



TASCA 5.4 BASES DE DATOS

Antonio Ferrer.

Actividades tarea 5.4 de bases de datos.

11/05/2020

1. Es demana realitzar 4 preguntes tipus test amb 4 solucions sobre el que hajem vist aquesta avaluació a bbdd.

1. Algunas de las características que nos proporciona el uso de vistas son:

- a. Nos permite restringir al acceso a los datos, independizar los datos, hace más sencillas las consultas y podemos eliminar la vista sin borrar los datos.
- b. No nos permite restringir al acceso a los datos, en cambio si hace más sencillas las consultas y podemos modificar los datos de las tablas más fácilmente.
- c. Nos permite limitar el acceso a los datos a los usuarios, aunque complica las consultas necesarias, podemos crear vistas diferentes para los mismos datos.
- d. Podemos crear varias vistas que proporcionen información diferente de los mismos datos en cambio si borramos las vistas se verán afectados los datos subyacentes.

2. Que afirmación es más correcta:

CREATE TRIGGER Log_nuevo_cliente AFTER INSERT ON clientes FOR EACH ROW....

- a. Antes de insertar un cliente (o varios clientes) en la base de datos el motor de la base de datos realizará las acciones que tiene establecidas entre BEGIN y END.
- b. Después de insertar un cliente (o varios clientes) en la base de datos el motor de la base de datos realizará las acciones que tiene establecidas entre BEGIN y END.
- c. Después de insertar un cliente (o varios clientes) en la base de datos el motor de la base de datos realizará las acciones que tiene establecidas entre BEGIN y END por cada uno de los INSERT que se envíen en bloque.
- d. Se disparará una acción por cada Insert que se ejecute sobre la base de datos clientes.

3. Que afirmación es más correcta si ejecutáramos esta sentencia:

GRANT SELECT (nombre,email,dni) ON admin.clientes TO contable@'localhost' identified by 'contable_admin';

- a. El usuario contable puede seleccionar para ver las tablas nombre, email, dni y todas sus columnas de la base de datos de admin.clientes y para identificarse tiene que usar como contraseña contable_admin.
- b. Que el usuario llamado contable podrá ejecutar una sentencia SELECT sobre las columnas nombre, email y dni de la tabla clientes situada en la base de datos admin, este usuario podrá logearse desde cualquier lugar empleando la contraseña contable_admin.
- c. El usuario contable no podrá realizar un SELECT * from clientes;
- d. El usuario solo podrá realizar un SELECT (nombre,email,dni) FROM clientes pero si podrá autorizar a otros usuarios para que realicen las consultas que el no puede.

4. Si ejecutamos este procedimiento mediante: call envit_truc(10)

```
delimiter //
create procedure envit_truc(in dos_dePalo int)
begin
    declare suma_carta int;
    set suma_carta = dos_de_palo +10;
    case suma_carta
        when 33 then insert into consejo
            values (TORNA A ENVIDAR);
        when 32 then insert into consejo
            values (PIENSATELO);
        else insert into consejo
            values (NO ENVIDES);
    end case;
end;
delimiter ;
```

- a. Retorna el valor 20
- b. Inserta dentro de la tabla consejo el valor 20
- c. Retorna NO ENVIDES
- d. Inserta el valor NO ENVIDES dentro de consejo

2. Tanmateix es demana realitzar una petita BBDD d'entre dos i 3 taules amb 3 o 4 insercions que experimente amb els triggers, cursors, handlers, procediments... es a dir es tria un o dos conceptes que pogam haver vist durant aquest trimestre. Ha de explicar quin procediment s'utilitza i que fa el mateix.

He decidido realizar una simple base de datos que mantendrá un registro de los usuarios de la tienda y las compras que realicen, hay que tener en cuenta que una compra solo llevará asociado un producto. En la tabla de compras disponemos de un campo para la fecha y hora en la que se realizó la compra, pero precisamos que se complete dicho campo de forma automática cuando se realice una compra junto con esto, también precisamos que se inserte el precio del producto elegido por el usuario en el momento de la compra desde la tabla de productos en el momento de realizar dicho pedido.

A esta base de datos accederán:

EL PROPIETARIO DE LA TIENDA: solo tendrá acceso a ver las compras realizadas y sus datos, pero no podrá modificar nada ni acceder a los datos de la tabla de usuarios.

Su usuario es propietario y la contraseña es propietarioPWD

LA ENCARGADA DE ENVIOS: Podrá ver los datos del cliente y los de la compra. No podrá modificar ningún dato en la base de datos EXCEPTO el precio de la compra ya que tiene que añadir los gastos de envío.

Su usuario es almacen1 y su contraseña es almacenista.

EL ADMINISTRADOR DE LA BASE DE DATOS: Tendrá acceso total a la base de datos y control sobre esta. Es el usuario ROOT del sistema.

No se tiene que permitir que un usuario se conecte desde fuera de la oficina a la base de datos.

Nuestra jefa de almacén nos ha solicitado que la base de datos le calcule el importe total de la compra dependiendo de la tarifa de la compañía de transportes que escoja, disponen de 5 compañías y de esta forma evita errores al calcular los precios, si se introduce una compañía que no está en lista, debe de calcular el importe base de gestión del pedido que es de 1.80€. Claro está le gustaría realizar esta tarea de forma simple por lo que diseñaremos un procedimiento por el que al introducir la compañía y el pedido que está preparando debe de actualizarse el importe de la compra con la suma del importe del producto + los gastos de envío.

La almacenista una vez prepare el pedido y sepa la compañía con la que enviará el producto ejecutará:

CALL calculoImporteConGastos("SEUR", 1);

Y el pedido 1 se actualizará según los criterios indicados anteriormente.

CREACIÓN DE BASE DE DATOS Y TABLAS :

```
CREATE DATABASE tienda;
```

```
USE tienda;
```

```
CREATE TABLE usuarios (  
    idUsuario INT AUTO_INCREMENT,  
    nombre VARCHAR(20) NOT NULL,  
    email VARCHAR(60) NOT NULL,  
    direccion VARCHAR(50) NOT NULL,  
    PRIMARY KEY (idUsuario)  
);
```

```
CREATE TABLE productos (  
    idProducto INT AUTO_INCREMENT,  
    descripcion VARCHAR(120) NOT NULL,  
    precio DOUBLE NOT NULL,  
    PRIMARY KEY (idProducto)  
);
```

```
CREATE TABLE compras (  
    idCompra INT AUTO_INCREMENT,  
    descripcion VARCHAR(120) NOT NULL DEFAULT "SIN DESCRIPCIÓN",  
    importe DOUBLE NOT NULL,  
    idUsuario INT NOT NULL,  
    idProducto INT NOT NULL,  
    fechaHoraCompra TIMESTAMP,  
    PRIMARY KEY (idCompra),  
    CONSTRAINT compra_usuario_FK FOREIGN KEY (idUsuario) REFERENCES usuarios(idUsuario),  
    CONSTRAINT compra_producto_FK FOREIGN KEY (idProducto) REFERENCES productos(idProducto)  
);
```

```
/*INSERCIONES*/
```

```
INSERT INTO usuarios (nombre,email,direccion) VALUES ("Ozores", "nohijano@losbingueros.es", "calle del destape 1");
```

```
INSERT INTO usuarios (nombre,email,direccion) VALUES ("Faemino", "elQueTienePelo@faeminoycansado.com", "avenida del humor inteligente 22");
```

```
INSERT INTO usuarios (nombre,email,direccion) VALUES ("Monty Python S.A", "montyPython@monty.com", "calle del sentido de la vida 42");
```

```
INSERT INTO productos (descripcion, precio) VALUES ("Curso de iniciación al humor", 250);
```

```
INSERT INTO productos (descripcion, precio) VALUES ("Pack de libros sobre Como ser director de cine y no morir en el intento", 89);
```

```
INSERT INTO productos (descripcion, precio) VALUES ("Yate + derechos de amarre en las Maldivas", 889900001);
```

CREACIÓN DE LOS USUARIOS Y ASIGNACIÓN DE PRIVILEGIOS:

El administrador de la base de datos está ya configurado puesto que empleará el usuario Root por lo que vamos a crear los dos usuarios restantes:

Propietario:

```
GRANT SELECT(idCompra,descripcion,importe) ON tienda.compras TO propietario@'localhost'  
IDENTIFIED BY 'propietarioPWD';
```

Almacenista:

```
GRANT SELECT ON tienda.usuarios TO almacen1@'localhost'  
IDENTIFIED BY 'almacenista';  
GRANT SELECT ON tienda.compras TO almacen1@'localhost'  
IDENTIFIED BY 'almacenista';  
GRANT UPDATE(importe) ON tienda.compras TO almacen1@'localhost'  
IDENTIFIED BY 'almacenista';
```

CREACIÓN DEL TRIGGER PARA LA AUTO-INSERCIÓN DEL REGISTRO DE LA COMPRA.

```
DELIMITER //  
CREATE TRIGGER registraCompra_AI BEFORE INSERT ON compras FOR EACH ROW  
BEGIN  
    DECLARE precioProducto DOUBLE;  
    SET precioProducto = (SELECT productos.precio FROM productos WHERE idProducto = NEW.idProducto);  
    SET NEW.fechaHoraCompra = NOW() , NEW.importe = precioProducto ;  
END  
// DELIMITER
```

CREACIÓN DEL PROCEDIMIENTO CON CÁLCULO DEL IMPORTE TOTAL DEL PEDIDO:

```
DELIMITER //  
create procedure calculoImporteConGastos(in transportista VARCHAR(20), idCompraRealizada int)  
begin  
    case transportista  
        when "seur" then UPDATE compras SET importe = (importe + 6.18) WHERE idCompra = idCompraRealizada;  
        when "MRW" then UPDATE compras SET importe = (importe + 9.99) WHERE idCompra = idCompraRealizada;  
        when "UPS" then UPDATE compras SET importe = (importe + 3.80) WHERE idCompra = idCompraRealizada;  
        when "NACEX" then UPDATE compras SET importe = (importe + 5.28) WHERE idCompra = idCompraRealizada;  
        when "CORREOS" then UPDATE compras SET importe = (importe + 666.66) WHERE idCompra = idCompraRealizada;  
        else UPDATE compras SET importe = (importe + 1.80) WHERE idCompra = idCompraRealizada;  
    end case;  
end;  
DELIMITER ;
```