Desenvolvendo em SQL

**Criando função:**

O compor da função deve estar entre aspas simples ou $$:

Ex1:

CREATE FUNCTION primeira\_funcao() returns integer as '

select (5 - 3) \* 2

' language SQL;

select primeira\_funcao();

select primeira\_funcao() as numero;

Ex2:

CREATE FUNCTION soma\_dois\_numeros( integer, integer) RETURNS INTEGER AS '

SELECT $1 + $2

' LANGUAGE SQL;

SELECT soma\_dois\_numeros(3, 12);

Também é possível fazer o mesmo com tabelas, n apenas para um valor único:

drop table instrutor;

CREATE TABLE instrutor(

id SERIAL PRIMARY KEY,

nome VARCHAR (255) NOT NULL,

salario DECIMAL (10, 2)

);

INSERT INTO instrutor (nome, salario) VALUES ('Vinicius Dias', 100);

CREATE FUNCTION dobro\_do\_salario(instrutor) returns decimal as $$

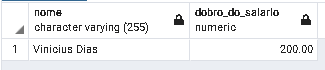
select $1.salario \* 2 as dobro

$$ language SQL;

select nome,dobro\_do\_salario(instrutor.\*) from instrutor;

ou

select nome,dobro\_do\_salario(instrutor) from instrutor;



select nome,dobro\_do\_salario(instrutor) from instrutor;

CREATE or replace FUNCTION cria\_intrutor\_falso() returns instrutor as $$

select 22, 'Nome Falso' , 200::decimal ;

$$ language SQL;

select \* from cria\_intrutor\_falso();

CREATE FUNCTION instrutores\_bem\_pagos (valor\_salario)

SELECT \* FROM instrutor WHERE salário > valor\_salario



Funções que retornam tabelas:

CREATE or replace FUNCTION instrutores\_bem\_pagos (valor\_salario decimal) returns setof instrutor as $$

SELECT \* FROM instrutor WHERE salario > valor\_salario;

$$ language SQL;

select \* from instrutores\_bem\_pagos(300);

**PLPGSQL**

Mais utilizado quando temos que fazer contas e uma programação mais comum, contem ideias do sql com mais funcionalidades:

CREATE OR REPLACE FUNCTION primeira\_pl() returns integer as $$

DECLARE declara variavel

primeira\_variavel integer DEFAULT 3;

BEGIN começa a parte do SQL

primeira\_variavel := primeira\_variavel \* 2;

return primeira\_variavel;

END

$$ language plpgsql;

select primeira\_pl();

