Actividad de clase: Transacciones

Parte A

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

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

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)

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;

producto (3r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)

producto (4r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)
4	4	Cuarto	(NULL)

   En la primera muestra solo tres porque se selecciona antes de la inserción.

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

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)

   Devuelve solo 3 porque el select está antes de la inserción.

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;
```

producto (4r × 3c) | Resultado #2 (1r × 1c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)
4	5	Quinto	(NULL)

En el primer select devuele la captura anterior debido a que en el ejercicio anterior, a pesar de hacer un rolback, no había punto al que volver.

#	@@AUTOCOMMIT
1	0

En el segundo select devuelve el estado de autocommit como en la captura anterior.

En el tercer select no devuelve nada debido a que se ha borrado el contenido de la tabla.

#	id	nombre	precio
1	6	Sexto	(NULL)
2	7	Séptimo	(NULL)

En el cuarto select devuelve la captura anterior porque se han añadido esas 2 entradas después de borrar las otras.

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;
```

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)
4	5	Quinto	(NULL)

En el primer select devuelve la captura anterior debido a que en el ejercicio anterior se ha desactivado el auto commit no se han aplicado los cambios, y tras los cambios se ha hecho un rolback que ha cancelado los cambios y ha activado el autocommit de vuelta.

6. **¿Se puede hacer ROLLBACK de sentencias del DDL tipo CREATE, ALTER, DROP, ...?**

   No debido a que las sentencias del DDL implican cambios permanentes en la estructura de la base de datos.

Parte 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?

Se podría presentar el problema de lectura fantasma donde se reciben x cantidad de entradas en la primera consulta y en la segunda se reciben más o menos de las que había anteriormente. Este problema se puede solucionar con el nivel de aislamiento serializable.

Parte 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.

1. ***¿Se podrá ver la tabla creada por la transacción 1? ¿por qué? ¿qué devolverá la consulta anterior?***

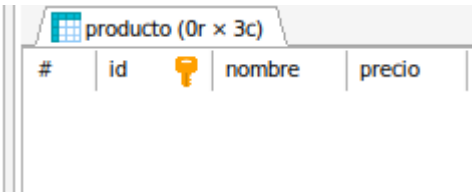
   Si se puede ver porque los cambios estructurales se reflejan en todas las sesiones independientemente de las transacciones.

   La consulta no devuelve nada debido a que no hay filas en la tabla.

2. ***¿Qué devolverá la consulta anterior?***

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)

3. ***¿Qué devolverá la consulta anterior? ¿por qué?***



   Porque no se han confirmado los cambios en la sesión 1.

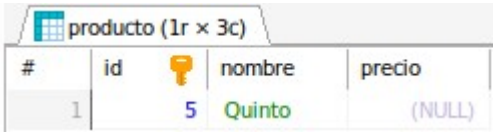
4. ***¿Qué devolverá la consulta anterior? ¿por qué?***

producto (4r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)
4	4	Cuarto	(NULL)

Porque del 1 al 3 ya los había introducido en esta sesión y el 4 se acaba de introducir.

5. ***¿Qué devolverá la consulta anterior? ¿por qué?***



producto (1r × 3c)

#	id	nombre	precio
1	5	Quinto	(NULL)

Porque todavía no se han confirmado los cambios en la sesión 1.

6. ***¿Qué devolverá la consulta anterior? ¿devuelve la fila con id 4? ¿y la fila con id 5? ¿por qué en cada caso?***

producto (3r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)

La fila con id 4 porque los cambios se han deshecho al volver al punto de guardado anterior, la fila con id 5 porque los cambios del commit de la sesión 2 se han deshecho en la sesión 1 por el rollback.

7. ***¿Qué devolverá la consulta anterior? ¿qué hace RELEASE SAVEPOINT? ¿La sentencia DELETE borra alguna fila?***

producto (5r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	2	Segundo	(NULL)
3	3	Tercero	(NULL)
4	4	444444	(NULL)
5	5	Quinto	(NULL)

La sentencia RELEASE SAVEPOINT elimina el punto de guardado creado. La sentencia DELETE borra la fila con id 4.

8. ***¿Qué devolverá la consulta anterior?***

producto (2r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	3	Tercero	(NULL)

9. ***¿Qué devolverá la consulta anterior?***

producto (2r × 3c)

#	id	nombre	precio
1	1	Primero	5
2	3	3333	(NULL)

10. ***¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 9? ¿por qué?***

producto (2r × 3c) producto (2r × 3c)

#	id	nombre	precio
1	1	Primero	(NULL)
2	3	Tercero	(NULL)

No coincide se encuentra en una transacción y tiene los datos sin modificar por la sesión 2 debido a esto.

11. ¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 9? ¿por qué?

producto (2r × 3c) producto (2r × 3c)

#	id	nombre	precio
1	1	Primero	5
2	3	3333	(NULL)

Coincide porque se ha terminado la transacción anterior y se han seleccionado los mismos ids que en la pregunta 9.

12. ¿Qué devolverá la consulta anterior? ¿sucede algo no esperado?

No devuelve nada, y no sucede nada inesperado porque se debe al FOR UPDATE de la sesión 1.

13. ¿Qué es lo que ha sucedido? ¿a qué se debe?

Se ha quedado esperando y ha lanzado un error de timeout. Esto se debe a que FOR UPDATE de la sesión 1 ha bloqueado las entradas que solicita la sesión 2.

14. ¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 11? ¿por qué?

producto (2r × 3c)

#	id	nombre	precio
1	1	Primero	5
2	3	3333	(NULL)

Si coincide porque se ha hecho rollback y se han cancelado los cambios.

15. ¿Qué devolverá la consulta anterior? ¿sucede algo no esperado?

producto (1r × 3c)

#	id	nombre	precio
1	3	3333	(NULL)

No sucede nada no esperado.

16. ¿Qué devolverá la consulta anterior? ¿coincide con el resultado de la pregunta 14? ¿por qué? ¿sucede algo no esperado?

No devuelve nada y no coincide con el 14.

17. ¿Qué es lo que ha sucedido? ¿a qué se debe?

Se ha quedado esperando porque la sesión 2 ha bloqueado la entrada con id 3 con el FOR UPDATE.

18. ¿Ha sucedido algo en la sesión 1 al hacer COMMIT en la sesión 2? ¿qué? ¿por qué?

Si, ha mostrado lo que pedía porque al hacer COMMIT en la sesión 2, se han desbloqueado las entradas bloqueadas.

19. ¿Qué devolverá la consulta anterior?

producto (2r × 3c)

#	id	nombre	precio
1	1	1	1
2	3	3	3

20. ¿Qué devolverá la consulta anterior? ¿coincide con lo que se devuelve en la pregunta 19? ¿por qué?

producto (2r × 3c)

#	id	nombre	precio
1	1	1	1
2	3	3	3

Si coincide porque en la sesión 1 se han confirmado los cambios con un COMMIT.

21. ¿Qué devolverá la consulta anterior?

producto (6r × 3c)

#	id	nombre	precio
1	1	1	1
2	2	Segundo	(NULL)
3	3	3	3
4	4	444444	(NULL)
5	5	Quinto	(NULL)
6	10	Décimo	10

22. ¿Qué devolverá la consulta anterior? ¿muestra la fila con id 10? ¿qué ha sucedido?

producto (5r × 3c)

#	id	nombre	precio
1	1	1	1
2	2	Segundo	(NULL)
3	3	3	3
4	4	444444	(NULL)
5	5	Quinto	(NULL)

No la muestra porque al no hacer un COMMIT, los cambios no se han aplicado y al sufrir una desconexión repentina, la transacción se ha cancelado, deshaciendo los cambios.

23. ¿Ha sucedido algo? ¿qué? ¿por qué? ¿qué se puede hacer para que no se produzcan estas situaciones?

Si, se ha desbloqueado la entrada con id 3 y la ha mostrado en la sesión 2 mientras en la sesión 1 ha dado el siguiente error:

Ha sucedido porque la sesión 1 ha intentado modificar los datos bloqueados por la sesión 2 y la sesión 2 ha intentado modificar los datos bloqueados por la sesión 1.

Se podrían cambiar las políticas para que un lector nunca bloqueé a un escritor, porque en esta secuencia, los bloqueos fueron leyendo las entradas.

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?).

25. ¿Ha sucedido algo en la sesión 2? ¿qué? ¿a qué se debe?

Se ha quedado esperando porque la sesión 1 tiene bloqueada la entrada con id 11 gracias a su transacción.