**1. ¿Qué es una transacción en una base de datos?**

Una transacción es un conjunto de operaciones **DML (Data Manipulation Language)** que se ejecutan como una **unidad lógica**. Esto significa que todas las operaciones incluidas en una transacción deben cumplirse exitosamente o deshacerse por completo si ocurre un error.

Por ejemplo, en una transferencia bancaria:

* Un débito en una cuenta debe ir acompañado de un crédito en otra.
* No es aceptable que uno de los dos pasos se ejecute sin el otro.

**2. Tipos de transacciones**

Oracle gestiona tres tipos de transacciones:

* **Un conjunto de comandos DML:** operaciones como INSERT, UPDATE y DELETE.
* **Un único comando DDL:** operaciones que alteran la estructura de las tablas, como CREATE, ALTER, y DROP.
* **Un único comando DCL:** comandos que gestionan permisos, como GRANT y REVOKE.

**3. ¿Cuándo comienza y termina una transacción?**

**Inicio de una transacción:**

* Una transacción inicia automáticamente cuando se ejecuta el primer comando DML.

**Fin de una transacción:**  
Una transacción termina de cualquiera de las siguientes formas:

1. **Con un COMMIT:** Hace permanentes todos los cambios realizados en la base de datos.
2. **Con un ROLLBACK:** Revierte todos los cambios realizados en la transacción actual.
3. **Automáticamente con comandos DDL/DCL:** Cada comando DDL o DCL incluye un COMMIT implícito.
4. **Salida del usuario del sistema:** Si sales de SQL\*Plus sin un COMMIT explícito, se realiza un ROLLBACK.
5. **Fallo del sistema:** En caso de caída, Oracle ejecuta un ROLLBACK automático para preservar la integridad de los datos.

**4. Comandos importantes en una transacción**

**4.1. COMMIT**

* Finaliza la transacción actual y **hace permanentes** los cambios realizados.
* Libera los bloqueos en las filas afectadas, permitiendo que otros usuarios las manipulen.
* Borra todos los puntos de guardado (SAVEPOINT) creados durante la transacción.

**4.2. SAVEPOINT**

* Marca un punto específico dentro de una transacción.
* Permite dividir la transacción en secciones más pequeñas para realizar un ROLLBACK parcial si es necesario.
  + Por ejemplo:

sql

Copiar código

SAVEPOINT punto1;

**4.3. ROLLBACK**

* Revierte los cambios realizados desde el inicio de la transacción o desde un punto de guardado.
* Sintaxis:

sql

Copiar código

ROLLBACK; -- Revierte toda la transacción

ROLLBACK TO punto1; -- Revierte hasta el SAVEPOINT "punto1"

**5. Estado de los datos antes y después de COMMIT**

**Antes de COMMIT o ROLLBACK:**

* Los cambios son **visibles sólo para el usuario que los hizo**.
* Otros usuarios no pueden ver los cambios.
* Las filas afectadas están **bloqueadas** para evitar conflictos.

**Después de COMMIT:**

* Los cambios se hacen permanentes y visibles para todos los usuarios.
* Se liberan los bloqueos.
* Se pierden las versiones previas de los datos.

**6. Consistencia en lectura**

La **consistencia en lectura** asegura que:

1. Los datos leídos por una consulta no reflejen cambios que aún no han sido confirmados (COMMIT).
2. Los datos visibles sean los mismos que existían al inicio de la transacción.

**Implementación técnica:**  
Oracle crea copias de los datos en segmentos de rollback antes de modificarlos. Esto permite a las consultas ver el estado previo de los datos, incluso si están siendo modificados.

**7. Bloqueos en Oracle**

**¿Qué son los bloqueos?**  
Son mecanismos que aseguran que las transacciones no interfieran entre sí.

* **Bloqueo exclusivo:** Solo una transacción puede modificar el recurso (por ejemplo, una fila).
* **Bloqueo compartido:** Permite que múltiples transacciones lean el mismo recurso, pero ninguna puede modificarlo mientras esté bloqueado.

**Ejemplo práctico**

sql

Copiar código

-- Inicio de transacción: Actualizamos una tabla

UPDATE EMPLEADOS SET salario = salario \* 1.1 WHERE deptno = 10;

-- Marcamos un punto de guardado

SAVEPOINT aumento\_salarios;

-- Insertamos un nuevo departamento

INSERT INTO DEPARTAMENTOS (deptno, nombre, ubicacion) VALUES (50, 'VENTAS', 'MADRID');

-- Decidimos revertir la última operación

ROLLBACK TO aumento\_salarios;

-- Confirmamos los cambios restantes

COMMIT;

**Consejos clave**

1. Siempre termina explícitamente una transacción con COMMIT o ROLLBACK.
2. Usa SAVEPOINT para controlar mejor transacciones largas o complejas.
3. Entiende cómo Oracle maneja los bloqueos automáticamente para evitar conflictos en tus operaciones.