

Introducción al lenguaje de consulta estructurado (SQL)

Structured Query Language

Transacciones

Transacción

- Es un conjunto ordenado de operaciones que se ejecutan de forma atómica (de forma unitaria), o sea, si se ejecuta la transacción pueden suceder 2 cosas:
 - Se ejecutan todas las operaciones
 - No se ejecuta ninguna

Una transacción nunca ejecuta una cantidad parcial de las operaciones

Motivaciones

- Acceso **concurrente** (simultáneo) a la base de datos, pueden existir múltiples usuarios o clientes (programas externos) accediendo y realizando operaciones en forma concurrente.
- **Fallas** en los sistemas (fallas eléctricas, interrupción de los procesos del motor de la base de datos que están en ejecución)

Problemas de la concurrencia

- Inconsistencia de los valores de un campo
- Inconsistencia de las Tuplas (Registros) de una tabla
- Otros, por ejemplo, inconsistencia para operaciones que se realizan en varias tablas

Inconsistencias en los valores

- Mecanismo que usa el motor de bases de datos para acceder y modificar un valor
 - **GET** (Lee el valor)
 - **MODIFY** (lo modifica en memoria)
 - **PUT** (Guarda el valor modificado)

Ejemplo de inconsistencias en los valores

- Sentencia 1:

- **UPDATE** Producto **SET** precio = precio + 100
WHERE id = 4;

- Sentencia 2:

- **UPDATE** Producto **SET** precio = precio + 200
WHERE id = 4;

Si las 2 sentencias se ejecutan en forma simultánea por 2 usuarios diferentes (el usuario 1 ejecuta la sentencia 1 y el usuario 2 ejecuta la sentencia 2), o 2 sesiones diferentes usando el mismo usuario, puede ocurrir algo inesperado

Ejemplo de inconsistencias en los valores

- Si el valor del precio era 800 entonces al ejecutar ambas sentencias de forma concurrente los posibles resultados son:
(A) Que el valor nuevo de precio sea 900:

S1:GET

S1:MODIFY

S2:GET

S2:MODIFY

S2:PUT

S1:PUT

Ejemplo de inconsistencias en los valores

- Si el valor del precio era 800 entonces al ejecutar ambas sentencias de forma concurrente los posibles resultados son:
(B) Que el valor nuevo de precio sea 1000:

S1:GET (S1)

S2:GET
S2:MODIFY

S1:MODIFY
S1:PUT

S2:PUT

Ejemplo de inconsistencias en los valores

- Si el valor del precio era 800 entonces al ejecutar ambas sentencias de forma concurrente los posibles resultados son:
(C) Que el valor nuevo de precio sea 1100:

S1:GET

S1:MODIFY

S1:PUT

S2:GET

S2:MODIFY

S2:PUT

Inconsistencias en las tuplas

- Mecanismo que usa el motor de bases de datos para acceder y modificar una tupla (un registro)
 - **GET** (Lee la tupla)
 - **MODIFY** (modifica algunos o todos los valores de la tupla en memoria)
 - **PUT** (Guarda la tupla modificada)

Ejemplo de inconsistencias en las tuplas

- Sentencia 3:

- **INSERT INTO** Empleado **SET** idJefe = 2
WHERE id = 5;

- Sentencia 4:

- **INSERT INTO** Empleado **SET** idSucursal = 3
WHERE id = 5;

Nuevamente, si las 2 sentencias se ejecutan en forma simultánea por 2 usuarios diferentes (el usuario 1 ejecuta la sentencia 3 y el usuario 2 ejecuta la sentencia 4), o 2 sesiones diferentes usando el mismo usuario, puede ocurrir algo inesperado

Ejemplo de inconsistencias en las tuplas

- Si la tupla original contenía los siguientes valores:
 - `(id = 5, idJefe = null, idSucursal = null)`
- Al ejecutar ambas sentencias (3 y 4) de forma concurrente los posibles resultados son:

(A) Que la tupla sea:

– `(id = 5, idJefe = 2, idSucursal = null)`

S3:GET Tupla

S4:GET Tupla

S4:MODIFY Tupla

S4:PUT Tupla

S3:MODIFY Tupla

S3:PUT Tupla

Ejemplo de inconsistencias en las tuplas

- Si la tupla original contenía los siguientes valores:
 - `(id = 5, idJefe = null, idSucursal = null)`
- Al ejecutar ambas sentencias (3 y 4) de forma concurrente los posibles resultados son:

(B) Que la tupla sea:

- `(id = 5, idJefe = null, idSucursal = 3)`

S3:GET Tupla

S4:GET Tupla

S3:MODIFY Tupla

S4:MODIFY Tupla

S3:PUT Tupla

S4:PUT Tupla

Ejemplo de inconsistencias en las tuplas

- Si la tupla original contenía los siguientes valores:
 - `(id = 5, idJefe = null, idSucursal = null)`
- Al ejecutar ambas sentencias (3 y 4) de forma concurrente los posibles resultados son:

(C) Que la tupla sea:

– `(id = 5, idJefe = 2, idSucursal = 3)`

S3:GET Tupla

S3:MODIFY Tupla

S3:PUT Tupla

S4:GET Tupla

S4:MODIFY Tupla

S4:PUT Tupla

Solución usando transacciones

Script 1:

```
START TRANSACTION;  
INSERT INTO Empleado SET idJefe = 2 WHERE id = 5;  
COMMIT;
```

Script 2:

```
START TRANSACTION;  
INSERT INTO Empleado SET idSucursal = 3 WHERE id = 5;  
COMMIT;
```

NOTA: También soluciona los problemas de fallas (eléctricas, procesos del motor de bases de datos que son detenidos por el sistema operativo, etc)

Propiedades de una Transacción (ACID)

- Atomicidad (Atomicity)
 - Asegura que el conjunto de operaciones de la transacción se ejecuten de forma unitaria
- Consistencia (Consistency)
 - Asegura que las restricciones aplicadas a la base de datos no se violen
- Aislamiento (Isolation)
 - Asegura que una transacción no puede afectar a la otra
- Permanencia (Durability)
 - Asegura que si se ejecuta la transacción esta persistirá y no se podrá deshacer aunque falle el sistema

FIN