

PRACTICO 3: Procedimientos Almacenados y Triggers

Esta práctica debe ser resuelta utilizando SQL. Para resolver la práctica se deben utilizar los motores de base de datos MYSQL 5.5, PostgreSQL 9.3 y ORACLE 10XE (o versiones superiores) este software, como sus manuales, están disponibles en la página de la materia.

Ejercicio 1

Teniendo en cuenta el ejercicio 1 a) de la práctica Nro 1(base de datos de facturación), agregue a la tabla factura el campo cantidad máxima de ítems permitidos. Controlar que la base de datos no acepte que una factura tenga más ítems que la cantidad de ítems permitidos por esta.

Nota: Resolver utilizando los motores que se utilizaron en la creación de las tablas (MySQL y Oracle)

Ejercicio 2

Utilizando el motor de base de datos PostgreSQL, cree las tablas:

Banco (cod_banco, nombre, precio, cotizacion_maxima)
Cotización (cod_cotizacion, cod_banco, cotización)
Cod_banco: clave foránea a Banco

Realizar una función que obtenga la máxima cotización para un banco dado y actualice el valor en la tabla Banco, la función debe retornar la cotización máxima en un parámetro de salida.

Nota: No utilizar la función MAX!!!.

Ejercicio 3

Utilizando el motor de base de datos Oracle, cree las tablas:

cuenta (nro_cuenta, saldo)
movimiento(nro_movimiento, nro_cuenta, fecha, debe, haber)
nro_cuenta: Clave foránea a cuenta

Luego, cree los siguientes procedimientos almacenados:

- a) Que inserte un movimiento y mantenga consistente el saldo en la tabla cuenta. El procedimiento debe tomar como parámetro el nro de cuenta, el debe y el haber. Al final del procedimiento imprima por pantalla el número de cuenta y el saldo actual.

- b) Que calcule el saldo de una cuenta a una determinada fecha. El procedimiento, además de mostrar por pantalla el saldo, lo debe retornar en un parámetro.

Ejercicio 4

Utilizando el motor MySQL con la base de datos definida en el ejercicio 1a) de la Práctica 1 de Repaso de SQL(Base de datos de clientes y productos), y haciéndole las modificaciones que crea necesarias, cree un procedimiento que guarde información en una tabla con las facturas que no son consistentes (una factura no es consistente si la suma de sus ítems es diferentes al monto total de la factura). Considerar:

- a) El procedimiento debe ser creado con el usuario vendedor3 (creado en la practica anterior) y el procedimiento debe ser creado para ejecutarse con los permisos del invocador.
- b) Ejecute el procedimiento con los usuarios vendedor3 y vendedor1.

Ejercicio 5

Cree triggers teniendo en cuenta las siguientes consideraciones:

- a) En la base de datos creada en el ejercicio 1a) de la Práctica 1) de Repaso de SQL, agregue un trigger para que antes de cada inserción de un producto se guarde la descripción en mayúsculas.
- b) En la base de datos creada en el ejercicio 1a) de la Práctica 1) de Repaso de SQL, agregue un trigger, para que luego de cada modificación en la cantidad de un producto se guarde información de auditoria. Esta información debe quedar almacenada en la tabla auditoriaProducto (crearla antes de definir el trigger), la información a almacenar es cod_producto, movimiento (diferencia de cantidades), fecha de realización de la actualización y el usuario que lo hizo. Con respecto al usuario que la realizó, debe ser el código de usuario del usuario logueado en la aplicación cliente de la base de datos

Hint: ver ejemplo Teórico

- c) Modifique la base de datos creada en el inciso b) del ejercicio 1 de la Práctica 1(bases de datos de estacionamiento) considerando que #parquimetro es auto numerado.

¿Cómo lo resolvería en Oracle? Hint: ver SEQUENCE

Nota: Resolver utilizando los motores que se utilizaron en la creación de las bases de datos.

Ejercicio 6

Utilizando triggers, solucione las restricciones de dominios de los ejercicios 1 a) y b) de la Práctica 1) de Repaso de SQL, que no pudieron resolver utilizando el comando check.