

# Tema: Fundamentos e Blocos Anônimos

---

## Objetivos de Aprendizagem

Ao final deste encontro, o aluno deverá ser capaz de:

- Compreender o que é PL/SQL e suas vantagens.
  - Estruturar e executar blocos PL/SQL anônimos.
  - Declarar variáveis e constantes.
  - Utilizar estruturas condicionais e de repetição.
  - Exibir informações com DBMS\_OUTPUT.
- 

## 1. Introdução ao PL/SQL

**PL/SQL (Procedural Language/SQL)** é a linguagem procedural da Oracle, que **combina o poder do SQL com estruturas de programação**, permitindo:

- Controle de fluxo (condições e laços);
  - Declaração de variáveis e constantes;
  - Tratamento de exceções (erros);
  - Criação de *procedures*, *functions* e *triggers*;
  - Execução de blocos anônimos diretamente no SQL Developer ou SQL\*Plus.
- 

## Estrutura de um Bloco PL/SQL

```
DECLARE
    -- Declaração de variáveis, constantes, cursores, etc.
BEGIN
    -- Comandos executáveis (SQL e PL/SQL)
EXCEPTION
```

```
-- Tratamento de erros (opcional)
END;
```

#### ♦ **Observação:**

Cada bloco PL/SQL termina com END; e deve conter pelo menos a parte BEGIN ... END;.  
O DECLARE e o EXCEPTION são opcionais.

---

## 2. Declarando Variáveis e Constantes

```
DECLARE
    v_nome      VARCHAR2(30);
    v_idade      NUMBER := 25;
    c_taxa       CONSTANT NUMBER := 0.05;
BEGIN
    v_nome := 'Vilson';
    DBMS_OUTPUT.PUT_LINE('Nome: ' || v_nome);
    DBMS_OUTPUT.PUT_LINE('Idade: ' || v_idade);
    DBMS_OUTPUT.PUT_LINE('Taxa: ' || c_taxa);
END;
```

#### ♦ **Tipos mais usados:**

- VARCHAR2(tamanho) — texto
- NUMBER(p, s) — números
- DATE — datas
- BOOLEAN — valores lógicos (TRUE, FALSE, NULL)

#### ♦ **Atribuição:**

- Direta: v\_nome := 'Ana';

Em SELECT:

```
SELECT salario INTO v_salario FROM empregados WHERE id = 1;
```

- 

---

### 3. Uso do DBMS\_OUTPUT

O pacote DBMS\_OUTPUT serve para exibir informações no console.

Antes de usar, ative a saída no SQL Developer:

```
SET SERVEROUTPUT ON
```

Exemplo:

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Olá, PL/SQL!');
END;
```

Saída esperada:

Olá, PL/SQL!

---

### 4. Estruturas de Controle

#### IF / ELSIF / ELSE

```
DECLARE
  v_idade NUMBER := 20;
BEGIN
  IF v_idade < 18 THEN
    DBMS_OUTPUT.PUT_LINE('Menor de idade');
  ELSIF v_idade < 60 THEN
    DBMS_OUTPUT.PUT_LINE('Adulto');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Idoso');
  END IF;
END;
```

---

#### LOOP

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

---

## WHILE LOOP

```
DECLARE
    v_num NUMBER := 1;
BEGIN
    WHILE v_num <= 3 LOOP
        DBMS_OUTPUT.PUT_LINE('Número: ' || v_num);
        v_num := v_num + 1;
    END LOOP;
END;
```

---

## FOR LOOP

```
BEGIN
    FOR i IN 1..5 LOOP
        DBMS_OUTPUT.PUT_LINE('Valor: ' || i);
    END LOOP;
END;
```

---

## 5. Tipos de Blocos em PL/SQL

### 1. Bloco Anônimo:

- Criado e executado diretamente, não fica armazenado no banco.

Exemplo:

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('Bloco anônimo executado!');  
END;
```

○

## 2. Bloco Nomeado (Procedures, Functions, Triggers):

- Armazenado no banco e reutilizável (veremos nos próximos encontros).

---

## 6. Usando %TYPE e %ROWTYPE

### %TYPE

Permite declarar variáveis com o mesmo tipo de uma coluna:

```
DECLARE  
    v_salario empregados.salario%TYPE;  
BEGIN  
    SELECT salario INTO v_salario FROM empregados WHERE id = 10;  
    DBMS_OUTPUT.PUT_LINE('Salário: ' || v_salario);  
END;
```

### %ROWTYPE

Permite armazenar uma linha inteira de uma tabela:

```
DECLARE  
    v_emp empregados%ROWTYPE;  
BEGIN  
    SELECT * INTO v_emp FROM empregados WHERE id = 10;  
    DBMS_OUTPUT.PUT_LINE('Nome: ' || v_emp.nome || ' - Salário: ' ||  
v_emp.salario);  
END;
```



## Lista de Atividades – Encontro 1

### ◆ Atividade 1 – Primeiro Bloco

Crie um bloco PL/SQL que exiba:

Olá, Mundo do PL/SQL!

---

### ♦ **Atividade 2 – Declaração de Variáveis**

Crie um bloco que:

- Declare duas variáveis: v\_nome e v\_idade;
- Atribua valores;

Exiba:

Nome: <nome> - Idade: <idade>

- 
- 

### ♦ **Atividade 3 – Estrutura Condicional**

Crie um bloco que leia uma variável v\_media e exiba:

- “Aprovado” se maior ou igual a 7
  - “Recuperação” se entre 5 e 6,9
  - “Reprovado” caso contrário
- 

### ♦ **Atividade 4 – Loop**

Crie um bloco que exiba os números de 1 a 10.

---

### ♦ **Atividade 5 – Loop com Condição**

Crie um loop que some os números de 1 a 5 e exiba o total:

Soma total: 15

### ♦ **Atividade 6 – Uso de %TYPE**

Crie uma variável que receba o mesmo tipo de uma coluna (ex: salario) de uma tabela empregados e exiba seu valor.

### ♦ **Atividade 7 – Mini Desafio**

Crie um bloco que:

- Declare uma variável v\_nome\_func
- Busque o nome do funcionário com id = 1 da tabela empregados
- Exiba:

Funcionário: <nome\_func>