PL/SQL Transacciones

Antonio Espín Herranz

Transacciones

- Transacciones.
- Puntos de control.
- Transacciones autónomas.

Transacciones

- Una transacción es un conjunto de instrucciones DML (Insert, Update, Delete) que se ejecutan entre dos comandos Connect, Commit y Rollback.
- La transacción nos asegura que un conjunto de instrucciones SQL se van a ejecutar como una sola instrucción o no se van a ejecutar.
- La transacción termina con una instrucción:
  - **Commit**: Confirmando los cambios.
  - **Rollback**: Deshaciendo los cambios.

Transacciones

- Una vez que cerramos una transacción (con Commit o Rollback) la siguiente instrucción DML iniciará otra transacción nueva.
- Oracle mantiene la **coherencia con los datos** y los usuarios. Un usuario no verá los cambios confirmados hasta que el otro usuario no haga el commit.
- Las instrucciones DDL no forman parte de transacciones.
- La ejecución de un comando Create, Alter o Drop confirmará la transacción que esté iniciada.

Ejemplo: Transferencia Bancaria

- **Set serveroutput on;**
- **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 (CUENTA_ORIGEN, CUENTA_DESTINO,IMPORTE, FECHA_MOVIMIENTO)
  VALUES (ctaOrigen, ctaDestino, importe*(-1), SYSDATE);
  INSERT INTO MOVIMIENTOS (CUENTA_ORIGEN, CUENTA_DESTINO,IMPORTE, FECHA_MOVIMIENTO)
  VALUES (ctaDestino,ctaOrigen, importe, SYSDATE);

  COMMIT; -- Confirma la transacción.

- **EXCEPTION**
  WHEN OTHERS THEN
  dbms_output.put_line('Error en la transaccion:'||**SQLERRM**);
  dbms_output.put_line('Se deshacen las modificaciones);
  ROLLBACK; -- Si se produce cualquier error se deshacen todos los cambios.
  END;

Puntos de Control

- Rollback nos ofrece la posibilidad de deshacer cambios hasta un determinado punto.
- Para ello tenemos que marcar un punto de control.
- Después en un momento dado podemos deshacer los cambios hasta ese punto de control.
- Sintaxis:
  - **Rollback [to punto_grabación]**

Esquema Puntos de Control

SQL>Savepoint s1;

SQL>Select * from personas; (por ej. lista 5)

SQL>**Delete** from personas where edad > 30;

SQL>Select * from personas; (lista 3)

SQL>Savepoint s2;

SQL>**insert into** Personas …

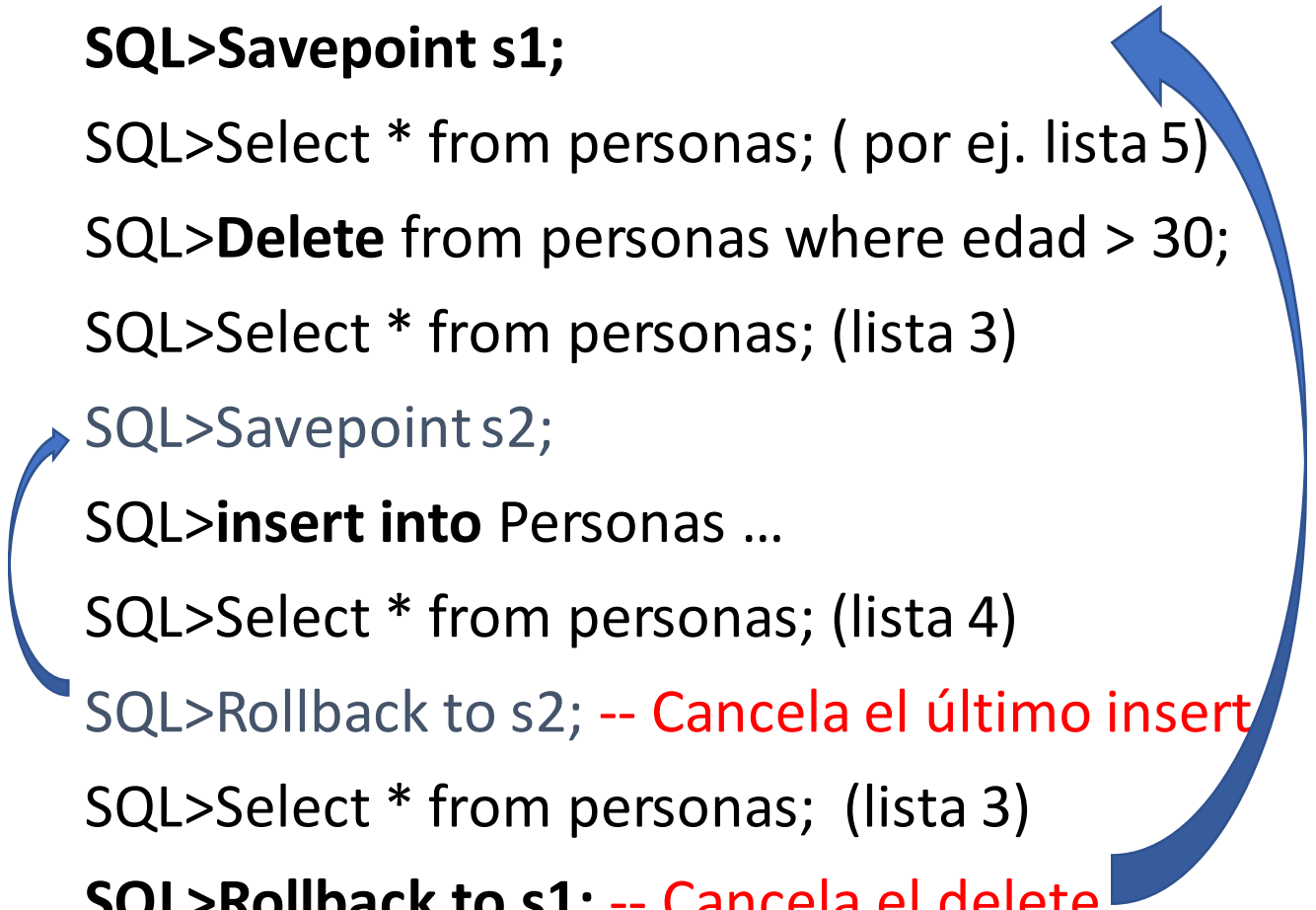
SQL>Select * from personas; (lista 4)

SQL>Rollback to s2; -- Cancela el último insert

SQL>Select * from personas; (lista 3)

SQL>Rollback to s1; -- Cancela el delete.

SQL>Select * from personas; (lista 5, las del principio)



Transacciones autónomas

- En ocasiones es necesario que los datos escritos por parte de una transacción sean persistentes a pesar de que la transaccion se deshaga con **ROLLBACK**.

- PL/SQL permite marcar un bloque con **PRAGMA AUTONOMOUS_TRANSACTION**. Con esta directiva marcamos el subprograma para que se comporte como transacción diferente a la del proceso principal, llevando el control de **COMMIT** o **ROLLBACK** independiente**.**

Ejemplo

```
CREATE OR REPLACE PROCEDURE Grabar_Log(descripcion VARCHAR2)
 IS
 PRAGMA AUTONOMOUS_TRANSACTION;
 BEGIN
   INSERT INTO LOG_APLICACION
   (CO_ERROR, DESCRIPICION, FX_ERROR)
   VALUES
   (SQ_ERROR.NEXTVAL, descripcion, SYSDATE);
 COMMIT;
 -- Este commit solo afecta a la transacción autónoma
 END ;
```

Ejemplo II

```
DECLARE
  producto PRECIOS%TYPE;
BEGIN
    producto := '100599';
    INSERT INTO PRECIOS
    (CO_PRODUCTO, PRECIO, FX_ALTA)
    VALUES
    (producto, 150, SYSDATE);
    COMMIT;
EXCEPTION
WHEN OTHERS THEN
   Grabar_Log(SQLERRM);
   ROLLBACK;
/* Los datos grabados por "Grabar_Log" se escriben en la base
de datos a pesar del ROLLBACK, ya que el procedimiento está
marcado como transacción autónoma.*/
END;
```