

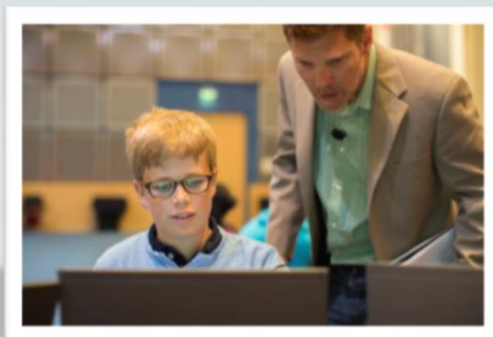


ACADEMY

Fundamentos de bases de datos

6-4

Lenguaje de manipulación de datos (DML)



ORACLE ACADEMY

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

Guía básica

Se encuentra aquí

Introducción a
Oracle
Application
Express (APEX)

Lenguaje de
consulta
estructurado
(SQL)

Lenguaje de
definición de
datos
(DDL)

Lenguaje de
manipulación
de datos
(DML)

Lenguaje de
control de
transacciones
(TCL)



Recuperación
de datos
mediante
SELECT

Restricción de
datos
mediante
WHERE

Ordenación de
datos
mediante
ORDER BY

Unión de
tablas
mediante
JOIN

Objetivos

En esta lección se abordan los siguientes objetivos:

- Describir la finalidad del lenguaje de manipulación de datos (DML)
- Explicar las operaciones DML que son necesarias para gestionar los datos de tabla de una base de datos:
 - INSERT
 - UPDATE
 - DELETE



Lenguaje de manipulación de datos

- Las sentencias DML se ejecutan al:
 - Agregar nuevas filas a una tabla (`INSERT`)
 - Modificar filas existentes en una tabla (`UPDATE`)
 - Eliminar filas existentes de una tabla (`DELETE`)
- Una transacción consta de una recopilación de sentencias DML que forman una unidad lógica de trabajo.



Piense en una base de datos bancaria. Cuando un cliente del banco transfiere dinero de su cuenta de ahorro a una cuenta corriente, la transacción puede constar de las siguientes tres acciones diferentes: reducir la cuenta de ahorro, aumentar la cuenta corriente y registrar la transacción en el diario de transacciones. El servidor de Oracle debe garantizar que se ejecuten las tres sentencias SQL para mantener el balance correcto de las cuentas. Si algo impide que una de las sentencias de la transacción se ejecute, las demás sentencias de la transacción se deben deshacer.

Adición de una nueva fila a una tabla

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
80	Sales	145	2500

Nueva fila

70	Public Relations	204	2700
----	------------------	-----	------

**Insertar nueva fila en la
tabla DEPARTMENTS.**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500

Sintaxis de la sentencia INSERT

- Agregue filas a una tabla mediante la sentencia `INSERT`:

```
INSERT INTO table [(column [, column...])]  
VALUES                (value [, value...]);
```

- Con esta sintaxis, solo se inserta una fila cada vez.

En la sintaxis:

- `table` es el nombre de la tabla.
- `column` es el nombre de la columna de la tabla que desea rellenar.
- `value` es el valor correspondiente para la columna.

Sintaxis de la sentencia INSERT

- Antes de ejecutar cualquier operación DML en una tabla, realice una copia de la tabla:

```
CREATE copy_departments  
AS (SELECT * FROM departments);
```

- Tenga en cuenta que al copiar una tabla de esta manera, no se copian todas las restricciones (solo las NOT NULL).

Inserción de filas

- Si inserta una fila que contiene valores para cada columna, no es necesaria la lista de columnas en la cláusula `INSERT`
- Mostrar valores en el orden por defecto de las columnas de la tabla
- Se debe proporcionar un valor para cada columna

```
INSERT INTO copy_departments  
VALUES (40, 'Advertising', 201, 1800);
```

Inserción de filas

- Opcionalmente, enumere las columnas en la cláusula `INSERT`.

```
INSERT INTO copy_departments (department_id,  
                               department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);
```

- Enumere los valores en el mismo orden en que aparecen los campos.
- Delimite los valores de caracteres y de fecha entre comillas simples.

Escenario de caso: Inserción de filas

Entiendo que la sentencia **INSERT** se utiliza para agregar filas a una tabla. ¿Es posible insertar una fila si algunas columnas no tienen valores?



Alumno



Profesor

Por supuesto. Puede utilizar la sentencia **INSERT** para especificar las columnas para las que no se introducirán valores.

Inserción de filas con valores nulos

- Método explícito : Omita la columna en la lista de columnas.

```
INSERT INTO copy_departments (department_id, department_name)
VALUES (30, 'Purchasing');
```

- Método implícito : Especifique la palabra clave `NULL` en la cláusula `VALUES`.

```
INSERT INTO copy_departments
VALUES (100, 'Finance', NULL, NULL);
```

Asegúrese de que puede utilizar valores nulos en la columna de destino mediante la verificación del estado `Null` con el comando `DESCRIBE`.

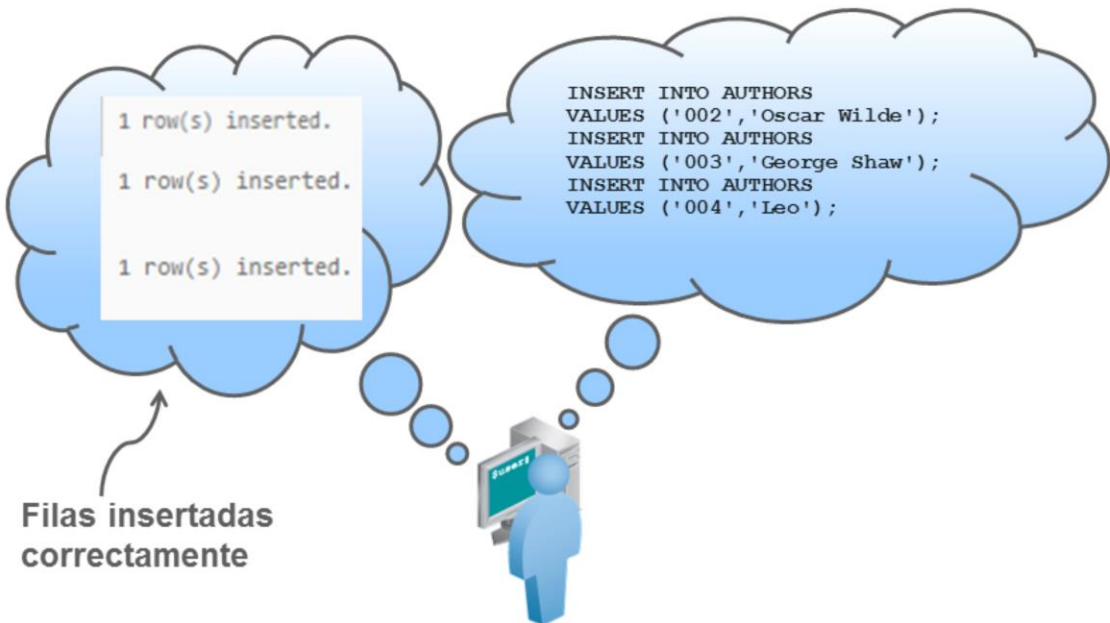
El servidor de Oracle aplica automáticamente todos los tipos de dato, rangos de datos y restricciones de integridad de los datos. Las columnas que no se muestran explícitamente obtienen un valor nulo en la nueva fila, a menos que haya valores por defecto para las columnas que faltan que se utilizan.

Los errores comunes que se producen durante la entrada del usuario se comprueban en el siguiente orden:

- Falta el valor obligatorio de una columna `NOT NULL`.
- Un valor duplicado viola cualquier restricción de clave única o primaria.
- El valor `Any` viola una restricción `CHECK`.
- Una clave ajena viola la restricción de integridad referencial.
- No coincidencias de tipos de dato o los valores son demasiado anchos para la columna.

Nota: Se recomienda el uso de la lista de columnas porque hace la sentencia `INSERT` más legible y fiable, y menos proclive a errores.

Escenario de caso: Inserción de filas



Inserción de valores especiales

- Puede utilizar funciones para introducir valores especiales en la tabla.
- La función `SYSDATE` registra la fecha y la hora actuales.

```
INSERT INTO copy_employees (employee_id,  
                             first_name, last_name,  
                             email, phone_number,  
                             hire_date, job_id, salary,  
                             commission_pct, manager_id,  
                             department_id)  
VALUES  
    (113,  
     'Louis', 'Popp',  
     'LPOPP', '515.124.4567',  
     SYSDATE, 'AC_ACCOUNT', 6900,  
     NULL, 205, 110);
```



DFo 6-4
Lenguaje de manipulación de datos (DML)

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 14

Nota: En primer lugar, cree la tabla `copy_employees`: `CREATE copy_employees AS (SELECT * FROM employees);`

En el ejemplo de la diapositiva se registra información sobre el empleado Popp en la tabla `EMPLOYEES`. Proporciona la fecha y hora actuales en la columna `HIRE_DATE`. Utiliza la función `SYSDATE` para devolver la fecha y hora actuales del servidor de base de datos.

También puede utilizar la función `CURRENT_DATE` para devolver la fecha actual en la zona horaria de la sesión.

Al insertar filas en una tabla, también puede utilizar la función `USER` para registrar el nombre de usuario actual.

Inserción de valores de fecha y hora específicos

- Agregue un empleado.

```
INSERT INTO copy_employees
VALUES      (114, 'Den', 'Raphealy',
             'DRAPHEAL', '515.127.4561',
             TO_DATE('Dec 7, 2002', 'MON DD, YYYY'),
             'SA_REP', 11000, 0.2, 100, 60, NULL);
```

1 rows inserted

- Verifique la adición.

114	Den	Raphealy	DRAPHEAL	515.127.4561	07-Dec-2002
-----	-----	----------	----------	--------------	-------------

Ejercicio 1 del proyecto

DFo_6_4_1_Project

Base de datos de la tienda Oracle Baseball League:
Uso de operaciones DML para gestionar las tablas
de la base de datos: Inserción de registros

Cambio de datos en la tabla

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	90
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

Actualizar filas en la tabla EMPLOYEES:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	60
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	60
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	60
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

Sintaxis de la sentencia UPDATE

- Modifique los valores existentes en una tabla con la sentencia UPDATE:

UPDATE	table
SET	column = value [, column = value, ...]
[WHERE	condition];

- En general, utilice la columna de clave primaria en la cláusula WHERE para identificar una única fila para la actualización.
- Actualice más de una fila cada vez (si es necesario).

En la sintaxis:

- `table` es el nombre de la tabla.
- `column` es el nombre de la columna de la tabla que se debe rellenar.
- `value` es el valor o subconsulta correspondiente para la columna.
- `condition` identifica las filas que se van a actualizar y consta de los nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

Actualización de filas en una tabla

- Si se especifica la cláusula `WHERE`, se modifican los valores de una fila o varias filas específicas:

```
UPDATE copy_employees
SET    department_id = 50
WHERE  employee_id = 113;
```

- Si se omite la cláusula `WHERE`, se modifican los valores de **todas** las filas de la tabla:

```
UPDATE    copy_employees
SET       department_id = 110;
```



Por ejemplo, un empleado que era `SA_REP` ahora ha cambiado su puesto a `IT_PROG`. Por lo tanto, se debe actualizar su `JOB_ID` y el campo de comisiones se debe definir en `NULL`.

```
UPDATE copy_employees
SET job_id = 'IT_PROG', commission_pct = NULL
...
WHERE employee_id = 114;
```

Nota: La tabla `copy_employees` tiene los mismos datos que la tabla `EMPLOYEES`.

Actualización de filas en una tabla

- Especifique `SET column_name= NULL` para actualizar un valor de columna a `NULL`.

```
UPDATE    copy_employees
SET       department_id = NULL
WHERE     employee_id = 124;
```

Violación de restricciones

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

Error starting at line 1 in command:

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110
```

Error report:

```
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
ORA-02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:      A foreign key value has no matching primary key value.
*Action:     Delete the foreign key or add a matching primary key.
```

- El departamento 55 no existe en la tabla principal, DEPARTMENTS.



DFo 6-4
Lenguaje de manipulación de datos (DML)

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 21

Nota: Tenga en cuenta que la consulta utiliza la tabla employees, que tiene restricciones de integridad aplicadas (copy_employees no tiene estas restricciones aplicadas).

Si hay restricciones aplicadas en las columnas, se devolverá un error si intenta violar la regla de restricción. Por ejemplo, si intenta actualizar un registro con un valor ligado a una restricción de integridad, se devuelve un error.

En el ejemplo de la diapositiva, el departamento 55 no existe en la tabla principal, DEPARTMENTS, por lo tanto, recibirá la violación ORA-02291 "parent key not found".

Eliminación de filas de una tabla

En esta diapositiva se muestra que el departamento de relaciones públicas se ha eliminado de la tabla DEPARTMENTS (asumiendo que no se han violado las restricciones en la tabla DEPARTMENTS).

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700

Fila suprimida de la tabla DEPARTMENTS:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
80	Sales	145	2500
90	Executive	100	1700

Sentencia DELETE

- Puede eliminar filas existentes de una tabla mediante la sentencia DELETE:

```
DELETE [FROM] table  
[WHERE condition];
```

En la sintaxis:

- `table` es el nombre de la tabla.
- `condition` identifica las filas que se van a suprimir y consta de los nombres de columna, expresiones, constantes, subconsultas y operadores de comparación.

Nota: Si no se suprime ninguna fila, se devuelve el mensaje “0 rows deleted”.

Supresión de filas de una tabla

- Si se incluye la cláusula `WHERE`, se suprimen filas concretas:

```
DELETE FROM copy_departments  
WHERE department_name = 'Purchasing';
```

- Si se omite la cláusula `WHERE`, se suprimen **todas** las filas de la tabla:

```
DELETE FROM copy_departments;
```

A continuación, se muestra un ejemplo de eliminación de las filas identificadas en la cláusula `WHERE`:

```
DELETE FROM copy_departments WHERE department_id IN (30, 40);
```


Violación de restricciones

- No se puede suprimir una fila que contenga una clave primaria utilizada como clave ajena en otra tabla.

```
DELETE FROM departments
WHERE department_id = 60;
```

Error starting at line 1 in command: DELETE FROM departments WHERE department_id = 60	
Error report: SQL Error: ORA-02292: integrity constraint (ORA1.JHIST_DEPT_FK) violated - child record found 02292. 00000 - "integrity constraint (%s.%s) violated - child record found" *Cause: attempted to delete a parent key value that had a foreign dependency. *Action: delete dependencies first then parent or disable constraint.	

Nota: Tenga en cuenta que la consulta utiliza la tabla departments, que tiene restricciones de integridad aplicadas (copy_departments no tiene estas restricciones aplicadas).

Por ejemplo, si intenta suprimir un registro con un valor ligado a una restricción de integridad, se devuelve un error.

En el ejemplo de la diapositiva, se muestra un intento de suprimir el departamento 60 de la tabla DEPARTMENTS, pero se produce un error porque ese número de departamento se utiliza como clave ajena en la tabla EMPLOYEES. Si el registro principal que intenta suprimir tiene registros secundarios, recibirá la violación ORA-02292 "child record found".

Escenario de caso: Supresión de filas

¿Es posible suprimir todas las filas de una tabla, pero dejar la estructura intacta?



Alumno



Profesor

Por supuesto. Puede utilizar la sentencia **TRUNCATE** para ello.

Sentencia TRUNCATE

- Elimina todas las filas de una tabla, dejando la tabla vacía y la estructura de la misma intacta.
- Es una sentencia DDL más que una sentencia DML; no se puede deshacer fácilmente.
- Sintaxis:

```
TRUNCATE TABLE table_name;
```

- Ejemplo:

```
TRUNCATE TABLE copy_employees;
```

**** Consulte las notas sobre el truncamiento de tablas principales**



DFo 6-4
Lenguaje de manipulación de datos (DML)

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados. 27

La sentencia `TRUNCATE` es un método más eficaz para eliminar todas las filas de una tabla o cluster.

La eliminación de filas con la sentencia `TRUNCATE` es más rápida que la eliminación con la sentencia `DELETE` por las siguientes razones:

- La sentencia `TRUNCATE` es una sentencia DDL y no genera ninguna información de rollback. La información de rollback se trata más adelante en esta lección.
- El truncamiento de una tabla no arranca los disparadores de supresión de la tabla.

Si la tabla es el principal de una restricción de integridad referencial, no puede truncarla. Debe desactivar la restricción antes de emitir la sentencia `TRUNCATE`. La desactivación de las restricciones se trata en la lección titulada "Introducción a sentencias DDL".

Ejercicio 2 del proyecto

DFo_6_4_2_Project

Base de datos de la tienda Oracle Baseball League:

Uso de operaciones DML para gestionar las tablas de la base de datos: Actualización y supresión de registros

Resumen

En esta lección, debe haber aprendido a hacer lo siguiente:

- Describir el objetivo del DML
- Mostrar las operaciones DML que son necesarias para gestionar los datos de tabla de una base de datos:
 - INSERT
 - UPDATE
 - DELETE





ACADEMY