

# Tarea 5.5 Bases de Datos (Transacciones)

+ 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');
```

0º) Comprueba que las filas se han insertado en la tabla de forma correcta :

-> **SELECT \* FROM producto;**

- La consulta `SELECT * FROM producto` devolverá las filas insertadas en la tabla producto, mostrando los valores de las columnas id, nombre y precio (siendo este último nulo para las filas insertadas).

+ 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;`

0.5°) ¿Qué devolverá esta consulta?

-> `SELECT * FROM producto;`

- Después de reconectar al servidor, al ejecutar la consulta “`SELECT * FROM producto;`”, devolverá un error indicando que la tabla no existe. Esto es porque la tabla fue creada dentro de una transacción que no fue confirmada antes de la desconexión, por lo tanto, la tabla no se ha guardado permanentemente en la base de datos.

+ Ejecuta las siguientes instrucciones y resuelve las cuestiones que se plantean en cada paso:

### 1º.) ¿Qué devolverá esta consulta?

-> **SELECT \* FROM producto;**

- La consulta “SELECT \* FROM producto;” devolverá las filas insertadas en la tabla “producto”, mostrando los valores de las columnas “id”, “nombre” y “precio”.

### 2º.) Vamos a intentar deshacer la transacción actual :

-> **ROLLBACK;**

- Intentar deshacer la transacción actual con “ROLLBACK” no tendrá ningún efecto ya que no hay ninguna transacción actual en curso.

### 3º.) ¿Qué devolverá esta consulta?

-> **SELECT \* FROM producto;**

- La consulta “SELECT \* FROM producto;” devolverá las filas insertadas en la tabla producto.

### 4º.) Ejecutamos la siguiente transacción :

-> **START TRANSACTION;**

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

-> **SELECT \* FROM producto;**

-> **ROLLBACK;**

- Al ejecutar la transacción, se insertará un registro con “id” 4 y “nombre” 'Cuarto' en la tabla “producto”, pero como la transacción se deshace inmediatamente después, la tabla quedará sin el registro insertado.

#### 5º) ¿Qué devolverá esta consulta?

-> **SELECT \* FROM producto;**

- La consulta “SELECT \* FROM producto;” devolverá las filas insertadas en la tabla producto antes de la transacción.

#### 6º) Ejecutamos la siguiente transacción :

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

-> **ROLLBACK;**

- Al intentar insertar una fila dentro de una transacción que luego se deshace, no se realizará ninguna modificación en la tabla producto.

#### 7º) ¿Qué devolverá esta consulta?

-> **SELECT \* FROM producto;**

- La consulta “SELECT \* FROM producto;” devolverá las filas insertadas en la tabla producto antes de la transacción.

#### 8º) Desactivamos el modo “AUTOCOMMIT” y borramos el contenido de la tabla :

- > SET AUTOCOMMIT = 0;
- > SELECT @@AUTOCOMMIT;
- > DELETE FROM producto WHERE id > 0;

- Después de desactivar el modo “AUTOCOMMIT” y borrar el contenido de la tabla, la tabla estará vacía.

### 9º.) Comprobamos que la tabla está vacía :

- > SELECT \* FROM producto;

- La consulta “SELECT \* FROM producto;” devolverá una tabla vacía.

### 10º.) Insertamos dos filas nuevas :

- > INSERT INTO producto (id, nombre) VALUES (6, 'Sexto');
- > INSERT INTO producto (id, nombre) VALUES (7, 'Séptimo');
- > SELECT \* FROM producto;

- Al insertar dos filas nuevas fuera de una transacción explícita, estas inserciones serán permanentes y la tabla ya no permanecerá vacía. Por lo tanto, al ejecutar “SELECT \* FROM producto;”, se mostrarán todas las filas insertadas, incluidas las dos nuevas ('Sexto' y 'Séptimo').

### 11º.) Hacemos un “ROLLBACK” :

- > ROLLBACK;

- Como las inserciones se realizaron fuera de una transacción explícita, no hay nada que deshacer con un “ROLLBACK”. Esta operación no tendrá ningún efecto en las filas insertadas anteriormente, que permanecerán en la tabla “producto”.

## 12°) ¿Qué devolverá esta consulta?

-> **SELECT \* FROM producto;**

- La consulta “SELECT \* FROM producto;” mostrará todas las filas insertadas en la tabla “producto”, incluyendo las dos inserciones realizadas antes del “ROLLBACK”.

## 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**

- Se inicia una transacción explícita y se intenta crear una tabla “fabricante” e insertar un registro en ella. Sin embargo, como la transacción se deshace con el “ROLLBACK”, ninguna de estas operaciones tendrá efecto y la tabla “fabricante” no se creará ni contendrá ningún registro.

**12°)** ¿Se puede hacer “ROLLBACK” de instrucciones de tipo “DDL” (“CREATE”, “ALTER”, “DROP”, “RENAME” y “TRUNCATE”)?

- No, no se puede hacer un “ROLLBACK” de instrucciones de tipo “DDL” como “CREATE”, “ALTER”, “DROP”, “RENAME” y “TRUNCATE”. Estas instrucciones no forman parte de una transacción y sus cambios son permanentes una vez ejecutados, por lo que no pueden revertirse directamente con un “ROLLBACK”.

Lucía González Rosas  
1º de DAM  
18 de febrero del 2024