

# SQL - Mysql

Presentación enfocada en el conocimiento de la herramienta “transaction”



# INDEX

- 1- ¿Qué es una transacción? + Expiación general + ejemplo
- 2- Principios ACID
- 3- Isolation levels y la importancia que acarrean
- 4- Bloqueos y deadlock o abrazo mortal
- 5- Instrucciones de manejo de transacciones
- 6-Ejercicio práctico de transacciones



# ¿Qué es una transacción?

Una transacción no es mas que una herramienta que nos brinda un Gestor de bases de datos para agrupar varias acciones y hacer que estas, se realicen al mismo tiempo

Es decir, permite establecer una serie de acciones para que se resuelvan al mismo tiempo.

Tambien añadir que no todos los motores de mysql permiten las transacciones, InnoDB es por antonomasia el motor de BBDD con el que se realizan, motores como Myisam, que era el predeterminado en versiones anteriores de Mysql, no tiene la función de crear transacciones



# Ejemplo cuenta bancaria

- El ejemplo más típico y mas ilustrativo es el de las cuentas bancarias.

Tº	SUCURSAL A	SUCURSAL B	SALDO INICIAL CUENTA
0			1.000 €
1	Disminuye 200 €		800 €
2		Aumenta 300 €	1.300 €
3	Confirma operación		800 €
4		Confirma operación	1.300 €

- Cuando se compra un producto, estamos nada mas y nada menos que realizando una transacción entre nosotros y el vendedor del bien que nosotros queremos adquirir.
- En el caso de que internamente no se realizara una transacción para poder realizar el pago. Ocurrirían problemas.(Explica Iván)



# Principios ACID

- Las transacciones siguen una serie de principios para ser efectivas:
- Por sus siglas se conoce como ACID, y cada una de estas letras entraña un significado oculto.
- A: Atomicidad. Todas las ordenes dentro de una transacción deben realizarse, si alguna falla o no se realiza, la transacción no se realiza.
- C: Consistencia. Si la transacción que vamos a realizar, compromete la coherencia de los datos de la base de datos, la transacción no se realiza
- I: "Isolation"(Aislamiento): Este principio viene a decir que toda transacción debe realizarse independientemente de las otras acciones que se realizan en la base de datos
- D: Duración. Cuando una transacción se realiza con éxito , los cambios en la base de datos perduraran aun cuando el sistema falle o el propio terminal falle



# Isolation levels

- Hemos hablado de aislamiento en la anterior diapositiva. Pero lo que no os he explicado es que el aislamiento va por niveles, y que dependiendo del nivel que elijamos, obtendremos unas características diferentes. Están estructurados para que a mayor nivel de aislamiento: mayor seguridad, pero menor fluidez, será tarea del DBA establecer un nivel que le aporte lo mejor de los dos “mundos”.
- También comentar que el predeterminado que utiliza Mysql es “repetible read / lectura repetible”



# Isolation levels

- Lectura no confirmada / read uncommitted: Este nivel de aislamiento permite que cuando se haga una transacción y se realice una acción de modificación o inserción de datos, sea visible por otro sin haber hecho el primero sin haber acabado la transacción, por eso también se llama de “lectura sucia”, porque ve valores que podrían modificar-se en un futuro o no. (Ejemplo breve)
- Lectura confirmada / Read committed: Este nivel viene para paliar el problema de la “lectura sucia”, que se realiza en el nivel anterior. En este nivel al iniciar una transacción e insertar o modificar un valor, otro sujeto que este viendo el contenido de la tabla que se este modificando, ( `select * from [tabla_transaccion]`). No observará el contenido que se esta modificando, sino el antiguo, y cuando se realice la consulta /commit, se podrá ver el contenido nuevo. Pero esto no resuelve una serie de problemas como la “lectura repetida”, [ejemplo+breve\_explicación](el preferido para los sistemas gestores de bases de datos)



# Isolation levels

- Lectura repetida / Repeatable read: Con este nuevo nivel de aislamiento arreglamos el problema de la “lectura repetida”. Ahora, cada vez que queramos modificar un valor de una tabla en una transacción, dicho valor en la tabla se bloqueará para que no podamos ni ver-lo ni modificar-lo.
- Secuenciable / Serializable: Este es el nivel que mayor aislamiento proporciona y también el que peor fluidez ofrece para la base de datos. Es verdad que elimina las lecturas fantasma, dado cuando se modifica un valor de la fila, este nivel bloquea la tabla entera HASTA que se realiza el commit en la transacción. Las transacciones en este nivel se realizan de manera secuencial(de ahí el nombre), dado que si quieres hacer una transacción sobre la tabla, deberas esperar hasta que acabe la transacción anterior.





# Bloqueos y dreadblock

- Hemos hablado de bloqueos en los dos niveles de aislamiento anteriores y hemos visto que la información se bloquea para un buen funcionamiento de la base de datos.
- Las sentencias UPDATE e INSERT, provocan un bloqueo natural cuando se introducen datos en Mysql, bloqueando estas las tablas hasta que se realice el commit.
- Pero existen alternativas para bloquear una BD, como:

LOCK TABLES, que bloquea la tabla entera, aunque, como pasa en el cuarto nivel de aislamiento, reduce la concurrencia.

- Hay también una forma para bloquear la instrucción de la tabla en la cual estamos trabajando. El truco es poner For update / Lock in share mode en una sentencia de select, tal que así:

- Se suele utilizar para c

```
SELECT opciones de la sentencia SELECT  
[FOR UPDATE | LOCK IN SHARE MODE]
```



# Dreadlock o abrazo mortal

- El dreadlock o abrazo mortal se da cuando dos tablas se bloquean al mismo tiempo. Observad la siguiente imagen:

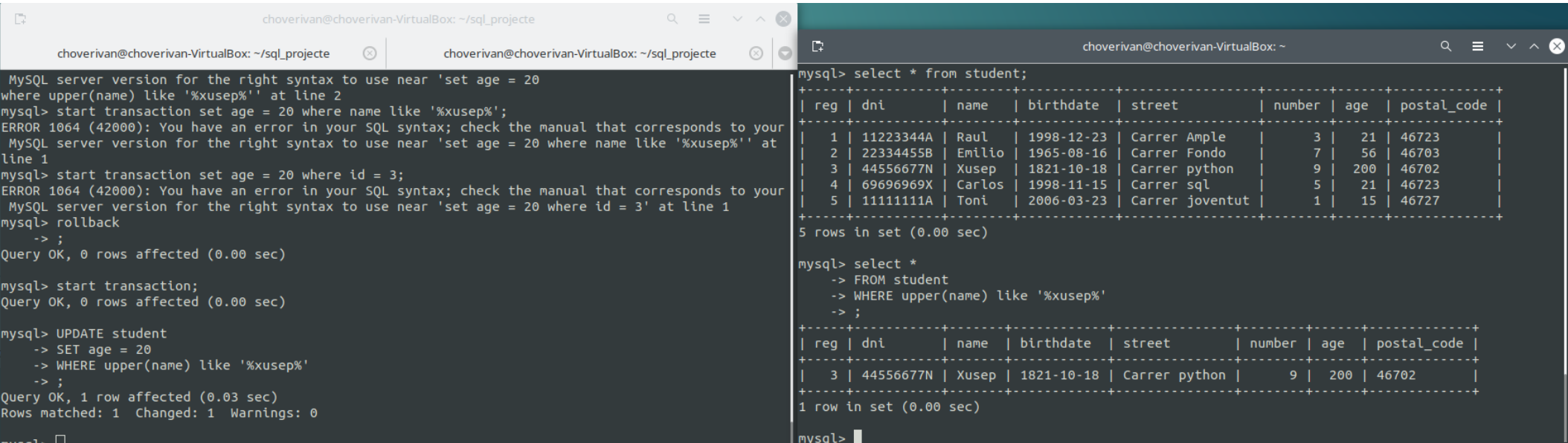
- 

Tº	SUCURSAL A		SUCURSAL B		SITUACION
1	Modifica	saldo			La cuenta X queda bloqueada por A
	cuenta X				
2			Modifica	saldo	La cuenta Y queda bloqueada por B
			cuenta Y		
3	Modifica	saldo			A queda a la espera pues Y está bloqueada
	cuenta Y				
4			Modifica	saldo	B queda a la espera pues X está bloqueada
			cuenta X		



# Manejo de transacciones

- **START TRANSACTION:** es la orden con la que empezaremos a iniciar nuestra transacción, aunque hay otras maneras como empezar con un: “Begin”. O poniendo la variable @@autocommit = 0



```
choverivan@choverivan-VirtualBox: ~/sql_projecte
MySQL server version for the right syntax to use near 'set age = 20
where upper(name) like '%xusep%' at line 2
mysql> start transaction set age = 20 where name like '%xusep%';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'set age = 20 where name like '%xusep%' at
line 1
mysql> start transaction set age = 20 where id = 3;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near 'set age = 20 where id = 3' at line 1
mysql> rollback
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE student
-> SET age = 20
-> WHERE upper(name) like '%xusep%'
-> ;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
```

```
choverivan@choverivan-VirtualBox: ~
mysql> select * from student;
+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name  | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 11223344A | Raul  | 1998-12-23 | Carrer Ample | 3 | 21 | 46723 |
| 2 | 22334455B | Emilio | 1965-08-16 | Carrer Fondo | 7 | 56 | 46703 |
| 3 | 44556677N | Xusep | 1821-10-18 | Carrer python | 9 | 200 | 46702 |
| 4 | 69696969X | Carlos | 1998-11-15 | Carrer sql | 5 | 21 | 46723 |
| 5 | 11111111A | Toni   | 2006-03-23 | Carrer joventut | 1 | 15 | 46727 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select *
-> FROM student
-> WHERE upper(name) like '%xusep%'
-> ;
+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name  | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+
| 3 | 44556677N | Xusep | 1821-10-18 | Carrer python | 9 | 200 | 46702 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

```
mysql> select @@autocommit;
+-----+
| @@autocommit |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

- Commit: Es la orden para terminar la transacción y realizar los cambios. Cualquier tipo de bloqueo durante la transacción queda liberado.

```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE student
-> SET age = 20
-> WHERE upper(name) like '%xusep%'
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM student WHERE upper(name) like '%xusep%';
+-----+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 44556677N | Xusep | 1821-10-18 | Carrer python | 9 | 20 | 46702 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> commit
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
mysql> select *
-> FROM student
-> WHERE upper(name) like '%xusep%'
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 44556677N | Xusep | 1821-10-18 | Carrer python | 9 | 200 | 46702 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT *
-> FROM student
-> WHERE upper(name) like '%xusep%'
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 44556677N | Xusep | 1821-10-18 | Carrer python | 9 | 20 | 46702 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

```
mysql> set autocommit=0
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE subject
-> SET description = 'mansanet vlan'
-> WHERE code = 'PAX'
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from subject;
+-----+-----+
| code | description |
+-----+-----+
| BBDD | the best subject |
| HDW  | git status -s |
| ISO  | linux sucks |
| LMI  | Garrido hacker |
| PAX  | mansanet vlan |
+-----+-----+
5 rows in set (0.00 sec)

mysql> rollback
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from subject;
+-----+-----+
| code | description |
+-----+-----+
| BBDD | the best subject |
| HDW  | git status -s |
| ISO  | linux sucks |
| LMI  | Garrido hacker |
| PAX  | alt f4 |
+-----+-----+
5 rows in set (0.00 sec)
```



# Rollback

- Rollback: És la instrucció que permet desfer tots els canvis realitzats en la transacció i donar-la per finalitzada

```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE student
-> SET age = 20
-> WHERE upper(name) like '%xusep%'
-> ;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> rollback
-> ;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
-> FROM student
-> WHERE upper(name) like '%xusep%'
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| reg | dni      | name | birthdate | street      | number | age | postal_code |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3   | 44556677N | Xusep | 1821-10-18 | Carrer python | 9      | 200 | 46702      |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```



# SAVEPOINT y Rollback to savepoint

- Savepoint [nombre]: Crea un punto de salvaguarda al que se puede retroceder con “Rollback to savepoint [nombre]”.
- Rollback to savepoint [nombre]: Con esta acción vuelves al punto de guardado definido y todo aquello que has construido después de este punto, se pierde. Cabe decir que esta herramienta no debe usarse de manera habitual.



```
mysql> start transaction
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE student
-> SET age = 22
-> WHERE upper(name) like '%raul%'
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SAVEPOINT raul
-> ;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> UPDATE student
-> SET age = 22
-> WHERE upper(name) like '%carlos%'
-> ;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> ROLLBACK TO SAVEPOINT raul;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> commit;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from student;
```

reg	dni	name	birthdate	street	number	age	postal_code
1	11223344A	Raul	1998-12-23	Carrer Ample	3	22	46723
2	22334455B	Emilio	1965-08-16	Carrer Fondo	7	56	46703
3	44556677N	Xusep	1821-10-18	Carrer python	9	20	46702
4	69696969X	Carlos	1998-11-15	Carrer sql	5	21	46723
5	11111111A	Toni	2006-03-23	Carrer joventut	1	15	46727

5 rows in set (0.00 sec)





# Set transaction

- Set transaction: permite definir el nivel de aislamiento para la transacción

```
mysql> set @@autocommit=0  
-> ;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;  
Query OK, 0 rows affected (0.00 sec)
```

