

use chollolandia;

-- 2. Crea una tabla temporal, una llamada productos2. Tendrá la misma estructura y contenido que la original. (0,2 puntos)

create temporary table productos2 select * from productos;

-- 3. Crea una transacción que realice las siguientes acciones sobre la tabla productos: (1,5 puntos)

-- 3.1 Inicia una transacción (0,15 p)

start transaction;

-- 3.2 Incrementar un 10% el precio de todos los productos con un stock menor o igual a 10. (0,2 p)

update productos set precio = precio*1.10 where stock <= 10;

-- 3.3 Introduce en la tabla productos, un nuevo producto, especificando las columnas sobre las que se va a insertar. (0,3 p)

insert into productos (id_producto, nombre, descripcion, precio, categoria, stock, id_vendedor)
values (21, 'balon', 'bota', 12, 'Deportes', 15,5);

-- 3.4 Elimina todos los productos de Hogar con un precio menor a 100€ (0,3 p)

delete from productos where (precio < 100) and (categoria = 'Hogar');

-- 3.5 Duplica el precio y establece un stock de 2 a todos los productos del vendedor llamado: Hugo Rodríguez (0,4 p)

update productos set precio = precio*2, stock = 2 where id_vendedor = (select id_vendedor from vendedores where nombre = 'Hugo Rodríguez');

-- 3.6 Cancela la transacción. (0,15 p)

rollback;

-- 4. Crea una transacción que realice las siguientes acciones: (1,75 puntos)

-- 4.1 Inicia una transacción

start transaction;

-- 4.2 Elimina todos los productos con id_producto entre 1 y 4. ¿Qué ha pasado? ¿Por qué? Explícalo. (0,3 p)

delete from productos where (id_producto >=1 and id_producto <=4);

-- Que han sido eliminados dichos productos y se ha actualizado en la tabla de carrito, desapareciendo los productos eliminados

-- 4.3 Elimina de todos los carritos el producto con id_producto 1 (0,3 p)

delete from carrito where id_producto = 1;

-- 4.4 Crea una tabla NO temporal llamada productos_aux con la misma estructura, no incluir su contenido. (0,3 p)

create table productos_aux like productos;

-- 4.5 Insertar en productos_aux todos los productos que contengan por 'Snapdragon' o cuesten 7500 (0,3 p)

insert into productos_aux select * from productos where precio = 7500 and nombre like '%Snapdragon%';

-- 4.6 Haz un ROLLBACK. ¿Qué ha pasado? Explícalo (0,55 p)

rollback;

-- Que la tabla produc_aux se ha vaciado al realizar el rollback, pero los cambios hechos desde el create table productos_aux han persistido por ser DDL

-- 5. Crea una transacción que realice las siguientes acciones: (2 puntos)

-- 5.1 Inicia una transacción

start transaction;

-- 5.2 Inserta en la tabla productos2 todas todos los productos de Elena Pérez que se encuentran en el carrito de Dominga Fernández (0,35 p)

```
insert into productos2 select * from productos
    where id_vendedor = (select id_vendedor from vendedores where nombre = 'Elena Pérez')
    and id_producto = (select id_producto from carrito where dniCliente = (select dniCliente
from clientes where nombre = 'DOMinga Fernández'));
```

-- 5.3 Insertar en la tabla productos2 todos los productos del carrito de aquellos clientes que nunca hayan realizado un pedido (0,3 p)

```
insert into productos2 select * from productos where id_producto in (select id_producto from
carrito where dniCliente not in (select dniCliente from pedidos));
```

-- 5.4 Crea un SAVEPOINT llamado punto1 (0,15 p)

savepoint punto1;

-- 5.5 Actualiza el estado a “Entregado” del pedido más reciente del cliente llamado Facundo Rodríguez (0,3 p)

```
update pedidos set estado = 'Entregado' where dniCliente = (select dniCliente from clientes where
nombre = 'Facundo Rodríguez') order by fecha_pedido desc limit 1;
```

-- 5.6 Elimina el Seat Panda 1998 de productos2 (0,15 p)

```
delete from productos2 where nombre = 'Seat Panda 1998';
```

-- 5.7 Actualiza todos los pedidos que tengan estado “En tránsito” y establécelos a “En aduanas” (0,2 p)

```
update pedidos set estado = 'En tránsito' where estado = 'En aduanas';
```

-- 5.8 ¿Podrías volver al estado en el que se encontraba la BD en el punto 5.4? En caso afirmativo, hazlo y explica que cambios ha habido con respecto al 5.7 (0,4 p)

```
rollback to savepoint punto1;
```

-- Si podría, porque estoy en medio de una transacción y después del savepoint solo he realizado sentencias DML, por lo que se han desecho los cambios desde el 5.7 hasta 5.5

-- 5.9 Confirma la transacción (0,15 p)

```
commit;
```

-- 6. Crea un usuario con el nombre Ramona, se podrá conectar desde cualquier host y se le establecerá la contraseña ‘1234’.

-- Además, una vez pasado un año (no bisiesto), caducará y no podrá establecer la misma hasta haberla cambiado 4 veces. (0,5 puntos)

```
create user ramona identified by '1234' password expire interval 365 day password history 4;
```

-- 7. Añade permisos de lectura y escritura sobre todas las tablas de todas las bases de datos a Ramona. (0,25 puntos)

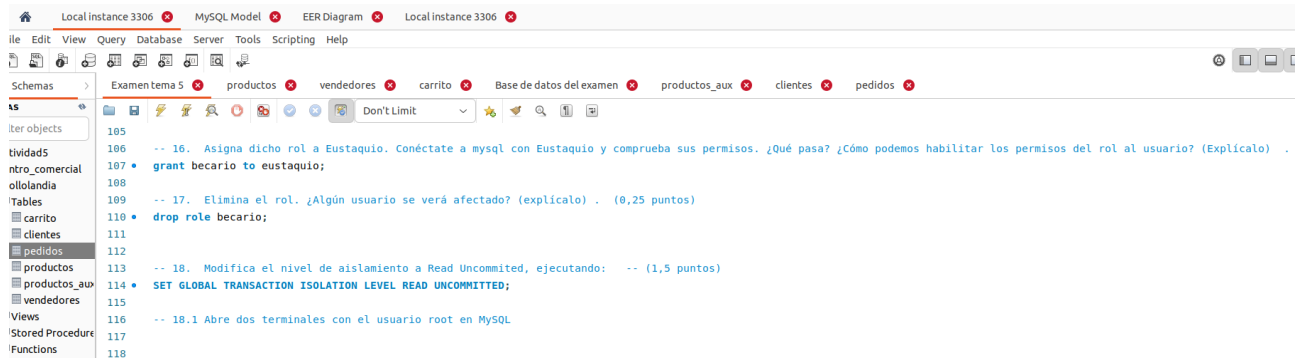
```
grant select, insert, update on *.* to ramona;
```

-- 8. Muestra los permisos de Ramona (0,2 puntos)

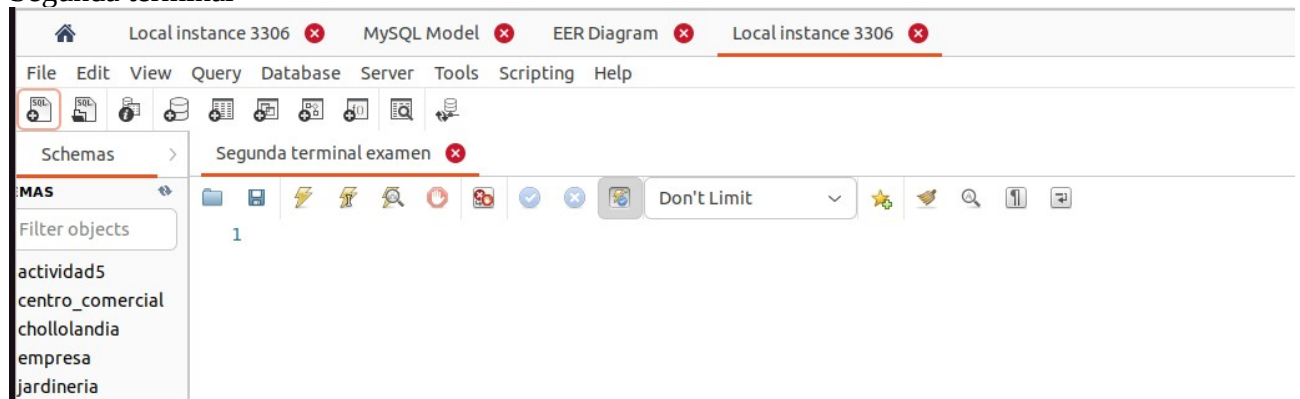
```
show grants for ramona;
```

- 9. Haz las modificaciones necesarias para que Ramona pueda crear tablas en todas las bases de datos. ¿Habrá más afectados? Explícalo (0,2 puntos)
grant create on *.* to ramona;
- No habrá mas afectados, ya que los cambios los estamos haciendo al usuario ramona, no a ningún roll
- 10. Establece a Ramona un máximo de 10 peticiones al servidor y 3 actualizaciones. (0,2 puntos)
alter user ramona with max_queries_per_hour 10 max_updates_per_hour 3;
- 11. Ramona ha sido despedida, bloquea su cuenta para que no pueda acceder al sistema. (0,2 puntos)
alter user ramona account lock;
- 12. Crea un nuevo usuario llamado Eustaquio su contraseña será 'habichuela' (0,15 puntos)
create user eustaquio identified by 'habichuela';
- 13. Establece permisos de lectura sobre todas las tablas de la base de dato 'chollolandia' . (0,2 puntos)
grant select on chollolandia.* to eustaquio;
- 14. Crea un rol llamado 'becario'. . (0,15 puntos)
create role becario;
- 15. Todos aquellos que sean becarios podrán visualizar todas las tablas del sistema,
-- podrán insertar o modificar los registros de las diferentes tablas de la base de datos. . (0,25 puntos)
grant select, insert, update on *.* to becario;
- 16. Asigna dicho rol a Eustaquio. Conéctate a mysql con Eustaquio y comprueba sus permisos. ¿Qué pasa? ¿Cómo podemos habilitar los permisos del rol al usuario? (Explícalo) . (0,5 puntos)
grant becario to eustaquio;
- 17. Elimina el rol. ¿Algún usuario se verá afectado? (explícalo) . (0,25 puntos)
drop role becario;
- Si esutaquio, que directamente no pierde privilegios, pero al desaparecer el rol de becario, pierde las capacidades que este le daba
- 18. Modifica el nivel de aislamiento a Read Uncommitted, ejecutando: -- (1,5 puntos)
SET GLOBAL TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
- 18.1 Abre dos terminales con el usuario root en MySQL

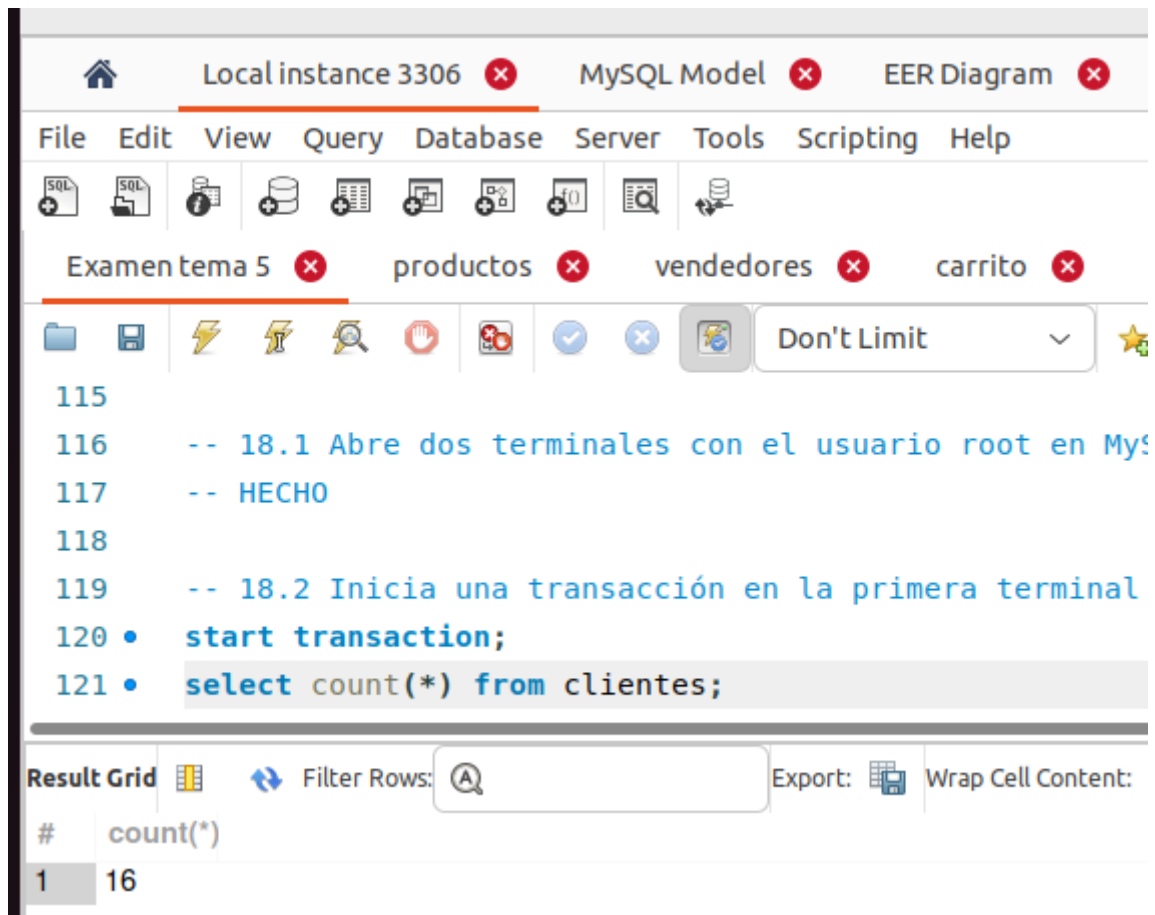
Primera terminal



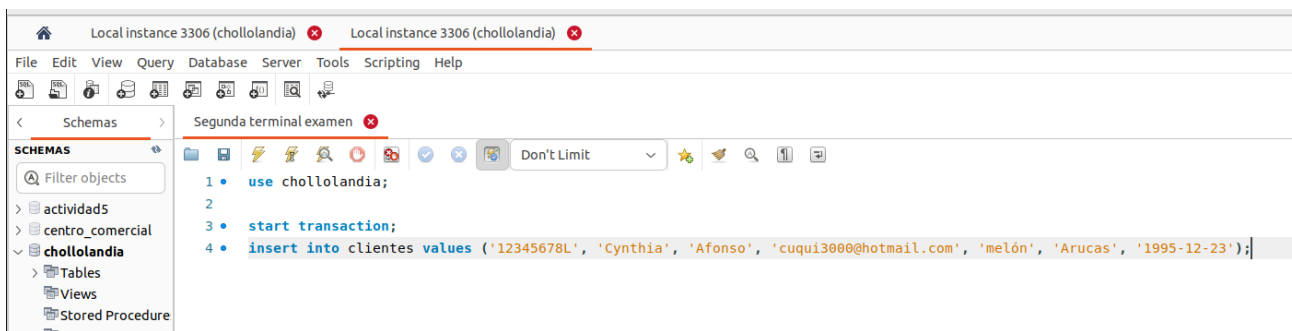
Segunda terminal



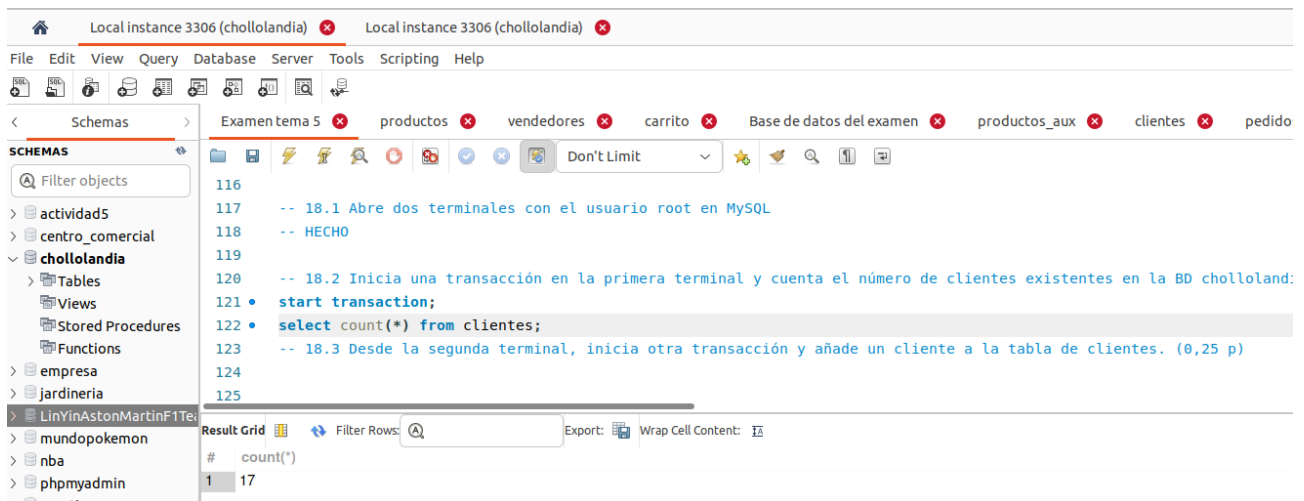
-- 18.2 Inicia una transacción en la primera terminal y cuenta el número de clientes existentes en la BD chollolandia. (0,25 p)



-- 18.3 Desde la segunda terminal, inicia otra transacción y añade un cliente a la tabla de clientes. (0,25 p)



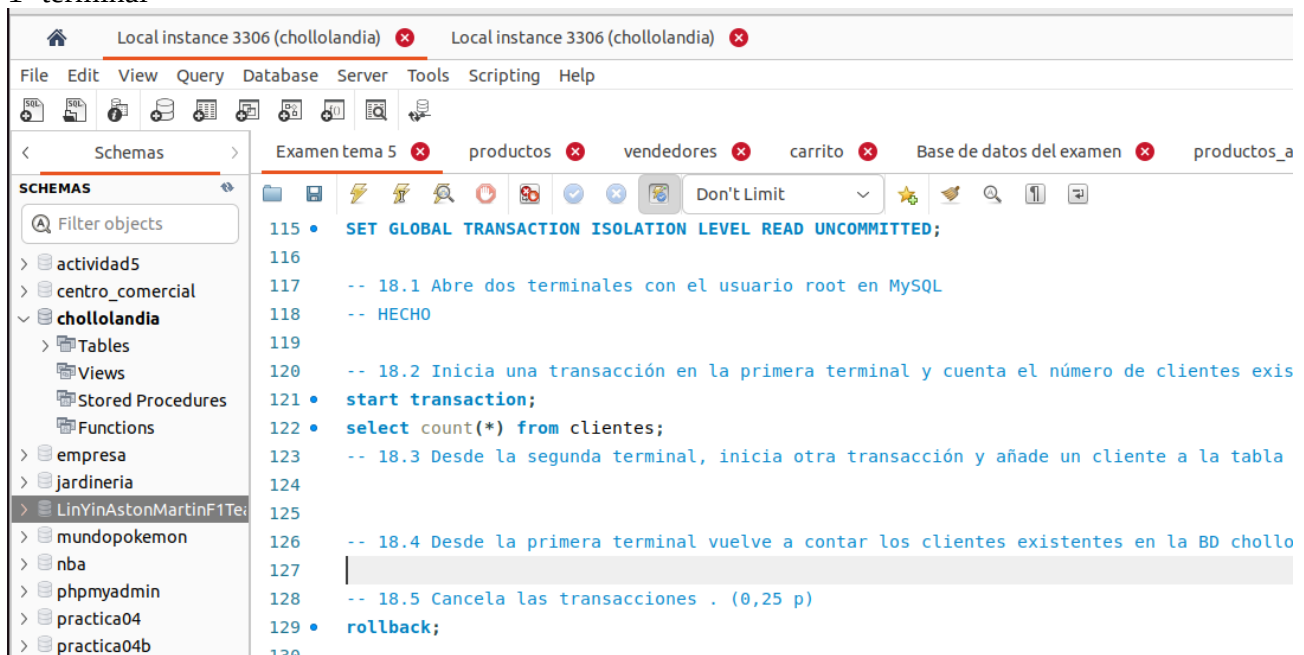
-- 18.4 Desde la primera terminal vuelve a contar los clientes existentes en la BD chollolandia. ¿Qué ha pasado? ¿Qué tipo de problema ha ocurrido? Especifica su nombre (1 p)



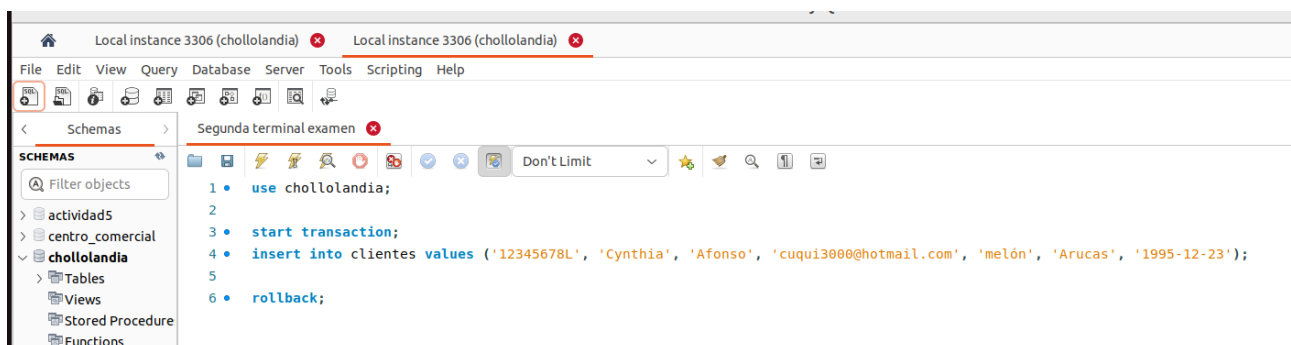
En la segunda cuenta ha aparecido un cliente más, ha ocurrido un dirty read por estar el sistema de aislamiento en read uncommited, el terminal 1 ha visto un cambio realizado por el terminal 2 aunque este no haya hecho el commit.

-- 18.5 Cancela las transacciones . (0,25 p)

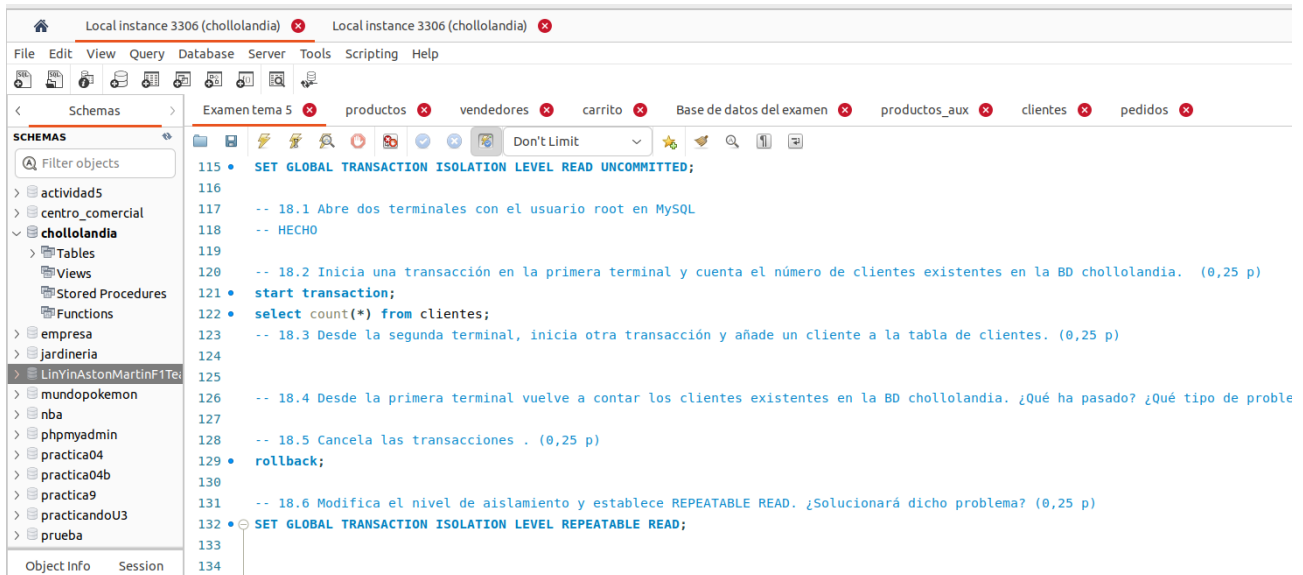
1º terminal



2º terminal



-- 18.6 Modifica el nivel de aislamiento y establece REPEATABLE READ. ¿Solucionará dicho problema? (0,25 p)



Si solucionará dicho problema, ya que este modo de aislamiento solo permite el problema del phantom read.