

## 6. Ejercicios para practicar: transacciones

### Ejercicio 1

Si iniciamos una transacción y de pronto cerramos la sesión, qué operación hace el servidor: ¿un COMMIT, un ROLLBACK o indefinido?

#### Solución

Con la variable AUTOCOMMIT con su valor por defecto (activada) el sistema gestor actúa como si ejecutara un ROLLBACK (no haciendo permanentes los cambios).

### Ejercicio 2

¿Qué diferencia hay entre START TRANSACTION y BEGIN?

#### Solución

Con START TRANSACTION, *autocommit* permanece no habilitado hasta el final de la transacción con COMMIT o ROLLBACK. Después, el modo *autocommit* vuelve a su estado previo.

BEGIN y BEGIN WORK se soportan como alias para START TRANSACTION para iniciar una transacción. La sentencia START TRANSACTION corresponde a la sintaxis SQL estándar y es la forma recomendada para iniciar una transacción.

### Ejercicio 3

¿Qué ocurre si en la mitad de una transacción sin confirmar modificamos la variable *autocommit* dándole el valor 1?

#### Solución

Al ejecutar START TRANSACCTION, el sistema se comporta como si la variable *autocommit* tomase el valor 0, aunque a efectos prácticos ignora su contenido, por lo que almacenar el valor 1 de forma manual no tiene ningún efecto.

```
START TRANSACTION;
INSERT INTO tbl_ProbaTransacciones ( id, s, si) VALUES (7, "setimo", NULL);
SET AUTOCOMMIT=1;
SELECT * FROM tbl_ProbaTransacciones;
```

	id	s	si
▶	1	primeiro	NULL
	2	segundo	NULL
	3	terceiro	NULL
	4	cuarto	NULL
	5	quinto	NULL
	6	sexto	NULL
	7	setimo	NULL

SHOW VARIABLES LIKE 'AUTOCOMMIT';

	Variable_name	Value
▶	autocommit	ON

ROLLBACK;  
SELECT \* FROM tbl\_ProbaTransacciones;

	id	s	si
▶	1	primeiro	NULL
	2	segundo	NULL
	3	terceiro	NULL
	4	cuarto	NULL
	5	quinto	NULL
	6	sexto	NULL

## Ejercicio 4

Continuando con la base de datos *bd\_ProbaTransacciones* creada y empleada hasta ahora, ejecuta el siguiente código y explica qué es lo que ocurre:

```

USE `bd_ProbaTransaccions`;
SET AUTOCOMMIT=1;
INSERT INTO tbl_ProbaTransaccions ( id, s, si) VALUES (7, "¿conformouse este dato?", NULL);
SET AUTOCOMMIT=0;
CREATE TABLE tbl_ProbaTransaccions2 (id INT NOT NULL PRIMARY KEY, s2 VARCHAR (50)) ENGINE = InnoDB ;
INSERT INTO tbl_ProbaTransaccions2 ( id, s2) VALUES (1, "¿existe tbl_ProbaTransaccions2?");
ROLLBACK;
SELECT * FROM tbl_ProbaTransaccions;
SELECT * FROM tbl_ProbaTransaccions2;

```

### Solución:

En este ejemplo, al poner de forma manual la variable *autocommit* a uno, el INSERT en la tabla *tbl\_ProbaTransaccions* se confirmará de manera automática.

	id	s	si
▶	1	primeiro	NULL
	2	segundo	NULL
	3	terceiro	NULL
	4	cuarto	NULL
	5	quinto	NULL
	6	sexto	NULL
	7	¿conformouse este dato?	NULL

La siguiente instrucción inhabilita la variable *autocommit*, por lo que la transacción permanecerá abierta hasta el ROLLBACK, deshaciendo las sentencias CREATE y el INSERT.

Pero al ejecutarse los SELECT se observa que la tabla *tbl\_ProbaTransaccions2* fue creada, debido a que ciertas instrucciones (como CREATE) se confirman automáticamente y de forma independiente al contenido de la variable *autocommit*.

id	s2
----	----

### Ejercicio 5

Continuando con la base de datos *bd\_ProbaTransaccions*, ejecuta el siguiente código y explica qué es lo que ocurre:

```

USE `bd_ProbaTransaccions`;
SET AUTOCOMMIT=0;
INSERT INTO tbl_ProbaTransaccions ( id, s, si) VALUES (8, "... comeza
a proba", NULL);
INSERT INTO tbl_ProbaTransaccions ( id, s, si) VALUES (9, "... supera-
se o limite do tamaño de numero de caracteres da cadea, prodúcese un
error...", NULL);
COMMIT;
SELECT * FROM tbl_ProbaTransaccions;

```

### Solución:

En este ejemplo, al inhabilitar de forma manual la variable *autocommit*, todas las instrucciones pasan a ser una única transacción hasta encontrar un COMMIT (o un ROLLBACK).

Al producirse un error, se puede pensar que de manera automática se hará un ROLLBACK de las instrucciones ya ejecutadas de la transacción, pero MésQL no actúa de este modo ya que incluirá en la tabla la fila con el *id*=8 a pesar de que la inserción *id*=9 da error.

	id	s	si
▶	1	primeiro	NULL
	2	segundo	NULL
	3	terceiro	NULL
	4	cuarto	NULL
	5	quinto	NULL
	6	sexto	NULL
	7	¿conformouse este dato?	NULL
	8	... comeza a proba	NULL