

STORED FUNCTION



REINALDO_{JR}

CRIE AS SEGUINTEs TABELAS

```
CREATE TABLE pessoas
(
  cpf numeric NOT NULL,
  nome character varying,
  CONSTRAINT "pessoas-cpf" PRIMARY KEY (cpf)
)
```

```
CREATE TABLE funcionarios
(
  id serial NOT NULL, --
  nome character varying(80),
  tipo integer,
  salario numeric(14,2),
  CONSTRAINT "PK_funcionarios" PRIMARY KEY (id)
)
```

```
CREATE OR REPLACE FUNCTION lista_pessoas()
  RETURNS text AS
$BODY$
DECLARE
  listaPessoas RECORD;
  retorno      TEXT DEFAULT '';
  contLog      INTEGER DEFAULT 0;
BEGIN
  FOR listaPessoas IN
    SELECT cpf, nome
      FROM pessoas
    LOOP

      retorno = retorno || ' ' || listaPessoas.cpf || ' - ' || listaPessoas.nome;

    END LOOP;
  return retorno;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION lista_pessoas()
  OWNER TO postgres;
```

```
CREATE OR REPLACE FUNCTION atualizar_funcionarios()
  RETURNS text AS
$BODY$
DECLARE
  listaFuncionarios RECORD;
  retorno          TEXT DEFAULT '';
  descTipo         TEXT DEFAULT '';
BEGIN
  FOR listaFuncionarios IN
    SELECT id, nome, tipo, salario
      FROM funcionarios
    LOOP
      IF (listaFuncionarios.tipo = 1) THEN
        descTipo = 'PLENO';
      ELSE
        descTipo = 'SENIOR';
      END IF;
      retorno = retorno || ' ' || listaFuncionarios.id || '-' || listaFuncionarios.nome || '-' ||
        descTipo || '-' || listaFuncionarios.salario;
    END LOOP;
  return retorno;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION atualizar_funcionarios()
  OWNER TO postgres;
```

CRIE AS SEGUINTE TABELAS

```
CREATE TABLE produtos
(
  id serial NOT NULL,
  descricao character varying(60),
  valor numeric(14,2),
  unidade character varying(20),
  CONSTRAINT "PK_id_produto" PRIMARY KEY (id)
)
```

```
CREATE TABLE municipios
(
  id serial NOT NULL,
  nome character varying(60),
  codg_ibge character varying(12)
)
```

```
CREATE TABLE aluno
(
  id serial NOT NULL,
  nome character varying(60),
  nota_n1 numeric(5,2),
  nota_n2 numeric(5,2),
  nota_media_final numeric(5,2),
  CONSTRAINT "PK_aluno" PRIMARY KEY (id)
)
```

EXERCÍCIOS

- 1) Crie uma stored function, que liste todos os campos da tabela “produtos”, mostrando todos os seus campos na listagem.
- 2) Crie uma stored function, que liste todos os campos da tabela “municipios”, mostrando todos os seus campos na listagem.

EXERCÍCIOS

3) Crie uma stored function, que liste todos os campos da tabela “alunos”, porém, o campo “nota_media_final”, será calculado no instante em que os dados forem mostrados e atualizado na tabela em questão. Durante a listagem das notas, se a média for < 7 , então escrever “fazer N3” e a nota mínima que o aluno necessita tirar, senão escrever “Aprovado!!”

EXERCÍCIOS

4) Crie uma stored function, que liste todos os campos da tabela “produtos”, somente para os produtos com valor > 100.

5) Crie uma stored function, que liste todos os campos da tabela “aluno”, mas que traga somente os alunos com média ≥ 7 .

EXERCÍCIOS

6) Crie uma stored function, que liste todos os alunos já aprovados sem precisar de fazer N3.