

# Práctica 5. Manipulación de datos en PL/SQL y PL/SQL como lenguaje de programación (lógica condicional, bucles)

Profesor: José M. Luna

Email: [jmluna@uco.es](mailto:jmluna@uco.es)

# Índice

- Sentencias condicionales
- Bucles
- Cursores

# Índice

- **Sentencias condicionales**
- Bucles
- Cursores

# Sentencias condicionales: CASE

- La sentencia CASE en PL/SQL es similar a una sentencia IF-THEN-ELSE

CASE [expresión]

    WHEN condicion1 THEN resultado1

    WHEN condicion2 THEN resultado2

    ...

    ELSE result

END CASE;

# Sentencias condicionales: CASE

- **Ejemplo** de un CASE en un **SELECT**

```
SELECT CASE estudiosuperiores
        WHEN 'Ninguno' THEN 'sin estudios'
        ELSE 'con estudios'
END FROM votantes;
```

- **Ejemplo** de un CASE en un **bloque**

```
DECLARE
    v_nombre votantes.nombrecompleto%type;
    mydni NUMBER := 30983712;
    myoutput VARCHAR2(15);
BEGIN
    SELECT nombrecompleto INTO v_nombre FROM votantes where dni=mydni;
    CASE
    WHEN length(v_nombre) > 10 THEN myoutput := 'Nombre corto';
    ELSE myoutput := 'Nombre Largo';
    END CASE;
    dbms_output.put_line(myoutput);
END;
```

# Sentencias condicionales: IF-THEN

- **Definición:**

```
IF condition1 THEN statement1;  
ELSIF condition2 THEN statement2;  
ELSIF condition3 THEN statement3;  
END IF;
```

- **Ejemplo** de un IF-THEN en un **bloque**

```
DECLARE  
    v_nombre votantes.nombrecompleto%type;  
    mydni NUMBER := 30983712;  
    myoutput VARCHAR2(15);  
BEGIN  
    SELECT nombrecompleto INTO v_nombre FROM votantes where dni=mydni;  
    IF length(v_nombre) > 10 THEN myoutput := 'Nombre corto';  
    ELSE myoutput := 'Nombre Largo';  
    END IF;  
    dbms_output.put_line(myoutput);  
END;
```

# Índice

- Sentencias condicionales
- **Bucles**
- Cursores

# Bucles

- **LOOP and EXIT**

```
DECLARE
    n NUMBER := 1;
BEGIN
    LOOP
        n := n + 1;
        EXIT WHEN n > 10;
    END LOOP;
END;
```



# Bucles

## FOR (incremental)

```
set serveroutput on;  
DECLARE  
    num NUMBER;  
BEGIN  
    FOR num IN 1..3 LOOP  
        dbms_output.put_line(num);  
    END LOOP;  
END;
```

## FOR (decremental)

```
set serveroutput on;  
DECLARE  
    num NUMBER;  
BEGIN  
    FOR num IN REVERSE 1..3 LOOP  
        dbms_output.put_line(num);  
    END LOOP;  
END;
```

# Bucles

## WHILE

```
set serveroutput on;  
DECLARE  
    num NUMBER := 1;  
BEGIN  
    WHILE num<= 3 LOOP  
        dbms_output.put_line(num);  
        num := num + 1;  
    END LOOP;  
END;
```

# Índice

- Sentencias condicionales
- Bucles
- **Cursores**

# Cursores

- Creamos un cursor cuando queremos recorrer el resultado de una consulta

```
set serveroutput on;  
DECLARE  
  CURSOR c IS SELECT * FROM votantes;  
  votantesData votantes%ROWTYPE;  
BEGIN  
  open c;  
  LOOP  
    FETCH c INTO votantesData;  
    EXIT WHEN c%NOTFOUND;  
    dbms_output.put_line(votantesData.nombrecompleto);  
  END LOOP;  
END;
```

# Cursores

- **Atributos útiles de los cursores:**
  - **SQL%ROWCOUNT:** Número de filas afectadas por la sentencia SQL (valor entero)
  - **SQL%FOUND:** TRUE si la sentencia afecta a uno o más registros
  - **SQL%NOTFOUND:** TRUE si la sentencia no afecta a ningún registro

# Cursores

- **Bucle FOR para cursores:**

```
set serveroutput on;  
DECLARE  
  CURSOR c IS SELECT * FROM votantes;  
BEGIN  
  FOR num_row IN c LOOP  
    dbms_output.put_line(num_row.nombrecompleto);  
  END LOOP;  
END;
```