

# Aula 11

## Introdução

### Programação PL/SQL Estruturas de Controle

Banco de Dados - Computação - UNISUL

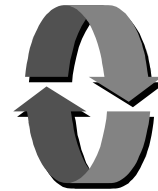


## Estruturas de Controle

# Controlando fluxo de execução do PL/SQL

## Comando Condicional IF:

- IF-THEN-END IF
- IF-THEN-ELSE-END IF
- IF-THEN-ELSIF-END IF



## Comandos IF

### Sintaxe

```
IF condição THEN  
    comandos;  
[ELSIF condição THEN  
    comandos;]  
[ELSE  
    comandos;]  
END IF;
```

### IF simples:

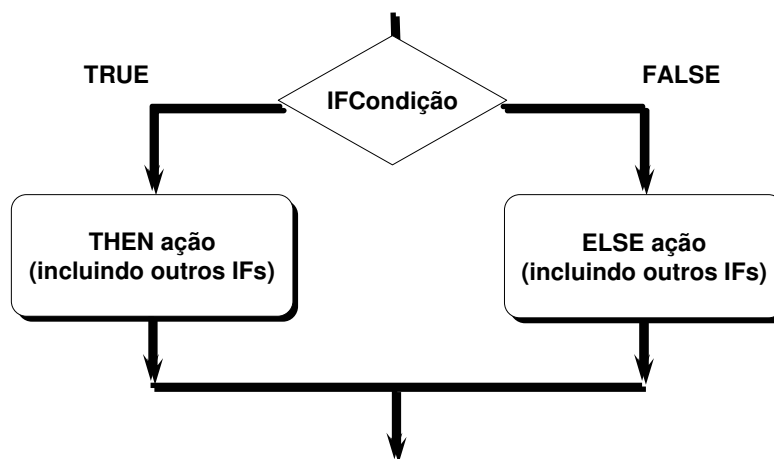
```
IF v_ename = 'OSBORNE' THEN  
    v_mgr := 22;  
END IF;
```

# IF simples

## Exemplo

```
. . .  
IF v_ename = 'MILLER' THEN  
  v_job := 'SALESMAN';  
  v_deptno := 35;  
  v_new_comm := sal * 0.20;  
END IF;  
. . .
```

## Comando IF-THEN-ELSE

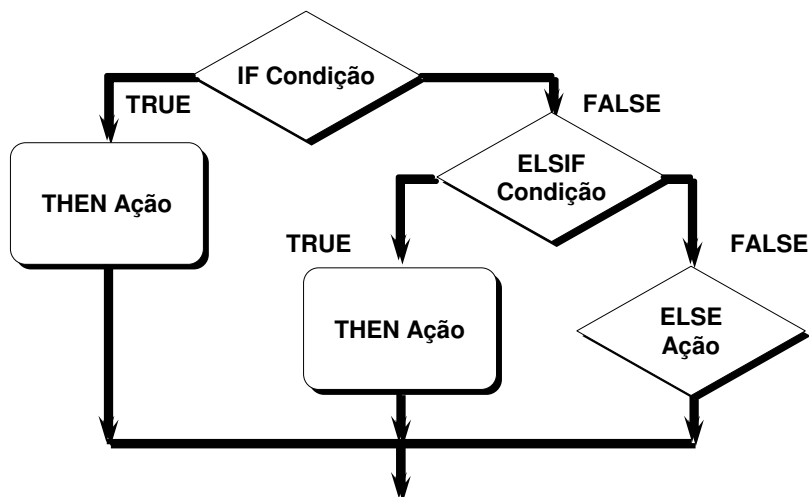


# Comando IF-THEN-ELSE

## Exemplo

```
...  
IF v_shipdate - v_orderdate < 5 THEN  
    v_ship_flag := 'Acceptable';  
ELSE  
    v_ship_flag := 'Unacceptable';  
END IF;  
...
```

# Comando IF-THEN-ELSIF



## Comando IF-THEN-ELSIF

### Exemplo

```
. . .  
IF v_start > 100 THEN  
    v_start := 2 * v_start;  
ELSIF v_start >= 50 THEN  
    v_start := .5 * v_start;  
ELSE  
    v_start := .1 * v_start;  
END IF;  
. . .
```

## Construindo Condições Lógicas

- Podemos manipular valores nulos com operador IS NULL.
- Alguma expressão que contenha um valor nulo resulta NULL.
- Concatenando valores nulos o resultado final será um *string* vazio

## Tabela Lógica

AND	TRUE	FALSE	NULL	OR	TRUE	FALSE	NULL	NOT	
TRUE	TRUE	FALSE	NULL	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	NULL	FALSE	TRUE
NULL	NULL	FALSE	NULL	NULL	TRUE	NULL	NULL	NULL	NULL

11

Banco de Dados – Computação - UNISUL

## Condições Booleanas

Qual o valor de V\_FLAG em cada caso?

```
v_flag := v_reorder_flag AND v_available_flag;
```

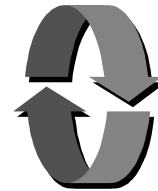
V_REORDER_FLAG	V_AVAILABLE_FLAG	V_FLAG
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
NULL	TRUE	NULL
NULL	FALSE	FALSE

12

Banco de Dados – Computação - UNISUL

## Comandos de Repetição: Comando LOOP

- Loop faz com que seja repetido um comando ou seqüência de comandos múltiplas vezes.
- Existem três tipos de loop:
  - Basic loop
  - FOR loop
  - WHILE loop



## Basic Loop

### Sintaxe

```
LOOP                                -- delimitador
  comando1;                        -- comandos
  . . .
  EXIT [WHEN condição];           -- comando EXIT
END LOOP;                          -- delimitador
```

onde:      *condição*              é uma variável ou expressão Boolean (TRUE, FALSE, ou NULL);

# Basic Loop

## Exemplo

```
DECLARE
  v_ordid    item.ordid%TYPE := 101;
  v_counter  NUMBER(2) := 1;
BEGIN
  LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_ordid, v_counter);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 10;
  END LOOP;
END;
```

# FOR Loop

## Sintaxe

```
FOR contador in [REVERSE]
  de..até LOOP
  comando1;
  comando2;
  . . .
END LOOP;
```

- índice não é declarado, está implícito.



## FOR Loop

Inserir as primeiras 10 linhas de itens na ordem 101.

### Exemplo

```
DECLARE
  v_ordid    item.ordid%TYPE := 101;
BEGIN
  FOR i IN 1..10 LOOP
    INSERT INTO item(ordid, itemid)
      VALUES(v_ordid, i);
  END LOOP;
END;
```

## WHILE Loop

### Sintaxe

```
WHILE condição LOOP ← Condição é
  comando1;           avaliada no
  comando2;           começo de
  . . .               cada iteração
END LOOP;
```

**WHILE loop é usado para repetir comandos quando uma condição é TRUE.**

# WHILE Loop

## Exemplo

```
ACCEPT p_price PROMPT 'Entre com o preço do item: '
ACCEPT p_itemtot PROMPT 'Entre com o valor máximo a
                        ser gasto: '

DECLARE
...
v_qty          NUMBER(8) := 1;
v_running_total NUMBER(7,2) := 0;
BEGIN
...
  WHILE v_running_total < &p_itemtot LOOP
    ...
    v_qty := v_qty + 1;
    v_running_total := v_qty * &p_price;
  END LOOP;
...

```

# Ninho de Loops

```
...
BEGIN
LOOP
  v_counter := v_counter+1;
  EXIT WHEN v_counter>10;
  LOOP
    ...
    EXIT loop_externo WHEN total_done = 'YES';
    -- sai de ambos os loops
    EXIT loop_interno WHEN total_done = 'YES';
    -- sai somente do loop interno
    ...
  END LOOP;
  ...
END LOOP;
END;
```

## **Resumo**

**Mudando o fluxo do programa usando estruturas de controle.**

- **Condicional (comando IF)**
- **Repetição**
  - **Basic loop**
  - **FOR loop**
  - **WHILE loop**
  - **comando EXIT**