**Manual Problemas en BD**

**Optimizar BD**

**Belcorp**

Menú Principal

[1. Objetivos 3](#_Toc246416207)

[2. BD SSiCC Producción 4](#_Toc246416208)

[2.1. Situación actual Procedimientos y Paquetes de BD 4](#_Toc246416209)

[2.2. Context Switch en los SELECT 5](#_Toc246416210)

[2.3. Paquetes de Interfaces (Uso de UTL\_FILE) 6](#_Toc246416211)

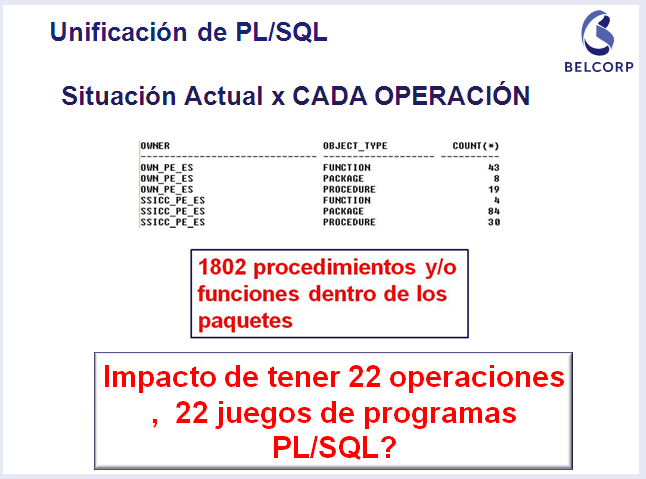
[2.4. Paquetes de Procesos 10](#_Toc246416212)

# Objetivos

El presente documento tiene como finalidad servir como guía para optimizar el código de la Aplicación SSiCC a nivel Oracle, de forma que se optimizen los procesos mencionados

# BD SSiCC Producción

## Situación actual Procedimientos y Paquetes de BD



Actualmente en los ambientes Productivos tenemos los siguientes tipos de Paquetes de BD:

* Paquetes relacionados al Envío y Recepción de Interfaces: Dichos paquetes tienen como características principales la ejecución de un Cursor la cual sus registros son procesados por el **UTL\_FILE** para generar un archivo de texto sin formato temporal.

Actualmente existen 25 paquetes de BD de este tipo

Ejemplo:



* Paquetes relacionados a Generación de data para Reportes pesados: Generalmente estos tipos de paquetes contiene lógica de negocio, la cual tiene como finalidad generar data para ser mostrada en un reporte. Para estos casos se estila usar tablas temporales las cuales serán invocadas en el reporte.

Actualmente existen 10 paquetes de BD de este tipo

Ejemplo:



* Paquetes relacionados a Ejecución de Procesos: Dichos paquetes contiene lógica de negocio necesaria para la ejecución de procesos pesados, la cual implica gran demanda de procesamiento

Actualmente existen 35 paquetes de BD de este tipo



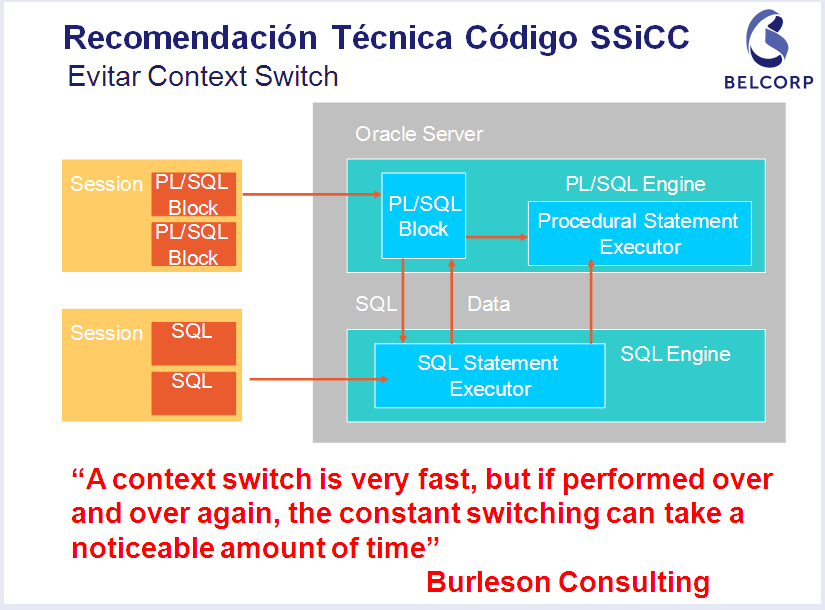
* Paquetes relacionados a Repositorio de Funciones Generales: Dichos paquetes contiene funciones generales utilizadas en los diferentes paquetes de BD.

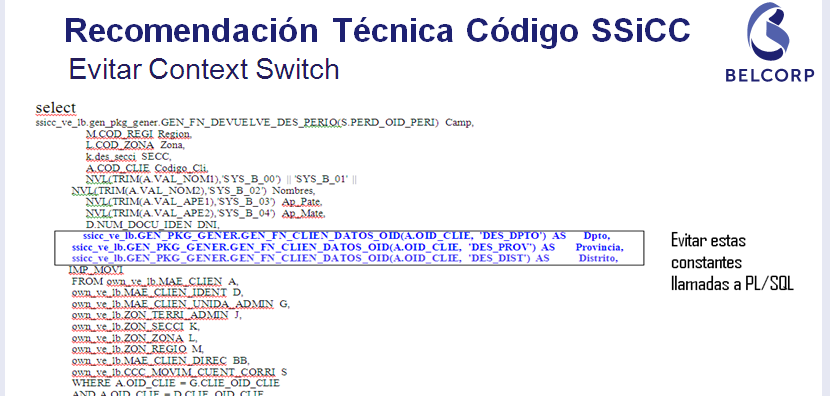
Actualmente existen 35 paquetes de BD de este tipo

**Preguntas:**

* Es conveniente usar paquetes de BD para lógica de negocio?
* Existen comentarios indicando que usar paquetes de BD es una mala práctica de programación. Es correcto esto?

## Context Switch en los SELECT





## Paquetes de Interfaces (Uso de UTL\_FILE)

Ejemplo:

**PROCEDURE** INT\_PR\_BEL\_FACTU\_CABEC  
 (psCodigoPais **VARCHAR2**,  
 psCodigoSistema **VARCHAR2**,  
 psCodigoInterfaz **VARCHAR2**,  
 psNombreArchivo **VARCHAR2**,  
 psCodigoMarca **VARCHAR2** ,  
 psCodigoCanal **VARCHAR2** ,  
 psCodigoAcceso **VARCHAR2** ,  
 psCodigoPeriodo **VARCHAR2** ,  
 psFechaFacturacion **VARCHAR2**  
 )  
  
**IS**  
 **CURSOR** c\_interfaz **IS**  
 **SELECT** **DISTINCT**  
 CABECERA.ANYCAM **AS** ANYCAM,  
 CABECERA.CODCLI **AS** CODCLI,  
 PED\_SOLIC\_CABEC.VAL\_NUME\_SOLI **AS** NUMFAC,  
 CABECERA.TIPPED **AS** TIPPED,  
 TO\_CHAR(CABECERA.FECEMI, 'YYYYMMDD') **AS** FECFAC,  
 CABECERA.TIPDOC **AS** TIPDOC  
 **FROM**  
 (**SELECT** NVL(SEG\_PERIO\_CORPO.COD\_PERI, ' ') **AS** ANYCAM,  
 NVL(MAE\_CLIEN.COD\_CLIE, ' ') **AS** CODCLI,  
 NVL(PED\_SOLIC\_CABEC.VAL\_NUME\_SOLI, 0) **AS** NUMFAC,  
 'N' **AS** TIPPED,  
 PED\_SOLIC\_CABEC.FEC\_FACT **AS** FECEMI,  
 (**CASE** FAC\_TIPO\_DOCUM\_LEGAL.IND\_TIPO\_DOCU  
 **WHEN** 'B' **THEN** '01'  
 **WHEN** 'F' **THEN** '02'  
 **END**) **AS** TIPDOC,  
 PED\_SOLIC\_CABEC.OID\_SOLI\_CABE,  
 PED\_SOLIC\_CABEC.SOCA\_OID\_SOLI\_CABE  
 **FROM** PED\_SOLIC\_CABEC,  
 SEG\_PAIS,  
 SEG\_MARCA,  
 SEG\_CANAL,  
 CRA\_PERIO,  
 SEG\_PERIO\_CORPO,  
 SEG\_TIPO\_PERIO,  
 SEG\_SUBAC,  
 SEG\_ACCES,  
 PED\_ESTAD\_SOLIC,  
 FAC\_TIPO\_DOCUM,  
 FAC\_TIPO\_DOCUM\_LEGAL,  
 PED\_TIPO\_SOLIC\_PAIS,  
 PED\_TIPO\_SOLIC,  
 MAE\_CLIEN  
 **WHERE** ((SEG\_PAIS.COD\_PAIS = psCodigoPais)  
 **AND** (SEG\_MARCA.COD\_MARC = psCodigoMarca)  
 **AND** (SEG\_CANAL.COD\_CANA = psCodigoCanal)  
 **AND** (SEG\_ACCES.COD\_ACCE = psCodigoAcceso)  
 **AND** (SEG\_PERIO\_CORPO.COD\_PERI = psCodigoPeriodo)  
 **AND** (PED\_SOLIC\_CABEC.FEC\_FACT = TO\_DATE(psFechaFacturacion,'DD/MM/YYYY'))  
 **AND** (PED\_SOLIC\_CABEC.IND\_TS\_NO\_CONSO = 1)  
 *-- AND (PED\_SOLIC\_CABEC.IND\_OC = 1)*  
 **AND** ((COD\_TIPO\_SOLI **LIKE** 'SE%') **OR** (COD\_TIPO\_SOLI = 'SOC'))  
 **AND** (PED\_SOLIC\_CABEC.IND\_PEDI\_PRUE = 0)  
 **AND** (PED\_ESTAD\_SOLIC.COD\_ESTA\_SOLI <> 'AN')  
 **AND** (PED\_TIPO\_SOLIC.IND\_DEVO = 0)  
 **AND** (PED\_TIPO\_SOLIC.IND\_ANUL = 0)  
 **AND** (FAC\_TIPO\_DOCUM\_LEGAL.IND\_ACTI\_DOCU = 1)  
 **AND** (PED\_SOLIC\_CABEC.PAIS\_OID\_PAIS = SEG\_PAIS.OID\_PAIS)  
 **AND** (PED\_SOLIC\_CABEC.PERD\_OID\_PERI = CRA\_PERIO.OID\_PERI)  
 **AND** (CRA\_PERIO.PAIS\_OID\_PAIS = SEG\_PAIS.OID\_PAIS)  
 **AND** (CRA\_PERIO.MARC\_OID\_MARC = SEG\_MARCA.OID\_MARC)  
 **AND** (CRA\_PERIO.CANA\_OID\_CANA = SEG\_CANAL.OID\_CANA)  
 **AND** (CRA\_PERIO.PERI\_OID\_PERI = SEG\_PERIO\_CORPO.OID\_PERI)  
 **AND** (SEG\_CANAL.TIPE\_OID\_TIPO\_PERI = SEG\_TIPO\_PERIO.OID\_TIPO\_PERI)  
 **AND** (SEG\_PERIO\_CORPO.TIPE\_OID\_TIPO\_PERI = SEG\_TIPO\_PERIO.OID\_TIPO\_PERI)  
 **AND** (PED\_SOLIC\_CABEC.SBAC\_OID\_SBAC = SEG\_SUBAC.OID\_SBAC)  
 **AND** (SEG\_SUBAC.ACCE\_OID\_ACCE = SEG\_ACCES.OID\_ACCE)  
 **AND** (PED\_SOLIC\_CABEC.ESSO\_OID\_ESTA\_SOLI = PED\_ESTAD\_SOLIC.OID\_ESTA\_SOLI)  
 **AND** (PED\_SOLIC\_CABEC.TIDO\_OID\_TIPO\_DOCU = FAC\_TIPO\_DOCUM.OID\_TIPO\_DOCU)  
 **AND** (FAC\_TIPO\_DOCUM.COD\_TIPO\_DOCU = FAC\_TIPO\_DOCUM\_LEGAL.COD\_TIPO\_DOCU)  
 **AND** (PED\_SOLIC\_CABEC.TSPA\_OID\_TIPO\_SOLI\_PAIS = PED\_TIPO\_SOLIC\_PAIS.OID\_TIPO\_SOLI\_PAIS)  
 **AND** (PED\_TIPO\_SOLIC\_PAIS.PAIS\_OID\_PAIS = SEG\_PAIS.OID\_PAIS)  
 **AND** (PED\_TIPO\_SOLIC\_PAIS.TSOL\_OID\_TIPO\_SOLI = PED\_TIPO\_SOLIC.OID\_TIPO\_SOLI)  
 **AND** (PED\_SOLIC\_CABEC.CLIE\_OID\_CLIE = MAE\_CLIEN.OID\_CLIE))  
 ) CABECERA, PED\_SOLIC\_CABEC  
 **WHERE** CABECERA.SOCA\_OID\_SOLI\_CABE = PED\_SOLIC\_CABEC.OID\_SOLI\_CABE  
 **ORDER** **BY** NUMFAC ;  
  
 **TYPE** interfazCab **IS** **RECORD**  
 (  
 anyoCampanya **VARCHAR2**(6) ,  
 codigoCliente **VARCHAR2**(15) ,  
 numeroFactura **VARCHAR2**(10) ,  
 tipoPedido **VARCHAR2**(1) ,  
 fechaFactura **VARCHAR2**(8) ,  
 tipoDocumento **VARCHAR2**(2)  
 );  
  
 **TYPE** interfazCabTab **IS** **TABLE** **OF** interfazCab ;  
 interfazRecord interfazCabTab;  
  
  
 */\* Variables usadas para la generacion del archivo de texto \*/*  
 lsDirTempo BAS\_INTER.DIR\_TEMP%**TYPE**;  
 W\_FILAS **NUMBER** := 1000 ;  
 v\_handle UTL\_FILE.FILE\_TYPE;  
 W\_DESC **VARCHAR2**(200);  
 lsLinea **VARCHAR2**(1000);  
 lsLineaCabecera **VARCHAR2**(1000);  
 lsNombreArchivo **VARCHAR2**(50);  
 lbAbrirUtlFile **BOOLEAN**;  
**BEGIN**  
  
 */\* Generando Archivo de Texto (Detalle) \*/*  
 lbAbrirUtlFile := **TRUE**;  
 **OPEN** c\_interfaz;  
 **LOOP**  
 **FETCH** c\_interfaz **BULK** **COLLECT** **INTO** interfazRecord **LIMIT** W\_FILAS;  
 */\* Procedimiento inicial para generar interfaz \*/*  
 **IF** lbAbrirUtlFile **THEN**  
 GEN\_PKG\_INTER\_ARCHI.GEN\_PR\_INICI\_INTER(psCodigoPais, psCodigosistema, psCodigoInterfaz,  
 psNombreArchivo, lsDirTempo, lsNombreArchivo, V\_HANDLE);  
 lbAbrirUtlFile := **FALSE**;  
 **END** **IF**;  
  
 **IF** interfazRecord.**COUNT** > 0 **THEN**  
 **FOR** x **IN** interfazRecord.**FIRST** .. interfazRecord.**LAST** **LOOP**  
 lsLinea :=  
 interfazRecord(x).anyoCampanya ||';'||  
 interfazRecord(x).codigoCliente ||';'||  
 interfazRecord(x).numeroFactura ||';'||  
 interfazRecord(x).tipoPedido ||';'||  
 interfazRecord(x).fechaFactura ||';'||  
 interfazRecord(x).tipoDocumento ;  
  
 UTL\_FILE.PUT\_LINE (V\_HANDLE, lslinea );  
  
 **END** **LOOP**;  
 **END** **IF**;  
 **EXIT** **WHEN** c\_interfaz%**NOTFOUND**;  
 **END** **LOOP**;  
 **CLOSE** c\_interfaz;  
  
 **IF** **NOT** lbAbrirUtlFile **THEN**  
 utl\_file.fclose(V\_HANDLE);  
  
 */\* Procedimiento final para generar interfaz \*/*  
 GEN\_PKG\_INTER\_ARCHI.GEN\_PR\_FINAL\_INTER('SICC\_DIR', lsDirTempo, psNombreArchivo, lsNombreArchivo);  
 **END** **IF**;  
  
 **RETURN**;  
  
**EXCEPTION**  
**WHEN** **OTHERS** **THEN**  
 ln\_sqlcode := **SQLCODE**;  
 ls\_sqlerrm := substr(**sqlerrm**,1,1000);  
 RAISE\_APPLICATION\_ERROR(-20123, 'ERROR INT\_PR\_BEL\_FACTU\_CABEC: '||ls\_sqlerrm);  
  
**END** INT\_PR\_BEL\_FACTU\_CABEC;

## Paquetes de Procesos

Ejemplo:

**PROCEDURE** COM\_PR\_GENER\_LISTA\_COMIS\_INGRE( psCodPais **VARCHAR2**,  
 psCodMarca **VARCHAR2**,  
 psCodCanal **VARCHAR2**,  
 psCodComision **VARCHAR2**,  
 psCodPeriodo **VARCHAR2**,  
 psAnioTramo **VARCHAR2**,  
 psNumTramo **VARCHAR2**,  
 psCodTipoCalculo **VARCHAR2**  
 ) **IS**  
 */\* Declaracion de Variables \*/*  
 lnIdPais SEG\_PAIS.OID\_PAIS%**TYPE**;  
 lnIdCanal SEG\_CANAL.OID\_CANA%**TYPE**;  
 lnIdMarca SEG\_MARCA.OID\_MARC%**TYPE**;  
 lnIdUnidadAdm MAE\_CLIEN\_UNIDA\_ADMIN.OID\_CLIE\_UNID\_ADMI%**TYPE**;  
 lsCodPeriMas2 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lsCodPeriSiguiente1 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lsCodPeri1 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lsCodPeri2 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lsCodPeri3 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lsCodPeri4 SEG\_PERIO\_CORPO.COD\_PERI%**TYPE**;  
 lnIdPeriMas2 CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeriSiguiente1 CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeriActual CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeri1 CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeri2 CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeri3 CRA\_PERIO.OID\_PERI%**TYPE**;  
 lnIdPeri4 CRA\_PERIO.OID\_PERI%**TYPE**;  
  
 lnIdProcCreacion CCC\_PROCE.OID\_PROC%**TYPE**;  
  
 lnIdSubprocCreacion CCC\_SUBPR.OID\_SUBP%**TYPE**;  
 lnIdEstado PED\_ESTAD\_SOLIC.OID\_ESTA\_SOLI%**TYPE**;  
 lnIdZona ZON\_ZONA.OID\_ZONA%**TYPE**;  
 lnIdSeccion ZON\_SECCI.OID\_SECC%**TYPE**;  
 ldFechaIniPeriodo **DATE**;  
 ldFechaFinPeriodo **DATE**;  
 lsNomLiderZona **VARCHAR2**(500);  
 lsNomLiderSeccion **VARCHAR2**(500);  
 lsCodZona ZON\_ZONA.COD\_ZONA%**TYPE**;  
 lsCodSecc ZON\_SECCI.COD\_SECC%**TYPE**;  
 lsCodRegi ZON\_REGIO.COD\_REGI%**TYPE**;  
 lsCodLiderZona MAE\_CLIEN.COD\_CLIE%**TYPE**;  
 lsCodLiderSeccion MAE\_CLIEN.COD\_CLIE%**TYPE**;  
  
 PORCENTAJE\_COMISION **NUMBER**:= 0.05;  
 lnIdComision **NUMBER**;  
 lnSumaComision **NUMBER**;  
 lnSumaImporteNeto **NUMBER**;  
 lnContaImporteMayorCero **NUMBER**;  
 lnSumaImporteCobranza **NUMBER**;  
 lsMensaje **VARCHAR2**(255);  
 lnValor **NUMBER**;  
 regRegistro tRegTablaClienComision;  
 tablaRegistro tTablaClienteComision;  
 lnContador **NUMBER**;  
 lnPagina **NUMBER**;  
 lbDemandaAnticipada **BOOLEAN**;  
 lnNumDiasDemandaTramo1 **NUMBER**;  
 lnNumDiasDemandaTramo2 **NUMBER**;  
 lv\_procesar **BOOLEAN**;  
 psCodTipoComision **VARCHAR2**(1);  
  
 **CURSOR** cursorComision(vnIdComision **NUMBER**)  
 **IS**  
 **SELECT**  
 B.ZORG\_OID\_REGI  
 **FROM**  
 COM\_COMIS A,  
 COM\_COMIS\_CLIEN B  
 **WHERE** A.OID\_COMI = vnIdComision  
 **AND** A.OID\_COMI = B.COMI\_OID\_COMI  
 **AND** B.ZORG\_OID\_REGI **IS** **NOT** **NULL**;  
  
 **CURSOR** cursorRegistro(vnIdPeri **NUMBER**, vnIdComision **NUMBER**, vnIdRegion **NUMBER**)  
 **IS**  
 **SELECT** **DISTINCT**  
 C.OID\_CLIE,  
 C.COD\_CLIE,  
 **TRIM**(NVL(C.VAL\_APE1,' ')) || ' ' ||  
 **TRIM**(NVL(C.VAL\_APE2,' ')) || ' ' ||  
 **TRIM**(NVL(C.VAL\_NOM1,' ')) **AS** NOM\_CLIE,  
 L.COD\_SUBG\_VENT,  
 E.COD\_REGI,  
 F.COD\_ZONA,  
 G.COD\_SECC,  
 K.COD\_TERR,  
 F.OID\_ZONA,  
 G.OID\_SECC,  
 D.ZTAD\_OID\_TERR\_ADMI **AS** OID\_TERR\_ADMI,  
 0 **AS** IMP\_PAGO\_ACTU,  
 0 **AS** IMP\_PAGO\_1,  
 0 **AS** IMP\_PAGO\_2,  
 0 **AS** IMP\_PAGO\_3,  
 0 **AS** IMP\_PAGO\_4,  
 0 **AS** IMP\_PAGO\_NETO,  
 0 **AS** IMP\_COMI,  
 '' **AS** COD\_LIDE\_ZONA,  
 '' **AS** NOM\_LIDE\_ZONA,  
 '' **AS** COD\_LIDE\_SECC,  
 '' **AS** NOM\_LIDE\_SECC,  
 0 **AS** IMP\_COBR\_1,  
 0 **AS** IMP\_COBR\_2,  
 0 **AS** IMP\_COBR\_3,  
 0 **AS** IMP\_COBR\_4,  
 0 **AS** IMP\_COBR\_ACTU,  
 0 **AS** IMP\_COBR\_TOTA  
 **FROM**  
 COM\_COMIS A,  
 COM\_COMIS\_CLIEN B,  
 MAE\_CLIEN C,  
 MAE\_CLIEN\_UNIDA\_ADMIN D,  
 ZON\_REGIO E,  
 ZON\_ZONA F,  
 ZON\_SECCI G,  
 ZON\_TERRI\_ADMIN H,  
 MAE\_CLIEN\_HISTO\_ESTAT I,  
 MAE\_ESTAT\_CLIEN J,  
 ZON\_TERRI K,  
 ZON\_SUB\_GEREN\_VENTA L  
 **WHERE** A.OID\_COMI = vnIdComision  
 **AND** I.PERD\_OID\_PERI = vnIdPeri  
 **AND** E.OID\_REGI = vnIdRegion  
 **AND** C.OID\_CLIE = I.CLIE\_OID\_CLIE  
  
 **AND** J.COD\_ESTA\_CLIE = '02'  
 **AND** J.OID\_ESTA\_CLIE = I.ESTA\_OID\_ESTA\_CLIE  
 **AND** A.OID\_COMI = B.COMI\_OID\_COMI  
 **AND** E.OID\_REGI = B.ZORG\_OID\_REGI  
 **AND** E.OID\_REGI = F.ZORG\_OID\_REGI  
 **AND** F.OID\_ZONA = G.ZZON\_OID\_ZONA  
 **AND** G.OID\_SECC = H.ZSCC\_OID\_SECC  
 **AND** H.OID\_TERR\_ADMI = D.ZTAD\_OID\_TERR\_ADMI  
 **AND** C.OID\_CLIE = D.CLIE\_OID\_CLIE  
 **AND** K.OID\_TERR = H.TERR\_OID\_TERR  
 **AND** L.OID\_SUBG\_VENT = E.ZSGV\_OID\_SUBG\_VENT  
 **ORDER** **BY** E.COD\_REGI, F.COD\_ZONA, G.COD\_SECC, K.COD\_TERR, C.COD\_CLIE ;  
  
**BEGIN**  
 lv\_procesar:=**FALSE**;  
 */\* Borrando tabla temporal de Comision x Ingresos \*/*  
 **DELETE** **FROM** COM\_REPOR\_COMIS\_INGRE A  
 **WHERE** A.COD\_PERI\_ACTU = psCodPeriodo  
 **AND** A.COD\_COMI = psCodComision;  
  
 */\* Obteniendo oids \*/*  
 lnIdPais := gen\_pkg\_gener.gen\_fn\_devuelve\_id\_pais(psCodPais);  
 lnIdCanal := gen\_pkg\_gener.gen\_fn\_devuelve\_id\_canal(psCodCanal);  
 lnIdMarca := gen\_pkg\_gener.gen\_fn\_devuelve\_id\_marca(psCodMarca);  
 lnIdPeriActual := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(psCodPeriodo, lnIdMarca, lnIdCanal);  
 lnIdComision := COM\_PKG\_REPOR.COM\_FN\_DEVUE\_ULTI\_OID\_COMI(psCodComision);  
  
 */\* obtenemos id subproceso banco\*/*  
 **DELETE** **FROM** PER\_GTT\_TABLA\_REPOS;  
 **INSERT** **INTO** PER\_GTT\_TABLA\_REPOS(CAM\_NUME, CAM\_CADE)  
 **SELECT** C.OID\_SUBP, D.TIP\_COMI  
 **FROM** SEG\_PAIS A,  
 CCC\_PROCE B,  
 CCC\_SUBPR C,  
 COM\_COMIS\_SUBPR\_PAGO D  
 **WHERE** A.OID\_PAIS = lnIdPais  
 **AND** B.PAIS\_OID\_PAIS = A.OID\_PAIS  
 **AND** B.OID\_PROC = C.CCPR\_OID\_PROC  
 **AND** D.COD\_PAIS = psCodPais  
 **AND** D.COD\_COMI = COMISION\_INGRESO  
 **AND** B.COD\_PROC = D.COD\_PROC  
 **AND** C.COD\_SUBP = D.COD\_SUBP;  
  
 */\* obtenemos id subproceso creacion \*/*  
 lnIdProcCreacion := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_PROCESO(lnIdPais, 'CCC001');  
 **BEGIN**  
 **SELECT** a.oid\_subp  
 **INTO** lnIdSubprocCreacion  
 **FROM** CCC\_SUBPR A  
 **WHERE**  
 a.ccpr\_oid\_proc = lnIdProcCreacion **AND**  
 a.cod\_subp = 1;  
 **EXCEPTION**  
 **WHEN** no\_Data\_found **THEN**  
 RAISE\_APPLICATION\_ERROR(-20123, 'No se encontro el ID del Subproceso de Creacion' );  
 **END**;  
  
 */\* Obteniendo periodos anteriores \*/*  
 lsCodPeriSiguiente1 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, 1);  
 lnIdPeriSiguiente1 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeriSiguiente1, lnIdMarca, lnIdCanal);  
 lsCodPeriMas2 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, 2);  
 lnIdPeriMas2 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeriMas2, lnIdMarca, lnIdCanal);  
  
 lsCodPeri1 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, -1);  
 lnIdPeri1 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeri1, lnIdMarca, lnIdCanal);  
 lsCodPeri2 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, -2);  
 lnIdPeri2 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeri2, lnIdMarca, lnIdCanal);  
 lsCodPeri3 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, -3);  
 lnIdPeri3 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeri3, lnIdMarca, lnIdCanal);  
 lsCodPeri4 := PER\_PKG\_REPOR\_PERCE.PER\_FN\_OBTIE\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, -4);  
 lnIdPeri4 := gen\_pkg\_gener.GEN\_FN\_DEVUELVE\_ID\_CRA\_PERIO(lsCodPeri4, lnIdMarca, lnIdCanal);  
  
 */\* Obteniendo fecha inicial y fecha final del periodo del proceso \*/*  
 lnValor := COM\_FN\_DEVUE\_RANGO\_FECHA\_PERIO(psCodPeriodo, lnIdPais, lnIdMarca, lnIdCanal, ldFechaIniPeriodo, ldFechaFinPeriodo, lsMensaje);  
 **IF** lnValor = -1 **THEN**  
 RAISE\_APPLICATION\_ERROR(-20123, lsMensaje);  
 **END** **IF**;  
  
 */\* Borrando informaci¿n en la tabla COM\_COMIS\_PERI\_CALCU \*/*  
 **DELETE** **FROM** COM\_COMIS\_PERIO\_CALCU  
 **WHERE** PERI\_COD\_PERI = psCodPeriodo  
 **AND** COD\_COMI = psCodComision  
 **AND** IND\_COMI = '01';  
  
 */\* obteniendo id de estado de pedidos \*/*  
 **SELECT** ES.OID\_ESTA\_SOLI  
 **INTO** lnIdEstado  
 **FROM** PED\_ESTAD\_SOLIC ES  
 **WHERE**  
 ES.COD\_ESTA\_SOLI = 'AN';  
  
 */\* Recorriendo Lista de Clientes \*/*  
 lnIdZona := -1;  
 lnIdSeccion := -1;  
 lsNomLiderZona := '';  
 lsNomLiderSeccion := '';  
 lnSumaComision := 0;  
 lnSumaImporteNeto := 0;  
 lnSumaImporteCobranza := 0;  
 lnPagina := 0;  
 lbDemandaAnticipada := **TRUE**;  
  
 **FOR** curComis **IN** cursorComision(lnIdComision) **LOOP**  
 lnPagina := lnPagina + 1;  
 **OPEN** cursorRegistro(lnIdPeri4, lnIdComision, curComis.Zorg\_Oid\_Regi);  
 **LOOP**  
 **FETCH** cursorRegistro **BULK** **COLLECT** **INTO** tablaRegistro **LIMIT** W\_FILAS;  
 **IF** tablaRegistro.**COUNT** > 0 **THEN**  
 **FOR** x **IN** tablaRegistro.**FIRST** .. tablaRegistro.**LAST** **LOOP**  
 regRegistro := tablaRegistro(x);  
 lnContador := x;  
  
 lnIdUnidadAdm := COM\_FN\_DEVUE\_OID\_UNADM\_HISTO(regRegistro.OID\_CLIE, regRegistro.OID\_TERR\_ADMI, psCodPeriodo);  
 **IF** lnIdUnidadAdm != -1 **THEN**  
  
  
 */\* Verificando Demanda Anticipada \*/*  
 lbDemandaAnticipada := **TRUE**;  
 **BEGIN**  
 **SELECT** A.NUM\_DIAS\_TRA1, A.NUM\_DIAS\_TRA2  
 **INTO**  
 lnNumDiasDemandaTramo1, lnNumDiasDemandaTramo2  
 **FROM** COM\_ZONA\_DEMAN\_ANTIC A  
 **WHERE** A.COD\_PAIS = psCodPais  
 **AND** A.COD\_ZONA = regRegistro.COD\_ZONA;  
 **EXCEPTION**  
 **WHEN** no\_data\_found **THEN**  
 lbDemandaAnticipada := **FALSE**;  
 **END**;  
  
 */\* Obteniendo importes \*/*  
 regRegistro.IMP\_PAGO\_ACTU := COM\_FN\_MONTO\_COMIS\_INGRE(lbDemandaAnticipada, lnIdPais, regRegistro.OID\_CLIE, lnIdPeriMas2, lnIdPeriSiguiente1, lnIdPeriActual, lnIdPeri1, lnIdSubprocCreacion, regRegistro.OID\_ZONA, lnIdEstado);  
 regRegistro.IMP\_PAGO\_1 := COM\_FN\_MONTO\_COMIS\_INGRE(lbDemandaAnticipada, lnIdPais, regRegistro.OID\_CLIE, lnIdPeriSiguiente1, lnIdPeriActual, lnIdPeri1, lnIdPeri2, lnIdSubprocCreacion, regRegistro.OID\_ZONA, lnIdEstado);  
 regRegistro.IMP\_PAGO\_2 := COM\_FN\_MONTO\_COMIS\_INGRE(lbDemandaAnticipada, lnIdPais, regRegistro.OID\_CLIE, lnIdPeriActual, lnIdPeri1, lnIdPeri2, lnIdPeri3, lnIdSubprocCreacion, regRegistro.OID\_ZONA, lnIdEstado);  
 regRegistro.IMP\_PAGO\_3 := COM\_FN\_MONTO\_COMIS\_INGRE(lbDemandaAnticipada, lnIdPais, regRegistro.OID\_CLIE, lnIdPeri1, lnIdPeri2, lnIdPeri3, lnIdPeri4, lnIdSubprocCreacion, regRegistro.OID\_ZONA, lnIdEstado);  
 regRegistro.IMP\_PAGO\_4 := COM\_FN\_MONTO\_COMIS\_INGRE(lbDemandaAnticipada, lnIdPais, regRegistro.OID\_CLIE, lnIdPeri2, lnIdPeri3, lnIdPeri4, lnIdPeri4, lnIdSubprocCreacion, regRegistro.OID\_ZONA, lnIdEstado);  
 regRegistro.IMP\_PAGO\_NETO := ROUND(regRegistro.IMP\_PAGO\_ACTU + regRegistro.IMP\_PAGO\_1 + regRegistro.IMP\_PAGO\_2 + regRegistro.IMP\_PAGO\_3 + regRegistro.IMP\_PAGO\_4, 2);  
  
 */\* Encontrando Lider de Seccion \*/*  
 **IF** regRegistro.OID\_SECC <> lnIdSeccion **THEN**  
  
 */\* Grabando importe en la tabla de Comisiones calculadas \*/*  
 **IF** lsNomLiderSeccion **IS** **NOT** **NULL** **AND** lv\_procesar **THEN**  
 **INSERT** **INTO** COM\_COMIS\_PERIO\_CALCU  
 (PERI\_COD\_PERI, COD\_COMI, COD\_LIDE\_SECC, NOM\_LIDE\_SECC,  
 IND\_COMI, COD\_REGI, COD\_ZONA, COD\_SECC, IMP\_COMI,  
 IMP\_NETO\_SIN\_RECL, IMP\_COMI\_DCTO, FEC\_CALC)  
 **VALUES**  
 (psCodPeriodo, psCodComision , lsCodLiderSeccion, lsNomLiderSeccion,  
 '01', lsCodRegi, lsCodZona, lsCodSecc, lnSumaComision,  
 lnSumaImporteNeto, lnSumaImporteCobranza, **SYSDATE**);  
 **END** **IF**;  
  
 lnSumaComision := 0;  
 lnSumaImporteNeto := 0;  
 lnSumaImporteCobranza := 0;  
 lnIdSeccion := regRegistro.OID\_SECC;  
 lsCodSecc := regRegistro.COD\_SECC;  
 lsCodZona := regRegistro.COD\_ZONA;  
 lsCodRegi := regRegistro.COD\_REGI;  
 lsNomLiderSeccion := COM\_FN\_DEVUE\_RESPO\_UNIAD\_HISTO(  
 lsCodLiderSeccion,  
 ldFechaIniPeriodo, ldFechaFinPeriodo,  
 lnIdPais, regRegistro.COD\_SUBG\_VENT,  
 regRegistro.COD\_REGI,  
 regRegistro.COD\_ZONA,  
 regRegistro.COD\_SECC);  
  
 lv\_procesar:= COM\_FN\_DEVUE\_PROCE\_TUTOR(psCodPais, psCodMarca,psCodCanal, lsCodLiderSeccion,psAnioTramo,psNumTramo,psCodTipoComision,psCodTipoCalculo);  
  
 **END** **IF**;  
 regRegistro.NOM\_LIDE\_SECC := lsNomLiderSeccion;  
  
 */\* Encontrando Lider de Zona \*/*  
 **IF**(lv\_procesar)**THEN**  
 **IF** regRegistro.OID\_ZONA <> lnIdZona **THEN**  
  
 lnIdZona := regRegistro.OID\_ZONA;  
 lsCodZona := regRegistro.COD\_ZONA;  
 lsNomLiderZona := COM\_FN\_DEVUE\_RESPO\_UNIAD\_HISTO(  
 lsCodLiderZona,  
 ldFechaIniPeriodo, ldFechaFinPeriodo,  
 lnIdPais, regRegistro.COD\_SUBG\_VENT,  
 regRegistro.COD\_REGI,  
 regRegistro.COD\_ZONA,  
 **NULL**);  
 **END** **IF**;  
 regRegistro.NOM\_LIDE\_ZONA := lsNomLiderZona;  
  
 */\* Verificando que existan al menos 4 importes mayores a cero \*/*  
 lnContaImporteMayorCero := 0;  
 **IF** regRegistro.IMP\_PAGO\_ACTU > 0 **THEN**  
 lnContaImporteMayorCero := lnContaImporteMayorCero + 1;  
 **END** **IF**;  
 **IF** regRegistro.IMP\_PAGO\_1 > 0 **THEN**  
 lnContaImporteMayorCero := lnContaImporteMayorCero + 1;  
 **END** **IF**;  
 **IF** regRegistro.IMP\_PAGO\_2 > 0 **THEN**  
 lnContaImporteMayorCero := lnContaImporteMayorCero + 1;  
 **END** **IF**;  
 **IF** regRegistro.IMP\_PAGO\_3 > 0 **THEN**  
 lnContaImporteMayorCero := lnContaImporteMayorCero + 1;  
 **END** **IF**;  
 **IF** regRegistro.IMP\_PAGO\_4 > 0 **THEN**  
 lnContaImporteMayorCero := lnContaImporteMayorCero + 1;  
 **END** **IF**;  
 lnSumaImporteNeto := lnSumaImporteNeto + regRegistro.IMP\_PAGO\_NETO;  
  
 */\* Calculando comision \*/*  
 **IF** lnContaImporteMayorCero >= 4 **AND** regRegistro.NOM\_LIDE\_SECC **IS** **NOT** **NULL** **THEN**  
 regRegistro.IMP\_COMI := regRegistro.IMP\_PAGO\_ACTU +  
 regRegistro.IMP\_PAGO\_1 + regRegistro.IMP\_PAGO\_2 +  
 regRegistro.IMP\_PAGO\_3 + regRegistro.IMP\_PAGO\_4;  
 regRegistro.IMP\_COMI := regRegistro.IMP\_COMI \* PORCENTAJE\_COMISION;  
 lnSumaComision := lnSumaComision + regRegistro.IMP\_COMI;  
  
 */\* obtieniendo cobranzas \*/*  
 regRegistro.IMP\_COBR\_1 := COM\_FN\_DEVUE\_COBRA\_CONSU(lsCodPeri1, regRegistro.COD\_CLIE);  
 regRegistro.IMP\_COBR\_2 := COM\_FN\_DEVUE\_COBRA\_CONSU(lsCodPeri2, regRegistro.COD\_CLIE);  
 regRegistro.IMP\_COBR\_3 := COM\_FN\_DEVUE\_COBRA\_CONSU(lsCodPeri3, regRegistro.COD\_CLIE);  
 regRegistro.IMP\_COBR\_4 := COM\_FN\_DEVUE\_COBRA\_CONSU(lsCodPeri4, regRegistro.COD\_CLIE);  
 regRegistro.IMP\_COBR\_ACTU := COM\_FN\_DEVUE\_COBRA\_CONSU(psCodPeriodo, regRegistro.COD\_CLIE);  
 regRegistro.IMP\_COBR\_TOTA := regRegistro.IMP\_COBR\_1 + regRegistro.IMP\_COBR\_2 +  
 regRegistro.IMP\_COBR\_3 + regRegistro.IMP\_COBR\_4 +  
 regRegistro.IMP\_COBR\_ACTU;  
 lnSumaImporteCobranza := lnSumaImporteCobranza + regRegistro.IMP\_COBR\_TOTA;  
 **END** **IF**;  
  
 */\* Insertando data en la tabla detalle de comision de ingreso \*/*  
 **INSERT** **INTO** COM\_REPOR\_COMIS\_INGRE  
 (OID\_CLIE, COD\_COMI, COD\_CLIE, NOM\_CLIE,  
 COD\_SUBG\_VENT, COD\_REGI, COD\_ZONA, COD\_SECC,  
 COD\_TERR, OID\_ZONA, OID\_SECC, OID\_TERR\_ADMI,  
 IMP\_PAGO\_ACTU, IMP\_PAGO\_1, IMP\_PAGO\_2, IMP\_PAGO\_3, IMP\_PAGO\_4,  
 IMP\_PAGO\_NETO, IMP\_COMI, COD\_LIDE\_ZONA, NOM\_LIDE\_ZONA, COD\_LIDE\_SECC, NOM\_LIDE\_SECC,  
 COD\_PERI\_ACTU, COD\_PERI\_1, COD\_PERI\_2, COD\_PERI\_3, COD\_PERI\_4,  
 IMP\_COBR\_1, IMP\_COBR\_2, IMP\_COBR\_3, IMP\_COBR\_4,  
 IMP\_COBR\_ACTU, IMP\_COBR\_TOTA)  
 **VALUES**  
 ( regRegistro.OID\_CLIE, psCodComision, regRegistro.COD\_CLIE, regRegistro.NOM\_CLIE,  
 regRegistro.COD\_SUBG\_VENT, regRegistro.COD\_REGI, regRegistro.COD\_ZONA, regRegistro.COD\_SECC,  
 regRegistro.COD\_TERR, regRegistro.OID\_ZONA, regRegistro.OID\_SECC, regRegistro.OID\_TERR\_ADMI,  
 regRegistro.IMP\_PAGO\_ACTU, regRegistro.IMP\_PAGO\_1, regRegistro.IMP\_PAGO\_2, regRegistro.IMP\_PAGO\_3, regRegistro.IMP\_PAGO\_4,  
 regRegistro.IMP\_PAGO\_NETO, regRegistro.IMP\_COMI,  
 lsCodLiderZona, regRegistro.NOM\_LIDE\_ZONA, lsCodLiderSeccion, regRegistro.NOM\_LIDE\_SECC,  
 psCodPeriodo, lsCodPeri1, lsCodPeri2, lsCodPeri3, lsCodPeri4,  
 regRegistro.IMP\_COBR\_1, regRegistro.IMP\_COBR\_2, regRegistro.IMP\_COBR\_3, regRegistro.IMP\_COBR\_4,  
 regRegistro.IMP\_COBR\_ACTU, regRegistro.IMP\_COBR\_TOTA );  
 **END** **IF**;  
  
 **END** **IF**;  
 **END** **LOOP**;  
 **END** **IF**;  
 **EXIT** **WHEN** cursorRegistro%**NOTFOUND**;  
 **END** **LOOP**;  
 **CLOSE** cursorRegistro;  
  
 **END** **LOOP**;  
 **RETURN** ;  
**EXCEPTION**  
**WHEN** **OTHERS** **THEN**  
 ln\_sqlcode := **SQLCODE**;  
 ls\_sqlerrm := substr(**sqlerrm**,1,250);  
 RAISE\_APPLICATION\_ERROR(-20123, 'ERROR COM\_PR\_GENER\_LISTA\_COMIS\_INGRE: '||lnPagina||' '||lnContador||' '|| ls\_sqlerrm);  
**END** COM\_PR\_GENER\_LISTA\_COMIS\_INGRE;