[](https://www.myperfectresume.com/membership/RegisterGuestUser.aspx?wizard=true&productid=17&utm_source=hloom-com&utm_medium=referral&utm_campaign=word-template)

|  |  |
| --- | --- |
| **Claro Club** | Documento Tecnico |
| Procesos |
|  |  |

Documento Tecnico

de procesos

**Claro Club**

**2019**

**Producido por**

Corporación Sapia.

Contenido

No se encontraron entradas de tabla de contenido.

|  |
| --- |
| REVISIÓN TECNICA |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCEPTO** | 16 – VENCIMIENTO DE PUNTOS | | |
| **FUNCIONALIDAD:** |  | | |
| **SHELL** | | | |
| **RUTA** | /home/usrclaroclub/CLAROCLUB/Interno/Prepago/Shells/SH011\_REPVENPUNT\_modif.sh | **FRECUENCIA EJECUCION** |  |
| **ARCHIVOS A PROCESAR** | | | |
| **RUTA** |  | **ORIGEN DATOS / HORA** |  |
| **PLATAFORMA** | | | |
| **APLICACIÓN** |  | | |
| **SP** |  | | |
| Se ejecuta la Shell: SH011\_REPVENPUNT\_modif.sh   1. Se considera el archivo de configuración importaPtosBajaHFCC.ctl para el inicio del proceso. 2. Se ingresa como parrámetro de entrada la fecha actual y tipo de Cliente “7”para realizar la baja de puntos en el procedimiento PKG\_CC\_PTOSFIJA. ADMPSI\_BAJA\_CC . 3. Valida que el cliente no este de baja.  |  | | --- | | SELECT COUNT(1)  INTO V\_CANT  FROM ADMPT\_TMP\_BAJA\_CC B,  (SELECT DISTINCT P.ADMPV\_COD\_CLI,  SUBSTR(P.ADMPV\_COD\_CLI\_PROD,  1,  INSTR(P.ADMPV\_COD\_CLI\_PROD, '\_') - 1) AS ADMPV\_COD\_CLI\_PROD  FROM ADMPT\_CLIENTEPRODUCTO P) C,  ADMPT\_CLIENTEFIJA D  WHERE B.ADMPD\_FEC\_OPER = K\_FEC\_PROCESO  AND B.ADMPV\_TIP\_CLIENTE = K\_TIPCLIENTE  AND (B.ADMPV\_MSJE\_ERROR IS NULL OR B.ADMPV\_MSJE\_ERROR = '')  AND B.ADMPV\_CUSTCODE = C.ADMPV\_COD\_CLI\_PROD  AND B.ADMPV\_CUSTCODE = vRECPTOBAJA.ADMPV\_CUSTCODE  AND C.ADMPV\_COD\_CLI = D.ADMPV\_COD\_CLI  AND D.ADMPC\_ESTADO = 'B'; |  1. Obtiene los servicios de las cuentas y saldos kardex del cliente a dar de baja de las cuentas servicios que tienen saldo.  |  | | --- | | DECLARE CURSOR KARDEX\_BAJA(vCUENTA VARCHAR2) IS  SELECT KF.ADMPV\_COD\_CLI\_PROD, SUM(KF.ADMPN\_SLD\_PUNTO)  FROM ADMPT\_KARDEXFIJA KF, (SELECT C.ADMPV\_COD\_CLI\_PROD FROM ADMPT\_CLIENTEPRODUCTO C  WHERE C.ADMPV\_ESTADO\_SERV = 'A' AND C.ADMPV\_COD\_CLI\_PROD LIKE vCUENTA||'%' ) CP  WHERE KF.ADMPV\_COD\_CLI\_PROD = CP.ADMPV\_COD\_CLI\_PROD AND KF.ADMPN\_SLD\_PUNTO <> 0  AND KF.ADMPC\_ESTADO = 'A'  GROUP BY KF.ADMPV\_COD\_CLI\_PROD; |  1. Registra el movimiento de salida para cada servicio en el kardex con concepto 69 (BAJA CLIENTE HFC) con estado “B” y tipo punto “C”.  |  | | --- | |  |  1. Actualiza los movimientos que tienen saldo en el Kardex para cada servicio con saldo en 0 y guarda el último saldo que tuvo con el idkardex del movimiento de baja.  |  | | --- | |  |  1. Actualiza la tabla saldo cliente con estado en “B” y saldo en 0 para los servicios activos.  |  | | --- | | UPDATE ADMPT\_SALDOS\_CLIENTEFIJA S  SET S.ADMPC\_ESTPTO\_CC = 'B', S.ADMPN\_SALDO\_CC = 0,  S.ADMPV\_USU\_MOD = K\_USUARIO, S.ADMPD\_FEC\_MOD = SYSDATE  WHERE S.ADMPV\_COD\_CLI\_PROD IN (SELECT C.ADMPV\_COD\_CLI\_PROD FROM ADMPT\_CLIENTEPRODUCTO C  WHERE C.ADMPV\_ESTADO\_SERV = 'A' AND C.ADMPV\_COD\_CLI\_PROD LIKE vRECPTOBAJA.ADMPV\_CUSTCODE||'%' )  AND S.ADMPC\_ESTPTO\_CC = 'A'; |      1. Actualiza la tabla cliente con estado en “B” para los servicios activos.  |  | | --- | | UPDATE ADMPT\_CLIENTEPRODUCTO P  SET P.ADMPV\_ESTADO\_SERV = 'B',  P.ADMPV\_USU\_MOD = K\_USUARIO, P.ADMPD\_FEC\_MOD = SYSDATE  WHERE P.ADMPV\_COD\_CLI\_PROD LIKE vRECPTOBAJA.ADMPV\_CUSTCODE||'%'  AND P.ADMPV\_ESTADO\_SERV = 'A'; |  1. Verifica si el cliente tiene otra cuenta activa.  |  | | --- | | SELECT COUNT(\*) INTO nVALTRANS FROM ADMPT\_CLIENTEPRODUCTO P  WHERE P.ADMPV\_COD\_CLI\_PROD NOT IN (SELECT C.ADMPV\_COD\_CLI\_PROD FROM ADMPT\_CLIENTEPRODUCTO C  WHERE C.ADMPV\_ESTADO\_SERV = 'A' AND C.ADMPV\_COD\_CLI\_PROD LIKE vRECPTOBAJA.ADMPV\_CUSTCODE||'%' )  AND P.ADMPV\_COD\_CLI = vCODCLIENTE  AND P.ADMPV\_ESTADO\_SERV = 'A'; |  1. Si tiene otra cuenta:  |  |  | | --- | --- | | Descripción de la secuencia | Secuencia de migración. | | Registra un movimiento de entrada para la última cuenta activa en el Kardex con concepto 71 (INGRESO POR BAJA CLIENTE HFC) con los puntos a transferir con estado “A” y tipo de punto “C”. | INSERT INTO PCLUB.ADMPT\_KARDEXFIJA (ADMPN\_ID\_KARDEX,ADMPV\_COD\_CLI\_PROD,ADMPV\_COD\_CPTO,  ADMPD\_FEC\_TRANS,ADMPN\_PUNTOS, ADMPC\_TPO\_OPER, ADMPC\_TPO\_PUNTO, ADMPN\_SLD\_PUNTO, ADMPC\_ESTADO,ADMPV\_USU\_REG,ADMPD\_FEC\_REG)  VALUES(PCLUB.ADMPT\_KARDEXFIJA\_SQ.NEXTVAL,vCODCLIPRODAUX, V\_COD\_CPTO2,SYSDATE,  nSUMSALDO,'E', 'C', nSUMSALDO, 'A',K\_USUARIO,SYSDATE); | | Actualiza el saldo cliente de esa cuenta servicio con los puntos transferidos. | UPDATE ADMPT\_SALDOS\_CLIENTEFIJA SC  SET SC.ADMPN\_SALDO\_CC = SC.ADMPN\_SALDO\_CC + nSUMSALDO, SC.ADMPD\_FEC\_MOD = SYSDATE,  SC.ADMPV\_USU\_MOD = K\_USUARIO  WHERE SC.ADMPV\_COD\_CLI\_PROD = vCODCLIPRODAUX; |  1. Si no tiene otra cuenta, le da de baja al cliente.  |  | | --- | | UPDATE ADMPT\_CLIENTEPRODUCTO P  SET P.ADMPV\_COD\_CLI = V\_COD\_CLINUE, P.ADMPD\_FEC\_MOD = SYSDATE,  P.ADMPV\_USU\_MOD = K\_USUARIO  WHERE P.ADMPV\_COD\_CLI = vCODCLIENTE;    UPDATE ADMPT\_CANJEFIJA  SET ADMPV\_COD\_CLI = V\_COD\_CLINUE, ADMPD\_FEC\_MOD = SYSDATE,  ADMPV\_USU\_MOD = K\_USUARIO  WHERE ADMPV\_COD\_CLI = vCODCLIENTE;    UPDATE ADMPT\_CLIENTEFIJA  SET ADMPC\_ESTADO = 'B', ADMPD\_FEC\_MOD = SYSDATE,  ADMPV\_USU\_MOD = K\_USUARIO  WHERE ADMPV\_COD\_CLI = V\_COD\_CLINUE; | | | | |

|  |
| --- |
| REVISIÓN TECNICA |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCEPTO** | 16 – VENCIMIENTO DE PUNTOS | | |
| **FUNCIONALIDAD:** | Realizar el vencimiento de puntos para el producto Postpago que cumplieron su tiempo de caducidad | | |
| **SHELL** | | | |
| **RUTA** | /home/usrclaroclub/CLAROCLUB/Interno/Prepago/Shells/SH011\_REPVENPUNT\_modif.sh | **FRECUENCIA EJECUCION** |  |
| **ARCHIVOS A PROCESAR** | | | |
| **RUTA** |  | **ORIGEN DATOS / HORA** |  |
| **PLATAFORMA** | | | |
| **APLICACIÓN** |  | | |
| **SP** |  | | |
| Se ejecuta la Shell: SH011\_REPVENPUNT\_modif.sh   1. Se considera el archivo de configuración importaCambioPlanHFCB.ctl para el inicio del proceso. 2. Se carga un archivo para obtener los conceptos que cumplan la caducidad mayor a 0 actualmente el periodo es de 18 meses.      1. Se calcula el total del saldo de los movimientos por cliente y concepto que tienen puntos por usar y que cumplieron el periodo de caducidad realizando una consulta a las tablas **PCLUB.ADMPT\_KARDEX K, PCLUB.ADMPT\_CLIENTE.**      1. Se actualizan el saldo a 0 de estos movimientos en la tabla **PCLUB.ADMPT\_KARDEX** | | | |
|  | | | |

|  |
| --- |
| REVISIÓN TECNICA |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONCEPTO** | 33– VENCIMIENTO DE PUNTO PREPAGO | | |
| **FUNCIONALIDAD:** |  | | |
| **SHELL** | | | |
| **RUTA** | /home/usrclaroclub/CLAROCLUB/Interno/Prepago/Shells/SH061\_MIGRACION\_PUNTOSCC.sh | **FRECUENCIA EJECUCION** |  |
| **ARCHIVOS A PROCESAR** | | | |
| **RUTA** |  | **ORIGEN DATOS / HORA** |  |
| **PLATAFORMA** | | | |
| **APLICACIÓN** |  | | |
| **SP** |  | | |
| Se ejecuta la Shell: SH061\_MIGRACION\_PUNTOSCC.sh   1. Se ingresa como parámetro de entrada la fecha actual para iniciar el proceso de migración en el procedimiento PKG\_CC\_MIGRACION.ADMPSI\_PREMIGPOS . 2. Se obtiene la lista de números postpago que han sido migrados a prepago del día anterior obtenidos de la BD Reptdm de la tabla **DW\_SUS\_M\_MIGRACIONPOSITIVA.**  |  | | --- | | **SELECT** **SUBSTR(**A**.**MSISDN**,** **LENGTH(**A**.**MSISDN**)** **-** 8**,** 9**)** **AS** MSISDN**,**  A**.**FECHAMIGRACION  **FROM** DM**.**DW\_SUS\_M\_MIGRACIONPOSITIVA@DBL\_REPTDM\_D A  **WHERE** IDSEGMENTOORIGEN **IN** **(**2**,** 3**)** --de donde vino Postpago Consumer  **AND** IDSEGMENTO **=** 1 --a donde ha llegado Prepago  **AND** FECHAMIGRACION **>=** K\_FECHINI  **AND** FECHAMIGRACION **<** K\_FECHFIN  **ORDER** **BY** A**.**MSISDN**;** |  1. Se obtiene la siguiente información del cliente en base a la línea por medio del siguiente procedimiento:  |  |  | | --- | --- | | **SP :**  PKG\_CLAROCLUB.ADMPSS\_DAT\_CLIE | **Consulta:** Obtiene el custcode (“código de cuenta”) mediante el número de documento del cliente  Luego realiza una segunda consulta mediante el custcode obtenido y se obtiene la siguiente información mostrada en la tercera línea del cuadro. | | **SELECT** cu**.**custcode**,** co**.**co\_id  **INTO** v\_custcode**,** v\_co\_id  **FROM** customer\_all@DBL\_BSCS CU**,** contract\_all@DBL\_BSCS co  **WHERE** cu**.**customer\_id **=** co**.**customer\_id  **AND** co**.**co\_id **=** tim**.**tfun006\_get\_coid\_from\_dn@DBL\_BSCS**(**K\_DN\_NUM**);**  **SELECT** /\*+ use\_nl(cu cc t) \*/  cu**.**custcode cuenta**,**  it**.**idtype\_name tip\_doc**,**  cc**.**passportno num\_doc**,**  v\_co\_id co\_id**,**  cu**.**billcycle ciclo\_fac**,**  cu**.**prgcode codigo\_tipo\_cliente**,**  pg**.**prgname tipo\_cliente  **FROM** customer\_all@DBL\_BSCS cu**,**  ccontact\_all@DBL\_BSCS cc**,**  ccontact\_all@DBL\_BSCS cc2**,**  id\_type@DBL\_BSCS it**,**  title@DBL\_BSCS t**,**  country@DBL\_BSCS c**,**  info\_cust\_combo@DBL\_BSCS icc**,**  marital\_status@DBL\_BSCS ms**,**  pricegroup\_all@DBL\_BSCS pg**,**  **language**@DBL\_BSCS la**,**  payment\_all@DBL\_BSCS pa**,**  paymenttype\_all@DBL\_BSCS pt  **WHERE** cu**.**customer\_id **=** cc**.**customer\_id  **AND** cc**.**ccbill **=** 'X'  **AND** cc2**.**ccforward**(+)** **=** 'X'  **AND** cc2**.**ccseq**(+)** **=** 2  **AND** cu**.**customer\_id **=** cc2**.**customer\_id**(+)**  **AND** cc**.**id\_type **=** it**.**idtype\_code**(+)**  **AND** cc**.**cctitle **=** t**.**ttl\_id**(+)**  **AND** cc**.**csnationality **=** c**.**country\_id**(+)**  **AND** cu**.**custcode **=** v\_custcode  **AND** icc**.**customer\_id**(+)** **=** cu**.**customer\_id  **AND** cu**.**marital\_status **=** ms**.**mas\_id**(+)**  **AND** cu**.**prgcode **=** pg**.**prgcode  **AND** cu**.**cslanguage **=** la**.**lng\_id**(+)**  **AND** pa**.**payment\_type **=** pt**.**payment\_id  **AND** cu**.**customer\_id **=** pa**.**customer\_id  **AND** pa**.**seq\_id **=** **(SELECT** **MAX(**seq\_id**)**  **FROM** payment\_all@DBL\_BSCS  **WHERE** customer\_id **=** pa**.**customer\_id**);** | |  |  1. Valida que el cliente PostPago existe y este activo.  |  | | --- | | Se valida que exista el cliente “1,2” mediante la variable V\_CUENTA y V\_ESTADO. | | **SELECT** ADMPC\_ESTADO  **INTO** V\_ESTADO  **FROM** PCLUB**.**ADMPT\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA  **AND** ADMPV\_COD\_TPOCL **IN** **(**'1'**,** '2'**);**  **SELECT** ADMPC\_ESTADO  **INTO** V\_ESTADO  **FROM** PCLUB**.**ADMPT\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA  **AND** ADMPV\_COD\_TPOCL **IN** **(**'1'**,** '2'**);** |  1. Valida que no tenga varias cuentas IB registradas para ese Cliente.  |  | | --- | | **SELECT** ADMPN\_COD\_CLI\_IB **INTO** V\_COD\_CLI\_IB  **FROM** PCLUB**.**ADMPT\_CLIENTEIB  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** |  1. Busca las líneas por cuenta de la BD Reptdm. Si encuentra datos realiza lo siguiente pero sin usar ningún datos de la Reptdm:  |  |  | | --- | --- | | Descripción de la secuencia | Secuencia de migración. | | Valida si ya está creada esa línea como PrePago, si no existe la crea. | **INSERT** **INTO** PCLUB**.**ADMPT\_IMP\_PREPOSPRE  **(**ADMPN\_ID\_FILA**,**  ADMPV\_COD\_CLI**,**  ADMPD\_FEC\_MIG**,**  ADMPD\_FEC\_OPER**,**  ADMPV\_MSJE\_ERROR**)**  **VALUES**  **(**PCLUB**.**ADMPT\_IMP\_PREPOSPRE\_SQ**.NEXTVAL,**  V\_CUENTA**,**  C\_FECHAMIGR**,**  **TO\_DATE(TO\_CHAR(SYSDATE,** 'dd/mm/yyyy'**),** 'dd/mm/yyyy'**),**  V\_ERROR**);** | | Valida si ya está creada el registro de la línea en la tabla saldo, si no existe lo crea. | **SELECT** ADMPN\_SALDO\_CC  **INTO** V\_SALDO  **FROM** PCLUB**.**ADMPT\_SALDOS\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** |  * + Si es cliente IB:  |  |  | | --- | --- | | Desvincula el cliente IB del cliente PostPago en la tabla ADMPT\_CLIENTEIB | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI**=**V\_TELEFONO**,**  ADMPD\_FEC\_MOD**=sysdate**  **WHERE** ADMPN\_COD\_CLI\_IB**=**V\_COD\_CLI\_IB**;** | | Genera el movimiento de salida para la cuenta PostPago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I” en la tabla ADMPT\_KARDEX. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  **(**V\_SALDO **\*** **(-**1**)),**  'S'**,**  'C'**,**  0**,**  'C'**,**  **sysdate);** | | Actualiza el movimiento con saldos a 0 del Kardex para los tipos “I” en la tabla ADMPT\_SALDOS\_CLIENTE. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_IB **=** 0**,**  ADMPN\_COD\_CLI\_IB **=** ''**,**  ADMPC\_ESTPTO\_IB **=** ''**,**  ADMPD\_FEC\_MOD **=** **sysdate**  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | Genera el movimiento de entrada para la línea PrePago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I”. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  V\_COD\_CLI\_IB**,**  V\_TELEFONO**,**  V\_COD\_CPTO\_IB**,**  **sysdate,**  V\_SALDO\_IB**,**  'E'**,**  'I'**,**  V\_SALDO\_IB**,**  'A'**,**  **sysdate);** | | Vincula el cliente IB con el cliente PrePago. | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI **=** V\_TELEFONO  **WHERE** ADMPN\_COD\_CLI\_IB **=** V\_COD\_CLI\_IB**;** |  1. Si no encuentra en BD Reptdm, realiza lo siguiente:  |  |  | | --- | --- | | Descripción de la secuencia | Secuencia de migración. | | Busca si la línea PostPago tiene otra cuenta por DNI | **SELECT** **COUNT(\*)**  **INTO** V\_COUNT  **FROM** **(SELECT** A**.**ADMPV\_COD\_CLI  **FROM** ADMPT\_CLIENTE A  **WHERE** A**.**ADMPV\_COD\_TPOCL **IN** **(**'1'**,** '2'**)**  **AND** A**.**ADMPC\_ESTADO **=** 'A'  **AND** A**.**ADMPV\_COD\_CLI **<>** K\_CUENTA  **AND** A**.**ADMPV\_TIPO\_DOC **=** K\_TIPO\_DOC  **AND** A**.**ADMPV\_NUM\_DOC **=** K\_NUM\_DOC**);** | | Si la cuenta PostPago tiene otra cuenta realiza la transferencia:  Obtiene el saldo de la cuenta que va a migrar de la tabla saldo cliente. | **SELECT** ADMPN\_SALDO\_CC  **INTO** V\_SALDO  **FROM** ADMPT\_SALDOS\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | * Actualiza el saldo en 0 los movimientos de kardex de la cuenta PostPago que va a migrar para los registros que tienen saldo y que tienen tipo de punto “C” y “L”. | **UPDATE** ADMPT\_KARDEX  **SET** ADMPN\_SLD\_PUNTO **=** 0**,** ADMPC\_ESTADO **=** 'C'  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA  **AND** ADMPC\_TPO\_OPER **=** 'E'  **AND** ADMPC\_TPO\_PUNTO **IN** **(**'C'**,** 'L'**)**  **AND** ADMPN\_SLD\_PUNTO **>** 0**;** | | * Genera el movimiento de salida en el kardex con el concepto 35 (MIGRACIONES POSTPAGO A PREPAGO CC) con estado “C” y tipo “C” para el cliente PostPago que va a migrar. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  **(**V\_SALDO **\*** **(-**1**)),**  'S'**,**  'C'**,**  0**,**  'C'**,**  **sysdate);** | | * Actualiza el saldo cliente en 0 y estado en “B” para el cliente PostPago que va a migrar. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_CC **=** 0**,** ADMPC\_ESTPTO\_CC **=** 'B'  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | * Genera el movimiento de entrada en el kardex con el concepto 35 (MIGRACIONES POSTPAGO A PREPAGO CC) con estado “A” y tipo “C” | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA\_1**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  V\_SALDO**,**  'E'**,**  'C'**,**  V\_SALDO**,**  'A'**);** | | * Actualiza el saldo cliente para el cliente PostPago que va a recibir la transferencia. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_CC **=** V\_SALDO **+**  **(SELECT** **NVL(**ADMPN\_SALDO\_CC**,** 0**)**  **FROM** ADMPT\_SALDOS\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA\_1**)**  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA\_1**;** | | * Le da de baja al cliente PostPago que va a migrar cambiando de estado a “B” en cliente. | **UPDATE** ADMPT\_CLIENTE  **SET** ADMPC\_ESTADO **=** 'B'**,**  ADMPV\_USU\_MOD **=** 'USRMIGCC'**,**  ADMPD\_FEC\_MOD **=** **sysdate**  **WHERE** ADMPV\_COD\_CLI**=**V\_CUENTA  **AND** ADMPV\_COD\_TPOCL **IN** **(**'1'**,**'2'**);** | | * Crea el nuevo cliente PrePago y su registro de saldo. | **INSERT** **INTO** ADMPT\_CLIENTE  **(**ADMPV\_COD\_CLI**,**  ADMPN\_COD\_CATCLI**,**  ADMPV\_TIPO\_DOC**,**  ADMPV\_NUM\_DOC**,**  ADMPV\_NOM\_CLI**,**  ADMPV\_APE\_CLI**,**  ADMPC\_SEXO**,**  ADMPV\_EST\_CIVIL**,**  ADMPV\_EMAIL**,**  ADMPV\_PROV**,**  ADMPV\_DEPA**,**  ADMPV\_DIST**,**  ADMPD\_FEC\_ACTIV**,**  ADMPC\_ESTADO**,**  ADMPV\_COD\_TPOCL**,**  ADMPD\_FEC\_REG**,**  ADMPV\_USU\_REG**)**  **VALUES**  **(**V\_TELEFONO**,**  2**,**  V\_TIPO\_DOC**,**  V\_NUM\_DOC**,**  V\_NOM\_CLI**,**  V\_APE\_CLI**,**  V\_SEXO**,**  V\_EST\_CIVIL**,**  V\_EMAIL**,**  V\_PROV**,**  V\_DEPA**,**  V\_DIST**,**  **TRUNC(SYSDATE),**  'A'**,**  '3'**,**  **SYSDATE,**  'USRMIGCC'**);**  **END** **IF;** | | Desvincula el cliente IB del cliente PostPago en la tabla ADMPT\_CLIENTEIB | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI**=**V\_TELEFONO**,**  ADMPD\_FEC\_MOD**=sysdate**  **WHERE** ADMPN\_COD\_CLI\_IB**=**V\_COD\_CLI\_IB**;** | | Genera el movimiento de salida para la cuenta PostPago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I” en la tabla ADMPT\_KARDEX. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  **(**V\_SALDO **\*** **(-**1**)),**  'S'**,**  'C'**,**  0**,**  'C'**,**  **sysdate);** | | Actualiza el movimiento con saldos a 0 del Kardex para los tipos “I” en la tabla ADMPT\_SALDOS\_CLIENTE. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_IB **=** 0**,**  ADMPN\_COD\_CLI\_IB **=** ''**,**  ADMPC\_ESTPTO\_IB **=** ''**,**  ADMPD\_FEC\_MOD **=** **sysdate**  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | Genera el movimiento de entrada para la línea PrePago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I”. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  V\_COD\_CLI\_IB**,**  V\_TELEFONO**,**  V\_COD\_CPTO\_IB**,**  **sysdate,**  V\_SALDO\_IB**,**  'E'**,**  'I'**,**  V\_SALDO\_IB**,**  'A'**,**  **sysdate);** | | Vincula el cliente IB con el cliente PrePago. | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI **=** V\_TELEFONO  **WHERE** ADMPN\_COD\_CLI\_IB **=** V\_COD\_CLI\_IB**;** |  1. Se valida que la línea no tiene otra cuenta :  |  |  | | --- | --- | | Descripción de la secuencia | Secuencia de migración. | | Obtiene el saldo de la cuenta que va a migrar de la tabla saldo cliente. | **SELECT** ADMPN\_SALDO\_CC **INTO** V\_SALDO  **FROM** ADMPT\_SALDOS\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | Actualiza el saldo en 0 los movimientos de kardex de la cuenta PostPago que va a migrar para los registros que tienen saldo y que tienen tipo de punto “C” y “L”. | **UPDATE** ADMPT\_KARDEX  **SET** ADMPN\_SLD\_PUNTO**=**0**,**  ADMPC\_ESTADO **=** 'C'  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA  **AND** ADMPC\_TPO\_OPER**=**'E'  **AND** ADMPC\_TPO\_PUNTO **IN** **(**'C'**,**'L'**)**  **AND** ADMPN\_SLD\_PUNTO **>** 0**;** | | Genera el movimiento de salida en el kardex con el concepto 35 (MIGRACIONES POSTPAGO A PREPAGO CC) con estado “C” y tipo “C” para el cliente PostPago que va a migrar. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  **(**V\_SALDO **\*** **(-**1**)),**  'S'**,**  'C'**,**  0**,**  'C'**,**  **sysdate);** | | Actualiza el saldo cliente en 0 y estado en “B” para el cliente PostPago que va a migrar. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_CC**=**0**,**  ADMPC\_ESTPTO\_CC**=**'B'  **WHERE** ADMPV\_COD\_CLI**=**V\_CUENTA**;** | | Crea el nuevo cliente PrePago y su registro de saldo. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA\_1**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  V\_SALDO**,**  'E'**,**  'C'**,**  V\_SALDO**,**  'A'**);** | | Genera el movimiento de entrada en el kardex con el concepto 35 (MIGRACIONES POSTPAGO A PREPAGO CC) con estado “C” y tipo “A” para el cliente PrePago. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA\_1**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  V\_SALDO**,**  'E'**,**  'C'**,**  V\_SALDO**,**  'A'**);** | | Actualiza el saldo del cliente PrePago. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_CC **=** V\_SALDO **+**  **(SELECT** **NVL(**ADMPN\_SALDO\_CC**,** 0**)**  **FROM** ADMPT\_SALDOS\_CLIENTE  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA\_1**)**  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA\_1**;** | | Obtiene el valor de Penalidad que es configurable y actualmente se encuentra en -150. | V\_VALOR **:=** V\_VALOR **\*** **(-**1**);**  **IF** V\_SALDO **>** V\_VALOR **THEN**  V\_PENALIDAD **:=** V\_VALOR**;**  **ELSE**  V\_PENALIDAD **:=** V\_SALDO**;**  **END** **IF;** | | Evalúa si el cliente tiene saldo mayor a la penalidad en caso sea menor la penalidad va a ser el total de su saldo. | **if** LK\_SLD\_PUNTOS**<=**V\_PUNTOS\_REQUERIDOS **then**  -- Actualiza Kardex  **update** admpt\_kardex  **set**  admpn\_sld\_punto **=** 0**,** admpc\_estado **=** 'C'  **where** admpn\_id\_kardex **=** LK\_ID\_KARDEX**;**  -- Actualiza Saldos\_cliente  **if** LK\_TPO\_PUNTO**=**'C' **or** LK\_TPO\_PUNTO**=**'L' **then**  **update** admpt\_saldos\_cliente  **set**  admpn\_saldo\_cc **=** **-** LK\_SLD\_PUNTOS **+** **(select** **NVL(**admpn\_saldo\_cc**,**0**)** **from** admpt\_saldos\_cliente  **where** admpv\_cod\_cli**=**LK\_COD\_CLI**)**  **where** admpv\_cod\_cli **=** LK\_COD\_CLI**;**  **end** **if;**  V\_PUNTOS\_REQUERIDOS**:=**V\_PUNTOS\_REQUERIDOS**-**LK\_SLD\_PUNTOS**;**  else | | Genera el movimiento de salida en el kardex con el concepto 37 (PENALIDAD POR MIGRACIONES POSTPAGO A PREPAGO) con estado “C” y tipo “S”. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  V\_COD\_CLI\_IB**,**  V\_CUENTA**,**  V\_COD\_CPTO\_IB**,**  **sysdate,**  **(**V\_SALDO\_IB **\*** **(-**1**)),**  'S'**,**  'I'**,**  0**,**  'C'**,**  **sysdate);** | | Se descuenta del saldo cliente la penalidad. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_IB**=**0**,**  ADMPN\_COD\_CLI\_IB **=** ''**,**  ADMPC\_ESTPTO\_IB **=** ''**,**  ADMPD\_FEC\_MOD **=** **sysdate**  **WHERE** ADMPV\_COD\_CLI**=**V\_CUENTA**;** | | Se descuenta del kardex la penalidad. | **UPDATE** ADMPT\_KARDEX  **SET** ADMPN\_SLD\_PUNTO**=**0**,**  ADMPC\_ESTADO **=** 'C'  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA  **AND** ADMPC\_TPO\_OPER**=**'E'  **AND** ADMPC\_TPO\_PUNTO **IN** **(**'C'**,**'L'**)**  **AND** ADMPN\_SLD\_PUNTO **>** 0**;** | | Le da de baja al cliente PostPago que va a migrar cambiando de estado a “B” en cliente. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_CC**=**0**,**  ADMPC\_ESTPTO\_CC**=**'B'  **WHERE** ADMPV\_COD\_CLI**=**V\_CUENTA**;** | | Si es cliente IB:   * + Desvincula el cliente IB del cliente PostPago en la tabla ADMPT\_CLIENTEIB | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI**=**V\_TELEFONO**,**  ADMPD\_FEC\_MOD**=sysdate**  **WHERE** ADMPN\_COD\_CLI\_IB**=**V\_COD\_CLI\_IB**;** | | * + Genera el movimiento de salida para la cuenta PostPago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I” en la tabla ADMPT\_KARDEX. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  ''**,**  V\_CUENTA**,**  V\_COD\_CPTO\_CC**,**  **sysdate,**  **(**V\_SALDO **\*** **(-**1**)),**  'S'**,**  'C'**,**  0**,**  'C'**,**  **sysdate);** | | * + Actualiza el movimiento con saldos a 0 del Kardex para los tipos “I” en la tabla ADMPT\_SALDOS\_CLIENTE. | **UPDATE** ADMPT\_SALDOS\_CLIENTE  **SET** ADMPN\_SALDO\_IB **=** 0**,**  ADMPN\_COD\_CLI\_IB **=** ''**,**  ADMPC\_ESTPTO\_IB **=** ''**,**  ADMPD\_FEC\_MOD **=** **sysdate**  **WHERE** ADMPV\_COD\_CLI **=** V\_CUENTA**;** | | * + Genera el movimiento de entrada para la línea PrePago con concepto 41 (MIGRACIONES POSTPAGO A PREPAGO IB) tipo “I”. | **INSERT** **INTO** ADMPT\_KARDEX  **(**ADMPN\_ID\_KARDEX**,**  ADMPN\_COD\_CLI\_IB**,**  ADMPV\_COD\_CLI**,**  ADMPV\_COD\_CPTO**,**  ADMPD\_FEC\_TRANS**,**  ADMPN\_PUNTOS**,**  ADMPC\_TPO\_OPER**,**  ADMPC\_TPO\_PUNTO**,**  ADMPN\_SLD\_PUNTO**,**  ADMPC\_ESTADO**,**  ADMPD\_FEC\_REG**)**  **VALUES**  **(**ADMPT\_KARDEX\_SQ**.NEXTVAL,**  V\_COD\_CLI\_IB**,**  V\_TELEFONO**,**  V\_COD\_CPTO\_IB**,**  **sysdate,**  V\_SALDO\_IB**,**  'E'**,**  'I'**,**  V\_SALDO\_IB**,**  'A'**,**  **sysdate);** | | * + Vincula el cliente IB con el cliente PrePago. | **UPDATE** ADMPT\_CLIENTEIB  **SET** ADMPV\_COD\_CLI **=** V\_TELEFONO  **WHERE** ADMPN\_COD\_CLI\_IB **=** V\_COD\_CLI\_IB**;** | | | | |