



Técnico en
< DESARROLLO DE SOFTWARE >

Bases de Datos II

(CC BY-NC-ND 4.0)
International

Attribution-NonCommercial-NoDerivatives 4.0



Atribución

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



No Comercial

Usted no puede hacer uso del material con fines comerciales.



Sin obra derivada

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



Bases de Datos II

Unidad VI

Transacciones

1. Transacciones

Una transacción es una unidad lógica de procesamiento de la base que incluye una o más operaciones de acceso a la base de datos, que pueden ser de inserción, eliminación, modificación o recuperación. A continuación, mencionaremos los tipos de entradas, denominadas registros de diario, que se escriben en el diario y la acción que realiza cada una de ellas. En estas entradas, T se refiere a un identificador de transacción único que el sistema genera automáticamente y que sirve para identificar cada transacción:

Comandos: **START TRANSACTION, COMMIT y ROLLBACK,**

Por defecto, MySQL se ejecuta con el modo autocommit activado. Esto significa que en cuanto ejecute un comando que actualice (modifique) una tabla, MySQL almacena la actualización en disco.

Si se necesita desactivar el comando autocommit utilice el comando:

SET AUTOCOMMIT=0;

Debe usar COMMIT para almacenar los cambios en disco, cuándo se ejecuta es cuándo hasta ese momento todas las sentencias que le preceden tendrán efecto antes no.

Ejecutando por ejemplo una sentencia UPDATE, esta sentencia no tenga ningún efecto hasta que nosotros lo indiquemos con COMMIT. Por ejemplo;

UPDATE bd SET campo="valor" WHERE id=1;

En cambio, se utiliza ROLLBACK si se quiere ignorar los cambios hechos desde el comienzo de la transacción. Si realiza un comando ROLLBACK tras actualizar una tabla no transaccional dentro de una transacción, ocurre una advertencia.

"ER_WARNING_NOT_COMPLETE_ROLLBACK."

Los cambios en tablas transaccionales se deshacen, pero no los cambios en tablas no transaccionales. Si quiere deshabilitar el modo autocommit para una serie única de comandos, puede usar el comando START TRANSACTION que indica que se ha iniciado la ejecución de la transacción:

Ejemplo:

START TRANSACTION;

SELECT @A:=SUM(salary) FROM table1 WHERE type=1;

UPDATE table2 SET summary=@A WHERE type=1;

COMMIT;

Veremos un ejemplo completo, extraído del artículo fuente de esta publicación, donde utilizaremos START TRANSACTION (no es necesario AUTOCOMMIT en 0)

CREATE TABLE `departamentos` (

`CODIGO` INTEGER(11) NOT NULL DEFAULT "0",

```
`NOMBRE` VARCHAR(100),  
`PRESUPUESTO` INTEGER(11) DEFAULT NULL,  
PRIMARY KEY (`CODIGO`)  
)ENGINE=InnoDB
```

Ahora, insertaremos registros de la tabla departamentos externos a departamentos mediante una transacción:

```
START TRANSACTION;
```

```
SELECT @A := presupuesto FROM departamentos_externos WHERE codigo =11;  
INSERT INTO departamentos( codigo, nombre, presupuesto ) VALUES ( 11,  
„Department test“, @A );  
COMMIT;
```

Las actualizaciones a las tablas no-transaccionales se almacenan en el registro binario inmediatamente después de su ejecución. Para las tablas transaccionales como las tablas BDB o InnoDB, todas las actualizaciones (UPDATE, DELETE, o INSERT) que cambian alguna tabla son almacenadas en cache hasta que se recibe una sentencia COMMIT en el servidor. En ese momento mysqld escribe la transacción completa al registro binario antes de que se ejecute el COMMIT. Cuando el flujo de ejecución que gestiona la transacción comienza, reserva un buffer de tamaño binlog_cache_size para almacenar consultas. Si una sentencia es más grande de esto, el flujo abre un archivo temporal para almacenar la transacción. El archivo temporal se borra cuando acaba el flujo.

Sintaxis de SET TRANSACTION

SET [GLOBAL | SESSION] TRANSACTION ISOLATION LEVEL

**{ READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ |
SERIALIZABLE }**

Este comando prepara el nivel de aislamiento de transacción para la siguiente transacción, globalmente, o para la sesión actual. El comportamiento por defecto de SET TRANSACTION es poner el nivel de aislamiento para la siguiente transacción (que no ha empezado todavía). Si usa la palabra clave GLOBAL el comando pone el nivel de aislamiento de transacción por defecto globalmente para todas las transacciones creadas desde ese momento. Las conexiones existentes no se ven afectadas. Necesita el permiso SUPER para hacerlo. Usar la palabra clave SESSION determina el nivel de transacción para todas las transacciones futuras realizadas en la conexión actual.

Descargo de responsabilidad

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educativos del curso.

