



2. Sentències per manipular les dades

RA5.Desenvolupa procediments emmagatzemats avaluant i utilitzant les sentències del llenguatge incorporat al sistema gestor de bases de dades.

1. Órdenes de manipulación de datos

- Consultar datos
- Eliminar datos
- Insertar datos
- Actualizar datos

2. Ejemplos

1. Ordenes de manipulació de dades

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

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

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

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

1. Ordenes de manipulación de datos

- 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.

1. Ordenes de manipulación de datos

- 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.

1. Ordenes de manipulación de datos

- 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.

1. Ordenes de manipulación de datos

- Actualizar datos

```
UPDATE <nombre_tabla>  
SET <campo> = <valor_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_campo es el nuevo 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. Ejemplos

- Ejemplo 1

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

Los datos del empleado son:

CODIGO: 100

NOMBRE: Steven

TRABAJO: AD_PRES

SALARIO: 24000

```
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;
/
```


2. Ejemplos

- Ejemplo 2

Crear un script que recupere el código, nombre, trabajo y salario del empleado que 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;  
/
```

2. Ejemplos

- Ejemplo 3, insert , update, delete

```
DECLARE
  var_emp  employees%ROWTYPE;
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');

  UPDATE employees
  SET commission_pct = commission_pct + 200
  WHERE employee_id = 1111;
  DBMS_OUTPUT.PUT_LINE ('MODIFICACION CORRECTA');

  DELETE FROM employees
  WHERE employee_id = 1111;
  DBMS_OUTPUT.PUT_LINE ('ELIMINADO CORRECTAMENTE');

END;
/
```

Preguntes!!!!

