



ACADEMY

Fundamentos de bases de datos

6-5

Lenguaje de control de transacciones (TCL)

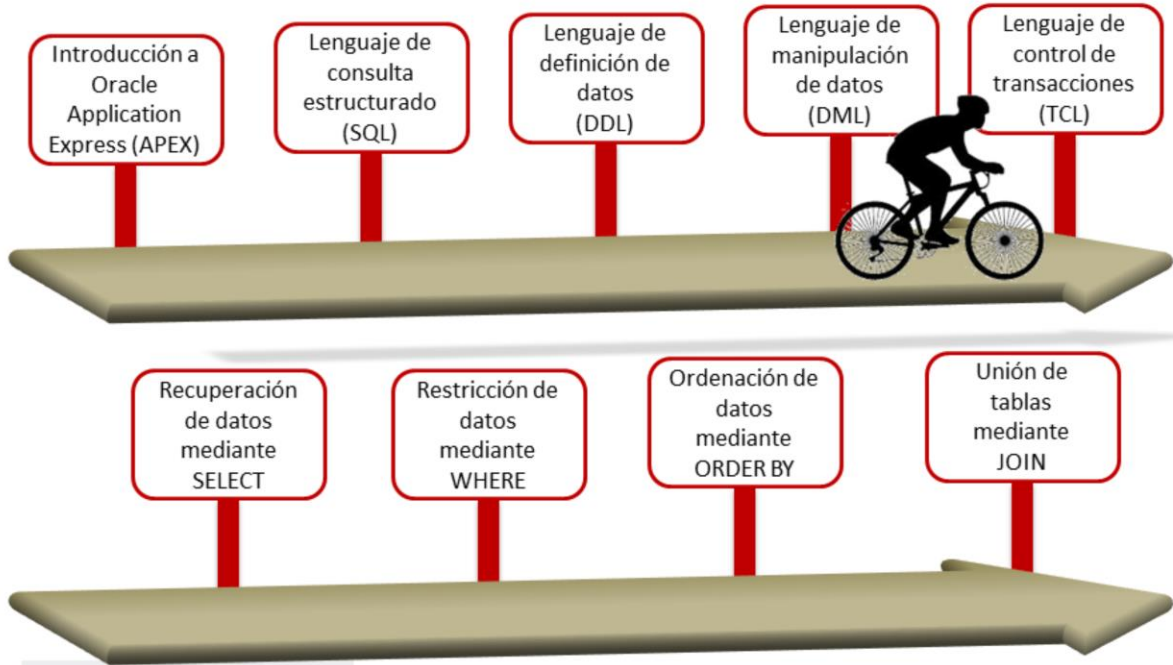


ORACLE ACADEMY

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

Guía básica

Se encuentra aquí



Objetivos

En esta lección se abordan los siguientes objetivos:

- Describir la finalidad del lenguaje de control de transacciones (TCL)
- Explicar las operaciones TCL que son necesarias para gestionar una transacción:
 - COMMIT
 - SAVEPOINT
 - ROLLBACK
- Describir la necesidad de consistencia de lectura



Application Express

- 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.



Transacciones de Base de Datos

- Las transacciones constan de sentencias DML que representan un cambio consistente en los datos.
- El servidor de Oracle garantiza la consistencia de los datos basada en transacciones.
- Las transacciones le proporcionan más flexibilidad y control al cambiar los datos y garantizan la consistencia de los datos en caso de fallo de proceso de usuario o del sistema.

Transacciones de Base de Datos

- Por ejemplo, una transferencia de fondos entre dos cuentas debe incluir el débito en una cuenta y el crédito en otra con la misma cantidad. Ambas acciones deben ser correctas o incorrectas por igual; el crédito no se debe confirmar sin el débito.

Transacciones de base de datos

Una transacción de base de datos consta de una de las siguientes sentencias:

- Sentencias DML que representan un cambio consistente en los datos
- Una sentencia DDL
- Una sentencia TCL

Transacciones de base de datos: Inicio y fin

- Una transacción empieza cuando se ejecuta la primera sentencia SQL de DML.
- Termina con uno de los siguientes eventos:
 - Se emite una sentencia `COMMIT` o `ROLLBACK`.
 - Se ejecuta una sentencia DDL o TCL (confirmación automática).
 - El usuario sale del software SQL en uso.
 - El sistema falla.

Ventajas de las sentencias COMMIT y ROLLBACK

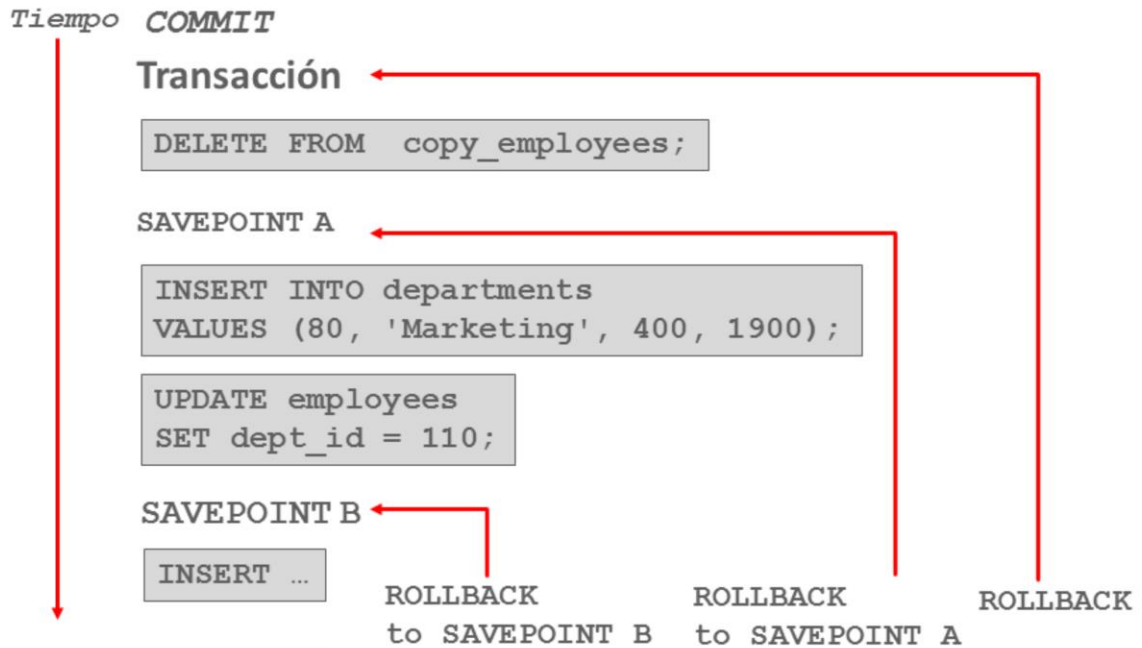
Con las sentencias COMMIT y ROLLBACK, puede:

- Garantizar la consistencia
- Visualizar una presentación preliminar de los cambios de los datos antes de hacerlos permanentes
- Agrupan componentes relacionados de forma lógica
- Tener control sobre cambios permanentes de los datos

Sentencias de Control de Transacciones

Sentencia	Descripción
COMMIT	Finaliza la transacción actual convirtiendo todos los cambios de datos pendientes en permanentes.
SAVEPOINT <i>nombre</i>	Marca un punto de grabación en la transacción actual.
ROLLBACK	Finaliza la transacción actual desechando todos los cambios de datos pendientes.
ROLLBACK TO SAVEPOINT <i>nombre</i>	Realiza un rollback de la transacción actual en el punto de grabación especificado, desechando de esta forma los cambios y/o puntos de grabación creados después del punto de grabación en el que está realizando el rollback. Si omite la cláusula TO SAVEPOINT, la sentencia ROLLBACK realiza un rollback de toda la transacción. Puesto que los puntos de grabación son lógicos, no hay forma de mostrar los puntos de grabación que ha creado.

Sentencias de Control de Transacciones Explícitas



Puede controlar la lógica de las transacciones mediante las sentencias **COMMIT**, **SAVEPOINT** y **ROLLBACK**.

Nota: No puede aplicar la sentencia **COMMIT** en **SAVEPOINT**. **SAVEPOINT** no es SQL estándar de ANSI.

Nota: **COMMIT** y **ROLLBACK** no están soportadas actualmente en APEX.

Cómo Deshacer Cambios de un Marcador

- Cree un marcador en una transacción actual mediante la sentencia `SAVEPOINT`.
- Descarte cambios pendientes realizando rollback en dicho marcador mediante la sentencia `ROLLBACK TO SAVEPOINT`.

```
UPDATE...  
SAVEPOINT update_done;
```

```
SAVEPOINT update_done succeeded.
```

```
INSERT...  
ROLLBACK TO update_done;
```

```
ROLLBACK TO succeeded.
```

Nota: Si crea un segundo punto de grabación con el mismo nombre que un punto de grabación anterior, éste se suprime.

Procesamiento de Transacciones Implícitas

- Una transacción automática se produce en las siguientes circunstancias:
 - Se emite una sentencia DDL.
 - Se emite una sentencia TCL.
 - Hay una salida normal del software SQL, sin emitir explícitamente las sentencias COMMIT o ROLLBACK
- Se produce un rollback automático cuando hay una terminación anormal del software SQL o cuando hay un fallo del sistema para proteger la integridad de la base de datos.

Estado de los datos antes de COMMIT o ROLLBACK

- Todos los cambios realizados durante una transacción son temporales si no se confirman, por lo tanto, se puede recuperar el estado anterior de los datos.
- La sesión actual puede revisar los resultados de las operaciones DML mediante la sentencia `SELECT`.
- Las demás sesiones no pueden ver los resultados de las sentencias DML emitidas por la sesión actual.
- Las filas afectadas están bloqueadas; otras sesiones no pueden cambiar los datos de las filas afectadas.

Estado de los datos después de COMMIT

- Los cambios de datos se guardan en la base de datos.
- Se sobrescribe el estado anterior de los datos.
- Todas las sesiones pueden ver los resultados.
- Los bloqueos de las filas afectadas se liberan y dichas filas quedan disponibles para que las manipulen otras sesiones.
- Se borran todos los puntos de grabación.

Confirmación de Datos

- Realice estos cambios:

```
DELETE FROM copy_employees  
WHERE employee_id=113;
```

1 row(s) deleted.

```
INSERT INTO copy_departments  
VALUES (290, 'Corporate Tax', NULL, 1700);
```

1 row(s) inserted.

- Confirme los cambios:

```
COMMIT;
```



ACADEMY

DFo 6-S
Lenguaje de control de transacciones(TCL)

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 17

En el siguiente ejemplo, se eliminan los departamentos 290 y 300 en la tabla copy_departments, actualiza una fila en la tabla copy_employees y guarda los cambios en los datos.

```
DELETE FROM copy_departments  
WHERE department_id IN (290, 300);
```

```
UPDATE copy_employees  
SET department_id = 80  
WHERE employee_id = 206;
```

```
COMMIT;
```

Nota: La sentencia COMMIT no está soportada actualmente en APEX.

Estado de los datos después de ROLLBACK

Deseche todos los cambios pendientes mediante la sentencia ROLLBACK:

- Se deshacen los cambios de datos.
- Se restaura el estado anterior de los datos.
- Se liberan los bloqueos de las filas afectadas.

```
DELETE FROM copy_employees;  
ROLLBACK ;
```

Estado de los datos después de ROLLBACK: Ejemplo

Imagine que hay una tabla de prueba con 4 registros:

```
DELETE FROM test;
```

4 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.

Nota: La sentencia ROLLBACK no está soportada actualmente en APEX.

Rollback a nivel de sentencia

- Si falla una sentencia DML durante la ejecución, solo se realiza un rollback de dicha sentencia.
- El servidor de Oracle implementa un punto de grabación implícito.
- Los demás cambios se retienen.
- El usuario debe terminar las transacciones explícitamente con la ejecución de una sentencia COMMIT o ROLLBACK .

El servidor de Oracle emite una confirmación implícita antes y después de cualquier sentencia DDL.

Por lo tanto, si la sentencia DDL no se ejecuta correctamente, no podrá realizar rollback de la sentencia anterior porque el servidor ha emitido una confirmación.

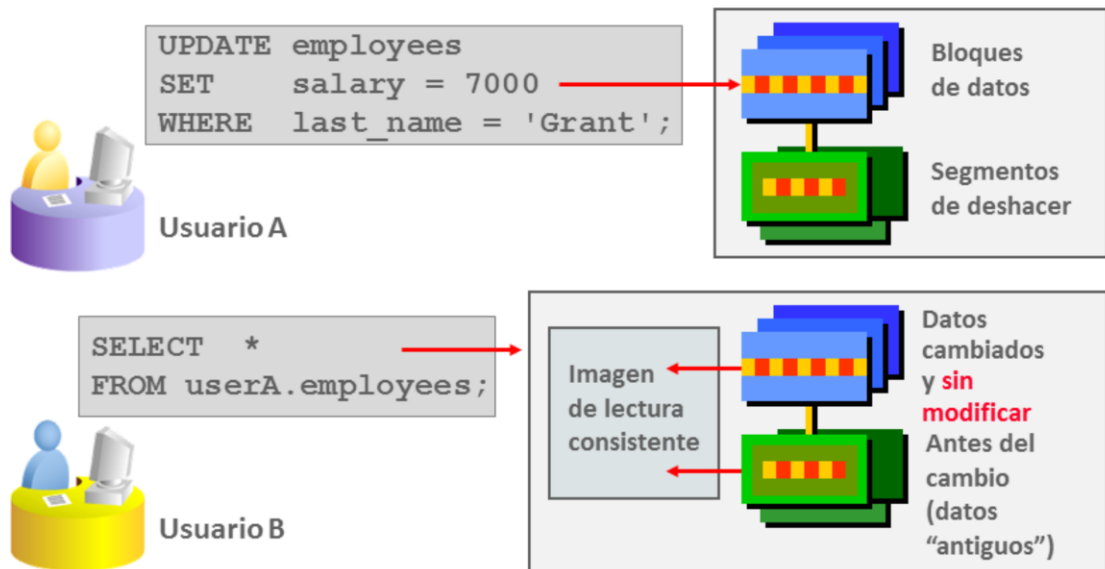
Termine las transacciones explícitamente con la ejecución de una sentencia COMMIT o ROLLBACK.

Consistencia de Lectura

- La consistencia de lectura garantiza una vista consistente de los datos en todo momento.
Cada usuario ve los datos como estaban en la última sentencia COMMIT.
- Los cambios realizados por un usuario no entran en conflicto con los cambios realizados por otro usuario.
- La consistencia de lectura garantiza que en los mismos datos:
 - Los lectores no esperen a los escritores.
 - Los escritores no esperen a los lectores.
 - Los escritores esperen a los escritores.

Nota: El mismo usuario se puede conectar a diferentes sesiones. Cada sesión mantiene la consistencia de lectura, independientemente de la sesión en la que esté conectado el usuario.

Implantación de Consistencia de Lectura



Nota: para obtener una explicación más detallada sobre la consistencia de lectura, consulte las notas de la diapositiva.

ORACLE

ACADEMY

DFo 6-5
Lenguaje de control de transacciones (TCL)

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 22

La consistencia de lectura es una implementación automática. Mantiene una copia parcial de la base de datos en los segmentos de deshacer. La imagen de lectura consistente se crea a partir de los datos confirmados de la tabla y de los datos antiguos que se están cambiando y que aún no se han confirmado del segmento de deshacer.

Al realizar una operación de inserción, actualización o supresión en la base de datos, el servidor de Oracle realiza una copia de los datos antes de cambiarlos y los escribe en un *segmento de deshacer*.

Todos los lectores, excepto el que haya realizado el cambio, seguirán viendo la base de datos como estaba antes de que comenzaran los cambios; verán una "instantánea" de los datos del segmento de deshacer.

Antes de confirmar los cambios en la base de datos, sólo el usuario que modifica los datos ve la base de datos con modificaciones. Todos los demás verán la instantánea en el segmento de deshacer. Esto garantiza que los lectores de los datos lean datos consistentes en los que no se esté realizando actualmente ningún cambio.

Al confirmar una sentencia DML, el cambio realizado en la base de datos se hace visible para todos aquellos que emiten una sentencia `SELECT` después de realizar la confirmación. El espacio ocupado por los datos antiguos en el archivo de segmentos de deshacer se libera para volver a utilizarlo.

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 y todos los usuarios ven la base de datos como estaba antes de comenzar la transacción.

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Describir la finalidad del lenguaje de control de transacciones (TCL)
- Explicar las operaciones TCL que son necesarias para gestionar una transacción:
 - COMMIT
 - SAVEPOINT
 - ROLLBACK
- Describir la necesidad de consistencia de lectura





ACADEMY