

CODIGO PRESENTACION - NIVELES DE AISLAMIENTO Y PROBLEMAS DE LECTURA

NIVELES DE AISLAMIENTO

Lecturas no Comprometidas:

Permite leer datos no confirmados (sin hacer commit)

Permite lecturas sucias, no repetibles y fantasmas

```
-- Configurar nivel de aislamiento
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

-- Verificar nivel actual
SELECT @@transaction_isolation;

-- Ejemplo práctico:
-- TERMINAL 1:
START TRANSACTION;
UPDATE cuentas SET saldo = 2000 WHERE id = 1;
-- NO hacer COMMIT todavía

-- TERMINAL 2:
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
START TRANSACTION;
SELECT * FROM cuentas WHERE id = 1; -- Ve el cambio no confirmado
COMMIT;

-- TERMINAL 1:
ROLLBACK; -- Deshace el cambio

-- TERMINAL 2 (nueva consulta):
SELECT * FROM cuentas WHERE id = 1; -- Ahora ve 1000 otra vez
```

Lecturas repetibles

Solo lee datos confirmados (Commit)

Permite lecturas no repetibles y fantasmas

```

-- Configurar nivel
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;

-- Ejemplo práctico:
-- TERMINAL 1:
START TRANSACTION;
UPDATE cuentas SET saldo = 2000 WHERE id = 1;
-- NO hacer COMMIT

-- TERMINAL 2:
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
SELECT * FROM cuentas WHERE id = 1; -- Ve el valor original (1000)

-- TERMINAL 1:
COMMIT; -- Confirma el cambio

-- TERMINAL 2:
SELECT * FROM cuentas WHERE id = 1; -- Ahora ve el nuevo valor (2000)
COMMIT;

```

Lecturas Comprometidas

Evita lecturas sucias y no repetibles

Permite lecturas fantasma

Es el nivel por defecto de mysql

```
-- Configurar nivel
SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ;

-- Ejemplo práctico:
-- TERMINAL 1:
START TRANSACTION;
SELECT * FROM cuentas WHERE id = 1; -- Ve saldo actual

-- TERMINAL 2:
START TRANSACTION;
UPDATE cuentas SET saldo = 3000 WHERE id = 1;
COMMIT;

-- TERMINAL 1:
SELECT * FROM cuentas WHERE id = 1; -- Sigue viendo el valor origi
COMMIT;

-- Nueva consulta fuera de transacción:
SELECT * FROM cuentas WHERE id = 1; -- Ahora ve el valor actualiza
```

Serializable

Es el máximo nivel de aislamiento

Menor rendimiento debido a bloqueos

```
-- Configurar nivel
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;

-- Ejemplo práctico:
-- TERMINAL 1:
START TRANSACTION;
SELECT * FROM cuentas WHERE saldo > 500;

-- TERMINAL 2:
START TRANSACTION;
INSERT INTO cuentas (nombre, saldo) VALUES ('Ana', 600); -- Se
-- Esta operación esperará hasta que TERMINAL 1 termine

-- TERMINAL 1:
COMMIT; -- Libera el bloqueo

-- TERMINAL 2:
-- Ahora se ejecuta la inserción
COMMIT;
```

PROBLEMAS DE LECTURA

Lecturas Sucias

Lee datos modificados por otra transacción que aún no ha confirmado los cambios.

```
-- Ejemplo de lectura sucia (solo ocurre en READ UNCOMMITTED)

-- TERMINAL 1:
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
START TRANSACTION;
UPDATE cuentas SET saldo = 5000 WHERE id = 1;
-- NO hacer COMMIT

-- TERMINAL 2:
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
START TRANSACTION;
SELECT saldo FROM cuentas WHERE id = 1; -- Lee 5000 (dato sucio)
COMMIT;

-- TERMINAL 1:
ROLLBACK; -- El cambio nunca se confirma

-- Verificación:
SELECT saldo FROM cuentas WHERE id = 1; -- Vuelve al valor origi
```

Lecturas Fantasmas

Obtiene diferentes resultados al ejecutar la misma consulta dentro de una transacción debido a inserciones/eliminaciones de otras transacciones.

Cambia: La cantidad de registros que cumplen una condición

```
-- Ejemplo de lectura fantasma

-- TERMINAL 1:
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
SELECT COUNT(*) FROM cuentas WHERE saldo > 500; -- Resultado:

-- TERMINAL 2:
START TRANSACTION;
INSERT INTO cuentas (nombre, saldo) VALUES ('Carlos', 700);
COMMIT;

-- TERMINAL 1:
SELECT COUNT(*) FROM cuentas WHERE saldo > 500; -- Resultado:
COMMIT;
```

Lecturas no repetibles

Obtiene diferentes valores al leer el mismo registro dentro de una transacción.

Cambia los valores de registros que ya existen

```
-- Ejemplo de lectura no repetible
```

```
-- TERMINAL 1:
```

```
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
START TRANSACTION;
```

```
SELECT saldo FROM cuentas WHERE id = 1; -- Resultado: 1000
```

```
-- TERMINAL 2:
```

```
START TRANSACTION;
```

```
UPDATE cuentas SET saldo = 1200 WHERE id = 1;
```

```
COMMIT;
```

```
-- TERMINAL 1:
```

```
SELECT saldo FROM cuentas WHERE id = 1; -- Resultado: 1200 (dirty read)  
COMMIT;
```