

Transacciones

[Introducción](#)

[Tipos de transacciones](#)

[Estados de una transacción](#)

[Niveles de Aislamiento](#)

[Ejemplo](#)

Introducción

Definición

Una transacción es una unidad única de trabajo. Si una transacción tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y se convierten en una parte permanente de la base de datos. Si una transacción encuentra errores y debe cancelarse o revertirse, se borran todas las modificaciones de los datos.

Una transacción debe exhibir cuatro propiedades conocidas como (ACID):

- Atomicidad (Atomicity)
- Coherencia (Consistency)
- Aislamiento (Isolation)
- Durabilidad (Durability)

Atomicidad: Una transacción debe ser una unidad atómica de trabajo, tanto si se realizan todas sus modificaciones en los datos, como si no se realiza ninguna de ellas.

Coherencia: Cuando finaliza, una transacción debe dejar todos los datos en un estado coherente. En una base de datos relacional, se deben aplicar todas las reglas a las modificaciones de la transacción para mantener la integridad de todos los datos. Todas las estructuras internas de datos, como índices de árbol b o listas doblemente vinculadas, deben estar correctas al final de la transacción.

Aislamiento: Las modificaciones realizadas por transacciones simultáneas se deben aislar de las modificaciones llevadas a cabo por otras transacciones simultáneas. Una transacción reconoce los datos en el estado en que estaban antes de que otra transacción simultánea los modificara o después de que la segunda transacción haya concluido, pero no reconoce un estado intermedio.

Durabilidad: Una vez concluida una transacción, sus efectos son permanentes en el sistema. Las modificaciones persisten aún en el caso de producirse un error del sistema.

Tipos de transacciones

Para agrupar varias sentencias **SQL** en una única transacción, disponemos de los siguientes métodos:

- Transacciones explícitas
- Transacciones implícitas

Explícitas:

Son aquellas en que se define explícitamente el inicio y el final de la transacción. En un motor SQL utilizan las instrucciones **START TRANSACTION**, **COMMIT** Y **ROLLBACK**.

START TRANSACTION

Marca el punto de inicio de una transacción explícita para una conexión.

COMMIT

Se utiliza para finalizar una transacción correctamente si no hubo errores. Todas las modificaciones de datos realizadas en la transacción se convierten en parte permanente de la base de datos. Se liberan los recursos ocupados por la transacción.

ROLLBACK

Se utiliza para eliminar una transacción en la que se encontraron errores. Todos los datos modificados por la transacción vuelven al estado en el que estaban al inicio de la transacción. Se liberan los recursos ocupados por la transacción.

Ejemplo:

START TRANSACTION

```
update cuenta
set monto = monto - 100
where id_cuenta = 101;

update cuenta
set monto = monto + 100
where id_cuenta = 102;
```

COMMIT;

Implícitas:

Cuando una conexión funciona en modo de transacciones implícitas, MySQL inicia automáticamente una nueva transacción después de confirmar o revertir la transacción actual. No tiene que realizar ninguna acción para delinear el inicio de una transacción, sólo tiene que confirmar o revertir cada transacción. El modo de transacciones implícitas genera una cadena continua de transacciones.

Ejemplo:

```
update cuenta
set monto = monto - 100
where id_cuenta = 101;

update cuenta
set monto = monto + 100
where id_cuenta = 102;

COMMIT;
```

Modo autocommit

En el modo de autocommit, no es necesario realizar un START TRANSACTION ni un COMMIT. Cada operación hace un commit automático sin que sea posible volver atrás la operación con un ROLLBACK.

Para definir el modo de autocommit se debe ejecutar el siguiente comando :

```
SET autocommit=1;
```

para deshabilitarlo :

```
SET autocommit=0;
```

Estados de una transacción

Activa (estado inicial): la transacción permanece en este estado durante su ejecución.

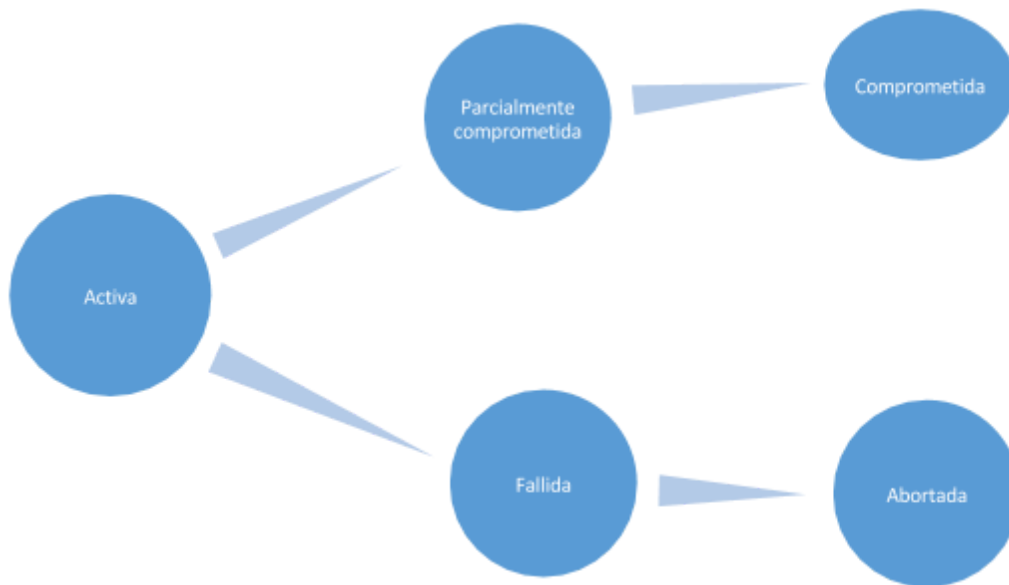
Parcialmente Comprometida: la transacción pasa a este estado cuando acaba de realizar la última instrucción.

Fallida: la transacción pasa a este estado tras descubrir que no puede continuar la ejecución normal.

Abortada: la transacción pasa a este estado después de haber restablecido la base de datos a su

estado anterior.

Comprometida: la transacción pasa a este estado tras completarse con éxito.



Niveles de aislamiento (Isolation level)

Las transacciones especifican un nivel de aislamiento que define el grado en que se debe aislar una transacción de las modificaciones de recursos o datos realizadas por otras transacciones. Los niveles de aislamiento se describen en función de los efectos secundarios de la simultaneidad que se permiten, como las lecturas de datos sucios o las lecturas fantasmas.

Los niveles de aislamiento de transacciones controlan lo siguiente:

Nivel de aislamiento	
READ UNCOMMITTED	puede recuperar datos modificados pero no confirmados por otras transacciones (lecturas sucias - dirty reads). En este nivel se pueden producir todos los efectos secundarios de simultaneidad (lecturas sucias, lecturas no repetibles y lecturas fantasma - ej: entre dos lecturas de un mismo registro en una transacción A, otra transacción B puede modificar dicho registro), pero no hay bloqueos ni versiones de lectura, por lo que se minimiza la sobrecarga
READ COMMITTED	Permite que entre dos lecturas de un mismo registro en una transacción A, otra transacción B pueda modificar dicho registro, obteniéndose diferentes resultados de la misma lectura
REPEATABLE READ	Evita que entre dos lecturas de un mismo registro en una transacción A, otra transacción B pueda modificar dicho registro, con el efecto de que en la segunda lectura de la transacción A se obtuviera un dato diferente
SERIALIZABLE	Garantiza que una transacción recuperará exactamente los mismos datos cada vez que repita una operación de lectura (es decir, la misma sentencia SELECT con la misma cláusula WHERE devolverá el mismo número de filas, luego no se podrán insertar filas nuevas en el rango cubierto por la WHERE, etc. - se evitarán las lecturas fantasma), aunque para ello aplicará un nivel de bloqueo que puede afectar a los demás usuarios en los sistemas multiusuario.

Sintaxis para declarar el nivel de aislamiento:

```
SET TRANSACTION ISOLATION LEVEL
{
  | READ UNCOMMITTED
  | READ COMMITTED
  | REPEATABLE READ
  | SERIALIZABLE
}
[ ; ]
```

El nivel de aislamiento por defecto para MySQL es REPEATABLE READ.

Ejemplo: