

# 4- ESTRUCTURES DE CONTROL DE FLUX

## Contenido

---

1.	Estructures condicionals IF-ELSIF .....	2
2.	Sentència CASE.....	4
	SEARCHED CASE statements .....	4
	EXPRESSIONS CASE: assignar valor a una variable .....	5
3.	Bucle LOOP (obsolet).....	6
4.	Bucle WHILE .....	7
5.	Bucle FOR .....	8

# 1. Estructures condicionals IF-ELSIF

---

## SINTAXI

---

```
IF condició THEN
    sentències;
[ELSIF condició THEN
    sentències;
    ... ]

[ELSE
    sentències]
END IF;
```

- ✓ **ELSE** del final s'avalua si cap de les condicions anteriors és certa

## EXEMPLE

---

Bloc de codi que declara una variable edat i mostra diferents missatge en funció de l'edat.

```
DECLARE
    v_edat NUMBER;
BEGIN
    v_edat := 5;
    IF v_edat <=10 THEN
        DBMS_OUTPUT.PUT_LINE('Ets un nen.');
```

```
    ELSIF v_edat>10 AND v_edat <=20 THEN
        DBMS_OUTPUT.PUT_LINE('Uf ... adolescent.');
```

```
    ELSE
        DBMS_OUTPUT.PUT_LINE('Ja tenim una edat.');
```

```
    END IF;
END;
```

## VALOR NULL EN LES CONDICIONS

---

- ✓ Una comparació en la que alguna de les parts és NULL s'avalua a NULL.
- ✓ Una condició NULL és **falsa**. En realitat, es considerada no certa.

## EXEMPLE

Modifiquem l'exemple de l'edat eliminat l'assignació a la variable v\_edat:

```
DECLARE
    v_edat NUMBER;
BEGIN
    IF v_edat <=10 THEN
        DBMS_OUTPUT.PUT_LINE('Ets un nen.');
```

```
    ELSIF v_edat>10 AND v_edat <=20 THEN
        DBMS_OUTPUT.PUT_LINE('Uf ... adolescent.');
```

```
    ELSE
        DBMS_OUTPUT.PUT_LINE('Ja tenim una edat.');
```

```
    ENDI IF;
```

Com la variable **v\_edat** no ha estat inicialitzada, la comparació de l'IF retorna NULL (què no és ni TRUE ni FALSE). Com no és TRUE, s'avalua la part de l'ELSE.

```
Ja tenim una edat.  
Statement processed.
```

## EXEMPLE

```
DECLARE  
  a NUMBER;  
  b NUMBER;  
BEGIN  
  a:= NULL;  
  b:= NULL;  
  IF a=b THEN  
    DBMS_OUTPUT.PUT_LINE('a i b són iguals.');  ELSE  
    DBMS_OUTPUT.PUT_LINE('a i b són diferents.');  END IF;  
END;
```

Es comparen dos valors NULL. Com el valor NULL indica valor desconegut, comparar qualsevol valor amb NULL és NULL (comparar amb un valor desconegut sempre és desconegut), encara que l'altre valor també sigui NULL.

```
a i b són diferents.  
Statement processed.
```

## 2. Sentència CASE

- ✓ Representen una seqüència de IF- ELSEIF. Més fàcil d'escriure i d'entendre.
- ✓ Habitualment utilitzat per a comparar varis valors d'una mateixa variable

```
DECLARE
  v_numvar    NUMBER;
BEGIN
  ...
  CASE v_numvar
    WHEN 5 THEN statement_1; statement_2;
    WHEN 10 THEN statement_3;
    WHEN 12 THEN statement_4; statement_5;
    WHEN 27 THEN statement_6;
    WHEN ... - and so on
    ELSE statement_15;
  END CASE;
  ...
END;
```

### SEARCHED CASE statements

Exemple que demana l'id d'empleat amb una **finestra emergent**. Consulta el cognom i el salari de l'empleat i obté la categoria segons els valors de la taula **JOB\_GRADES**. Mostra el cognom, el salari i la categoria de l'id introduït:

```
DECLARE
  -- Declara les variables
  v_salari    EMPLOYEES.SALARY%TYPE;
  v_nom       EMPLOYEES.LAST_NAME%TYPE;
  v_id        EMPLOYEES.EMPLOYEE_ID%TYPE;
  v_categoria  JOB_GRADES.GRADE_LEVEL%TYPE;
BEGIN
  -- Assigna ide de l'empleat
  v_id := 100;

  -- Consulta nom i salari
  SELECT LAST_NAME, SALARY
  INTO v_nom, v_salari
  FROM EMPLOYEES
  WHERE EMPLOYEE_ID = v_id;

  -- Obté la categoria comparant els valors de la taula JOB_GRADES
  CASE
    WHEN v_salari BETWEEN 1000 AND 2999 THEN v_categoria := 'A';
    WHEN v_salari BETWEEN 3000 AND 5999 THEN v_categoria := 'B';
    WHEN v_salari BETWEEN 6000 AND 9999 THEN v_categoria := 'C';
    WHEN v_salari BETWEEN 10000 AND 14999 THEN v_categoria := 'D';
    WHEN v_salari BETWEEN 15000 AND 24999 THEN v_categoria := 'E';
    WHEN v_salari BETWEEN 25000 AND 40000 THEN v_categoria := 'F';
    ELSE v_categoria := 'SENSE CATEGORIA';
  END CASE;

  DBMS_OUTPUT.PUT_LINE(v_nom || ' cobra ' || v_salari || '- Categoria ' || v_categoria);
END;
```

```
King cobra 24000-- Categoria E
Statement processed.
```

## EXPRESSIONS CASE: assignar valor a una variable

Expressions CASE que assignen un valor a una variable en funció del valor d'una altra variable.

Modifiquem la sintaxi de l'exemple anterior per assignar valor a la variable categoria en funció del salari:

```
DECLARE
    v_salari    EMPLOYEES.SALARY%TYPE;
    v_nom       EMPLOYEES.LAST_NAME%TYPE;
    v_id        EMPLOYEES.EMPLOYEE_ID%TYPE;
    v_categoria  JOB_GRADES.GRADE_LEVEL%TYPE;
BEGIN
    -- Demana id de l'empleat
    v_id := :id;

    -- Consulta nom i salari
    SELECT LAST_NAME, SALARY
    INTO v_nom, v_salari
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = v_id;
    -- Obté la categoria comparant els valors de la taula JOB_GRADES
    CASE
        v_categoria := CASE
            WHEN v_salari BETWEEN 1000 AND 2999 THEN 'A'
            WHEN v_salari BETWEEN 3000 AND 5999 THEN 'B'
            WHEN v_salari BETWEEN 6000 AND 9999 THEN 'C'
            WHEN v_salari BETWEEN 10000 AND 14999 THEN 'D'
            WHEN v_salari BETWEEN 15000 AND 24999 THEN 'E'
            WHEN v_salari BETWEEN 25000 AND 40000 THEN 'F'
            ELSE 'SENSE CATEGORIA'
        END;

    DBMS_OUTPUT.PUT_LINE(v_nom || ' cobra ' || v_salari || '-
    - Categoria ' || v_categoria);
END;
```

No acaben amb ;

Finalitza amb un END,  
no amb un END CASE !!!!

Amb id 100:

```
King cobra 24000-- Categoria E
Statement processed.
```

### 3. Bucle LOOP (obsolet)

#### SINTAXI

```
LOOP
  [sentències;]
  EXIT WHEN condició;
  [sentències]
END LOOP;
```

```
LOOP
  [sentències;]
  IF condició THEN EXIT;
  END IF;
  [sentències]
END LOOP;
```

#### EXEMPLE

Bucle que escriu els números de l'1 fins al 10.

```
DECLARE
  i NUMBER;
BEGIN
  i:=1;
  LOOP
    DBMS_OUTPUT.PUT_LINE(i);
    i:=i+1;
    EXIT WHEN i>5;
  END LOOP;
END;
```

```
DECLARE
  i NUMBER;
BEGIN
  i:=1;
  LOOP
    DBMS_OUTPUT.PUT_LINE(i);
    i:=i+1;
    IF i>5 THEN EXIT;
    END IF;
  END LOOP;
END;
```

```
1
2
3
4
5

Statement processed.
```

## 4.Bucle WHILE

---

### SINTAXI

---

```
WHILE condició LOOP
    sentències;
END LOOP;
```

El bucle finalitza quan la condició és falsa o NULL.

### EXEMPLE

---

Bucle que escriu els números de l'1 fins al 10.

```
DECLARE
    i NUMBER;
BEGIN
    i:=1;
    WHILE i<=5 LOOP
        DBMS_OUTPUT.PUT_LINE(i);
        i:=i+1;
    END LOOP;
END;
```

```
1
2
3
4
5
Statement processed.
```

## 5.Bucle FOR

### SINTAXI

```
FOR variable IN [REVERSE] valor_inicial..valor_final LOOP
    sentències;
END LOOP;
```

- ✓ No s'ha de declarar al variable del bucle, **es declara implícitament**. És un enter, que pren per valor el valor\_inicial, al final de cada iteració del bucle s'incrementa amb 1, fins que té el valor\_final
- ✓ **REVERSE** fa que el comptador comenci des del valor\_final fins al valor\_inicial
- ✓ L'àmbit del comptador és només el FOR. No es pot referenciar fora del FOR.
- ✓ Ni el valor\_inicial ni el valor\_final poden ser NULL.
- ✓ El valor\_inicial i el valor final poden ser expressions.

### EXEMPLE

<pre>BEGIN     FOR i IN 1..5 LOOP         DBMS_OUTPUT.PUT_LINE('i');     END LOOP; END;</pre>	<pre>BEGIN     FOR i IN REVERSE 1..5 LOOP         DBMS_OUTPUT.PUT_LINE('i');     END LOOP; END;</pre>
<pre>1 2 3 4 5  Statement processed.</pre>	<pre>5 4 3 2 1  Statement processed.</pre>