

# Atividade SQL

🕒 Created	@March 22, 2023 10:18 AM
🏷️ Tags	

Considere o seguinte modelo relacional:

Empresas (id, nome, localização)

Produtos (codigo, nome, preço, id->empresas)

Apresente o respectivo script de criação das tabelas.

```
CREATE TABLE Empresas(  
    id INTEGER,  
    nome CHAR(20),  
    localizacao CHAR(20),  
    PRIMARY KEY(id),  
);  
  
CREATE TABLE Produtos(  
    codigo INTEGER,  
    nome CHAR(20),  
    preco REAL,  
    PRIMARY KEY (codigo),  
    FOREIGN KEY(id) REFERENCES Empresas  
);
```

1. Considere o seguinte modelo relacional e apresente as interrogações SQL correspondentes:  
Colaboradores (codigo, primeiroNome, ultimoNome, DataNasc, codigodept->Departamentos, Endereço, Cidade, Salario)  
Departamentos(codigodept, nome, localização, codigoColabGerente)

```
CREATE TABLE Colaboradores(  
    codigo INTEGER,  
    primeiroNome CHAR(20),  
    ultimoNome CHAR(20),  
    dataNasc CHAR(10),  
    FOREIGN KEY (codigoDept) REFERENCES Departamentos,
```

```

        endereco CHAR(20),
        cidade CHAR(20),
        salario REAL
        PRIMARY KEY(codigo)
    );

CREATE TABLE Departamentos(
    codigoDept INTEGER,
    nome CHAR(20),
    localizacao CHAR(20),
    codigoColabGerent INTEGER,
    PRIMARY KEY(codigoDept),
    PRIMARY KEY(codigoColabGerent)
);

```

a) Liste o nome completo e cidade dos colaboradores que têm salário superior a 1000€ ordenados pelo nome completo.

```

SELECT primeiroNome, ultimoNome, cidade
FROM Colaboradores
WHERE salario > 1000
ORDER BY primeiroNome+ultimoNome;

```

b) Liste o total e a média dos salários de todos os colaboradores da empresa.

```

SELECT SUM(salario),AVG(salario)
FROM Colaboradores;

```

c) Mostre a data de nascimento dos colaboradores cujo apelido termina na letra “a”.

```

SELECT dataNasc
FROM Colaboradores
WHERE primeiroNome LIKE '%a';

```

d) Liste o primeiro nome dos colaboradores que não têm o último nome registado na

## BD. Ordene

a informação apresentada alfabeticamente por ordem inversa.

```
SELECT primeiroNome
FROM Colaboradores
WHERE ultimoNome is null
ORDER BY primeiroNome DESC;
```

1. Considere as seguintes tabelas:

Alunos (id, nomeA, idade, cod->cursos)

Cursos (cod, nomeC, dataCriado, maxCandid)

a) Crie o script da tabela Alunos.

```
CREATE TABLE Alunos(
    id INTEGER,
    nomeA CHAR(20),
    idade INTEGER,
    FOREIGN KEY(cod) REFERENCES Cursos
    PRIMARY KEY(id)
);

CREATE TABLE Cursos(
    cod INTEGER,
    nomeC CHAR(20),
    dataCriado CHAR(20),
    maxCandid INTEGER,
    PRIMARY KEY(cod)
);
```

b) Selecione o nome e idade de todos os alunos. Ordene o resultado pela idade.

```
SELECT nomeA, idade
FROM Alunos
ORDER BY idade ASC;
```

c) Selecione a idade de todos os alunos.

```
SELECT idade  
FROM Alunos;
```

d) Apresente a média da idade de todos os alunos

```
SELECT AVG(idade)  
FROM Alunos;
```

e) Selecione o nome e idade do aluno mais velho.

```
SELECT MAX(idade)  
FROM Alunos;
```

f) Selecione o nome dos cursos em que os alunos com mais de 20 anos de idade se encontram inscritos.

```
SELECT nomeA, nomeC  
FROM Alunos, Cursos  
WHERE idade > 20;
```

g) Apague todos os registos dos alunos para os quais não se saiba o nome e tenham mais de 30 anos de idade

```
DELETE  
FROM Alunos  
WHERE nomeA is null  
AND idade>30;
```

h) Remova a tabela Alunos da Base de Dados

```
DELETE Alunos;
```