**EJERCICIO 1 PIENSA Y RESPONDE**

**DROP DATABASE IF EXISTS** test;

**CREATE DATABASE** test CHARACTER **SET** utf8mb4;

**USE** test;

**CREATE TABLE** cliente (

**Id** INT UNSIGNED **PRIMARY KEY**,

nombre CHAR (20)

);

**START TRANSACTION**;

**INSERT INTO** cliente **VALUES** (1, 'Pepe');

**COMMIT**;

***-- 1. ¿Qué devolverá esta consulta?***

**SELECT** \* **FROM** cliente;

**SET** AUTOCOMMIT=0;

**INSERT INTO** cliente **VALUES** (2, 'Maria');

**INSERT INTO** cliente **VALUES** (20, 'Juan');

**DELETE FROM** cliente **WHERE** nombre ='Pepe';

***-- 2. ¿Qué devolverá esta consulta?***

**SELECT** \* **FROM** cliente;

**ROLLBACK**;

***-- 3. ¿Qué devolverá esta consulta?***

**SELECT** \* **FROM** cliente;

**EJERCICIO 2 PIENSA Y RESPONDE**

**DROP DATABASE IF EXISTS** test;

**CREATE DATABASE** test CHARACTER **SET** utf8mb4;

**USE** test;

**CREATE TABLE** cuentas (

**Id** INTEGER UNSIGNED **PRIMARY KEY**,

saldo DECIMAL(11,2) **CHECK** (saldo >=0)

);

**INSERT INTO** cuentas **VALUES** (1,1000);

**INSERT INTO** cuentas **VALUES** (2,2000);

**INSERT INTO** cuentas **VALUES** (3,0);

*-- 1. Consultamos el estado actual de las cuentas*

**SELECT** \* **FROM** cuentas;

*-- 2. Suponga que queremos realizar una transferencia de dinero entre dos cuentas bancarias con la siguiente transacción:*

**START TRANSACTION**;

**UPDATE** cuentas **SET** saldo = saldo -100 **WHERE id**=1;

**UPDATE** cuentas **SET** saldo = saldo +100 **WHERE id**=2;

**COMMIT**;

*-- 3. ¿Qué devolverá esta consulta?*

**SELECT**\* **FROM** cuentas;

*-- 4. Suponga que queremos realizar una transferencia de dinero entre dos cuentas bancarias con la siguiente transacción y una de las dos cuentas no existe:*

**START TRANSACTION**;

**UPDATE** cuentas **SET** saldo = saldo -100 **WHERE id**=9999;

**UPDATE** cuentas **SET** saldo = saldo +100 **WHERE id**=2;

**COMMIT**;

*-- 6. Suponga que queremos realizar una transferencia de dinero entre dos cuentas bancarias con la siguiente transacción y la cuenta origen no tiene saldo:*

**START TRANSACTION**;

**UPDATE** cuentas **SET** saldo = saldo -100 **WHERE id**=3;

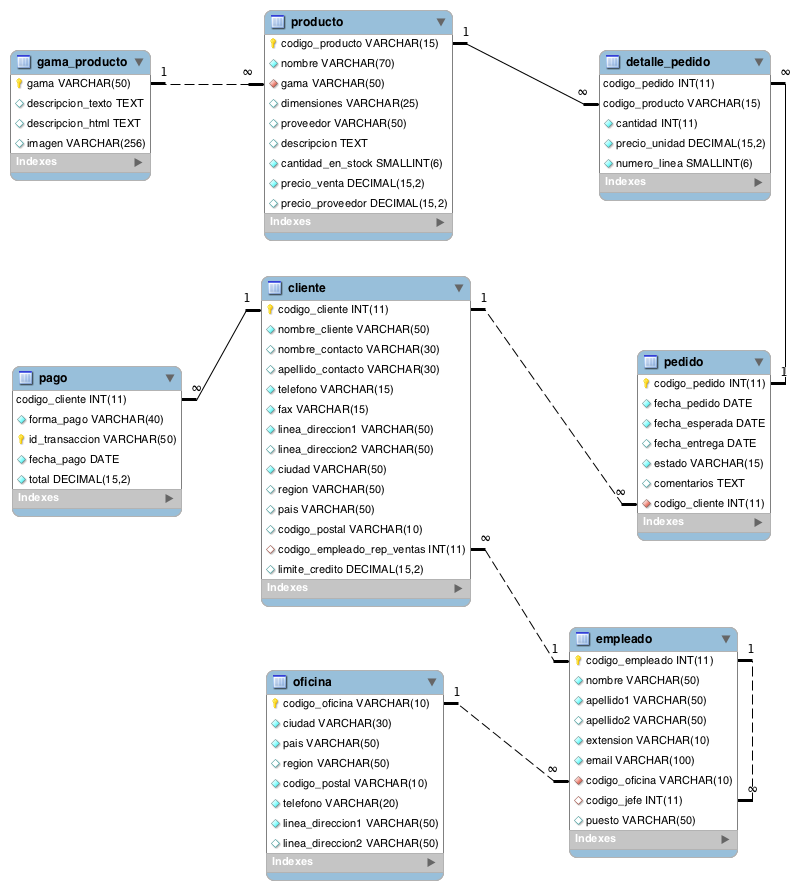
**UPDATE** cuentas **SET** saldo = saldo +100 **WHERE id**=2;

**COMMIT**;

*-- 7. ¿Qué devolverá esta consulta?*

**SELECT**\* **FROM** cuentas;

**EJERCICIO 3 BD JARDINERÍA Modelo entidad/relación**



**Realice las siguientes operaciones sobre la base de datos jardineria.**

1. Inserta una nueva oficina en Almería.
2. Inserta un empleado para la oficina de Almería que sea representante de ventas.
3. Inserta un cliente que tenga como representante de ventas el empleado que hemos creado en el paso anterior.
4. Inserte un pedido para el cliente que acabamos de crear, que contenga al menos dos productos.
5. Actualiza el código del cliente que hemos creado en el paso anterior y averigua si hubo cambios en las tablas relacionadas.
6. Borra el cliente y averigua si hubo cambios en las tablas relacionadas.
7. Elimina los clientes que no hayan realizado ningún pedido.
8. Incrementa en un 20% el precio de los productos que no tengan pedidos.
9. Borra los pagos del cliente con menor límite de crédito.
10. Establece a 0 el límite de crédito del cliente que menos unidades pedidas tenga del producto OR-179.
11. Modifica la tabla detalle\_pedido para insertar un campo numérico llamado iva. Mediante una transacción, establece el valor de ese campo a 18 para aquellos registros cuyo pedido tenga fecha a partir de Enero de 2009. A continuación actualiza el resto de pedidos estableciendo el iva al 21.
12. Modifica la tabla detalle\_pedido para incorporar un campo numérico llamado total\_linea y actualiza todos sus registros para calcular su valor con la fórmula:

total\_linea = precio\_unidad\*cantidad \* (1 + (iva/100));

1. Borra el cliente que menor límite de crédito tenga. ¿Es posible borrarlo solo con una consulta? ¿Por qué?
2. Inserta una oficina con sede en Granada y tres empleados que sean representantes de ventas.
3. Inserta tres clientes que tengan como representantes de ventas los empleados que hemos creado en el paso anterior.
4. **Realiza una transacción que inserte un pedido para cada uno de los clientes. Cada pedido debe incluir dos productos.**
5. **Borra uno de los clientes y comprueba si hubo cambios en las tablas relacionadas. Si no hubo cambios, modifica las tablas necesarias estableciendo la clave foránea con la clausula ON DELETE CASCADE.**
6. **Realiza una transacción que realice los pagos de los pedidos que han realizado los clientes del ejercicio anterior.**

**EJERCICIO 4 Ejercicios prácticos de transacciones**

1. Ejecuta las siguientes instrucciones y resuelve las cuestiones que se plantean en cada paso.

**SET** AUTOCOMMIT =0;

**SELECT** @@AUTOCOMMIT;

**DROP DATABASE IF EXISTS** test;

**CREATE DATABASE** test CHARACTER **SET** utf8mb4;

**USE** test;

**CREATE TABLE** producto (

**Id** INT UNSIGNED AUTO\_INCREMENT **PRIMARY KEY**,

nombre VARCHAR(100) **NOT NULL**,

precio DOUBLE

);

**INSERT INTO** producto (**id**, nombre) **VALUES** (1, 'Primero');

**INSERT INTO** producto (**id**, nombre) **VALUES** (2, 'Segundo');

**INSERT INTO** producto (**id**, nombre) **VALUES** (3, 'Tercero');

*-- 1. Comprueba que las filas se han insertado en la tabla de forma correcta.*

**SELECT** \* **FROM** producto;

Ahora vamos a simular que perdemos la conexión con el servidor antes de que la transacción sea completada (Observa que hemos ejecutado SET AUTOCOMMIT = 0).

Para simular que perdemos la conexión desde *MySQL Workbench* hay que cerrar la pestaña de conexión con el servidor. Si estás conectado al servidor desde la consola de MySQL sólo tienes que ejecutar el comando EXIT.

Volvemos a conectar con el servidor y ejecutamos las siguientes instrucciones:

**USE** test;

*-- ¿Qué devolverá esta consulta?*

**SELECT** \* **FROM** producto;

1. Ejecuta las siguientes instrucciones y resuelve las cuestiones que se plantean en cada paso.

**SET** AUTOCOMMIT =1;

**SELECT** @@AUTOCOMMIT;

**DROP DATABASE IF EXISTS** test;

**CREATE DATABASE** test CHARACTER **SET** utf8mb4;

**USE** test;

**CREATE TABLE** producto (

**Id** INT UNSIGNED AUTO\_INCREMENT **PRIMARY KEY**,

nombre VARCHAR(100) **NOT NULL**,

precio DOUBLE

);

**INSERT INTO** producto (**id**, nombre) **VALUES** (1, 'Primero');

**INSERT INTO** producto (**id**, nombre) **VALUES** (2, 'Segundo');

**INSERT INTO** producto (**id**, nombre) **VALUES** (3, 'Tercero');

*-- 1. ¿Qué devolverá esta consulta?*

**SELECT** \* **FROM** producto;

*-- 2. Vamos a intentar deshacer la transacción actual*

**ROLLBACK**;

*-- 3. ¿Qué devolverá esta consulta? Justifique su respuesta.*

**SELECT** \* **FROM** producto;

*-- 4. Ejecutamos la siguiente transacción*

**START TRANSACTION**;

**INSERT INTO** producto (**id**, nombre) **VALUES** (4, 'Cuarto');

**SELECT** \* **FROM** producto;

**ROLLBACK**;

*-- 5. ¿Qué devolverá esta consulta? Justifique su respuesta.*

**SELECT** \* **FROM** producto;

*-- 6. Ejecutamos la siguiente transacción*

**INSERT INTO** producto (**id**, nombre) **VALUES** (5, 'Quinto');

**ROLLBACK**;

*-- 7. ¿Qué devolverá esta consulta? Justifique su respuesta.*

**SELECT** \* **FROM** producto;

*-- 8. Desactivamos el modo AUTOCOMMIT y borramos el contenido de la tabla*

**SET** AUTOCOMMIT =0;

**SELECT** @@AUTOCOMMIT;

**DELETE FROM** producto **WHERE id**>0;

*-- 9. Comprobamos que la tabla esta vacia*

**SELECT** \* **FROM** producto;

*-- 10. Insertamos dos filas nuevas*

**INSERT INTO** producto (**id**, nombre) **VALUES** (6, 'Sexto');

**INSERT INTO** producto (**id**, nombre) **VALUES** (7, 'Séptimo');

**SELECT** \* **FROM** producto;

*-- 11. Hacemos un ROLLBACK*

**ROLLBACK**;

*-- 12. ¿Qué devolverá esta consulta? Justifique su respuesta.*

**SELECT** \* **FROM** producto;

*-- 13. Ejecutamos la siguiente transacción*

**SET** AUTOCOMMIT =0;

**START TRANSACTION**;

**CREATE TABLE** fabricante (**id** INT UNSIGNED);

**INSERT INTO** fabricante (**id**) **VALUES** (1);

**SELECT** \* **FROM** fabricante;

**ROLLBACK**;

*-- 14. ¿Se puede hacer ROLLBACK de instrucciones de tipo DDL (CREATE, ALTER, DROP, RENAME y TRUNCATE)?*