

# **Transacciones y Bloqueos en Bases de Datos**

Conceptos clave sobre  
transacciones, bloqueos y  
concurrency

# Modelo ACID en Transacciones

1. Atomicidad: Todo o nada.
2. Consistencia: Mantiene la integridad.
3. Aislamiento: Evita interferencias.
4. Durabilidad: Cambios permanentes.

# Estados de una Transacción

1. Activa: En ejecución.
2. Parcialmente confirmada:  
Operaciones completadas, sin COMMIT.
3. Fallida: Error detectado.
4. Abortada: Se revierte (ROLLBACK).
5. Confirmada: Se almacena permanentemente.

# Ejemplo de Transacción en SQL

```
START TRANSACTION;
```

```
UPDATE cuentas SET saldo = saldo -  
100 WHERE id = 1;
```

```
UPDATE cuentas SET saldo = saldo +  
100 WHERE id = 2;
```




```
COMMIT; -- Confirma los cambios
```

```
ROLLBACK; -- Revierte los cambios  
en caso de error
```

# **Tipos de Bloqueos**

1. Bloqueo Compartido (S): Permite lectura concurrente.
2. Bloqueo Exclusivo (X): Solo una transacción puede modificar.

# Esquema de Compatibilidad de Bloqueos

- Compartido (S) con Compartido (S):  
 Permitido
- Compartido (S) con Exclusivo (X):  
 No permitido
- Exclusivo (X) con Exclusivo (X):  
 No permitido

# Problemas de Concurrency

1. Lectura sucia: Se previene con bloqueos exclusivos.
2. Lectura no repetible: Se evita con bloqueos de fila.
3. Lectura fantasma: Se previene con bloqueos de tabla o índice.

# Niveles de Aislamiento

1. Read Uncommitted: Lecturas sucias permitidas.
2. Read Committed: Bloquea lecturas sucias.
3. Repeatable Read: Bloquea lecturas no repetibles.
4. Serializable: Máxima seguridad, menor rendimiento.



# Ejemplo de Bloqueos en SQL

```
BEGIN TRANSACTION;  
SELECT * FROM cuentas WHERE id = 1  
FOR UPDATE;  
UPDATE cuentas SET saldo = saldo -  
100 WHERE id = 1;  
COMMIT;
```

# Conclusión

Las transacciones y bloqueos garantizan la consistencia y confiabilidad de los datos en entornos multiusuario.