

8

Manipulación de Datos

ORACLE®
DATABASE **11^g**

ORACLE®

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Describir cada sentencia DML**
- **Insertar filas en una tabla**
- **Actualizar las filas de una tabla**
- **Suprimir filas de una tabla**
- **Fusionar las filas de una tabla**
- **Controlar transacciones**

Lenguaje de Manipulación de Datos (DML)

- Se ejecuta una sentencia DML cuando:
 - Agrega filas nuevas a una tabla
 - Modifica las filas existentes de una tabla
 - Elimina filas existentes de una tabla
- Una transacción consta de una recopilación de sentencias DML que forman una unidad de trabajo lógica.

Adición de una Fila Nueva a una Tabla

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

**Fila
Nueva**

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
100	Finance	108	1700
110	Accounting	205	1700
120	Treasury		1700

**...inserte una fila
nueva en la tabla
DEPARTMENTS**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
250	Retail Sales		1700
260	Recruiting		1700
270	Payroll		1700
70	Public Relations	204	2700

Sintaxis de la Sentencia INSERT

- Agregue nuevas filas a una tabla mediante la sentencia INSERT

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

- Con esta sintaxis sólo se inserta una fila cada vez.

Inserción de Filas Nuevas

- Inserte una fila nueva que contenga valores para cada columna.
- Enumere los valores en el orden por defecto de las columnas de la tabla.
- Opcionalmente, liste las columnas en la cláusula INSERT.

```
INSERT INTO departments(department_id, department_name,  
                        manager_id, location_id)  
VALUES      (700, 'Public Relations', 100, 1700);  
1 rows created.
```

- Escriba los valores de caracteres y de fecha entre comillas simples.

Inserción de Filas con Valores Nulos

- **Método implícito: Omita columnas de la lista de columnas.**

```
INSERT INTO departments(department_id, department_name)
VALUES      (301, 'Purchasing');
```

1 rows created.

- **Método explícito: Especifique la palabra clave NULL en la cláusula VALUES.**

```
INSERT INTO departments
VALUES      (101, 'Finance', NULL, NULL);
```

1 rows created.

Inserción de Valores Especiales

La función SYSDATE registra la fecha y la hora actuales.

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES (207,  
        'Louis', 'Popp',  
        'LPOPP1', '515.124.4567',  
        SYSDATE, 'AC_ACCOUNT', 6900,  
        NULL, 205, 100);
```

1 row created.

Inserción de Valores de Fecha Específicos

- Agregue un empleado nuevo.

```
INSERT INTO employees
VALUES      (208,
             'Den', 'Raphealy',
             'DRAPHEAL1', '515.127.4561',
             TO_DATE('FEB 3,1999', 'MON DD,YYYY'),
             'AC_ACCOUNT', 11000, NULL, 100, 30 );
```

1 rows created.

- Verifique la adición.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DAT	JOB_ID	SALARY	COMMISSION_PCT
114	Den	Raphaely	DRAPHEAL	515.127.4561	07/12/94	PU_MAN	11000	

Creación de un Archivo de Comandos

- Utilice la sustitución & en una sentencia SQL para solicitar valores.
- & es una variable pendiente de asignación para el valor de la variable.

```
INSERT INTO departments
      (department_id,department_name, location_id)
VALUES (&department_id, '&department_name', &location);
```

1 row created

Copia de Filas desde otra Tabla

- Cree la tabla **SALES_REPS**

```
CREATE TABLE sales_reps(  
  id NUMBER(6) PRIMARY KEY,  
  name VARCHAR2(25),  
  salary NUMBER(8,2),  
  commission_pct NUMBER(2,2)  
);
```

Table **SALES_REPS** creado.

Copia de Filas desde otra Tabla

- Escriba la sentencia INSERT con una subconsulta.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
SELECT employee_id, last_name, salary, commission_pct
FROM   employees
WHERE  job_id LIKE '%REP%';
```

4 rows created

- No utilice la cláusula VALUES.
- Haga coincidir el número de columnas de la cláusula INSERT con el de la subconsulta.

Cambio de los datos de una Tabla

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DAT	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_PCT
100	Steven	King	SKING	17/06/87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21/09/89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13/01/93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03/01/90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21/05/91	IT_PROG	6000	60	
105	David	Austin	DAUSTIN	25/06/97	IT_PROG	4800	60	
106	Valli	Pataballa	VPATABAL	05/02/98	IT_PROG	4800	60	
107	Diana	Lorentz	DLORENTZ	07/02/99	IT_PROG	4200	60	
108	Nancy	Greenberg	NGREENBE	17/08/94	FI_MGR	12000	100	

Actualice las filas de la tabla EMPLOYEES.



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DAT	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_PCT
100	Steven	King	SKING	17/06/87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21/09/89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13/01/93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03/01/90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21/05/91	IT_PROG	6000	30	
105	David	Austin	DAUSTIN	25/06/97	IT_PROG	4800	30	
106	Valli	Pataballa	VPATABAL	05/02/98	IT_PROG	4800	30	
107	Diana	Lorentz	DLORENTZ	07/02/99	IT_PROG	4200	30	
108	Nancy	Greenberg	NGREENBE	17/08/94	FI_MGR	12000	100	

La Sintaxis de la Sentencia UPDATE

- Modifique las filas existentes con la sentencia UPDATE.

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

- Actualice más de una fila cada vez si es necesario.

Actualización de las Filas de una Tabla

- Se incluye la cláusula **WHERE**, las filas específicas se modifican.

```
UPDATE    employees
SET       department_id = 70
WHERE     employee_id = 113;
1 row updated.
```

- Se modifican todas las filas de la tabla si omite la cláusula **WHERE**.

```
UPDATE    sales_reps
SET       salary = 10000;
33 rows updated.
```

Actualización de Dos Columnas con una Subconsulta

```
UPDATE    employees
SET        job_id =      (SELECT job_id
                           FROM    employees
                           WHERE   employee_id = 205),
          salary =      (SELECT salary
                           FROM    employees
                           WHERE   employee_id = 205)
WHERE      employee_id = 114;
1 row updated.
```


Eliminación de una Fila de 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
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

Suprima una fila 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
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
110	Accounting	205	1700
120	Treasury		1700
130	Corporate Tax		1700

La Sentencia DELETE

Puede eliminar las filas existentes de una tabla utilizando la sentencia DELETE.

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

Supresión de Filas de una Tabla

- Se suprimen filas específicas si incluye la cláusula WHERE.

```
DELETE FROM departments  
WHERE    department_name = 'Contracting';  
1 row deleted.
```

- Se suprimen todas las filas de la tabla si omite la cláusula WHERE.

```
DELETE FROM sales_reps;  
33 rows updated.
```

Supresión de Filas Basándose en otra Tabla

Utilice subconsultas en sentencias DELETE para eliminar las filas de una tabla basándose en los valores de otra tabla.

```
DELETE FROM employees
WHERE department_id IN (SELECT department_id
                        FROM departments
                        WHERE department_name LIKE '%Public%');
```

1 row deleted.

Instrucción TRUNCATE

- Quita todas las filas de una tabla, dejando la tabla vacía y deja la estructura intacta.
- Es una instrucción (DDL) más bien que una declaración DML; no puede ser deshecho fácilmente.
- Sintaxis

```
TRUNCATE TABLE table_name;
```

- Ejemplo

```
TRUNCATE TABLE sales_reps;
```

Validación de los Datos

Deseche todos los cambios pendientes utilizando la sentencia ROLLBACK:

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

```
DELETE FROM sales_reps;
```

```
33 rows deleted.
```

```
ROLLBACK;
```

```
Rollback complete.
```

Validación de Datos

- Realice los Cambios.

```
DELETE FROM employees  
WHERE employee_id = 104;  
1 row deleted.
```

```
INSERT INTO departments  
VALUES (290, 'Corporate Tax', NULL, 1700);  
1 row inserted.
```

- Valide los Cambios

```
COMMIT;  
Commit complete.
```

Estado de los Datos Después de ROLLBACK

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.

Resumen

En esta lección, debería haber aprendido a utilizar sentencias DML y a controlar transacciones.

Sentencia	Descripción
INSERT	Agrega una fila nueva a la tabla.
UPDATE	Modifica las filas existentes de la tabla.
DELETE	Elimina las filas existentes de la tabla.
COMMIT	Hace que todos los cambios pendientes sean permanentes.
ROLLBACK	Revierte todos los cambios de datos pendientes.

Visión General de la Práctica 8

Esta práctica cubre los siguientes temas:

- **Inserción de filas en las tablas.**
- **Actualización y supresión de filas de la tabla**
- **Control de transacciones**