

# TRANSACCIONES

**CREAR UN DOCUMENTO DE TEXTO CON LAS PREGUNTAS NUMERADAS, CAPTURAS DE PANTALLA DE LA EJECUCIÓN DE LAS SENTENCIAS Y LAS RESPUESTAS A LAS PREGUNTAS PLANTEADAS (SE ENCUENTRAN MARCADAS EN NEGRITA).**

## A. Ejecutar las siguientes sentencias desde una misma sesión y resolver las cuestiones que se plantean.

```
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'), (2, 'Segundo'), (3, 'Tercero');

-- 1. ¿Qué devolverá la siguiente consulta?
SELECT * FROM producto;

ROLLBACK;
-- 2. ¿Qué devolverá la siguiente consulta? Justificar la respuesta.
SELECT * FROM producto;

START TRANSACTION;
INSERT INTO producto (id, nombre) VALUES (4, 'Cuarto');
SELECT * FROM producto;
ROLLBACK;
-- 3. ¿Qué devolverá la siguiente consulta? Justificar la respuesta.
SELECT * FROM producto;

INSERT INTO producto (id, nombre) VALUES (5, 'Quinto');
ROLLBACK;
-- 4. ¿Qué devolverá la siguiente consulta? Justificar la respuesta.
SELECT * FROM producto;

SET AUTOCOMMIT = 0;
SELECT @@AUTOCOMMIT;
DELETE FROM producto WHERE id > 0;
SELECT * FROM producto;
INSERT INTO producto (id, nombre) VALUES (6, 'Sexto'), (7, 'Séptimo');
SELECT * FROM producto;
ROLLBACK;
-- 5. ¿Qué devolverá la siguiente consulta? Justificar la respuesta.
SELECT * FROM producto;

SET AUTOCOMMIT = 0;
START TRANSACTION;
CREATE TABLE fabricante (id INT UNSIGNED);
INSERT INTO fabricante (id) VALUES (1);
SELECT * FROM fabricante;
ROLLBACK;
-- 6. ¿Se puede hacer ROLLBACK de sentencias del DDL tipo CREATE, ALTER, DROP, ...?
```

## B. ¿Qué problema o problemas podrían llegar a presentar las siguientes transacciones si se ejecutan concurrentemente? ¿qué nivel de aislamiento sería el adecuado para que no se llegue a producir ninguno de los posibles problemas?

### Transacción 1

```
START TRANSACTION;
Obtiene n registros
Obtiene n registros
Obtiene n+1 registros
COMMIT;
```

### Transacción 2

```
START TRANSACTION;
Insertar registro;
COMMIT;
```

**C. Teniendo en cuenta el orden de ejecución de las sentencias SQL dado en las sesiones siguientes y haciendo uso del nivel de aislamiento por defecto de *MariaDB* (lectura repetible), dar respuesta a las cuestiones que se plantean.**

SESIÓN 1	SESIÓN 2
<pre> DROP DATABASE IF EXISTS test; CREATE DATABASE test CHARACTER SET utf8mb4; USE test; START TRANSACTION; CREATE TABLE producto (   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,   nombre VARCHAR(100) NOT NULL,   precio DOUBLE ); </pre>	
	<pre> USE test; SHOW TABLES; START TRANSACTION; SELECT * FROM producto; -- 1) ¿Se podrá ver la tabla creada por la transacción --    1? ¿por qué? ¿qué devolverá la consulta anterior? </pre>
<pre> START TRANSACTION; -- Cierra la transacción anterior INSERT INTO producto (id, nombre) VALUES (1, 'Primero'), (2, 'Segundo'), (3, 'Tercero'); SELECT * FROM producto; -- 2) ¿Qué devolverá la consulta anterior? </pre>	
	<pre> SELECT * FROM producto; -- 3) ¿Qué devolverá la consulta anterior? ¿por qué? </pre>
<pre> SAVEPOINT sp1; INSERT INTO producto (id, nombre) VALUES (4, 'Cuarto'); SELECT * FROM producto; -- 4) ¿Qué devolverá la consulta anterior? ¿por qué? </pre>	
	<pre> INSERT INTO producto (id, nombre) VALUES (5, 'Quinto'); SELECT * FROM producto; -- 5) ¿Qué devolverá la consulta anterior? ¿por qué? COMMIT; </pre>
<pre> ROLLBACK TO SAVEPOINT sp1; SELECT * FROM producto; -- 6) ¿Qué devolverá la consulta anterior? ¿devuelve la --    fila con id 4? ¿y la fila con id 5? ¿por qué en --    cada caso? COMMIT; </pre>	
	<pre> START TRANSACTION; DELETE FROM producto WHERE id = 4; SAVEPOINT sp1; INSERT INTO producto (id, nombre) VALUES (4, '444444'); RELEASE SAVEPOINT sp1; SELECT * FROM producto; -- 7) ¿Qué devolverá la consulta anterior? ¿qué hace --    RELEASE SAVEPOINT? ¿La sentencia DELETE borra --    alguna fila? COMMIT; </pre>
<pre> START TRANSACTION; SELECT * FROM producto WHERE id IN (1, 3); -- 8) ¿Qué devolverá la consulta anterior? </pre>	
	<pre> START TRANSACTION; UPDATE producto SET precio = 5 WHERE id = 1; UPDATE producto SET nombre = '3333' WHERE id = 3; COMMIT; SELECT * FROM producto WHERE id = 1 OR id = 3; -- 9) ¿Qué devolverá la consulta anterior? </pre>
<pre> SELECT * FROM producto WHERE id IN (1, 3); -- 10) ¿Qué devolverá la consulta anterior? ¿coincide --      con el resultado de la pregunta 9? ¿por qué? COMMIT; START TRANSACTION; SELECT * FROM producto WHERE id IN (1, 3) FOR UPDATE; -- 11) ¿Qué devolverá la consulta anterior? ¿coincide --      con el resultado de la pregunta 9? ¿por qué? </pre>	
	<pre> START TRANSACTION; SELECT * FROM producto WHERE id = 3 FOR UPDATE; -- 12) ¿Qué devolverá la consulta anterior? ¿sucede --      algo no esperado? </pre>

SESIÓN 1		SESIÓN 2	
En caso de suceder algo en la sesión 2, esperar y no continuar ejecutando las sentencias de la sesión 1. 13) ¿Qué es lo que ha sucedido? ¿a qué se debe?			
UPDATE producto SET precio=1, nombre='1' WHERE id = 1; UPDATE producto SET precio=3, nombre='3' WHERE id = 3; ROLLBACK; SELECT * FROM producto WHERE id IN (1, 3); -- 14) ¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 11? ¿por qué?			
		START TRANSACTION; SELECT * FROM producto WHERE id = 3 FOR UPDATE; -- 15) ¿Qué devolverá la consulta anterior? ¿sucede algo no esperado? UPDATE producto SET precio=3, nombre='3' WHERE id = 3;	
START TRANSACTION; SELECT * FROM producto WHERE id IN (1, 3) FOR UPDATE; -- 16) ¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 14? ¿por qué? ¿sucede algo no esperado?			
En caso de suceder algo en la sesión 1, no esperar y continuar ejecutando las sentencias de la sesión 2. 17) ¿Qué es lo que ha sucedido? ¿a qué se debe?			
		COMMIT;	
18) ¿Ha sucedido algo en la sesión de la transacción 1 al hacer COMMIT en la transacción 2? ¿qué? ¿por qué?			
UPDATE producto SET precio=1, nombre='1' WHERE id = 1; COMMIT; SELECT * FROM producto WHERE id IN (1, 3); -- 19) ¿Qué devolverá la consulta anterior?			
		SELECT * FROM producto WHERE id IN (1, 3); -- 20) ¿Qué devolverá la consulta anterior? ¿coincide con lo que se devuelve en la pregunta 19? ¿por qué?	
START TRANSACTION; INSERT INTO producto VALUES (10, 'Décimo', 10); SELECT * FROM producto; -- 21) ¿Qué devolverá la consulta anterior?			
Simular que se pierde la conexión con el servidor (cerrar la aplicación de la sesión 1) antes de que la transacción sea completada. Volver a conectar con el servidor.			
USE test; SELECT * FROM producto; -- 22) ¿Qué devolverá la consulta anterior? ¿muestra la fila con id 10? ¿qué ha sucedido?			
		START TRANSACTION; SELECT * FROM producto WHERE id = 3 FOR UPDATE;	
START TRANSACTION; SELECT * FROM producto WHERE id = 1 FOR UPDATE;			
		SELECT * FROM producto WHERE id = 1 FOR UPDATE;	
Aunque se quede la sesión 2 en espera, continuar ejecutando las sentencias de la sesión 1.			
SELECT * FROM producto WHERE id = 3 FOR UPDATE;			
23) ¿Ha sucedido algo? ¿qué? ¿por qué? ¿qué se puede hacer para que no se produzcan estas situaciones?			
ROLLBACK;			
		ROLLBACK;	
24) Proponer una forma alternativa en la que se podrían ejecutar las dos transacciones anteriores sin que se produzca ningún problema (¿se podría cambiar el orden de los SELECT ... FOR UPDATE?).			
START TRANSACTION; INSERT INTO producto VALUES (11, 'Undécimo', 11); SELECT * FROM producto;			
		START TRANSACTION; INSERT INTO producto VALUES (11, 'Undécimo', 11);	
25) ¿Ha sucedido algo en la sesión 2? ¿qué? ¿a qué se debe?			
ROLLBACK;			
		ROLLBACK;	