

FUNCTIONS

(Funções)



O que é uma FUNCTION?

- São rotinas definidas no banco de dados, identificadas por um nome pelo qual podem ser invocadas.
- Podem executar instruções, receber parâmetros e retornar valores com tipos de dados definidos.



Sintaxe

```
Create function nome_funcao (lista_parametros)
returns tipo_retorno
language plpgsql AS
$$
declare
-- declaração de variáveis
begin
-- lógica
end;
$$
```

Sintaxe

- **nome_funcao**: o nome que identificará a função (o nome que quiser).
- **parâmetro**: nome do(s) parâmetro(s) – são opcionais e, caso não sejam necessários, devem permanecer apenas os parênteses vazios na declaração da função.
- **tipo_parâmetro**: O tipo de dado do parâmetro da função, caso exista: int, double, varchar(30), float, char(4), etc.
- **tipo_retorno**: tipo de dado retornado pela função (int, double, etc).

Create function getNumeroFilmes (dataInicial DATE,
dataFinal DATE)

returns INTEGER

language plpgsql AS

\$\$

declare

 contadorFilmes INTEGER;

begin

 SELECT count(*) INTO film_count FROM film WHERE
 dataLancamento BETWEEN dataInicial AND dataFinal;
 RETURN contadorFilmes;

end;

\$\$

Sintaxe

- OBSERVAÇÃO:
- A declaração de novas variáveis pode ser necessária no corpo da função!
- Tendo criado a function, como executá-la? Como na maioria das vezes haverá um tipo de retorno, a chamada da mesma poderá ser feita **após o SELECT** ou **após a cláusula WHERE**, sendo, para este último caso, o valor retornado deve ser comparado a alguma condição.

Exemplo

- `SELECT contar_filmes('ACAO') FROM filmes;`
- `SELECT *`
- `FROM Filmes`
- `WHERE tipo = pegar_tipo_com_mais_filmes();`

Alterando uma FUNCTION

```
CREATE OR REPLACE FUNCTION nome_funcao  
(lista_parametros)  
returns tipo_retorno  
language plpgsql AS  
$$  
declare  
-- declaração de variáveis  
begin  
-- lógica  
end;  
$$
```


Excluir uma FUNCTION

- `DROP FUNCTION nome_funcao;`