### **EXERCÍCIOS DE SQL 4**

Acesse o site <https://sqlfiddle.com/sqlite/online-compiler> para começar a programar em sqlite!

**EXERCÍCIO 1**

CREATE TABLE Alunos (

id INTEGER PRIMARY KEY,

nome TEXT,

turma TEXT,

nota REAL

);

INSERT INTO Alunos (id, nome, turma, nota) VALUES

(1, 'Ana', 'A', 8.5),

(2, 'Bruno', 'A', 5.0),

(3, 'Carla', 'A', 6.0),

(4, 'Diego', 'B', 7.5),

(5, 'Eduarda', 'B', 4.0),

(6, 'Felipe', 'B', 9.0),

(7, 'Giovana', 'C', 6.5),

(8, 'Henrique', 'C', 3.5),

(9, 'Isabela', 'C', 8.0),

(10, 'Joao', 'C', 5.5);

**Exercícios:**

1. Mostre a média das notas para cada turma.

2. Liste apenas as turmas cuja média de notas seja maior que 6.

3. Para cada aluno, informe nome, turma, nota e se está 'Aprovado' (nota >= 6) ou 'Reprovado'.

4. Conte quantos alunos foram aprovados e quantos foram reprovados.

5. Conte quantos alunos existem em cada turma.

**Resposta:**

**--1. Mostre a média das notas para cada turma.**

SELECT turma, AVG(nota) FROM Alunos GROUP BY turma;

**--2. Liste apenas as turmas cuja média de notas seja maior que 6.**

SELECT turma, AVG(nota) FROM Alunos GROUP BY turma HAVING AVG(nota) > 6;

**--3. Para cada aluno, informe nome, turma, nota e se está 'Aprovado' (nota >= 6) ou 'Reprovado'.**

SELECT nome, turma, nota,

CASE

WHEN nota >= 6 THEN 'Aprovado!'

ELSE 'Reprovado!'

END AS Situacao

FROM Alunos;

**--4. Conte quantos alunos foram aprovados e quantos foram reprovados.**

SELECT

CASE

WHEN nota >= 6 THEN 'Aprovado!'

ELSE 'Reprovado!'

END AS Situacao,

COUNT(\*) AS Quantidade

FROM Alunos

GROUP BY Situacao

;

**--5. Conte quantos alunos existem em cada turma.**

SELECT turma,

COUNT(\*) AS Quantidade

FROM Alunos

GROUP BY turma

;

**EXERCÍCIO 2**

CREATE TABLE Produtos (

id INTEGER PRIMARY KEY,

nome TEXT,

categoria TEXT,

preco REAL,

estoque INTEGER

);

INSERT INTO Produtos (id, nome, categoria, preco, estoque) VALUES

(1, 'Caneta', 'Papelaria', 2.50, 100),

(2, 'Lapis', 'Papelaria', 1.50, 200),

(3, 'Caderno', 'Papelaria', 15.00, 80),

(4, 'Mouse', 'Informatica', 50.00, 40),

(5, 'Teclado', 'Informatica', 120.00, 30),

(6, 'Monitor', 'Informatica', 900.00, 10),

(7, 'Borracha', 'Papelaria', 1.20, 150),

(8, 'Pen Drive', 'Informatica', 70.00, 25),

(9, 'Livro', 'Educacao', 35.00, 60),

(10, 'Agenda', 'Papelaria', 18.00, 20);

**Exercícios:**

1. Mostre o preço médio dos produtos para cada categoria.

2. Liste apenas as categorias cujo preço médio seja maior que R$ 50.

3. Para cada produto, mostre nome, categoria, preço e classifique-o como:

'Barato' (preço até 20),

'Intermediário' (preço entre 21 e 100)

'Caro' (preço acima de 100).

4. Conte quantos produtos existem em cada categoria de preço (Barato, Intermediário, Caro).

5. Conte quantos produtos existem em cada categoria.

**EXERCÍCIO 3**

CREATE TABLE Funcionarios (

id INTEGER PRIMARY KEY,

nome TEXT,

setor TEXT,

salario REAL

);

INSERT INTO Funcionarios (id, nome, setor, salario) VALUES

(1, 'Carlos', 'Vendas', 3000),

(2, 'Fernanda', 'Vendas', 3500),

(3, 'Julia', 'RH', 2800),

(4, 'Lucas', 'TI', 4000),

(5, 'Mariana', 'TI', 4500),

(6, 'Pedro', 'RH', 2700),

(7, 'Rafael', 'Vendas', 3200),

(8, 'Sofia', 'TI', 4300);

**Exercícios:**

1. Mostre o salário médio por setor.  
2. Liste apenas os setores com salário médio maior que R$ 3500.  
3. Classifique os funcionários como 'Bem Pago' (salário > 4000) ou 'Regular' (salário <= 4000).  
4. Conte quantos funcionários são 'Bem Pago' e 'Regular'.  
5. Conte quantos funcionários há em cada setor.

**EXERCÍCIO 4**

CREATE TABLE Pedidos (

id INTEGER PRIMARY KEY,

cliente TEXT,

data TEXT,

valor REAL

);

INSERT INTO Pedidos (id, cliente, data, valor) VALUES

(1, 'Ana', '2024-01-10', 250.00),

(2, 'Bruno', '2024-01-15', 120.00),

(3, 'Carla', '2024-02-02', 300.00),

(4, 'Diego', '2024-02-10', 75.00),

(5, 'Ana', '2024-02-20', 180.00),

(6, 'Bruno', '2024-03-01', 450.00),

(7, 'Fernanda', '2024-03-05', 600.00),

(8, 'Carla', '2024-03-12', 90.00),

(9, 'Ana', '2024-03-25', 230.00),

(10, 'Diego', '2024-04-01', 120.00);

1. Mostre o valor total dos pedidos por cliente.  
2. Liste apenas os clientes com valor total de pedidos maior que R$ 500.  
3. Para cada pedido, mostre cliente, data, valor e uma classificação:  
 ‘Baixo' (até 100),  
 'Médio' (de 101 a 300),  
 'Alto' (acima de 300).  
4. Conte quantos pedidos há em cada categoria de valor (Baixo, Médio, Alto).

**EXERCÍCIO 5**

CREATE TABLE Filmes (

id INTEGER PRIMARY KEY,

titulo TEXT,

genero TEXT,

ano INTEGER,

avaliacao REAL

);

INSERT INTO Filmes (id, titulo, genero, ano, avaliacao) VALUES

(1, 'Matrix', 'Ficcao Cientifica', 1999, 8.7),

(2, 'Titanic', 'Romance', 1997, 7.9),

(3, 'O Senhor dos Aneis', 'Fantasia', 2001, 8.8),

(4, 'Vingadores: Ultimato', 'Acao', 2019, 8.4),

(5, 'Parasita', 'Drama', 2019, 8.6),

(6, 'Toy Story', 'Animacao', 1995, 8.3),

(7, 'Coringa', 'Drama', 2019, 8.5),

(8, 'Pantera Negra', 'Acao', 2018, 7.3),

(9, 'Divertida Mente', 'Animacao', 2015, 8.1),

(10, 'A Origem', 'Ficcao Cientifica', 2010, 8.8);

1. Mostre a avaliação média dos filmes por gênero.

2. Liste apenas os gêneros com avaliação média maior que 8.

3. Para cada filme, mostre título, gênero, ano, avaliação e uma classificação:

'Ruim' (avaliação < 7),

'Boa' (entre 7 e 8.4),

'Excelente' (8.5 ou mais).

4. Conte quantos filmes existem em cada categoria de avaliação (Ruim, Boa, Excelente).