
ORACLE
ACADEMY

Programación de bases de datos con SQL

18-1
Transacciones de Base de Datos

ORACLE ACADEMY

Objetivos

En esta lección, aprenderá a:

- Definir los términos COMMIT, ROLLBACK y SAVEPOINT y su relación con las transacciones de datos
- Enumerar tres ventajas de las sentencias COMMIT, ROLLBACK y SAVEPOINT
- Explicar por qué es importante, desde una perspectiva de negocio, poder controlar el flujo de procesamiento de transacciones

Objetivo

- ¿Qué ocurre si un banco no tenía ningún proceso sistemático para registrar los depósitos y retiradas?
- ¿Cómo sabe si un depósito se ha abonado en su cuenta antes de que tenga que retirar dinero?
- Puede imaginar la confusión que produciría.

Objetivo

- Afortunadamente, los bancos controlan los procesos de transacciones para garantizar la consistencia de los datos.
- En esta lección aprenderá cómo se gestiona el proceso de cambio de datos y cómo se confirman o cancelan los cambios en una base de datos.
- COMMIT, ROLLBACK y SAVEPOINT no están soportados en Oracle Application Express, debido a la forma en que Oracle Application Express gestiona las conexiones a la base de datos.

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

Transactions

- Las transacciones son un concepto fundamental de todos los sistemas de base de datos.
- Las transacciones permiten a los usuarios realizar cambios en los datos y, a continuación, decidir si desean guardar o desechar el trabajo.
- Las transacciones de base agrupar varios pasos en una unidad lógica de trabajo.

Transactions

- Una transacción consiste en una de las siguientes opciones:
  - Sentencias DML que constituyen un cambio consistente de los datos.
  - Las sentencias DML incluyen INSERT, UPDATE, DELETE y MERGE
  - Una sentencia DDL como CREATE, ALTER, DROP, RENAME o TRUNCATE
  - Una sentencia DCL como GRANT o REVOKE

Analogía de Transacción

- La base de datos de un banco contiene saldos de varias cuentas de cliente, así como saldos de depósito totales para otras sucursales.
- Supongamos que un cliente desea retirar y transferir dinero de su cuenta y depositarlo en otra cuenta del cliente en una sucursal diferente.

Analogía de Transacción

- Hay varios pasos independientes implicados para lograr esta operación bastante sencilla.
- Las dos sucursales bancarias desean asegurarse de que se producen todos los pasos de la transacción, o no se produce ninguno de ellos, y si el sistema falla, la transacción no se deja parcialmente terminada.
- La agrupación de los pasos de retirada y depósito en una transacción proporciona esta garantía.
- Una transacción se produce completamente o no se produce en absoluto.

Control de Transacciones

- Las transacciones se controlan mediante las siguientes sentencias:
  - COMMIT: representa el punto en el tiempo en el que el usuario ha realizado todos los cambios que quería para agruparlos lógicamente y, puesto que no se ha cometido ningún error, el usuario está listo para guardar el trabajo.
    - Cuando se emite una sentencia COMMIT, la transacción actual finaliza haciendo que todos los cambios pendientes sean permanentes.
  - ROLLBACK: permite al usuario desechar los cambios realizados en la base de datos.
    - Cuando se emite una sentencia ROLLBACK, se desechan todos los cambios pendientes.

Control de Transacciones

- Las transacciones se controlan mediante las siguientes sentencias:
  - SAVEPOINT: crea un marcador en una transacción, que divide la transacción en varias partes más pequeñas
  - ROLLBACK TO SAVEPOINT: permite al usuario realizar un rollback de la transacción actual hasta un punto de grabación especificado.
    - Si se ha cometido un error, el usuario puede emitir una sentencia ROLLBACK TO SAVEPOINT desechando solo los cambios realizados después de establecer SAVEPOINT.

Ejemplo de Transacción

- En el ejemplo, el usuario ha emitido una sentencia UPDATE e inmediatamente creado SAVEPOINT one.

```
UPDATE copy_departments
SET manager_id= 101
WHERE department_id = 60;
```

```
SAVEPOINT one;
```

- Después de una sentencia INSERT y una sentencia UPDATE (en la siguiente diapositiva), el usuario se ha dado cuenta de que no se ha incluido una cláusula WHERE en la última UPDATE.
- Para solucionar el error, el usuario ha emitido ROLLBACK TO SAVEPOINT one.

Consulte la diapositiva 5

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

Ejemplo de Transacción

- Los datos se han restaurado a su estado en SAVEPOINT one.

```
INSERT INTO copy_departments(department_id, department_name,
  manager_id, location_id)
VALUES(130, 'Estate Management', 102, 1500);
```

```
UPDATE copy_departments
SET department_id = 140;
```

← Cláusula WHERE omitida

```
ROLLBACK TO SAVEPOINT one;
```

```
COMMIT;
```


Consulte la diapositiva 5

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

¿Cuándo Empieza y Termina una Transacción?

- Una transacción empieza con la primera sentencia DML (INSERT, UPDATE, DELETE o MERGE).
  - Una transacción termina cuando se produce una de las siguientes situaciones:
  - Se emite una sentencia COMMIT o ROLLBACK.
  - Se emite una sentencia DDL (CREATE, ALTER, DROP, RENAME o TRUNCATE).
  - Se emite una sentencia DCL (GRANT o REVOKE).
  - Un usuario sale normalmente de la utilidad de la base de datos Oracle, lo que hace que la transacción actual que se confirme de forma implícita.

¿Cuándo Empieza y Termina una Transacción?

- Tras la finalización de una transacción, la siguiente sentencia SQL ejecutable inicia automáticamente la próxima transacción.
- Se confirma automáticamente una nueva sentencia DDL o DCL y, por lo tanto, finaliza de forma implícita una transacción.
- Todos los cambios de datos realizados durante una transacción son temporales hasta que se confirma la transacción.

Consistencia de los Datos

- Imagine dedicar varias horas a realizar cambios en los datos de los empleados solo para averiguar que alguien más está introduciendo información que ha entrado en conflicto con sus cambios.
- Para evitar estas interrupciones o conflictos y permitir que varios usuarios accedan a la base de datos al mismo tiempo, los sistemas de base de datos utilizan una implantación automática denominada "consistencia de lectura".

Consistencia de Lectura

- La consistencia de lectura garantiza una vista consistente de los datos para todos los usuarios en todo momento
- Que los lectores no visualicen datos en proceso de cambio.
- Que los escritores tengan garantizado que los cambios en la base de datos se realizarán de forma consistente.
- Que los cambios realizados por un escritor no destruyan ni entren en conflicto con los cambios que esté realizando otro escritor.

Consistencia de Lectura

- La consistencia de lectura es una implementación automática.
- Una copia parcial de la base de datos se mantiene en segmentos de deshacer. Cuando el usuario A emite una operación de inserción, actualización o supresión en la base de datos, Oracle Server toma una instantánea (copia) de los datos antes de cambiarlos y los escribe en un segmento de deshacer (rollback).
- El usuario B sigue viendo la base de datos como existía antes de que se iniciaran los cambios; ve la instantánea de los datos del segmento de deshacer.

Consistencia de Lectura

- Antes de confirmar los cambios en la base de datos, solo el usuario que está cambiando los datos ve los cambios; todos los demás ven la instantánea en el segmento de deshacer.
- Esto garantiza que los lectores de los datos vean datos consistentes en los que no se esté realizando actualmente ningún cambio.

Cambios Visibles

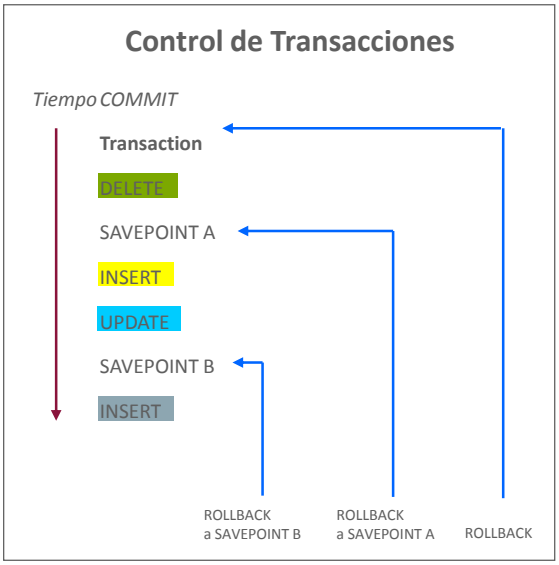
- Cuando se confirma una sentencia DML, cualquiera que ejecute la sentencia SELECT puede ver el cambio realizado.
- Si se realiza un rollback de la transacción, los cambios se deshacen:
  - La versión original anterior de los datos del segmento de deshacer se vuelve a escribir en la tabla.
  - Todos los usuarios ven la base de datos como existía antes de comenzar la transacción.

COMMIT, ROLLBACK y SAVEPOINT

- COMMIT y ROLLBACK garantizan la consistencia de los datos, posibilitando obtener una vista previa de los cambios en los datos antes de hacer que los cambios sean permanentes y para agrupar lógicamente operaciones relacionadas.
- SAVEPOINT crea un punto en una transacción al que puede realizar un rollback sin tener que deshacer toda la transacción.
- COMMIT, ROLLBACK y SAVEPOINT se denominan lenguaje de control de transacciones o TCL.

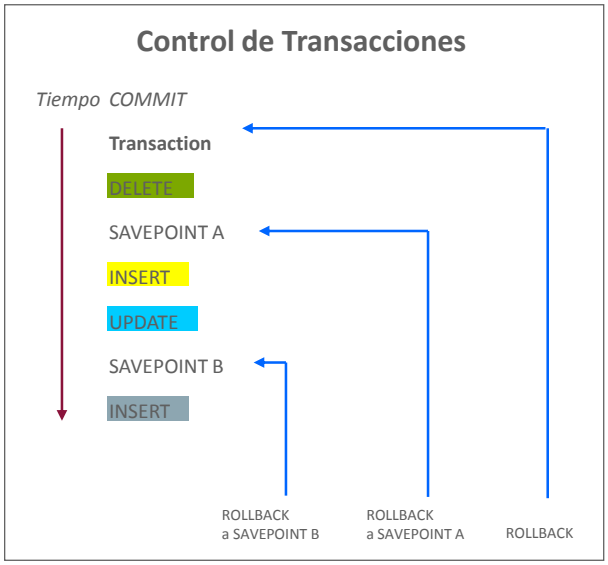
COMMIT, ROLLBACK y SAVEPOINT

- En la transacción que se muestra en el gráfico, se ha emitido una sentencia DELETE y, a continuación, se ha establecido SAVEPOINT A.
- Este SAVEPOINT actúa como un marcador que permitirá al usuario para realizar un rollback de los cambios posteriores en los datos al estado de los datos tal como existían en este punto.



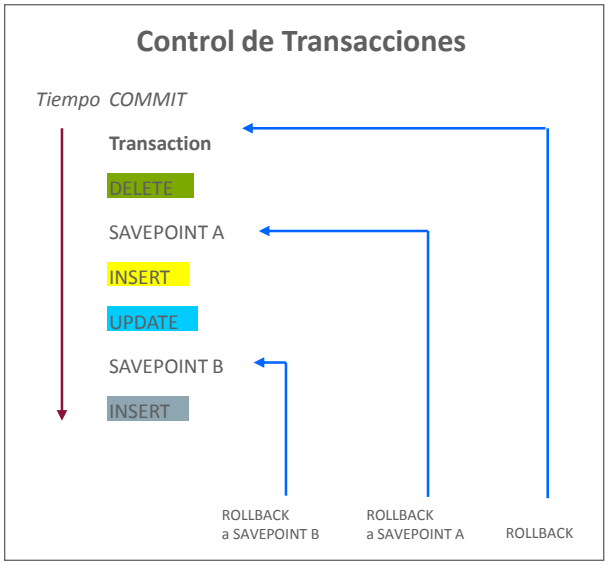
COMMIT, ROLLBACK y SAVEPOINT

- En el ejemplo, después de SAVEPOINT A, el usuario emite una sentencia INSERT y UPDATE y, a continuación, establece otro marcador de rollback a SAVEPOINT B.



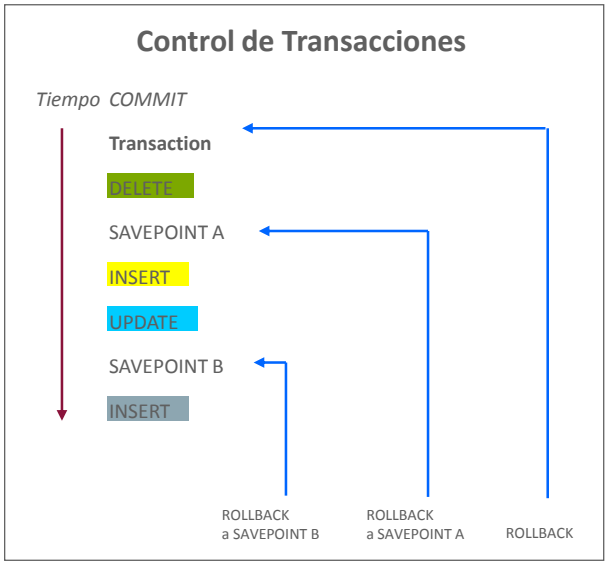
COMMIT, ROLLBACK y SAVEPOINT

- Si, por algún motivo el usuario no desea que se produzcan estas sentencias INSERT y/o UPDATE, puede emitir una sentencia ROLLBACK TO SAVEPOINT A.
- Esta acción realizará un rollback al estado de los datos tal como se encontraba en el marcador SAVEPOINT A.



COMMIT, ROLLBACK y SAVEPOINT

- La adición de otros SAVEPOINTS crea otros marcadores para puntos de rollback.
- Si un usuario emite una sentencia ROLLBACK sin una sentencia ROLLBACK TO SAVEPOINT, se finaliza toda la transacción y se desechan todos los cambios de datos pendientes.



SAVEPOINTS no son objetos de esquema y no se puede hacer referencia a ellos en el diccionario de datos. Cuando una transacción termina (por COMMIT o ROLLBACK) todos los SAVEPOINTS definidos en la transacción se pierden.

Procesamiento de Transacciones Implícitas

- La confirmación automática de cambios en los datos se produce en las siguientes circunstancias:
  - se emite una sentencia DDL
  - se emite una sentencia DCL
  - un usuario sale normalmente de la utilidad de la base de datos Oracle, lo que hace que la transacción actual que se confirme de forma implícita
  - se emite explícitamente sentencias COMMIT o ROLLBACK

Procesamiento de Transacciones Implícitas

- El rollback automático se produce tras una terminación anormal de la utilizad de base de datos Oracle o cuando se produce un fallo del sistema.
- Esto evita que cualquier error en los datos provoque cambios no deseados en las tablas subyacentes.
- Por lo tanto, la integridad de los datos está protegida.

Bloqueo

- Es importante evitar que más de un usuario cambie los datos a la vez.
- Oracle utiliza bloqueos para evitar una interacción destructiva entre transacciones que acceden al mismo recurso, tanto objetos de usuarios (como tablas o filas) como objetos del sistema no visibles para los usuarios (como estructuras de datos compartidos y filas del diccionario de datos).

Cómo Bloquea la Base de Datos Oracle los Datos

- El bloqueo de Oracle se realiza automáticamente y no necesita ninguna acción del usuario.
- El bloqueo implícito se produce para sentencias SQL según sea necesario, según la acción solicitada.
- El bloqueo implícito se produce para todas las sentencias SQL excepto SELECT.
- Los usuarios también pueden bloquear los datos de forma manual, lo que se denomina bloqueo explícito.
- Cuando se emite una sentencia COMMIT o ROLLBACK, se liberan los bloqueos en las filas afectadas.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Transaction
- confirmación
- Savepoint
- Rollback
- Lenguaje de control de transacciones
- Consistencia de lectura
- Locks

Resumen

En esta lección, ha aprendido lo siguiente:

- Definir los términos COMMIT, ROLLBACK y SAVEPOINT y su relación con las transacciones de datos
- Enumerar tres ventajas de las sentencias COMMIT, ROLLBACK y SAVEPOINT
- Explicar por qué es importante, desde una perspectiva de negocio, poder controlar el flujo de procesamiento de transacciones


ORACLE
ACADEMY