

Instituto Federal Sul-rio-grandense – Campus Camaquã
Disciplina: Banco de Dados II – Turma 2024
Professor: Vinícius Alves Hax
Assunto: Revisão sobre PL/pgSQL

Instruções gerais: para os exercícios abaixo você pode conversar com seus colegas (em voz baixa) e até consultar a Internet. Mas NÃO use o chatGPT, ele nem sempre estará lá :P

Para os exercícios abaixo utilize as tabelas e registros abaixo:

```
CREATE TABLE teacher(  
    id INT,  
    name VARCHAR(200),  
    side VARCHAR(200),  
    level REAL  
);
```

```
CREATE TABLE student(  
    id SERIAL,  
    name VARCHAR(200),  
    teacher_id INT  
);
```

```
INSERT INTO teacher (id, name, side, level) VALUES (1, 'Palpatine', 'Dark', 100.0);  
INSERT INTO teacher (id, name, side, level) VALUES (2, 'Yoda', 'Light', 100.0);  
INSERT INTO teacher (id, name, side, level) VALUES (3, 'Dookan', 'Light', 90.0);  
INSERT INTO teacher (id, name, side, level) VALUES (4, 'Qui-gon Jin', 'Light', 90.0);  
INSERT INTO teacher (id, name, side, level) VALUES (5, 'Obi Wan', 'Light', 85.0);  
INSERT INTO teacher (id, name, side, level) VALUES (6, 'Anakin', 'Light', 95.0);  
INSERT INTO teacher (id, name, side, level) VALUES (7, 'Luke', 'Light', 100.0);  
INSERT INTO teacher (id, name, side, level) VALUES (8, 'Mace Windu', 'Light', 100.0);
```

```
INSERT INTO student (name, teacher_id) VALUES ('Luke', 2);  
INSERT INTO student (name, teacher_id) VALUES ('Anakin', 5);  
INSERT INTO student (name, teacher_id) VALUES ('Anakin', 1);  
INSERT INTO student (name, teacher_id) VALUES ('Obi Wan', 4);  
INSERT INTO student (name, teacher_id) VALUES ('Qui-gon Jin', 3);  
INSERT INTO student (name, teacher_id) VALUES ('Ahsoka', 6);  
INSERT INTO student (name, teacher_id) VALUES ('Rei', 7);  
INSERT INTO student (name, teacher_id) VALUES ('Kylo Ren', 7);  
INSERT INTO student (name) VALUES ('Palpatine');
```

Crie funções anônimas, usando PL/pgSQL, para cada um dos exercícios de maneira a fazer o que se pede.

A saída deverá sempre ser mostrada usando a função “raise notice”

1) Crie uma função anônima, em PL/pgSQL, que mostre o número de alunos que cujo professor (teacher) chama-se “Luke”.

```
do $$  
declare  
    id_luke integer;
```

```

    contador integer;
begin
    -- Primeiro pegamos o id do professor
    select teacher.id
    into id_luke
    from teacher where name = 'Luke';

    select count(*)
    into contador
    from student where student.teacher_id = id_luke;

    raise notice 'O professor Luke tem % alunos', contador;
end $$;

```

2) Mostre o ID do aluno (student) cujo nome é “Anakin”.

```

do $$
declare
    id_anakin integer;
begin
    -- Primeiro pegamos o id do professor
    select student.id
    into id_anakin
    from student where name = 'Anakin';

    raise notice 'O ID de Anakin é %', id_anakin;
end $$;

```

3) Mostre o nome do aluno (student) cujo ID é 3.

```

do $$
declare
    name_student varchar(200);
begin
    -- Primeiro pegamos o id do professor
    select student.name
    into name_student
    from student where id = 3;

    raise notice 'O nome do aluno(a) com ID 3 é %', name_student;
end $$;

```

4) Faça um programa que mostre os os números ímpares no intervalo entre 1 e 10.

```

do $$
begin
    -- Usando o for não precisamos declarar a variável de controle
    for numero in 1..10 by 2 loop
        raise notice 'Número: %', numero;
    end loop;
end; $$

```

5) Faça um programa que mostre, em ordem decrescente, os números pares situados no intervalo entre 90 e 100.

```
do $$
begin
-- Usando o for não precisamos declarar a variável de controle
for numero in reverse 100..90 by 2 loop
raise notice 'Número: %', numero;
end loop;
end; $$
```

6) Faça um programa cuja variável seja declarada com o tipo “%type” e que mostre o nível de poder do teacher com maior nível de poder. Dica: Use o parâmetro LIMIT para limitar a quantidade de resultados.

```
do $$
declare
    max_level teacher.level%type;
begin
-- Primeiro pegamos o id do professor
select teacher.level
into max_level
from teacher order by teacher.level DESC
limit 1;

raise notice 'Nível máximo: %', max_level;
end $$;
```

7) Faça um programa que pegue o nível de poder (coluna level) do teacher com ID = 2, aumente esse nível em 50% e mostre o resultado.

```
do $$
declare
    level real;
begin
select teacher2.level
into level
from teacher2 where id=2;
raise notice 'Nível inicial %', level;
level = level * 1.5;
raise notice 'Novo nível %', level;
end $$;
```

8) Refaça o programa anterior porém aumentando o nível em 100%.

```
do $$
declare
    level real;
begin
select teacher2.level
into level
```

```

from teacher2 where id=2;
raise notice 'Nível inicial %', level;
level = level * 2;
raise notice 'Novo nível %', level;
end $$;

```

9) Mostre todos os professores que tem pelo menos um aluno, e ao seu lado o nome do aluno. A saída deverá ser formatada como mostrado a seguir:

Professor X treinou aluno Y (onde X e Y são os nomes)

Exemplo da formatação:

**Professor Girafales treinou aluno Chaves
Professor Girafales treinou aluno Quico
Professor Madruga treinou aluno Girafales**

-- Solução 1 usando sub-consultas

```

DO $$
DECLARE
    professor varchar(100);
    aluno varchar(100);
BEGIN
    -- Usamos um loop FOR para iterar sobre cada linha da tabela 'clientes'
    FOR professor,aluno IN
        SELECT * FROM
        (select teacher.name as teacher_name, student.name as student_name from teacher left
        join student on
        teacher.id = student.teacher_id) AS linhas
    LOOP
        -- Exibimos os valores das colunas da linha atual
        RAISE NOTICE 'Professor(a) % treinou aluno(a) %', professor, aluno;
    END LOOP;
END $$;

```

-- Solução 2 sem usar uma sub-consulta e usando o tipo record

-- com o tipo record podemos pegar o resultado de uma linha mesmo

-- não sendo uma tabela específica

```

DO $$
DECLARE
    linha record;
BEGIN
    -- Usamos um loop FOR para iterar sobre cada linha da tabela 'clientes'
    FOR linha IN
        (SELECT teacher.name AS teacher_name, student.name AS student_name
        FROM teacher
        LEFT JOIN student ON teacher.id = student.teacher_id)
    LOOP
        -- Exibimos os valores das colunas da linha atual
        RAISE NOTICE 'Professor(a) % treinou aluno(a) %', linha.teacher_name, linha.student_name;
    END LOOP;
END $$;

```