

# **Arquitetura e Desempenho de Bancos de Dados**

## **Funções em SQL e Funções em PL/pgSQL - Particularidades**

---

# Funções

## Funções em Linguagem Procedural

Exemplo:

```
CREATE OR REPLACE FUNCTION exemplo (codFunc integer)
RETURNS varchar AS $$
DECLARE
    nome varchar;
BEGIN
    SELECT INTO nome lower(nome) FROM funcionarios
    WHERE codigo = codFunc;
END;
$$ LANGUAGE plpgsql;
```

# Funções Definidas pelo Usuário

## Funções retornando conjunto - SETOF

```
CREATE FUNCTION get_amigo(int)  
RETURNS SETOF amigos AS $$  
    SELECT *  
    FROM amigos WHERE codigo = $1;  
$$ LANGUAGE SQL;
```

**//executando agora a função acima:**

```
SELECT * FROM get_amigo(1);
```

# Funções Definidas pelo Usuário

## IF

**A linguagem PL/pgSQL possui cinco formas de IF:**

- IF ... THEN
- IF ... THEN ... ELSE
- IF ... THEN ... ELSE IF
- IF ... THEN ... ELSIF ... THEN ... ELSE
- IF ... THEN ... ELSEIF ... THEN ... ELSE

# Funções Definidas pelo Usuário

## IF

**Veja sintaxe e exemplos em:**

**<http://pgdocptbr.sourceforge.net/pg80/plpgsql-control-structures.html#PLPGSQL-CONDITIONALS>**

# Funções Definidas pelo Usuário

## CASE

```
1 CASE
2     WHEN condition_1 THEN result_1
3     WHEN condition_2 THEN result_2
4     [WHEN ...]
5     [ELSE result_n]
6 END
```

# Funções Definidas pelo Usuário

## CASE

### 1º Exemplo

Cenário: Serão distribuídos uniformes para times de futebol, de acordo com a seguinte regra:

- O time "A" receberá uniforme da cor "Amarela";
- O time "B" receberá uniforme da cor "Azul";
- O time "C" receberá uniforme da cor "Vermelha";

Podemos ver a imagem da tabela "time\_futebol":

	time character varying(1)
1	A
2	B
3	C

# Funções Definidas pelo Usuário

## CASE

```
SELECT time, CASE  
    WHEN time = 'A' THEN 'Amarela'  
    WHEN time = 'B' THEN 'Azul'  
    WHEN time = 'C' THEN 'Vermelha'  
    END AS cor_uniforme  
FROM time_futebol;
```



# Funções Definidas pelo Usuário

## CASE

Após a execução poderemos ver o resultado da distribuição das cores dos uniformes na imagem abaixo:

	<b>time character varying(1)</b>	<b>cor_uniforme text</b>
<b>1</b>	A	Amarela
<b>2</b>	B	Azul
<b>3</b>	C	Vermelha

# Funções Definidas pelo Usuário

## Laços

```
WHILE
```

```
  [<<rótulo>>]
```

```
  WHILE expressão LOOP
```

```
    instruções
```

```
  END LOOP;
```

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.

# Funções Definidas pelo Usuário

## WHILE - EXEMPLO

```
WHILE NOT expressão_booleana LOOP  
    -- algum processamento  
END LOOP;
```

### Exemplo:

```
WHILE quantia_devida > 0 AND saldo_do_certificado_de_bonus > 0 LOOP  
    -- algum processamento  
END LOOP;
```

# Funções Definidas pelo Usuário

## Laços

### FOR (variação inteira)

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

Esta forma do FOR cria um laço que interage num intervalo de valores inteiros. A variável nome é definida automaticamente como sendo do tipo integer, e somente existe dentro do laço. As duas expressões que fornecem o limite inferior e superior do intervalo são avaliadas somente uma vez, ao entrar no laço. Normalmente o passo da interação é 1.

# Funções Definidas pelo Usuário

## FOR - Exemplos

Alguns exemplos de laços FOR inteiros:

```
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;
```

Se o limite inferior for maior do que o limite superior, o corpo do laço não é executado nenhuma vez. Nenhum erro é gerado.