresas
--
CURSOR C_st_empleados IS
SELECT stemp.nombre empresa
FROM   st_empleados stemp
WHERE  NOT EXISTS (SELECT 'X'
                  FROM   dm_es_empleado dmesemp
                  WHERE   dmesemp.es_empleado = stemp.nombre_empresa
                  )
GROUP BY stemp.nombre_empresa;
--
-- Obtiene las Fechas de Vencimientos de todas las cuotas de todas las empresas
--
CURSOR C_fechas_vencimiento IS
SELECT trunc(vencimiento) vencimiento
FROM   forma_pago_comprobante forpagcom
WHERE  forpagcom.vencimiento > to_date('2006-01-01','yyyy-mm-dd')
AND    NOT EXISTS (SELECT 'X'
                  FROM   dm_fechas_vencimientos dmfecven
                  WHERE   dmfecven.fecha_vencimiento = trunc(forpagcom.vencimiento)
                  )
GROUP BY trunc(vencimiento)
ORDER BY trunc(vencimiento);
--
-- Obtiene las fechas de las ventas en Anio_Mes Venta y Dia de Venta
--
CURSOR C_comprobante_aaaamm_dia IS
SELECT to_number(to_char(com.fecha,'yyyymm')) anio_mes_venta, to_number(to_char(com.fecha,'dd')) dia_venta,
com.empresa codigo_empresa
FROM   comprobante com
WHERE  com.tipo_comprobante in ('A0','NV','D0','DB')
```



```
AND NOT EXISTS (SELECT 'X'
                 FROM   dm_fechas_ventas dmfecven
                 WHERE  dmfecven.anio_mes_venta = to_number(to_char(com.fecha, 'yyyymm'))
                 AND    dmfecven.dia_venta      = to_number(to_char(com.fecha, 'dd'))
                 AND    dmfecven.codigo_empresa = com.empresa
                 )
GROUP BY to_number(to_char(com.fecha, 'yyyymm')), to_number(to_char(com.fecha, 'dd')), com.empresa
ORDER BY to_number(to_char(com.fecha, 'yyyymm')), to_number(to_char(com.fecha, 'dd')), com.empresa;
BEGIN
Lv_mensaje := NULL;
--
-- Obtiene los periodos cerrados de los balances
--
FOR C IN C_periodos_balances LOOP
Lv_mensaje := 'Tabla DM_PERIODOS_BALANCES. Periodo='||C.periodo_cierre||' Empr='||C.codigo_empresa;
INSERT INTO dm_periodos_balances (PERIODO
                                CODIGO_EMPRESA
                                )
VALUES (C.periodo_cierre,
        C.codigo_empresa
        );
END LOOP;-- FOR C IN C_periodos_balances LOOP
--
--
commit;
Lv_mensaje := NULL;
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
BEGIN
Lv_mensaje := 'Tabla DM_PERIODOS_BALANCES. Periodo='||to_number(to_char(sysdate, 'yyyymm'))||'
Empr='||J.codigo_empresa;
INSERT INTO dm_periodos_balances (PERIODO
                                CODIGO_EMPRESA
                                )
```



```
VALUES (to_number(to_char(sysdate,'yyyymm')),
        J.codigo_empresa
        );
--
--
--
commit;
EXCEPTION
  WHEN others THEN
    Lv_CodigoError := null;
END;-- BEGIN
END LOOP;-- FOR J IN C_tipos_carteras LOOP
Lv_mensaje := NULL;
--
-- Obtiene las fechas de las ventas
--
FOR C IN C_comprobante LOOP
  Lv_mensaje := 'Tabla DM_ANIO_MES_VENTAS. AnioMesVentao='||C.anio_mes_venta||' Empr='||C.codigo_empresa;
  INSERT INTO dm_anio_mes_ventas (ANIO_MES_VENTA
                                CODIGO_EMPRESA
                                )
    VALUES (C.anio_mes_venta,
            C.codigo_empresa
            );
END LOOP;-- FOR C IN C_comprobante LOOP
--
--
--
commit;
Lv_mensaje := NULL;
--
-- Obtiene el plazo de las ventas
--
FOR C IN C_comprobante_x_plazo LOOP
  Lv_mensaje := 'Tabla DM_PLAZOS_VENTAS. Plazo='||C.numero_pagos||' Empr='||C.codigo_empresa;
  INSERT INTO dm_plazos_ventas (PLAZO
                                CODIGO_EMPRESA
                                )
```



```

                                )
VALUES (C.numero_pagos,
        C.codigo_empresa
        );
END LOOP;-- FOR C IN C_comprobante_x_plazo LOOP
--
--
--
commit;
Lv mensaje := NULL;
--
-- Obtiene las Lineas de todas las empresas
--
FOR C IN C_tg_modelo_item LOOP
Lv_mensaje := 'Tabla DM_LINEA. Empr='||C.empresa||' Modelo_Linea='||C.cod_modelo||'
Item_Linea='||C.cod_item;
INSERT INTO dm_lineas (EMPRESA
                      COD_MODELO_LINEA
                      COD_ITEM_LINEA
                      NOMBRE
                      )
VALUES (C.empresa,
        C.cod_modelo,
        C.cod_item,
        C.nombre
        );
END LOOP;-- FOR C IN C_tg_modelo_item LOOP
--
--
--
commit;
--
-- Obtiene las Marcas de todas las empresas
--
FOR C IN C_marca LOOP
Lv_mensaje := 'Tabla DM_MARCAS. Marca='||C.cod_marca||' Empr='||C.empresa;
INSERT INTO dm_marcas (COD_MARCA
                      )
```



```

                EMPRESA
                NOMBRE
            )
VALUES (C.cod_marca,
        C.empresa,
        C.nombre
    );
END LOOP; -- FOR C IN C_marca LOOP
--
--
--
commit;
--
-- Obtiene los Productos de todas las empresas
--
FOR C IN C_productos LOOP
    INSERT INTO dm_productos (COD_PRODUCTO
                            EMPRESA
                            NOMBRE
                            COD_MARCA
                            )
VALUES (C.cod_producto,
        C.empresa,
        C.nombre,
        C.cod_marca
    );
END LOOP; -- FOR C IN C_productos LOOP
--
--
--
commit;
--
-- Obtiene las Formas de Ventas de todas las empresas
--
FOR C IN C_forma_pago LOOP
    INSERT INTO dm_forma_pago (COD_FORMA_PAGO,
                              EMPRESA,

```




```
        )
VALUES (C.cod_politica,
        C.empresa,
        C.nombre
        );
END LOOP;-- FOR C IN C_politica_credito LOOP
--
--
--
commit;
--
-- Obtiene el codigo de las calificaciones de los clientes de todas las empresas
--
FOR C IN C_st_cat_cliente LOOP
    INSERT INTO dm_calificacion_del_cliente (COD_CAT_CLIENTE,
                                            EMPRESA,
                                            NOMBRE
                                            )

VALUES (C.cod_cat_cliente,
        C.empresa,
        C.nombre
        );
END LOOP;-- FOR C IN C_st_cat_cliente LOOP
--
--
--
commit;
--
-- Obtiene las empresas donde trabajan los empleados de todas las empresas
--
FOR C IN C_st_empleados LOOP
    INSERT INTO dm_es_empleado (ES_EMPLEADO
                                )

VALUES (C.nombre_empresa
        );
END LOOP;-- FOR C IN C_st_empleados LOOP
--
```



```
--  
--  
commit;  
--  
-- Obtiene las Fechas de Vencimientos de todas las cuotas de todas las empresas  
--  
FOR C IN C_fechas_vencimiento LOOP  
    INSERT INTO dm_fechas_vencimientos (FECHA_VENCIMIENTO  
                                       )  
        VALUES (C.vencimiento  
                );  
END LOOP; -- FOR C IN C_fechas_vencimiento LOOP  
--  
--  
commit;  
Lv_mensaje := NULL;  
--  
-- Obtiene las fechas de las ventas en Anio_Mes Venta y Dia de Venta  
--  
FOR C IN C_comprobante_aaaamm_dia LOOP  
    Lv_mensaje := 'Tabla DM_FECHAS_VENTAS. AnioMesVenta='||C.anio_mes_venta||' DiaVta='||C.dia_venta||'  
Empr='||C.codigo_empresa;  
    INSERT INTO dm_fechas_ventas (ANIO_MES_VENTA  
                                  DIA_VENTA,  
                                  CODIGO_EMPRESA  
                                  )  
        VALUES (C.anio_mes_venta,  
                C.dia_venta,  
                C.codigo_empresa  
                );  
END LOOP; -- FOR C IN C_comprobante_aaaamm_dia LOOP  
--  
--  
--  
commit;  
EXCEPTION
```



```
WHEN others THEN
  --
  -- Envia correo de Error a DBA
  --
  Lv_CodigoError := null;
  P_ENVIAR_MAILS_DBA (null, --Pv_directorio      IN      VARCHAR2,
                    null, --Pv_archivo         IN      VARCHAR2,
                    2, --Pn_codigo_empresa    IN      NUMBER,
                    'Error. No pudo cargar las tablas de Dimensionamiento.', --Pv_asunto      IN
  VARCHAR2,
                    'Error. No pudo cargar las tablas de Dimensionamiento. ' || chr(10) || chr(10) ||
                    'Paquete Almacenado:
  PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS NUEVOS' || chr(10) || chr(10) ||
                    Lv_mensaje || chr(10) || chr(10) ||
                    'Error: ' || SQLERRM, --Pv_mensaje      IN      VARCHAR2,
                    Lv_CodigoError --Pv_CodigoError      IN OUT VARCHAR2
                    );
  raise_application_error (-20003, SQLERRM);
END P_INSERTAR_REGISTROS NUEVOS;
END;
```



Anexo A31.

Script actualización índice de cumplimiento de ventas.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_IND_CUPLIM_VTA IS
--
--
*****
**
--
-- Procedimiento      : Carga el Indice de Cumplimiento de Ventas
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla DM_INDICE_CUPLIMIENTO_VENTAS
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Viernes 31 de Julio de 2010.
PROCEDURE P_CARGAR_IND_CUMVTA_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA DEL INDICE DE CUMPLIMIENTO DE VENTAS
--                    : Inserta en la Tabla DM_INDICE_CUPLIMIENTO_VENTAS
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Viernes 31 de Julio de 2010.
PROCEDURE P_ACTUALIZACION_DIA_IND_CUMVTA (Pv_CodigoError  IN OUT VARCHAR2);
END;
/
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_IND_CUPLIM_VTA IS
--
--
*****
**
--
-- Procedimiento      : Carga el Indice de Cumplimiento de Ventas
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
```



```
--          Inserta en la Tabla DM INDICE CUMPLIMIENTO_VENTAS
-- Autor          : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion  : Viernes 31 de Julio de 2010.
PROCEDURE P_CARGAR_IND_CUMVTA_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Ld_fecha_inicial          DATE;
Ld_fecha_final            DATE;
Ln_periodo_balance        cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_codigo_nacion          ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton          ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region          ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia       ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa        ad_empresas.codigo_empresa%TYPE;
Ln_contador               NUMBER;
Ln_anio_inicial           NUMBER(4);
Ln_anio_final             NUMBER(4);
Ln_anio                   NUMBER(4);
Ln_mes                    NUMBER(2);
Ln_dia                    NUMBER(2);
Ln_dia_final              NUMBER(2);
Lv_CodigoError            VARCHAR2(3000);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM   cc_tipos_carteras tipcar
WHERE  tipcar.estado = 'N'
AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel que
utilizaba Franklin Bernal
```



```
--  
CURSOR C_ventas_detalladas IS  
  SELECT /*+ first_rows */  
    e.cod_agencia,  
    e.nombre Agencia,  
    to_char(a.fecha,'yyyymm') Fecha,  
    to_char(a.fecha,'yyyy-mm-dd') Fecha_dia,  
    to_char(a.fecha,'dd') dia_venta,  
    d.nombre Linea,  
    g.nombre Marca,  
    f.nombre Vendedor,  
    a.numero_pagos,  
    a.forma_pago,  
    k.nombre Tipo_cliente,  
    j.cod_cat_cliente Categoria_cliente,  
    decode(a.tipo_comprobante,'D0',(b.total_linea-b.iva) * -1,(b.total_linea-b.iva))Valor  
  FROM  
    (  
      SELECT ven.cod_comprobante, ven.empresa, ven.tipo_comprobante,  
ven.fecha,ven.numero_pagos,ven.forma_pago,ven.cod_agencia,  
          ven.cod_agente,ven.cod_persona,secuen_certificado,observaciones  
        FROM stock.comprobante VEN  
        WHERE ven.tipo_comprobante in ('A0','NV')  
        AND ven.anulado='N'  
        AND ven.fecha between to_date(to_char(ld_fecha_inicial,'yyyy-mm-dd')||' 00:00:00','yyyy-mm-dd hh24:mi:ss')  
AND to_date(to_char(ld_fecha_final,'yyyy-mm-dd')||' 23:59:59','yyyy-mm-dd hh24:mi:ss')  
        AND ven.empresa=ln_codigo_empresa  
      UNION ALL  
      SELECT dev.cod_comprobante,dev.empresa, dev.tipo_comprobante,  
dev.fecha,dev.numero_pagos,dev.forma_pago,ven_dev.cod_agencia,  
          dev.cod_agente,dev.cod_persona,dev.secuen_certificado,dev.observaciones  
        from  
        STOCK.COMPROBANTE DEV,  
        STOCK.ST_COMPROBANTE_COMPROBANTE X,  
        STOCK.COMPROBANTE VEN_DEV,  
        STA_PARAMETROS_VISTA_ELEC par  
        WHERE dev.cod_comprobante=x.cod_comprobante_s
```



```
AND dev.tipo_comprobante=x.tipo_comprobante_s
AND dev.empresa=x.empresa_s
AND ven_dev.cod_comprobante=x.cod_comprobante
AND ven_dev.tipo_comprobante=x.tipo_comprobante
AND ven_dev.empresa=x.empresa
AND dev.tipo_comprobante in ('D0')
AND dev.anulado='N'
AND dev.empresa=Ln_codigo_empresa
AND dev.fecha between to_date(to_char(Ld_fecha_inicial,'yyyy-mm-dd')||' 00:00:00','yyyy-mm-dd hh24:mi:ss')
AND to date(to_char(Ld_fecha_final,'yyyy-mm-dd')||' 23:59:59','yyyy-mm-dd hh24:mi:ss')
)a
,stock.movimiento b
,stock.producto c
,computo.tg_modelo_item d
,computo.tg_agencia e
,stock.persona f
,stock.marca g
,stock.cliente h
,stock.cliente_hor i
,stock.st_cliente_cupo j
,computo.tg_modelo_item k
,stock.st_comprobante_comprobante l
,stock.comprobante m --devolucion
,stock.forma_pago_comprobante n
,stock.st_tipo_devolucion_anticipo o
,stock.st_credijaher p
,stock.st_credijaher_origen q
WHERE b.cod_comprobante=a.cod_comprobante
AND b.tipo_comprobante=a.tipo_comprobante
AND b.empresa=a.empresa
AND c.cod_producto=b.cod_producto
AND c.empresa=b.empresa
AND d.cod_modelo(+) =c.cod_modelo_cat
AND d.cod_item(+) =c.cod_item_cat
AND d.empresa(+) =c.empresa
AND e.cod_agencia=a.cod_agencia
AND e.empresa=a.empresa
```



```
AND f.cod_persona=a.cod_agente
AND f.empresa=a.empresa
AND f.cod_tipo_persona='VEN'
AND g.cod_marca=c.cod_marca
AND g.empresa=c.empresa
AND h.cod_cliente=a.cod_persona
AND h.empresa=a.empresa
AND i.cod_clienteh=h.cod_cliente
AND i.empresa=h.empresa
AND j.empresa(+) =h.empresa
AND j.cod_cliente(+) =h.cod_cliente
AND k.empresa=i.empresa
AND k.cod_modelo = 'CL11'
AND k.cod_item = i.cod_tipo_clienteh
--Devoluciones
AND l.empresa(+) =a.empresa
AND l.tipo_comprobante_s(+) =a.tipo_comprobante
AND l.cod_comprobante_s(+) =a.cod_comprobante
AND m.empresa(+) =l.empresa
AND m.tipo_comprobante(+) =l.tipo_comprobante
AND m.cod_comprobante(+) =l.cod_comprobante
AND a.empresa=n.empresa(+)
AND a.tipo_comprobante=n.tipo_comprobante(+)
AND a.cod_comprobante=n.cod_comprobante(+)
AND n.cod_forma_pago(+) = 'ADI'
AND o.empresa(+) =a.empresa
AND o.cod_tipo_devolucion(+) =a.secuen_certificado
AND p.cod_cliente(+) =a.cod_persona
AND q.cod_origen(+) =p.cod_origen;
BEGIN
--
-- Inserta Tablas Maestras
--
--
--
*****
**
```



```
--
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS
TABLAS DE DIMENSIONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Miercoles 4 de Julio de 2010.
PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS_NUEVOS;
--
-- Inserta las Fechas de Ventas de Todas las empresas
--
--
-- Obtiene los Tipos de Cartera para Insertar las Fechas de Ventas de todas las empresas
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    : de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Obtiene el Año Inicial y Año Final
--
Ln_anio_inicial := to_number(to_char(Ld_fecha_inicial,'yyyy'));
Ln_anio_final   := to_number(to_char(Ld_fecha_final,'yyyy'));
```



```
--  
-- Lee el rango de años  
--  
FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP  
  --  
  -- Lee los meses del año  
  --  
  FOR Ln_mes IN 1..12 LOOP  
    Ln_dia_final := to_number(to_char(last_day(to_date(Ln_anio||lpad(Ln_mes,2,'0')||'01','yyyy-mm-  
dd')), 'dd'));  
    FOR Ln_dia IN 1..Ln_dia_final LOOP  
      --  
      -- Inserta los dias  
      --  
      BEGIN  
        INSERT INTO dm_fechas_ventas (ANIO_MES_VENTA  
          DIA_VENTA  
          CODIGO_EMPRESA  
        )  
          VALUES (to_number(Ln_anio||lpad(Ln_mes,2,'0')), --ANIO_MES_VENTA  
            NUMBER(6) NOT NULL  
            Ln_dia, --DIA_VENTA NOT NULL NUMBER(2)  
            J.codigo_empresa--CODIGO_EMPRESA NOT NULL NUMBER(2)  
          );  
      EXCEPTION  
        WHEN others THEN  
          NULL;  
      END;  
    END LOOP; -- FOR Ln_dia IN 1..Ln_dia_final LOOP  
  END LOOP; -- FOR Ln_mes IN 1..12 LOOP  
END LOOP; -- FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP  
END LOOP; -- FOR J IN C_tipos_carteras LOOP  
--  
-- ***** Borrar los Registros del Periodo Abierto  
--  
--  
--  
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
```



```
--
FOR J IN C_tipos_carteras LOOP
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--           Como datos definitos del cierre de mes
--
--
DELETE dm_indice_cumplimiento_ventas indcumven
WHERE indcumven.mes_cerrado = 'A'--A=MES ABIERTO, C=MES CERRADO
AND indcumven.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- ***** Inserta las Ventas del Periodo Cerrado
--
--
--
Ln_contador := 0;
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Funcion : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--           de la tabla AD PARAMETROS GENERALES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa IN
NUMBER
);
```



```
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel
que utilizaba Franklin Bernal
--
FOR C IN C_ventas_detalladas LOOP
  --
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                Lv_codigo_canton,--Pv_codigo_canton    IN OUT VARCHAR2,
                                                Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                                Lv_CodigoError
                                                );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
```



```
-- Actualiza las ventas netas del Cumplimiento de ventas
--
UPDATE dm_indice_cumplimiento_ventas indcumven
SET    indcumven.valor_venta_neta = indcumven.valor_venta_neta + C.valor
WHERE  indcumven.anio_mes_venta   = to_number(C.fecha)
AND    indcumven.dia_venta        = to_number(C.dia_venta)
AND    indcumven.codigo_agencia   = C.cod_agencia
AND    indcumven.codigo_canton    = Lv_codigo_canton
AND    indcumven.codigo_provincia = Lv_codigo_provincia
AND    indcumven.codigo_region    = Ln_codigo_region
AND    indcumven.codigo_nacion    = Ln_codigo_nacion
AND    indcumven.codigo_empresa  = J.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
    INSERT INTO dm_indice_cumplimiento_ventas (ANIO_MES_VENTA
                                                DIA_VENTA
                                                CODIGO_AGENCIA
                                                CODIGO_EMPRESA
                                                VALOR_VENTA_NETA
                                                MES_CERRADO
                                                )
        VALUES (to_number(C.fecha), --PERIODO                NOT NULL NUMBER(6)
                to_number(C.dia_venta), --DIA_VENTA          NOT NULL NUMBER(2)
                C.cod_agencia, --CODIGO_AGENCIA              NOT NULL NUMBER(4)
                J.codigo_empresa, --CODIGO_EMPRESA          NOT NULL NUMBER(2)
                C.valor, --VALOR_VENTA_NETA                 NOT NULL NUMBER(30,2)
                'C' --MES_CERRADO                          NOT NULL VARCHAR2(1)
                );
END IF; -- IF Sql%Notfound THEN
--
--
--
Ln_contador := Ln_contador + 1;
IF Ln_contador > 2000 THEN
    commit;
```



```
Ln_contador := 0;
END IF;-- IF Ln_contador > 2000 THEN
END LOOP;-- FOR C IN C_ventas_detalladas LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
commit;
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := Pv_CodigoError||'. '||SQLERRM;
END P_CARGAR_IND_CUMVTA_CIERRE_MES;
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA DEL INDICE DE CUMPLIMIENTO DE VENTAS
--                    : Inserta en la Tabla DM_INDICE_CUMPLIMIENTO_VENTAS
-- Autor              : Ing. Giovanny Chasifan Chicaiza.
-- Fecha creacion    : Viernes 31 de Julio de 2010.
PROCEDURE P_ACTUALIZACION_DIA_IND_CUMVTA (Pv_CodigoError IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Ld_fecha_inicial          DATE;
Ld_fecha_final            DATE;
Ld_fecha_inicial_mes_sig  DATE;
Ld_fecha_final_mes_sig    DATE;
Ln_periodo_balance        cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_mes_sig        cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_codigo_nacion          ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton          ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region          ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia       ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa        ad_empresas.codigo_empresa%TYPE;
```



```
Ln_contador          NUMBER;
Ln_anio_inicial      NUMBER(4);
Ln_anio_final        NUMBER(4);
Ln_anio              NUMBER(4);
Ln_mes              NUMBER(2);
Ln_dia              NUMBER(2);
Ln_dia_final         NUMBER(2);
Lv_CodigoError       VARCHAR2(3000);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel que
utilizaba Franklin Bernal
--
CURSOR C_ventas_detalladas IS
  SELECT /*+ first_rows */
    e.cod_agencia,
    e.nombre Agencia,
    to_char(a.fecha,'yyyymm') Fecha,
    to_char(a.fecha,'yyyy-mm-dd') Fecha_dia,
    to_char(a.fecha,'dd') dia_venta,
    d.nombre Linea,
    g.nombre Marca,
    f.nombre Vendedor,
    a.numero_pagos,
    a.forma_pago,
    k.nombre Tipo_cliente,
    j.cod_cat_cliente Categoria_cliente,
    decode(a.tipo_comprobante,'D0',(b.total_linea-b.iva) * -1,(b.total_linea-b.iva))Valor
```



```
FROM
(
SELECT ven.cod_comprobante, ven.empresa, ven.tipo_comprobante,
ven.fecha, ven.numero_pagos, ven.forma_pago, ven.cod_agencia,
ven.cod_agente, ven.cod_persona, secuen_certificado, observaciones
FROM stock.comprobante VEN
WHERE ven.tipo_comprobante in ('A0', 'NV')
AND ven.anulado='N'
AND ven.fecha between to_date(to_char(Ld_fecha_inicial, 'yyyy-mm-dd') || ' 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
AND to_date(to_char(Ld_fecha_final, 'yyyy-mm-dd') || ' 23:59:59', 'yyyy-mm-dd hh24:mi:ss')
AND ven.empresa=Ln_codigo_empresa
UNION ALL
SELECT dev.cod_comprobante, dev.empresa, dev.tipo_comprobante,
dev.fecha, dev.numero_pagos, dev.forma_pago, ven_dev.cod_agencia,
dev.cod_agente, dev.cod_persona, dev.secuen_certificado, dev.observaciones
from
STOCK.COMPROBANTE DEV,
STOCK.ST_COMPROBANTE_COMPROBANTE X,
STOCK.COMPROBANTE VEN_DEV,
STA_PARAMETROS_VISTA_ELEC par
WHERE dev.cod_comprobante=x.cod_comprobante_s
AND dev.tipo_comprobante=x.tipo_comprobante_s
AND dev.empresa=x.empresa_s
AND ven_dev.cod_comprobante=x.cod_comprobante
AND ven_dev.tipo_comprobante=x.tipo_comprobante
AND ven_dev.empresa=x.empresa
AND dev.tipo_comprobante in ('D0')
AND dev.anulado='N'
AND dev.empresa=Ln_codigo_empresa
AND dev.fecha between to_date(to_char(Ld_fecha_inicial, 'yyyy-mm-dd') || ' 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
AND to_date(to_char(Ld_fecha_final, 'yyyy-mm-dd') || ' 23:59:59', 'yyyy-mm-dd hh24:mi:ss')
)a
, stock.movimiento b
, stock.producto c
, computo.tg_modelo_item d
, computo.tg_agencia e
, stock.persona f
```



```
,stock.marca g
,stock.cliente h
,stock.cliente_hor i
,stock.st_cliente_cupo j
,computo.tg_modelo_item k
,stock.st_comprobante_comprobante l
,stock.comprobante m --devolucion
,stock.forma_pago_comprobante n
,stock.st_tipo_devolucion_anticipo o
,stock.st_credijaher p
,stock.st_credijaher_origen q
WHERE b.cod_comprobante=a.cod_comprobante
AND b.tipo_comprobante=a.tipo_comprobante
AND b.empresa=a.empresa
AND c.cod_producto=b.cod_producto
AND c.empresa=b.empresa
AND d.cod_modelo(+)=c.cod_modelo_cat
AND d.cod_item(+)=c.cod_item_cat
AND d.empresa(+)=c.empresa
AND e.cod_agencia=a.cod_agencia
AND e.empresa=a.empresa
AND f.cod_persona=a.cod_agente
AND f.empresa=a.empresa
AND f.cod_tipo_persona='VEN'
AND g.cod_marca=c.cod_marca
AND g.empresa=c.empresa
AND h.cod_cliente=a.cod_persona
AND h.empresa=a.empresa
AND i.cod_clienteh=h.cod_cliente
AND i.empresah=h.empresa
AND j.empresa(+)=h.empresa
AND j.cod_cliente(+)=h.cod_cliente
AND k.empresa=i.empresah
AND k.cod_modelo = 'CLI1'
AND k.cod_item = i.cod_tipo_clienteh
--Devoluciones
AND l.empresa(+)=a.empresa
```



```
AND l.tipo_comprobante_s(+)=a.tipo_comprobante
AND l.cod_comprobante_s(+)=a.cod_comprobante
AND m.empresa(+)=l.empresa
AND m.tipo_comprobante(+)=l.tipo_comprobante
AND m.cod_comprobante(+)=l.cod_comprobante
AND a.empresa=n.empresa(+)
AND a.tipo_comprobante=n.tipo_comprobante(+)
AND a.cod_comprobante=n.cod_comprobante(+)
AND n.cod_forma_pago(+)'ADI'
AND o.empresa(+)=a.empresa
AND o.cod_tipo_devolucion(+)=a.secuen_certificado
AND p.cod_cliente(+)=a.cod_persona
AND q.cod_origen(+)=p.cod_origen;
BEGIN
--
-- Inserta Tablas Maestras
--
--
--
--
*****
**
--
-- Procedimiento : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS
TABLAS DE DIMENSIONES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Miercoles 4 de Julio de 2010.
PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS_NUEVOS;
--
-- Inserta las Fechas de Ventas de Todas las empresas
--
--
-- Obtiene los Tipos de Cartera para Insertar las Fechas de Ventas de todas las empresas
--
FOR J IN C_tipos_carteras LOOP
--
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    : de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK OBTENER CC PER CER.P ULT CIERRE X CART EMPR ROWNU (J.codigo empresa,--Pn codigo empresa  IN      NUMBER,
                                                       J.cartera,--Pv_cartera          IN      VARCHAR2,
                                                       Ln_periodo_balance,--Pn periodo_balance  IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
                                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Obtiene el Año Inicial y Año Final
--
Ln_anio_inicial := to_number(to_char(Ld_fecha_inicial,'yyyy'));
Ln_anio_final   := to_number(to_char(Ld_fecha_final,'yyyy'));
--
-- Lee el rango de años
--
FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP
    --
    -- Lee los meses del año
    --
    FOR Ln_mes IN 1..12 LOOP
        Ln_dia_final := to_number(to_char(last_day(to_date(Ln_anio||lpad(Ln_mes,2,'0')||'01','yyyy-mm-
dd')), 'dd'));
        FOR Ln_dia IN 1..Ln_dia_final LOOP
            --
            -- Inserta los dias
            --

```



```
BEGIN
  INSERT INTO dm_fechas_ventas (ANIO_MES_VENTA
                                DIA_VENTA
                                CODIGO_EMPRESA
                                )
  VALUES (to_number(Ln_anio||lpad(Ln_mes,2,'0')),--ANIO_MES_VENTA          NOT NULL
          NUMBER(6)
          Ln_dia,--DIA_VENTA          NOT NULL NUMBER(2)
          J.codigo_empresa--CODIGO_EMPRESA          NOT NULL NUMBER(2)
          );
EXCEPTION
  WHEN others THEN
    NULL;
END;
END LOOP;-- FOR Ln_dia IN 1..Ln_dia_final LOOP
END LOOP;-- FOR Ln_mes IN 1..12 LOOP
END LOOP;-- FOR Ln_anio IN Ln_anio_inicial..Ln_anio_final LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
-- ***** Borrarr los Registros del Periodo Abierto
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--          Como datos definitos del cierre de mes
--
DELETE dm_indice_cumplimiento_ventas indcumven
WHERE indcumven.mes_cerrado = 'A'--A=MES ABIERTO, C=MES CERRADO
AND indcumven.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
```



```
--
-- ***** Inserta las Ventas del Periodo Abierto
--
--
--
Ln_contador := 0;
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
--
*****
**
--
-- Funcion      : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--              de la tabla AD_PARAMETROS_GENERALES
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);
IF Ln_codigo_nacion IS NULL THEN
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
```



```
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--
--                   de la tabla CC_PERIODOS_CERRADOS
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                    J.cartera,--Pv_cartera                IN      VARCHAR2,
                                                    Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                    Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                    Ld fecha final,--Pd_fecha_final      IN OUT DATE,
                                                    Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Obtiene la Fecha Inicial y Final del Siguiente Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
Ln_periodo_mes_sig := to_number(to_char(Ld_fecha_inicial_mes_sig,'yyyymm'));
--
--
Ld_fecha_inicial := Ld_fecha_inicial_mes_sig;
Ld_fecha_final := Ld_fecha_final_mes_sig;
--
-- Obtiene Las Ventas Detalladas de un rango de fechas
-- Este select es el mismo que la vista VT_VTAS_DETALLADAS esta vista es utilizada para cargar la hoja excel
que utilizaba Franklin Bernal
--
FOR C IN C_ventas_detalladas LOOP
  --
  --
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
```



```
-- de la tabla AD AGENCIAS X CANTONES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia IN NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion IN NUMBER,
J.codigo_empresa,--Pn_codigo_empresa IN NUMBER,
Lv_codigo_canton,--Pv_codigo_canton IN OUT VARCHAR2,
Ln_codigo_region,--Pn_codigo_region IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
-- de la tabla AD_AGENCIAS_X_CANTONES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia IN NUMBER,
Ln_codigo_region,--Pn_codigo_region IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion IN NUMBER,
J.codigo_empresa,--Pn_codigo_empresa IN NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Actualiza las ventas netas del Cumplimiento de ventas
--
UPDATE dm_indice_cumplimiento_ventas indcumven
SET indcumven.valor_venta_neta = indcumven.valor_venta_neta + C.valor
WHERE indcumven.anio_mes_venta = to_number(C.fecha)
```



```
AND      indcumven.dia_venta      = to_number(C.dia_venta)
AND      indcumven.codigo_agencia = C.cod_agencia
AND      indcumven.codigo_canton  = Lv_codigo_canton
AND      indcumven.codigo_provincia = Lv_codigo_provincia
AND      indcumven.codigo_region  = Ln_codigo_region
AND      indcumven.codigo_nacion  = Ln_codigo_nacion
AND      indcumven.codigo_empresa = J.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO dm_indice_cumplimiento_ventas (ANIO_MES_VENTA      ,
                                             DIA_VENTA          ,
                                             CODIGO_AGENCIA      ,
                                             CODIGO_EMPRESA     ,
                                             VALOR_VENTA_NETA   ,
                                             MES_CERRADO       )
  VALUES (to_number(C.fecha), --PERIODO                NOT NULL NUMBER(6)
           to_number(C.dia_venta), --DIA_VENTA          NOT NULL NUMBER(2)
           C.cod_agencia, --CODIGO_AGENCIA              NOT NULL NUMBER(4)
           J.codigo_empresa, --CODIGO_EMPRESA          NOT NULL NUMBER(2)
           C.valor, --VALOR_VENTA_NETA                 NOT NULL NUMBER(30,2)
           'A' --MES_CERRADO                            NOT NULL VARCHAR2(1)
           );
END IF; -- IF Sql%Notfound THEN
--
--
--
Ln_contador := Ln_contador + 1;
IF Ln_contador > 2000 THEN
  commit;
  Ln_contador := 0;
END IF; -- IF Ln_contador > 2000 THEN
END LOOP; -- FOR C IN C_ventas_detalladas LOOP
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
```



```
--  
--  
commit;  
EXCEPTION  
WHEN OTHERS THEN  
    Pv_CodigoError := Pv_CodigoError||'. '||SQLERRM;  
END P_ACTUALIZACION_DIA_IND_CUMVTA;  
END PK_CC_PROCESO_IND_CUMPLIM_VTA;  
/
```



Anexo A32.

Script actualización índice de rentabilidad.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_IND_RENTABILIDAD IS
--
--
*****
**
--
-- Procedimiento      : Carga el Indice de rentabilidad Para el Nuevo Mes
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla DM_INDICE_RENTABILIDAD
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Viernes 1 de Abril de 2010.
PROCEDURE P_CARGAR_IND_REN_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA DM_INDICE_RENTABILIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                    costo_ventas      = 0,
--                    precio_ventas     = 0,
--                    costo_devoluciones = 0,
--                    precio_devoluciones = 0.
--                    costo_netto       = 0,
--                    venta_neta        = 0,
--                    utilidad           = 0
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Lunes 4 de Abril de 2010.
PROCEDURE P_ACTUALIZACION_DIARIA_IND_REN (Pv_CodigoError  IN OUT VARCHAR2);
```



```
END PK_CC_PROCESO_IND_RENTABILIDAD;
/
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_IND_RENTABILIDAD IS
  --
  --
  ****
  **
  --
  -- Procedimiento      : Carga el Indice de rentabilidad Para el Nuevo Mes
  --                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
  --                   : Inserta en la Tabla DM_INDICE_RENTABILIDAD
  -- Autor              : Ing. Giovanni Chasifan Chicaiza.
  -- Fecha creacion    : Viernes 1 de Abril de 2010.
  PROCEDURE P_CARGAR_IND_REN_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2)
  IS
    --
    -- Variables Locales
    --
    Lv_CodigoError          VARCHAR2(3000);
    Ln_codigo_empresa      ad_empresas.codigo_empresa%TYPE;
    Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
    Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
    Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
    Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;
    Ln_contador            NUMBER(30);
    Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;
    Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;
    Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
    Lv_cod_modelo_cat_linea producto.cod_modelo_cat%TYPE;
    Lv_cod_item_cat_linea  producto.cod_item_cat%TYPE;
    Ln_cod_marca            cc_balances_mensuales.codigo_marca%TYPE;
    --
    -- Obtiene los Tipos de Cartera
    --
    CURSOR C_tipos_carteras IS
      SELECT *
      FROM   cc_tipos_carteras tipcar
```



```
WHERE tipcar.estado = 'N'
AND tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
CURSOR C_rentabilidad IS
select a.empresa,
a.cod_agencia,
e.nombre agencia,
to_number(to_char(a.fecha,'yyyymm')) fecha,
a.cod_producto,
trim(b.nombre) nombre,
d.nombre linea,
c.nombre marca,
a.pagos num_pagos,
g.nombre tipo_cliente,
g.cod_modelo,
g.cod_item,

round(sum(decode(a.deb_cred,2,decode(a.costo_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costo_reexpresado),0)),2) costo_ventas,
round(sum(decode(a.deb_cred,2,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1),0)),2) precio_ventas,

round(sum(decode(a.deb_cred,1,decode(a.costo_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costo_reexpresado)*-1,0)),2) costo_devoluciones,
round(sum(decode(a.deb_cred,1,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1)*-1,0)),2) precio_devoluciones,
i.forma_pago
from sta_rentabilidad_temporal a,
producto b,
marca c,
tg_modelo_item d,
tg_agencia e,
```



```
cliente_hor f,  
sta_parametros_vista_elec h,  
tg_modelo_item g,  
comprobante i  
where a.empresa = h.empresa  
and h.activo='S'  
and a.usuario=user  
and a.cod_persona!='0190331024001'  
and b.empresa=a.empresa  
and b.cod_producto=a.cod_producto  
and c.empresa=b.empresa  
and c.cod_marca=b.cod_marca  
and d.empresa=b.empresa  
and d.cod_modelo='PRO2'  
and d.cod_item=b.cod_item_cat  
and e.empresa=a.empresa  
and e.cod_agencia=a.cod_agencia  
and f.empresah=a.empresa  
and f.cod_clienteh=a.cod_persona  
and g.empresa=f.empresah  
and g.cod_modelo='CLI1'  
and g.cod_item=f.cod_tipo_clienteh  
and d.nombre<>'Electropolis S.A.'  
and i.empresa=a.empresa  
and i.tipo_comprobante=a.tipo_comprobante  
and i.cod_comprobante=a.cod_comprobante  
group by a.empresa,  
a.cod_agencia,  
to_number(to_char(a.fecha,'yyyymm')),  
a.cod_producto ,  
b.nombre ,  
c.nombre ,  
a.pagos,  
d.nombre ,  
e.nombre ,  
i.forma_pago,  
g.nombre,
```



```
g.cod_modelo,  
g.cod_item  
order by fecha asc;  
BEGIN  
--  
-- Inserta Tablas Maestras del DM_INDICE_RENTABILIDAD  
--  
--  
--  
*****  
**  
--  
-- Procedimiento      : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS  
TABLAS DE DIMENSIONES  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación     : Miercoles 4 de Julio de 2010.  
PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS_NUEVOS;  
--  
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto  
--  
FOR J IN C_tipos_carteras LOOP  
--  
--  
*****  
**  
--  
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y  
CARTERA  
--                    de la tabla CC_PERIODOS_CERRADOS  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación     : Martes 1 de Junio de 2010.  
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,  
J.cartera,--Pv_cartera      IN      VARCHAR2,  
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,  
ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,  
ld_fecha_final,--Pd_fecha_final      IN OUT DATE,  
Lv_CodigoError);
```



```
IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- ***** Borrando las Transacciones ABIertas porque los datos seran cargados mas adelante
-- Como datos definitos del cierre de mes
--
DELETE dm_indice_rentabilidad indren
WHERE indren.periodo      = Ln_periodo_balance
AND indren.mes cerrado    = 'A'--A=MES ABIERTO, C=MES CERRADO
AND indren.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- ***** INSERTA TODAS LAS VENTAS EN EL PERIODO CERRADO
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  --
  --
  *****
  **
  --
  -- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
  --                  : de la tabla AD_PARAMETROS_GENERALES
  -- Autor            : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación  : Martes 19 de Octubre de 2010.
  Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                                    );
  IF Ln_codigo_nacion IS NULL THEN
```



```
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera      IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.Codigo_Empresa;
Ln_contador := 0;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta_rentabilidad_temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
```



```
--
COMMIT;
--
-- ***** Obtiene a Cargar los Valores para el Calculo de la Rentabilidad
--
ps_rentabilidad (Ln_codigo_empresa,--pcod_empresa          number,
                USER,--p_user                          varchar2,
                Ld_fecha_inicial,--p_fecha_inicial        date,
                Ld_fecha_final--p_fecha_final            date
                );
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
FOR C IN C_rentabilidad LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.empresa,--Pn_codigo_empresa IN      NUMBER,
                                           C.cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat_linea,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat_linea,--Pv_cod_item_cat  IN OUT VARCHAR2,
                                           Lv_CodigoError
                                           );
IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
--
--
*****
**
--
```



```
-- Procedimiento      : Obtiene EL CODIGO DE LA MARCA DE UN PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 15 de Diciembre de 2009.
PK_OBTENER_PRODUCTO.P_COD_MARCA (C.empresa,--Pn_codigo_empresa IN      NUMBER,
                                   C.cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                   Ln_cod_marca,--Pn_codigo_marca  IN OUT NUMBER,
                                   Lv_CodigoError
                                   );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;

--
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia  IN      NUMBER,
                                                  Ln_codigo_nacion,--Pn_codigo_nacion  IN      NUMBER,
                                                  C.empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                  Lv_codigo_canton,--Pv_codigo_canton  IN OUT VARCHAR2,
                                                  Ln_codigo_region,--Pn_codigo_region  IN OUT NUMBER,
                                                  Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN

--
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
```



```
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia IN NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion IN NUMBER,
                                             C.empresa,--Pn_codigo_empresa IN NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
UPDATE dm_indice_rentabilidad indren
SET    indren.costo_ventas = indren.costo_ventas + C.costo_ventas,
       indren.precio_ventas = indren.precio_ventas + C.precio_ventas,
       indren.costo_devoluciones = indren.costo_devoluciones + C.costo_devoluciones,
       indren.precio_devoluciones = indren.precio_devoluciones + C.precio_devoluciones
WHERE indren.periodo = Ln_periodo_balance
AND   indren.cod_producto = C.cod_producto
AND   indren.anio_mes_venta = C.fecha
AND   indren.forma_pago = C.forma_pago
AND   indren.plazo = C.num_pagos
AND   indren.codigo_agencia = C.cod_agencia
AND   indren.codigo_canton = Lv_codigo_canton
AND   indren.codigo_provincia = Lv_codigo_provincia
AND   indren.codigo_region = Ln_codigo_region
AND   indren.cod_modelo_cat_linea = Lv_cod_modelo_cat_linea
AND   indren.cod_item_cat_linea = Lv_cod_item_cat_linea
AND   indren.cod_marca = Ln_cod_marca
AND   indren.tipo_cliente = C.tipo_cliente
AND   indren.codigo_nacion = Ln_codigo_nacion
AND   indren.codigo_empresa = C.empresa;
--
-- Verifica si Existe Registro
```



```
--  
IF SQL%NOTFOUND THEN  
  INSERT INTO dm_indice_rentabilidad (PERIODO  
    COD_PRODUCTO  
    ANIO_MES_VENTA  
    FORMA_PAGO  
    PLAZO  
    CODIGO_AGENCIA  
    TIPO_CLIENTE  
    COD_ITEM TIPO CLIENTE  
    COD MODELO ITEM TIPO CLIENTE  
    CODIGO_NACION  
    CODIGO_EMPRESA  
    COSTO_VENTAS  
    PRECIO_VENTAS  
    COSTO_DEVOLUCIONES  
    PRECIO_DEVOLUCIONES  
    MES_CERRADO  
  )  
VALUES (Ln_periodo_balance, --PERIODO          NOT NULL NUMBER(6)  
  C.cod_producto, --COD_PRODUCTO          NOT NULL VARCHAR2(14)  
  C.fecha, --ANIO_MES_VENTA          NOT NULL NUMBER(6)  
  C.forma_pago, --FORMA_PAGO          NOT NULL VARCHAR2(3)  
  C.num_pagos, --PLAZO          NOT NULL NUMBER(3)  
  C.cod_agencia, --CODIGO_AGENCIA          NOT NULL NUMBER(4)  
  C.tipo_cliente, --TIPO_CLIENTE          NOT NULL VARCHAR2(50)  
  C.cod_item, --COD_ITEM_TIPO_CLIENTE          NOT NULL VARCHAR2(3)  
  C.cod_modelo, --COD_MODELO_ITEM_TIPO_CLIENTE          NOT NULL VARCHAR2(8)  
  Ln_codigo_nacion, --CODIGO_NACION          NOT NULL NUMBER(3)  
  C.empresa, --CODIGO_EMPRESA          NOT NULL NUMBER(2)  
  C.costos_ventas, --COSTO_VENTAS          NOT NULL NUMBER(30,2)  
  C.precio_ventas, --PRECIO_VENTAS          NOT NULL NUMBER(30,2)  
  C.costos_devoluciones, --COSTO_DEVOLUCIONES          NOT NULL NUMBER(30,2)  
  C.precio_devoluciones, --PRECIO_DEVOLUCIONES          NOT NULL NUMBER(30,2)  
  'C' --MES_CERRADO          NOT NULL VARCHAR2(1)  
);  
END IF; -- IF SQL%NOTFOUND THEN
```



```
END LOOP;-- FOR C IN C_rentabilidad LOOP
--
--
--
COMMIT;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta rentabilidad temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
--
COMMIT;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := Pv_CodigoError||'. '||SQLERRM;
END P_CARGAR_IND_REN_CIERRE_MES;
--
--
*****
**
--
-- Procedimiento : ACTUALIZA DIARIAMENTE LA TABLA DM_INDICE_RENTABILIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
-- costo_ventas = 0,
-- precio_ventas = 0,
-- costo_devoluciones = 0,
-- precio_devoluciones = 0.
-- costo_neto = 0,
-- venta_neta = 0,
-- utilidad = 0
--
-- Obtiene los comprobantes del ultimo cierre de Balance
```



```
--
-- Autor           : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion  : Lunes 4 de Abril de 2010.
PROCEDURE P_ACTUALIZACION_DIARIA_IND_REN (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_codigo_empresa      ad_empresas.codigo_empresa%TYPE;
Ln_periodo_balance     cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;
Ln_contador            NUMBER(30);
Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia    ad_cantones_x_provincias.codigo_provincia%TYPE;
Lv_cod_modelo_cat_linea producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat_linea  producto.cod_item_cat%TYPE;
Ln_cod_marca           cc_balances_mensuales.codigo_marca%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM   cc_tipos_carteras tipcar
WHERE  tipcar.estado = 'N'
AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
CURSOR C_rentabilidad IS
select a.empresa,
a.cod_agencia,
e.nombre agencia,
```



```
to_number(to_char(a.fecha,'yyyymm')) fecha,
a.cod_producto,
trim(b.nombre) nombre,
d.nombre linea,
c.nombre marca,
a.pagos num_pagos,
g.nombre tipo_cliente,
g.cod_modelo,
g.cod_item,

round(sum(decode(a.deb_cred,2,decode(a.costos_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costos_reexpresado),0)),2) costos_ventas,
round(sum(decode(a.deb_cred,2,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1),0)),2) precio_ventas,

round(sum(decode(a.deb_cred,1,decode(a.costos_reexpresado,0,decode(b.serie,'S',stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,a.numero_serie,sysdate),stock.ks_inventario_serie.obt_costo_serie(a.empresa,a.cod_producto,null,sysdate))*decode(a.cantidad_i,0,1,a.cantidad_i)),a.costos_reexpresado)*-1,0)),2) costos_devoluciones,
round(sum(decode(a.deb_cred,1,(a.total_linea-a.iva)/decode(b.serie,'S',decode(a.cantidad_i,0,1,a.cantidad_i),1)*-1,0)),2) precio_devoluciones,
i.forma_pago
from sta_rentabilidad_temporal a,
producto b,
marca c,
tg_modelo_item d,
tg_agencia e,
cliente_hor f,
sta_parametros_vista_elec h,
tg_modelo_item g,
comprobante i
where a.empresa = h.empresa
and h.activo='S'
and a.usuario=user
and a.cod_persona!='0190331024001'
and b.empresa=a.empresa
and b.cod_producto=a.cod_producto
```



```
and c.empresa=b.empresa
and c.cod_marca=b.cod_marca
and d.empresa=b.empresa
and d.cod_modelo='PRO2'
and d.cod_item=b.cod_item_cat
and e.empresa=a.empresa
and e.cod_agencia=a.cod_agencia
and f.empresah=a.empresa
and f.cod_clienteh=a.cod_persona
and g.empresa=f.empresah
and g.cod_modelo='CLI1'
and g.cod_item=f.cod_tipo_clienteh
and d.nombre<>'Electropolis S.A.'
and i.empresa=a.empresa
and i.tipo_comprobante=a.tipo_comprobante
and i.cod_comprobante=a.cod_comprobante
group by a.empresa,
a.cod_agencia,
to_number(to_char(a.fecha,'yyyymm')),
a.cod_producto ,
b.nombre ,
c.nombre ,
a.pagos,
d.nombre ,
e.nombre ,
i.forma_pago,
g.nombre,
g.cod_modelo,
g.cod_item
order by fecha asc;
--
-- Obtiene las fechas de las ventas
--
CURSOR C_comprobante IS
SELECT to_number(to_char(com.fecha,'yyyymm')) anio_mes_venta, com.empresa codigo_empresa
FROM comprobante com
WHERE com.tipo_comprobante in ('A0','NV','D0','DB')
```



```
AND NOT EXISTS (SELECT 'X'
                FROM dm_anio_mes_ventas animesven
                WHERE animesven.anio_mes_venta = to_number(to_char(com.fecha, 'yyyymm'))
                AND animesven.codigo_empresa = com.empresa
                )
GROUP BY to_number(to_char(com.fecha, 'yyyymm')), com.empresa
ORDER BY to_number(to_char(com.fecha, 'yyyymm')), com.empresa;
--
-- Obtiene el plazo de las ventas
--
CURSOR C_comprobante_x_plazo IS
SELECT com.numero_pagos, com.empresa codigo_empresa
FROM comprobante com
WHERE com.tipo_comprobante in ('A0', 'NV', 'D0', 'DB')
AND NOT EXISTS (SELECT 'X'
                FROM dm_plazos_ventas plaven
                WHERE plaven.plazo = com.numero_pagos
                AND plaven.codigo_empresa = com.empresa
                )
GROUP BY com.numero_pagos, com.empresa
ORDER BY com.empresa, com.numero_pagos;
BEGIN
--
-- Inserta Tablas Maestras del DM_INDICE_RENTABILIDAD
--
--
--
*****
**
--
-- Procedimiento : INSERTA LOS NUEVOS REGISTROS QUE EXISTEN EN LA TABLAS OLTP Y QUE NO EXISTEN EN LAS
TABLAS DE DIMENSIONES
-- DEL MES ACTUAL
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Miercoles 4 de Julio de 2010.
PK_DM_PROCESOS_TB_DIMENSIONES.P_INSERTAR_REGISTROS_NUEVOS;
--
```



```
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
BEGIN
    INSERT INTO dm_periodos_balances (PERIODO
                                     CODIGO_EMPRESA
                                    )
    VALUES (to_number(to_char(sysdate,'yyyymm')),
            J.codigo_empresa
           );

    --
    --
    --
    commit;
EXCEPTION
    WHEN others THEN
        Lv_CodigoError := null;
END; -- BEGIN
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
-- Obtiene las fechas de las ventas
--
FOR C IN C_comprobante LOOP
    INSERT INTO dm_anio_mes_ventas (ANIO_MES_VENTA
                                   CODIGO_EMPRESA
                                  )
    VALUES (C.anio_mes_venta,
            C.codigo_empresa
           );
END LOOP; -- FOR C IN C_comprobante LOOP
--
--
--
commit;
--
-- Obtiene el plazo de las ventas
--
```



```
FOR C IN C_comprobante_x_plazo LOOP
  INSERT INTO dm_plazos_ventas (PLAZO
                                CODIGO_EMPRESA
                              )
  VALUES (C.numero_pagos,
           C.codigo_empresa
          );
END LOOP;-- FOR C IN C_comprobante_x_plazo LOOP
--
--
--
commit;
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
  --                    de la tabla CC_PERIODOS_CERRADOS
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Martes 1 de Junio de 2010.
  PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                         J.cartera,--Pv_cartera          IN      VARCHAR2,
                                                         Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                         Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                         Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
                                                         Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  -- Suma un Mes al Ultimo Cierre de Balance
```



```
--
Ln_periodo_balance := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--           Como datos definitos del cierre de mes
--
UPDATE dm_indice_rentabilidad indren
SET    indren.costo_ventas      = 0,
       indren.precio_ventas    = 0,
       indren.costo_devoluciones = 0,
       indren.precio_devoluciones = 0,
       indren.costo_neto       = 0,
       indren.venta_neta       = 0,
       indren.utilidad         = 0
WHERE  indren.periodo         = Ln_periodo_balance
AND    indren.mes_cerrado     = 'A'--A=MES ABIERTO, C=MES CERRADO
AND    indren.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- ***** INSERTA TODAS LAS VENTAS EN EL PERIODO CERRADO
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
Ln_codigo_nacion := null;
--
--
*****
**
--
-- Funcion           : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                   de la tabla AD_PARAMETROS_GENERALES
```



```
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--
--                   de la tabla CC PERIODOS CERRADOS
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
Lv_CodigoError);
IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Suma un Mes al Ultimo Cierre de Balance
--
Ln_periodo_balance := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
Ld_fecha_inicial := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final := add_months(trunc(Ld_fecha_final),1);
--
--
```



```
--
Ln_codigo_empresa := J.Codigo_Empresa;
Ln_contador := 0;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta_rentabilidad_temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
COMMIT;
--
-- ***** Obtiene a Cargar los Valores para el Calculo de la Rentabilidad
--
ps_rentabilidad (Ln_codigo_empresa,--pcod_empresa          number,
                USER,--p_user                          varchar2,
                Ld_fecha_inicial,--p_fecha_inicial        date,
                Ld_fecha_final--p_fecha_final            date
                );
--
-- Obtiene los Costos y Precios para el Calculo de la Rentabilidad
--
FOR C IN C_rentabilidad LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.empresa,--Pn_codigo_empresa IN    NUMBER,
                                           C.cod_producto,--Pv_cod_producto IN    VARCHAR2,
```



```

                                Lv_cod_modelo_cat_linea,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                Lv_cod_item_cat_linea,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                Lv_CodigoError
                                );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DE LA MARCA DE UN PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 15 de Diciembre de 2009.
PK_OBTENER_PRODUCTO.P_COD_MARCA (C.empresa,--Pn_codigo_empresa IN      NUMBER,
                                C.cod_producto,--Pv_cod_producto IN  VARCHAR2,
                                Ln_cod_marca,--Pn_codigo_marca  IN OUT NUMBER,
                                Lv_CodigoError
                                );
IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.cod_agencia,--Pn_codigo_agencia IN      NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion IN  NUMBER,
                                                C.empresa,--Pn_codigo_empresa  IN  NUMBER,
                                                Lv_codigo_canton,--Pv_codigo_canton IN OUT VARCHAR2,
```



```
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.cod_agencia,--Pn_codigo_agencia      IN      NUMBER,
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
C.empresa,--Pn_codigo_empresa      IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
UPDATE dm_indice_rentabilidad indren
SET   indren.costo_ventas = indren.costo_ventas + C.costo_ventas,
      indren.precio_ventas = indren.precio_ventas + C.precio_ventas,
      indren.costo_devoluciones = indren.costo_devoluciones + C.costo_devoluciones,
      indren.precio_devoluciones = indren.precio_devoluciones + C.precio_devoluciones
WHERE indren.periodo      = Ln_periodo_balance
AND   indren.cod_producto = C.cod_producto
AND   indren.anio_mes_venta = C.fecha
AND   indren.forma_pago    = C.forma_pago
AND   indren.plazo        = C.num_pagos
```



```
AND indren.codigo_agencia = C.cod_agencia
AND indren.codigo_canton = Lv_codigo_canton
AND indren.codigo_provincia = Lv_codigo_provincia
AND indren.codigo_region = Ln_codigo_region
AND indren.cod_modelo_cat_linea = Lv_cod_modelo_cat_linea
AND indren.cod_item_cat_linea = Lv_cod_item_cat_linea
AND indren.cod_marca = Ln_cod_marca
AND indren.tipo_cliente = C.tipo_cliente
AND indren.codigo_nacion = Ln_codigo_nacion
AND indren.codigo_empresa = C.empresa;
--
-- Verifica si Existe Registro
--
IF SQL%NOTFOUND THEN
  INSERT INTO dm_indice_rentabilidad (PERIODO
    COD_PRODUCTO
    ANIO_MES_VENTA
    FORMA_PAGO
    PLAZO
    CODIGO_AGENCIA
    TIPO_CLIENTE
    COD_ITEM_TIPO_CLIENTE
    COD_MODELO_ITEM_TIPO_CLIENTE
    CODIGO_NACION
    CODIGO_EMPRESA
    COSTO_VENTAS
    PRECIO_VENTAS
    COSTO_DEVOLUCIONES
    PRECIO_DEVOLUCIONES
    MES_CERRADO
  )
VALUES (Ln_periodo_balance, --PERIODO NOT NULL NUMBER(6)
  C.cod_producto, --COD_PRODUCTO NOT NULL VARCHAR2(14)
  C.fecha, --ANIO_MES_VENTA NOT NULL NUMBER(6)
  C.forma_pago, --FORMA_PAGO NOT NULL VARCHAR2(3)
  C.num_pagos, --PLAZO NOT NULL NUMBER(3)
  C.cod_agencia, --CODIGO_AGENCIA NOT NULL NUMBER(4)
```



```

C.tipo_cliente,--TIPO_CLIENTE                                NOT NULL VARCHAR2(50)
C.cod_item,--COD_ITEM_TIPO_CLIENTE                          NOT NULL VARCHAR2(3)
C.cod_modelo,--COD_MODELO_ITEM_TIPO_CLIENTE                 NOT NULL VARCHAR2(8)
Ln_codigo_nacion,--CODIGO_NACION                            NOT NULL NUMBER(3)
C.empresa,--CODIGO_EMPRESA                                  NOT NULL NUMBER(2)
C.costos_ventas,--COSTO_VENTAS                              NOT NULL NUMBER(30,2)
C.precio_ventas,--PRECIO_VENTAS                            NOT NULL NUMBER(30,2)
C.costos_devoluciones,--COSTO_DEVOLUCIONES                 NOT NULL NUMBER(30,2)
C.precio_devoluciones,--PRECIO_DEVOLUCIONES               NOT NULL NUMBER(30,2)
'A'--MES_CERRADO                                           NOT NULL VARCHAR2(1)
);
END IF;-- IF SQL%NOTFOUND THEN
END LOOP;-- FOR C IN C_rentabilidad LOOP
--
--
--
COMMIT;
--
-- ***** Elimina los registros de la Tabla donde se Cargan los Valores para el Calculo de la
Rentabilidad
--
DELETE sta_rentabilidad_temporal rentem
WHERE rentem.usuario = USER
AND rentem.empresa = J.codigo_empresa;
--
--
--
COMMIT;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := Pv_CodigoError||'. '||SQLERRM;
END P_ACTUALIZACION_DIARIA_IND_REN;
END PK_CC_PROCESO_IND_RENTABILIDAD;
/
```



Anexo A33.

Script actualización índice de morosidad.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_INDICE_MOROSIDAD IS
--
--
*****
**
--
-- Procedimiento      : Carga el Indice de Morosidad Para el Nuevo Mes
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_INDICES_DE_MOROSIDAD
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Miercoles 1 de Diciembre de 2010.
PROCEDURE P_CARGAR_IND_MOR_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_INDICE_DE_MOROSIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                    SALDO_CARTERA,
--                    TOTAL_CARTERA_AL_CIERRE (Tenia_que_cobrar)
--                    VALOR_COBRADO
--                    CARTERA_VENCIDA
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Jueves 2 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIARIA_IND_MOR;
END;
/
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_INDICE_MOROSIDAD IS
--
```



```
--
*****
**
--
-- Procedimiento      : Carga el Indice de Morosidad Para el Nuevo Mes
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_INDICE_DE_MOROSIDAD
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Miercoles 1 de Diciembre de 2010.
PROCEDURE P_CARGAR_IND_MOR_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_tenia_que_cobrar     NUMBER(30,2);
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_siguiete_periodo_trabajo NUMBER(6);
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
Ln_contador             NUMBER(30);
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;
--
-- Obtiene los Tipos de Cartera
--
```



```
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;-- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene todos los Ventas realizadas
--
CURSOR C siembra x vencimiento IS
  SELECT /*+ first_rows */ sum(siexven.valor_vencimiento) valor_ventas,
    siexven.anio_mes_venta,
    siexven.codigo_agencia,
    siexven.codigo_canton,
    siexven.codigo_provincia,
    siexven.codigo_region,
    siexven.codigo_politica,
    siexven.cod_modelo_cat_linea,
    siexven.cod_item_cat_linea,
    siexven.cod_marca,
    siexven.codigo_nacion,
    siexven.codigo_empresa
  FROM   cc_siembra_x_vencimiento siexven
  WHERE  siexven.codigo_empresa = Ln_codigo_empresa
  GROUP BY siexven.anio_mes_venta,
    siexven.codigo_agencia,
    siexven.codigo_canton,
    siexven.codigo_provincia,
    siexven.codigo_region,
    siexven.codigo_politica,
    siexven.cod_modelo_cat_linea,
    siexven.cod_item_cat_linea,
    siexven.cod_marca,
    siexven.codigo_nacion,
    siexven.codigo_empresa;
--
-- Obtiene los creditos de un determinado periodo
```



```
--
CURSOR C_balances IS
  SELECT /*+ first_rows */ *
  FROM   cc_balances_mensuales balmen
  WHERE  balmen.período_cierre = Ln_período_balance
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  ORDER BY balmen.ano_venta, balmen.mes_venta;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--
-- de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                       J.cartera,--Pv_cartera                IN      VARCHAR2,
                                                       Ln_período_balance,--Pn_período_balance  IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final       IN OUT DATE,
                                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
-- Como datos definitos del cierre de mes
--
DELETE cc_indice_de_morosidad indmor
WHERE  indmor.período          = Ln_período_balance
```



```
AND indmor.mes_cerrado = 'A'--A=MES ABIERTO, C=MES CERRADO
AND indmor.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- ***** INSERTA TODAS LAS VENTAS EN EL PERIODO CERRADO
--
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
-- de la tabla CC_PERIODOS_CERRADOS
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa IN NUMBER,
J.cartera,--Pv_cartera IN VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final IN OUT DATE,
Lv_CodigoError);
IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.Codigo_Empresa;
```



```
Ln_contador := 0;
--
-- ***** Obtiene todos los Ventas realizadas
--
FOR C IN C_siembra_x_vencimiento LOOP
  --
  --
  *****
  **
  --
  -- Función          : OBTIENE EL TOTAL DE VENCIMIENTOS (SIEMBRA) A UN DETERMINADO ANIO Y MES DE
VENCIMIENTO
  --
  -- en la tabla CC_SIEMBRA_X_VENCIMIENTO
  -- Autor           : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación  : Martes 14 de Diciembre de 2010.
  Ln_tenia_que_cobrar := PK_OBTENER_CC_SIE_X_VEN.F_TENIA_QUE_COBRAR (Ln_periodo_balance, --
Pn_anio_mes_vencimiento IN NUMBER, --ANIO_MES_VENCIMIENTO                                NOT NULL NUMBER(6)
  C.anio_mes_venta, --Pn_anio_mes_venta
IN NUMBER, --ANIO_MES_VENTA                                NOT NULL NUMBER(6)
  C.codigo_agencia, --Pn_codigo_agencia
IN NUMBER, --CODIGO_AGENCIA                                NOT NULL NUMBER(4)
  C.codigo_canton, --Pv_codigo_canton
IN VARCHAR2, --CODIGO_CANTON                              NOT NULL VARCHAR2(4)
  C.codigo_provincia, --Pv_codigo_provincia
IN VARCHAR2, --CODIGO_PROVINCIA                          NOT NULL VARCHAR2(6)
  C.codigo_region, --Pn_codigo_region
IN NUMBER, --CODIGO_REGION                                NOT NULL NUMBER(3)
  C.codigo_politica, --Pn_codigo_politica
IN NUMBER, --CODIGO_POLITICA                              NOT NULL NUMBER(2)
  C.cod_modelo_cat_linea, --
Pv_cod_modelo_cat_linea IN VARCHAR2, --COD_MODELO_CAT_LINEA                                NOT NULL VARCHAR2(8)
  C.cod_item_cat_linea, --
Pv_cod_item_cat_linea IN VARCHAR2, --COD_ITEM_CAT_LINEA                                NOT NULL VARCHAR2(3)
  C.cod_marca, --Pn_cod_marca
IN NUMBER, --COD_MARCA                                    NOT NULL NUMBER(3)
  C.codigo_nacion, --Pn_codigo_nacion
IN NUMBER, --CODIGO_NACION                                NOT NULL NUMBER(3)
```



```
IN NUMBER--CODIGO_EMPRESA NOT NULL NUMBER(2)
--
-- Inserta el Indice de Morosidad
--
INSERT INTO cc_indice_de_morosidad (PERIODO
ANIO_MES_VENTA
CODIGO_AGENCIA
CODIGO_CANTON
CODIGO_PROVINCIA
CODIGO_REGION
CODIGO_POLITICA
COD_MODELO_CAT_LINEA
COD_ITEM_CAT_LINEA
COD_MARCA
CODIGO_NACION
CODIGO_EMPRESA
VALOR_VENTAS
SALDO_CARTERA
TOTAL_CARTERA_AL_CIERRE
CARTERA_VENCIDA
MES_CERRADO
)
VALUES (Ln_periodo_balance,--PERIODO
C.anio_mes_venta,--ANIO_MES_VENTA
C.codigo_agencia,--CODIGO_AGENCIA
C.codigo_canton,--CODIGO_CANTON
C.codigo_provincia,--CODIGO_PROVINCIA
C.codigo_region,--CODIGO_REGION
C.Codigo_Politica,--CODIGO_POLITICA
C.cod_modelo_cat_linea,--COD_MODELO_CAT_LINEA
C.cod_item_cat_linea,--COD_ITEM_CAT_LINEA
C.cod_marca,--COD_MARCA
C.codigo_nacion,--CODIGO_NACION
C.codigo_empresa,--CODIGO_EMPRESA
C.valor_ventas,--VALOR_VENTAS
C.codigo_empresa--Pn_codigo_empresa
);
NOT NULL NUMBER(6)
NOT NULL NUMBER(6)
NOT NULL NUMBER(4)
NOT NULL VARCHAR2(4)
NOT NULL VARCHAR2(6)
NOT NULL NUMBER(3)
NOT NULL NUMBER(2)
NOT NULL VARCHAR2(8)
NOT NULL VARCHAR2(3)
NOT NULL NUMBER(3)
NOT NULL NUMBER(3)
NOT NULL NUMBER(2)
NOT NULL NUMBER(30,2)
```



```

    0,--SALDO_CARTERA                                NOT NULL NUMBER(30,2)
    Ln_tenia_que_cobrar,--TOTAL_CARTERA_AL_CIERRE    NOT NULL NUMBER(30,2)
    0,--CARTERA_VENCIDA                                NOT NULL NUMBER(30,2)
    'C'--MES_CERRADO                                  NOT NULL VARCHAR2(1)
);
--
--
--
Ln_contador := Ln_contador + 1;
IF Ln contador > 2000 THEN
    COMMIT;
    Ln_contador := 0;
END IF;-- IF Ln_contador > 2000 THEN
END LOOP;-- FOR C_siembra_x_vencimiento LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
    Ln_codigo_nacion := null;
--
--
*****
**
--
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                  de la tabla AD_PARAMETROS_GENERALES
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);
IF Ln_codigo_nacion IS NULL THEN
```



```
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.  
Empresa='||to_char(J.codigo_empresa);  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Ln_codigo_nacion IS NULL THEN  
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y  
CARTERA  
--                    de la tabla CC_PERIODOS_CERRADOS  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación     : Martes 1 de Junio de 2010.  
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,  
J.cartera,--Pv_cartera      IN      VARCHAR2,  
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,  
Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,  
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,  
Lv_CodigoError);  
  
IF Lv_CodigoError IS NOT NULL THEN  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
-- Obtiene la Fecha Inicial y Final del Siguiente Mes del Balance  
--  
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);  
--  
--  
--  
Ln_codigo_empresa := J.codigo_empresa;  
--  
-- Obtiene los creditos de un determinado periodo  
--  
FOR C IN C_balances LOOP  
  Ln_tenia_que_cobrar := 0;
```



```
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA  
--                    : de la tabla AD_AGENCIAS_X_CANTONES  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Jueves 21 de Octubre de 2010.  
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,  
                                                Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,  
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,  
                                                Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,  
                                                Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,  
                                                Lv_CodigoError);  
  
IF Lv_CodigoError IS NOT NULL THEN  
    raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA  
--                    : de la tabla AD_AGENCIAS_X_CANTONES  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Viernes 26 de Noviembre de 2010.  
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,  
                                                Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,  
                                                Ln_codigo_nacion,--Pn_codigo_nacion     IN      NUMBER,  
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,  
                                                Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,  
                                                Lv_CodigoError  
                                                );  
  
IF Lv_CodigoError IS NOT NULL THEN  
    raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
```



```
--  
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la  
Factura  
--  
Lv_cod_modelo_cat := C.cod_modelo_cat;  
Lv_cod_item_cat := C.cod_item_cat;  
--  
-- Verifica SI existe dato en el codigo de la Linea  
--  
IF Lv_cod_modelo_cat IS NULL THEN  
--  
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante  
--  
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,  
                                                    C.tipo_comprobante,  
                                                    C.cod_comprobante);  
  
IF Lv_cod_producto = ' ' THEN  
    Lv_cod_producto := NULL;  
END IF;-- IF :NEW.cod_producto = ' ' THEN  
--  
--  
--  
IF Lv_cod_producto IS NOT NULL THEN  
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO  
--                    : de la tabla STOCK.PRODUCTO  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Domingo 7 de Enero de 2007.  
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,  
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,  
                                       Lv_nombre_producto,--Pv_nombre     IN OUT VARCHAR2,  
                                       Lv_CodigoError);  
  
IF Lv_CodigoError IS NOT NULL THEN
```



```
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                    Lv_cod_modelo_cat,--Pv_cod_modelo   IN      VARCHAR2,
                                    Lv_cod_item_cat,--Pv_cod_item     IN      VARCHAR2,
                                    Lv_nombre_linea,--Pv_nombre       IN OUT VARCHAR2,
                                    Lv_CodigoError);
```



```
        IF Lv_CodigoError IS NOT NULL THEN
            raise_application_error (-20003,Lv_CodigoError);
        END IF;
    END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
    END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_indice_de_morosidad indmor
SET     indmor.saldo_cartera          = indmor.saldo_cartera + C.Saldo_Del_Credito,
        indmor.valor_cobrado          = indmor.valor_ventas - indmor.saldo_cartera,
        indmor.cartera_vencida        = indmor.cartera_vencida + C.Total_Vencido
WHERE   indmor.periodo                = Ln_periodo_balance
AND     indmor.anio_mes_venta         = C.anio_venta||LPAD(C.mes_venta,2,0)
AND     indmor.codigo_agencia         = C.codigo_agencia
AND     indmor.codigo_canton         = Lv_codigo_canton
AND     indmor.codigo_provincia       = Lv_codigo_provincia
AND     indmor.codigo_region         = Ln_codigo_region
AND     indmor.codigo_politica        = C.codigo_politica
AND     indmor.cod_modelo_cat_linea    = Lv_cod_modelo_cat
AND     indmor.cod_item_cat_linea     = Lv_cod_item_cat
AND     indmor.cod_marca               = Ln_codigo_marca
AND     indmor.codigo_nacion          = Ln_codigo_nacion
AND     indmor.codigo_empresa        = C.codigo_empresa;
--
```



```
--  
--  
IF Sql%Notfound THEN  
  raise_application_error (-20003, 'No Existe el Registro.' || chr(10) ||  
    ' Compr=' || C.cod_comprobante ||  
    ' TipCompr=' || C.tipo_comprobante ||  
    ' Periodo=' || Ln_perodo_balance ||  
    ' AnioMesVta=' || C.anio_venta || LPAD(C.mes_venta, 2, 0) ||  
    ' Agencia=' || C.codigo_agencia ||  
    ' Canton=' || Lv_codigo_canton ||  
    ' Provin=' || Lv_codigo_provincia ||  
    ' Region=' || Ln_codigo_region ||  
    ' Polit=' || C.codigo_politica ||  
    ' ModelLinea=' || Lv_cod_modelo_cat ||  
    ' ItemLinea=' || Lv_cod_item_cat ||  
    ' Marca=' || Ln_codigo_marca ||  
    ' Nacion=' || Ln_codigo_nacion ||  
    ' Empresa=' || C.codigo_empresa || chr(10) ||  
    ' en la Tabla CC_INDICE_DE_MOROSIDAD.  
PK_CC_PROCESO_INDICE_MOROSIDAD.P_CARGAR_IND_MOR_CIERRE_MES. ' || chr(10) ||  
    SQLERRM  
  );  
  END IF; -- IF Sql%Notfound THEN  
END LOOP; -- FOR C IN C_balances LOOP  
END LOOP; -- FOR J IN C_tipos_carteras LOOP  
--  
--  
--  
COMMIT;  
--  
-- ***** CARGA LOS DATOS PARA EL SIGUIENTE MES DE TRABAJO  
--          PARA PODER EVALUAR DIARIAMENTE COMO ESTA EL INDICE DE MOROSIDAD  
--  
--  
-- Obtiene los Tipos de Cartera  
--  
FOR J IN C_tipos_carteras LOOP
```



```
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN    NUMBER,
                                                       J.cartera,--Pv_cartera                IN    VARCHAR2,
                                                       Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
                                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Obtiene el Anio y Mes del siguiente periodo de Trabajo
--
Ln_siguiente_periodo_trabajo := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABiertas porque los datos seran cargados mas adelante
--          Como datos definitos del cierre de mes
--
INSERT INTO cc_indice_de_morosidad
SELECT
    Ln_siguiente_periodo_trabajo,--PERIODO
    ANIO_MES_VENTA
    CODIGO_AGENCIA
    CODIGO_CANTON
    CODIGO_PROVINCIA
    CODIGO_REGION
    CODIGO_POLITICA
    COD_MODELO_CAT_LINEA
```



```

COD_ITEM_CAT_LINEA      ,
COD_MARCA               ,
CODIGO_NACION           ,
CODIGO_EMPRESA          ,
NOMBRE_AGENCIA          ,
NOMBRE_CANTON           ,
NOMBRE_PROVINCIA        ,
NOMBRE_REGION           ,
NOMBRE_LINEA            ,
NOMBRE_MARCA            ,
NOMBRE_POLITICA         ,
VALOR_VENTAS            ,
SALDO_CARTERA           ,
TOTAL_CARTERA_AL_CIERRE ,
VALOR_COBRADO           ,
CARTERA_VENCIDA         ,
0, --PORCENTAJE_PRODUCTIVIDAD      ,
0, --PORCENTAJE_INDICE_MOROSIDAD   ,
0, --PAGOS_ANTICIPADOS             ,
'A', --MES_CERRADO                  ,
FECHA_ULT_EJECUCION              ,
ADICIONADO_POR                   ,
FECHA_ADICION                     ,
MODIFICADO_POR                    ,
FECHA_MODIFICACION                ,
FROM cc_indice_de_morosidad indmor
WHERE indmor.periodo              = Ln_periodo_balance
AND indmor.mes_cerrado             = 'C'--A=MES ABIERTO, C=MES CERRADO
AND indmor.codigo_empresa = J.codigo_empresa;
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
    Pv_CodigoError := SQLERRM;
```



```
END P_CARGAR_IND_MOR_CIERRE_MES;
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_INDICE_DE_MOROSIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                  SALDO_CARTERA,
--                  TOTAL CARTERA_AL_CIERRE (Tenia_que_cobrar)
--                  VALOR_COBRADO
--                  CARTERA_VENCIDA
--
--                  Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 2 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIARIA_IND_MOR
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_saldo_comprobante    NUMBER(30,2);
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto     producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
```



```
Ln_cartera_vencida          NUMBER(30,2);
Ln_periodo_cobradores       NUMBER(6);
Ln_codigo_marca             cc_balances_mensuales.codigo_marca%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT *
  FROM   cc_balances_mensuales balmen
  WHERE  balmen.periodo_cierre = Ln_periodo_balance
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  ORDER BY balmen.anio_venta, balmen.mes_venta;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
  --                    de la tabla CC_PERIODOS_CERRADOS
  -- Autor             : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación   : Martes 1 de Junio de 2010.
  PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                         J.cartera,--Pv_cartera                        IN      VARCHAR2,
```



```
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,  
Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,  
Ld_fecha_final,--Pd_fecha_final         IN OUT DATE,  
Lv_CodigoError);  
  
IF Lv_CodigoError IS NOT NULL THEN  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
--  
--  
Ln_periodo_cobradores := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1), 'yyyymm'));  
--  
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante  
--           Como datos definitos del cierre de mes  
--  
UPDATE cc_indice_de_morosidad indmor  
  SET   indmor.saldo_cartera      = 0,  
        indmor.valor_cobrado      = 0,  
        indmor.cartera_vencida     = 0  
WHERE  indmor.periodo            = Ln_periodo_cobradores  
AND    indmor.mes_cerrado        = 'A'--A=MES ABIERTO, C=MES CERRADO  
AND    indmor.codigo_empresa    = J.codigo_empresa;  
END LOOP;-- FOR J IN C_tipos_carteras LOOP  
--  
--  
--  
COMMIT;  
--  
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado  
--  
FOR J IN C_tipos_carteras LOOP  
  Ln_codigo_nacion := null;  
  --  
  --  
  *****  
  **  
  --
```



```
-- Funcion      : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--              de la tabla AD_PARAMETROS_GENERALES
-- Autor        : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--              de la tabla CC_PERIODOS_CERRADOS
-- Autor        : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera      IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance      IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene los creditos de un determinado periodo
```



```
--
FOR C IN C_balances LOOP
  --
  --
  ****
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton    IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                    Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Viernes 26 de Noviembre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                              Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                              Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                              J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                              Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                              Lv_CodigoError
                                              );

  IF Lv_CodigoError IS NOT NULL THEN
```



```
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL SALDO DE UN DETERMINADO COMPROBANTE
--                    : de la tabla STOCK.FORMA_PAGO_COMPROBANTE
-- Autor              : Ing. Giovanni Chasifán Chicaiza
-- Fecha creación    : Jueves 20 de Septiembre de 2007.
Ln_saldo_comprobante := PK_OBTENER_FOR_PAG_COM.F_SALDO_COMPROBANTE (J.codigo_empresa,--Pn_codigo_empresa
IN   NUMBER,
--
--                    C.cod_comprobante,--Pv_cod_comprobante
IN   VARCHAR2,
--
--                    C.tipo_comprobante--Pv_tipo_comprobante
IN   VARCHAR2
--
--                    );
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
--
--                    C.tipo_comprobante,
--                    C.cod_comprobante);
--
IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
```



```
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**

--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**

--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
```



```
Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
  --                    : de la tabla TG_MODELO_ITEM
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Domingo 7 de Enero de 2007.
  PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                     Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                     Lv_cod_item_cat,--Pv_cod_item          IN      VARCHAR2,
                                     Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                     Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
  END IF;-- IF Lv_cod_producto IS NOT NULL THEN
  END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento      : Obtiene EL SALDO VENCIDO DE UN DETERMINADO COMPROBANTE
  --                    : de la tabla STOCK.FORMA_PAGO_COMPROBANTE
  -- Autor              : Ing. Giovanni Chasifán Chicaiza
  -- Fecha creación    : Martes 10 de Abril de 2007.
  Ln_cartera_vencida := PK_OBTENER_FOR_PAG_COM.F_SALDO_VENCIDO (C.codigo_empresa,--Pn_codigo_empresa      IN
NUMBER,
```



```

--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_indice_de_morosidad indmor
SET   indmor.saldo_cartera      = indmor.saldo_cartera + Ln_saldo_comprobante,
      indmor.valor_cobrado     = indmor.valor_ventas - indmor.saldo_cartera,
      indmor.cartera_vencida   = indmor.cartera_vencida + Ln_cartera_vencida
WHERE indmor.periodo          = Ln_periodo_cobradores
AND   indmor.anio_mes_venta   = C.anio_venta||LPAD(C.mes_venta,2,0)
AND   indmor.codigo_agencia   = C.codigo_agencia
AND   indmor.codigo_canton    = Lv_codigo_canton
AND   indmor.codigo_provincia = Lv_codigo_provincia
AND   indmor.codigo_region    = Ln_codigo_region
AND   indmor.codigo_politica  = C.codigo_politica
AND   indmor.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND   indmor.cod_item_cat_linea = Lv_cod_item_cat
AND   indmor.cod_marca        = Ln_codigo_marca
AND   indmor.codigo_nacion    = Ln_codigo_nacion
AND   indmor.codigo_empresa   = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  raise_application_error (-20003,'No Existe el Registro.'||chr(10)||
    ' Periodo='||Ln_periodo_cobradores||

```



```

        ' AnioMesVta='||C.anio_venta||LPAD(C.mes_venta,2,0)||
        ' Agencia='||C.codigo_agencia||
        ' Canton='||Lv_codigo_canton||
        ' Provin='||Lv_codigo_provincia||
        ' Region='||Ln_codigo_region||
        ' Politica='||C.codigo_politica||
        ' Modellinea='||Lv_cod_modelo_cat||
        ' ItemLinea='||Lv_cod_item_cat||
        ' Marca='||Ln_codigo_marca||
        ' Nacion='||Ln_codigo_nacion||
        ' Empresa='||C.codigo_empresa||chr(10)||
        ' en la Tabla CC_INDICE_DE_MOROSIDAD.
PK_CC_PROCESO_INDICE_MOROSIDAD.P_ACTUALIZACION_DIARIA_IND_MOR. '||chr(10)||
        SQLERRM
    );
    END IF;-- IF Sql%Notfound THEN
    END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
END P_ACTUALIZACION_DIARIA_IND_MOR;
END;
/
```



Anexo A34.

Script actualización índice flujo corriente.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_FLUJO_CORRIENTE IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS QUE SE VENDE EN EL MES DEL BALANCE
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_FLUJO_CORRIENTE
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Lunes 13 de Diciembre de 2010.
PROCEDURE P_CARGAR_FLUJO_CORRIENTE (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_MES_PRESENTE_TOTAL DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                    VALOR_VENCIMIENTO,
--                    SALDO_VENCIMIENTO,
--                    COBRO_VENCIMIENTO
--                    COBRO_EFECTIVO_VENCIMIENTO
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Lunes 13 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_FLU_COR;
END;
/
```



```
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_FLUJO_CORRIENTE IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS QUE SE VENCE EN EL MES DEL BALANCE
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC FLUJO CORRIENTE
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Lunes 13 de Diciembre de 2010.
PROCEDURE P_CARGAR_FLUJO_CORRIENTE (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant  cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo       NUMBER(30,2);
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig    NUMBER(30,2);
Ln_descuento            NUMBER(30,2);
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;
```



```
Lv_cod_item_cat          producto.cod_item_cat%TYPE;
Lv_nombre_linea          tg_modelo_item.nombre%TYPE;
Ln_codigo_marca          cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion     NUMBER(30,2);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2; -- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
  FROM   cc_balances_mensuales balmen,
         forma_pago_comprobante forpagcom
  WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
  AND    balmen.total_vencido = 0
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  AND    forpagcom.cod_comprobante = balmen.cod_comprobante
  AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
  AND    (forpagcom.cod_forma_pago = 'CRE'
          OR forpagcom.cod_forma_pago = 'NDB'
          OR forpagcom.cod_forma_pago = 'TCR'
          OR forpagcom.cod_forma_pago = 'BAN'
          OR forpagcom.cod_forma_pago = 'ADI')
  AND    forpagcom.vencimiento >= Ld_fecha_inicial
  AND    forpagcom.vencimiento <= Ld_fecha_final
  ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
```



```
FOR J IN C_tipos_carteras LOOP
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
  --                    de la tabla CC_PERIODOS_CERRADOS
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Martes 1 de Junio de 2010.
  PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                         J.cartera,--Pv_cartera          IN      VARCHAR2,
                                                         Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                         Id_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                         Id_fecha_final,--Pd_fecha_final      IN OUT DATE,
                                                         Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  -- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
  --          Como datos definitos del cierre de mes
  --
  DELETE cc_flujo_corriente flucor
  WHERE  flucor.periodo      = Ln_periodo_balance
  AND    flucor.mes_cerrado  = 'A'--A=MES ABIERTO, C=MES CERRADO
  AND    flucor.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
  --
  --
  --
  COMMIT;
  --
  -- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
  --
  FOR J IN C_tipos_carteras LOOP
```



```
Ln_codigo_nacion := null;
--
--
*****
**
--
-- Funcion : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
-- : de la tabla AD_PARAMETROS_GENERALES
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa IN
NUMBER
);

IF Ln_codigo_nacion IS NULL THEN
Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
-- : de la tabla CC_PERIODOS_CERRADOS
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa IN NUMBER,
J.cartera,--Pv_cartera IN VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
```



```
--  
--  
--  
Ln_codigo_empresa := J.codigo_empresa;  
--  
-- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance  
--  
Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);  
Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));  
--  
-- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance  
--  
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);  
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);  
--  
-- Obtiene los creditos de un determinado periodo  
--  
FOR C IN C_balances LOOP  
  --  
  --  
*****  
**  
  --  
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA  
  --                               de la tabla AD_AGENCIAS_X_CANTONES  
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.  
  -- Fecha creación     : Jueves 21 de Octubre de 2010.  
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,  
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,  
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,  
                                                    Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,  
                                                    Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,  
                                                    Lv_CodigoError);  
  
  IF Lv_CodigoError IS NOT NULL THEN  
    raise_application_error (-20003,Lv_CodigoError);  
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
  --
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                             J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                       C.tipo_comprobante,
                                                       C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
```



```
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat  := '0';
  Lv_CodigoError  := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
```



```
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item        IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                   Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
```



```
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
--
-- Función : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
Pn_empresa IN NUMBER,
01,'yyyy-mm-dd'), --Ld_fecha_inicial, --Pd_fecha_inicial IN DATE,
Pd_fecha_final IN DATE
);
--
--
*****
**
--
-- Función : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Jueves 3 de Junio de 2010.
```



```
Ln_Cobro_efectivo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, -
- Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa, -
- Pn_empresa              IN      NUMBER,
Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig --Pd_fecha_final      IN      DATE
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante, --
Pv tipo comprobante     IN      VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa, --Pn_empresa
IN      NUMBER,
```



```
dd'),--Ld_fecha_inicial,--Pd_fecha_inicial      IN      DATE,
IN      DATE
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante  IN      VARCHAR2,
Pv_tipo_comprobante IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
Pn_empresa          IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Pd_fecha_final      IN      DATE
Ld_fecha_final_mes_sig--
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
```



```
--
-- Función          : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN   VARCHAR2,
--                  C.Tipo_Comprobante, --
Pv tipo comprobante  IN   VARCHAR2,
--                  C.secuencia, --
Pn_numero_vencimiento IN   NUMBER,
--                  C.Codigo_Empresa, --
Pn empresa           IN   NUMBER,
--                  to_date('2000-01-
01','yyyy-mm-dd'), --Pd_fecha_inicial   IN   DATE,
--                  Ld_fecha_final--
Pd fecha final       IN   DATE
--                  );
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_flujo_corriente flucor
SET     flucor.valor_vencimiento      = flucor.valor_vencimiento + (C.valor - Ln_valor_devolucion),
        flucor.saldo_vencimiento      = flucor.saldo_vencimiento + C.saldo + Ln_Cobro_efectivo_mes_sig
+ Ln_descuento_mes_sig,
        flucor.cobro_efectivo_vencimiento = flucor.cobro_efectivo_vencimiento + Ln_Cobro_efectivo
WHERE   flucor.periodo                = Ln_periodo_balance
AND     flucor.fecha_vencimiento       = C.vencimiento
AND     flucor.codigo_agencia           = C.codigo_agencia
AND     flucor.codigo_canton           = Lv_codigo_canton
AND     flucor.codigo_provincia        = Lv_codigo_provincia
AND     flucor.codigo_region           = Ln_codigo_region
AND     flucor.codigo_politica         = C.Codigo_Politica
AND     flucor.cod_modelo_cat_linea     = Lv_cod_modelo_cat
AND     flucor.cod_item_cat_linea      = Lv_cod_item_cat
AND     flucor.cod_marca                = Ln_codigo_marca
```



```
AND flucor.cod_cat_cliente = C.cod_cat_cliente
AND flucor.es_empleado = C.es_empleado
AND flucor.codigo_nacion = Ln_codigo_nacion
AND flucor.codigo_empresa = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_flujo_corriente (PERIODO
                                FECHA VENCIMIENTO
                                CODIGO_AGENCIA
                                CODIGO_CANTON
                                CODIGO_PROVINCIA
                                CODIGO_REGION
                                CODIGO_POLITICA
                                COD_MODELO_CAT_LINEA
                                COD_ITEM_CAT_LINEA
                                COD_MARCA
                                COD_CAT_CLIENTE
                                ES_EMPLEADO
                                CODIGO_NACION
                                CODIGO_EMPRESA
                                valor_vencimiento
                                saldo_vencimiento
                                cobro_efectivo_vencimiento
                                MES_CERRADO
                                )
VALUES (Ln_periodo_balance, --PERIODO NOT NULL NUMBER(6)
        C.vencimiento, --fecha_vencimiento NOT NULL DATE
        C.codigo_agencia, --CODIGO_AGENCIA NOT NULL NUMBER(4)
        Lv_codigo_canton, --CODIGO_CANTON NOT NULL VARCHAR2(4)
        Lv_codigo_provincia, --CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
        Ln_codigo_region, --CODIGO_REGION NOT NULL NUMBER(3)
        C.Codigo_Politica, --CODIGO_POLITICA NOT NULL NUMBER(2)
        Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA NOT NULL VARCHAR2(8)
        Lv_cod_item_cat, --COD_ITEM_CAT_LINEA NOT NULL VARCHAR2(3)
        Ln_codigo_marca, --COD_MARCA NOT NULL NUMBER(3)
```



```
C.cod_cat_cliente,--COD_CAT_CLIENTE                NOT NULL VARCHAR2(8)
C.es_empleado,--ES_EMPLEADO                        NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                   NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                  NOT NULL NUMBER(2)
(C.valor - Ln_valor_devolucion),--C.valor,--valor_vencimiento            NOT
NULL NUMBER(30,2)
(C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--saldo_vencimiento
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo,--cobro_efectivo                NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                                  NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := SQLERRM;
END P_CARGAR_FLUJO_CORRIENTE;
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA cc_flujo_corriente DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--
--                    VALOR_VENCIMIENTO,
--                    SALDO_VENCIMIENTO,
--                    COBRO_VENCIMIENTO
--                    COBRO_EFECTIVO_VENCIMIENTO
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
```



```
-- Fecha creacion      : Lunes 13 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_FLU_COR
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia    ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa      ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto        cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto     producto.nombre%TYPE;
Lv_cod_modelo_cat      producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat        producto.cod_item_cat%TYPE;
Lv_nombre_linea        tg_modelo_item.nombre%TYPE;
Ln_periodo_cobradores  NUMBER(6);
Ln_codigo_marca        cc_balances_mensuales.codigo_marca%TYPE;
Ln_Cobro_efectivo      NUMBER(30,2);
Ln_periodo_mes_sig     NUMBER(6);
Ln_valor_devolucion    NUMBER(30,2);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2;
--
-- Obtiene los creditos de un determinado periodo
```



```
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
  FROM   cc_balances_mensuales balmen,
         forma_pago_comprobante forpagcom
  WHERE  balmen.periodo_cierre = Ln_periodo_balance
  AND    balmen.total_vencido  = 0
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  AND    forpagcom.cod_comprobante = balmen.cod_comprobante
  AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
  AND    forpagcom.empresa      = balmen.codigo_empresa
  AND    (forpagcom.cod_forma_pago = 'CRE'
         OR forpagcom.cod_forma_pago = 'NDB'
         OR forpagcom.cod_forma_pago = 'TCR'
         OR forpagcom.cod_forma_pago = 'BAN'
         OR forpagcom.cod_forma_pago = 'ADI')
  AND    forpagcom.vencimiento   >= Ld_fecha_inicial_mes_sig
  AND    forpagcom.vencimiento   <= Ld_fecha_final_mes_sig
  ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC PERIODOS CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa, --Pn_codigo_empresa  IN      NUMBER,
                                                       J.cartera, --Pv_cartera                IN      VARCHAR2,
                                                       Ln_periodo_balance, --Pn_periodo_balance  IN OUT NUMBER,
```



```

Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final          IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_periodo_cobradores := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--           Como datos definitos del cierre de mes
--
UPDATE cc_flujo_corriente flucor
  SET   flucor.valor_vencimiento      = 0,
        flucor.saldo_vencimiento      = 0,
        flucor.cobro_efectivo_vencimiento = 0
WHERE  flucor.periodo                = Ln_periodo_cobradores
AND    flucor.mes_cerrado              = 'A'--A=MES ABIERTO, C=MES CERRADO
AND    flucor.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  --
  --
  *****
  **
  --
  -- Funcion           : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
```



```
--                               de la tabla AD PARAMETROS GENERALES
-- Autor                         : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación                : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                                    );
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise application error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento                : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                               de la tabla CC PERIODOS CERRADOS
-- Autor                         : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación                : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial   IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);
IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Siguiente Mes del Balance
--
```



```
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
Ln_periodo_mes_sig := to_number(to_char(Ld_fecha_inicial_mes_sig,'yyyymm'));
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  ****
  **
  --
  -- Procedimiento : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  -- de la tabla AD AGENCIAS X CANTONES
  -- Autor : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia IN NUMBER,
  Ln_codigo_nacion,--Pn_codigo_nacion IN NUMBER,
  J.codigo_empresa,--Pn_codigo_empresa IN NUMBER,
  Lv_codigo_canton,--Pv_codigo_canton IN OUT VARCHAR2,
  Ln_codigo_region,--Pn_codigo_region IN OUT NUMBER,
  Lv_CodigoError);
  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  -- de la tabla AD AGENCIAS X CANTONES
  -- Autor : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación : Viernes 26 de Noviembre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia IN NUMBER,
  Ln_codigo_region,--Pn_codigo_region IN OUT NUMBER,
  Ln_codigo_nacion,--Pn_codigo_nacion IN NUMBER,
```



```
J.codigo_empresa,--Pn_codigo_empresa    IN    NUMBER,  
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,  
Lv_CodigoError  
);  
  
IF Lv_CodigoError IS NOT NULL THEN  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la  
Factura  
--  
Lv_cod_modelo_cat := C.cod_modelo_cat;  
Lv_cod_item_cat := C.cod_item_cat;  
--  
-- Verifica SI existe dato en el codigo de la Linea  
--  
IF Lv_cod_modelo_cat IS NULL THEN  
  --  
  -- Obtiene el Código del Producto de Mayor Valor de un Comprobante  
  --  
  Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,  
                                                       C.tipo_comprobante,  
                                                       C.cod_comprobante);  
  
  IF Lv_cod_producto = ' ' THEN  
    Lv_cod_producto := NULL;  
  END IF;-- IF :NEW.cod_producto = ' ' THEN  
  --  
  --  
  --  
  IF Lv_cod_producto IS NOT NULL THEN  
    --  
    --  
    *****  
    **  
  
    --  
    -- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO  
    --                    de la tabla STOCK.PRODUCTO
```



```
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
  --                    : de la tabla STOCK.PRODUCTO
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **

  --
  -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
  --                    : de la tabla TG_MODELO_ITEM
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación     : Domingo 7 de Enero de 2007.
```



```
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                     Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                     Lv_cod_item_cat,--Pv_cod_item        IN      VARCHAR2,
                                     Lv_nombre_linea,--Pv_nombre          IN OUT VARCHAR2,
                                     Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Jueves 3 de Junio de 2010.
```



```
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,
Pn_numero_vencimiento  IN      NUMBER,
Pn_empresa             IN      NUMBER,
Ld_fecha_inicial_mes_sig, -- Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig -- Pd_fecha_final      IN      DATE
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
Pv tipo comprobante     IN      VARCHAR2,
Pn_numero_vencimiento  IN      NUMBER,
Pn_empresa             IN      NUMBER,
C.Tipo_Comprobante, --
C.secuencia, --
C.Codigo_Empresa, --
```



```
01','yyyy-mm-dd'),--Pd_fecha_inicial      IN      DATE,                                to_date('2000-01-
Pd_fecha_final      IN      DATE                                Ld_fecha_final_mes_sig--
                                                                );
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_flujo_corriente flucor
SET     flucor.valor_vencimiento      = flucor.valor_vencimiento + (C.valor - Ln_valor_devolucion),
        flucor.saldo_vencimiento      = flucor.saldo_vencimiento + C.saldo,
        flucor.cobro_efectivo_vencimiento = flucor.cobro_efectivo_vencimiento + Ln_Cobro_efectivo
WHERE   flucor.periodo                = Ln_periodo_mes_sig
AND     flucor.fecha_vencimiento      = C.vencimiento
AND     flucor.codigo_agencia         = C.codigo_agencia
AND     flucor.codigo_canton         = Lv_codigo_canton
AND     flucor.codigo_provincia       = Lv_codigo_provincia
AND     flucor.codigo_region         = Ln_codigo_region
AND     flucor.codigo_politica       = C.Codigo_Politica
AND     flucor.cod_modelo_cat_linea   = Lv_cod_modelo_cat
AND     flucor.cod_item_cat_linea     = Lv_cod_item_cat
AND     flucor.cod_marca              = Ln_codigo_marca
AND     flucor.cod_cat_cliente       = C.cod_cat_cliente
AND     flucor.es_empleado           = C.es_empleado
AND     flucor.codigo_nacion         = Ln_codigo_nacion
AND     flucor.codigo_empresa       = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_flujo_corriente (PERIODO, FECHA_VENCIMIENTO, CODIGO_AGENCIA, CODIGO_CANTON, CODIGO_PROVINCIA, CODIGO_REGION, CODIGO_POLITICA,
```



```

COD_MODELO_CAT_LINEA
COD_ITEM_CAT_LINEA
COD_MARCA
COD_CAT_CLIENTE
ES_EMPLEADO
CODIGO_NACION
CODIGO_EMPRESA
valor_vencimiento
saldo_vencimiento
cobro efectivo_vencimiento
MES_CERRADO
)
VALUES (Ln_periodo_mes_sig,--PERIODO
C.vencimiento,--fecha_vencimiento
C.codigo_agencia,--CODIGO_AGENCIA
Lv_codigo_canton,--CODIGO_CANTON
Lv_codigo_provincia,--CODIGO_PROVINCIA
Ln_codigo_region,--CODIGO_REGION
C.codigo_politica,--CODIGO_REGION
Lv_cod_modelo_cat,--COD_MODELO_CAT_LINEA
Lv_cod_item_cat,--COD_ITEM_CAT_LINEA
Ln_codigo_marca,--COD_MARCA
C.cod_cat_cliente,--COD_CAT_CLIENTE
C.es_empleado,--ES_EMPLEADO
Ln_codigo_nacion,--CODIGO_NACION
C.codigo_empresa,--CODIGO_EMPRESA
(C.valor - Ln_valor_devolucion),--C.valor,--valor_vencimiento
C.saldo,--saldo_vencimiento
Ln_Cobro_efectivo,--cobro_efectivo
'A'--MES_CERRADO
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
```



```
--  
  COMMIT;  
END P_ACTUALIZACION_FLU_COR;  
END;  
/
```



Anexo A35.

Script actualización índice mes presente puntual.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_MES_PRESENTE_PUN IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC_PRODUC_MES_PRESENTE_PUNTUAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
PROCEDURE P_CARGAR_MES_PRESENTE_PUNTUAL (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_MES_PRESENTE_TOTAL DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                    VALOR_VENCIMIENTO,
--                    SALDO_VENCIMIENTO,
--                    COBRO_VENCIMIENTO
--                    COBRO_EFECTIVO_VENCIMIENTO
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIA_MES_PRE_PU;
END;
/
```



```
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_MES_PRESENTE_PUN IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                    : SIEMPRE Y CUANDO EL CREDITO NO TENGA CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC PRODUC MES_PRESENTE_PUNTUAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Jueves 9 de Diciembre de 2010.
PROCEDURE P_CARGAR_MES_PRESENTE_PUNTUAL (Pv_CodigoError IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance     cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo     NUMBER(30,2);
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig  NUMBER(30,2);
Ln_descuento          NUMBER(30,2);
Ln_codigo_nacion      ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton      ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region      ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia   ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa     ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto       cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto    producto.nombre%TYPE;
Lv_cod_modelo_cat     producto.cod_modelo_cat%TYPE;
```



```
Lv_cod_item_cat          producto.cod_item_cat%TYPE;
Lv_nombre_linea          tg_modelo_item.nombre%TYPE;
Ln_codigo_marca          cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion      NUMBER(30,2);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2; -- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
  FROM   cc_balances_mensuales balmen,
         forma_pago_comprobante forpagcom
  WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
  AND    balmen.total_vencido = 0
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  AND    forpagcom.cod_comprobante = balmen.cod_comprobante
  AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
  AND    forpagcom.secuencia <> 1
  AND    forpagcom.empresa = balmen.codigo_empresa
  AND    (forpagcom.cod_forma_pago = 'CRE'
  OR     forpagcom.cod_forma_pago = 'NDB'
  OR     forpagcom.cod_forma_pago = 'TCR'
  OR     forpagcom.cod_forma_pago = 'BAN'
  OR     forpagcom.cod_forma_pago = 'ADI')
  AND    forpagcom.vencimiento >= Ld_fecha_inicial
  AND    forpagcom.vencimiento <= Ld_fecha_final
  ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
```



```
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
  --                    de la tabla CC_PERIODOS_CERRADOS
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Martes 1 de Junio de 2010.
  PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                         J.cartera,--Pv_cartera          IN      VARCHAR2,
                                                         Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                         Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                         Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
                                                         Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  -- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
  --                    Como datos definitos del cierre de mes
  --
  DELETE cc_produc_mes_presente_puntual promesprepun
  WHERE  promesprepun.periodo          = Ln_periodo_balance
  AND    promesprepun.mes_cerrado      = 'A'--A=MES ABIERTO, C=MES CERRADO
  AND    promesprepun.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
```



```
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
--
--
*****
**
--
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                  : de la tabla AD PARAMETROS GENERALES
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                    );
IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento    : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                  : de la tabla CC_PERIODOS_CERRADOS
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial   IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
```



```
raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance
--
Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);
Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));
--
-- Obtiene la Fecha Inicial y Final del Siguiente Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Jueves 21 de Octubre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                Lv_codigo_canton,--Pv_codigo_canton      IN OUT VARCHAR2,
                                                Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                Lv_CodigoError);
IF Lv_CodigoError IS NOT NULL THEN
raise_application_error (-20003,Lv_CodigoError);
```



```
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    : de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                             J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                       C.tipo_comprobante,
                                                       C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
```



```
Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
```



```

                                                                    Lv_CodigoError);
IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
  --                    : de la tabla TG MODELO_ITEM
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Domingo 7 de Enero de 2007.
  PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                     Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                     Lv_cod_item_cat,--Pv_cod_item         IN      VARCHAR2,
                                     Lv_nombre_linea,--Pv_nombre           IN OUT VARCHAR2,
                                     Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
  END IF;-- IF Lv_cod_producto IS NOT NULL THEN
  END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
  --
  -- Verifica si el codigo de la Marca es NULL
  --
  Ln_codigo_marca := C.codigo_marca;
  IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
  END IF;-- IF C.codigo_marca IS NULL THEN

```



```
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
Pn_empresa IN NUMBER,
01, 'yyyy-mm-dd'), -- Ld_fecha_inicial, -- Pd_fecha_inicial IN DATE,
Pd_fecha_final IN DATE
C.secuencia, --
C.Codigo_Empresa, --
to_date('2000-01-
Ld_fecha_final --
);
--
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
```



```
-- Fecha creación      : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, -
- Pv_cod_comprobante   IN      VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante   IN      VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa, -
- Pn_empresa           IN      NUMBER,
Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial   IN      DATE,
Ld_fecha_final_mes_sig --Pd_fecha_final     IN      DATE
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función      : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--
--              de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN      VARCHAR2,
C.Tipo_Comprobante, --
Pv_tipo_comprobante IN      VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa, --Pn_empresa
IN      NUMBER,
```



```
dd'),--Ld_fecha_inicial,--Pd_fecha_inicial      IN      DATE,
IN      DATE
--
--
*****
**
--
-- Función      : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--              de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor        : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante      IN      VARCHAR2,
Pv_tipo_comprobante     IN      VARCHAR2,
Pn_numero_vencimiento   IN      NUMBER,
Pn_empresa              IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
Pd_fecha_final          IN      DATE
Ld_fecha_final_mes_sig--
);
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
```



```
--  
-- Función : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,  
VENCIMIENTO Y RANGO DE FECHAS  
-- de la tabla ST_CANCELACION_D y ST_CANCELACION  
-- Autor : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación : Viernes 28 de Enero de 2011.  
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --  
Pv_cod_comprobante IN VARCHAR2, -- C.Tipo_Comprobante, --  
Pv tipo comprobante IN VARCHAR2, -- C.secuencia, --  
Pn_numero_vencimiento IN NUMBER, -- C.Codigo_Empresa, --  
Pn empresa IN NUMBER, -- to_date('2000-01-  
01','yyyy-mm-dd'), --Pd_fecha_inicial IN DATE, -- Ld_fecha_final--  
Pd fecha final IN DATE -- ) ;  
  
--  
-- Actualiza el Indice de Morosidad  
--  
UPDATE cc_produc_mes_presente_puntual promesprepun  
SET promesprepun.valor_vencimiento = promesprepun.valor_vencimiento + (C.valor -  
Ln_valor_devolucion),  
promesprepun.saldo_vencimiento = promesprepun.saldo_vencimiento + C.saldo +  
Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig,  
promesprepun.cobro_efectivo_vencimiento = promesprepun.cobro_efectivo_vencimiento +  
Ln_Cobro_efectivo - Ln_descuento  
WHERE promesprepun.periodo = Ln_periodo_balance  
AND promesprepun.fecha_vencimiento = C.vencimiento  
AND promesprepun.codigo_agencia = C.codigo_agencia  
AND promesprepun.codigo_canton = Lv_codigo_canton  
AND promesprepun.codigo_provincia = Lv_codigo_provincia  
AND promesprepun.codigo_region = Ln_codigo_region  
AND promesprepun.codigo_politica = C.codigo_politica  
AND promesprepun.cod_modelo_cat_linea = Lv_cod_modelo_cat
```



```
AND promesprepun.cod_item_cat_linea = Lv_cod_item_cat
AND promesprepun.cod_marca = Ln_codigo_marca
AND promesprepun.cod_cat_cliente = C.cod_cat_cliente
AND promesprepun.es_empleado = C.es_empleado
AND promesprepun.codigo_nacion = Ln_codigo_nacion
AND promesprepun.codigo_empresa = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_produc_mes_presente_puntual (PERIODO ,
      FECHA_VENCIMIENTO ,
      CODIGO_AGENCIA ,
      CODIGO_CANTON ,
      CODIGO_PROVINCIA ,
      CODIGO_REGION ,
      CODIGO_POLITICA ,
      COD_MODELO_CAT_LINEA ,
      COD_ITEM_CAT_LINEA ,
      COD_MARCA ,
      COD_CAT_CLIENTE ,
      ES_EMPLEADO ,
      CODIGO_NACION ,
      CODIGO_EMPRESA ,
      valor_vencimiento ,
      saldo_vencimiento ,
      cobro_efectivo_vencimiento ,
      MES_CERRADO ,
      )
VALUES (Ln_periodo_balance, --PERIODO NOT NULL NUMBER(6)
      C.vencimiento, --fecha_vencimiento NOT NULL DATE
      C.codigo_agencia, --CODIGO_AGENCIA NOT NULL NUMBER(4)
      Lv_codigo_canton, --CODIGO_CANTON NOT NULL VARCHAR2(4)
      Lv_codigo_provincia, --CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
      Ln_codigo_region, --CODIGO_REGION NOT NULL NUMBER(3)
      C.codigo_politica, --CODIGO_POLITICA NOT NULL NUMBER(2)
      Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA NOT NULL VARCHAR2(8)
```



```

Lv_cod_item_cat,--COD_ITEM_CAT_LINEA                NOT NULL VARCHAR2(3)
Ln_codigo_marca,--COD_MARCA                          NOT NULL NUMBER(3)
C.cod_cat_cliente,--COD_CAT_CLIENTE                  NOT NULL VARCHAR2(8)
C.es_empleado,--ES_EMPLEADO                          NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                     NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                    NOT NULL NUMBER(2)
(C.valor - Ln_valor_devolucion),--C.valor,--valor_vencimiento      NOT
NULL NUMBER(30,2)
(C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--saldo_vencimiento
NOT NULL NUMBER(30,2)
(Ln_Cobro_efectivo - Ln_descuento),--cobro_efectivo              NOT NULL NUMBER(30,2)
'C'--MES_CERRADO          NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := SQLERRM;
END P_CARGAR_MES_PRESENTE_PUNTUAL;
--
--
*****
**
--
-- Procedimiento : ACTUALIZA DIARIAMENTE LA TABLA cc_produc_mes_presente_puntual DEL PERIODO ABIERTO LOS
SIGUIENTES CAMPOS:
--
-- VALOR VENCIMIENTO,
-- SALDO VENCIMIENTO,
-- COBRO VENCIMIENTO
-- COBRO_EFECTIVO_VENCIMIENTO
--
-- Obtiene los comprobantes del ultimo cierre de Balance
```



```
--
-- Autor           : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion  : Jueves 9 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIA_MES_PRE_PU
IS
  --
  -- Variables Locales
  --
  Lv_CodigoError          VARCHAR2(3000);
  Ln_periodes_balance     cc_periodes_cerrados.periodes_cierre%TYPE;
  Ld_fecha_inicial        cc_periodes_cerrados.fecha_inicial%TYPE;
  Ld_fecha_final          cc_periodes_cerrados.fecha_final%TYPE;
  Ld_fecha_inicial_mes_sig cc_periodes_cerrados.fecha_inicial%TYPE;
  Ld_fecha_final_mes_sig  cc_periodes_cerrados.fecha_final%TYPE;
  Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
  Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
  Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
  Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
  Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
  Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
  Lv_nombre_producto      producto.nombre%TYPE;
  Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;
  Lv_cod_item_cat          producto.cod_item_cat%TYPE;
  Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
  Ln_periodes_cobradores  NUMBER(6);
  Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
  Ln_Cobro_efectivo       NUMBER(30,2);
  Ln_periodes_mes_sig     NUMBER(6);
  Ln_descuento            NUMBER(30,2);
  Ln_valor_devolucion     NUMBER(30,2);
  --
  -- Obtiene los Tipos de Cartera
  --
  CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
```



```
AND tipcar.codigo_empresa = 2;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
FROM cc_balances_mensuales balmen,
forma_pago_comprobante forpagcom
WHERE balmen.periodo_cierre = Ln_periodo_balance
AND balmen.total_vencido = 0
AND balmen.codigo_empresa = Ln_codigo_empresa
AND forpagcom.cod_comprobante = balmen.cod_comprobante
AND forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND forpagcom.secuencia <> 1
AND forpagcom.empresa = balmen.codigo_empresa
AND (forpagcom.cod_forma_pago = 'CRE'
OR forpagcom.cod_forma_pago = 'NDB'
OR forpagcom.cod_forma_pago = 'TCR'
OR forpagcom.cod_forma_pago = 'BAN'
OR forpagcom.cod_forma_pago = 'ADI')
AND forpagcom.vencimiento >= Ld_fecha_inicial_mes_sig
AND forpagcom.vencimiento <= Ld_fecha_final_mes_sig
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA de la tabla CC PERIODOS CERRADOS
--
-- Autor : Ing. Giovanni Chasifán Chicaiza.
```



```
-- Fecha creación      : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera      IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_periodo_cobradores := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
-- Como datos definitos del cierre de mes
--
UPDATE cc_produc_mes_presente_puntual promesprepun
  SET   promesprepun.valor_vencimiento      = 0,
        promesprepun.saldo_vencimiento      = 0,
        promesprepun.cobro_efectivo_vencimiento = 0
WHERE  promesprepun.periodo                = Ln_periodo_cobradores
AND    promesprepun.mes_cerrado            = 'A'--A=MES ABIERTO, C=MES CERRADO
AND    promesprepun.codigo_empresa        = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
--
```



```
--
*****
**
--
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                  : de la tabla AD_PARAMETROS_GENERALES
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                    );

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento    : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                  : de la tabla CC_PERIODOS_CERRADOS
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
```



```
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
Ln_periodo_mes_sig := to_number(to_char(Ld_fecha_inicial_mes_sig,'yyyymm'));
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  ****
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanny Chasifán Chicaiza.
  -- Fecha creación    : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,
                                                    Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    de la tabla AD_AGENCIAS_X_CANTONES
```



```
-- Autor          : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia  IN    NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region  IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion  IN    NUMBER,
                                             J.codigo_empresa,--Pn_codigo_empresa  IN    NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
  --
  -- Obtiene el Código del Producto de Mayor Valor de un Comprobante
  --
  Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                         C.tipo_comprobante,
                                                         C.cod_comprobante);

  IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
  END IF;-- IF :NEW.cod_producto = ' ' THEN
  --
  --
  --
  IF Lv_cod_producto IS NOT NULL THEN
    --
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
```



```
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM  
--                    de la tabla TG_MODELO_ITEM  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Domingo 7 de Enero de 2007.  
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,  
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,  
                                   Lv_cod_item_cat,--Pv_cod_item        IN      VARCHAR2,  
                                   Lv_nombre_linea,--Pv_nombre          IN OUT VARCHAR2,  
                                   Lv_CodigoError);  
  
IF Lv_CodigoError IS NOT NULL THEN  
    raise_application_error (-20003,Lv_CodigoError);  
END IF;  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
ELSE-- IF Lv_cod_producto IS NOT NULL THEN  
    Lv_cod_modelo_cat := '0';  
    Lv_cod_item_cat := '0';  
END IF;-- IF Lv_cod_producto IS NOT NULL THEN  
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN  
--  
-- Verifica si el codigo de la Marca es NULL  
--  
Ln_codigo_marca := C.codigo_marca;  
IF C.codigo_marca IS NULL THEN  
    Ln_codigo_marca := 0;  
END IF;-- IF C.codigo_marca IS NULL THEN  
--  
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado  
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a  
nivel de Oracle  
--  
--
```



```
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln Cobro efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN      VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante   IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
Pn_empresa           IN      NUMBER,
01', 'yyyy-mm-dd'), -- Ld_fecha_inicial_mes_sig, -- Pd_fecha_inicial           IN      DATE,
Ld_fecha_final_mes_sig -- Pd_fecha_final           IN      DATE
);
--
--
*****
**
--
-- Función           : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Viernes 6 de Agosto de 2010.
Ln descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN      VARCHAR2,
Pv_tipo_comprobante IN      VARCHAR2,
C.Tipo_Comprobante, --
```



```
Pn_numero_vencimiento IN NUMBER,
IN NUMBER,
dd'),--Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial IN DATE,
Pd_fecha_final IN DATE
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
--
-- Función : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
Pn_empresa IN NUMBER,
01','yyyy-mm-dd'),--Pd_fecha_inicial IN DATE,
Pd_fecha_final IN DATE
);
```



```
--  
-- Actualiza el Indice de Morosidad  
--  
UPDATE cc_produc_mes_presente_puntual promesprepun  
SET     promesprepun.valor_vencimiento      = promesprepun.valor_vencimiento + (C.valor -  
Ln_valor_devolucion),  
        promesprepun.saldo_vencimiento      = promesprepun.saldo_vencimiento + C.saldo,  
        promesprepun.cobro_efectivo_vencimiento = promesprepun.cobro_efectivo_vencimiento +  
Ln_Cobro_efectivo - Ln_descuento  
WHERE   promesprepun.periodo                = Ln periodo mes_sig  
AND     promesprepun.fecha_vencimiento      = C.vencimiento  
AND     promesprepun.codigo_agencia         = C.codigo_agencia  
AND     promesprepun.codigo_canton         = Lv_codigo_canton  
AND     promesprepun.codigo_provincia      = Lv_codigo_provincia  
AND     promesprepun.codigo_region         = Ln_codigo_region  
AND     promesprepun.codigo_politica       = C.codigo_politica  
AND     promesprepun.cod_modelo_cat_linea   = Lv_cod_modelo_cat  
AND     promesprepun.cod_item_cat_linea    = Lv_cod_item_cat  
AND     promesprepun.cod_marca              = Ln_codigo_marca  
AND     promesprepun.cod_cat_cliente       = C.cod_cat_cliente  
AND     promesprepun.es_empleado           = C.es_empleado  
AND     promesprepun.codigo_nacion         = Ln_codigo_nacion  
AND     promesprepun.codigo_empresa       = C.codigo_empresa;  
--  
--  
--  
IF Sql%Notfound THEN  
INSERT INTO cc_produc_mes_presente_puntual (PERIODO  
                                             FECHA_VENCIMIENTO  
                                             CODIGO_AGENCIA  
                                             CODIGO_CANTON  
                                             CODIGO_PROVINCIA  
                                             CODIGO_REGION  
                                             CODIGO_POLITICA  
                                             COD_MODELLO_CAT_LINEA  
                                             COD_ITEM_CAT_LINEA  
                                             COD_MARCA
```



```

COD_CAT_CLIENTE
ES_EMPLEADO
CODIGO_NACION
CODIGO_EMPRESA
valor_vencimiento
saldo_vencimiento
cobro_efectivo_vencimiento
MES_CERRADO
)
VALUES (Ln periodo mes sig, --PERIODO
C.vencimiento, --fecha_vencimiento
C.codigo_agencia, --CODIGO_AGENCIA
Lv_codigo_canton, --CODIGO_CANTON
Lv_codigo_provincia, --CODIGO_PROVINCIA
Ln_codigo_region, --CODIGO_REGION
C.codigo_politica, --CODIGO_POLITICA
Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA
Lv_cod_item_cat, --COD_ITEM_CAT_LINEA
Ln_codigo_marca, --COD_MARCA
C.cod_cat_cliente, --COD_CAT_CLIENTE
C.es_empleado, --ES_EMPLEADO
Ln_codigo_nacion, --CODIGO_NACION
C.codigo_empresa, --CODIGO_EMPRESA
(C.valor - Ln_valor_devolucion), --C.valor, --valor_vencimiento
C.saldo, --saldo_vencimiento
Ln_Cobro_efectivo, --cobro_efectivo
'A' --MES_CERRADO
);
END IF; -- IF Sql%Notfound THEN
END LOOP; -- FOR C IN C_balances LOOP
END LOOP; -- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
END P_ACTUALIZACION_DIA_MES_PRE_PU;
```



END;
/



Anexo A36.

Script actualización índice mes presente total.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_MES_PRESENTE_TOT IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                     SIN VALIDAR SI EL CREDITO TIENE CARTERA VENCIDA.
--                     Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                     Inserta en la Tabla CC_PRODUC_MES_PRESENTE_TOTAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
PROCEDURE P_CARGAR_MES_PRESENTE_TOTAL (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_MES_PRESENTE_TOTAL DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--                     VALOR_VENCIMIENTO,
--                     SALDO_VENCIMIENTO,
--                     COBRO_VENCIMIENTO
--                     COBRO_EFECTIVO_VENCIMIENTO
--
--                     Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 9 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIA_MES_PRE_TO;
END;
/
```



```
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_MES_PRESENTE_TOT IS
--
--
*****
**
--
-- Procedimiento      : Carga los VENCIMIENTOS DEL MES DEL CIERRE DEL BALANCE QUE DEBE SER COBRADOS
--                    : SIN VALIDAR SI EL CREDITO TIENE CARTERA VENCIDA.
--                    : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                    : Inserta en la Tabla CC PRODUC MES_PRESENTE_TOTAL
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Jueves 9 de Diciembre de 2010.
PROCEDURE P_CARGAR_MES_PRESENTE_TOTAL (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance     cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo      NUMBER(30,2);
Ln_Cobro_efectivo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig   NUMBER(30,2);
Ln_descuento           NUMBER(30,2);
Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia    ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa      ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto        cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto     producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
```



```
Lv_cod_item_cat          producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2; -- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
  FROM   cc_balances_mensuales balmen,
        forma_pago_comprobante forpagcom
  WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  AND    forpagcom.cod_comprobante = balmen.cod_comprobante
  AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
  AND    forpagcom.empresa = balmen.codigo_empresa
  AND    (forpagcom.cod_forma_pago = 'CRE'
  OR     forpagcom.cod_forma_pago = 'NDB'
  OR     forpagcom.cod_forma_pago = 'TCR'
  OR     forpagcom.cod_forma_pago = 'BAN'
  OR     forpagcom.cod_forma_pago = 'ADI')
  AND    forpagcom.vencimiento >= Ld_fecha_inicial
  AND    forpagcom.vencimiento <= Ld_fecha_final
  ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
```



```
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                       J.cartera,--Pv_cartera                      IN      VARCHAR2,
                                                       Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
                                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--          Como datos definitos del cierre de mes
--
DELETE cc_produc_mes_presente_total promespretot
WHERE  promespretot.periodo          = Ln_periodo_balance
AND    promespretot.mes_cerrado      = 'A'--A=MES ABIERTO, C=MES CERRADO
AND    promespretot.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
    Ln_codigo_nación := null;
```



```
--
--
*****
**
--
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                  : de la tabla AD_PARAMETROS_GENERALES
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                    );

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento    : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                  : de la tabla CC_PERIODOS_CERRADOS
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
                                                    J.cartera,--Pv_cartera          IN      VARCHAR2,
                                                    Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                    Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
                                                    Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
                                                    Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
```



```
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance
--
Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);
Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));
--
-- Obtiene la Fecha Inicial y Final del Siguiete Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  -- de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación     : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,
                                                    Lv_CodigoError);
  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                             J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
  --
  -- Obtiene el Código del Producto de Mayor Valor de un Comprobante
  --
  Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                         C.tipo_comprobante,
                                                         C.cod_comprobante);

  IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
  END IF;-- IF :NEW.cod_producto = ' ' THEN
```



```
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat  := '0';
  Lv_CodigoError  := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
```



```
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item          IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                   Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
```



```
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln Cobro efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante IN VARCHAR2,
Pn_numero_vencimiento IN NUMBER,
Pn_empresa IN NUMBER,
Pd_fecha_inicial IN DATE,
Pd_fecha_final IN DATE,
C.secuencia, --
C.Codigo_Empresa, --
Ld_fecha_inicial, --
Ld_fecha_final --
);
--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln Cobro efectivo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, -
- Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante IN VARCHAR2,
```



```
Pn_numero_vencimiento IN NUMBER,
- Pn_empresa IN NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial IN DATE,
Ld_fecha_final_mes_sig--Pd_fecha_final IN DATE
);
--
--
*****
**
--
-- Función : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2,
C.Tipo_Comprobante, --
Pv tipo comprobante IN VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN NUMBER,
C.Codigo_Empresa, --Pn_empresa
IN NUMBER,
Ld_fecha_inicial, --
Pd_fecha_inicial IN DATE,
Ld_fecha_final--Pd_fecha_final
IN DATE
);
--
--
*****
**
--
```



```
-- Función : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
-- de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN VARCHAR2, --
Pv_tipo_comprobante IN VARCHAR2, --
Pn_numero_vencimiento IN NUMBER, --
Pn_empresa IN NUMBER, --
C.Tipo_Comprobante, --
C.secuencia, --
C.Codigo_Empresa, --
Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial IN DATE, --
Pd_fecha_final IN DATE --
Ld_fecha_final_mes_sig--
);
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_produc_mes_presente_total promespretot
SET promespretot.valor_vencimiento = promespretot.valor_vencimiento + C.valor,
promespretot.saldo_vencimiento = promespretot.saldo_vencimiento + C.saldo +
Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig,
promespretot.cobro_efectivo_vencimiento = promespretot.cobro_efectivo_vencimiento +
Ln_Cobro_efectivo
WHERE promespretot.periodo = Ln_periodo_balance
AND promespretot.fecha_vencimiento = C.vencimiento
AND promespretot.codigo_agencia = C.codigo_agencia
AND promespretot.codigo_canton = Lv_codigo_canton
AND promespretot.codigo_provincia = Lv_codigo_provincia
AND promespretot.codigo_region = Ln_codigo_region
AND promespretot.codigo_politica = C.codigo_politica
AND promespretot.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND promespretot.cod_item_cat_linea = Lv_cod_item_cat
AND promespretot.cod_marca = Ln_codigo_marca
```



```
AND      promespretot.cod_cat_cliente      = C.cod_cat_cliente
AND      promespretot.es_empleado         = C.es_empleado
AND      promespretot.codigo_nacion      = Ln_codigo_nacion
AND      promespretot.codigo_empresa     = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_produc_mes_presente_total (PERIODO
      FECHA_VENCIMIENTO
      CODIGO_AGENCIA
      CODIGO_CANTON
      CODIGO_PROVINCIA
      CODIGO_REGION
      CODIGO_POLITICA
      COD_MODELO_CAT_LINEA
      COD_ITEM_CAT_LINEA
      COD_MARCA
      COD_CAT_CLIENTE
      ES_EMPLEADO
      CODIGO_NACION
      CODIGO_EMPRESA
      valor_vencimiento
      saldo_vencimiento
      cobro_efectivo_vencimiento
      MES_CERRADO
  )
VALUES (Ln_periodo_balance, --PERIODO          NOT NULL NUMBER(6)
      C.vencimiento, --fecha_vencimiento     NOT NULL DATE
      C.codigo_agencia, --CODIGO_AGENCIA   NOT NULL NUMBER(4)
      Lv_codigo_canton, --CODIGO_CANTON    NOT NULL VARCHAR2(4)
      Lv_codigo_provincia, --CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
      Ln_codigo_region, --CODIGO_REGION    NOT NULL NUMBER(3)
      C.codigo_politica, --CODIGO_POLITICA  NOT NULL NUMBER(2)
      Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA NOT NULL VARCHAR2(8)
      Lv_cod_item_cat, --COD_ITEM_CAT_LINEA NOT NULL VARCHAR2(3)
      Ln_codigo_marca, --COD_MARCA        NOT NULL NUMBER(3)
```



```

C.cod_cat_cliente,--COD_CAT_CLIENTE                                NOT NULL VARCHAR2(8)
C.es_empleado,--ES_EMPLEADO                                       NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                                   NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                                 NOT NULL NUMBER(2)
C.valor,--valor_vencimiento                                       NOT NULL NUMBER(30,2)
(C.saldo + Ln_Cobro_efectivo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--saldo_vencimiento
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo,--cobro_efectivo                               NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                                               NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := SQLERRM;
END P_CARGAR_MES_PRESENTE_TOTAL;
--
--
*****
**
--
-- Procedimiento          : ACTUALIZA DIARIAMENTE LA TABLA cc_produc_mes_presente_total DEL PERIODO ABIERTO LOS
SIGUIENTES CAMPOS:
--
--                          VALOR_VENCIMIENTO,
--                          SALDO_VENCIMIENTO,
--                          COBRO_VENCIMIENTO
--                          COBRO_EFECTIVO_VENCIMIENTO
--
--                          Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor                   : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion          : Jueves 9 de Diciembre de 2010.
```



```
PROCEDURE P_ACTUALIZACION_DIA_MES_PRE_TO
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial       cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final         cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion       ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton       ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region       ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia    ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa     ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto       cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto     producto.nombre%TYPE;
Lv_cod_modelo_cat      producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat       producto.cod_item_cat%TYPE;
Lv_nombre_linea       tg_modelo_item.nombre%TYPE;
Ln_periodo_cobradores  NUMBER(6);
Ln_codigo_marca       cc_balances_mensuales.codigo_marca%TYPE;
Ln_Cobro_efectivo     NUMBER(30,2);
Ln_periodo_mes_sig     NUMBER(6);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM   cc_tipos_carteras tipcar
WHERE  tipcar.estado = 'N'
AND    tipcar.codigo_empresa = 2;
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
```



```
SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
FROM   cc_balances_mensuales balmen,
       forma_pago_comprobante forpagcom
WHERE  balmen.periodo_cierre = Ln_periodo_balance
AND    balmen.codigo_empresa = Ln_codigo_empresa
AND    forpagcom.cod_comprobante = balmen.cod_comprobante
AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
AND    forpagcom.empresa = balmen.codigo_empresa
AND    (forpagcom.cod_forma_pago = 'CRE'
        OR forpagcom.cod_forma_pago = 'NDB'
        OR forpagcom.cod_forma_pago = 'TCR'
        OR forpagcom.cod_forma_pago = 'BAN'
        OR forpagcom.cod_forma_pago = 'ADI')
AND    forpagcom.vencimiento >= Ld_fecha_inicial_mes_sig
AND    forpagcom.vencimiento <= Ld_fecha_final_mes_sig
ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa   IN      NUMBER,
                                                       J.cartera,--Pv_cartera                        IN      VARCHAR2,
                                                       Ln_periodo_balance,--Pn_periodo_balance   IN OUT NUMBER,
                                                       Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
                                                       Ld_fecha_final,--Pd_fecha_final         IN OUT DATE,
                                                       Lv_CodigoError);
```



```
IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_periodo_cobradores := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--           Como datos definitos del cierre de mes
--
UPDATE cc_produc_mes_presente_total promespretot
  SET   promespretot.valor_vencimiento      = 0,
        promespretot.saldo_vencimiento     = 0,
        promespretot.cobro_efectivo_vencimiento = 0
WHERE  promespretot.periodo                = Ln_periodo_cobradores
AND    promespretot.mes_cerrado            = 'A'--A=MES ABIERTO, C=MES CERRADO
AND    promespretot.codigo_empresa        = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  --
  --
  *****
  **
  --
  -- Funcion           : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
  --                   : de la tabla AD_PARAMETROS_GENERALES
  -- Autor            : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación   : Martes 19 de Octubre de 2010.
```



```
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                      de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera      IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance      IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
Ln_periodo_mes_sig := to_number(to_char(Ld_fecha_inicial_mes_sig,'yyyymm'));
```



```
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton    IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                    Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Viernes 26 de Noviembre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                               Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                               Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                               J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                               Lv_codigo_provincia,--Pv_codigo_provincia  IN OUT VARCHAR2,
                                               Lv_CodigoError
```



```
);  
IF Lv_CodigoError IS NOT NULL THEN  
  raise_application_error (-20003,Lv_CodigoError);  
END IF;-- IF Lv_CodigoError IS NOT NULL THEN  
--  
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la  
Factura  
--  
Lv_cod_modelo_cat := C.cod_modelo_cat;  
Lv_cod_item_cat := C.cod_item_cat;  
--  
-- Verifica SI existe dato en el codigo de la Linea  
--  
IF Lv_cod_modelo_cat IS NULL THEN  
--  
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante  
--  
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,  
                                                    C.tipo_comprobante,  
                                                    C.cod_comprobante);  
  
IF Lv_cod_producto = ' ' THEN  
  Lv_cod_producto := NULL;  
END IF;-- IF :NEW.cod_producto = ' ' THEN  
--  
--  
--  
IF Lv_cod_producto IS NOT NULL THEN  
--  
--  
*****  
**  
  
--  
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO  
--                    : de la tabla STOCK.PRODUCTO  
-- Autor              : Ing. Giovanni Chasifán Chicaiza.  
-- Fecha creación    : Domingo 7 de Enero de 2007.  
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
```



```

Lv_cod_producto,--Pv_cod_producto  IN    VARCHAR2,
Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
  --                    : de la tabla STOCK.PRODUCTO
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Domingo 7 de Enero de 2007.
  PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN    NUMBER,
                                             Lv_cod_producto,--Pv_cod_producto  IN    VARCHAR2,
                                             Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                             Lv_cod_item_cat,--Pv_cod_item_cat  IN OUT VARCHAR2,
                                             Lv_CodigoError);

  IF Lv_CodigoError IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
  ELSE-- IF Lv_CodigoError IS NOT NULL THEN
    --
    --
    ****
    **
    --
    -- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
    --                    : de la tabla TG MODELO ITEM
    -- Autor              : Ing. Giovanni Chasifán Chicaiza.
    -- Fecha creación    : Domingo 7 de Enero de 2007.
    PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN    NUMBER,
                                       Lv_cod_modelo_cat,--Pv_cod_modelo      IN    VARCHAR2,
                                       Lv_cod_item_cat,--Pv_cod_item          IN    VARCHAR2,
```



```

                                Lv_nombre_linea,--Pv_nombre           IN OUT VARCHAR2,
                                Lv_CodigoError);
    IF Lv_CodigoError IS NOT NULL THEN
        raise_application_error (-20003,Lv_CodigoError);
    END IF;
    END IF;-- IF Lv_CodigoError IS NOT NULL THEN
    END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
    END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor             : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante,--
Pv_cod_comprobante   IN      VARCHAR2,
C.Tipo_Comprobante,-- Pv_tipo_comprobante   IN      VARCHAR2,
C.secuencia,--
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa,--
Pn_empresa           IN      NUMBER,
```



```
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,

Ld_fecha_final_mes_sig--Pd_fecha_final          IN      DATE

--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_produc_mes_presente_total promespretot
SET     promespretot.valor_vencimiento          = promespretot.valor_vencimiento + C.valor,
        promespretot.saldo_vencimiento         = promespretot.saldo_vencimiento + C.saldo,
        promespretot.cobro_efectivo_vencimiento = promespretot.cobro_efectivo_vencimiento +

Ln_Cobro_efectivo
WHERE   promespretot.periodo                    = Ln_periodo_mes_sig
AND     promespretot.fecha_vencimiento         = C.vencimiento
AND     promespretot.codigo_agencia            = C.codigo_agencia
AND     promespretot.codigo_canton            = Lv_codigo_canton
AND     promespretot.codigo_provincia          = Lv_codigo_provincia
AND     promespretot.codigo_region            = Ln_codigo_region
AND     promespretot.codigo_politica          = C.codigo_politica
AND     promespretot.cod_modelo_cat_linea      = Lv_cod_modelo_cat
AND     promespretot.cod_item_cat_linea       = Lv_cod_item_cat
AND     promespretot.cod_marca                 = Ln_codigo_marca
AND     promespretot.cod_cat_cliente          = C.cod_cat_cliente
AND     promespretot.es_empleado              = C.es_empleado
AND     promespretot.codigo_nacion            = Ln_codigo_nacion
AND     promespretot.codigo_empresa           = C.codigo_empresa;

--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_produc_mes_presente_total (PERIODO
                                           FECHA_VENCIMIENTO
                                           CODIGO_AGENCIA
                                           CODIGO_CANTON
                                           CODIGO_PROVINCIA
                                           CODIGO_REGION
```



```

CODIGO_POLITICA
COD_MODELO_CAT_LINEA
COD_ITEM_CAT_LINEA
COD_MARCA
COD_CAT_CLIENTE
ES_EMPLEADO
CODIGO_NACION
CODIGO_EMPRESA
valor_vencimiento
saldo_vencimiento
cobro_efectivo_vencimiento
MES_CERRADO
)
VALUES (Ln_periodo_mes_sig,--PERIODO
C.vencimiento,--fecha_vencimiento
C.codigo_agencia,--CÓDIGO_AGENCIA
Lv_codigo_canton,--CODIGO_CANTON
Lv_codigo_provincia,--CODIGO_PROVINCIA
Ln_codigo_region,--CODIGO_REGION
C.codigo_politica,--CODIGO_POLITICA
Lv_cod_modelo_cat,--COD_MODELO_CAT_LINEA
Lv_cod_item_cat,--COD_ITEM_CAT_LINEA
Ln_codigo_marca,--COD_MARCA
C.cod_cat_cliente,--CÓD_CAT_CLIENTE
C.es_empleado,--ES_EMPLEADO
Ln_codigo_nacion,--CODIGO_NACION
C.codigo_empresa,--CODIGO_EMPRESA
C.valor,--valor_vencimiento
C.saldo,--saldo_vencimiento
Ln_Cobro_efectivo,--cobro_efectivo
'A'--MES_CERRADO
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
```



```
--  
  COMMIT;  
END P_ACTUALIZACION_DIA_MES_PRE_TO;  
END;  
/
```



Anexo A37.

Script actualización índice primeras cuotas.

```
CREATE OR REPLACE PACKAGE PK_CC_PROCESO_PRIMERAS_CUOTAS IS
--
--
*****
**
--
-- Procedimiento      : Carga las PRIMERAS CUOTAS, DONDE SECUENCIA = 1 Y LA FECHA DE VENCIMIENTO SEA DEL MES DEL
CIERRE DEL BALANCE
--
--                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                   : Inserta en la Tabla CC_PRIMERAS_CUOTAS
-- Autor             : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Miercoles 1 de Diciembre de 2010.
PROCEDURE P_CARGAR_PRI_CUO_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2);
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_INDICE_DE_MOROSIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--
--                   VALOR_PRIMERAS_CUOTAS,
--                   SALDO_PRIMERAS_CUOTAS
--                   COBRO_EFECTIVO_PRI_CUO
--
--                   : Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor             : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion    : Jueves 2 de Diciembre de 2010.
PROCEDURE P_ACTUALIZACION_DIARIA_PRI_CUO;
END;
/
CREATE OR REPLACE PACKAGE BODY PK_CC_PROCESO_PRIMERAS_CUOTAS IS
```



```
--
--
*****
**
--
-- Procedimiento      : Carga las PRIMERAS CUOTAS, DONDE SECUENCIA = 1 Y LA FECHA DE VENCIMIENTO SEA DEL MES DEL
CIERRE DEL BALANCE
--                   : Este siempre se debe ejecutar despues del proceso de Cierre de Mes
--                   : Inserta en la Tabla CC_PRIMERAS_CUOTAS
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Miercoles 1 de Diciembre de 2010.
PROCEDURE P_CARGAR_PRI_CUO_CIERRE_MES (Pv_CodigoError  IN OUT VARCHAR2)
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ln_periodo_balance_ant  cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_ant cc_periodos_cerrados.fecha_inicial%TYPE;
Ln_Cobro_efectivo_pri_cuo NUMBER(30,2);
Ln_Cob_efe_pri_cuo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig    NUMBER(30,2);
Ln_descuento            NUMBER(30,2);
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat       producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat         producto.cod_item_cat%TYPE;
```



```
Lv_nombre_linea          tg_modelo_item.nombre%TYPE;
Ln_codigo_marca          cc_balances_mensuales.codigo_marca%TYPE;
Ln_valor_devolucion     NUMBER(30,2);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
  SELECT *
  FROM   cc_tipos_carteras tipcar
  WHERE  tipcar.estado = 'N'
  AND    tipcar.codigo_empresa = 2; -- La empresa 6 falta definir las agencias a que cantones corresponden, razon
por la cual se ubica siempre al empresa 2 Jaher
--
-- Obtiene los creditos de un determinado periodo
--
CURSOR C_balances IS
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia
  FROM   cc_balances_mensuales balmen,
        forma_pago_comprobante forpagcom
  WHERE  balmen.periodo_cierre = Ln_periodo_balance_ant
  AND    balmen.codigo_empresa = Ln_codigo_empresa
  AND    forpagcom.cod_comprobante = balmen.cod_comprobante
  AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante
  AND    forpagcom.empresa = balmen.codigo_empresa
  AND    forpagcom.secuencia = 1
  AND    (forpagcom.cod_forma_pago = 'CRE'
  OR     forpagcom.cod_forma_pago = 'NDB'
  OR     forpagcom.cod_forma_pago = 'TCR'
  OR     forpagcom.cod_forma_pago = 'BAN'
  OR     forpagcom.cod_forma_pago = 'ADI')
  AND    forpagcom.vencimiento >= Ld_fecha_inicial
  AND    forpagcom.vencimiento <= Ld_fecha_final
  ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;
BEGIN
--
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto
--
```



```
FOR J IN C_tipos_carteras LOOP
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                    de la tabla CC_PERIODOS_CERRADOS
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                                       J.cartera,--Pv_cartera                IN      VARCHAR2,
                                                       Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
                                                       Id_fecha_inicial,--Pd_fecha_inicial    IN OUT DATE,
                                                       Id_fecha_final,--Pd_fecha_final        IN OUT DATE,
                                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
--                    Como datos definitos del cierre de mes
--
DELETE cc_primeras_cuotas pricuo
WHERE pricuo.periodo      = Ln_periodo_balance
AND   pricuo.mes_cerrado  = 'A'--A=MES ABIERTO, C=MES CERRADO
AND   pricuo.codigo_empresa = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
```



```
Ln_codigo_nacion := null;
--
--
*****
**
--
-- Funcion          : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--                  : de la tabla AD_PARAMETROS_GENERALES
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
                                                    );

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento    : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--                  : de la tabla CC_PERIODOS_CERRADOS
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera          IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final         IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
```



```
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
-- Obtiene la Fecha Inicial y Final del Anterior Mes del Balance
--
Ld_fecha_inicial_mes_ant := add_months(trunc(Ld_fecha_inicial),-1);
Ln_periodo_balance_ant := to_number(to_char(Ld_fecha_inicial_mes_ant,'yyyymm'));
--
-- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  *****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  -- de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación     : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                                    Lv_CodigoError);
  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
```



```
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
--                    de la tabla AD_AGENCIAS_X_CANTONES
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Viernes 26 de Noviembre de 2010.
PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                             Ln_codigo_region,--Pn_codigo_region    IN OUT NUMBER,
                                             Ln_codigo_nacion,--Pn_codigo_nacion    IN      NUMBER,
                                             J.codigo_empresa,--Pn_codigo_empresa  IN      NUMBER,
                                             Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
                                             Lv_CodigoError
                                             );

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
--
-- Obtiene el Código del Producto de Mayor Valor de un Comprobante
--
Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                       C.tipo_comprobante,
                                                       C.cod_comprobante);

IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
END IF;-- IF :NEW.cod_producto = ' ' THEN
```



```
--
--
--
IF Lv_cod_producto IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre       IN OUT VARCHAR2,
                                       Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
  Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto   IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
```



```
Lv_cod_modelo_cat := '0';
Lv_cod_item_cat := '0';
Lv_CodigoError := null;
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación    : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item        IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre          IN OUT VARCHAR2,
                                   Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
    Lv_cod_modelo_cat := '0';
    Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
    Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
```



```
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_pri_cuo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, -
- Pv_cod_comprobante      IN      VARCHAR2,
C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,
Pn_numero_vencimiento IN      NUMBER,
Pn_empresa            IN      NUMBER,
01,'yyyy-mm-dd'), --Ld_fecha_inicial, --Pd_fecha_inicial      IN      DATE,
Pd_fecha_final        IN      DATE
C.secuencia, --
C.Codigo_Empresa, --
to_date('2000-01-
Ld_fecha_final--
);
--
--
*****
**
--
-- Función          : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Jueves 3 de Junio de 2010.
```



```
Ln_Cob_efe_pri_cuo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante,-- Pv_cod_comprobante      IN      VARCHAR2,

C.Tipo_Comprobante,-- Pv_tipo_comprobante      IN      VARCHAR2,

Pn_numero_vencimiento IN      NUMBER,

- Pn_empresa          IN      NUMBER,

Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,

Ld_fecha_final_mes_sig--Pd_fecha_final      IN      DATE

);

--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante,--
Pv_cod_comprobante      IN      VARCHAR2,

C.Tipo_Comprobante,--
Pv tipo comprobante      IN      VARCHAR2,

C.secuencia,--
Pn_numero_vencimiento IN      NUMBER,

C.Codigo_Empresa,--Pn_empresa
IN      NUMBER,
```



```
dd'),--Ld_fecha_inicial,--Pd_fecha_inicial      IN      DATE,
                                                    to_date('2000-01-01','yyyy-mm-
                                                    Ld_fecha_final--Pd_fecha_final
                                                    );
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN      VARCHAR2,
                                                    C.Tipo_Comprobante, --
Pv_tipo_comprobante  IN      VARCHAR2,
                                                    C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
                                                    C.Codigo_Empresa, --
Pn_empresa           IN      NUMBER,
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,
                                                    Ld_fecha_final_mes_sig--
Pd_fecha_final       IN      DATE
                                                    );
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
*****
**
```



```
--
-- Función          : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN   VARCHAR2,
--                  C.Tipo_Comprobante, --
Pv tipo comprobante  IN   VARCHAR2,
--                  C.secuencia, --
Pn_numero_vencimiento IN   NUMBER,
--                  C.Codigo_Empresa, --
Pn empresa           IN   NUMBER,
--                  to_date('2000-01-
01','yyyy-mm-dd'), --Pd_fecha_inicial   IN   DATE,
--                  Ld_fecha_final--
Pd fecha final       IN   DATE
--                  );
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_primeras_cuotas pricuo
SET     pricuo.valor_primeras_cuotas = pricuo.valor_primeras_cuotas + (C.valor - Ln_valor_devolucion),
        pricuo.saldo_primeras_cuotas = pricuo.saldo_primeras_cuotas + (C.saldo +
Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig),
        pricuo.cobro_efectivo_pri_cuo = pricuo.cobro_efectivo_pri_cuo + Ln_Cobro_efectivo_pri_cuo
WHERE   pricuo.periodo = Ln_periodo_balance
AND     pricuo.fecha_vencimiento_pri_cuo = C.vencimiento
AND     pricuo.codigo_agencia = C.codigo_agencia
AND     pricuo.codigo_canton = Lv_codigo_canton
AND     pricuo.codigo_provincia = Lv_codigo_provincia
AND     pricuo.codigo_region = Ln_codigo_region
AND     pricuo.codigo_politica = C.codigo_politica
AND     pricuo.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND     pricuo.cod_item_cat_linea = Lv_cod_item_cat
AND     pricuo.cod_marca = Ln_codigo_marca
```



```
AND      pricuo.cod_cat_cliente          = C.cod_cat_cliente
AND      pricuo.es_empleado             = C.es_empleado
AND      pricuo.codigo_nacion           = Ln_codigo_nacion
AND      pricuo.codigo_empresa          = C.codigo_empresa;
--
--
--
IF Sql%Notfound THEN
  INSERT INTO cc_primeras_cuotas (PERIODO
                                FECHA_VENCIMIENTO_PRI_CUO
                                CODIGO_AGENCIA
                                CODIGO_CANTON
                                CODIGO_PROVINCIA
                                CODIGO_REGION
                                CODIGO_POLITICA
                                COD_MODELO_CAT_LINEA
                                COD_ITEM_CAT_LINEA
                                COD_MARCA
                                COD_CAT_CLIENTE
                                ES_EMPLEADO
                                CODIGO_NACION
                                CODIGO_EMPRESA
                                VALOR_PRIMERAS_CUOTAS
                                SALDO_PRIMERAS_CUOTAS
                                COBRO_EFECTIVO_PRI_CUO
                                MES_CERRADO
                                )
VALUES (Ln_periodo_balance, --PERIODO          NOT NULL NUMBER(6)
        C.vencimiento, --FECHA_VENCIMIENTO_PRI_CUO NOT NULL DATE
        C.codigo_agencia, --CODIGO_AGENCIA      NOT NULL NUMBER(4)
        Lv_codigo_canton, --CODIGO_CANTON      NOT NULL VARCHAR2(4)
        Lv_codigo_provincia, --CODIGO_PROVINCIA NOT NULL VARCHAR2(6)
        Ln_codigo_region, --CODIGO_REGION      NOT NULL NUMBER(3)
        C.codigo_politica, --CODIGO_POLITICA  NOT NULL NUMBER(2)
        Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA NOT NULL VARCHAR2(8)
        Lv_cod_item_cat, --COD_ITEM_CAT_LINEA  NOT NULL VARCHAR2(3)
        Ln_codigo_marca, --COD_MARCA          NOT NULL NUMBER(3)
```



```
C.cod_cat_cliente,--COD_CAT_CLIENTE                NOT NULL VARCHAR2(8)
C.es_empleado,--ES_EMPLEADO                        NOT NULL VARCHAR2(100)
Ln_codigo_nacion,--CODIGO_NACION                   NOT NULL NUMBER(3)
C.codigo_empresa,--CODIGO_EMPRESA                  NOT NULL NUMBER(2)
(C.valor - Ln_valor_devolucion),--C.valor,--VALOR_PRIMERAS_CUOTAS          NOT
NULL NUMBER(30,2)
(C.saldo + Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig),--C.saldo,--SALDO_PRIMERAS_CUOTAS
NOT NULL NUMBER(30,2)
Ln_Cobro_efectivo_pri_cuo,--COBRO_EFECTIVO_PRI_CUO          NOT NULL NUMBER(30,2)
'C'--MES_CERRADO                NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
EXCEPTION
WHEN OTHERS THEN
Pv_CodigoError := SQLERRM;
END P_CARGAR_PRI_CUO_CIERRE_MES;
--
--
*****
**
--
-- Procedimiento      : ACTUALIZA DIARIAMENTE LA TABLA CC_INDICE_DE_MOROSIDAD DEL PERIODO ABIERTO LOS SIGUIENTES
CAMPOS:
--
--                    VALOR_PRIMERAS_CUOTAS,
--                    SALDO_PRIMERAS_CUOTAS
--                    COBRO_EFECTIVO_PRI_CUO
--
--                    Obtiene los comprobantes del ultimo cierre de Balance
--
-- Autor              : Ing. Giovanni Chasifan Chicaiza.
-- Fecha creacion     : Jueves 2 de Diciembre de 2010.
```



```
PROCEDURE P_ACTUALIZACION_DIARIA_PRI_CUO
IS
--
-- Variables Locales
--
Lv_CodigoError          VARCHAR2(3000);
Ln_periodo_balance      cc_periodos_cerrados.periodo_cierre%TYPE;
Ld_fecha_inicial        cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final          cc_periodos_cerrados.fecha_final%TYPE;
Ld_fecha_inicial_mes_sig cc_periodos_cerrados.fecha_inicial%TYPE;
Ld_fecha_final_mes_sig  cc_periodos_cerrados.fecha_final%TYPE;
Ln_codigo_nacion        ad_naciones.codigo_nacion%TYPE;
Lv_codigo_canton        ad_agencias_x_cantones.codigo_canton%TYPE;
Ln_codigo_region        ad_agencias_x_cantones.codigo_region%TYPE;
Lv_codigo_provincia     ad_cantones_x_provincias.codigo_provincia%TYPE;
Ln_codigo_empresa       ad_empresas.codigo_empresa%TYPE;
Lv_cod_producto         cc_balances_mensuales.cod_producto%TYPE;
Lv_nombre_producto      producto.nombre%TYPE;
Lv_cod_modelo_cat        producto.cod_modelo_cat%TYPE;
Lv_cod_item_cat          producto.cod_item_cat%TYPE;
Lv_nombre_linea         tg_modelo_item.nombre%TYPE;
Ln_periodo_cobradores   NUMBER(6);
Ln_codigo_marca         cc_balances_mensuales.codigo_marca%TYPE;
Ln_Cobro_efectivo_pri_cuo NUMBER(30,2);
Ln_periodo_mes_sig      NUMBER(6);
Ln_Cob_efe_pri_cuo_mes_sig NUMBER(30,2);
Ln_descuento_mes_sig    NUMBER(30,2);
Ln_descuento            NUMBER(30,2);
Ln_valor_devolucion     NUMBER(30,2);
--
-- Obtiene los Tipos de Cartera
--
CURSOR C_tipos_carteras IS
SELECT *
FROM   cc_tipos_carteras tipcar
WHERE  tipcar.estado = 'N'
AND    tipcar.codigo_empresa = 2;
```



```
--  
-- Obtiene los creditos de un determinado periodo  
--  
CURSOR C_balances IS  
  SELECT balmen.*, forpagcom.valor, forpagcom.saldo, forpagcom.vencimiento, forpagcom.secuencia  
  FROM   cc_balances_mensuales balmen,  
         forma_pago_comprobante forpagcom  
 WHERE  balmen.periodo_cierre = Ln_periodo_balance  
 AND    balmen.codigo_empresa = Ln_codigo_empresa  
 AND    forpagcom.cod_comprobante = balmen.cod_comprobante  
 AND    forpagcom.tipo_comprobante = balmen.tipo_comprobante  
 AND    forpagcom.empresa = balmen.codigo_empresa  
 AND    forpagcom.secuencia = 1  
 AND    (forpagcom.cod_forma_pago = 'CRE'  
        OR forpagcom.cod_forma_pago = 'NDB'  
        OR forpagcom.cod_forma_pago = 'TCR'  
        OR forpagcom.cod_forma_pago = 'BAN'  
        OR forpagcom.cod_forma_pago = 'ADI')  
 AND    forpagcom.vencimiento >= Ld_fecha_inicial_mes_sig  
 AND    forpagcom.vencimiento <= Ld_fecha_final_mes_sig  
 ORDER BY forpagcom.vencimiento, balmen.cod_comprobante, balmen.tipo_comprobante;  
BEGIN  
--  
-- Obtiene los Tipos de Cartera para borrar las transacciones del periodo abierto  
--  
FOR J IN C_tipos_carteras LOOP  
  --  
  --  
  *****  
  **  
  --  
  -- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y  
CARTERA  
  --                    de la tabla CC_PERIODOS_CERRADOS  
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.  
  -- Fecha creación     : Martes 1 de Junio de 2010.  
  PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa, --Pn_codigo_empresa IN NUMBER,
```



```
J.cartera,--Pv_cartera          IN    VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance  IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial     IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final        IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_periodo_cobradores := to_number(to_char(add_months(trunc(Ld_fecha_inicial),1),'yyyymm'));
--
-- ***** Borra las Transacciones ABIertas porque los datos seran cargados mas adelante
-- Como datos definitos del cierre de mes
--
UPDATE cc_primeras_cuotas pricuo
  SET   pricuo.valor_primeras_cuotas      = 0,
        pricuo.saldo_primeras_cuotas     = 0,
        pricuo.cobro_efectivo_pri_cuo    = 0
 WHERE pricuo.periodo                    = Ln_periodo_cobradores
 AND   pricuo.mes_cerrado                 = 'A'--A=MES ABIERTO, C=MES CERRADO
 AND   pricuo.codigo_empresa             = J.codigo_empresa;
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
--
-- Obtiene los Tipos de Cartera Para cargar los datos definitos del mes cerrado
--
FOR J IN C_tipos_carteras LOOP
  Ln_codigo_nacion := null;
  --
  --
  *****
**
```



```
--
-- Funcion      : Obtiene EL CODIGO DE LA NACION DE LA EMPRESA
--              de la tabla AD_PARAMETROS_GENERALES
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 19 de Octubre de 2010.
Ln_codigo_nacion := PK_OBTENER_AD_PAR_GEN.F_CODIGO_NACION (J.codigo_empresa--Pn_codigo_empresa      IN
NUMBER
);

IF Ln_codigo_nacion IS NULL THEN
  Lv_CodigoError := 'No Existe la Empresa en la Tabla AD_PARAMETROS_GENERALES.
Empresa='||to_char(J.codigo_empresa);
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Ln_codigo_nacion IS NULL THEN
--
--
*****
**
--
-- Procedimiento      : Obtiene EL ULTIMO PERIODO Y LAS FECHAS DEL BALANCE DE UNA DETERMINADA EMPRESA Y
CARTERA
--              de la tabla CC_PERIODOS_CERRADOS
-- Autor       : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación : Martes 1 de Junio de 2010.
PK_OBTENER_CC_PER_CER.P_ULT_CIERRE_X_CART_EMPR_ROWNU (J.codigo_empresa,--Pn_codigo_empresa      IN      NUMBER,
J.cartera,--Pv_cartera      IN      VARCHAR2,
Ln_periodo_balance,--Pn_periodo_balance      IN OUT NUMBER,
Ld_fecha_inicial,--Pd_fecha_inicial      IN OUT DATE,
Ld_fecha_final,--Pd_fecha_final      IN OUT DATE,
Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
--
--
Ln_codigo_empresa := J.codigo_empresa;
--
```



```
-- Obtiene la Fecha Inicial y Final del Siguinte Mes del Balance
--
Ld_fecha_inicial_mes_sig := add_months(trunc(Ld_fecha_inicial),1);
Ld_fecha_final_mes_sig := add_months(trunc(Ld_fecha_final),1);
Ln_periodo_mes_sig := to_number(to_char(Ld_fecha_inicial_mes_sig,'yyyymm'));
--
-- Obtiene los creditos de un determinado periodo
--
FOR C IN C_balances LOOP
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Jueves 21 de Octubre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_CANTON_REGION (C.codigo_agencia,--Pn_codigo_agencia      IN      NUMBER,
                                                    Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
                                                    J.codigo_empresa,--Pn_codigo_empresa    IN      NUMBER,
                                                    Lv_codigo_canton,--Pv_codigo_canton     IN OUT VARCHAR2,
                                                    Ln_codigo_region,--Pn_codigo_region     IN OUT NUMBER,
                                                    Lv_CodigoError);
  IF Lv_CodigoError IS NOT NULL THEN
    raise_application_error (-20003,Lv_CodigoError);
  END IF;-- IF Lv_CodigoError IS NOT NULL THEN
  --
  --
  ****
  **
  --
  -- Procedimiento      : Obtiene EL CODIGO DEL CANTON Y LA REGION EN FUNCION DE LA AGENCIA, NACION Y EMPRESA
  --                    : de la tabla AD_AGENCIAS_X_CANTONES
  -- Autor              : Ing. Giovanni Chasifán Chicaiza.
  -- Fecha creación    : Viernes 26 de Noviembre de 2010.
  PK_OBTENER_AD_AGE_X_CAN.P_CODIGO_PROVINCIA (C.codigo_agencia,--Pn_codigo_agencia    IN      NUMBER,
```



```
Ln_codigo_region,--Pn_codigo_region      IN OUT NUMBER,
Ln_codigo_nacion,--Pn_codigo_nacion      IN      NUMBER,
J.codigo_empresa,--Pn_codigo_empresa     IN      NUMBER,
Lv_codigo_provincia,--Pv_codigo_provincia IN OUT VARCHAR2,
Lv_CodigoError
);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
--
-- Verifica Si el codigo de la Linea es NULO, vuelve a Buscar la linea del Producto de Mayor Valor de la
Factura
--
Lv_cod_modelo_cat := C.cod_modelo_cat;
Lv_cod_item_cat := C.cod_item_cat;
--
-- Verifica SI existe dato en el codigo de la Linea
--
IF Lv_cod_modelo_cat IS NULL THEN
  --
  -- Obtiene el Código del Producto de Mayor Valor de un Comprobante
  --
  Lv_cod_producto := ksa_balance_cartera.producto_valor(C.codigo_empresa,
                                                         C.tipo_comprobante,
                                                         C.cod_comprobante);

  IF Lv_cod_producto = ' ' THEN
    Lv_cod_producto := NULL;
  END IF;-- IF :NEW.cod_producto = ' ' THEN
  --
  --
  --
  IF Lv_cod_producto IS NOT NULL THEN
    --
    --
    *****
    **
    --
```



```
-- Procedimiento      : Obtiene EL NOMBRE DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_NOMBRE_PRODUCTO(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                       Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                       Lv_nombre_producto,--Pv_nombre      IN OUT VARCHAR2,
                                       Lv_CodigoError);
```

```
IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
```

```
*****
**
```

```
--
-- Procedimiento      : Obtiene EL CODIGO DEL MODELO E ITEM DEL PRODUCTO
--                    : de la tabla STOCK.PRODUCTO
-- Autor              : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación     : Domingo 7 de Enero de 2007.
PK_OBTENER_PRODUCTO.P_COD_MODELO_ITEM_CAT(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                           Lv_cod_producto,--Pv_cod_producto  IN      VARCHAR2,
                                           Lv_cod_modelo_cat,--Pv_cod_modelo_cat IN OUT VARCHAR2,
                                           Lv_cod_item_cat,--Pv_cod_item_cat   IN OUT VARCHAR2,
                                           Lv_CodigoError);
```

```
IF Lv_CodigoError IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
ELSE-- IF Lv_CodigoError IS NOT NULL THEN
```

```
*****
**
```

```
--
-- Procedimiento      : Obtiene EL NOMBRE DEL MODELO ITEM
--                    : de la tabla TG_MODELO_ITEM
```



```
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Domingo 7 de Enero de 2007.
PK_OBTENER_TG_MODELO_ITEM.P_NOMBRE(C.codigo_empresa,--Pn_codigo_empresa IN      NUMBER,
                                   Lv_cod_modelo_cat,--Pv_cod_modelo      IN      VARCHAR2,
                                   Lv_cod_item_cat,--Pv_cod_item         IN      VARCHAR2,
                                   Lv_nombre_linea,--Pv_nombre            IN OUT VARCHAR2,
                                   Lv_CodigoError);

IF Lv_CodigoError IS NOT NULL THEN
  raise_application_error (-20003,Lv_CodigoError);
END IF;
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
END IF;-- IF Lv_CodigoError IS NOT NULL THEN
ELSE-- IF Lv_cod_producto IS NOT NULL THEN
  Lv_cod_modelo_cat := '0';
  Lv_cod_item_cat := '0';
END IF;-- IF Lv_cod_producto IS NOT NULL THEN
END IF;-- IF Lv_cod_modelo_cat IS NULL THEN
--
-- Verifica si el codigo de la Marca es NULL
--
Ln_codigo_marca := C.codigo_marca;
IF C.codigo_marca IS NULL THEN
  Ln_codigo_marca := 0;
END IF;-- IF C.codigo_marca IS NULL THEN
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
```



```
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Jueves 3 de Junio de 2010.
Ln_Cobro_efectivo_pri_cuo := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA (C.cod_comprobante, -
- Pv_cod_comprobante      IN      VARCHAR2,

C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,

Pn_numero_vencimiento IN      NUMBER,

Pn_empresa            IN      NUMBER,

01', 'yyyy-mm-dd'), --Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial      IN      DATE,

Ld_fecha_final_mes_sig --Pd_fecha_final      IN      DATE

);

--
--
*****
**
--
-- Función           : Obtiene EL COBRO REALIZADO EN EFECTIVO DE UN DETERMINADO COMPROBANTE, VENCIMIENTO Y
RANGO DE FECHAS
--                   de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Jueves 3 de Junio de 2010.
Ln_Cob_efe_pri_cuo_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_COBRO_EFEC_X_COMP_VENC_FECHA
(C.cod_comprobante, -- Pv_cod_comprobante      IN      VARCHAR2,

C.Tipo_Comprobante, -- Pv_tipo_comprobante      IN      VARCHAR2,

Pn_numero_vencimiento IN      NUMBER,

- Pn_empresa            IN      NUMBER,
```



```
Ld_fecha_inicial_mes_sig,--Pd_fecha_inicial      IN      DATE,

Ld_fecha_final_mes_sig--Pd_fecha_final          IN      DATE

);

--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función           : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                   de la tabla ST_CANCELACION D y ST_CANCELACION
-- Autor            : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación   : Viernes 6 de Agosto de 2010.
Ln_descuento := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante IN      VARCHAR2,
C.Tipo_Comprobante, --
Pv_tipo_comprobante IN      VARCHAR2,
C.secuencia, --
Pn_numero_vencimiento IN      NUMBER,
C.Codigo_Empresa, --Pn_empresa
IN      NUMBER,
to_date('2000-01-01','yyyy-mm-
dd'),--Ld_fecha_inicial,--Pd_fecha_inicial      IN      DATE,
Ld_fecha_final_mes_sig--
Pd_fecha_final          IN      DATE
);
--
--
*****
**
```



```
--
-- Función          : Obtiene EL VALOR DEL DESCUENTO REALIZADO EN UN DETERMINADO COMPROBANTE, VENCIMIENTO
Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 6 de Agosto de 2010.
Ln_descuento_mes_sig := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DESC_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN   VARCHAR2,
--                  C.Tipo_Comprobante, --
Pv tipo comprobante  IN   VARCHAR2,
--                  C.secuencia, --
Pn_numero_vencimiento IN   NUMBER,
--                  C.Codigo_Empresa, --
Pn_empresa           IN   NUMBER,
Ld_fecha_inicial_mes_sig, --Pd_fecha_inicial      IN   DATE,
--                  Ld_fecha_final_mes_sig--
Pd fecha final       IN   DATE
--                  );
--
-- ***** Se ubica la Fecha Inicial 1 de Enero de 2000 porque puede cancelar con n meses adelantado
-- ***** Razon por la cual ubicamos esta fecha desde que comienza Jaher con las transacciones a
nivel de Oracle
--
--
--
*****
**
--
-- Función          : Obtiene EL VALOR DE LA DEVOLUCION REALIZADO EN UN DETERMINADO COMPROBANTE,
VENCIMIENTO Y RANGO DE FECHAS
--                  de la tabla ST_CANCELACION_D y ST_CANCELACION
-- Autor           : Ing. Giovanni Chasifán Chicaiza.
-- Fecha creación  : Viernes 28 de Enero de 2011.
Ln_valor_devolucion := PK_OBTENER_ST_CANCELACION_D.F_VALOR_DEVO_X_COMP_VENC_FECHA (C.cod_comprobante, --
Pv_cod_comprobante   IN   VARCHAR2,
```



```
Pv_tipo_comprobante      IN      VARCHAR2,
Pn_numero_vencimiento    IN      NUMBER,
Pn_empresa               IN      NUMBER,
01','yyyy-mm-dd'),--Pd_fecha_inicial      IN      DATE,
Pd fecha final           IN      DATE
--
-- Actualiza el Indice de Morosidad
--
UPDATE cc_primeras_cuotas pricuo
SET pricuo.valor_primeras_cuotas = pricuo.valor_primeras_cuotas + (C.valor - Ln_valor_devolucion),
    pricuo.saldo_primeras_cuotas = pricuo.saldo_primeras_cuotas + (C.saldo +
Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig),
    pricuo.cobro_efectivo_pri_cuo = pricuo.cobro_efectivo_pri_cuo + Ln_Cobro_efectivo_pri_cuo
WHERE pricuo.periodo = Ln_periodo_mes_sig
AND pricuo.fecha_vencimiento_pri_cuo = C.vencimiento
AND pricuo.codigo_agencia = C.codigo_agencia
AND pricuo.codigo_canton = Lv_codigo_canton
AND pricuo.codigo_provincia = Lv_codigo_provincia
AND pricuo.codigo_region = Ln_codigo_region
AND pricuo.codigo_politica = C.codigo_politica
AND pricuo.cod_modelo_cat_linea = Lv_cod_modelo_cat
AND pricuo.cod_item_cat_linea = Lv_cod_item_cat
AND pricuo.cod_marca = Ln_codigo_marca
AND pricuo.cod_cat_cliente = C.cod_cat_cliente
AND pricuo.es_empleado = C.es_empleado
AND pricuo.codigo_nacion = Ln_codigo_nacion
AND pricuo.codigo_empresa = C.codigo_empresa;
--
--
IF Sql%Notfound THEN
```



```
INSERT INTO cc_primeras_cuotas (PERIODO
                                FECHA_VENCIMIENTO_PRI_CUO
                                CODIGO_AGENCIA
                                CODIGO_CANTON
                                CODIGO_PROVINCIA
                                CODIGO_REGION
                                CODIGO_POLITICA
                                COD_MODELO_CAT_LINEA
                                COD_ITEM_CAT_LINEA
                                COD_MARCA
                                COD_CAT_CLIENTE
                                ES_EMPLEADO
                                CODIGO_NACION
                                CODIGO_EMPRESA
                                VALOR_PRIMERAS_CUOTAS
                                SALDO_PRIMERAS_CUOTAS
                                COBRO_EFECTIVO_PRI_CUO
                                MES_CERRADO
                                )
VALUES (Ln_periodo_mes_sig, --PERIODO
        C.vencimiento, --FECHA_VENCIMIENTO_PRI_CUO
        C.codigo_agencia, --CODIGO_AGENCIA
        Lv_codigo_canton, --CODIGO_CANTON
        Lv_codigo_provincia, --CODIGO_PROVINCIA
        Ln_codigo_region, --CODIGO_REGION
        C.codigo_politica, --CODIGO_POLITICA
        Lv_cod_modelo_cat, --COD_MODELO_CAT_LINEA
        Lv_cod_item_cat, --COD_ITEM_CAT_LINEA
        Ln_codigo_marca, --COD_MARCA
        C.cod_cat_cliente, --COD_CAT_CLIENTE
        C.es_empleado, --ES_EMPLEADO
        Ln_codigo_nacion, --CODIGO_NACION
        C.codigo_empresa, --CODIGO_EMPRESA
        (C.valor - Ln_valor_devolucion), --C.valor, --VALOR_PRIMERAS_CUOTAS
        (C.saldo + Ln_Cob_efe_pri_cuo_mes_sig + Ln_descuento_mes_sig), --C.saldo, --SALDO_PRIMERAS_CUOTAS
        )
NOT NULL NUMBER(30,2)
NOT NULL NUMBER(6)
NOT NULL DATE
NOT NULL NUMBER(4)
NOT NULL VARCHAR2(4)
NOT NULL VARCHAR2(6)
NOT NULL NUMBER(3)
NOT NULL NUMBER(2)
NOT NULL VARCHAR2(8)
NOT NULL VARCHAR2(100)
NOT NULL NUMBER(3)
NOT NULL NUMBER(2)
NOT NULL NUMBER(2)
NOT NULL VARCHAR2(8)
NOT NULL NUMBER(30,2)
```



```
Ln_Cobro_efectivo_pri_cuo,--COBRO_EFECTIVO_PRI_CUO          NOT NULL NUMBER(30,2)
'A'--MES_CERRADO          NOT NULL VARCHAR2(1)
);
END IF;-- IF Sql%Notfound THEN
END LOOP;-- FOR C IN C_balances LOOP
END LOOP;-- FOR J IN C_tipos_carteras LOOP
--
--
--
COMMIT;
END P_ACTUALIZACION_DIARIA_PRI_CUO;
END;
/
```



Anexo A38.

MANUALES DE OPERACIÓN DE LA HERRAMIENTA.

8.1 Ingreso al sistema.

Para ingresar al sistema de indicadores gerenciales en el navegador digitamos la siguiente url <http://apldmcue.jaher.com.ec:8080/apex/f?p=105:101> a continuación visualizará la siguiente pantalla de ingreso.

Login

Username

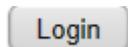
Password

Login

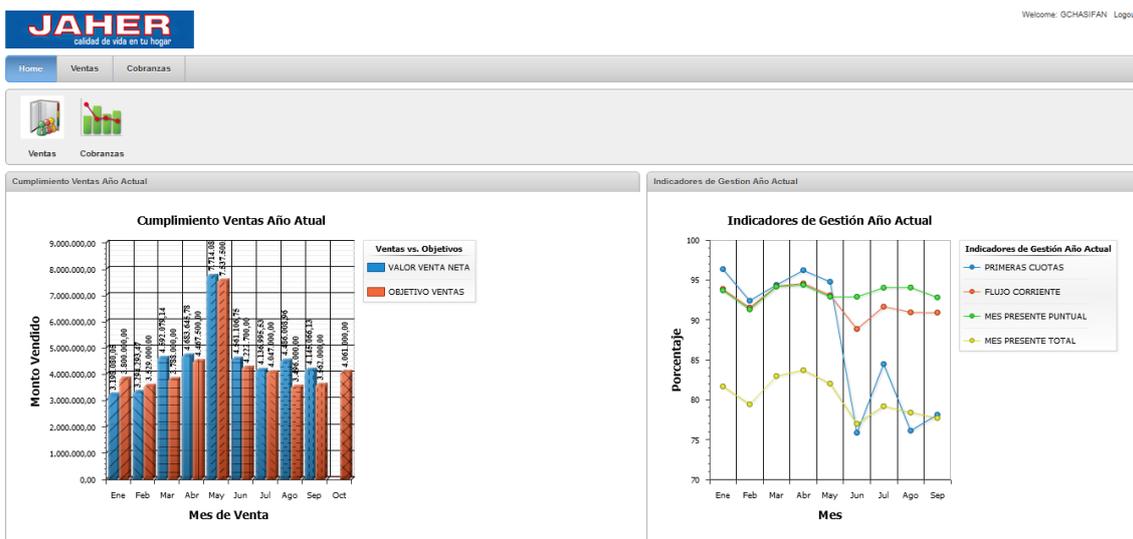
Username: ingresar el nombre de usuario asignado.

Password: ingresar la contraseña.

Para ingresar al sistema presionar la tecla **<Enter>** o dar un clic en el botón



A continuación visualizará la siguiente pantalla principal:



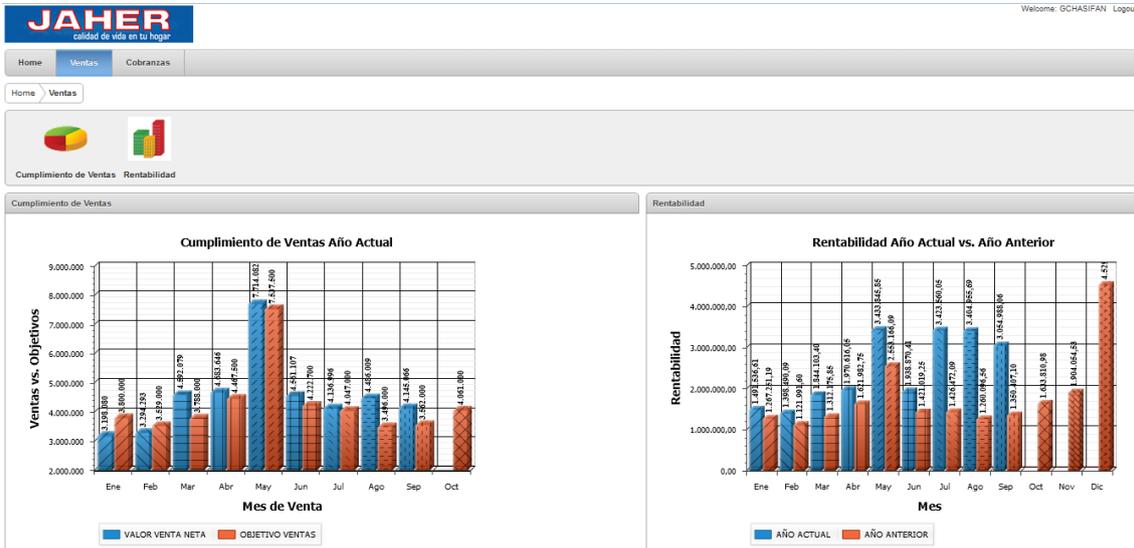
8.1.1 Cumplimiento de Ventas Año Actual: gráfico de barras indica el objetivo de ventas por cada mes del año vigente vs las ventas netas.

8.1.2 Indicadores de Gestión Año Actual: gráfico lineal de todos los indicadores de cartera del año en curso.



8.2 Ventas.

En esta opción se visualizarán los indicadores de ventas, a continuación presentará la siguiente pantalla:

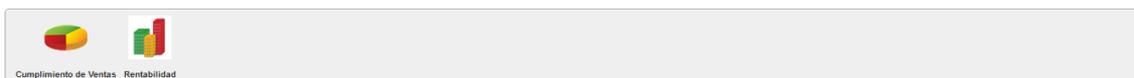


En esta pantalla se visualiza la siguiente información:

Cumplimiento de Ventas Año Actual: gráfico de barras indica el objetivo de ventas por cada mes del año vigente vs las ventas netas.

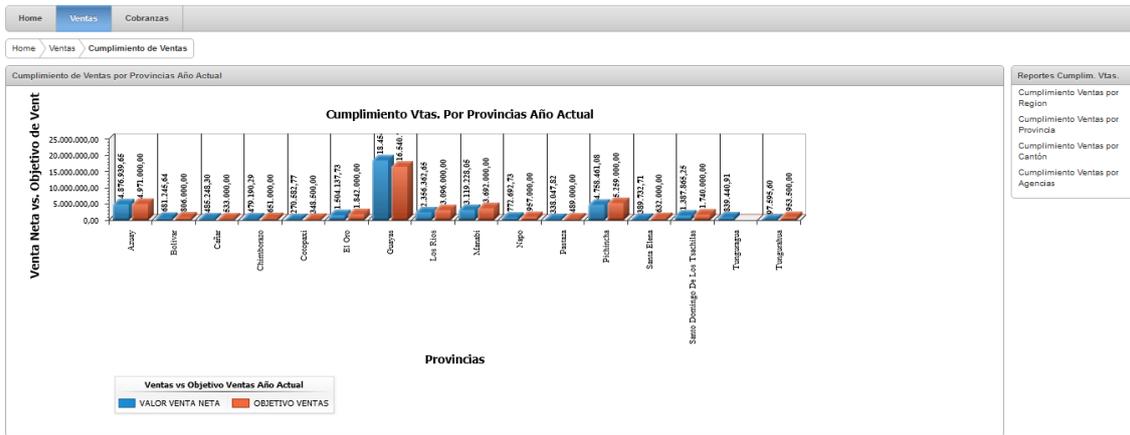
Rentabilidad Año Actual vs. Año Anterior: gráfico de barras indica la rentabilidad del año actual vs la rentabilidad del año anterior.

Menú de Ventas tiene las siguientes opciones:



8.2.1 Cumplimiento de Ventas

En esta opción se visualizará los indicadores del Cumplimiento de ventas, a continuación presentará la siguiente pantalla:

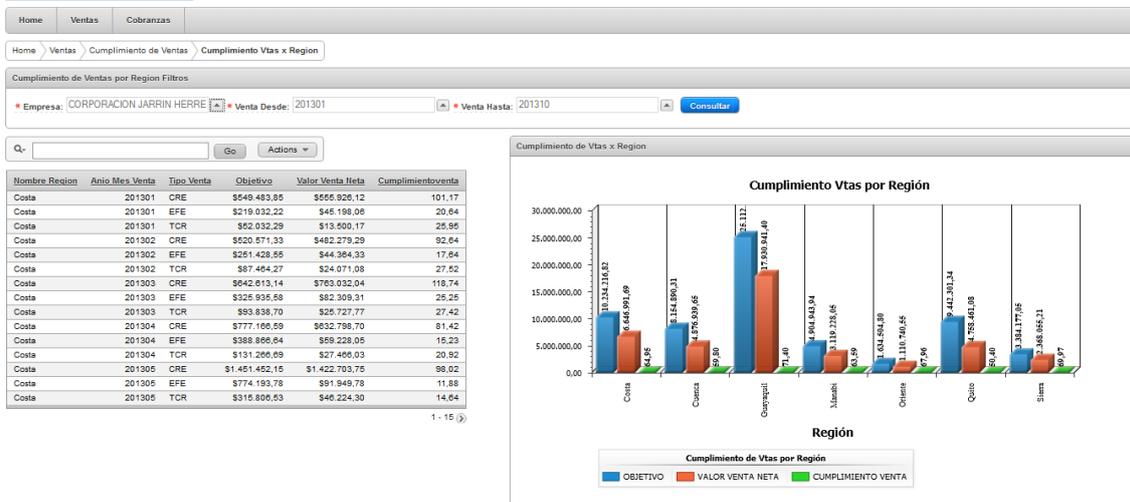


Cumplimiento Vtas. Por Provincias Año Actual: gráfico de barras indica las ventas netas vs. El objetivo de ventas del año actual por provincias.

Reportes Cumplim. Vtas.: Reporteria del cumplimiento de ventas.

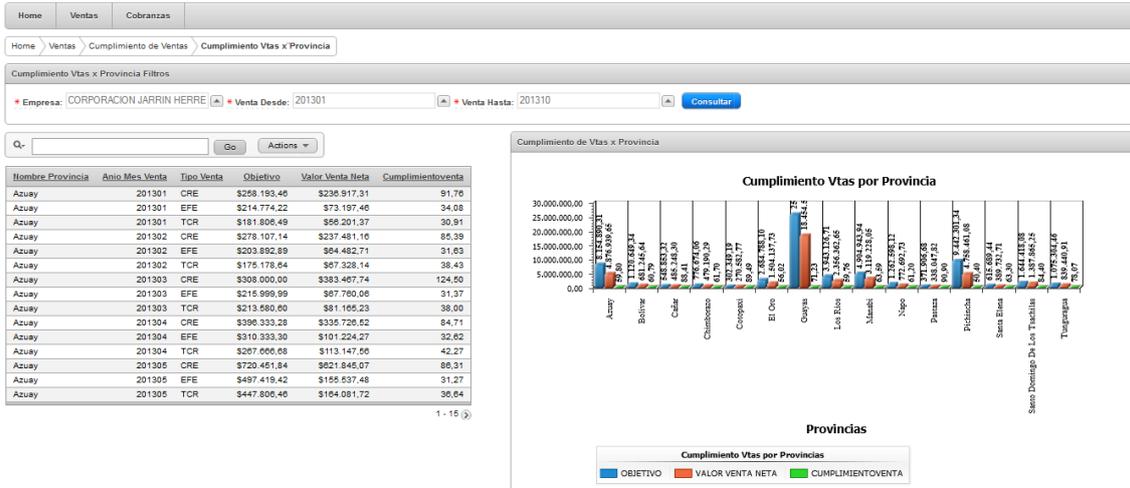
Cumplimiento Vtas x Región

En esta opción se visualizará el cumplimiento de ventas por región, a continuación presentará la siguiente pantalla:



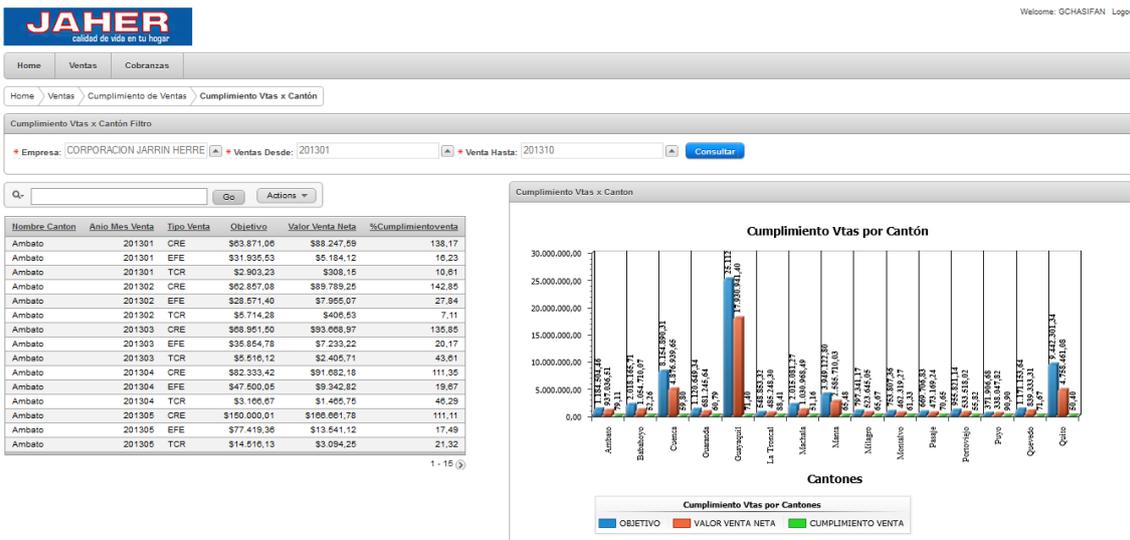
Cumplimiento Vtas x Provincia

En esta opción se visualizará el cumplimiento de ventas por provincias, a continuación presentará la siguiente pantalla:



Cumplimiento Vtas x Cantón

En esta opción se visualizará el cumplimiento de ventas por cantones, a continuación presentará la siguiente pantalla:



Cumplimiento Vtas x Agencia

En esta opción se visualizará el cumplimiento de ventas por agencias, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Ventas > Cumplimiento de Ventas > Cumplimiento Vtas x Agencia

Cumplimiento Vtas x Agencia Filtro

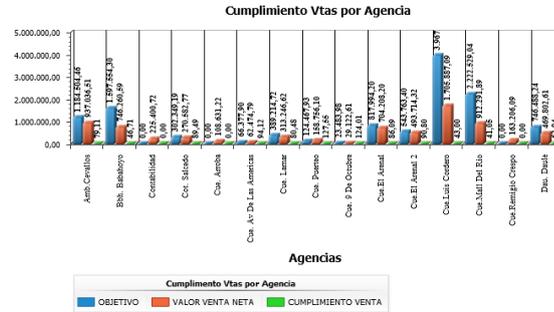
Empresa: CORPORACION JARRIN HERRE [v] Ventas Desde: 201301 Ventas Hasta: 201310 Consultar

Q: Go Actions

Nombre Agencia	Anio Mes Venta	Tipo Venta	Objetivo	Valor Venta Neta	Cumplimientoventa
Amb.Cevallos	201301	CRE	\$83.871,06	\$88.247,59	138,17
Amb.Cevallos	201301	EFE	\$31.935,53	\$5.184,12	16,23
Amb.Cevallos	201301	TCR	\$2.903,23	\$308,15	10,61
Amb.Cevallos	201302	CRE	\$62.857,08	\$69.789,25	142,85
Amb.Cevallos	201302	EFE	\$28.571,40	\$7.555,07	27,84
Amb.Cevallos	201302	TCR	\$5.714,28	\$406,53	7,11
Amb.Cevallos	201303	CRE	\$68.951,50	\$93.668,97	135,85
Amb.Cevallos	201303	EFE	\$35.854,78	\$7.233,22	20,17
Amb.Cevallos	201303	TCR	\$5.516,12	\$2.405,71	43,01
Amb.Cevallos	201304	CRE	\$82.333,42	\$91.682,18	111,35
Amb.Cevallos	201304	EFE	\$47.500,05	\$9.342,82	19,67
Amb.Cevallos	201304	TCR	\$3.166,07	\$1.465,75	46,29
Amb.Cevallos	201305	CRE	\$150.000,01	\$166.661,78	111,11
Amb.Cevallos	201305	EFE	\$77.419,36	\$13.541,12	17,49
Amb.Cevallos	201305	TCR	\$14.516,13	\$3.054,25	21,32

1 - 15

Cumplimiento Vtas x Agencia

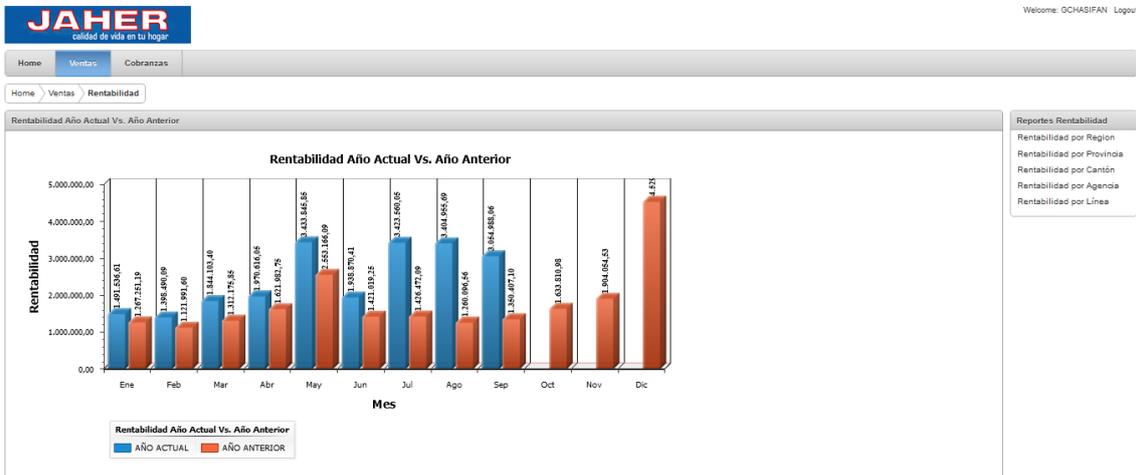




8.2.2 Rentabilidad

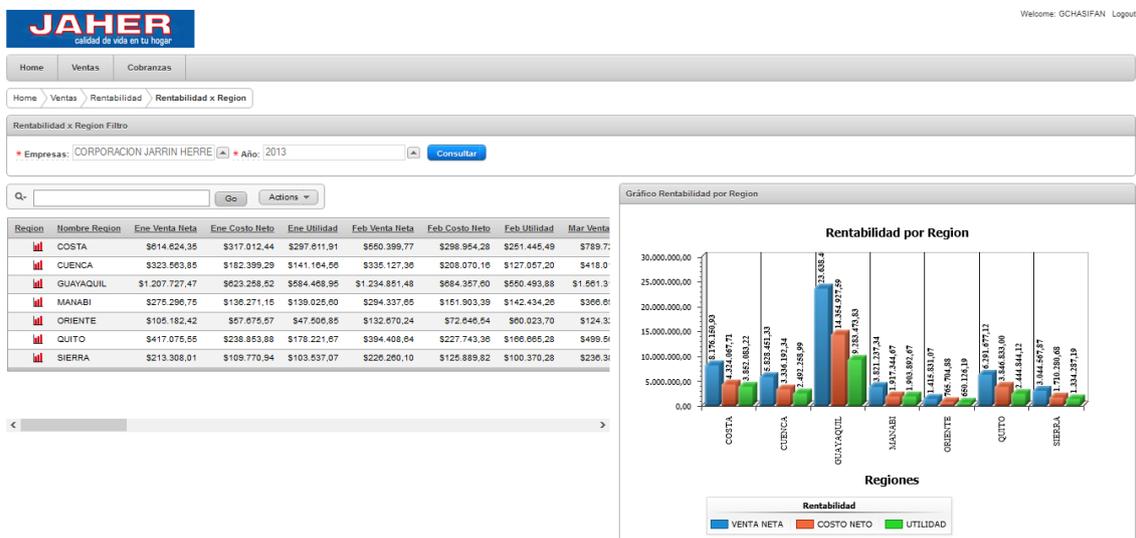
Rentabilidad Año Actual Vs. Año Anterior

En esta opción se visualizará el indicador de la rentabilidad de la empresa del año actual vs. El año anterior, a continuación presentará la siguiente pantalla:



Rentabilidad por Región

En esta opción visualizará la rentabilidad de la empresa por región, a continuación presentará la siguiente pantalla:



En la parte derecha presenta un gráfico de barras de la rentabilidad por regiones de la empresa y el año seleccionado.

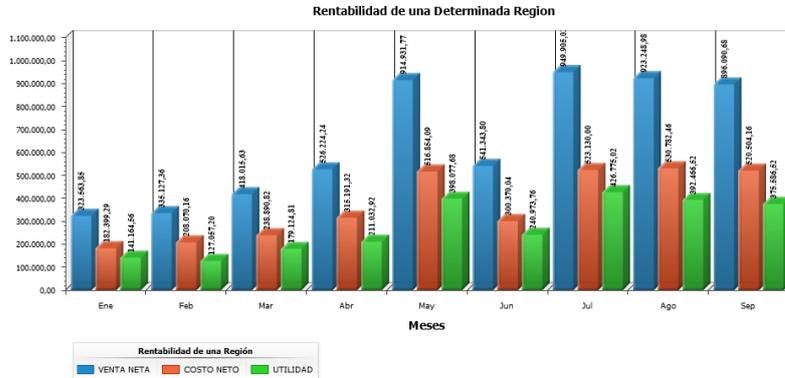
Puede obtener el gráfico de una determinada región presionando el botón  que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Ventas Rentabilidad Rentabilidad x Region Grafico Rentabilidad de una Region

Rentabilidad de una Determinada Region
Año: 2013 Region CUENCA



Rentabilidad por Provincia

En esta opción visualizará la rentabilidad de la empresa por provincias, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

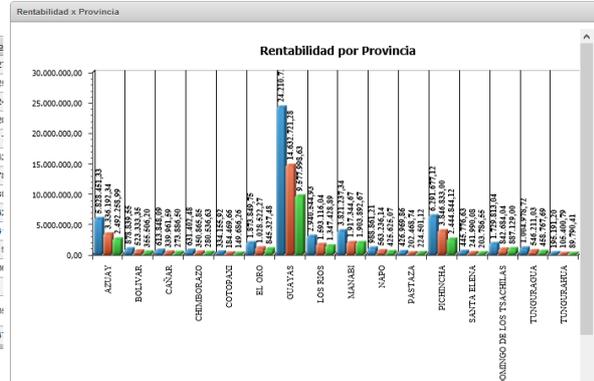
Home Ventas Rentabilidad Rentabilidad x Provincia

Rentabilidad x Provincia Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

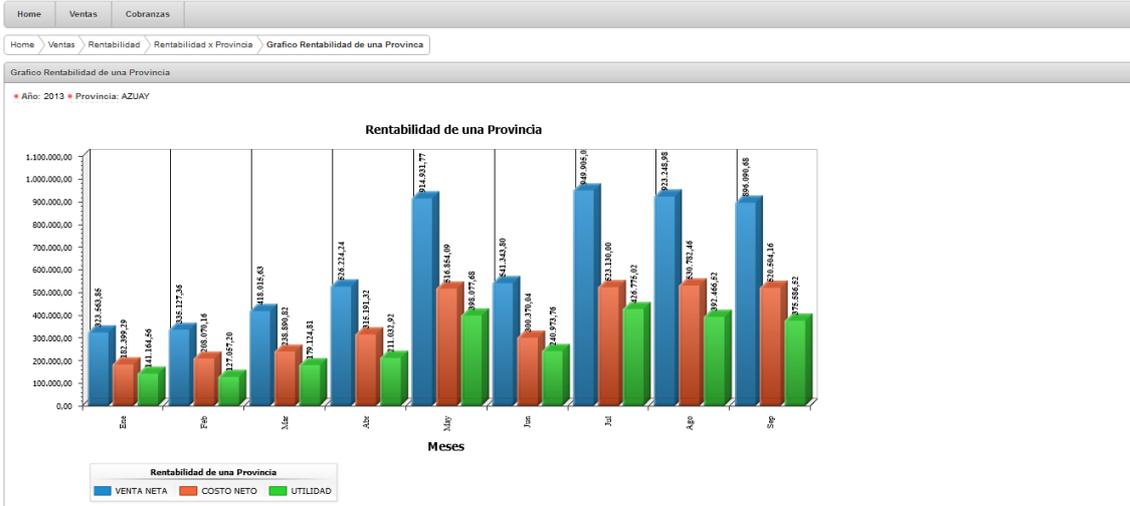
Q: Go Actions

Provincia	Hombre Provincia	Ene. Venta Neta	Ene. Costo Neto	Ene. Utilidad	Feb. Venta Neta	Feb. Costo Neto	Feb. Utilidad
AZUAY		323.683,85	182.399,29	141.164,56	335.127,30	208.070,16	127.057,14
BOLIVAR		57.684,45	32.517,82	25.166,83	63.939,76	38.750,79	25.188,97
CAÑAR		45.853,08	24.778,40	21.075,28	54.388,19	29.756,42	24.631,77
CHIMBORAZO		37.046,55	19.316,46	17.730,09	45.171,79	25.064,49	20.107,30
COTOPAXI		24.837,15	12.195,91	12.641,54	18.997,70	11.240,37	7.757,33
EL ORO		135.709,01	70.857,32	64.851,09	130.428,48	77.692,57	52.735,91
GUAYAS		1.286.977,23	650.441,65	610.535,58	1.290.369,24	712.074,71	578.294,53
LOS RIOS		236.155,39	128.280,42	107.874,97	178.588,53	97.917,45	80.671,08
MANABI		275.296,76	198.271,15	159.025,80	294.337,65	151.903,39	142.021,96
NAPO		74.252,56	43.130,81	31.121,75	94.715,54	53.653,51	41.062,03
PASTAZA		20.929,88	14.544,76	16.385,10	37.954,70	18.953,03	17.000,06
PICHINCHA		417.075,55	238.853,88	178.221,07	394.408,64	227.743,36	166.665,28
SANTA ELENA							
SANTO DOMINGO DE LOS TSACHILAS		137.656,51	65.913,17	71.743,34	131.476,81	65.870,73	65.566,08
TUNGURAGUA		93.739,88	45.741,05	47.998,81	98.150,85	50.834,17	47.264,68



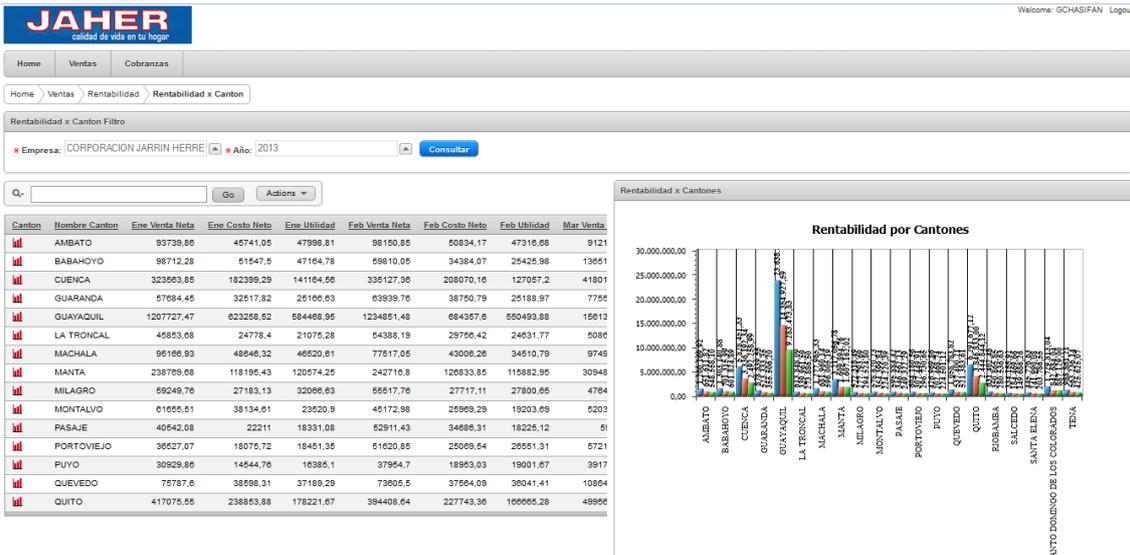
En la parte derecha presenta un gráfico de barras de la rentabilidad por provincias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada provincia presionando el botón  que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



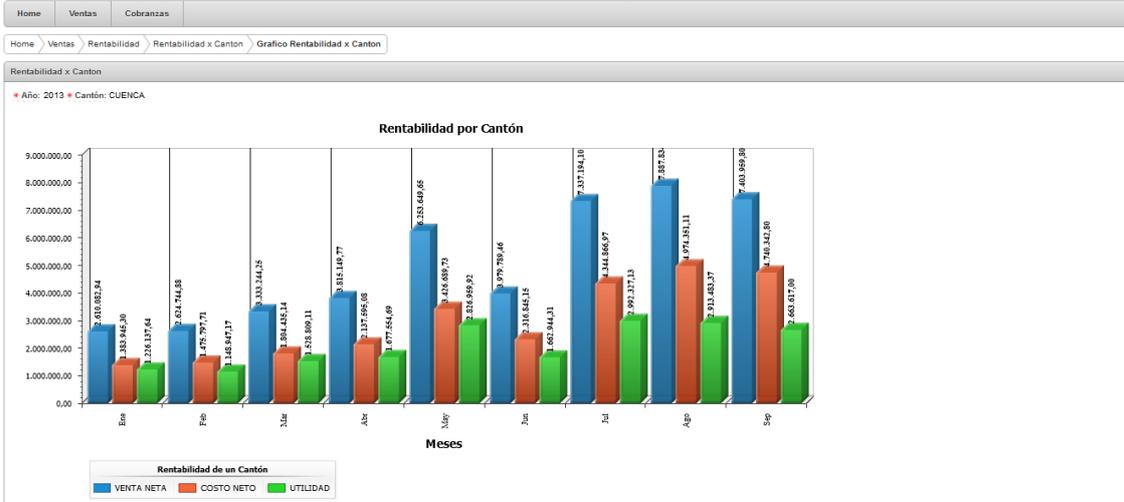
Rentabilidad por Cantón

En esta opción visualizará la rentabilidad de la empresa por cantones, a continuación presentará la siguiente pantalla:



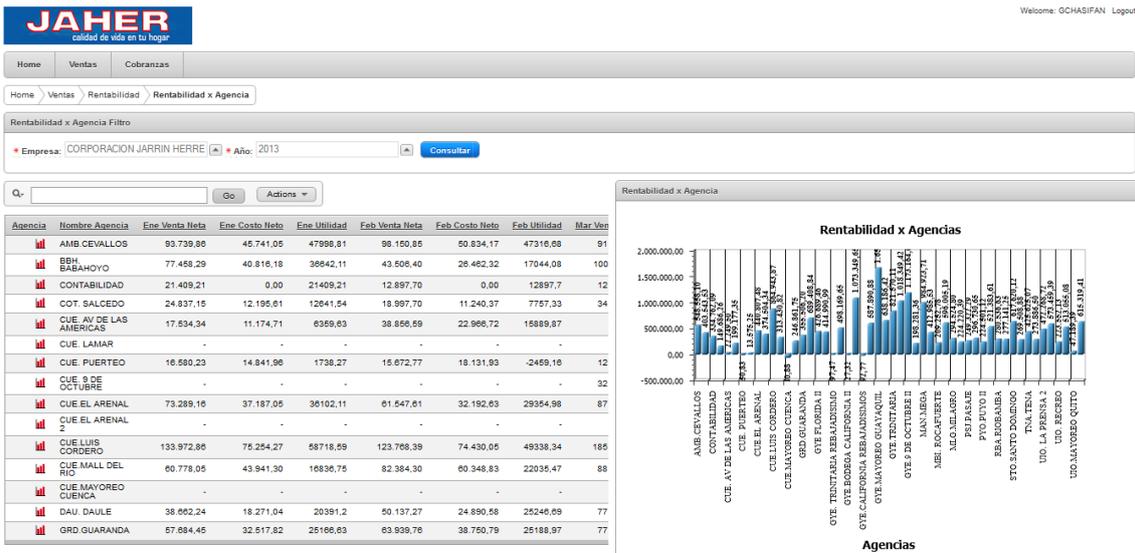
En la parte derecha presenta un gráfico de barras de la rentabilidad por cantones de la empresa y el año seleccionado.

Puede obtener el gráfico de un determinado cantón presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



Rentabilidad por Agencia

En esta opción visualizará la rentabilidad de la empresa por agencias, a continuación presentará la siguiente pantalla:



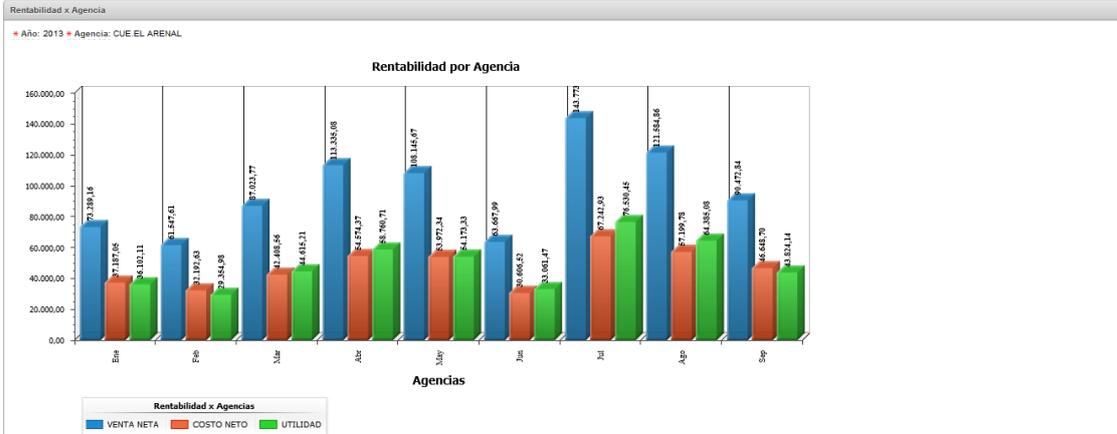
En la parte derecha presenta un gráfico de barras de la rentabilidad por agencias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada agencia presionando el botón  que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Ventas Rentabilidad > Rentabilidad x Agencia > Grafico Rentabilidad x Agencia



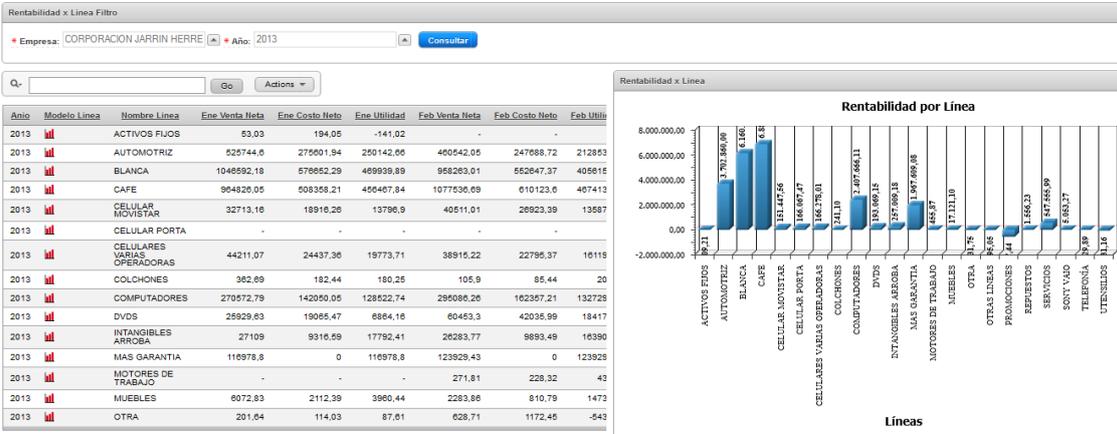
Rentabilidad por Línea

En esta opción visualizará la rentabilidad de la empresa por líneas, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Ventas Rentabilidad > Rentabilidad x Línea



En la parte derecha presenta un gráfico de barras de la rentabilidad por líneas de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada línea presionando el botón  que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

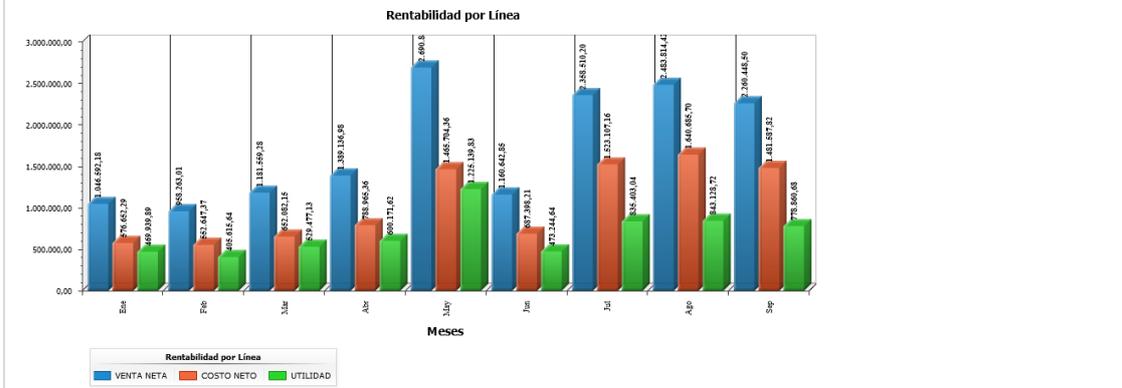


Home Ventas Cobranzas

Home Ventas Rentabilidad Rentabilidad x Línea Gráfico Rentabilidad x Línea

Rentabilidad x Línea

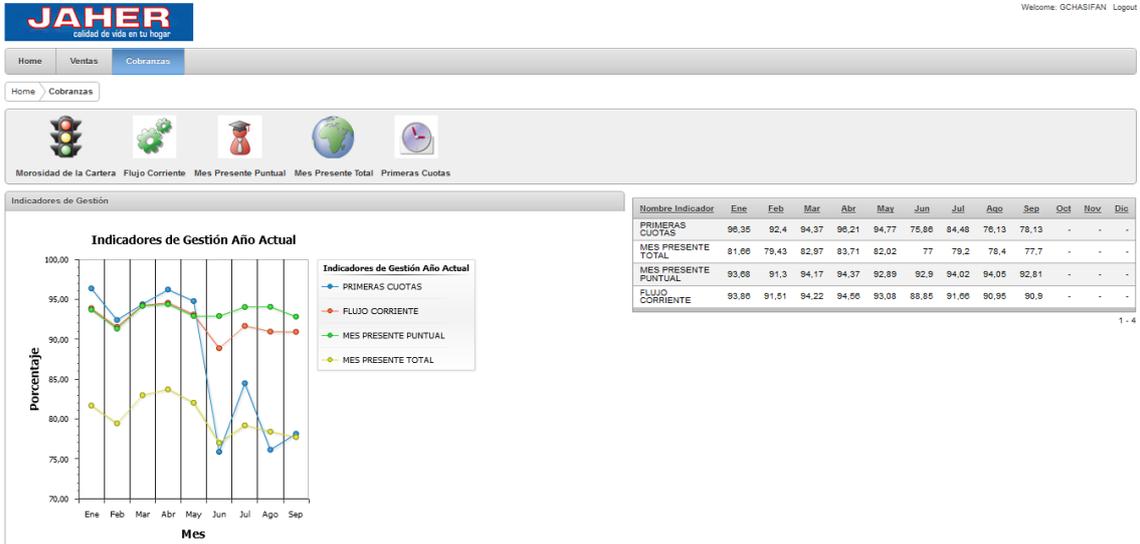
Año: 2013 Línea: BLANCA





8.3 Cobranzas.

En esta opción se visualizará los indicadores de cobranzas, a continuación presentará la siguiente pantalla:



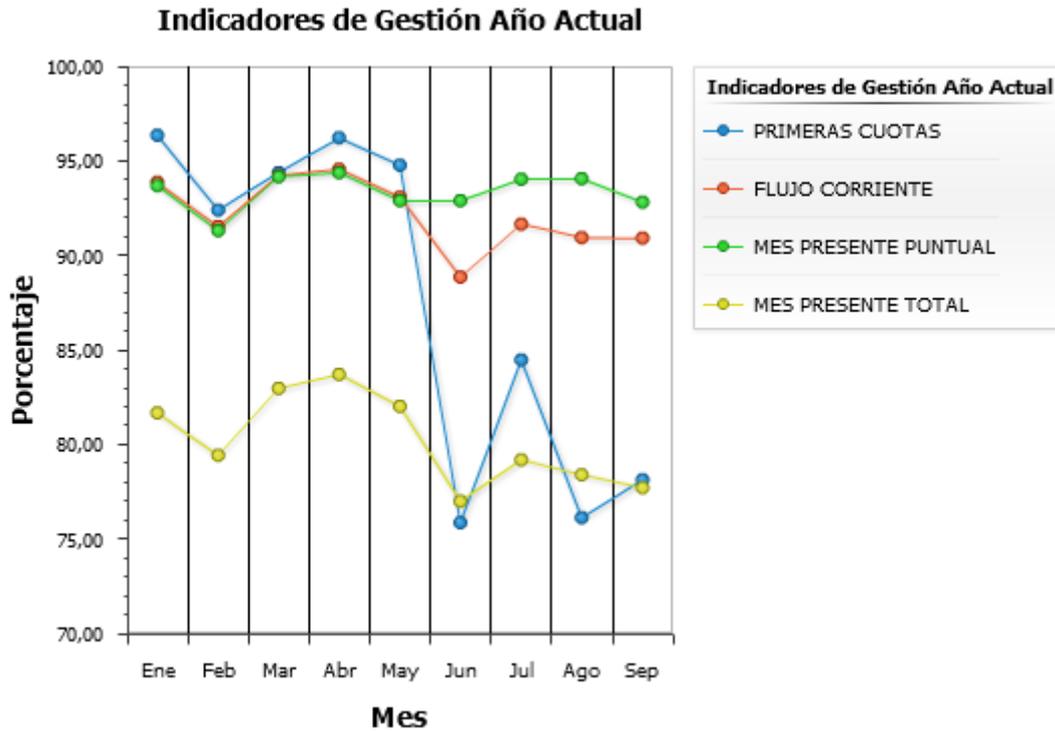
En esta pantalla se visualiza la siguiente información:

Menú de los indicadores de cobranzas





Gráfico lineal de los indicadores de gestión por meses del año actual.



Reporte de los indicadores de gestión en porcentaje por meses:

Nombre Indicador	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
PRIMERAS CUOTAS	96,35	92,4	94,37	96,21	94,77	75,86	84,48	76,13	78,13	-	-	-
MES PRESENTE TOTAL	81,66	79,43	82,97	83,71	82,02	77	79,2	78,4	77,7	-	-	-
MES PRESENTE PUNTUAL	93,68	91,3	94,17	94,37	92,89	92,9	94,02	94,05	92,81	-	-	-
FLUJO CORRIENTE	93,88	91,51	94,22	94,56	93,08	88,85	91,66	90,95	90,9	-	-	-

1 - 4

8.3.1 Morosidad de la cartera

En esta opción visualizará el gráfico del indicador de la morosidad de la cartera en función del mes de siembra, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Morosidad de la Cartera



- Reportes Ind. Morosidad
- Indice de Morosidad por Año y Mes de Siembra
- Indice Morosidad por Cantón Agencia
- Indice Morosidad por Políticas
- Indice Morosidad por Línea

Visualizará los diferentes reportes que se puede obtener de la morosidad de la cartera:

Reportes Ind. Morosidad

- Indice de Morosidad por Año y Mes de Siembra
- Indice Morosidad por Cantón Agencia
- Indice Morosidad por Políticas
- Indice Morosidad por Línea

Índice de Morosidad por Año y Mes de Siembra

En esta opción visualizará la morosidad de la cartera de acuerdo a la siembra, a continuación presentará la siguiente pantalla:



Home Cobranzas Morosidad de la Cartera Índice de Morosidad

Indice de Morosidad por Año y Mes de Siembra

* Empresa: * Periodo: * Política:

* Provincia: * Cantón: * Agencia:

Q: Go Reports 1. Primary Report Actions

Añoventa	Año Mes Venta	Siembra	Ítema Que Cobrar	Valor Cobrado	Saldo Cartera	Cartera Morosida	% Índice Morosidad	% Productividad
2013	201301	3.494.696,17	2.079.971,61	2.090.990,82	1.413.605,36	111.052,22	4,35	100,19
2013	201302	3.690.098,09	2.020.711,76	2.048.401,00	1.547.697,69	96.246,21	4,76	101,27
2013	201303	6.368.114,94	2.797.230,62	2.797.498,46	2.680.616,48	201.424,69	7,20	100,01
2013	201304	5.786.097,91	2.514.933,65	2.542.212,05	3.243.885,86	231.463,84	9,20	101,08
2013	201305	8.011.691,97	2.318.628,49	2.464.699,75	5.656.992,22	179.247,06	7,73	105,87
2013	201306	4.281.758,47	1.138.388,76	1.199.480,96	3.082.277,51	140.841,96	12,37	105,37
2013	201307	4.062.006,38	869.429,61	864.885,70	3.197.120,68	172.180,88	19,80	98,33
2013	201308	4.414.328,83	403.482,85	388.707,84	4.025.620,79	172.036,10	42,64	98,34
2013	201309	4.037.150,08	107.502,64	38.736,54	3.998.414,54	104.063,09	96,80	96,03
2012	201201	2.263.981,00	2.293.009,60	2.194.922,37	99.069,63	98.667,88	3,88	96,12
2012	201202	2.086.949,13	2.060.023,65	1.978.902,36	106.946,78	94.691,97	4,60	96,66
2012	201203	2.469.729,84	2.376.319,91	2.296.298,77	164.431,07	120.657,51	5,08	95,69
2012	201204	2.960.244,89	2.796.871,60	2.722.298,06	267.946,83	127.122,56	4,65	97,37
2012	201205	4.719.459,88	4.288.336,62	4.163.284,79	666.175,09	218.492,08	5,10	96,86
2012	201206	2.498.431,69	2.201.141,96	2.127.718,70	370.712,99	127.103,71	6,77	96,66

Índice Morosidad por Cantón Agencia



En esta opción visualizará la morosidad de la cartera por cantones y agencias, a continuación presentará la siguiente pantalla:

Home > Cobranzas > Morosidad de la Cartera > Índice Morosidad Cantón Agencia

Índice Morosidad por Cantón Agencia

Empresa: CORPORACION JARRIN HERRE | Período: 201310 | Política: TODAS

Provincia: TODAS | Cantón: TODOS | Año Venta: TODOS

Añoventa	Nombre Agencia	Tenia Que Cobrar	Cartera Vencida	Índice Morosidad
2013	AMB CEVALLOS	252.311,08	7.904,67	3,13
2013	BBH BABAHUYO	173.231,56	6.576,60	3,80
2013	COT. SALCEDO	72.977,26	1.728,67	2,37
2013	CUE LAMAR	71.768,87	3.360,73	4,72
2013	CUE FUERTEO	35.639,08	2.154,21	6,04
2013	CUE EL ARENAL	195.325,75	7.137,82	3,65
2013	CUE EL ARENAL 2	71.022,65	3.020,48	4,25
2013	CUE LUIS CORDERO	389.409,07	10.465,55	2,69
2013	CUE MALL DEL RIO	60.892,04	2.500,31	4,11
2013	CUE MAYOREO CUENCA	5.606.670,31	828.226,47	14,77
2013	DAU DAULE	97.018,88	8.400,30	8,66
2013	GRD GUARANDA	204.979,02	3.936,07	1,90
2013	GVE DURAN III	282.441,09	11.536,01	4,40
2013	GVE FLORIDA II	121.051,53	6.427,46	5,31
2013	GVE GUASMO III	153.666,70	9.098,20	5,92

Índice Morosidad por Políticas

En esta opción visualizará la morosidad de la cartera por Políticas de ventas, a continuación presentará la siguiente pantalla:

Home > Cobranzas > Morosidad de la Cartera > Índice de Morosidad por Política

Índice de Morosidad por Política

Empresa: CORPORACION JARRIN HERRE | Período: 201310 | Agencia: TODAS

Go Reports | Primary Report

Añoventa	Nombre Política	Siembra	Tenia Que Cobrar	Valor Cobrado	Cartera Vencida	%Índice Morosidad	Pagos Anticipados	%Productividad
2013	COLEGIOS	40.211,17	4.334,52	7.546,45	398,69	9,20	3.610,62	174,10
2013	COMPAÑÍAS	27.372,23	19.835,00	20.663,19	750,00	3,78	1.577,59	104,17
2013	DISTRIBUCION LINEA BLANCA	686.900,01	29.847,95	35.187,17	18.256,69	66,35	22.594,91	121,97
2013	DISTRIBUCION LINEA CAFE	18.113,95	2.135,51	979,80	1.294,15	60,90	138,44	45,63
2013	DISTRIBUCION MOTOS	6.856.971,23	4.077.317,53	3.672.511,77	633.942,90	15,55	228.137,20	90,07
2013	EMPLREADOS TIPO "R"	672.336,39	202.783,16	215.895,15	14.301,36	7,05	27.413,35	106,47
2013	GENERAL	23.765.740,18	5.855.842,40	6.562.660,08	247.934,90	4,23	954.752,58	112,07
2013	INSTITUCIONES	14.091,28	8.435,72	7.327,20	1.281,71	15,19	173,19	80,86
2013	MAS PLAZO A 6 MESES	129.542,12	21.815,30	14.562,81	10.615,27	48,86	3.362,78	66,76
2013	MESES DE GRACIA	3.305.154,29	2.456.688,60	2.206.775,92	389.251,01	15,84	139.338,33	89,83
2013	MOTOS SG	1.426.427,15	286.715,27	309.829,33	11.468,73	4,00	34.562,79	108,06
2013	OTROS MAYOREO	12.377,91	2.868,25	2.096,03	607,32	28,25	45,10	73,33
2013	PERIFÉRICOS	21.795,95	6.400,28	6.372,72	44,00	0,89	516,44	107,36
2013	PRODUCTO DE LA SEMANA	5.688.546,89	1.097.940,70	1.192.987,56	45.610,92	4,17	140.857,73	108,66
2013	FUERTEO	5.730,03	1.320,16	1.446,71	0,00	0,00	128,55	109,59

Índice Morosidad por Línea

En esta opción visualizará la morosidad de la cartera por líneas, a continuación presentará la siguiente pantalla:



Home > Cobranzas > Morosidad de la Cartera > Índice de Morosidad por Línea

Índice de Morosidad por Línea

* Empresa: CORPORACION JARRIN HERRER * Período: 201310 * Agencia: TODAS

Q:

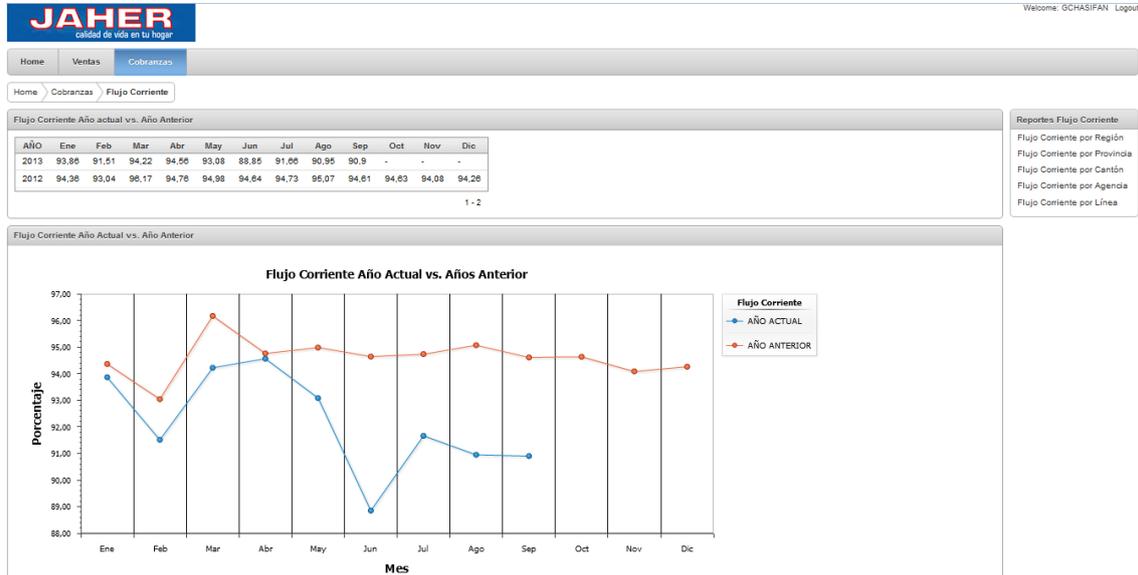
Añoventa	Hombre Línea	Siembra	Ítema Que Cobrar	Valor Cobrado	Cartera Vencida	%Índice Morosidad	Pagos Anticipados	%Productividad
2000	BLANCA	26 544,90	26 544,90	26 544,90	0,00	0,00	0,00	100,00
2000	CAFE	83 541,19	83 541,19	83 541,19	0,00	0,00	0,00	100,00
2001	BLANCA	106 490,79	106 490,79	106 490,00	0,73	0,00	0,00	100,00
2001	CAFE	309 318,40	309 318,40	309 318,40	0,00	0,00	0,00	100,00
2001	COLCHONES	310,73	310,73	310,73	0,00	0,00	0,00	100,00
2001	SIN LINEA	772,18	772,18	772,18	0,00	0,00	0,00	100,00
2002	BLANCA	309 028,28	309 028,28	309 028,28	0,00	0,00	0,00	100,00
2002	CAFE	484 625,56	484 625,56	484 625,56	0,00	0,00	0,00	100,00
2002	COLCHONES	495,00	495,00	495,00	0,00	0,00	0,00	100,00
2002	COMPUTADORES	30 512,00	30 512,00	30 512,00	0,00	0,00	0,00	100,00
2002	DVDS	6 548,00	6 548,00	6 548,00	0,00	0,00	0,00	100,00
2002	OTRAS LINEAS	1 275,00	1 275,00	1 275,00	0,00	0,00	0,00	100,00
2002	SIN LINEA	4 717,46	4 717,46	4 717,46	0,00	0,00	0,00	100,00
2002	TELEFONIA	210,00	210,00	210,00	0,00	0,00	0,00	100,00
2003	BLANCA	725 145,00	725 145,00	725 145,00	0,00	0,00	0,00	100,00

1 - 15 (3)



8.3.2 Flujo Corriente

En esta opción visualizará el gráfico del indicador del flujo Corriente, a continuación presentará la siguiente pantalla:



En la parte derecha presenta el menú de los reportes que puede generar del indicador del flujo corriente:

- Reportes Flujo Corriente
 - Flujo Corriente por Región
 - Flujo Corriente por Provincia
 - Flujo Corriente por Cantón
 - Flujo Corriente por Agencia
 - Flujo Corriente por Línea

Reporte Flujo Corriente por Región

En esta opción generará el reporte del indicador del flujo corriente por región y por meses, a continuación presentará la siguiente pantalla:



JAHER
calidad de vida en tu hogar

Welcome: GCHASIFAN Logout

Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Region

Reporte Flujo Corriente x Region Filtro

* Empresa: CORPORACION JARRIN HERRERA Año: 2013 Consultar

Q- Go Actions

Region	Nombre Region	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	COSTA	93,76	91,36	93,90	93,86	92,46	92,68	94,56	94,86	93,29	-	-	-
	CUENCA	93,80	92,13	92,59	94,29	93,60	79,65	91,32	93,31	90,97	-	-	-
	GUAYAQUIL	72,75	89,59	86,71	93,04	75,68	69,67	88,44	85,68	88,29	-	-	-
	MANABI	93,33	91,78	93,59	94,89	92,67	93,41	93,90	94,94	93,34	-	-	-
	ORIENTE	91,22	88,98	92,62	93,38	94,85	90,58	96,00	94,82	92,99	-	-	-
	QUITO	95,95	90,52	93,87	95,17	95,00	95,50	95,79	95,21	92,90	-	-	-
	SIERRA	95,20	93,72	95,19	95,61	94,56	94,39	95,65	96,00	95,32	-	-	-

1 - 7

Reporte Flujo Corriente x Region

Flujo Corriente por Region

Regiones

Flujo Corriente x Region

■ COSTA ■ CUENCA ■ GUAYAQUIL ■ MANABI ■ ORIENTE ■ QUITO ■ SIERRA

En la parte derecha presenta un gráfico de barras del flujo corriente por regiones de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada región presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

JAHER
calidad de vida en tu hogar

Welcome: GCHASIFAN Logout

Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Region Grafico Flujo Corriente x Region

Flujo Corriente x Region

* Año: 2013 * Región: CUENCA

Flujo Corriente x Región-Meses

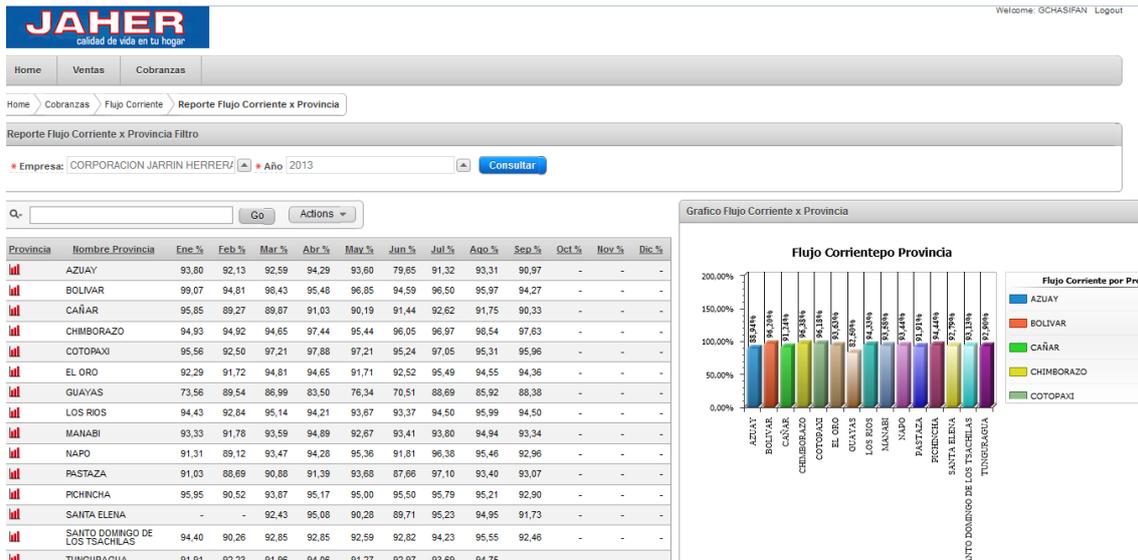
Meses

Flujo Corriente

■ Ene ■ Feb ■ Mar ■ Abr ■ May ■ Jun ■ Jul ■ Ago ■ Sep

Reporte Flujo Corriente por Provincia

En esta opción generará el reporte del indicador del flujo corriente por provincia y por meses, a continuación presentará la siguiente pantalla:





Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Canton

Reporte Filtro Corriente x Canton Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013

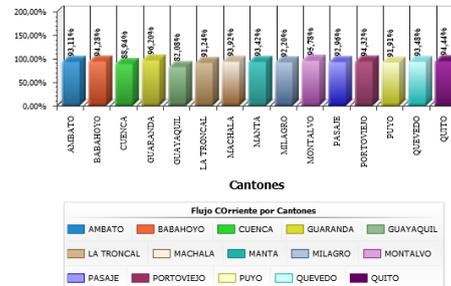
Q:

Canton	Nombre Canton	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMBATO	91.91	92.23	91.96	94.06	91.27	92.97	93.69	94.75	94.53	-	-	-
	BABAHOYO	93.57	92.40	95.63	93.90	94.37	93.22	94.05	95.77	94.36	-	-	-
	CUENCA	93.80	92.13	92.59	94.29	93.60	79.65	91.32	93.31	90.97	-	-	-
	GUARANDA	99.07	94.81	96.43	95.48	96.85	94.59	96.50	95.97	94.27	-	-	-
	GUAYAQUIL	72.75	89.59	86.71	83.04	75.68	69.67	88.44	85.68	88.29	-	-	-
	LA TRONCAL	95.85	89.27	89.87	91.03	90.19	91.44	92.62	91.75	90.33	-	-	-
	MACHALA	92.06	91.98	95.02	94.34	92.37	93.21	95.37	94.62	95.13	-	-	-
	MANTA	93.11	91.99	93.10	94.97	92.30	92.89	93.89	94.70	93.29	-	-	-
	MILAGRO	92.63	88.47	92.59	93.47	92.33	93.15	93.96	91.78	91.03	-	-	-
	MONTALVO	98.62	95.46	96.04	94.52	95.85	95.56	94.67	92.22	95.58	-	-	-
	PASAJE	91.38	91.08	94.30	95.36	90.24	90.98	95.76	94.39	92.59	-	-	-
	PORTOVIEJO	94.12	90.97	95.75	94.00	94.48	95.90	93.36	90.02	93.59	-	-	-
	PUYO	91.03	88.89	90.88	91.39	93.68	87.66	97.10	93.40	93.07	-	-	-
	QUEVEDO	92.69	91.63	94.00	94.37	91.66	92.16	94.22	96.13	94.01	-	-	-
	QUITO	95.95	90.52	93.87	95.17	95.00	95.50	95.79	95.21	92.90	-	-	-

1 - 15

Reporte Flujo Corriente x Canton

Flujo Corriente por Canton



En la parte derecha presenta un gráfico de barras del flujo corriente por cantones de la empresa y el año seleccionado.

Puede obtener el gráfico de un determinado cantón presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

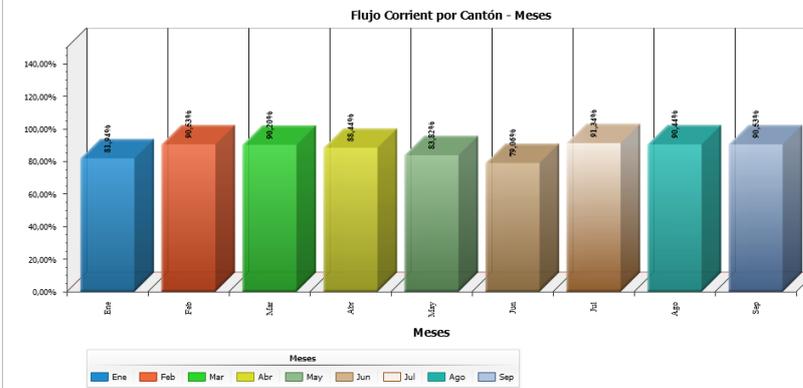


Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Canton Graficar Flujo Corriente x Canton

Reporte Flujo Corriente x Canton

* Año: 2013 * Canton: CUENCA



Reporte Flujo Corriente por Agencia

En esta opción generará el reporte del indicador del flujo corriente por agencias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Agencia

Reporte Flujo Corriente x Agencia Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013 Consultar

Q: Go Actions

Agencia	Nombre Agencia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMB CEVALLOS	91,91	92,23	91,96	94,06	91,27	92,97	93,69	94,76	94,53	-	-	-
	BBH BABAHYO	94,12	92,68	96,57	93,49	94,75	93,13	94,56	95,60	95,25	-	-	-
	COT SALCEDO	95,56	92,50	97,21	97,88	97,21	95,24	97,05	95,31	95,98	-	-	-
	CUE AV DE LAS AMERICAS	93,22	92,86	92,12	-	-	-	-	-	-	-	-	-
	CUE LAMAR	-	-	-	94,24	90,82	95,67	93,25	94,75	90,56	-	-	-
	CUE PUERTO	90,36	90,29	83,36	96,09	87,02	87,59	96,98	88,42	94,13	-	-	-
	CUE 10 DE AGOSTO	95,88	95,33	100,00	67,96	100,00	100,00	100,00	100,00	100,00	-	-	-
	CUE EL ARENAL	91,99	92,54	91,76	93,99	93,66	91,29	94,34	95,75	92,95	-	-	-
	CUE EL ARENAL 2	-	-	-	-	-	100,00	93,46	95,49	93,83	-	-	-
	CUE LUIS CORDERO	94,84	92,62	93,78	95,28	94,16	95,15	95,42	96,29	94,34	-	-	-
	CUE MALL DEL RIO	93,39	87,78	91,11	88,84	95,15	88,84	91,67	93,20	89,45	-	-	-
	CUE MAYOREO CUENCA	-	-	-	-	-	77,03	90,33	92,33	89,82	-	-	-
	CUE SEMINIO CHESPO II	100,00	100,00	-	-	-	-	-	-	-	-	-	-
	DAU DAULE	88,67	92,42	90,33	95,05	89,78	89,62	95,14	93,41	89,58	-	-	-
	GRD GUARANDA	89,07	94,81	98,43	95,48	96,85	94,59	96,50	95,87	94,27	-	-	-

Reporte Flujo Corriente por Agencia



En la parte derecha presenta un gráfico de barras del flujo corriente por agencias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada agencia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

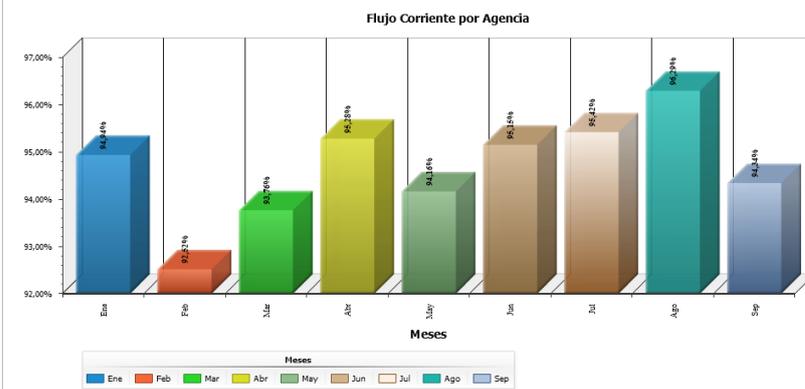


Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Agencia Grafico Flujo Corriente por Agencia

Reporte Flujo Corriente por Agencia

* Año: 2013 * Agencia: CUE LUIS CORDERO



Reporte Flujo Corriente por Línea

En esta opción generará el reporte del indicador del flujo corriente por líneas y por meses, a continuación presentará la siguiente pantalla:



JAHER
calidad de vida en tu hogar

Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Línea

Reporte Flujo Corriente x Línea Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Línea	Nombre Línea	Ene %	Feb %	Mar %	Abr %	May %	Jun %	Jul %	Ago %	Sep %	Oct %	Nov %	Dic %
	ACTIVOS FIJOS	-	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	-	-	-
	AUTOMOTRIZ	91.88	91.44	94.97	94.59	78.37	90.01	82.57	78.16	83.75	-	-	-
	BLANCA	94.05	91.39	94.70	93.80	91.86	74.54	93.20	92.47	92.15	-	-	-
	CAFE	78.89	89.62	98.96	84.53	79.08	84.32	94.25	94.53	93.30	-	-	-
	CELULAR MOVISTAR	92.80	92.48	92.36	93.33	93.55	91.05	92.78	93.18	89.56	-	-	-
	CELULAR PORTA	-	-	-	91.57	83.56	90.90	93.20	93.08	91.09	-	-	-
	CELULARES VARIAS OPERADORAS	91.71	88.50	93.41	90.91	91.75	91.48	91.75	92.95	91.30	-	-	-
	COLCHONES	100.00	68.52	100.00	100.00	64.85	100.00	86.13	100.00	100.00	-	-	-
	COMPUTADORES	88.23	91.49	88.49	88.35	92.37	88.78	93.73	94.03	92.81	-	-	-
	DVDS	52.69	78.67	90.17	91.52	93.29	64.58	88.23	92.46	91.63	-	-	-
	INTANGIBLES ARROBA	75.04	37.28	5.49	100.00	69.59	100.00	64.60	83.56	73.98	-	-	-
	MOTORES DE TRABAJO	100.00	100.00	80.41	-	-	-	-	92.73	67.49	-	-	-
	MUEBLES	92.27	90.56	96.28	93.82	96.09	90.25	89.21	95.16	87.85	-	-	-
	OTRA	100.00	100.00	100.00	100.00	3.30	-	100.00	-	-	-	-	-
	OTRAS LINEAS	-	-	-	-	-	-	-	0.89	32.61	-	-	-

1 - 15

Reporte Flujo Corriente x Línea

Flujo Corriente por Línea

Líneas

En la parte derecha presenta un gráfico de barras del flujo corriente por líneas de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada línea presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

JAHER
calidad de vida en tu hogar

Home Ventas Cobranzas

Home Cobranzas Flujo Corriente Reporte Flujo Corriente x Línea Gráfico Flujo Corriente x Línea

Reporte Flujo Corriente x Línea

Año: 2013 Línea: BLANCA

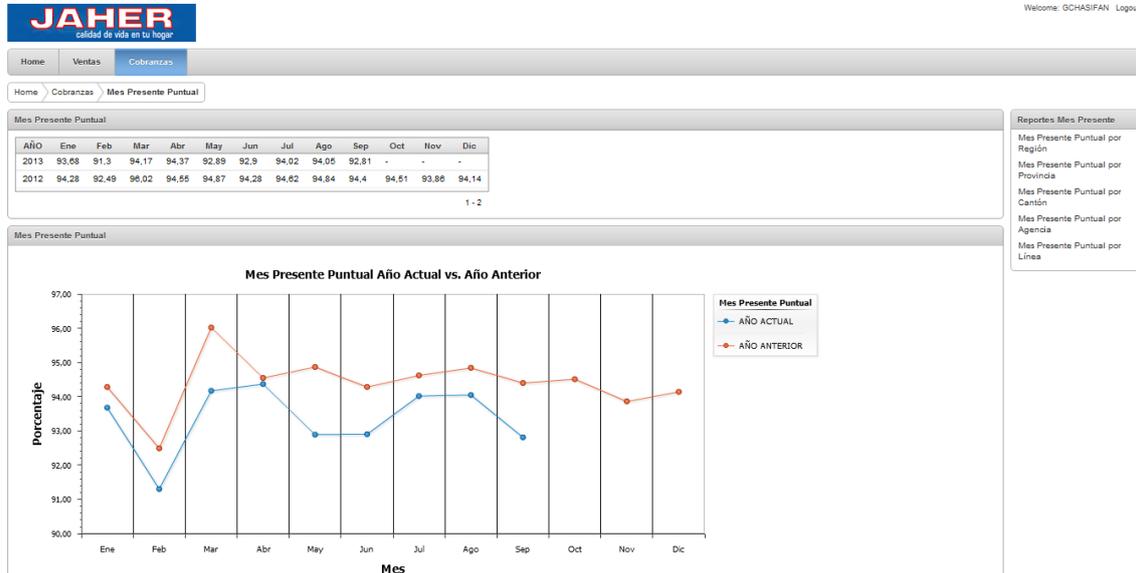
Flujo Corriente por Línea

Meses



8.3.3 Mes Presente Puntual

En esta opción visualizará el gráfico del indicador del Mes Presente Puntual, a continuación presentará la siguiente pantalla:



En la parte derecha presenta el menú de los reportes que puede generar del indicador del mes presente puntual:



Mes Presente Puntual por Región

En esta opción generará el reporte del indicador del mes presente puntual por regiones y por meses, a continuación presentará la siguiente pantalla:



calidad de vida en tu hogar

Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Region

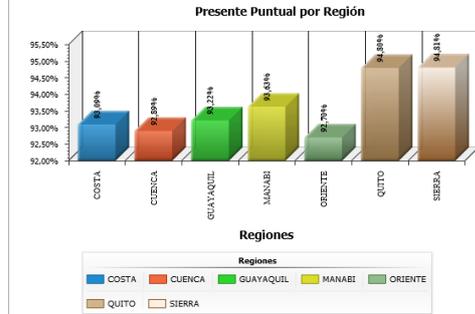
Reporte Presente Puntual x Region Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013

Q: Go

Region	Nombre Region	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	COSTA	93.40	90.78	93.45	93.96	91.89	92.41	94.10	94.00	93.05	-	-	-
	CUENCA	93.46	91.66	92.57	94.20	93.66	92.91	94.14	93.24	90.91	-	-	-
	GUAYAQUIL	93.57	91.22	94.85	94.58	92.30	92.51	93.08	93.59	93.17	-	-	-
	MANABI	93.49	91.06	94.30	94.98	93.04	92.27	93.82	94.66	93.26	-	-	-
	ORIENTE	91.12	89.28	92.95	93.02	94.50	90.52	96.09	94.17	92.37	-	-	-
	QUITO	95.77	92.22	94.17	94.90	95.01	95.33	95.87	95.48	93.93	-	-	-
	SIERRA	94.74	93.21	94.66	95.14	94.55	94.20	95.47	95.96	94.95	-	-	-

Reporte Presente Puntual x Region



En la parte derecha presenta un gráfico de barras del mes presente puntual por regiones de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada región presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

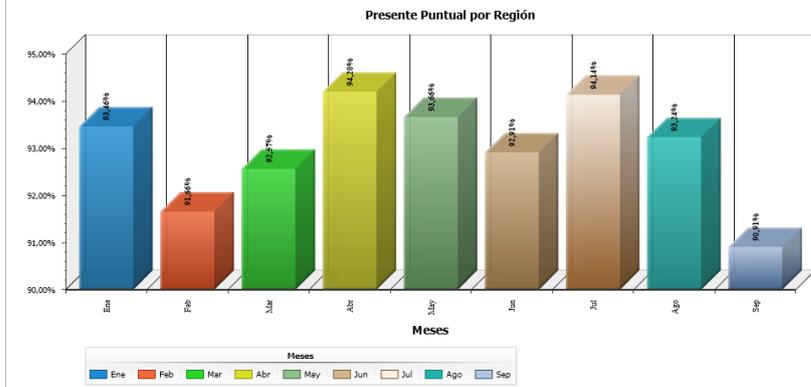


Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Region Gráfico Presente Puntual x Region

Reporte Presente Puntual x Region

* Año: 2013 * Región: CUENCA



Mes Presente Puntual por Provincia

En esta opción generará el reporte del indicador del mes presente puntual por provincias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Provincia

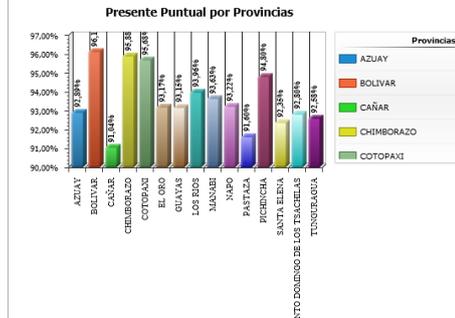
Reporte Presente Puntual x Provincia Filtro

* Empresa: CORPORACION JARRIN HERRERA * Año: 2013 Consultar

Q: Go Actions

Provincia	Nombre Provincia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AZUAY	93,46	91,66	92,57	94,20	93,66	92,91	94,14	93,24	90,91	-	-	-
	BOLIVAR	99,48	95,24	98,88	94,77	96,46	94,32	96,65	96,47	93,53	-	-	-
	CAÑAR	95,94	90,41	89,00	90,92	88,79	91,55	92,25	91,42	89,98	-	-	-
	CHIMBORAZO	94,32	94,04	93,33	97,60	95,04	95,72	95,98	98,38	97,28	-	-	-
	COTOPAXI	94,95	90,35	95,77	97,42	97,15	94,96	96,92	94,71	95,49	-	-	-
	EL ORO	91,84	91,50	94,11	94,45	91,38	92,12	94,59	94,25	93,98	-	-	-
	GUAYAS	93,52	91,06	94,75	94,50	92,29	92,53	93,10	93,49	93,08	-	-	-
	LOS RIOS	93,98	91,65	94,61	93,89	93,24	93,14	94,46	95,77	94,33	-	-	-
	MANABI	93,48	91,06	94,30	94,98	93,04	93,27	93,82	94,66	93,36	-	-	-
	NAPO	91,40	89,46	94,19	94,03	94,95	91,83	95,64	94,74	92,13	-	-	-
	PASTAZA	90,59	88,95	90,35	90,88	93,49	87,60	97,16	92,97	92,86	-	-	-
	PICHINCHA	95,77	92,22	94,17	94,90	95,01	95,33	95,87	95,48	93,93	-	-	-
	SANTA ELENA	-	-	92,13	94,87	89,22	89,28	94,83	94,68	91,68	-	-	-
	SANTO DOMINGO DE LOS TSACHILAS	94,10	89,98	92,39	92,53	92,19	92,63	93,49	95,33	92,24	-	-	-

Reporte Presente Puntual x Provincia



En la parte derecha presenta un gráfico de barras del mes presente puntual por provincias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada provincia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

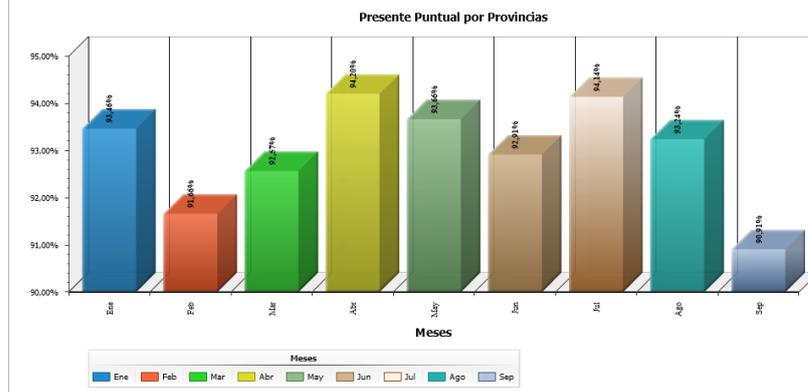


Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Provincia Grafico Presente Puntual x Provincias

Reporte Presente Puntual x Provincias

* Año: 2013 * Provincia: AZUAY



Mes Presente Puntual por Cantón

En esta opción generará el reporte del indicador del mes presente puntual por cantones y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Canton

Reporte Presente Puntual x Cantones Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q- Go Actions

Canton	Nombre Canton	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMBATO	91.42	91.10	91.23	93.43	91.84	92.99	93.04	94.48	94.47	-	-	-
	BABAHYOYO	93.12	90.96	95.10	93.50	93.95	93.02	94.16	95.54	93.92	-	-	-
	CUENCA	93.40	91.06	92.57	94.20	93.66	92.91	94.14	93.24	90.91	-	-	-
	GUARANDA	99.48	95.24	98.88	94.77	96.46	94.32	96.65	96.47	93.53	-	-	-
	GUAYAQUIL	93.57	91.22	94.85	94.88	92.30	92.51	93.08	93.59	93.17	-	-	-
	LA TRONCAL	95.94	90.41	89.00	90.92	88.79	91.65	92.25	91.42	89.98	-	-	-
	MACHALA	92.26	91.00	94.47	94.21	91.91	92.81	94.36	94.38	94.89	-	-	-
	MANTA	93.41	91.44	94.07	95.11	92.73	92.76	93.76	94.42	93.22	-	-	-
	MILAGRO	92.60	88.06	92.77	92.92	92.19	92.82	93.41	91.56	90.93	-	-	-
	MONTALVO	98.54	93.89	95.58	93.74	95.44	95.57	94.28	95.97	95.43	-	-	-
	PASAJE	90.73	91.22	93.20	95.08	90.12	90.55	95.10	93.92	91.88	-	-	-
	PORTOVIJEJO	93.73	89.76	95.24	94.37	94.50	95.76	94.12	95.83	94.00	-	-	-
	PUYO	90.59	88.95	90.35	90.86	93.49	87.00	97.10	92.97	92.88	-	-	-
	QUEVEDO	92.36	91.03	93.45	94.43	91.27	91.90	94.92	95.52	94.18	-	-	-
	QUITO	95.77	92.22	94.17	94.90	95.01	95.33	95.87	95.48	93.93	-	-	-

1 - 15

Reporte Presente Puntual x Cantones

Presente Puntual por Cantones

Cantones

- AMBATO
- BABAHYOYO
- CUENCA
- GUARANDA
- GUAYAQUIL
- LA TRONCAL
- MACHALA
- MANTA
- MILAGRO
- MONTALVO
- PASAJE
- PORTOVIJEJO
- PUYO
- QUEVEDO
- QUITO

En la parte derecha presenta un gráfico de barras del mes presente puntual por cantones de la empresa y el año seleccionado.

Puede obtener el gráfico de un determinado cantón presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Canton Grafico Presente Puntual x Canton

Reporte Presente Puntual x Canton

Año: 2013 Cantón: CUENCA

Presente Puntual por Cantón

Meses

- Ene
- Feb
- Mar
- Abr
- May
- Jun
- Jul
- Ago
- Sep

Mes Presente Puntual por Agencia

En esta opción generará el reporte del indicador del mes presente puntual por agencias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Agencia

Reporte Presente Puntual x Agencia Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Agencia	Nombre Agencia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMB CEVALLOS	91,42	91,10	91,23	93,43	91,84	92,99	93,64	94,48	94,47	-	-	-
	BBH BABAHYO	93,69	91,25	95,06	93,02	94,26	93,03	93,73	95,46	94,84	-	-	-
	COT. SALCEDO	94,95	90,35	95,77	97,42	97,15	94,96	96,92	94,71	95,49	-	-	-
	CUE AV DE LAS AMERICAS	92,49	92,88	92,39	-	-	-	-	-	-	-	-	-
	CUE LAMAR	-	-	-	93,61	90,28	88,05	92,12	96,42	91,26	-	-	-
	CUE PUERTO	88,85	91,19	82,13	96,62	90,34	87,47	96,35	91,52	85,57	-	-	-
	CUE 10 DE AGOSTO	95,88	95,33	100,00	67,96	100,00	100,00	100,00	100,00	100,00	-	-	-
	CUE EL ARENAL	91,44	92,11	92,24	94,88	94,22	91,74	94,20	95,84	92,84	-	-	-
	CUE EL ARENAL 2	-	-	-	-	-	-	97,58	96,14	92,75	-	-	-
	CUE LUIS CORDERO	94,79	92,04	93,73	94,80	93,98	95,04	95,06	96,04	94,15	-	-	-
	CUE MALL DEL RIO	92,51	88,29	89,57	88,82	94,62	88,67	92,87	92,65	91,48	-	-	-
	CUE MAYOREO CUENCA	-	-	-	-	-	92,97	94,02	92,33	89,82	-	-	-
	CUE REMIGIO CRESPO II	100,00	100,00	-	-	-	-	-	-	-	-	-	-
	DAU DAULE	88,14	92,62	90,51	94,98	89,65	89,22	95,50	93,36	89,89	-	-	-
	GRD GUARANDA	99,48	95,24	98,88	94,77	96,46	94,32	96,65	96,47	93,53	-	-	-

1 - 11

Reporte Presente Puntual x Agencia

Mes Presente Puntual por Agencias

Agencia	Porcentaje
AMB CEVALLOS	93,84%
BBH BABAHYO	93,69%
COT. SALCEDO	94,95%
CUE AV DE LAS AMERICAS	92,49%
CUE LAMAR	92,88%
CUE 10 DE AGOSTO	95,88%
CUE EL ARENAL	92,11%
CUE EL ARENAL 2	92,11%
CUE LUIS CORDERO	94,79%
CUE MALL DEL RIO	92,51%
CUE MAYOREO CUENCA	94,02%
CUE REMIGIO CRESPO II	100,00%
DAU DAULE	92,62%
GRD GUARANDA	98,88%

En la parte derecha presenta un gráfico de barras del mes presente puntual por agencias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada agencia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Agencia Grafico Presente Puntual x Agencia

Reporte Presente Puntual x Agencia

Año: 2013 Agencia: CUE LUIS CORDERO

Mes Presente Puntual por Agencia

Mes	Porcentaje
Ene	94,79%
Feb	92,04%
Mar	93,73%
Abr	94,80%
May	93,98%
Jun	95,04%
Jul	95,06%
Ago	96,04%
Sep	94,15%

Mes Presente Puntual por Línea

En esta opción generará el reporte del indicador del mes presente puntual por líneas y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Línea

Reporte Presente Puntual x Línea Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Modelo Línea	Hombre Línea	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
ACTIVOS FIJOS	-	-	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	-	-	-
AUTOMOTRIZ	93.05	91.00	95.00	94.22	92.16	92.76	92.95	94.86	92.46	-	-	-	-
BLANCA	93.95	91.83	94.50	94.81	92.59	93.57	94.39	94.35	94.25	-	-	-	-
CAFE	93.82	90.93	93.84	94.22	93.53	92.94	94.11	94.73	93.23	-	-	-	-
CELULAR MOVISTAR	91.70	92.38	92.06	93.56	93.60	91.49	93.27	93.27	88.53	-	-	-	-
CELULAR PORTA	-	-	-	-	84.83	89.67	93.75	93.01	90.10	-	-	-	-
CELULARES VARIAS OPERADORAS	91.23	87.31	93.32	90.91	91.49	91.45	91.34	93.47	91.27	-	-	-	-
COLCHONES	100.00	97.95	100.00	100.00	94.85	100.00	80.12	100.00	100.00	-	-	-	-
COMPUTADORES	93.15	91.49	94.08	94.19	92.72	91.86	94.29	93.88	92.93	-	-	-	-
DVDOS	97.02	90.07	92.28	90.27	92.96	88.71	92.89	91.47	90.61	-	-	-	-
INTANGIBLES ARROBA	75.04	1.98	5.49	-	100.00	100.00	55.44	56.78	90.20	-	-	-	-
MOTORES DE TRABAJO	100.00	100.00	99.98	-	-	-	-	-	-	-	-	-	-
MUEBLES	-	91.99	93.05	98.85	90.20	89.90	89.97	94.45	87.28	-	-	-	-
OTRA	100.00	100.00	100.00	100.00	3.30	-	100.00	-	-	-	-	-	-
PROMOCIONES	100.00	100.00	100.00	86.25	88.50	100.00	73.20	100.00	69.16	-	-	-	-

1 - 15

Reporte Presente Puntual x Línea

Mes Presente Puntual por Línea

Líneas

- ACTIVOS FIJOS
- AUTOMOTRIZ
- BLANCA
- CAFE

En la parte derecha presenta un gráfico de barras del mes presente puntual por líneas de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada línea presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Mes Presente Puntual Reporte Presente Puntual x Línea Gráfico Reporte Presente Puntual x Línea

Reporte Presente Puntual x Línea

Año: 2013 Línea: BLANCA

Mes Presente Puntual por Línea

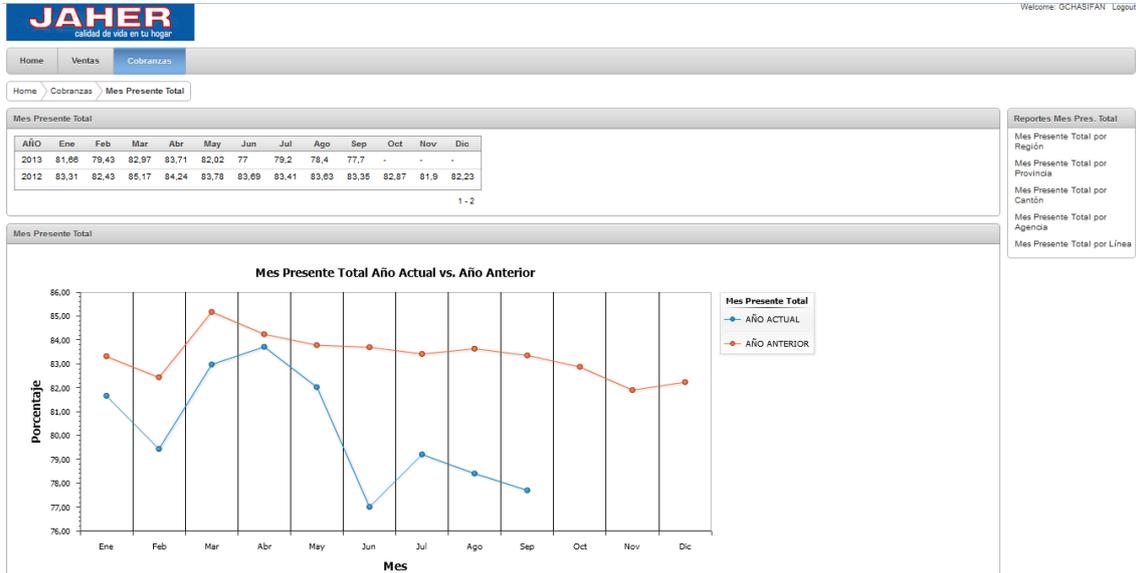
Meses

- Ene
- Feb
- Mar
- Abr
- May
- Jun
- Jul
- Ago
- Sep



8.3.4 Mes Presente Total

En esta opción visualizará el gráfico del indicador del Mes Presente Total, a continuación presentará la siguiente pantalla:



En la parte derecha presenta el menú de los reportes que puede generar del indicador del mes presente total:



Mes Presente Total por Región

En esta opción generará el reporte del indicador del mes presente total por regiones y por meses, a continuación presentará la siguiente pantalla:



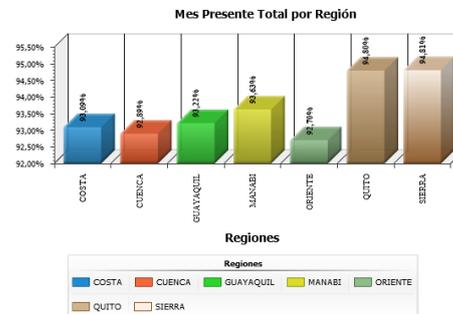
Reporte Presente Total x Region Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Region	Nombre Region	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	COSTA	81,42	78,96	82,57	82,49	80,62	78,93	81,90	83,01	81,82	-	-	-
	CUENCA	86,95	84,68	85,78	87,64	87,24	70,47	76,03	74,24	73,44	-	-	-
	GUAYAQUIL	64,89	76,02	75,43	73,49	67,40	62,23	77,21	75,64	74,74	-	-	-
	MANABI	82,88	80,18	83,47	84,72	82,08	81,49	82,21	82,98	81,71	-	-	-
	ORIENTE	79,02	75,83	80,43	82,31	84,53	80,20	85,52	85,65	83,36	-	-	-
	QUITO	84,19	81,49	85,38	87,21	86,74	86,69	88,25	88,31	86,13	-	-	-
	SIERRA	89,34	87,59	90,31	90,07	88,91	88,22	89,67	90,67	90,39	-	-	-

Region Presente Total x Region

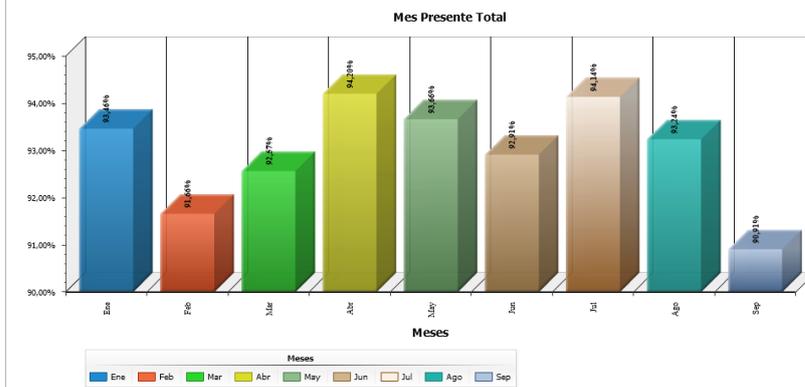


En la parte derecha presenta un gráfico de barras del mes presente total por regiones de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada región presionando el botón  que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Reporte Presente Total x Region

Año: 2013 Región: CUENCA



Mes Presente Total por Provincia

En esta opción generará el reporte del indicador del mes presente total por provincias y por meses, a continuación presentará la siguiente pantalla:



Reporte Presente Total x Provincia Filtro

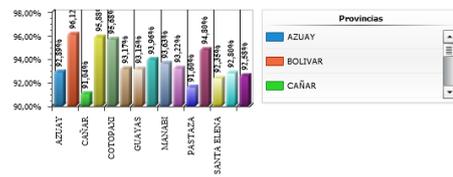
Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Go Actions

Provincia	Nombre Provincia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
■	AZUAY	86,95	84,68	85,78	87,64	87,24	70,47	76,03	74,24	73,44	-	-	-
■	BOLIVAR	97,01	93,07	96,00	92,79	91,64	90,89	91,84	92,11	89,47	-	-	-
■	CAÑAR	84,68	81,62	82,48	82,90	81,22	81,38	85,01	82,98	82,49	-	-	-
■	CHIMBORAZO	87,89	88,24	89,88	91,32	89,30	89,62	91,15	93,51	94,30	-	-	-
■	COTOPAXI	93,97	92,39	96,38	94,71	93,69	92,29	91,94	91,05	90,07	-	-	-
■	EL ORO	80,25	77,99	83,84	84,63	82,69	81,28	84,99	86,46	87,03	-	-	-
■	GUAYAS	65,43	75,87	75,49	73,70	67,78	62,81	77,33	75,75	74,85	-	-	-
■	LOS RIOS	84,65	82,93	85,23	84,07	82,98	80,54	82,79	83,95	83,10	-	-	-
■	MANABI	82,88	80,18	83,47	84,72	82,08	81,49	82,21	82,98	81,71	-	-	-
■	NAPO	79,31	76,55	82,03	85,24	87,59	83,12	87,36	87,93	84,93	-	-	-
■	PASTAZA	78,49	74,40	77,31	78,35	78,35	73,98	81,77	80,86	80,10	-	-	-
■	PICHINCHA	84,19	81,49	85,38	87,21	86,74	86,69	88,25	88,31	86,13	-	-	-
■	SANTA ELENA	-	-	83,88	85,77	79,95	75,12	80,59	81,58	77,35	-	-	-
■	SANTO DOMINGO DE LOS TSACUILLAS	78,97	75,85	79,34	78,75	76,84	76,74	77,95	80,30	78,00	-	-	-
■	TUNGURAGUA	83,68	82,18	84,60	86,05	85,16	84,00	86,99	87,83	-	-	-	-

Reporte Presente Total x Provincia

Mes Presente Total por Provincia



Provincias

En la parte derecha presenta un gráfico de barras del mes presente total por provincias de la empresa y el año seleccionado.

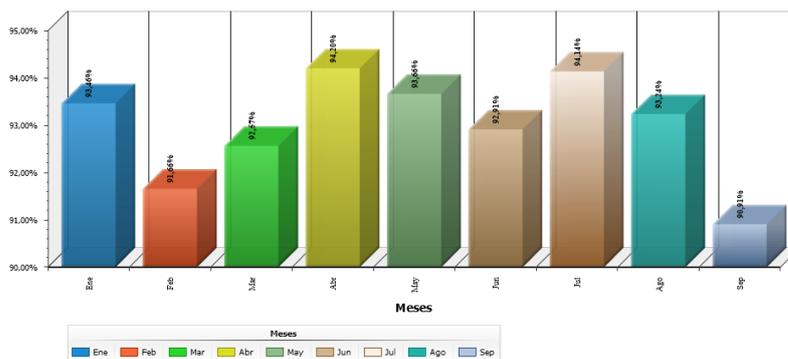
Puede obtener el gráfico de una determinada provincia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



Reporte Presente Total x Provincia

Año: 2013 Provincia: AZUAY

Mes Presente Total por Provincia



Mes Presente Total por Cantón

En esta opción generará el reporte del indicador del mes presente total por cantones y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Reporte Presente Total por Canton

Reporte Presente Total x Canton Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013

Q-

Canton	Nombre Canton	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMBATO	83,68	82,18	84,60	86,05	85,16	84,00	86,99	87,83	88,96	-	-	-
	BABAHYOY	83,35	81,84	85,37	83,39	82,85	81,36	83,67	84,49	83,34	-	-	-
	CUENCA	86,95	84,08	85,78	87,64	87,24	70,47	76,03	74,24	73,44	-	-	-
	GUARANDA	97,01	93,07	96,06	92,79	91,64	90,89	91,84	92,11	89,47	-	-	-
	GUAYAQUIL	64,89	76,02	75,43	73,49	67,40	62,23	77,21	75,64	74,74	-	-	-
	LA TRONCAL	84,68	81,62	82,48	82,90	81,22	81,38	85,01	82,96	82,49	-	-	-
	MACHALA	79,71	78,99	83,45	83,07	81,70	80,96	84,70	86,32	87,96	-	-	-
	MANTA	82,95	81,00	83,97	85,71	83,00	82,11	82,95	83,65	82,17	-	-	-
	MILAGRO	77,52	72,97	76,70	77,87	76,05	76,59	79,80	79,39	77,71	-	-	-
	MONTALVO	93,54	90,98	90,16	88,87	90,30	87,83	89,51	89,64	88,01	-	-	-
	PASAJE	81,67	80,50	84,79	88,35	84,97	81,93	85,65	88,80	84,93	-	-	-
	PORTOVIJO	82,61	77,15	81,43	80,45	77,97	78,70	78,88	80,09	79,66	-	-	-
	PUYO	78,49	74,40	77,31	76,35	78,35	73,98	81,77	80,86	80,10	-	-	-
	QUEVEDO	80,94	79,44	82,33	82,19	79,26	75,75	78,00	80,27	80,15	-	-	-
	QUITO	84,19	81,49	85,38	87,21	85,74	86,69	88,25	88,31	86,13	-	-	-

1 - 15

Reporte Presente Total x Canton

Mes Presente Total por Canton

En la parte derecha presenta un gráfico de barras del mes presente total por cantones de la empresa y el año seleccionado.

Puede obtener el gráfico de un determinado cantón presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Reporte Presente Total por Canton Grafico Reporte Presente Total x Canton

Reporte Presente Total x Canton

* Año: 2013 * Cantón: CUENCA

Mes Presente Total por Cantón

Mes Presente Total por Agencia

En esta opción generará el reporte del indicador del mes presente total por agencias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Report Presente Total x Agencia

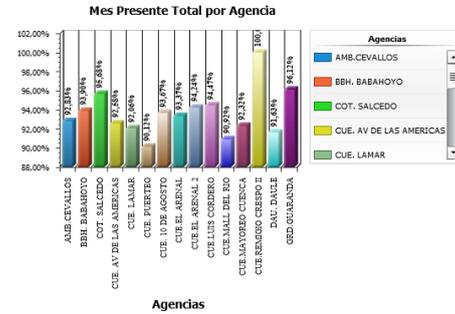
Reporte Presente Total x Agencia Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Agencia	Nombre Agencia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMB CEVALLOS	83.08	82.18	84.00	85.05	85.10	84.00	86.99	87.83	88.96	-	-	-
	BBH BABAHOYO	83.91	82.47	86.04	83.65	83.21	81.54	83.99	84.62	84.50	-	-	-
	COT. SALCEDO	93.97	92.39	95.38	94.71	93.69	92.29	91.94	91.05	90.07	-	-	-
	CUE AV DE LAS AMERICAS	85.29	84.89	84.76	-	-	-	-	-	-	-	-	-
	CUE LAMAR	-	-	-	87.62	84.70	79.48	85.02	80.78	84.63	-	-	-
	CUE PUERTO	80.19	72.23	69.56	86.91	80.44	78.57	85.57	81.28	74.97	-	-	-
	CUE 10 DE AGOSTO	86.67	83.17	70.20	45.33	57.17	44.63	47.65	47.65	100.00	-	-	-
	CUE EL ARENAL	85.35	85.24	84.59	86.31	87.04	82.35	85.73	88.17	85.26	-	-	-
	CUE EL ARENAL 2	-	-	-	-	-	100.00	93.48	94.82	90.82	-	-	-
	CUE LUIS CORDERO	88.26	85.39	87.49	89.10	87.67	87.54	88.62	89.18	88.12	-	-	-
	CUE MALL DEL RIO	91.46	85.78	87.68	85.03	92.00	84.45	86.68	88.69	85.10	-	-	-
	CUE MAYOREO CUENCA	-	-	-	-	-	88.04	73.50	69.89	68.83	-	-	-
	CUE REMIGIO CRESPO II	46.00	100.00	-	-	-	-	-	-	-	-	-	-
	DAU DAULE	72.51	73.96	73.63	75.51	74.28	71.15	79.24	78.91	74.72	-	-	-
	GRD GUARANDA	87.01	93.07	86.06	92.79	91.64	90.89	91.64	92.11	89.47	-	-	-

Reporte Presente Total x Agencia



En la parte derecha presenta un gráfico de barras del mes presente total por agencias JARRIN HERRE y el año seleccionado.

Puede obtener el gráfico de una determinada agencia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

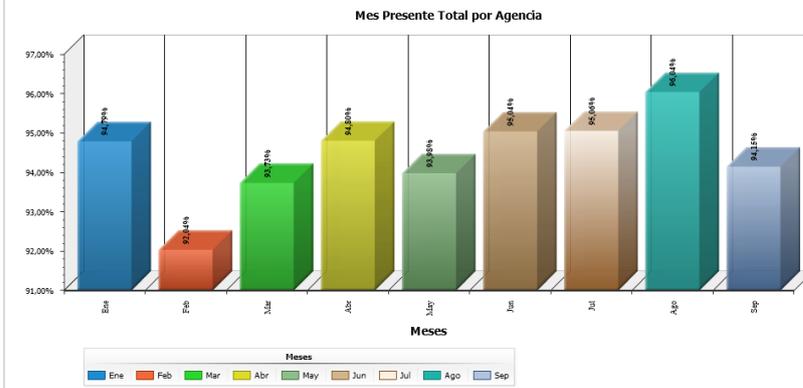


Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Report Presente Total x Agencia Grafico Reporte Presente Total x Agencia

Reporte Presente Total x Agencia

Año: 2013 Agencia: CUE LUIS CORDERO



Mes Presente Total por Línea

En esta opción generará el reporte del indicador del mes presente total por líneas y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Reporte Presente Total x Línea

Reporte Presente Total x Línea Filtro

Empresa: CORPORACION JARRIN HERRE Años: 2013 Consultar

Q: Go Actions

Modelo Línea	Nombre Línea	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
ACTIVOS FIJOS	ACTIVOS FIJOS	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	-	-	-
AUTOMOTRIZ	AUTOMOTRIZ	80.12	78.73	83.58	84.40	71.58	79.91	75.85	71.19	70.89	-	-	-
BLANCA	BLANCA	82.50	79.95	83.97	83.71	81.43	67.25	82.78	82.94	81.66	-	-	-
CAFE	CAFE	71.97	78.46	77.68	76.09	71.31	74.32	82.41	83.24	82.23	-	-	-
CELULAR MOVISTAR	CELULAR MOVISTAR	85.25	85.09	84.51	85.95	84.67	82.56	84.88	84.23	81.46	-	-	-
CELULAR PORTA	CELULAR PORTA	-	-	-	91.57	81.48	83.24	87.98	87.54	84.18	-	-	-
CELULARES VARIAS OPERADORAS	CELULARES VARIAS OPERADORAS	82.29	77.88	83.88	83.05	82.83	81.34	81.00	82.95	80.82	-	-	-
COLCHONES	COLCHONES	62.24	38.83	46.64	100.00	64.85	58.50	56.31	63.88	100.00	-	-	-
COMPUTADORES	COMPUTADORES	61.13	77.11	77.00	77.43	80.20	76.17	81.10	82.28	80.85	-	-	-
DVDS	DVDS	52.87	76.87	87.22	89.18	89.44	62.94	84.84	83.71	81.93	-	-	-
INTANGIBLES ARROBA	INTANGIBLES ARROBA	61.03	33.51	5.49	49.82	69.09	86.36	55.71	82.00	84.51	-	-	-
MOTORES DE TRABAJO	MOTORES DE TRABAJO	100.00	100.00	80.41	100.00	-	-	-	92.80	71.52	-	-	-
MUEBLES	MUEBLES	92.27	80.76	96.51	91.56	92.69	88.45	84.03	83.53	78.15	-	-	-
OTRA	OTRA	100.00	100.00	100.00	100.00	3.30	100.00	100.00	-	-	-	-	-
OTRAS LINEAS	OTRAS LINEAS	-	-	-	-	-	-	-	0.89	32.51	-	-	-

Mes Presente Total por Línea

Lineas: ACTIVOS FIJOS, AUTOMOTRIZ, BLANCA, CAFE, CELULAR MOVISTAR, CELULAR PORTA, CELULARES VARIAS OPERADORAS, COLCHONES, COMPUTADORES, DVDS, INTANGIBLES ARROBA, MOTORES DE TRABAJO, MUEBLES, OTRA, PROLOGADORES

En la parte derecha presenta un gráfico de barras del mes presente total por líneas de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada línea presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Mes Presente Total Reporte Presente Total x Línea Gráfico Reporte Presente Total x Línea

Reporte Presente Total x Línea

Año: 2013 Línea: BLANCA

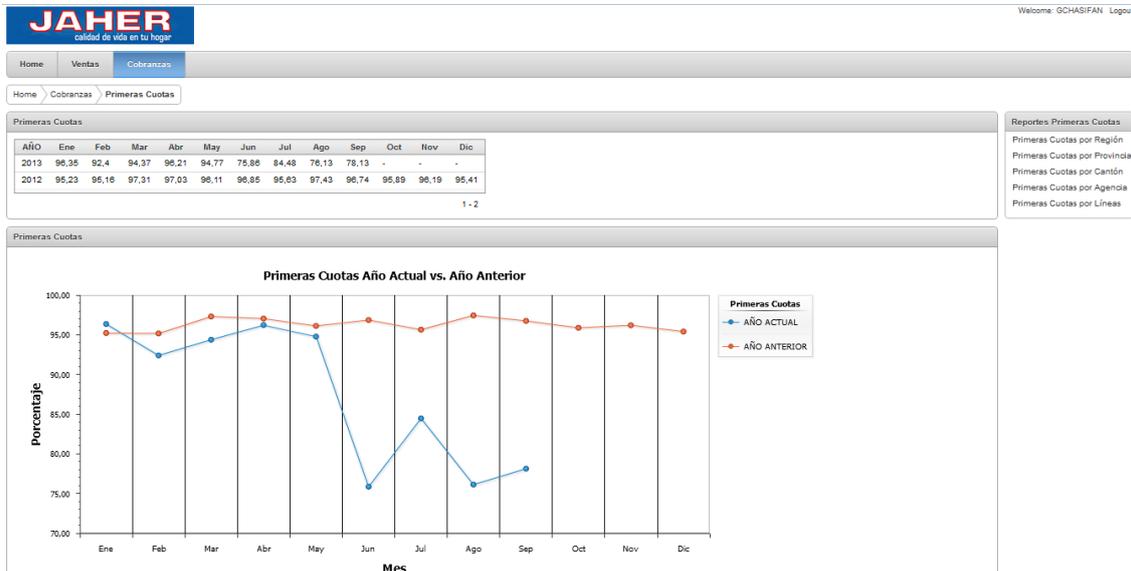
Mes Presente Total por Línea

Meses: Ene, Feb, Mar, Abr, May, Jun, Jul, Ago, Sep



8.3.5 Primeras Cuotas

En esta opción visualizará el gráfico del indicador de las Primeras Cuotas, a continuación presentará la siguiente pantalla:



En la parte derecha presenta el menú de los reportes que puede generar del indicador de las Primeras Cuotas:

- Reportes Primeras Cuotas
- Primeras Cuotas por Región
- Primeras Cuotas por Provincia
- Primeras Cuotas por Cantón
- Primeras Cuotas por Agencia
- Primeras Cuotas por Líneas

Primeras Cuotas por Región

En esta opción generará el reporte del indicador de las primeras cuotas por regiones y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Region

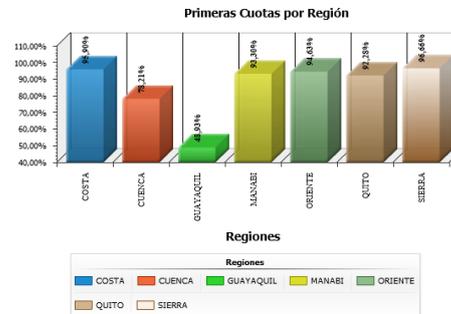
Reporte Primeras Cuotas x Region Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Region	Nombre Region	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	COSTA	98.12	93.89	95.30	96.53	96.67	95.88	96.48	96.90	96.34	-	-	-
	CUENCA	96.80	93.80	92.68	94.79	93.15	88.75	85.90	94.77	92.47	-	-	-
	GUAYAQUIL	13.89	82.65	63.65	38.00	23.13	17.29	74.83	67.04	69.91	-	-	-
	MANABI	90.90	94.43	91.61	94.00	89.88	95.09	93.67	97.01	93.13	-	-	-
	ORIENTE	92.48	87.45	91.65	95.92	97.88	91.14	88.83	99.39	88.85	-	-	-
	QUITO	97.95	85.14	92.94	97.22	94.95	97.41	95.37	93.38	88.04	-	-	-
	SIERRA	98.20	95.67	96.78	98.42	94.06	95.95	96.30	96.23	97.89	-	-	-

Primeras Cuotas x Region



En la parte derecha presenta un gráfico de barras de las primeras cuotas por regiones de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada región presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

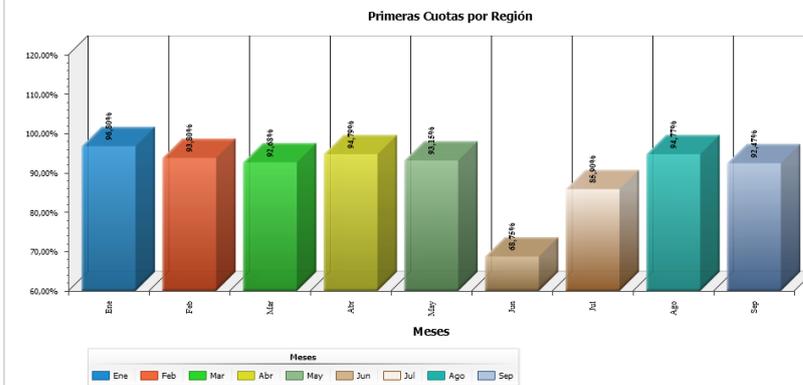


Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Region Grafico Reporte Primeras Cuotas x Region

Reporte Primeras Cuotas x Region

Año: 2013 Región: CUENCA



Primeras Cuotas por Provincia

En esta opción generará el reporte del indicador de las primeras cuotas por provincias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Provincia

Reporte Primeras Cuotas x Provincia Filtro

Empresa: CORPORACION JARRIN HERRE Año: 2013 Consultar

Q: Go Actions

Provincia	Nombre Provincia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AZUAY	96.80	93.80	92.88	94.79	93.15	68.75	85.90	94.77	92.47	-	-	-
	BOLIVAR	97.56	92.85	96.80	100.00	100.00	90.27	95.80	93.93	98.85	-	-	-
	CAÑAR	94.37	85.17	91.73	91.90	97.81	90.68	94.11	93.71	93.35	-	-	-
	CHIMBORAZO	100.00	99.04	99.12	96.24	99.99	99.31	100.00	100.00	100.00	-	-	-
	COTOPAXI	100.00	95.38	98.89	100.00	97.53	97.92	97.53	99.63	100.00	-	-	-
	EL ORO	98.41	92.74	96.95	96.87	95.18	90.57	99.13	96.88	98.72	-	-	-
	GUAYAS	14.58	82.98	84.93	39.17	29.75	17.95	75.03	67.31	70.06	-	-	-
	LOS RIOS	98.74	96.76	97.04	97.37	97.06	95.56	94.70	97.70	96.32	-	-	-
	MANABI	90.90	94.43	91.01	84.00	88.88	95.09	93.07	97.01	93.13	-	-	-
	NAPO	90.09	87.89	91.29	95.91	98.84	91.65	99.98	98.74	99.30	-	-	-
	PASTAZA	98.20	86.88	92.36	95.95	95.48	88.68	96.88	98.06	96.76	-	-	-
	PICHINCHA	97.95	85.14	92.94	97.22	94.95	97.41	95.37	93.36	86.04	-	-	-
	SANTA ELENA	-	-	93.39	96.14	98.02	96.43	96.95	99.63	94.47	-	-	-
	SANTO DOMINGO DE LOS TSACHILAS	98.98	91.50	94.18	95.38	96.60	95.72	97.44	96.91	95.49	-	-	-
	TUNGURAGUA	98.04	96.38	94.32	97.61	87.05	92.74	93.83	96.42	-	-	-	-

Reporte Primeras Cuotas x Provincia



En la parte derecha presenta un gráfico de barras de las primeras cuotas por provincias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada provincia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

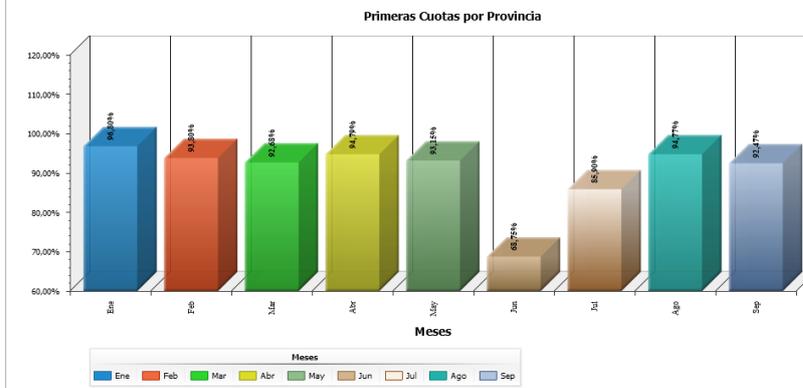


Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Provincia Grafico Reporte Primeras Cuotas

Reporte Primeras Cuotas

Año: 2013 Provincia: AZUAY



Primeras Cuotas por Cantón

En esta opción generará el reporte del indicador de las primeras cuotas por cantones y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Canton

Reporte Primeras Cuotas x Canton Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013

Q: Go

Canton	Nombre Canton	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMBATO	98.04	96.38	94.32	97.01	97.05	92.74	93.83	96.42	95.03	-	-	-
	BABAHoyo	99.20	96.90	97.51	98.69	97.75	95.35	96.48	97.82	99.37	-	-	-
	CUENCA	96.80	93.80	92.05	94.79	93.15	68.75	85.90	94.77	92.47	-	-	-
	GUARANDA	97.56	92.95	96.80	100.00	100.00	96.27	95.80	93.93	98.85	-	-	-
	GUAYAQUIL	13.89	82.65	63.65	38.00	29.13	17.29	74.83	67.04	69.91	-	-	-
	LA TRONCAL	94.37	85.17	91.73	91.90	97.81	90.08	94.11	93.71	93.35	-	-	-
	MACHALA	100.00	93.90	96.72	96.53	98.04	97.52	99.34	96.69	98.08	-	-	-
	MANTA	87.80	93.93	90.58	93.60	89.22	94.52	94.50	96.96	94.15	-	-	-
	MILAGRO	93.26	90.00	92.07	98.32	94.49	97.71	96.01	94.54	93.91	-	-	-
	MONTALVO	96.94	99.98	97.55	100.00	98.54	95.52	96.21	98.18	98.82	-	-	-
	PASAJE	96.54	96.53	97.48	97.21	91.10	94.72	98.00	97.40	100.00	-	-	-
	PORTOVIÑO	98.53	96.88	97.65	96.06	94.29	97.34	89.93	97.18	86.00	-	-	-
	PUYO	98.20	86.68	92.36	95.95	95.48	88.68	96.88	98.08	96.75	-	-	-
	QUEVEDO	97.75	93.94	96.11	93.76	95.14	95.97	90.08	97.54	91.69	-	-	-
	QUITO	97.95	85.14	92.94	97.22	94.96	97.41	95.37	93.36	86.04	-	-	-

1 - 15

Reporte Primeras Cuotas x Canton



En la parte derecha presenta un gráfico de barras de las primeras cuotas por cantones de la empresa y el año seleccionado.

Puede obtener el gráfico de un determinado cantón presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

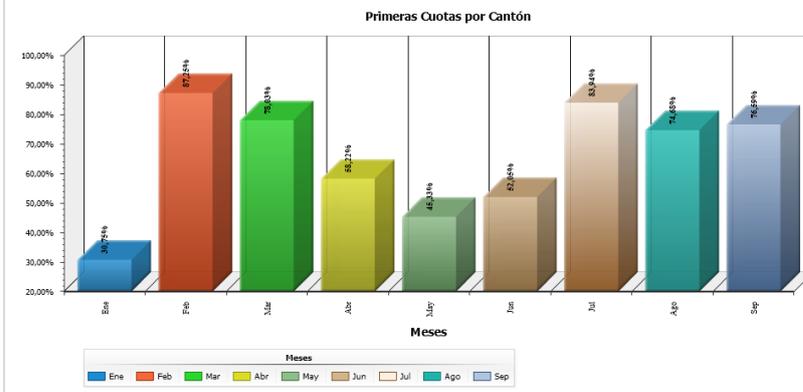


Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Canton Grafico Primeras Cuotas x Canton

Primeras Cuotas x Canton

* Año: 2013 * Canton: CUENCA



Primeras Cuotas por Agencia

En esta opción generará el reporte del indicador de las primeras cuotas por agencias y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Agencia

Reporte Primeras Cuotas x Agencia Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013 Consultar

Q- Go Actions

Agencia	Nombre Agencia	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
	AMB CEVALLOS	98.04	96.38	94.32	97.61	87.05	92.74	93.83	96.42	95.03	-	-	-
	BBH BABAHOYO	100.00	97.11	97.52	96.62	98.44	94.11	97.58	96.72	99.84	-	-	-
	COT SALCEDO	100.00	95.38	98.89	100.00	97.53	97.92	97.53	99.63	100.00	-	-	-
	CUE AV DE LAS AMERICAS	100.00	92.81	90.96	-	-	-	-	-	-	-	-	-
	CUE LAMAR	-	-	-	97.07	96.64	78.71	97.23	88.43	86.59	-	-	-
	CUE PUERTO	93.62	85.44	86.98	96.95	89.80	88.03	100.00	66.38	76.90	-	-	-
	CUE EL ARENAL	97.10	93.77	90.20	88.70	90.12	87.22	95.04	95.10	94.01	-	-	-
	CUE EL ARENAL 2	-	-	-	-	-	100.00	92.93	94.60	98.28	-	-	-
	CUE LUIS CORDERO	96.47	94.30	93.85	98.11	95.55	95.98	96.57	97.99	96.01	-	-	-
	CUE MALL DEL RIO	97.71	93.90	96.29	87.61	97.77	90.07	86.04	88.23	77.65	-	-	-
	CUE MAYOREO CUENCA	-	-	-	-	-	67.72	84.02	-	-	-	-	-
	DAU DAULE	100.00	92.11	89.87	95.44	90.71	93.45	93.99	93.81	80.75	-	-	-
	GRD GUARANDA	97.56	92.95	96.80	100.00	100.00	96.27	95.80	93.83	98.85	-	-	-
	GYE DURAN III	96.82	91.98	97.42	98.09	96.65	81.07	96.67	97.08	97.50	-	-	-
	GYE FLORIDA II	100.00	97.02	94.37	96.01	96.39	93.32	97.20	98.12	89.78	-	-	-

1 - 15

Reporte Primeras Cuotas x Agencia

En la parte derecha presenta un gráfico de barras de las primeras cuotas por agencias de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada agencia presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:

Home Ventas Cobranzas

Home Cobranzas Primeras Cuotas Reporte Primeras Cuotas x Agencia Grafico Reporte Primeras Cuotas x Agencia

Reporte Primeras Cuotas x Agencia

* Año: 2013 * Agencia: CUE LUIS CORDERO

Primeras Cuotas por Agencia

Primeras Cuotas por Línea

En esta opción generará el reporte del indicador de las primeras cuotas por líneas y por meses, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

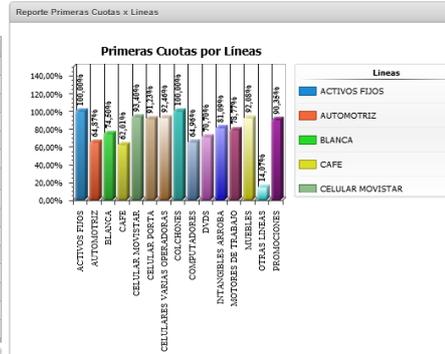
Home > Cobranzas > Primeras Cuotas > Reporte Primeras Cuotas x Línea

Reporte Primeras Cuotas x Líneas Filtro

* Empresa: CORPORACION JARRIN HERRE * Año: 2013 Consultar

Q- Go Actions

Modelo Línea	Nombre Línea	Ene. %	Feb. %	Mar. %	Abr. %	May. %	Jun. %	Jul. %	Ago. %	Sep. %	Oct. %	Nov. %	Dic. %
ACTIVOS FIJOS		-	100.00	-	-	-	-	-	-	-	-	-	-
AUTOMOTRIZ		61.11	93.20	94.68	97.41	96.50	67.06	62.38	58.45	65.51	-	-	-
BLANCA		95.47	89.57	95.42	85.78	85.57	25.30	89.00	83.02	78.44	-	-	-
CAFE		29.09	95.11	70.50	45.33	34.18	41.89	94.84	93.23	94.03	-	-	-
CELULAR MOVISTAR		95.98	92.79	93.95	91.93	93.15	94.36	90.68	92.63	94.87	-	-	-
CELULAR PORTA		-	-	-	91.57	83.09	82.51	92.53	93.27	96.25	-	-	-
CELULARES VARIAS OPERADORAS		93.63	93.46	93.88	88.73	93.22	91.58	94.36	89.40	91.51	-	-	-
COLCHONES		-	100.00	100.00	-	-	-	-	-	-	-	-	-
COMPUTADORES		10.57	91.52	70.99	54.74	88.85	61.23	91.36	94.88	91.73	-	-	-
DVDS		22.19	58.08	87.19	92.23	93.37	33.39	79.00	92.39	21.68	-	-	-
INTANGIBLES ARROBA		-	100.00	-	100.00	66.29	100.00	100.00	92.39	21.68	-	-	-
MOTORES DE TRABAJO		100.00	-	0.00	-	-	-	-	92.73	67.49	-	-	-
MUEBLES		92.27	90.05	99.97	67.11	100.00	100.00	87.12	100.00	100.00	-	-	-
OTRAS LINEAS		-	-	-	-	-	-	-	0.89	32.51	-	-	-
PROMOCIONES		100.00	100.00	100.00	100.00	24.02	96.75	100.00	90.64	100.00	-	-	-



En la parte derecha presenta un gráfico de barras de las primeras cuotas por líneas de la empresa y el año seleccionado.

Puede obtener el gráfico de una determinada línea presionando el botón que se encuentra en la primera columna del reporte, a continuación presentará la siguiente pantalla:



Home Ventas Cobranzas

Home > Cobranzas > Primeras Cuotas > Reporte Primeras Cuotas x Línea > Gráfico Primeras Cuotas x Líneas

Primeras Cuotas x Líneas
* Año: 2013 * Línea: BLANCA

