



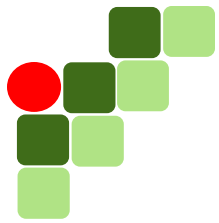
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Araraquara

Banco de Dados II

Cristiane Yaguinuma

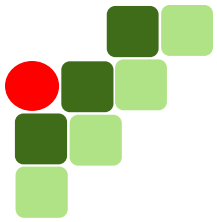
cristiane.yaguinuma@ifsp.edu.br

- **Introdução a PL/SQL – blocos anônimos**



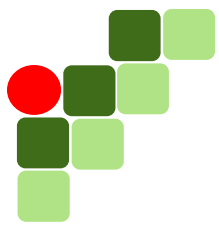
Roteiro da aula

- ▶ **Conceitos de programação de banco de dados**
- ▶ **Introdução a PL/SQL**
 - Unidades de programa
 - Blocos anônimos
 - Variáveis e tipos de dados
 - SELECT INTO
 - Comandos DML



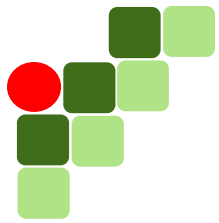
Acesso ao banco de dados

- ▶ **Biblioteca de funções de banco de dados**
 - API – Application Programming Interface
 - Exemplo: JDBC – Java DataBase Connectivity
- ▶ **Linguagem de programação de banco de dados**
 - Programas armazenados e executados pelo SGBD
 - Exemplo: Linguagem PL/SQL da Oracle



Por que utilizar programas executados pelo SGBD?

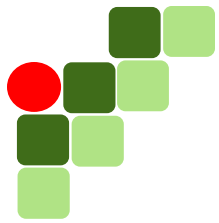
- ▶ Reutilizar programas em diversas aplicações
 - Redução de duplicação de esforço e melhoria da modularidade de software
- ▶ Reduzir a transferência de dados e o custo de comunicação entre aplicação e SGBD
 - Dados brutos x dados processados
 - Desempenho otimizado
- ▶ Definir restrições ou dados derivados mais complexos
 - Não realizados com restrições, visões, restrições de integridade referencial



Divergência de impedância

- ▶ Termo usado para descrever problemas relacionados às diferenças entre o modelo de banco de dados e o modelo da linguagem de programação da aplicação
- ▶ Tipos de dados diferentes
- ▶ Estruturas de dados
 - Modelo relacional vs. Modelo orientado a objetos
- ▶ Frameworks de Mapeamento Objeto–Relacional (ORM)
 - Exemplo: Hibernate

	Vantagens	Desvantagens
JDBC	<ul style="list-style-type: none"> • Maior portabilidade: caso seja necessário migrar a aplicação para outro SGBD, modifica-se a configuração de drivers. 	<ul style="list-style-type: none"> • Divergência de impedância <ul style="list-style-type: none"> • tipos de dados • estruturas de dados
Programas armazenados e executados pelo SGBD	<ul style="list-style-type: none"> • Apropriados para aplicações que possuem intensa interação com o banco de dados • Menor impacto de divergência de impedância 	<ul style="list-style-type: none"> • Aprendizado de uma nova linguagem de programação • Linguagens de BD proprietárias podem dificultar a portabilidade de programas para outros SGBDs

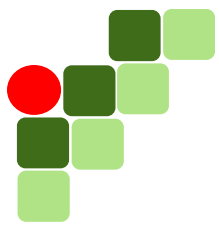


PL/SQL

- ▶ Linguagem de programação para acesso a BDs relacionais da Oracle

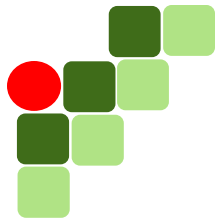
PL/SQL: Procedural Language extension to Structured Query Language

- ▶ Une o estilo modular e procedural de linguagens de programação à versatilidade no acesso a banco de dados via SQL
- ▶ Oracle Database PL/SQL User's Guide and Reference
 - https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/toc.htm



Tipos de unidades de programa

Anônimo	Bloco executável de comandos PL/SQL que não possui um nome explícito.
Trigger	Rotina disparada automaticamente antes ou depois de comandos update, insert ou delete.
Procedure	Pode receber parâmetros de entrada ou de saída. Ativado como se fosse um comando da linguagem.
Function	Pode receber parâmetros apenas de entrada e, necessariamente, retorna um valor em seu nome. A ativação ocorre em expressões.



Unidade de programa

- ▶ Um bloco anônimo possui a seguinte estrutura:

```
[DECLARE]
```

```
-- declaração de variáveis, constantes
```

```
-- contém inicializações
```

```
BEGIN
```

```
-- comandos SQL
```

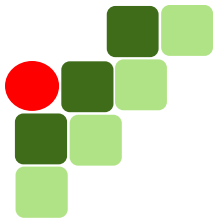
```
-- comandos PL/SQL
```

```
[EXCEPTION]
```

```
-- tratamento de erros e emissão de mensagens
```

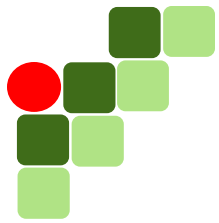
```
END;
```

```
/
```



Bloco anônimo PL/SQL

```
SET SERVEROUTPUT ON
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello world!');
END;
/
```



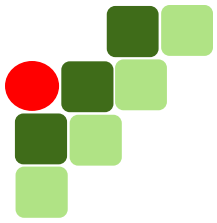
Variáveis em PL/SQL

- ▶ Declaradas e inicializadas na seção DECLARE

```
nome_variavel [CONSTANT] datatype [NOT NULL]
               [:= | DEFAULT expr];
```

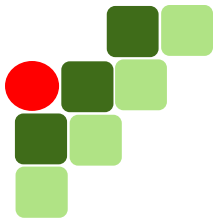
DECLARE

```
emp_hire_date DATE;
emp_dept      NUMBER(4) NOT NULL := 10;
city          VARCHAR2(20) := 'Araraquara';
num_const     CONSTANT NUMBER := 500;
emp_job       VARCHAR2(10) DEFAULT 'IT_PROG';
```



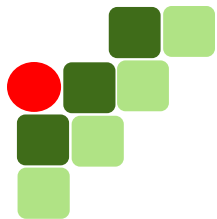
Bloco anônimo PL/SQL

```
DECLARE
    l_message VARCHAR2(100) := 'Hello World!';
BEGIN
    DBMS_OUTPUT.put_line (l_message);
END;
```



Bloco anônimo PL/SQL

```
DECLARE
    MY_VAR VARCHAR2(30);
BEGIN
    MY_VAR := '&INPUT';
    DBMS_OUTPUT.PUT_LINE('HELLO ' || MY_VAR);
END;
/
```



Atributo %TYPE

- ▶ Usado para declarar uma variável a partir de
 - Uma definição de coluna de tabela
 - Outra variável declarada

```
nome_variavel tabela.nome_coluna%TYPE;
```

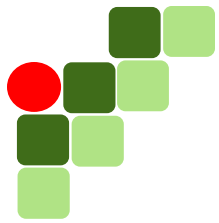
```
nome_variavel variavel%TYPE;
```

```
DECLARE
```

```
emp_hire_date employees.hire_date%TYPE;
```

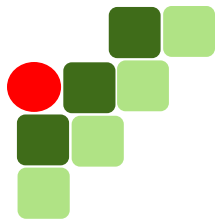
```
commission      NUMBER(3,2);
```

```
min_commission  commission%TYPE;
```



Operações em PL/SQL

- ▶ Operações e funções são as mesmas usadas em comandos SQL
 - Operações aritméticas
 - Comparadores lógicos
 - Concatenação
 - Precedência e uso de parênteses
 - Funções SQL



SELECT INTO

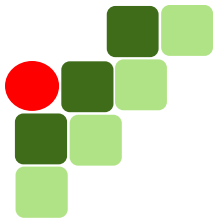
- ▶ Recuperar dados do BD para variáveis em blocos PL/SQL

```
SELECT coluna1[, coluna2]...  
INTO nome_variavel1[, nome_variavel2]...  
FROM tabela1[, tabela2]...  
[WHERE condições]
```

- ▶ Consultas devem retornar apenas uma tupla
- ▶ Não utilizar nomes de variáveis iguais aos nomes das colunas das tabelas



/



Comandos DML em PL/SQL

```
DECLARE
```

```
    deptno NUMBER := 9999;
```

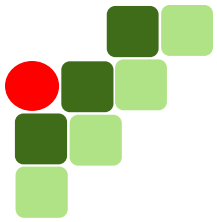
```
BEGIN
```

```
    INSERT INTO departments (department_id,  
                             department_name)
```

```
    VALUES (deptno, 'Teste');
```

```
END;
```

```
/
```



Comandos DML em PL/SQL

```
DECLARE
```

```
    deptno NUMBER := 9999;
```

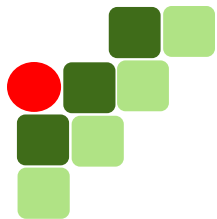
```
BEGIN
```

```
    DELETE FROM departments
```

```
    WHERE department_id = deptno;
```

```
END;
```

```
/
```



Exercícios de fixação

1. **Escreva um bloco de programa PL/SQL que descreva na tela o menor, o maior e o salário médio dos empregados do departamento com id = 60**
2. **Escreva um bloco de programa PL/SQL para atualizar o salário dos empregados do departamento com id = 50 para o mesmo valor do salário médio dos empregados do departamento com id = 60**

DECLARE

min_sal employees.salary%TYPE;

max_sal min_sal%TYPE;

deptno employees.department_id%TYPE := 60;

BEGIN

SELECT MIN(salary), MAX(salary)

INTO min_sal, max_sal

FROM employees

WHERE department_id = deptno;

DBMS_OUTPUT.PUT_LINE ('O menor salário do

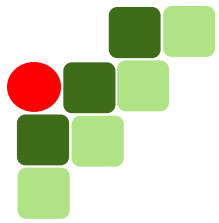
departamento ' || deptno || ' é ' ||

min_sal || ' e o maior salário é ' ||

max_sal);

END;

/



Material da aula

- ▶ <http://www.oracle.com/technetwork/issue-archive/2011/11-mar/o21plsql-242570.html>
- ▶ https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/overview.htm#LNPLS001