



# Stored Procedures: Funções em Linguagem SQL



**Professor**

Fernando Martins Medeiros

# Estruturas condicionais

## IF ... THEN

*As instruções IF-THEN são a forma mais simples de IF. As instruções entre o THEN e o END IF são executadas se a condição for verdade. Senão, são saltadas.*

Sintaxe:

```
IF expressão_booleana THEN
    instruções
END IF;
```

Exemplo:

```
IF v_id_usuario <> 0 THEN
    UPDATE usuarios SET email = v_email WHERE id_usuario = v_id_usuario;
END IF;
```

# Estruturas condicionais

## IF ... THEN ... ELSE

*s instruções IF-THEN-ELSE ampliam o IF-THEN permitindo especificar um conjunto alternativo de instruções a serem executadas se a condição for avaliada como falsa.*

Sintaxe:

```
IF expressão_booleana THEN
    instruções
ELSE
    instruções
END IF;
```

Exemplo:

```
IF v_contador > 0 THEN
    INSERT INTO contador_de_usuários (contador) VALUES (v_contador);
    RETURN 't';
ELSE
    RETURN 'f';
END IF;
```

# Estruturas condicionais

## IF ... THEN ... ELSE IF

*As instruções IF podem ser aninhadas, quando esta forma é utilizada uma instrução IF está sendo aninhada dentro da parte ELSE da instrução IF externa. Portanto, há necessidade de uma instrução END IF para cada IF aninhado, mais um para o IF-ELSE pai.*

Sintaxe:

```
IF expressão_booleana THEN
    instruções
[ ELSIF expressão_booleana THEN
    instruções
END IF;
END IF;
```

Exemplo:

```
IF linha_demo.sexo = 'm' THEN
    sexo_extenso := 'masculino';
ELSE
    IF linha_demo.sexo = 'f' THEN
        sexo_extenso := 'feminino';
    END IF;
END IF;
```

# Estruturas condicionais

## IF ... THEN ... ELSIF ... THEN ... ELSE

A instrução IF-THEN-ELSIF-ELSE fornece um método mais conveniente para verificar muitas alternativas em uma instrução. Formalmente equivale aos comandos IF-THEN-ELSE-IF-THEN aninhados, mas somente necessita de um END IF. *ELSEIF é um aliás para ELSIF.*

Sintaxe:

```
IF expressão_booleana THEN
    instruções
[ ELSIF expressão_booleana THEN
    instruções
[ ELSIF expressão_booleana THEN
    instruções
    ...]]
[ ELSE
    instruções ]
END IF;
```

Exemplo:

```
IF numero = 0 THEN
    resultado := 'zero';
ELSIF numero > 0 THEN
    resultado := 'positivo';
ELSIF numero < 0 THEN
    resultado := 'negativo';
ELSE
    -- hmm, a única outra possibilidade é que o número seja nulo
    resultado := 'NULL';
END IF;
```

# Estruturas condicionais

## Laços simples

A instrução `LOOP` define um laço incondicional, repetido indefinidamente até ser terminado por uma instrução `EXIT` ou `RETURN`. Nos laços aninhados pode ser utilizado um rótulo opcional na instrução `EXIT` para especificar o nível de aninhamento que deve ser terminado.

Sintaxe:

```
[<<rótulo>>]  
LOOP  
    instruções  
END LOOP;
```

```
EXIT [ rótulo ] [ WHEN expressão ];
```

Exemplo:

```
LOOP  
    -- algum processamento  
    IF contador > 0 THEN  
        EXIT; -- sair do laço  
    END IF;  
END LOOP;  
  
LOOP  
    -- algum processamento  
    EXIT WHEN contador > 0; -- mesmo resultado do exemplo acima  
END LOOP;  
  
BEGIN  
    -- algum processamento  
    IF estoque > 100000 THEN  
        EXIT; -- causa a saída do bloco BEGIN  
    END IF;  
END;
```

# Estruturas condicionais

## WHILE

A instrução *WHILE* repete uma seqüência de instruções enquanto a expressão de condição for avaliada como verdade. A condição é verificada logo antes de cada entrada no corpo do laço.

Sintaxe:

```
[<<rótulo>>]  
WHILE expressão LOOP  
    instruções  
END LOOP;
```

Exemplo:

```
WHILE quantia_devida > 0 AND saldo_do_certificado_de_bonus > 0 LOOP  
    -- algum processamento  
END LOOP;  
  
WHILE NOT expressão_booleana LOOP  
    -- algum processamento  
END LOOP;
```

# Estruturas condicionais

## FOR

A instrução *WHILE* repete uma seqüência de instruções enquanto a expressão de condição for avaliada como verdade. A condição é verificada logo antes de cada entrada no corpo do laço.

Sintaxe:

```
[<<rótulo>>]  
FOR nome IN [ REVERSE ] expressão .. expressão LOOP  
    instruções  
END LOOP;
```

Exemplo:

```
FOR i IN 1..10 LOOP  
    -- algum processamento  
    RAISE NOTICE 'i é %', i;  
END LOOP;
```

```
FOR i IN REVERSE 10..1 LOOP  
    -- algum processamento  
END LOOP;
```