
ORACLE
ACADEMY

Programación de bases de datos con SQL

12-2
Actualización de Valores de Columna y Supresión de Filas

ORACLE ACADEMY

Objetivos

En esta lección, aprenderá a:

- Construir y ejecutar una sentencia UPDATE
- Crear y ejecutar una sentencia DELETE
- Crear y ejecutar una consulta que utilice una subconsulta para actualizar y suprimir datos de una tabla
- Crear y ejecutar una consulta que utilice una subconsulta correlacionada para actualizar y suprimir datos de una tabla

Objetivos

En esta lección, aprenderá a:

- Explicar cómo las restricciones de integridad de clave ajena y clave primaria afectan a las sentencias UPDATE y DELETE
- Explicar el objetivo de la cláusula FOR UPDATE en una sentencia SELECT

Objetivo

- ¿No sería un mundo maravilloso si, una vez que se ha obtenido algo, nunca fuera necesario cambiarlo o rehacerlo?
- La cama se quedaría hecha, la ropa se mantendría limpia y siempre obtendría aprobados.
- Lamentablemente en las bases de datos, como en la vida, "No hay nada permanente excepto el cambio".
- La actualización, inserción, supresión y gestión de datos es un trabajo del administrador de base de datos (DBA).
- En esta lección, se convertirá en el DBA de su propio esquema y aprenderá a gestionar la base de datos.

UPDATE:

- La sentencia UPDATE se utiliza para modificar filas existentes de una tabla.
- UPDATE necesita cuatro valores:
  - el nombre de la tabla
  - el nombre de las columnas cuyos valores se van a modificar.
  - un nuevo valor para cada una de las columnas que se están modificando.
  - una condición que identifique las filas de la tabla que se van a modificar
- El nuevo valor para una columna puede ser el resultado de una subconsulta de una sola fila.

Utilice solo tablas copy_tablename en todas las transacciones de DML.

UPDATE

- En el ejemplo, se utiliza una sentencia UPDATE para cambiar el número de teléfono de un empleado en la tabla de empleados.
- Tenga en cuenta que la tabla copy_employees se utiliza en esta transacción.

```
UPDATE copy_employees
SET phone_number = '123456'
WHERE employee_id = 303;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Wright	123456

UPDATE

- Podemos cambiar varias columnas y/o varias filas en una sentencia UPDATE.
- En este ejemplo, se cambia el número de teléfono y el apellido de dos empleados.

```
UPDATE copy_employees
SET phone_number = '654321', last_name = 'Jones'
WHERE employee_id >= 303;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Jones	654321
304	Test	Jones	654321

Al actualizar varias columnas, las columnas deben estar separadas por una coma.

UPDATE

- Tenga cuidado al actualizar los valores de columna.
- Si la cláusula WHERE se omite, todas las filas de la tabla se actualizarán.
- Puede que esto no sea lo deseado.

```
UPDATE copy_employees
SET phone_number = '654321', last_name = 'Jones';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
100	Steven	Jones	654321
101	Neena	Jones	654321
102	Lex	Jones	654321
200	Jennifer	Jones	654321
205	Shelley	Jones	654321
206	William	Jones	654321
149	Eleni	Jones	654321
174	Ellen	Jones	654321
…	…	…	…

Actualización de una Columna con un Valor de una Subconsulta

- Podemos utilizar el resultado de una subconsulta de una sola fila para proporcionar el valor nuevo para una columna actualizada.

```
UPDATE copy_employees
SET salary = (SELECT salary
      FROM copy_employees
           WHERE employee_id = 100)
WHERE employee_id = 101;
```

- En este ejemplo, se cambia el salario de un empleado (id = 101) al mismo salario que otro empleado (id = 100).

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	24000

Actualización de una Columna con un Valor de una Subconsulta

```
UPDATE copy_employees
SET salary = (SELECT salary
      FROM copy_employees
          WHERE employee_id = 100)
WHERE employee_id = 101;
```

- Como es habitual, la subconsulta se ejecuta en primer lugar y recupera el salario del identificador de empleado = 100.
- A continuación, este valor de salario se utiliza para actualizar el salario del identificador de empleado = 101.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	24000

Actualización de Dos Columnas con Dos Sentencias de Subconsulta

- Para actualizar varias columnas en una sentencia UPDATE, es posible escribir varias subconsultas de una sola fila, una para cada columna.
- En el ejemplo siguiente, la sentencia UPDATE cambia el salario de un empleado y el identificador de cargo (id = 206) a los mismos valores que otro empleado (id = 205).

Actualización de Dos Columnas con Dos Sentencias de Subconsulta

```
UPDATE copy_employees
SET salary = (SELECT salary
      FROM copy_employees
      WHERE employee_id = 205),
  job_id = (SELECT job_id
      FROM copy_employees
      WHERE employee_id = 205)
WHERE employee_id = 206;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOB_ID
205	Shelley	Higgins	12000	AC_MGR
206	William	Gietz	12000	AC_MGR

Actualización de Filas Basada en Otra Tabla

- Como puede que se espere, la subconsulta puede recuperar información de una tabla que, a continuación, se utiliza para actualizar otra tabla.
- En este ejemplo, se ha creado una copia de la tabla de empleados.
- A continuación, los datos de la tabla de empleados original se han recuperado, copiado y utilizado para rellenar la tabla copy_employees.

```
UPDATE  copy_employees
SET salary  = (SELECT salary
     FROM employees
     WHERE employee_id = 205)
WHERE employee_id  = 202;
```

Actualización de Filas con Subconsulta Correlacionada

- Como ya sabe, las subconsultas pueden independientes o correlacionadas.
- En una subconsulta correlacionada, actualiza una fila de una tabla según una selección de la misma tabla.

Actualización de Filas con Subconsulta Correlacionada

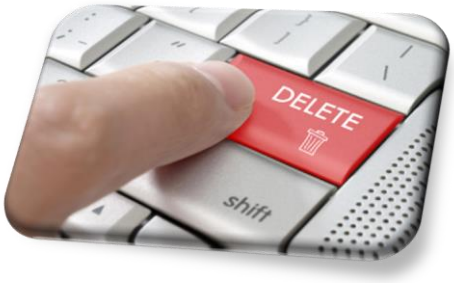
- En el siguiente ejemplo, se ha creado una copia de la columna de nombre de departamento en la tabla de empleados.
- A continuación, los datos de la tabla de departamentos original se han recuperado, copiado y utilizado para rellenar la copia de la columna en la tabla copy_employees.

```
ALTER TABLE  copy_employees
ADD (department_name varchar2(30) NOT NULL);
```

```
UPDATE  copy_employees e
SET e.department_name  = (SELECT d.department_name
                FROM departments d
                WHERE e.department_id = d.department_id);
```

DELETE

- La sentencia DELETE se utiliza para eliminar filas existentes de una tabla.
- La sentencia necesita dos valores:
  - el nombre de la tabla
  - la condición que identifica las filas que se van a suprimir



DELETE

- En el ejemplo, se utiliza la copia de la tabla de empleados para suprimir una fila: el empleado con el número de identificador 303.

```
DELETE from copy_employees
WHERE employee_id = 303;
```

- ¿Qué predice que se suprimirá si se elimina la cláusula WHERE en una sentencia DELETE?
- Se suprimen todas las filas de la tabla si omite la cláusula WHERE:

Subconsulta DELETE

- Las subconsultas también se pueden utilizar en sentencias DELETE.
- En el ejemplo, se suprimen las filas de la tabla de empleados para todos los empleados que trabajan en el departamento de envíos.
- Es posible que este departamento haya cambiado de nombre o se haya cerrado.

```
DELETE FROM copy_employees
WHERE  department_id =
 (SELECT department_id
  FROM departments
  WHERE  department_name = 'Shipping');
```

5 filas eliminadas.

Subconsulta DELETE

- El ejemplo siguiente elimina filas de todos los empleados que trabajan para un gerente que administra menos de 2 empleados .

```
DELETE FROM copy_employees e
WHERE  e.manager_id IN
       (SELECT d.manager_id
        FROM employees d
        HAVING count (d.department_id) < 2
        GROUP BY d.manager_id);
```

Errores de Restricción de Integridad

- Las restricciones de integridad garantizan que los datos se ajusten a un juego anidado de reglas.
- Las restricciones se comprueban automáticamente cada vez que se ejecuta una sentencia DML que puede romper las reglas.
- Si se rompe alguna regla, la tabla no se actualiza y se devuelve un error.

Errores de Restricción de Integridad

- En este ejemplo, se viola una restricción NOT NULL porque last_name tiene una restricción de no nulo e id=999 no existe, por lo que la subconsulta devuelve un resultado nulo.

```
UPDATE copy_employees
SET last_name = (SELECT last_name
       FROM copy_employees
       WHERE employee_id = 999)
WHERE employee_id = 101;
```

```
ORA-01407: cannot update
("US_A009EMEA815_PLSQL_T01"."COPY_EMPLOYEES"."LAST_NAME") to NULL
```

Errores de Restricción de Integridad

- ¿Cuándo se comprobarán las restricciones clave primaria-clave ajena?
- La tabla EMPLOYEES tiene una restricción de clave ajena en la columna department_id que hace referencia a department_id de la tabla DEPARTMENTS.
- Esto garantiza que cada empleado pertenezca a un departamento válido.
- En la tabla DEPARTMENTS, department_id 10 y 20 existen pero 15 no.

Errores de Restricción de Integridad

- ¿Cuál de las siguientes sentencias devolverá un error?

```
1.      UPDATE employees SET department_id = 15
        WHERE employee_id = 100;

2.      DELETE FROM departments WHERE department_id = 10;

3.      UPDATE employees SET department_id = 10
        WHERE   department_id = 20;
```


Consulta 1: devolverá un error porque department_id 15 no existe.

Consulta 2: devolverá un error si hay empleados con un department_id de 10

Consulta 3 se realizará correctamente suponiendo que department_id 10 esté en la tabla de departamentos.

Errores de Restricción de Integridad

- Al modificar las copias de tablas (por ejemplo, copy_employees), es posible que vea errores de restricción de valor no nulo, pero no verá ningún error de restricción de clave primaria-clave ajena.
- El motivo es que la sentencia CREATE TABLE ... AS (SELECT …) que se utiliza para crear la copia de la tabla, copia las filas y las restricciones de valores no nulos, pero no copia las restricciones de clave primaria-clave ajena.
- Por lo tanto, no existe ninguna restricción de clave primaria-clave ajena en ninguna de las tablas copiadas.
- La adición de restricciones se abordará en otra lección.

Cláusula FOR UPDATE en una Sentencia SELECT

- Cuando se emite una sentencia SELECT en una tabla de base de datos, no se emite ningún bloqueo en la base de datos en las filas devueltas por la consulta que está emitiendo.
- La mayoría de las veces, esta es así como desea que se comporte la base de datos, para mantener el número de bloqueos emitidas en un mínimo.
- Sin embargo, a veces, desea asegurarse de que nadie más puede actualizar o suprimir los registros que la consulta está devolviendo mientras trabaja en esos registros.

Cláusula FOR UPDATE en una Sentencia SELECT

- Entonces es cuando se utiliza la cláusula FOR UPDATE.
- En cuanto se ejecuta su consulta, la base de datos emitirá automáticamente bloqueos de nivel de fila exclusivos en todas las filas devueltas por la sentencia SELECT, que se mantendrán hasta que emita un comando COMMIT o ROLLBACK.
- Recordatorio: la instancia en línea/alojado de APEX se confirmará automáticamente y el bloqueo de nivel de fila no se realizará.

Cláusula FOR UPDATE en una Sentencia SELECT

- Si utiliza una cláusula FOR UPDATE en una sentencia SELECT con varias tablas en ella, todas las filas de todas las tablas se bloquearán.

```
SELECT e.employee_id, e.salary, d.department_name
FROM employees e JOIN departments d  USING (department_id)
WHERE job_id = 'ST_CLERK'  AND location_id = 1500
FOR UPDATE
ORDER BY e.employee_id;
```

Cláusula FOR UPDATE en una Sentencia SELECT

- Estas cuatro filas ahora están bloqueadas por el usuario que ha ejecutado la sentencia SELECT hasta que el usuario emita un COMMIT o ROLLBACK.

EMPLOYEE_ID	SALARY	DEPARTMENT_NAME
141	3500	Envío
142	3100	Envío
143	2600	Envío
144	2500	Envío

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- DELETE
- Restricción de integridad
- UPDATE
- Subconsulta correlacionada UPDATE
- Subconsulta correlacionada DELETE
- FOR UPDATE de SELECT

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Construir y ejecutar una sentencia UPDATE
- Crear y ejecutar una sentencia DELETE
- Crear y ejecutar una consulta que utilice una subconsulta para actualizar y suprimir datos de una tabla
- Crear y ejecutar una consulta que utilice una subconsulta correlacionada para actualizar y suprimir datos de una tabla

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Explicar cómo las restricciones de integridad de clave ajena y clave primaria afectan a las sentencias UPDATE y DELETE
- Explicar el objetivo de la cláusula FOR UPDATE en una sentencia SELECT


ORACLE
ACADEMY