

Práctica 6

Automatización de tareas

En esta práctica se trata la automatización de tareas en bases de datos MySQL. Se presentará en formato .pdf y siguiendo unas normas de presentación correctas.

Se parte del docker-compose del aula virtual que lanza un phpmyadmin y una base de datos MySQL.

Ejercicio 1: Puesta a punto:

- Arrancar el docker de php y mysql
- Crea las siguientes tablas:

```
Articulo(  
    idarticulo int primary key,  
    codigo varchar(50) null,  
    nombre varchar(100) not null unique,  
    precio_venta decimal(11,2) not null, // opcionalmente float  
    stock integer not null,  
    descripcion varchar(256) null  
);
```

```
Persona(  
    idpersona int primary key ,  
    nombre varchar(100) not null,  
    dni varchar(9) null,  
    direccion varchar(70) null,  
    telefono varchar(20) null,  
    email varchar(50) null  
);
```

Sospechosos_de_error (igual que artículo)

Dinero_total (tendrá solo un número: el total de los precios de venta multiplicados por stock)

En persona introduce lo siguiente:

```
insert into persona values(1, "Pablo", "12345678L", "calle falsa 123", "915555556", "p@g.com");  
insert into persona values(2, "Cristina", "12365678L", "calle otra 12", "915565556", "c@g.com");  
insert into persona values(3, "Titi", "12345378L", "calle dos 13", "915535556", "t@g.com");  
insert into persona values(4, "Irene", "12345278L", "calle tres 23", "915535556", "i@g.com");  
insert into persona values(5, "Miguel", "12645678L", "calle m 153", "915255556", "m@g.com");  
insert into persona values(6, "Tomás", "12745678L", "calle do 53", "915555556", "t@g.com");  
insert into persona values(7, "Álvaro", "12945678L", "calle re 63", "915555856", "al@g.com");  
insert into persona values(8, "Ana", "12349678L", "calle mi 127", "915556856", "a@g.com");  
insert into persona values(9, "Inés", "12336678L", "calle fa 138", "915455556", "ine@g.com");  
insert into persona values(10, "Diego", "12445678L", "calle sol 132", "915556556", "dg@g.com");
```

Ejercicio 2: Triggers

Crea los siguientes triggers:

- 2.1.- Si se inserta un artículo se actualiza dinero total (sumándole el dinero x el stock añadido)
- 2.2.- Si se actualiza un artículo se actualiza dinero total (sumándole el dinero x el stock añadido o restándolo si hay menos)
- 2.3.- Si se borra un artículo (restándole el dinero x el stock que tuviera)
- 2.4.- Si se ingresa un artículo con precio_venta negativo se meterá en sospechosos_de_error y se ingresará en la tabla artículo con precio 0.
- 2.5.- Se hará lo mismo que en el trigger4 si se actualiza un precio a 0.

Entrega del código de los 5. Puedes mostrar ejemplo de funcionamiento.

Después de crear los triggers (puedes haber insertado y borrado elementos) borra todo lo que hayas insertado.

Inserta

```
insert into articulos values(1, "XXXXXXXX1", "Ejemplo", 12.54, 1, "xxxxxxxxxx");
insert into articulos values(2, "XXXXXXXX2", "Ejemplo", 2.5, 2, "xxxxxxxxxx");
insert into articulos values(3, "XXXXXXXX3", "Ejemplo", 15.24, 5, "xxxxxxxxxx");
insert into articulos values(4, "XXXXXXXX4", "Ejemplo", 7.24, -10, "xxxxxxxxxx");
insert into articulos values(5, "XXXXXXXX5", "Ejemplo", 8.54, 7, "xxxxxxxxxx");
insert into articulos values(6, "XXXXXXXX6", "Ejemplo", 16.99, 2, "xxxxxxxxxx");
insert into articulos values(7, "XXXXXXXX7", "Ejemplo", 25.99, 1, "xxxxxxxxxx");
insert into articulos values(8, "XXXXXXXX8", "Ejemplo", 70.3, 3, "xxxxxxxxxx");
insert into articulos values(9, "XXXXXXXX9", "Ejemplo", 80, -1, "xxxxxxxxxx");
insert into articulos values(10, "XXXXXXXX10", "Ejemplo", 80, 2, "xxxxxxxxxx");
insert into articulos values(11, "XXXXXXXX11", "Ejemplo", 26, 34, "xxxxxxxxxx");
insert into articulos values(12, "XXXXXXXX12", "Ejemplo", 10.2, 11, "xxxxxxxxxx");
insert into articulos values(13, "XXXXXXXX13", "Ejemplo", 63, 10, "xxxxxxxxxx");
```

Muestra el valor de Dinero_total y de sospechosos_de_error

Inserta un artículo (14) y muestra que se actualiza dinero total.

Borra el artículo 6 y muestra cómo se ha actualizado

Actualiza el 9 para ponerle 5 de stock y muestra cómo se ha actualizado.

Ejercicio 3: Procedimientos y funciones

3.1 Crea una función que dada una cadena la devuelva pasada a letras mayúsculas

3.2 Crea un procedimiento que reciba un id y muestre la información de la persona de la tabla personas con ese id si existe. (utilizad el formato del ejemplo)

Ej:

La información solicitada es de Pablo, con DNI 12345678L, con domicilio en calle falsa 123, correo p@g.com y número de teléfono 915555556.

3.3 Crea un procedimiento que muestre los artículos que tengan un precio entre 2 valores dados.

3.4 Crea un procedimiento que reciba 2 valores de descuento (entre 0 y 100%).

Le rebajará al precio de los artículos el primer valor si su precio actual es menor o igual a 20 y el segundo valor si es mayor de 20.

Por ejemplo:

call Procedimiento 70 30

Los artículos con precio inferior a 20 se les baja un 70% su precio.

Los artículos con precio superior a 20 se les baja un 30% su precio.

3.5 Crea un procedimiento que analice si existe un sospechosos de error resuelto y lo elimina.

Analiza los Sospechosos_de_error.

Si alguno existe en artículos y tiene stock distinto de 0 (significará que se ha actualizado), se eliminará de Sospechosos_de_error y se mostrará un mensaje:

“Eliminados: xx,yyy,zzz.....”

Se entregará el código de cada apartado y una muestra de su ejecución.