



2. Ordres de manipulació

NF2. Procediments i funcions. Ordres per manipular les dades

UF3 - Llenguatges SQL: DCL i extensió procedimental

Desenvolupament d'Aplicacions Multiplataforma

M02 – Bases de dades. Versió 1.0

© M^a Carmen Brito Ruiz



2.1. Órdenes de manipulación de datos

2.1.1. Consultar datos

2.1.2. Eliminar datos

2.1.3. Insertar datos

2.1.4. Actualizar datos

2.2. Ejemplos

2.1. Órdenes de manipulación de datos

Las mismas órdenes o sentencias que hemos visto en la unidad didáctica 2 para manipular datos, las podemos usar en PL/SQL.

```
SELECT <lista_campos>  
INTO <variables>  
FROM <nombre_tabla>  
[WHERE <condición>];
```

```
DELETE FROM <nombre_tabla>  
[WHERE <condición>];
```

```
INSERT INTO <nombre_tabla>  
VALUES <valor i/o variables>;
```

```
UPDATE <nombre_tabla>  
SET <campo> = <valor_nuevo>  
[WHERE <condición>];
```

2.1.1. Consultar datos

```
SELECT <lista_campos>  
INTO <variables>  
FROM <nombre_tabla>  
[WHERE <condición>];
```

donde:

lista_campos ⇒ es una lista de campo o campos, y puede incluir expresiones SQL, funciones de fila o funciones de grupo.

variables ⇒ es la variable escalar que almacena el valor que se recupera en la consulta.

nombre_tabla ⇒ especifica el nombre de la tabla de la base de datos.

condición ⇒ esta compuesta por nombres de campos, expresiones, constantes y operadores de comparación, incluyendo constantes y variables PL/SQL.

2.1.2. Eliminar datos

```
DELETE FROM <nombre_tabla>  
[WHERE <condición>];
```

donde:

nombre_tabla ⇒ especifica el nombre de la tabla de la base de datos.

condición ⇒ esta compuesta por nombres de campos, expresiones, constantes y operadores de comparación, incluyendo constantes y variables PL/SQL.

2.1.3. Insertar datos

```
INSERT INTO <nombre_tabla>  
VALUES <valor i/o variables>;
```

donde:

nombre_tabla ⇒ especifica el nombre de la tabla de la base de datos.

valor/variables ⇒ es el valor a insertar o bien, es la variable escalar que almacena el valor que se va a insertar.

2.1.4. Actualizar datos

```
UPDATE <nombre_tabla>  
SET <campo> = <valor_nuevo_para_el_campo>  
[WHERE <condición>];
```

donde:

nombre_tabla ⇒ especifica el nombre de la tabla de la base de datos.

campo ⇒ especifica el campo de la tabla a modificar.

valor_nuevo_para_el_campo ⇒ es el valor que tomará el campo que se va a modificar.

Puede contener expresiones, constantes y variables PL/SQL. *condición* ⇒ esta compuesta por nombres de campos, expresiones, constantes y operadores de comparación, incluyendo constantes y variables PL/SQL.

2.2. Ejemplos órdenes de manipulación de datos

Employees

Campo	Tipo de dato
employee_id	NUMBER(6)
first_name	VARCHAR2(20)
last_name	VARCHAR2(25)
email	VARCHAR2(25)
phone_number	VARCHAR2(20)
hire_date	DATE
job_id	VARCHAR2(10)
salary	NUMBER(8,2)
commission_pct	NUMBER(2,2)
manager_id	NUMBER(6)
department_id	NUMBER(4)

Departments

Campo	Tipo de dato
department_id	NUMBER(4)
department_name	VARCHAR2(30)
manager_id	NUMBER(6)
location_id	NUMBER(4)

Ejemplo 1:

Crear un script que recupere el código, nombre, trabajo y salario del empleado que lleva por código el 100.

```
SET SERVEROUTPUT ON
SET VERIFY OFF
SET ECHO OFF
DECLARE
    var_numero employees.employee_id%TYPE;
    var_nombre employees.first_name%TYPE;
    var_trabajo employees.job_id%TYPE;
    var_salario employees.salary%TYPE;
BEGIN
    SELECT employee_id, first_name, job_id,salary
    INTO var_numero, var_nombre, var_trabajo,
        var_salario
    FROM employees
    WHERE employee_id = 100;
    DBMS_OUTPUT.PUT_LINE('Los datos del empleado son: ');
    DBMS_OUTPUT.PUT_LINE('CODIGO: ' || var_numero);
    DBMS_OUTPUT.PUT_LINE('NOMBRE: ' || var_nombre);
    DBMS_OUTPUT.PUT_LINE('TRABAJO: ' || var_trabajo);
    DBMS_OUTPUT.PUT_LINE('SALARIO: ' || var_salario);
END;
/
```

Los datos del empleado son:

CODIGO: 100

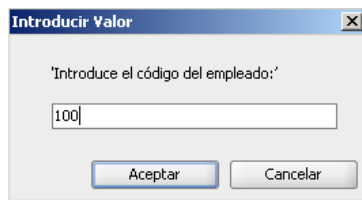
NOMBRE: Steven

TRABAJO: AD_PRES

SALARIO: 24000

Ejemplo 2:

Crear un script que recupere el código, nombre, trabajo y salario del empleado cuyo código se introduce por teclado.



```
Los datos del empleado son:
CODIGO: 100
NOMBRE: Steven
TRABAJO: AD_PRES
SALARIO: 24000
```

```
ACCEPT codigo PROMPT 'Introduce el código del
empleado: '
SET SERVEROUTPUT ON
SET VERIFY OFF
SET ECHO OFF
DECLARE
    var_numero employees.employee_id%TYPE;
    var_nombre employees.first_name%TYPE;
    var_trabajo employees.job_id%TYPE;
    var_salario employees.salary%TYPE;
BEGIN
    SELECT employee_id, first_name, job_id,salary
    INTO var_numero, var_nombre, var_trabajo, var_salario
    FROM employees
    WHERE employee_id = &codigo;
    DBMS_OUTPUT.PUT_LINE('Los datos del empleado son: ');
    DBMS_OUTPUT.PUT_LINE('CODIGO: ' || var_numero);
    DBMS_OUTPUT.PUT_LINE('NOMBRE: ' || var_nombre);
    DBMS_OUTPUT.PUT_LINE('TRABAJO: ' || var_trabajo);
    DBMS_OUTPUT.PUT_LINE('SALARIO: ' || var_salario);
END;
```

Desenvolupament d'Aplicacions Multiplataforma – M02 Bases de dades

UF3: Llenguatges SQL: DCL i extensió procedimental - NF2: Procediments i funcions. Ordres per manipular les dades -

EA 3.2.2. Ordres de manipulació

Versió 1.0 - © M^a Carmen Brito

Ejemplo 3:

Crear un script que introduzca un empleado en la tabla correspondiente

Campo	Valor
employee_id	1111
first_name	Maria
last_name	González
email	mgonz@gmail.com
phone_number	432311111
hire_date	10/10/2000
job_id	AD_PRES
salary	1243
commission_pct	
manager_id	
department_id	10

```
SET SERVEROUTPUT ON
SET ECHO OFF
SET VERIFY OFF
BEGIN
    INSERT INTO employees
    VALUES (1111,'Maria', 'Gonzalez',
    'mgonz@gmail.com', 432311111
    , '10/10/2000',
    'AD_PRES' ,1243,NULL,NULL,10);
    DBMS_OUTPUT.PUT_LINE ( 'INSERTADO
CORRECTAMENTE' );
END;
/
```

Ejemplo 4:

Crear un script que actualice los empleados cuyo código sea el 1111, aumentando en 200 la comisión.

```
SET SERVEROUTPUT ON
SET ECHO OFF
SET VERIFY OFF
BEGIN
    UPDATE employees
    SET commission_pct = commission_pct + 200
    WHERE employee_id = 1111;
    DBMS_OUTPUT.PUT_LINE ( 'MODIFICACION CORRECTA' );
END;
/
```

Ejemplo 5:

Crear un script que elimine el registro que acabamos de introducir, cuyo código de empleado es el 1111.

```
SET SERVEROUTPUT ON
SET ECHO OFF
SET VERIFY OFF
BEGIN
    DELETE FROM employees
    WHERE employee_id = 1111;
    DBMS_OUTPUT.PUT_LINE ( 'ELIMINADO CORRECTAMENTE' );
END;
/
```

Preguntes!!!!



Desenvolupament d'Aplicacions Multiplataforma – M02 Bases de dades

UF3: Llenguatges SQL: DCL i extensió procedimental - NF2: Procediments i funcions. Ordres per manipular les dades -

EA 3.2.2. Ordres de manipulació

Versió 1.0 - © M^a Carmen Brito