



Instituto Federal
Campus Goiânia

Bacharelado em Sistemas de Informação

Banco de Dados II



Prof. Dory Gonzaga Rodrigues





Agenda

- STORED PROCEDURE





Lista de Exercícios

PL/PgSQL: Exercícios

1) Crie uma função que receba um número inteiro e imprima a tabuada conforme apresentado abaixo:

```
select * from tabuada(5);
```

pane	
a Output Explain Messages	
tabuada	character varying
5 X 1 =	5
5 X 2 =	10
5 X 3 =	15
5 X 4 =	20
5 X 5 =	25
5 X 6 =	30
5 X 7 =	35
5 X 8 =	40
5 X 9 =	45
5 X 10 =	50





Lista de Exercícios

PL/PgSQL: Exercícios

1) Crie uma função que receba um número inteiro e imprima a tabuada conforme apresentado abaixo:

```
CREATE OR REPLACE FUNCTION tabuada (IN n INTEGER) RETURNS SETOF varchar AS $$  
BEGIN  
    FOR I IN 1..10 LOOP  
        RETURN NEXT n || ' x ' || i || ' = ' || n*i;  
    END LOOP;  
END;$$  
LANGUAGE plpgsql;  
  
select * from tabuada(5);
```





Lista de Exercícios

PL/PgSQL: Exercícios

2) Criar uma função que receba dois número inteiros (representando o intervalo entre dois anos) e retorne os anos bissextos existentes neste período. Um ano será bissexto quando for possível dividi-lo por 4, mas não por 100 ou quando for possível dividi-lo por 400.

```
select * from bissexto(2000, 2010);
```

pane

a Output

Explain

Messages

History

bissexto character varying

2000

2004

2008





Lista de Exercícios

PL/PgSQL: Exercícios

2) Criar uma função que receba dois número inteiros (representando o intervalo entre dois anos) e retorne os anos bissextos existentes neste período. Um ano será bissexto quando for possível dividi-lo por 4, mas não por 100 ou quando for possível dividi-lo por 400.

```
CREATE OR REPLACE FUNCTION bissexto (IN ano_i INTEGER, IN ano_f INTEGER) RETURNS SETOF varchar AS $$
BEGIN
    FOR ano IN ano_i .. ano_f LOOP
        IF (MOD(ano,4) = 0 AND MOD(ano,100) != 0) OR (MOD(ano,400) = 0) THEN
            RETURN NEXT ano;
        END IF;
    END LOOP;
END;$$
LANGUAGE plpgsql;

select * from bissexto(2000, 2010);
```





Lista de Exercícios

PL/PgSQL: Exercícios

3) Crie uma função que receba o código da turma e o da disciplina e retorne o nome da disciplina, o nome do professor e a lista dos alunos com a média das notas:

```
SELECT * FROM Lista_Turma_Resultado('52A',205195);
```

pane

a Output Explain Messages History

lista_turma_resultado
character varying

DISCIPLINA:DIREITO APLICADO A INFORMÁTICA

PROFESSOR: NILDETE

ALUNOS || MÉDIA

Nome: Henrique Nota: 0

Nome: Noriel Nota: 0

Nome: Alan Nota: 6

Nome: Alberto Nota: 0

Nome: Leonardo Nota: 0

Nome: Pedro Nota: 0





Lista de Exercícios

PL/PgSQL: Exercícios

```
3) CREATE OR REPLACE FUNCTION Lista_Turma_Resultado(IN cod_t VARCHAR, IN cod_d INTEGER) RETURNS SETOF VARCHAR AS $$
DECLARE
    registro RECORD;
    v_nome_disc VARCHAR;
    v_nome_prof VARCHAR;
    v_nome_aluno VARCHAR;
    v_media FLOAT;

BEGIN
    SELECT d.nom_disc INTO v_nome_disc FROM disciplinas d WHERE cod_disc = cod_d;

    SELECT DISTINCT p.nom_prof INTO v_nome_prof
    FROM turmas_matriculadas tm NATURAL JOIN turmas t NATURAL JOIN professores p
    WHERE turma = cod_t
    AND cod_disc = cod_d;

    RETURN NEXT 'DISCIPLINA:' || v_nome_disc;
    RETURN NEXT 'PROFESSOR: ' || v_nome_prof;
    RETURN NEXT 'ALUNOS || MÃ&DIA';

    FOR registro IN SELECT a.nom_alu, (tm.nota_1 + tm.nota_2 + tm.nota_3 + tm.nota_4)/4 AS v_media
    FROM turmas_matriculadas tm NATURAL JOIN alunos a
    WHERE turma = cod_t
    AND cod_disc = cod_d LOOP
        EXIT WHEN NOT FOUND;

        IF (registro.v_media != 0) THEN
            RETURN NEXT 'Nome: ' || registro.nom_alu || ' Nota: ' || registro.v_media;
        ELSE
            RETURN NEXT 'Nome: ' || registro.nom_alu || ' Nota: 0';
        END IF;
    END LOOP;
    RETURN;
END; $$
LANGUAGE 'plpgsql';
```