APUNTES ABD

Author
B.L.B

Contents

1 Control de Acceso

Definicion 1.1. Administrador: persona responsable de garantizar la seguridad e integridad de los datos.

- Seguridad: previene la revelación, alteración y destrucción de los datos. Los usuarios solo pueden realizar operaciones autorizadas.
- Integridad: asegura la exactitud y la validez de los datos (consistentes)

Es el responsable principal de la **seguridad** del SGBD. Tiene una cuenta de súper-usuario (forma de acceso protegida) y especifica qué usuarios tienen acceso a qué datos y qué operaciones pueden realizar sobre esos datos.

El acceso se controla mediante privilegios (permisos para realizar operaciones).

1.1 Gestión de privilegios

En esta sección se explorará la forma de crear usuarios y concederles privilegios.

1.1.1 Crear cuenta de usuario

```
CREATE USER <usuario> IDENTIFIED BY 'contraseña';
```

https://dev.mysql.com/doc/refman/8.0/en/create-user.html

1.1.2 Borrar cuenta de usuario

```
DROP USER <usuario>;
```

1.1.3 Conceder privilegios

```
GRANT <privilegio> ON <objeto> TO <sujeto> [WITH GRANT OPTION]
```

La cláusula "With Grang Option" permite al usuario receptor de un permiso propagarlo a otros usuarios.

1.1.4 Retirar privilegios

```
REVOKE [GRANT OPTION FOR] <privilegio> ON <objeto> FROM <sujeto>
```

1.1.5 Tipos de objeto

```
*.* (Global)
<BD>.*
<BD>.<tabla>
<BD>.<tabla>(columna)
```

Mostrar permisos otorgados para nuestro usuario:

```
SHOW GRANTS;
```

Mostrar permisos para otro usuario:

```
SHOW GRANTS FOR <usuario>;
```

1.1.6 Ejercicio 1

Crear BD llamada “ej1DB”

```
CREATE DATABASE ej1DB;
```

Crear usuario llamado “ej1user” y darle permisos para crear tablas en ej1DB y leer, insertar y borrar filas en las tablas de ej1DB

```
CREATE USER ej1user IDENTIFIED BY '1234';
GRANT CREATE, SELECT, INSERT, DELETE ON ej1DB.* TO ej1user;
```

Verificar los permisos asignados

```
SHOW GRANTS;
```

1.2 Roles

Un rol es una colección de privilegios con un nombre.

1.2.1 Crear rol

```
CREATE ROLE <rol>;
```

1.2.2 Asignar privilegios a rol

```
GRANT <privilegio> ON <objeto> TO <rol>;
```

1.2.3 Asignar rol a usuario

```
GRANT <rol> TO <usuario>;
```

1.2.4 Activar y mostrar rol

```
SET ROLE <rol>;
SELECT current_role();
```

1.2.5 Definir rol por defecto

```
SET DEFAULT ROLE <rol> TO <usuario>;
```

1.2.6 Mostrar privilegios de rol

```
SHOW GRANTS FOR <usuario> USING <rol>;
```

1.2.7 Retirar privilegios

```
REVOKE <privilegio> ON <objeto> FROM <rol>;
```

1.2.8 Eliminar rol

```
DROP <rol>;
```

1.2.9 Ejercicio 2

Crear una BD "ej2DB" y usuario "ej2user"

```
CREATE DATABASE ej2DB;
CREATE USER ej2user IDENTIFIED BY '1234';
```

Crear un rol "ej2rol" con permisos para crear tablas e insertar y leer filas de las tablas.

```
CREATE ROLE ej2rol;
GRANT CREATE, INSERT, SELECT ON ej2DB.* TO ej2rol;
```

Asignar el rol al usuario

```
GRANT ej2rol TO ej2user;
```

2 Recuperación (06-02-2024)

La SGBD debe asegurar la integridad y la seguridad de los datos (LOPD) ante los riesgos de que estos datos se pierdan. Dos situaciones:

- A nivel tabla/BD
- **A nivel de transacción**

Se hacen **transacciones** para agrupar varias operaciones, si algo va mal en alguna de las operaciones la transacción se interrumpe y no hay cambios en la BD.

> **Definicion 2.1. Transacción:** Unidad lógica de procesamiento, de integridad y de recuperación.

```
START TRANSACTION;
\\OPERACIONES
COMMIT;(éxito) O  ROLLBACK;(fracaso)
```

Por defecto mySQL funciona en AutoCommit: crea una transacción por cada operación. Se puede desactivar para gestionar las transacciones de forma manual.

```
SET AUTOCOMMIT=0;
```

Para garantizar las validez de datos frente a fallos, los SGBD cumplen las propiedades ACID:

- Atomicity: unidad lógica de procesamiento, no puede ejecutarse a medias.
- Consistency: unidad lógica de integridad
- Isolation: ejecución de una transacción no afecta a otra ejecución.
- Durability: los cambios de una transacción confirmados deben persistir.

Durante la ejecución de una base de datos, los cambios y cálculos ocurren en la memoria RAM. Una vez deban reflejarse, se envían al disco duro y se organizan en distintos ficheros. Para que la SGBD sea lo más eficiente posible, se trata de acceder lo menos posible al disco duro. Existen 3 tipos de fallos posibles:

- Fallos físicos
- Fallos de software
- Fallos al ejecutar la transacción

La SGBD debe asegurar que se cumplan las propiedades ACID a pesar de los fallos. Las SGBD disponen de un diario en el que se escriben las operaciones realizadas por las transacciones. Se escribe mediante "write ahead logging". Todo comando que pasa por la RAM, se escribe en un diario (fichero) del disco antes de realizar los cambios en la BD. Este diario es la base del proceso de recuperación y sirve para monitorizar la ejecución de las transacciones.

```
START_TRANSACTION, <T>, <time>
READ, <T>, <X>, <V>, <time>
WRITE, <T>, <X>, <oldV>, <newV>, <time>
COMMIT, <T>, <time>
ROLLBACK, <T>, <time>
```

En caso de fallo en medio de una transacción, el sistema de recuperación revisa el diario y aplica las operaciones REDO(T) o UNDO(T). En mySQL el diario se llama **Redo Log**. Por defecto se encuentra activado. Se puede repartir entre varios ficheros y se puede ver cuales se hallan activos. (Escrito en binario).

2.1 Proceso de recuperación

Tras el commit o el failed hay un paso más: terminated. Llega ahí tras pasar un punto de confirmación. En este tema se considerará un único diario. Cuando una transacción T realiza COMMIT: todas las operaciones de la transacción se ejecutaron con éxito, el efecto se ha anotado en el diario y T ha llegado a un punto de confirmación. Cuando una transacción T realiza ROLLBACK: las operaciones se han anotado en el diario, pero sus operaciones no deben escribirse en la BD. Si sucede un fallo durante una transacción, T no alcanza un punto de confirmación. El diario contiene alguna de sus operaciones, pero no está el COMMIT. El proceso de recuperación aplica UNDO(T): deshace las operaciones en orden inverso. Si sucede un fallo cuando una transacción ya ha sido confirmada, se debe rehacer. En este caso debe aplicarse REDO(T): rehace las operaciones en el orden original.
Frecuencia de la actualización del diario:

- Inmediata: por cada operación, la RAM envía dicha operación al diario. Mayor fiabilidad, menor rendimiento (más escrituras en disco).
- Diferida: las operaciones se envían al diario por bloques. Mayor rendimiento (menos escrituras), mejor jerarquía de memoria, mayor riesgo.

Cada cierto tiempo, el SGBD realiza el **Checkpoint**:
En *escritura diferida* de diario, el bloque de RAM se envía al diario aunque no haya alcanzado su tamaño total. Revisa el diario, y aquellas con COMMIT (finalizadas) las escribe en la BD. Crea una lista con las transacciones activas.

```
CHECKPOINT, <time>
```

Estos puntos permiten recorrer el diario para ignorar las confirmadas anteriores al checkpoint. En *escritura diferida*, hay 3 operaciones que fuerzan las escritura de los bloques: CHECKPOINT, ROLLBACK, COMMIT.

2.2 Actualización de BD

Hay dos maneras de gestionarlo: Inmmediata y diferida.
Actualización diferida: después del commit. **Algoritmo no-deshacer/rehacer**
Pueden suceder 2 tipos de fallo:

- Antes de realizar el COMMIT: no hay que deshacer nada. No ocurre nada.
- Después de ralizar el COMMIT: es necesario rehacer las operaciones.

Actualización **inmediata**: **Algoritmo deshacer/rehacer**

- Antes de realizar commit: es necesario deshacer las operaciones
- Después de realizar commit: es necesario rehacer las operaciones

```
ALGORITMO:
1) Crear listas vacías ACTIVAS y CONFIRMADAS
2) Llenar listas con transacciones activas previas al checkpoint.
3) Añadir transacciones que comienzan después del checkpoint.
     3.1) Buscar posibles confirmaciones de las transacciones
         -COMMIT: Mover a confirmados
         -ROLLBACK: Borrar de activas
         ¿BD DIFERIDA?
           No: deshacer operaciones de transacciones activas
           Sí: rehacer WRITEs de transacciones confirmadas y reiniciar transacciones
           activas.
```

2.2.1 Ejercicio 3: (08/02/2024)

```
***** DATOS *****
DIARIO Nº 2
Instante de fallo: 405
Diario: inmediato
Actualización de diario: Diferida
Bloque: 3
***** RESOLUCIÓN *****
Mirar último comando que ha llegado al diario:
[COMMIT, T1, 390]
1) ACT ={} CONF={}
2)Mirar transacciones previas al último checkpoint:
ACT={T6,T3,T4,T2,T7,T8,T1,T5}
Eliminar las confirmadas:
```

```
ACT={T6,T2,T8,T1,T5}
3) Mismas activas.
3.1) Buscar confirmaciones:
ACT={T2,T8}
CONF={T5,T6,T1}
4) Operaciones: actualización inmediata. BD diferida? no:
   DESHACER (orden inverso): {T2, T8}
    WRITE: F:84; X:77; W:30; D:171
   REHACER (mismo orden): {T5, T6, T1}
    WRITE: Q:67; A:6; R:14; B:190; S:99; C:105; N:112; O:187; P:195
   REINICIAR ACTIVAS:
    T2, T8
```

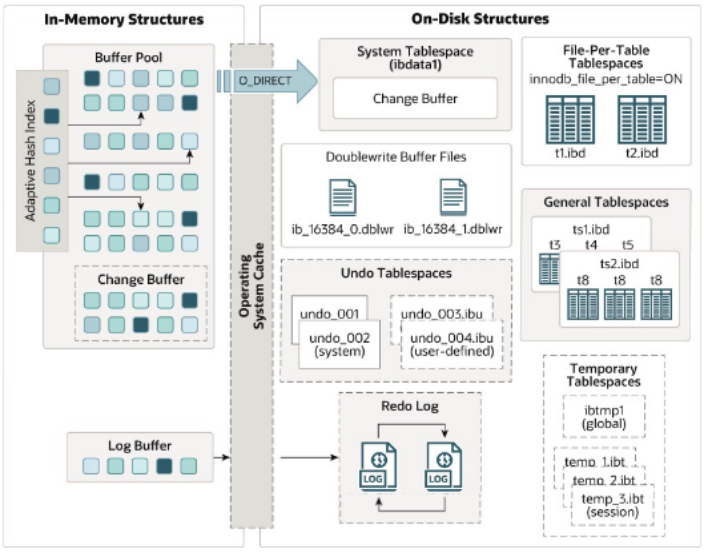
2.3 Estructura de InnoDB

Motor de almacenamiento de MySQL 8.0: InnoDB. Soporta propiedades ACID y control de concurrencia multi-usuario.
Mecanismos importantes:

- Flujo de operación de lectura: cliente -¿ RAM y CPU. Consulta una tabla llamada **Buffer Pool**, que guarda las consultas más habituales en la RAM, de modo que no tiene que acceder al disco todas las veces que se haga esa petición. Si no se encuetra allí, se accede al disco, **General Tablespaces**.

- Flujo de operación de escritura: cliente -¿ RAM. Se registran los cambios en **Log Buffer**, y estos pasan al disco al **Redo Log** y estos pasarán a **General Tablespaces**.

  Estructuras en memoria:



- *Buffer Pool*: caché de datos. Organizado con LRU. 128 MB. Se puede calcular su eficiencia con las variables de estado buffer.pool.reads y buffer.pool.read.requests. Para

evitar problemas de rendimiento, al reiniciar MySQL se guarda un porcentaje del Buffer Pool en el disco, que cuando MySQL se inicia pasan a la RAM.

- *Log Buffer*: diario de la RAM que aún no se ha pasado a disco. Contiene los datos que deben escribirse en el Redo Log. 16 MB. La gestión de copia de datos se hace mediante frecuencia.

Los ajustes de estos parámetros pueden tener comportamientos no esperados en la DB. A mayor % de Buffer Pool a disco, mayor rendimento de consultas desde el comienzo, pero más tiempo de carga de MySQL.
En MySQL 8.0 hay una variables de sistema: innodb.dedicated.server. Sólo se debe activar en entornos donde mysql sea la única aplicación, puesto que gestiona la RAM automáticamente y puede afectar a otros procesos. Afecta a: buffer.pool.size, redo.log.capacity y a flush.method.

2.4 Examen:

- Ejercicios de cada tema (teoría)
- Ejercicios prácticos