

# Bases de datos relacionales. Aspectos claves.

**Ing. Luciano Straccia**

# TEMARIO



- ▶ Transacciones
- ▶ Constraints

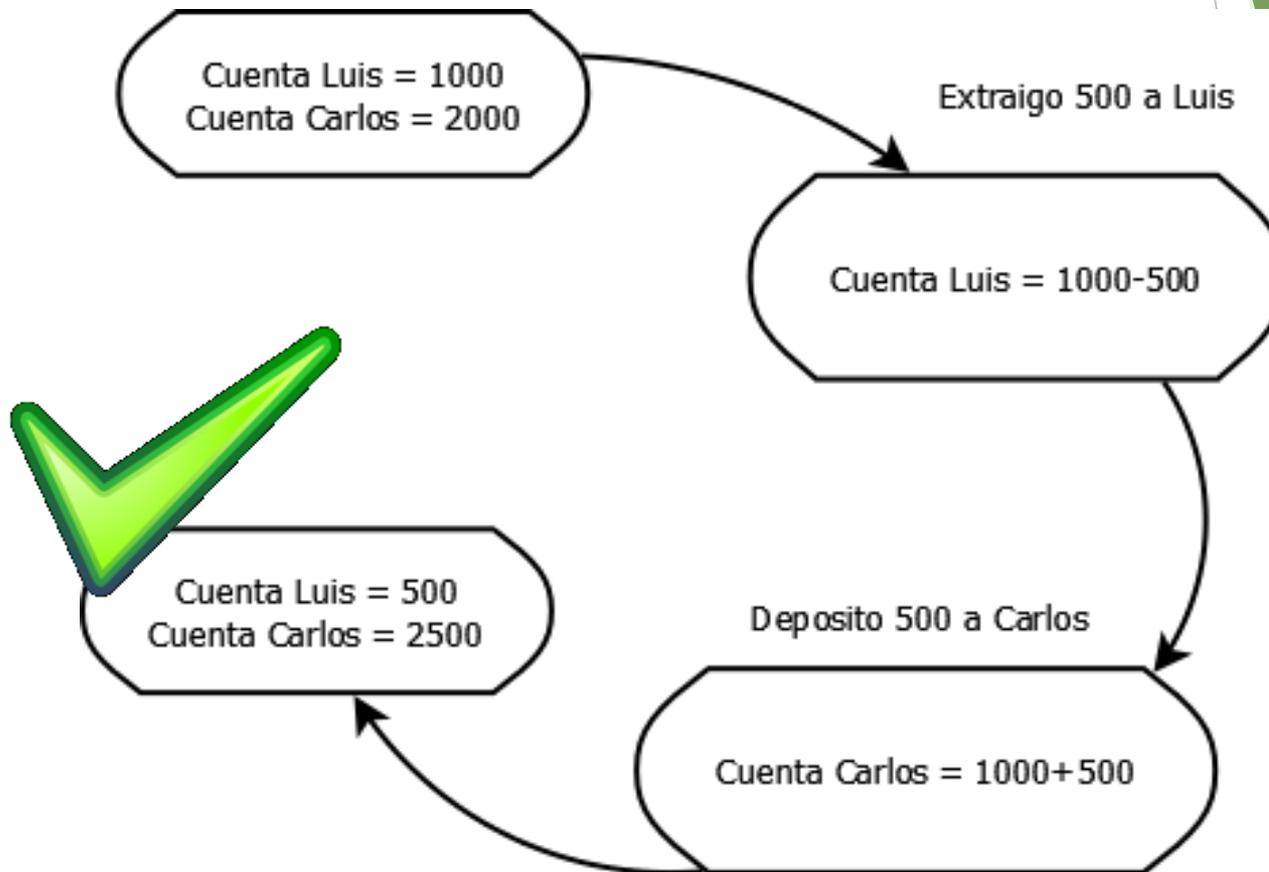
transacciones

# Qué es una transacción

- Conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.

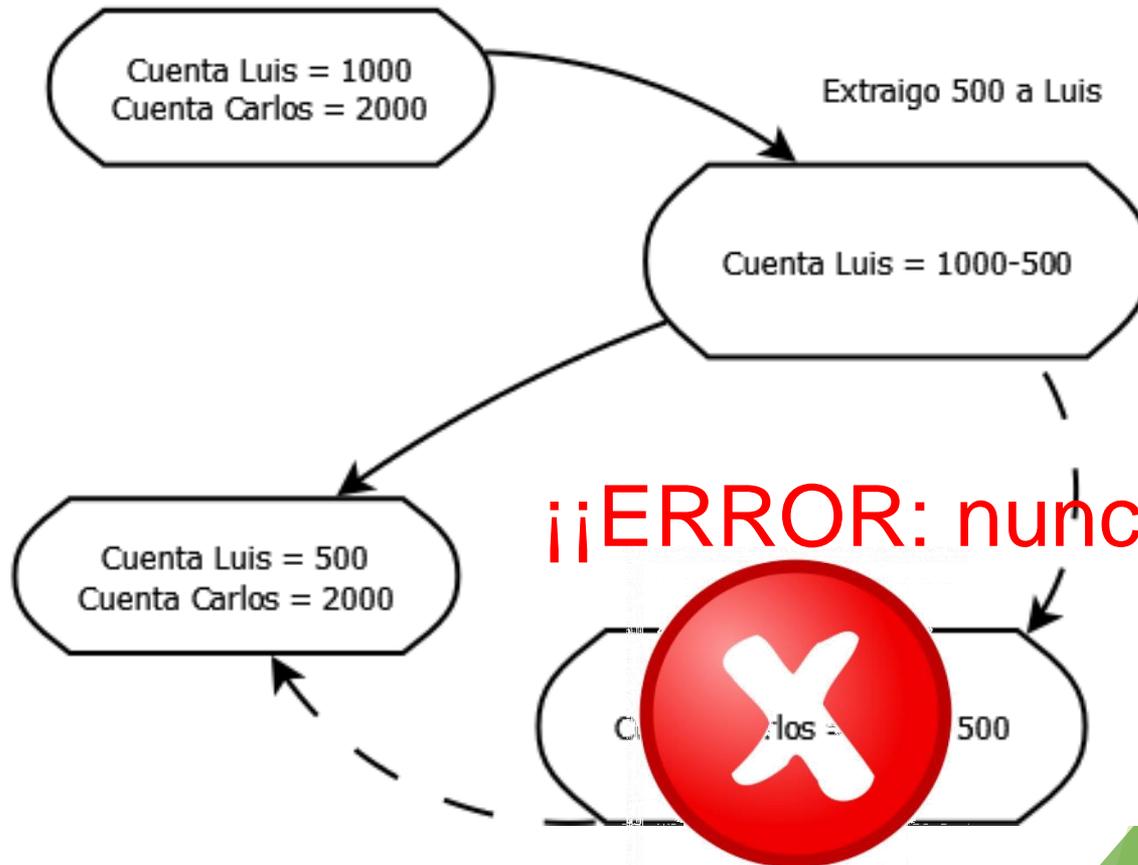
# Porqué usar transacciones

- ▶ Sin transacciones, con resultado exitoso:



# Porqué usar transacciones

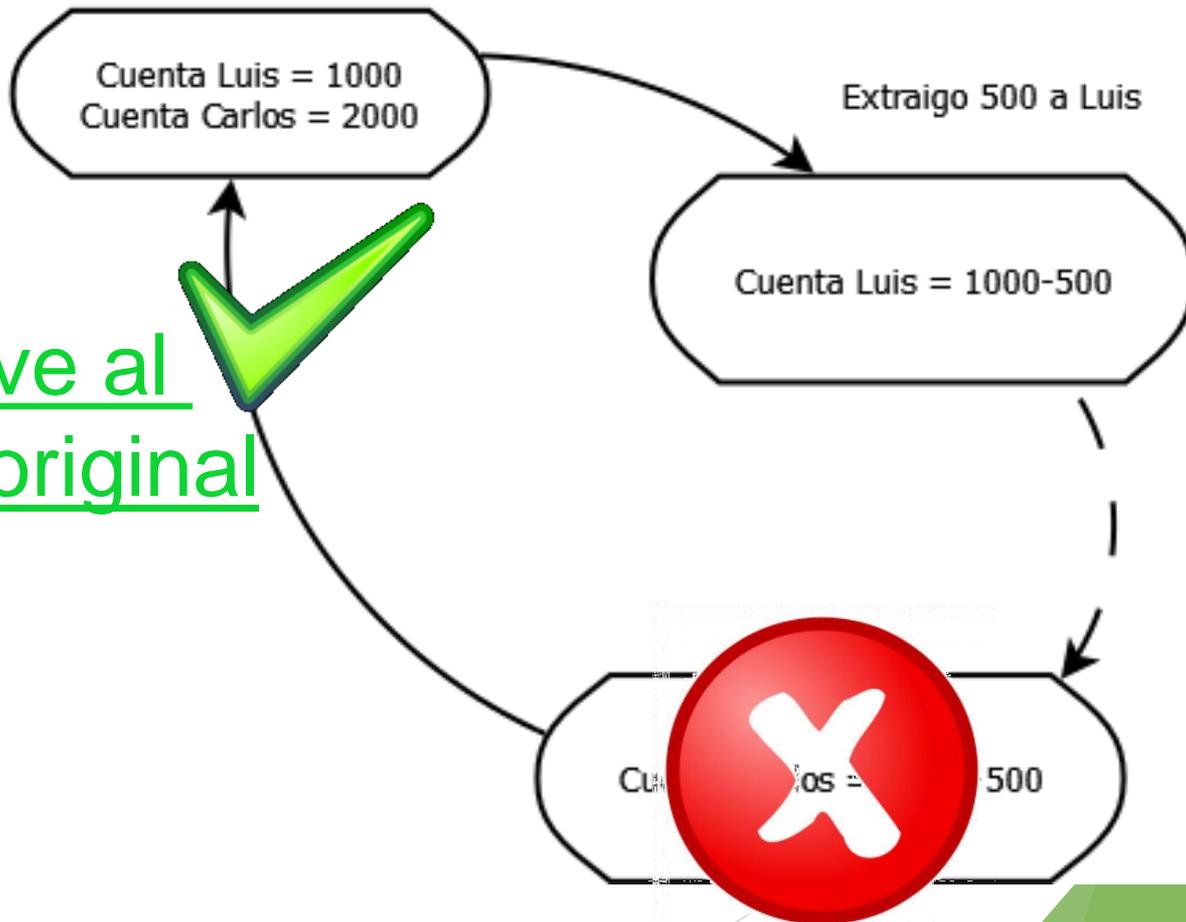
- Sin transacciones, ERROR inesperado:



# Porqué usar transacciones

- Con transacciones, ERROR inesperado:

Se vuelve al estado original



# Propiedades ACID

- ▶ **Atomicidad:** cada transacción es ejecución "todo o nada".
- ▶ **Consistencia:** todos los datos tienen que ser válidos
- ▶ **Aislamiento:** cada transacción es independiente de otra.
- ▶ **Durabilidad:** que una vez que se confirmó una transacción (commit), los datos quedaran persistidos

# Porqué puede fallar una transacción

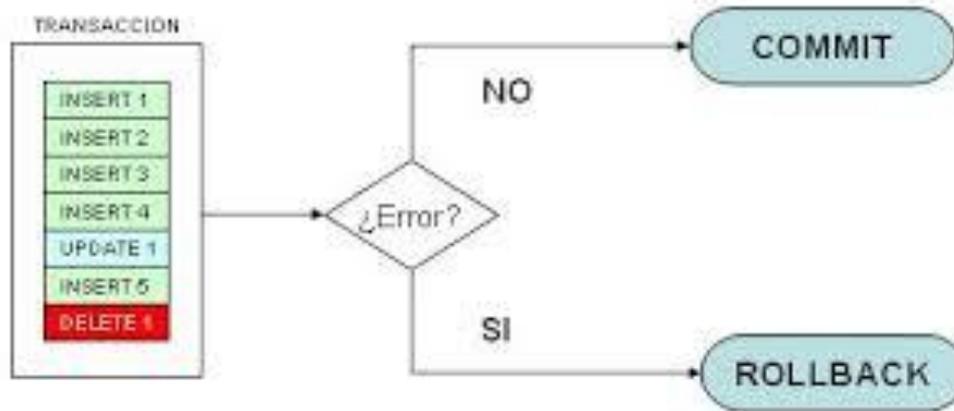
Una transacción puede no llegar a su término debido a muchas razones:

- ▶ situación excepcional detectada que hace que el programa no pueda continuar
- ▶ falla del programa
- ▶ falla del software de BD
- ▶ falla del Sistema Operativo
- ▶ falla del hardware
- ▶ falla de energía eléctrica
- ▶ control de concurrencia ha detectado un conflicto
- ▶ control de concurrencia ha detectado un deadlock o bloqueo infinito

# Transacciones

- **BEGIN TRAN:** Especifica que va a empezar una transacción.
- **COMMIT TRAN:** Le indica al motor que puede considerar la transacción completada con éxito.
- **ROLLBACK TRAN:** Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

# Transacciones



# Ejemplo

## Ejemplo en Oracle

```
DECLARE
    importe NUMBER;
    ctaOrigen VARCHAR2(23);
    ctaDestino VARCHAR2(23);
BEGIN
    importe := 100;
    ctaOrigen := '2530 10 2000 1234567890';
    ctaDestino := '2532 10 2010 0987654321';
    UPDATE CUENTAS SET SALDO = SALDO - importe WHERE CUENTA = ctaOrigen;
    UPDATE CUENTAS SET SALDO = SALDO + importe WHERE CUENTA = ctaDestino;
    INSERT INTO MOVIMIENTOS VALUES (ctaOrigen, ctaDestino, importe*(-1), SYSDATE);
    INSERT INTO MOVIMIENTOS VALUES (ctaDestino, ctaOrigen, importe, SYSDATE);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('Error en la transaccion: ' || SQLERRM);
        dbms_output.put_line('Se deshacen las modificaciones);
        ROLLBACK;
END;
```

# Constraints

# Qué es una constraints / restricción

- Una restricción es una condición que obliga el cumplimiento de ciertas condiciones en la base de datos.

# Cuáles son las constraints

- **NOT NULL:** Indica si una columna puede tener un valor nulo
- **UNIQUE:** Asegura que una columna tenga un valor único
- **PRIMARY KEY**
- **FOREIGN KEY**
- **CHECK:** Indica si una columna posee una condición específica
- **DEFAULT:** Especifica un valor default para la columna

# Sintaxis de una constraints

```
CREATE TABLE table_name  
(  
  column_name1 data_type(size) constraint_name,  
  column_name2 data_type(size) constraint_name,  
  column_name3 data_type(size) constraint_name,  
  ....  
);
```