
Transacciones de Base de Datos

Objetivos

En este capítulo se revisará lo siguiente:

- Control de transacciones

Database Transactions

A database transaction consists of one of the following:

- DML statements which constitute one consistent change to the data
- One DDL statement
- One DCL statement

ORACLE

Transacciones de la Base de datos

El servidor de Oracle garantiza la consistencia de los datos con base en transacciones. Las transacciones proporcionan mayor flexibilidad y control cuando los datos cambian y ello asegura la consistencia de los datos en el caso de un fallo en el proceso del usuario o del sistema.

Las transacciones consisten de sentencias DML que componen un cambio consistente en los datos. Por ejemplo, una transferencia de fondos entre dos cuentas debe incluir el débito de una cuenta y el crédito a otra cuenta en la misma cantidad. En su conjunto ambas acciones deben fallar o triunfar; el crédito no se debe cometer sin el débito.

Tipos de transacciones

Tipo	Descripción
Lenguaje de Manipulación de Datos (DML Data manipulation Language)	Consiste de cualquier número de sentencias DML que el servidor de Oracle trate como una sola entidad o una unidad lógica de trabajo
Lenguaje de Definición de datos (DDL Data Definition Language)	Consiste de solo una sentencia DDL
Lenguaje de Control de Datos (DCL Data Control Language)	Consiste de solo una sentencia DCL

¿Cuándo inicia y finaliza una transacción?

Una transacción inicia cuando la primera sentencia DML es encontrada y finaliza cuando ocurre alguno de los siguientes puntos:

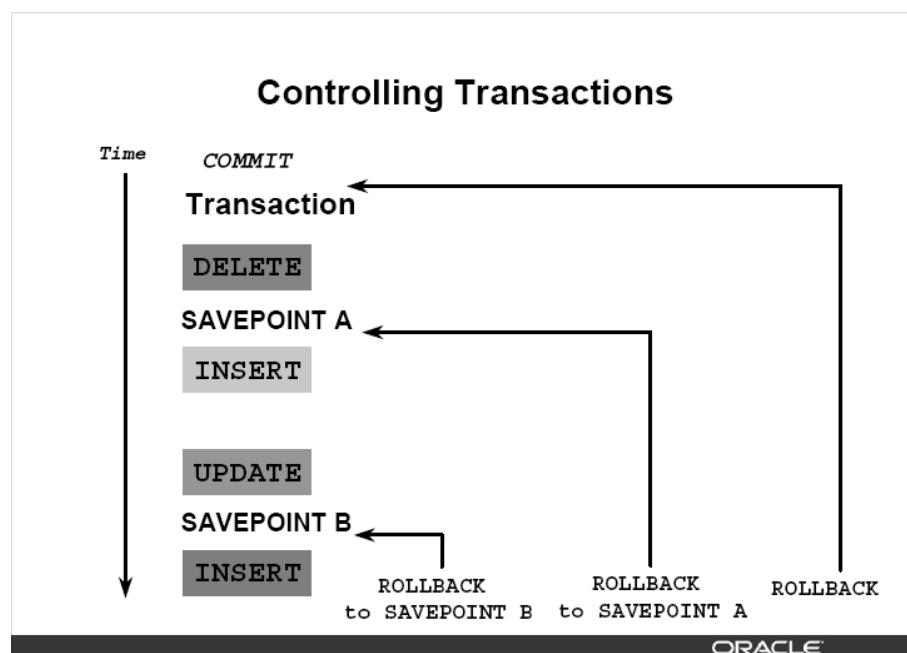
- Una sentencia COMMIT o ROLLBACK es usada
- Una sentencia DDL, como CREATE es utilizada
- Una sentencia DCL es usada
- El usuario sale de iSQL*Plus
- Una computadora falla o el sistema falla

Después de que una transacción finaliza, la siguiente sentencia SQL ejecutada automáticamente inicia la siguiente transacción.

Una sentencia DDL o DCL es automáticamente completada y por consiguiente implícitamente finaliza una transacción.

Ventajas de las sentencias COMMIT y ROLLBACK

- Asegurar la consistencia de datos
- Ver previamente los cambios de los datos antes de hacerlos permanentes
- Agrupar operaciones lógicamente relacionadas

**Control explícito del control de transacciones**

Se puede controlar la lógica de transacciones con el uso de las sentencias COMMIT, SAVEPOINT y ROLLBACK.

Sentencia	Descripción
COMMIT	Finaliza la transacción actual haciendo permanentes todos los cambios pendientes
SAVEPOINT name	Marca un punto de la transacción actual
ROLLBACK	Finaliza la transacción actual descartando todos los cambios pendientes

ROLLBACK to SAVEPOINT name	ROLLBACK TO SAVEPOINT retorna la transacción actual a la marca especificada, en consecuencia deshace todos los cambios y/o SAVEPOINTS creados después de la marca especificada a la cual retorna. Si se omite la cláusula TO SAVEPOINT, la sentencia ROLLBACK deshace toda la transacción.
-------------------------------	---

Nota: SAVEPOINT no es estándar ANSI de SQL.

Rolling Back Changes to a Marker

- Create a marker in a current transaction by using the SAVEPOINT statement.
- Roll back to that marker by using the ROLLBACK TO SAVEPOINT statement.

```

UPDATE...
SAVEPOINT update_done;
Savepoint created.
INSERT...
ROLLBACK TO update_done;
Rollback complete.

```

ORACLE

Deshaciendo cambios a un SAVEPOINT

Se pueden crear marcas en una transacción con la utilización de la sentencia SAVEPOINT la cual divide la transacción en secciones más pequeñas. Por lo que se pueden descartar los cambios pendientes hasta la marca con el uso de la sentencia ROLLBACK TO SAVEPOINT.

Si se crea una segunda marca con el mismo nombre de un savepoint anterior, el savepoint anterior es eliminado.

Procesando transacciones implícitas

Estatus	Circunstancias
COMMIT automático	Sentencias DDL o DCL son usadas. Abandonas /SQL*Plus de forma normal, sin el uso explícito de comandos COMMIT o ROLLBACK
ROLLBACK automático	Terminación anormal de /SQL*Plus o falla del sistema

Nota: Un tercer comando está disponible en /SQL*Plus. El comando AUTOCOMMIT puede ser habilitado o deshabilitado. Si se habilita, cada sentencia DML individual es cometida e inmediatamente es ejecutada. No se pueden deshacer estos cambios. Si se deshabilita, la sentencia COMMIT debe ser usada explícitamente. También, la sentencia COMMIT es usada cuando una sentencia DDL es usada o cuando sales de /SQL*Plus.

Fallas en el sistema

Cuando una transacción es interrumpida por que el sistema falla, la transacción entera es automáticamente deshecha. Esto previene errores por causas no deseadas en el cambio de los datos y regresa las tablas de su estado al estado del último commit. De esta manera, el servidor de Oracle protege la integridad de las tablas.

Desde iSQL*Plus, una salida normal de una sesión es lograda oprimiendo el botón "Exit". Con SQL*Plus, una salida normal es lograda escribiendo el comando EXIT en la línea de comandos. Cerrar la ventana es interpretado como una salida anormal.

Cometiando cambios (committing)

Cada dato que cambia durante la transacción es temporal hasta que la transacción es cometida (committed). El estado de los datos antes de usar las sentencias COMMIT o ROLLBACK es:

- Las operaciones de manipulación de datos afectan en primera instancia el buffer de la base de datos; por lo tanto, el estado previo de los datos se puede recuperar.
- El usuario actual puede revisar los resultados de las operaciones de manipulación de datos para consultar las tablas con el uso de la sentencia SELECT.
- Otros usuarios no pueden ver los resultados de las operaciones de manipulación de datos hechas por el usuario actual. El servidor de Oracle establece la consistencia para asegurar que cada usuario vea los datos como existieron desde el último commit.
- Las filas afectadas son bloqueadas; otros usuarios no pueden cambiar los datos de las filas afectadas.

Haga permanentes todos los cambios pendientes con el uso de la sentencia COMMIT. Después de una sentencia COMMIT:

- Los cambios de datos son escritos en la base de datos
- El estado previo de los datos es perdido permanentemente
- Todos los usuarios pueden ver los resultados de la transacción
- Los bloqueos en las filas afectadas son liberados; las filas son ahora disponibles para que otros usuarios realicen nuevos cambios a los datos
- Todos los savepoints son borrados.

Committing Data

- **Make the changes.**

```
DELETE FROM employees
WHERE employee_id = 99999;
1 row deleted.

INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
1 row inserted.
```
- **Commit the changes.**

```
COMMIT;
Commit complete.
```

ORACLE

En el ejemplo anterior se elimina una fila de la tabla EMPLOYEES y se inserta una nueva fila en la tabla DEPARTMENTS. Para hacer estos cambios permanentes se utiliza la sentencia COMMIT.

Ejemplo

Remueva los departamentos 290 y 300 de la tabla DEPARTMENTS, y actualice una fila en la tabla COPY_EMP. Haga todos los cambios permanentes.

```
DELETE FROM departments
WHERE department_id IN (290, 300);
2 rows deleted.
```

```
UPDATE copy_emp
SET department_id = 80
WHERE employee_id = 206;
1 row updated.
```

```
COMMIT;
Commit Complete.
```

State of the Data After ROLLBACK

Discard all pending changes by using the ROLLBACK statement:

- Data changes are undone.
- Previous state of the data is restored.
- Locks on the affected rows are released.

```
DELETE FROM copy_emp;  
22 rows deleted.  
ROLLBACK;  
Rollback complete.
```

ORACLE

Deshaciendo cambios (rolling back)

Deshaga todos los cambios pendientes con el uso de la sentencia ROLLBACK. Después de una sentencia ROLLBACK:

- Los cambios a los datos son deshechos
- El estado previo de los datos es restablecido
- Los bloqueos de las filas afectadas son liberados

Ejemplo

Mientras se intenta eliminar un registro de la tabla TEST, se puede accidentalmente vaciar la tabla. Se puede corregir el error, utilizando la sentencia apropiada, y haciendo permanente el cambio a los datos.

```
DELETE FROM test;  
25,000 rows deleted.
```

```
ROLLBACK;  
Rollback complete.
```

```
DELETE FROM test  
WHERE id = 100;  
1 row deleted.
```

```
SELECT *  
FROM test  
WHERE id = 100;  
No rows selected.
```

```
COMMIT;  
Commit complete.
```

Sentencia de Nivel ROLLBACK

La parte de una transacción puede ser excluida con un ROLLBACK implícito si en la ejecución de una sentencia es detectado un error. Si una sola sentencia DML falla durante la ejecución en una transacción, su efecto es deshacer por medio de una sentencia de nivel ROLLBACK, pero los cambios hechos por las anteriores sentencias DML en la transacción no son deshechos. Ellos pueden ser cometidos o deshechos explícitamente por el usuario.

Oracle utiliza un commit implícito antes y después de cualquier sentencia DDL. Así, incluso si la sentencia DDL no se ejecuta correctamente, no se puede deshacer la sentencia previa puesto que el servidor a usado un commit.

Termine sus transacciones explícitamente con la ejecución de sentencias COMMIT y ROLLBACK.

Consistencia en la lectura

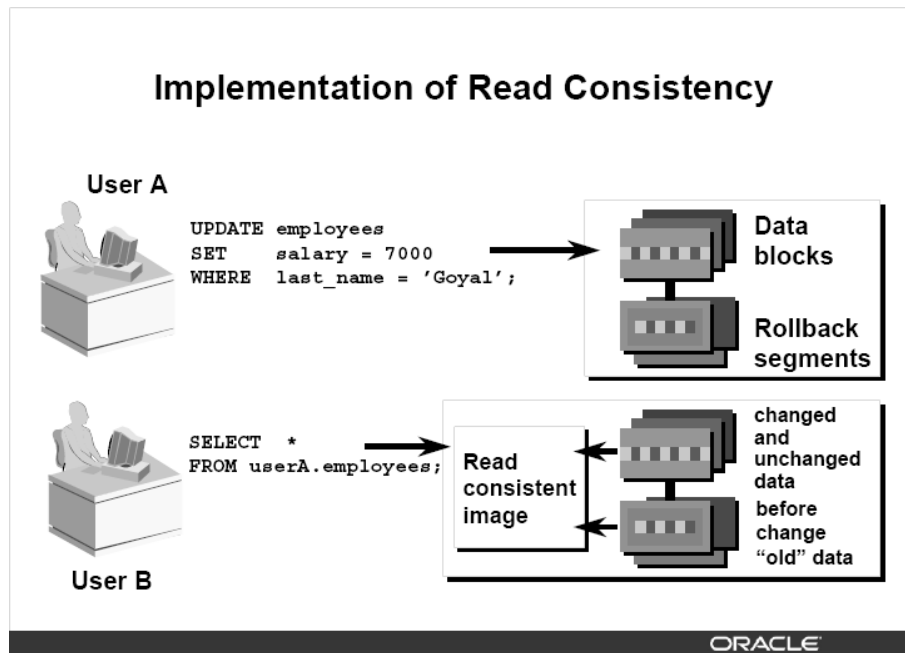
Los usuarios de la base de datos acceden a la base de datos de dos maneras:

- Con operaciones de lectura (Sentencia SELECT)
- Con operaciones de escritura (Sentencias INSERT, UPDATE y DELETE)

Se necesitan lecturas consistentes para que ocurra lo siguiente:

- Las operaciones de lectura y escritura a la base de datos aseguran una vista consistente de los datos
- Las operaciones de lectura no ven datos que estén en un proceso de cambio
- Las operaciones de escritura aseguran que los cambios a la base de datos sean hechos en un camino consistente.
- Los cambios hechos por una operación de escritura no interrumpen ni entran en conflicto con los cambios hechos por otra operación de escritura.

El propósito de las lecturas consistentes es asegurar que cada usuario vea los datos que existieron hasta el último commit, antes de que una operación DML inicie.



Implementación de lecturas consistentes

Las lecturas consistentes son una implementación automática. Esto mantiene una copia parcial de la base de datos en segmentos de deshecho (undo segments).

Cuando una operación INSERT, UPDATE o DELETE es hecha en la base de datos, el servidor Oracle toma una copia de los datos antes de que estos sean cambiados y escribe estos en un segmento de deshecho (*undo segment*).

Todas las lecturas, exceptuando la que es utilizada para hacer cambios, ve la base de datos tal como existió antes de que los cambios inicien; Ellos ven los segmentos de rollback ("snapshot"¹) de los datos.

Antes de que los cambios sean cometidos en la base de datos, solamente el usuario que esta modificando los datos ve los cambios realizados; todos los demás ven la imagen (snapshot) en el segmento de desecho. Esto garantiza que las lecturas de los datos obtengan datos consistentes que no estén en un proceso de cambio.

Cuando una sentencia DML es cometida, los cambios hechos a la base de datos se hacen visibles a cualquier usuario que ejecute una sentencia SELECT. El espacio ocupado por el antiguo dato en el archivo del segmento de desecho es liberado para su uso.

Si una transacción es deshecha, los cambios son deshechos:

- El original, la versión mas antigua del dato en el segmento de desecho es escrito de nuevo en la tabla
- Todos los usuarios ven la base de datos como existía antes que de la transacción iniciara.

¿Qué son los bloqueos?

Los bloqueos son mecanismos que impiden destruir interacciones entre transacciones acensando al mismo recurso, cualquier objeto de usuario (como

¹ Imagen de un segmento o porción de la base de datos que está siendo modificada

tablas o filas) u objetos del sistema no visibles a los usuarios (como estructuras de datos compartidas y filas del diccionario de datos).

Como la base de datos de Oracle bloquea datos

Los bloqueos de Oracle son ejecutados automáticamente y no requiere la acción del usuario. Implícitamente los bloqueos ocurren necesariamente a causa de sentencias SQL, dependiendo de la acción solicitada. Implícitamente los bloqueos ocurren para todas las sentencias SQL con excepción del SELECT.

Los usuarios también pueden bloquear datos manualmente, esto es llamado un bloqueo explícito.

Bloqueos DML

Cuando se ejecutan operaciones DML (Lenguaje de manipulación de Datos). El servidor de Oracle proporciona concurrencia a los datos por medio de bloqueos DML. Los bloqueos DML ocurren en dos niveles:

- Un bloqueo compartido (Share lock) es automáticamente adquirido a nivel de tabla durante operaciones DML. Con el modo de bloqueo compartido, diversas transacciones pueden adquirir bloqueos compartidos en el mismo recurso.
- Un bloqueo exclusivo (exclusive lock) es adquirido automáticamente para cada fila modificada por una sentencia DML. Los bloqueos exclusivos previenen que la fila sea cambiada por otras transacciones hasta que la transacción que bloqueo la fila, sea cometida o deshecha. Este bloqueo asegura otros usuarios no puedan modificar la misma fila al mismo tiempo y sobrescribir los cambios no cometidos por otro usuario.
- Los bloqueos DDL ocurren cuando modificas un objeto de la base de datos como una tabla.

Resumen

En este capítulo se ha visto como utilizar las sentencias para el control de transacciones.

Sentencia	Descripción
COMMIT	Hace permanentes todos los cambios pendientes
SAVEPOINT	Es usado para deshacer hasta la marca indicada
ROLLBACK	Descarta todos los cambios pendientes

El servidor Oracle garantiza consistencia en la vista de los datos todo el tiempo.

Los bloqueos pueden ser implícitos o explícitos.