

# SINTAXE DE UMA PROCEDURE PLPGSQL

Uma **desvantagem das funções** definidas pelo usuário é que elas não podem executar transações



```
create [or replace] procedure nome_do_procedimento(lista_de_parametros)
language plpgsql
as
$$
declare
-- declaração de variáveis
begin
-- lógica do procedimento
end;
$$
```

As **procedures** foram introduzidas a partir da versão 11 do PostgreSQL

# REESCREVENDO A NOSSA FUNÇÃO COMO UMA PROCEDURE



```
create or replace procedure proc_exclui_aluno(id int) as
$$
begin
    perform * from alunos where id_aluno = id;
    if not found then
        raise notice 'aluno não encontrado!';

    else
        delete from alunos where id_aluno = id;
        raise notice 'aluno excluído com sucesso!';
    end if;
end;

$$
language plpgsql;
```

# COMO EXECUTAR UMA PROCEDURE



```
call nome_da_procedure(argumentos)
```

```
-- executando a procedure criada  
call proc_exclui_aluno(1);
```

# IMPRIMINDO MENSAGENS EM PL/PGSQL



```
RAISE nível mensagem
```

Níveis: DEBUG, LOG, INFO, NOTICE, WARNING, ou EXCEPTION

# RAISE INFO/WARNING/NOTICE



```
-- mensagens de aviso
```

```
RAISE notice 'Produto não castrado!'
```

```
RAISE info 'Produto não castrado!'
```

```
RAISE warning 'Produto não castrado!'
```

Exibem mensagens de aviso

# RAISE EXCEPTION



-- mensagens de exceção

```
RAISE exception 'Não foi possível conectar ao BD!'
```

**RAISE EXCEPTION**, normalmente, aborta a transação corrente.

# TRATAMENTO DE EXCEÇÕES



```
declare
    ...
begin
    -- instruções
exception
    when condição [or outra_condição ...] then
        manipulador da exceção;
    [when condição [or outra_condição ...] then
        manipulador da exceção;
    [when outras then
        manipulador da exceção;
    ]
end;
```

# EXEMPLO DE USO DO TRATAMENTO DE EXCEÇÕES



```
create or replace function add_aluno(novo_id int, novo_aluno varchar)
returns int as
$$
begin
    insert into alunos values(novo_id, novo_aluno);

    if found then
        raise notice 'Aluno cadastrado com sucesso!';
        return 1;
    end if;

exception
    when unique_violation then
        raise notice 'Já existe aluno com este ID';
        return -1;
end;
$$
language plpgsql;
```



# HORA DE PRATICAR!

Crie uma função para cadastrar a participação dos empregados em projetos (ou seja, para inserir dados na tabela 'participacao'). Lembre-se de testar, no código, se o funcionário e projetos passados por parâmetros existem. Além disso, as horas trabalhadas devem ser maiores que zero. SE a participação for cadastrada de forma correta, retorne o valor 1, caso contrário, retorne um valor negativo.

[https://drive.google.com/file/d/1M-YOyo5hjuAv3HpPk\\_Lno6oC0s5lPh7A/view?usp=sharing](https://drive.google.com/file/d/1M-YOyo5hjuAv3HpPk_Lno6oC0s5lPh7A/view?usp=sharing)

