Unidad I - TRABAJO PRACTICO I – Procedimientos y Disparadores

Sobre BD Firebird utilizada en Base de Datos I (11077), implementar:

1- Implementación Generator sobre tabla para generar ID en forma automática.
2- Implementación Tipo-Subtipo mutuamente exclusivos.
3- Implementación de cardinalidad máxima.
4- Implementación Entidad Débil-Entidad Fuerte con discriminador ascendente consecutivo.
5- Implementación de reglas semánticas sobre modelos implementados en Base de Datos I (11077).

Dadas las siguientes tablas mencionadas en [1]:

FACTURA(NRO,IMPORTE)
DETALLE(NRO,ID,CANTIDAD,PRECIO)
NRO FK FACTURA
ID FK PRODUCTO
PRODUCTO(ID,DESCR,STOCK)

-agregue la columna PRECIO _BASE a tabla producto (use el mismo tipo de dato que DETALLE.PRECIO), para guardar allí el precio de referencia del producto
-agregue la columna PRECIO_COSTO a tabla producto (use el mismo tipo de dato que DETALLE.PRECIO), para guardar allí el precio de costo del producto
-agregue la columna ESTADO a tabla factura (tipo SMALLINT), la cual indica el estado de la factura (0 iniciada, 1 finalizada, 2 anulada)
-agregue la columna FECHA a tabla factura (tipo DATE), la cual indica la fecha de la factura.

En un esquema de implementación B[1] (el usuario/aplicaciones NO interactúa directamente con las tablas), **resuelva** (agregue los campos o tablas que considere necesarios):

1. Cada vez que se vende un producto, se descuenta su cantidad de stock (cada vez que deja de vender un producto, sume su cantidad al stock). Un producto no puede venderse si el stock no es suficiente.

2. El importe de la factura es igual a la sumatoria de cantidad * precio; actualice el importe de la factura a medida que vende o deja de vender productos.

3. Un producto no puede venderse a un precio por debajo de su precio base.

4. El estado de una factura comienza con estado 0 (iniciada).
De estado 0 iniciada, puede pasar a estado 1 finalizada.
De estado 0 iniciada, puede pasar a estado 2 anulada.
De estado 1 finalizada, puede pasar a estado 2 anulada, pero no puede volver a estado 0 iniciada De estado 2 anulada, puede pasar a estado 1 finalizada, pero no puede volver a estado 0 iniciada.
Si una factura pasa de estado 1 finalizada a estado 2 anulada, se debe devolver toda la cantidad de producto al stock.
Si una factura estaba en estado 2 anulada y pasa a estado 1 finalizada, se debe restar toda la cantidad de producto del stock.

Julio 2016

Unidad I - TRABAJO PRACTICO I – Procedimientos y Disparadores

5. Implemente, de forma genérica, para todas las facturas, el siguiente control: por ejemplo, si la factura 100 tiene fecha 02-SEP-14, entonces, la factura 101 deberá tener una fecha mayor igual al 02-SEP-14 y nunca menor.

Resuelva las siguientes consultas utilizando SELECT PROCEDURES:

1. Obtenga los datos de las facturas a través de un select procedure que posee 2 parámetros de entrada: importe desde e importe hasta; los cuales indican el rango de importes a seleccionar de la tabla de factura.

2. Modifique el select procedure anterior para que:
- en caso de no indicar ningún valor desde se asuma el más pequeño.
- en caso de no indicar ningún valor hasta se asuma el más alto.

3. Se indican 2 parámetros de entrada: número de factura desde y número de factura hasta (ambos inclusive); se pretende listar los datos de todos los productos, indicando el total facturado en $ y en cantidad; de todos los productos facturados en facturas cuyo número sea mayor igual al número de factura desde y menor o igual al número de factura hasta.

4. Se desea imprimir la tabla de productos con los siguientes datos: ID, DESCR, STOCK y se desea agregar 3 columnas más: FACTURA1, FACTURA2, FACTURA3 en donde se indiquen los números (si es que existen) de las 3 ultimás facturas en donde fue vendido dicho producto. Se asume que las facturas con números más altos, son las facturas más recientes.

5. Se desea confeccionar una siguiente consulta cuyas columás son: FAC100,FAC1000,FAC10000,FAC100000 en donde en la primer columna se almacenan la cantidad de facturas con importes menores o iguales a $ 100, en la segunda la cantidad de facturas con importes menores o iguales a $ 1000 y mayores a $ 100 y así sucesivamente.

6. Se desean imprimir los datos de las facturas en las cuales el segundo producto facturado tenga un stock actual de menos de 1000 unidades. Se asume que el orden de los productos facturados de cada factura, van desde los ID's de productos más pequeños hacia los más grandes.
Sugerencia: hacer primero un select procedure que enumere los productos facturados en cada factura y luego otro select procedure que use al anterior para resolver el problema.

7. Implemente un select procedure que reciba como argumento de entrada un número entero que representa un año (ejemplo: 2016) y devuelve las siguientes columnas: MES01, MES02, ... MES12 que contienen el total de ventas (sumatoria de FACTURA.IMPORTE) de cada mes, para el año indicado como argumento de entrada.
Sugerencia: puede utilizar la función EXTRACT() de la siguiente forma, para obtener el mes de una fecha, de todas las facturas del año 2016 (observe que pasa cuando sumamos 1 a una fecha):
SELECT FECHA,EXTRACT(MONTH FROM FECHA) AS MES,FECHA+1 AS
FECHA2 FROM FACTURA WHERE EXTRACT(YEAR FROM FECHA) = 2016
Lea [2] para ver otras opciones o consulte el Language Reference Update 2.1, función extract().

Trabajos Prácticos

Unidad I - TRABAJO PRACTICO I – Procedimientos y Disparadores

Resuelva los siguientes procesos utilizando EXECUTE PROCEDURES:

1. Borre todas las facturas que están anuladas a partir de una fecha desde y hasta que se indican como argumentos de entrada del procedimiento. No hay argumentos de salida.

2. Implemente un stored procedure que permite hacer altas bajas o modificaciones sobre la tabla producto, los argumentos de entrada son: ACCION ('A' es alta, 'B' es baja, 'C' es cambio), ID,DESCR,STOCK se supone que no hay cambios sobre la clave primaria. El procedimiento tiene un parámetro de salida, de tipo integer el cual devuelve 0 cuando no hay ningún error y en caso de algún error devuelve SQLSTATE. Este procedimiento nunca generaría ninguna excepción, no debería dar error nunca, solo habría que chequear su retorno para determinar si la acción fue satisfactoria o no.
Sugerencia: hacer otro stored procedure que llame a éste para comprobar los valores devueltos o insertarlo en una tabla temporaria y asegurarse de que funciona correctamente.

3. Implemente un stored procedure para generar tuplas en la tabla FACTURA, los argumentos de entrada son: CANTIDAD, FECHA_DESDE. CANTIDAD se refiere a la cantidad de tuplas a insertar, FECHA_DESDE es la fecha de la primer factura a insertar, a partir de la segunda en adelante, se suma 1 a FECHA_DESDE para obtener el día siguiente. Asumir ESTADO = 0 (iniciada), IMPORTE = 0. Ignorar todo error que pueda suceder durante el proceso. El procedimiento no debería lanzar ninguna excepción ni dar error. El procedimiento no tiene ningún argumento de salida.

Referencias

[1] Cherencio, G., "Implementación de Triggers en Firebird/Interbase SQL Server", apunte asignatura 11078 BD II UNLu, disponible en http://www.grch.com.ar/docs/bdd/apuntes/unidad.i/11078-programacion-triggers.pdf
[2] Documentacion Firebird, Date Literals, disponible en http://www.firebirdsql.org/en/firebird-date-literals/

Julio 2016