

CBDFR PASO A PASO

TABLAS QUE CONSULTA:

BS_MOR_MOMA_MONTO_MAXIMO *(Esta tabla contiene el monto maximo permitido para que un cliente no ingrese al proceso)*

bs_mor_recl_reclamos rc ,

bs_cob_tick_coll_tickets ct ,

bs_mor_clie_cliente cl ,

bs_mor_reha_rehabilitacion re,

bs_mor_estado_actual ea

bs_mor_pdrd_pedidored *(Cabecera de pedidos de acciones de red para responsable de pago)*

bs_mor_estado_actual

bs_mor_comm_comentarios *(Mensajes para programas batch de Mora)*

bs_cob_tick_coll_tickets *(Realiza INSERT sobre esta tabla)*

BS_MOR_REDE_RECLAMO_DESCARTADO *(Esta tabla contiene los reclamos que no hayan sido tenidos en cuenta por el Demonio de Facturas Reclamadas)*

bs_mor_esta_estado

bs_mor_acci_accion *(Especificaciones de cada paso de morosidad)*

TABLAS EN LAS QUE MODIFICA DATOS (UPDATE – DELETE - INSERT):

bs_cob_tick_coll_tickets *(Realiza INSERT sobre esta tabla)*

BS_MOR_CLHP_HPROC *(Pasos de morosidad para clientes en proceso de Morosidad)*

bs_mor_this_tickets_history

BS_MOR_CBDFR_UNIVERSO_DATOS

1. Busca Monto Maximo Permitido para que el proceso lo tenga en cuenta Consultando la tabla : **BS_MOR_MOMA_MONTO_MAXIMO**
2. Ejecutar **CURSOR MOROSOS: C_MOROSOS**
3. Verifica la cantidad de líneas en Pedidos de red con estado 'd' Desactivacion, consulta la tabla : **bs_mor_pdrd_pedidored** Busca el estado de Mora de cada cliente: Esta_id de la tabla : **bs_mor_estado_actual**.
4. Calcula la Deuda del cliente: **PKG_MOROSIDAD.FMO_DEUDA_CLIENTE**(:lhCustomerId);
5. Busca en la tabla : **bs_mor_comm_comentarios** los comentarios que luego insertara en la Coll Ticket.

6. Busca los contratos para insertar en la coll tickets con el CURSOR c_contratos
7. Inserta en la tabla: **bs_cob_tick_coll_tickets** ejecutando la función:
PMO_INS_CASO_VANTIVE_SIMPLE.
8. Inserta Cliente descartado en la tabla: **BS_MOR_REDE_RECLAMO_DESCARTADO**
9. Cuenta la cantidad de clientes descartados buscando por TICK_SEQ
10. Suspende proceso de mora: PKG_MOROSIDAD.PMO_SUSPENDE_PROCESO
11. Inserta un registro en la tabla **BS_MOR_CLHP_HPROC**. Suspende proceso: Valida si el proceso de MORA está Abierto entonces lo Suspende: UPDATE en la tabla : **bs_mor_clhp_hproc** al último registro poniéndole N para luego insertar el nuevo registro con estado último : 's' sobre la **bs_mor_clhp_hproc**, Busca la Descripción del estado del por qué se suspende la Mora en la tabla: **bs_mor_esta_estado** y luego impacta la actualización sobre la: **bs_cob_tick_coll_tickets**
12. Para los casos de casos cerrados se evalúa si existe un segundo reclamo, Consulta estas tablas: **BS_MOR_PETI_PERFIL_TIPF PETI**, **BS_CLI_CUST_CUSTOMERPROFILE PR**, **BS_COB_TICK_COLL_TICKETS CT** (Busca la cantidad) .
13. Si fallo suspencion o desuspencion terminamos procesamiento de moros: busca en la **BS_MOR_CLIA_ACCION** el perfil y el código.
14. Busca el Monto de IVR
15. Reactiva HOT **PKG_MOROSIDAD.PMO_HOT_VIGENTE**(:lhMorosId, :ihHotaActivar, :ihPaso, :ihEstadoPaso); Valida si es paso 0, si es HOT 1 o si es HOT 2.
16. Si la validación verifica que hay que activar alguno de los dos HOT 1 ó 2 o entonces ejecuta: **PKG_MOROSIDAD.FMO_VAL_Y_SUSPENDER**
 - a. -- Nombre : FMO_VAL_Y_SUSPENDER
 - b. -- Descripción : Llama al código que valida si se puede
 - c. ejecutar las acciones de red pretendidas
 - d. para c/u de los contratos por los cuales el cliente es responsable de pago.
 - e. Esta función opera en dos modos :
 - f. 1.- Se informa el cliente y no el contrato :
 - g. Suspende todos los contratos del cliente
 - h. 2.- Se informa el cliente y el contrato:
 - i. Suspende solamente el contrato informado
 - j. (Es el caso de límite de crédito que opera por contratos.)
17. Luego inserta en la Coll Tickets la actualización
18. Desactiva HOT si es necesario ejecutando el PKG: **PKG_MOROSIDAD.PMO_HOT_VIGENTE** si el HOT debe ser desactivado ejecuta: **PKG_MOROSIDAD.FMO_VAL_Y_REHABILITAR**
19. Busca el estado actual del moroso: **BS_MOR_ESTADO_ACTUAL**
20. Busca el paso en el que se encuentra el moroso: **bs_mor_paso_actual** y lo cierra ejecutando el PKG: **PKG_MOROSIDAD.PMO_CERRAR_PASO**:
 - a. Si el saldo del cliente es = 0 Entonces Busca el estado actual de la mora que sea <> 's' y el usuario último que lo tocó no sea DRZ.
 - b. Si el proceso no está suspendido es decir que es <> s, realiza un UPDATE sobre la tabla: **bs_mor_clia_accion** Cerrando el paso actual en 4 .

21. Procesar Moroso, hace un UPDATE sobre la tabla : bs_cob_tick_coll_tickets dejando una 'X' en el campo: tick_cod__est_inter si tenia la Mora abierta (a)

Si la mora era 'c' Cerrada lo que hace es borrar DELETE el registro de la tabla bs_mor_this_tickets_history.

22. Actualiza las Gestiones Pendientes: PCB_CBDFR.ACTUALIZAR_GESTIONS_PENDIENTES:

a. EXECUTE IMMEDIATE 'TRUNCATE TABLE BS_MOR_CBDFR_UNIVERSO_DATOS';

b. Busca el limite configurado de registros al que hace commit;

c. Abre cursor: p_cursor para recolectar todas las gestiones de la tabla :
BS_MOR_THIS_TICKETS_HISTORY (histórica)

d. Hace un INSERT sobre la tabla : BS MOR CBDFR UNIVERSO DATOS

e. Hace un UPDATE sobre la BS_MOR_THIS_TICKETS_HISTORY dejando en el campo
THIS_COD_EST_INTER = 'X'