Universidad Andrés Bello®
Conectar · Innovar · Liderar

FUNDAMENTOS DE BASES DE DATOS RELACIONALES

Transaccionalidad en las operaciones

Qué es una transacción y por qué son importantes

Una transacción es una unidad de trabajo que se realiza en una base de datos. Las transacciones son unidades o secuencias de trabajo realizadas en un orden lógico, yasea de forma manual por un usuario o automáticamente por algún tipo de programa de base de datos.

Una transacción es la propagación de uno o más cambios a la base de datos. Porejemplo, si está creando, actualizando o eliminando un registro de una tabla, entonces se está realizando una transacción en la tabla. Es importante controlar las transacciones para garantizar la integridad de los datos y manejar los errores de la base de datos.

En la práctica, se usa agrupar múltiples consultas y ejecutarlas todas juntas como parte de una transacción. A fin de entender la importancia de este concepto, se explicará en primer lugar la diferencia entre las tablas MyISAM e InnoDB.

Características de MyISAM

- Se establece por defecto cuando se crea una tabla, salvo que se indique lo contrario.
- Soporta transacciones.
- Realizar bloqueo de registros.
- Soporta un gran número de consultas SQL, lo que se refleja en una velocidad decarga muy rápida para nuestra web.

Como desventaja, señalamos que no realiza bloqueo de tablas, esto puede ser un problema si como se ha mencionado anteriormente hay un acceso simultáneo al mantenimiento de registros por parte de varios usuarios.

Características de InnoDB

- Bloqueo de registros. Importante para accesos múltiples al mantenimiento de tablas, es decir, ejecuciones de sentencias tipo INSERT o UPTATE, éstas ejecuciones tienen una velocidad optimizada.
- Capacidad para soportar transacciones e integridad de datos, es decir previene elalta de datos no adecuados.
- Aplica las características propias de ACID (Atomicity, Consistency, Isolation and Durability), consistentes en garantizar la integridad de las tablas.

Como desventaja, marcamos que al ser un tipo de motor que define un sistema más complejo de diseño de tablas, reduce el rendimiento en velocidad para desarrollo que requieren de un elevado número de consultas.

Qué tipo de tabla usar

Se pueden dar las siguientes situaciones:

- Un solo gestor de mantenimiento para una plataforma que requerirá muchas consultas o visitas: MyISAM
- Necesitas velocidad y mínimo consumo de recursos en servidor, espacio, RAM, etc.: MyISAM
- Varios o muchos gestores de mantenimiento: InnoDB
- Desarrollo donde se prioriza el diseño relacional de bases de datos: InnoDB

Independientemente del sistema que se elija, hay que hacer un buen diseño de la estructura y funcionalidad de la base de datos. Al mismo tiempo, la información o datos no deben almacenarse de cualquier manera. Hay que buscar el mayor aprovechamiento de los recursos que tenemos a nuestra disposición, tanto a nivel de almacenamiento como de rendimiento.

Además, hay que mantener la consistencia de la información durante todo el ciclo de vida de la base de datos, más aún si los datos que se manejan son críticos; por ejemplo, los salarios de una organización.

Los primeros factores que realizará el analista serán el análisis del sistema, que servirá de modelo, la observación de los elementos que lo componen y la descomposición en partes mucho más pequeñas.

Las tablas del tipo InnoDB están estructuradas de forma distinta que MyISAM, ya que se almacenan en un sólo archivo en lugar de tres, y sus principales características son que permite trabajar con transacciones, y definir reglas de integridad referencial.

El soporte de transacciones que provee MySQL no es algo nuevo en MySQL, ya que desde la versión 3.23 se podía hacer uso de tablas InnoDB; la única diferencia es que con la llegada de la versión 4.0 de MySQL, el soporte para este tipo de tablas es habilitado por default.

Las transacciones aportan una fiabilidad superior a las bases de datos. Si disponemos de una serie de consultas SQL que deben ejecutarse en conjunto, con el uso de transacciones podemos tener la certeza de que nunca nos quedaremos a medio camino de su ejecución. De hecho, podríamos decir que las transacciones aportan una característica de "deshacer" a las aplicaciones de bases de datos.

Para este fin, las tablas que soportan transacciones, como es el caso de InnoDB, son mucho más seguras y fáciles de recuperar si se produce algún fallo en el servidor, ya que las consultas se ejecutan o no en su totalidad. Por otra parte, las transacciones pueden hacer que las consultas tarden más tiempo en ejecutarse.

Propiedades de las transacciones: atomicidad, consistencia, aislamiento,durabilidad

Las transacciones tienen las siguientes cuatro propiedades estándar, a las que generalmente se hace referencia con el acrónimo ACID (del inglés Atomicity, Coherence,Isolation, Durability):

- **Atomicidad:** Garantiza que todas las operaciones dentro de la unidad de trabajo se completen con éxito; de lo contrario, la transacción se aborta en el punto de falla y las operaciones anteriores se revierten a su estado anterior.
- **Coherencia:** Garantiza que la base de datos cambie correctamente de estado tras una transacción confirmada con éxito.
- **Aislamiento:** Permite que las transacciones funcionen de forma independiente y transparente entre sí.
- **Durabilidad:** Asegura que el resultado o efecto de una transacción comprometida persista en caso de falla del sistema.

Confirmación de una transacción

Los comandos de control transaccional se utilizan con los comandos DML INSERT, UPDATE y DELETE únicamente. No se pueden usar al crear tablas o descartarlas porque estas operaciones se confirman automáticamente en la base de datos.

Las transacciones se pueden iniciar usando START. Estas transacciones generalmente persisten hasta que se encuentra el siguiente comando COMMIT o ROLLBACK. Una transacción también hará ROLLBACK si la base de datos se cierra o si ocurre un error.

La siguiente es la sintaxis simple para iniciar una transacción:

```
BEGIN TRANSACTION;
```

El comando COMMIT es el comando transaccional que se utiliza para guardar los cambios invocados por una transacción en la base de datos. Este comando guarda todas las transacciones en la base de datos desde el último comando COMMIT o ROLLBACK.

La sintaxis del comando COMMIT es la siguiente:

```
COMMIT;
```

Vuelta atrás de una transacción

El comando ROLLBACK es el comando transaccional que se utiliza para deshacer transacciones que aún no se han guardado en la base de datos. El comando ROLLBACK solo se puede utilizar para deshacer transacciones desde que se emitió el último comando COMMIT o ROLLBACK.

La sintaxis del comando ROLLBACK es la siguiente:

```
ROLLBACK;
```

Modo autocommit

El modo autocommit indica si los resultados de las consultas DML realizadas serán almacenadas directamente en la base de datos. Por defecto esté dato es igual a Verdadero, lo que significa que, por defecto, las transacciones se reflejarán inmediatamente.

Sin embargo, este modo se puede desactivar, a fin de poder confirmar o deshacer el resultado de una transacción. Después de deshabilitar el modo de confirmación automática, estableciendo la variable de confirmación automática en cero, los cambios en las tablas de transacciones seguras (como las de InnoDB o NDB) no se hacen permanentes de inmediato. Debe utilizar COMMIT para almacenar sus cambios en el disco o ROLLBACK para ignorar los cambios.

El ***autocommit*** es una variable de sesión y debe configurarse para cada sesión. Para

entender mejor este punto, haremos un ejemplo:

- En primer lugar, iniciamos la transacción.
- Desactiva el autocommit.
- Inserta dos registros en la tabla profesor.
- Muestra los registros existentes.
- Usa un comando para deshacer la acción.
- Muestra los registros existentes.
- Activa el autocommit.

El código antes indicado se vería de la siguiente manera:

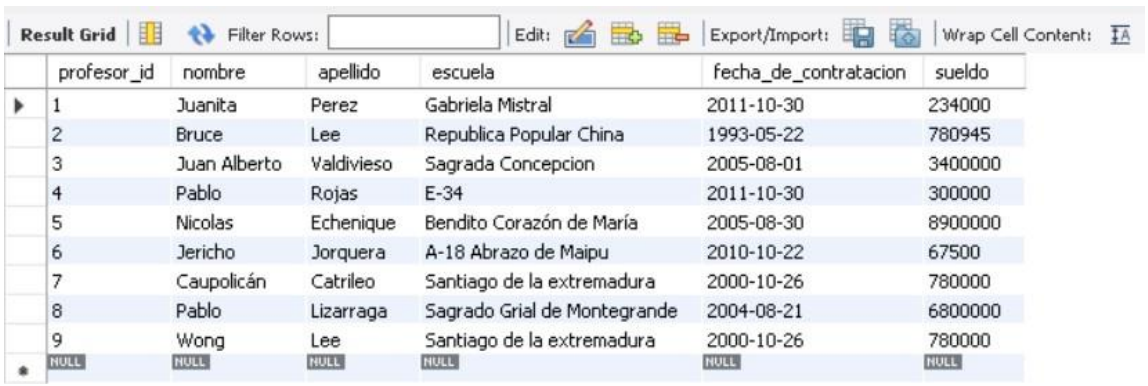
```
START TRANSACTION;
SET autocommit=0;

INSERT INTO profesor VALUES (10, 'Alfonsina', 'Araya',
'Escuela D-255', '2021-01-01', 600000);
INSERT INTO profesor VALUES (11, 'Bernardo', 'Bustos',
'Escuela Z-001', '2021-02-02', 850000);

SELECT * FROM profesor;

ROLLBACK;

SELECT * FROM profesor;

SET autocommit=1;
```

La primera consulta de selección arrojará el siguiente resultado:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

profesor_id	nombre	apellido	escuela	fecha_de_contratacion	sueldo
1	Juanita	Perez	Gabriela Mistral	2011-10-30	234000
2	Bruce	Lee	Republica Popular China	1993-05-22	780945
3	Juan Alberto	Valdivieso	Sagrada Concepcion	2005-08-01	3400000
4	Pablo	Rojas	E-34	2011-10-30	300000
5	Nicolas	Echenique	Bendito Corazón de María	2005-08-30	8900000
6	Jericho	Jorquera	A-18 Abrazo de Maipu	2010-10-22	67500
7	Caupolicán	Catrileo	Santiago de la extremadura	2000-10-26	780000
8	Pablo	Lizarraga	Sagrado Grial de Montegrande	2004-08-21	6800000
9	Wong	Lee	Santiago de la extremadura	2000-10-26	780000
10	Alfonsina	Araya	Escuela D-255	2021-01-01	600000
11	Bernardo	Bustos	Escuela Z-001	2021-02-02	850000
NULL	NULL	NULL	NULL	NULL	NULL

La segunda consulta de selección, en tanto, mostrará los registros originales, ya que al hacer ROLLBACK se deshacen los cambios.



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

profesor_id	nombre	apellido	escuela	fecha_de_contratacion	sueldo
1	Juanita	Perez	Gabriela Mistral	2011-10-30	234000
2	Bruce	Lee	Republica Popular China	1993-05-22	780945
3	Juan Alberto	Valdivieso	Sagrada Concepcion	2005-08-01	3400000
4	Pablo	Rojas	E-34	2011-10-30	300000
5	Nicolas	Echenique	Bendito Corazón de María	2005-08-30	8900000
6	Jericho	Jorquera	A-18 Abrazo de Maipu	2010-10-22	67500
7	Caupolicán	Catrileo	Santiago de la extremadura	2000-10-26	780000
8	Pablo	Lizarraga	Sagrado Grial de Montegrande	2004-08-21	6800000
9	Wong	Lee	Santiago de la extremadura	2000-10-26	780000
NULL	NULL	NULL	NULL	NULL	NULL

Anexo: Referencias

1.- Sentencias START TRANSACTION, COMMIT, y ROLLBACK

Referencia: https://dev.mysql.com/doc/refman/8.0/en/commit.html

2.- Transacciones en MySQL

Referencia: https://programacion.net/articulo/transacciones_en_mysql_242

3.- MyISAM vs InnoDB

Referencia: https://www.webreunidos.es/myisam-vs-innodb/