

**CENTRO PAULA SOUZA**

GOVERNO DO ESTADO DE  
**SÃO PAULO**

Faculdade de Tecnologia de Bauru

# PROGRAMAÇÃO AVANÇADA DE BANCO DE DADOS

A01 - PL-SQL / PROCEDURES

Prof. Esp. Luis Alexandre da Silva

[luis.silva51@fatec.sp.gov.br](mailto:luis.silva51@fatec.sp.gov.br)

[www.ldstec.com.br](http://www.ldstec.com.br)



# PL/SQL

- PL/SQL é uma linguagem ORACLE que tem por objetivo processar informações do banco de dados.
- Não tem telas gráficas.
- Operações permitidas:
  - Executar comando SQL DML;
  - Criar variáveis e constantes, podendo ser herdados de tipos de dados já existentes;
  - Criar cursores para tratar resultados de SQLs de mais de uma linha de registros;



# PL/SQL

- Tratar Erros;
- Executar comandos de comparação (decisão) e de repetição (laço).
- Vantagens:
  - Portabilidade;
  - Integração com o Gerenciamento de Banco de Dados;
  - Capacidade Procedural;
  - Produtividade.



# BLOCO PL/SQL

BOLOCO PL/SQL:

## **DECLARE**

```
preco_produto PRODUTO.VALOR_PRODUTO%TYPE;  
varCONTROLE NUMBER(3);  
nomeCLIENTE VARCHAR2(100);
```

## **BEGIN**

```
SELECT VALOR_PRODUTO INTO preco_produto FROM PRODUTO  
WHERE CODIGO_PRODUTO = &Codigo;  
IF preco_produto IS NOT NULL THEN  
    DBMS_OUTPUT.PUT_LINE('O Preço do produto é: ' || preco_produto);  
END IF;
```

## **END;**



# TIPOS DE DECLARAÇÃO

- Na área DECLARE podemos declarar:
  - Variáveis;
  - Constantes;
  - Cursors;
  - Estruturas;
  - Tabelas.

Exemplos:

```
Nr_Lidos Constant Number(4) := 15;  
Nr_tentativas NUMBER(5) := 0;
```

SCRIPT:

```
INSERT INTO Teste_TMP (cd_teste, vl_teste) VALUES (SELECT cd_pais,  
nm_pais FROM PAIS);
```



# COMANDO SQL DENTRO PL/SQL

- Os comando de manipulação de dados (DML) podem ser utilizados dentro dos blocos PL/SQL (INSERT, UPDATE, DELETE e SELECT), bem como as funções de manipulação de string, números e datas.
- A cláusula INTO serve para alocar a uma variável o valor de um retorno de um SELECT.

Exemplo:

**DECLARE**

```
Nome Cliente.Nm_Cliente%Type;  
Codigo Cliente.Cd_Cliente%Type;
```

**BEGIN**

```
SELECT Nm_Cliente, Cd_Cliente  
      INTO nome, codigo  
      FROM Cliente  
      WHERE Cd_codigo = 10;
```

**END;**



# TRATAMENTO DE TRANSAÇÃO

- Ao realizar INSERTs, UPDATEs, ou DELETEs é necessário ao final do procedimento utilizar o COMMIT para confirmar a transação. Através de controles de Erros é possível realizar o ROLLBACK para desfazer um conjunto de transações já realizadas e não confirmadas pelo COMMIT.



# IF...THEN... ELSEIF... END IF

- Comando condicional.

Exemplo:

**BEGIN**

...

```
IF valor_compra <= 3000 THEN
    valor_desconto := 10;
ELSEIF valor_compra <= 5000 THEN
    valor_desconto := 20;
ELSE
    valor_desconto := 30;
END IF;
```

...

**END;**



# FOR... IN... REVERSE... LOOP... END LOOP

- Comando Laço.

Exemplo:

**BEGIN**

...

FOR contador IN 0..10

LOOP

...

...

END LOOP;

...

**END;**



# WHILE... LOOP... END LOOP

- Comando Laço.

Exemplo:

**BEGIN**

...

WHILE (codigo > 10) and (situacao = 'A')

LOOP

...

...

END LOOP;

...

**END;**



# CURSORES

- São áreas compostas de linhas e colunas em memória que servem para armazenar o resultado de uma seleção (SELECT), que retorna 0 (zero), uma, ou várias linhas de registros.

Exemplo:

**BEGIN**

...

FOR RS IN (SELECT...)

LOOP

...RS.COLUNA;

...

END LOOP;

...

**END;**



# PROCEDURES

- Blocos PL/SQL que podem ser invocados dentro do banco de dados.
- Os nomes das procedures podem ter no máximo 30 caracteres.
- Usuário tem que ter privilégios para criar e acessar as procedures.

Exemplo:

```
CREATE OR REPLACE PROCEDURE nome_proc  
    (arg1 modo (IN/OUT/IN OUT) tipo_dado,  
     arg2 modo (IN/OUT/IN OUT) tipo_dado,  
     argn modo (IN/OUT/IN OUT) tipo_dado)  
IS ou AS  
    declaração de variáveis e contrantes;  
BEGIN  
  
    bloco PL/SQL;  
  
END nome_proc;
```



# PROCEDURES

Exemplo:

```
CREATE OR REPLACE PROCEDURE formata_telefone
    (param_fone IN varchar2)
IS
    varFone Varchar2(20);
BEGIN
    varFONE := '(' || SUBSTR(param_fone,1,2) || ')' ||
                SUBSTR(param_fone,3,4) || '-' ||
                SUBSTR(param_fone,7,4);
    DBMS_OUTPUT.PUT_LINE (varFONE);
END formata_telefone;
```



# DÚVIDAS??

