Nome: Bruna Rodrigues da Costa

Entregar até: 09/08/2022 às 12:00

**Atividade 9**

1. Nessa atividade vamos exercitar um pouco mais o aprendizado sobre o JOIN. Como ja conversamos, o JOIN é uma consulta que envolve duas ou mais tabelas. Essas tabelas geralmente possuem relação entre si. Para uma melhor experiência nesse exercício, vamos ampliar nosso banco de dados contas\_publicas. Antes de iniciarmos as consultas, vcs terão que incluir algumas tabelas e dados no banco. Segue abaixo:

- Implemente a tabela **funcionario** com os campos: id (inteiro, chave primária, não nulo, auto incremento), nome (texto, não nulo), cpf (texto, não nulo), rg (texto, não nulo), funcionario\_cargo\_id (inteiro, não nulo, FK), unidade\_administrativa\_id (inteiro, não nulo, FK), data\_admissao (data, não nulo), salario\_base (inteiro, não nulo), ativo (booleano, não nulo). Cole abaixo o script SQL.

**CREATE** **TABLE** funcionario(

id **INT** **PRIMARY** **KEY** **NOT** **NULL** **AUTO\_INCREMENT**,

nome **VARCHAR** (70) **NOT** **NULL**,

cpf **CHAR** (11) **NOT** **NULL**,

rg **VARCHAR** (15) **NOT** **NULL**,

funcionario\_cargo\_id **INT** **NOT** **NULL**,

unidade\_administrativa\_id **INT** **NOT** **NULL**,

data\_emissao **DATE** **NOT** **NULL**,

salario\_base **INT** **NOT** **NULL**,

ativo **BOOL** **NOT** **NULL**,

**FOREIGN** **KEY** (funcionario\_cargo\_id) **REFERENCES** funcionario\_cargo (id),

**FOREIGN** **KEY** (unidade\_administrativa\_id) **REFERENCES** unidade\_administrativa (id)

);

- Insira 200 funcionarios fictícios na tabela. Você pode utilizar esse [gerador de pessoas](https://www.4devs.com.br/gerador_de_pessoas) como auxilio. Cole o script de insert no pastebin e cole abaixo o link aqui no exercício.

https://pastebin.com/5HtDgup5

- Implemente a tabela **funcionario\_cargo** com os campos: id (inteiro, chave primária, não nulo, auto incremento), nome\_cargo (texto, não nulo), funcionario\_setor\_atuacao\_id (inteiro, não nulo, FK), unidade\_administrativa\_id (inteiro, não nulo, FK), ativo (booleano, não nulo). Cole abaixo o script SQL.

**CREATE** **TABLE** funcionario\_cargo(

id **INT** **PRIMARY** **KEY** **NOT** **NULL** **AUTO\_INCREMENT**,

nome\_cargo **VARCHAR** (70) **NOT** **NULL**,

funcionario\_setor\_atuacao\_id **INT** **NOT** **NULL**,

unidade\_administrativa\_id **INT** **NOT** **NULL**,

ativo **BOOL** **NOT** **NULL**,

**FOREIGN** **KEY** (funcionario\_setor\_atuacao\_id) **REFERENCES** funcionario\_setor\_atuacao (id),

**FOREIGN** **KEY** (unidade\_administrativa\_id) **REFERENCES** unidade\_administrativa (id)

);

- Insira os cargos: Coordenador (setor Tecnologia da Informação), Gerente (setor Administração), Secretário (setor Administração), Desenvolvedor de Software Junior (setor Tecnologia da Informação), Desenvolvedor de Software Pleno (setor Tecnologia da Informação), Desenvolvedor de Software Senior (setor Tecnologia da Informação), Analista I (setor Tecnologia da Informação), Analista II (setor Tecnologia da Informação), Analista III (setor Tecnologia da Informação), Contador (setor Contabilidade), Gestor (setor Administração), Administrador (setor Administração). Atribua Unidades administrativas aleatórias para cada cargo. Cole o script de insert no pastebin e cole abaixo o link aqui no exercício.

<https://pastebin.com/ca4KrvXp>

- Implemente a tabela **funcionario\_setor\_atuacao** com os campos: id (inteiro, chave primária, não nulo, auto incremento), nome\_setor (texto, não nulo), ativo (booleano, não nulo). Cole abaixo o script SQL.

**CREATE** **TABLE** funcionario\_setor\_atuacao(

id **INT** **PRIMARY** **KEY** **NOT** **NULL** **AUTO\_INCREMENT**,

nome\_setor **VARCHAR**(70) **NOT** **NULL**,

ativo **BOOL** **NOT** **NULL**

);

- Insira os setores: Tecnologia da Informação, Contabilidade, Administração. Cole o script de insert no pastebin e cole abaixo o link aqui no exercício.

<https://pastebin.com/MR7Xjp1q>

Agora que ampliamos nosso banco, vamos implementar algumas consultas. Para cada consulta cole abaixo o script SQL.

- Gere uma query que exiba uma coluna contendo os cargo existentes, e ao lado uma coluna contendo a quantidade de funcionários que ocupam cada cargo. Ordene pela quantidade de funcionários (cargo que tiver mais funcionário aparecendo primeiro...)

**SELECT** fc.nome\_cargo **AS** cargos, **COUNT**(\*) **AS** qtd\_funcionario

**FROM** funcionario\_cargo **AS** fc

**INNER** **JOIN** funcionario f **ON** f.funcionario\_cargo\_id = fc.id

**GROUP** **BY** fc.nome\_cargo

**ORDER** **BY** qtd\_funcionario **DESC**;

- Gere um relatório contendo as colunas: nome do funcionário, cargo do funcionário, setor de atuação do funcionário.

**SELECT** f.nome **AS** nome, fc.nome\_cargo **AS** cargo, s.nome\_setor **AS** setor

**FROM** funcionario\_cargo **AS** fc

**INNER** **JOIN** funcionario f **ON** f.funcionario\_cargo\_id = fc.id

**INNER** **JOIN** funcionario\_setor\_atuacao s **ON** s.nome\_setor = s.nome\_setor;

DESAFIO:

- Gere um relatório contendo duas colunas: setor de atuação, funcionário mais antigo do setor. Considere que a data de admissão do funcionário é a mesma data que ele ingressou no setor.

**SELECT** f.nome, f.data\_emissao, fc.nome\_cargo, fsa.nome\_setor

**FROM** funcionario f

**INNER** **JOIN** funcionario\_cargo fc **ON** fc.id = f.funcionario\_cargo\_id

**INNER** **JOIN** funcionario\_setor\_atuacao fsa **ON** fsa.id = fc.funcionario\_setor\_atuacao\_id

**WHERE** data\_emissao = (**SELECT** **min**(f.data\_emissao))

**ORDER** **BY** f.data\_emissao **ASC**;