

BASES DE DATOS AVANZADAS

Transacciones en MYSQL

Ejercicio Transacciones

```
mysql> CREATE TABLE innotest (campo INT NOT NULL PRIMARY KEY) TYPE = InnoDB;
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> INSERT INTO innotest VALUES(1);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> INSERT INTO innotest VALUES(2);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO innotest VALUES(3);
Query OK, 1 row affected (0.04 sec)
```

```
mysql> SELECT * FROM innotest;
```

```
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
+-----+
3 rows in set (0.00 sec)
```

De acuerdo, nada espectacular. Ahora cómo usar transacciones.

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

```
mysql> INSERT INTO innotest VALUES(4);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM innotest;
```

```
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
|     4 |
+-----+
4 rows in set (0.00 sec)
```

Si en este momento se ejecuta un ROLLBACK, la transacción no será completada, y los cambios realizados sobre la tabla no tendrán efecto.

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SELECT * FROM innotest;
```

```
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
+-----+
3 rows in set (0.00 sec)
```

Ahora qué sucede si se pierde la conexión al servidor antes de que la transacción sea completada.

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

```
mysql> INSERT INTO innotest VALUES(4);
```

BASES DE DATOS AVANZADAS

Transacciones en MYSQL

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
|     4 |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> EXIT;
Bye
```

Cuando se obtenga de nuevo la conexión, podemos verificar que el registro no se insertó, ya que la transacción no fue completada.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 449 to server version: 4.0.13
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
+-----+
3 rows in set (0.00 sec)
```

Ahora repetir la sentencia INSERT ejecutada anteriormente, pero haremos un COMMIT antes de perder la conexión al servidor al salir del monitor de MySQL.

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

mysql> INSERT INTO innotest VALUES(4);
Query OK, 1 row affected (0.00 sec)
mysql> COMMIT;
Query OK, 0 rows affected (0.02 sec)

mysql> EXIT;
Bye
```

Una vez que se comete o confirma la transacción con un COMMIT, la transacción es completada, y todas las sentencias SQL que han sido ejecutadas previamente afectan de manera permanente a las tablas de la base de datos.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 450 to server version: 4.0.13
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
|     1 |
|     2 |
|     3 |
|     4 |
+-----+
4 rows in set (0.00 sec)
```

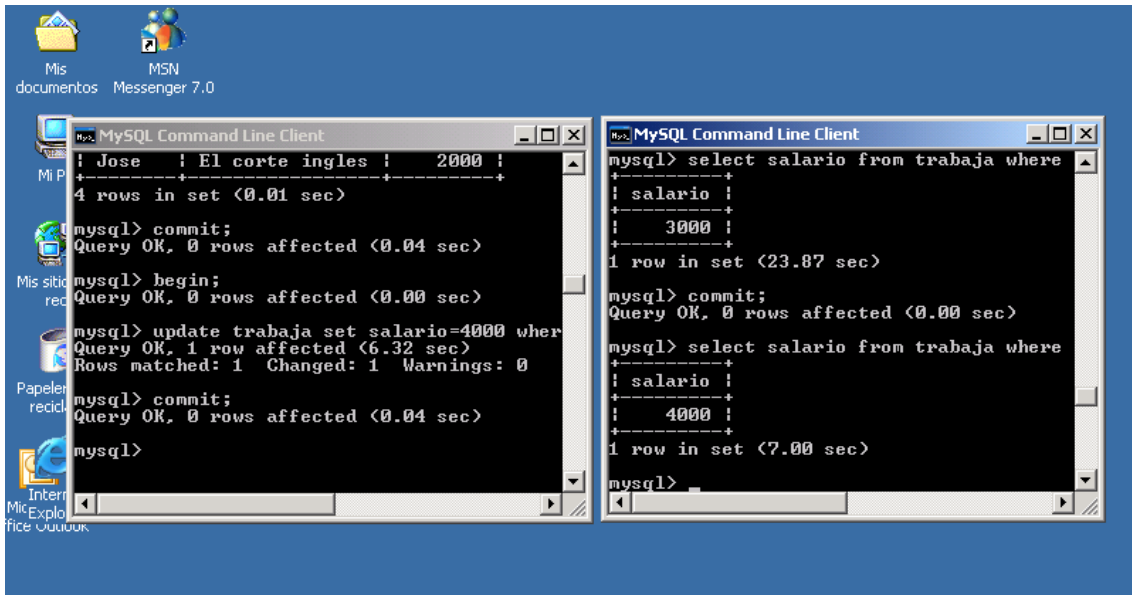
BASES DE DATOS AVANZADAS

Transacciones en MYSQL

Lecturas consistentes

Por default, las tablas InnoDB ejecutan una lectura consistente (*consistent read*). Esto significa que cuando una sentencia SELECT es ejecutada, MySQL regresa los valores presentes en la base de datos hasta la transacción más reciente que ha sido completada. Si alguna transacción está en progreso, los cambios hechos por alguna sentencia INSERT o UPDATE no serán reflejados. Sin embargo, existe una excepción: las transacciones abiertas si pueden ver sus propios cambios. Para demostrar esto, se necesita establecer dos conexiones al servidor MySQL.

Abrir dos sesiones de mysql a la vez:



Agregar un registro dentro de una transacción desde la primera conexión:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 524 to server version: 4.0.13

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

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

mysql> INSERT INTO innotest VALUES(5);
Query OK, 1 row affected (0.00 sec)
```

Ahora, desde una segunda conexión se consultan los datos de nuestra tabla.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 525 to server version: 4.0.13

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
| 1     |
| 2     |
| 3     |
| 4     |
+-----+
4 rows in set (0.00 sec)
```

BASES DE DATOS AVANZADAS

Transacciones en MYSQL

Como se puede observar, el registro que se ha insertado desde la primera conexión no es regresado puesto que forma parte de una transacción que no ha sido completada. Ahora, desde la primera conexión se ejecuta la misma consulta SELECT.

```
mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----+
5 rows in set (0.00 sec)
```

Después de completar la transacción con una sentencia COMMIT en la 1ra conexión se puede verificar que desde la 2da conexión los cambios ya son visibles.

```
mysql> SELECT * FROM innotest;
+-----+
| campo |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
+-----+
5 rows in set (0.00 sec)
```

Otro ejemplo

En el ejemplo anterior se ha usado únicamente sentencias INSERT, sin embargo, sucede lo mismo con sentencias UPDATE o DELETE. Crear una sencilla tabla llamada ventas.

```
mysql> CREATE TABLE ventas(id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
-> producto VARCHAR(30) NOT NULL, cantidad TINYINT NOT NULL) TYPE = InnoDB;
Query OK, 0 rows affected (0.96 sec)
```

Insertar un registro.

```
mysql> INSERT INTO ventas VALUES(0,'Mouse USB',3);
Query OK, 1 row affected (0.16 sec)
```

```
mysql> SELECT * FROM ventas;
+-----+-----+-----+
| id | producto      | cantidad |
+-----+-----+-----+
| 1 | 'Mouse USB'   | 3        |
+-----+-----+-----+
1 row in set (0.01 sec)
```

Ahora iniciar una transacción con la sentencia BEGIN.

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

Actualizar el registro.

```
mysql> UPDATE ventas SET cantidad=4 WHERE id=1;
Query OK, 1 row affected (0.07 sec)
Líneas correspondientes: 1 Cambiadas: 1 Avisos: 0
```

BASES DE DATOS AVANZADAS

Transacciones en MYSQL

Verificar que los cambios han sucedido.

```
mysql> SELECT * FROM ventas;
+-----+-----+-----+
| id | producto          | cantidad |
+-----+-----+-----+
| 1  | 'Mouse USB       | 4        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Si se quiere deshacer los cambios, entonces ejecutamos un ROLLBACK.

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.06 sec)
```

Verificar que se deshicieron los cambios.

```
mysql> SELECT * FROM ventas;
+-----+-----+-----+
| id | producto          | cantidad |
+-----+-----+-----+
| 1  | 'Mouse USB       | 3        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Actualizar el registro usando otra transacción.

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

mysql> UPDATE ventas SET cantidad=2 WHERE id=1;
Query OK, 1 row affected (0.00 sec)
Líneas correspondientes: 1 Cambiadas: 1 Avisos: 0
```

```
mysql> SELECT * FROM ventas;
+-----+-----+-----+
| id | producto          | cantidad |
+-----+-----+-----+
| 1  | 'Mouse USB       | 2        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Confirmar la transacción. Después de ejecutar el COMMIT los cambios serán permanentes y definitivos.

```
mysql> COMMIT;
Query OK, 0 rows affected (0.05 sec)

mysql> SELECT * FROM ventas;
+-----+-----+-----+
| id | producto          | cantidad |
+-----+-----+-----+
| 1  | 'Mouse USB       | 2        |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Bibliografía: Mysql-Hispano
Artículo: Transacciones