### **EXERCÍCIOS DE SQL 9**

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

**QUESTÃO 1**

-- CRIAÇÃO DAS TABELAS

CREATE TABLE clientes (

id INTEGER PRIMARY KEY,

nome TEXT,

idade INTEGER,

cidade TEXT

);

CREATE TABLE pedidos (

id INTEGER PRIMARY KEY,

cliente\_id INTEGER,

produto TEXT,

quantidade INTEGER,

preco\_unitario REAL,

data TEXT,

FOREIGN KEY (cliente\_id) REFERENCES clientes(id)

);

-- INSERÇÃO DE DADOS

INSERT INTO clientes (id, nome, idade, cidade) VALUES

(1, 'Ana', 22, 'Sao Paulo'),

(2, 'Bruno', 35, 'Rio de Janeiro'),

(3, 'Carla', 17, 'Belo Horizonte'),

(4, 'Daniel', 68, 'Sao Paulo'),

(5, 'Eduarda', 45, 'Porto Alegre');

INSERT INTO pedidos (id, cliente\_id, produto, quantidade, preco\_unitario, data) VALUES

(1, 1, 'Camiseta', 2, 50.0, '2024-01-10'),

(2, 1, 'Calça', 1, 120.0, '2024-01-15'),

(3, 2, 'Tênis', 1, 200.0, '2024-02-05'),

(4, 3, 'Boné', 3, 30.0, '2024-03-01'),

(5, 4, 'Jaqueta', 1, 300.0, '2024-03-10'),

(6, 4, 'Camiseta', 2, 50.0, '2024-03-15'),

(7, 5, 'Tênis', 2, 200.0, '2024-04-01');

**EXERCÍCIOS**

1. Selecione os nomes e idades dos clientes que moram em "São Paulo" e têm mais de 30 anos.

Dica: WHERE, operadores de comparação e lógicos

2. Liste todos os pedidos mostrando nome do cliente, produto e valor total (quantidade \* preco\_unitario).

Dica: INNER JOIN, operadores matemáticos, aliases

3. Liste os clientes que **não** fizeram nenhum pedido.

Dica: Subconsulta com NOT EXISTS

4. Mostre a quantidade total de pedidos feitos por cada cliente.

Dica: GROUP BY, COUNT, INNER JOIN

5. Exiba os nomes dos clientes e uma classificação com base na idade: “Menor de idade", "Adulto" ou "Idoso".

Dica: SELECT com CASE

6. Liste os nomes dos clientes e os produtos comprados ordenados pela data do pedido (mais recente primeiro).

Dica: ORDER BY com JOIN

7. Mostre os produtos cujo valor total da venda (quantidade \* preco\_unitario) foi maior que 150.

Dica: WHERE com operação matemática

8. Liste os nomes dos clientes que compraram "Tênis".

Dica: JOIN com filtro WHERE

9. Calcule o valor total gasto por cada cliente.

Dica: GROUP BY, SUM, JOIN, operadores matemáticos

**QUESTÃO 2**

-- CRIAÇÃO DAS TABELAS

CREATE TABLE funcionarios (

id INTEGER PRIMARY KEY,

nome TEXT,

idade INTEGER,

departamento TEXT,

salario REAL,

cidade TEXT

);

CREATE TABLE projetos (

id INTEGER PRIMARY KEY,

nome\_projeto TEXT,

funcionario\_id INTEGER,

horas\_trabalhadas INTEGER,

valor\_hora REAL,

FOREIGN KEY (funcionario\_id) REFERENCES funcionarios(id)

);

-- INSERÇÃO DE DADOS

INSERT INTO funcionarios (id, nome, idade, departamento, salario, cidade) VALUES

(1, 'Ana', 28, 'TI', 4500, 'Sao Paulo'),

(2, 'Bruno', 35, 'Financeiro', 5200, 'Rio de Janeiro'),

(3, 'Carla', 42, 'RH', 4800, 'Belo Horizonte'),

(4, 'Daniel', 30, 'TI', 4700, 'Sao Paulo'),

(5, 'Eduarda', 25, 'Marketing', 4000, 'Curitiba');

INSERT INTO projetos (id, nome\_projeto, funcionario\_id, horas\_trabalhadas, valor\_hora) VALUES

(1, 'Intranet', 1, 40, 50),

(2, 'ERP', 2, 35, 60),

(3, 'Treinamento', 3, 20, 55),

(4, 'Website', 4, 30, 65),

(5, 'Campanha', 5, 25, 45),

(6, 'App Mobile', 1, 50, 70);

**EXERCÍCIOS**

1. Liste o nome e salário dos funcionários que trabalham no departamento de "TI" e têm salário acima de 4500.

2. Mostre o nome do projeto, o nome do funcionário e o valor total do projeto (horas\_trabalhadas \* valor\_hora).

3. Conte quantos projetos cada funcionário participou. Mostre o nome do funcionário e a quantidade de projetos.

4. Classifique os funcionários como "Júnior" (idade < 30), "Pleno" (30 a 40), ou "Sênior" (mais de 40).

5. Mostre os nomes dos funcionários que participaram de projetos com mais de 40 horas.

6. Calcule o total de horas trabalhadas por cada funcionário em todos os seus projetos.

7. Liste os nomes e cidades dos funcionários que moram fora de São Paulo.

8. Mostre os nomes dos funcionários e os nomes dos projetos que comecem com a letra 'A'.  
 (Obs: Esse último usa apenas LIKE simples, sem LOWER).

**EXERCÍCIO 3**

-- Criação da tabela de produtos

CREATE TABLE produtos (

id INTEGER PRIMARY KEY,

nome TEXT,

preco REAL,

categoria TEXT,

estoque INTEGER

);

-- Criação da tabela de vendas

CREATE TABLE vendas (

id INTEGER PRIMARY KEY,

produto\_id INTEGER,

quantidade INTEGER,

data\_venda TEXT,

FOREIGN KEY (produto\_id) REFERENCES produtos(id)

);

-- Inserção de dados na tabela de produtos

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

(1, 'Notebook', 3000.00, 'Eletronicos', 10),

(2, 'Smartphone', 2000.00, 'Eletronicos', 20),

(3, 'Cafeteira', 250.00, 'Eletrodomesticos', 15),

(4, 'Livro', 50.00, 'Livros', 100),

(5, 'Fone de ouvido', 150.00, 'Eletronicos', 25);

-- Inserção de dados na tabela de vendas

INSERT INTO vendas (id, produto\_id, quantidade, data\_venda) VALUES

(1, 1, 2, '2023-08-01'),

(2, 2, 1, '2023-08-02'),

(3, 3, 3, '2023-08-03'),

(4, 5, 4, '2023-08-04'),

(5, 4, 5, '2023-08-05'),

(6, 1, 1, '2023-08-06');

**EXERCÍCIOS:**

1. Selecione os produtos da categoria 'Eletrônicos' com preço acima de 1000.

2. Liste o nome do produto, quantidade vendida e o valor total da venda (quantidade \* preco).

3. Mostre o total de vendas (em reais) por categoria.

4. Classifique os produtos como 'Baixo estoque', 'Médio estoque' ou 'Alto estoque' com base na seguinte lógica:

Estoque < 10 = 'Baixo', entre 10 e 20 = 'Médio', > 20 = 'Alto'

5. Mostre os nomes dos produtos vendidos no dia '2023-08-01'.

6. Liste os produtos com valor total de vendas superior a 3000.

7. Mostre a soma total de vendas (em reais) por produto.

8. Selecione os produtos cujo nome começa com 'C'.

9. Liste os produtos cujo preço está entre 100 e 500.

**EXERCÍCIO 4**

-- Criação da tabela de cursos

CREATE TABLE cursos (

id INTEGER PRIMARY KEY,

nome TEXT,

area TEXT,

duracao\_horas INTEGER,

preco REAL

);

-- Criação da tabela de matriculas

CREATE TABLE matriculas (

id INTEGER PRIMARY KEY,

curso\_id INTEGER,

aluno TEXT,

data\_matricula TEXT,

FOREIGN KEY (curso\_id) REFERENCES cursos(id)

);

-- Inserção de dados na tabela de cursos

INSERT INTO cursos (id, nome, area, duracao\_horas, preco) VALUES

(1, 'Python Basico', 'Programacao', 40, 300.00),

(2, 'HTML e CSS', 'Web Design', 30, 200.00),

(3, 'Excel Avancado', 'Produtividade', 25, 250.00),

(4, 'SQL para Iniciantes', 'Banco de Dados', 35, 280.00),

(5, 'JavaScript', 'Programacao', 45, 350.00);

-- Inserção de dados na tabela de matrículas

INSERT INTO matriculas (id, curso\_id, aluno, data\_matricula) VALUES

(1, 1, 'Carlos', '2023-07-10'),

(2, 2, 'Ana', '2023-07-11'),

(3, 3, 'Bruno', '2023-07-12'),

(4, 1, 'Daniela', '2023-07-13'),

(5, 4, 'Fernanda', '2023-07-14'),

(6, 5, 'Carlos', '2023-07-15');

**EXERCÍCIOS:**

1. Liste os cursos da área de 'Programação' com duração maior que 40 horas.

2. Mostre o nome do curso e o nome do aluno matriculado.

3. Calcule o total arrecadado (em reais) por curso.

4. Classifique os cursos como 'Curto', 'Médio' ou 'Longo': até 30h = Curto, entre 31 e 40 = Médio, acima de 40h = Longo

5. Liste os nomes dos alunos matriculados no curso de 'SQL para Iniciantes'.

6. Calcule a média de preços dos cursos por área.

7. Liste os cursos cujo nome começa com 'P'.

9. Liste os cursos cujo preço é menor ou igual a 250.

**EXERCÍCIO 5**

-- Criação da tabela de livros

CREATE TABLE livros (

id INTEGER PRIMARY KEY,

titulo TEXT,

autor TEXT,

genero TEXT,

preco REAL,

estoque INTEGER

);

-- Criação da tabela de pedidos

CREATE TABLE pedidos (

id INTEGER PRIMARY KEY,

livro\_id INTEGER,

quantidade INTEGER,

data\_pedido TEXT,

FOREIGN KEY (livro\_id) REFERENCES livros(id)

);

-- Inserção de dados na tabela de livros

INSERT INTO livros (id, titulo, autor, genero, preco, estoque) VALUES

(1, 'Dom Casmurro', 'Machado de Assis', 'Romance', 40.00, 50),

(2, '1984', 'George Orwell', 'Ficcao', 60.00, 30),

(3, 'O Pequeno Principe', 'Antoine de Saint-Exupery', 'Fabula', 35.00, 70),

(4, 'It: A Coisa', 'Stephen King', 'Terror', 90.00, 20),

(5, 'Clean Code', 'Robert C. Martin', 'Programacao', 120.00, 15);

-- Inserção de dados na tabela de pedidos

INSERT INTO pedidos (id, livro\_id, quantidade, data\_pedido) VALUES

(1, 1, 2, '2023-08-10'),

(2, 2, 1, '2023-08-11'),

(3, 5, 1, '2023-08-12'),

(4, 4, 3, '2023-08-13'),

(5, 3, 2, '2023-08-14');

**EXERCÍCIOS:**

1. Liste os livros do gênero 'Ficção' com preço acima de 50.

2. Mostre o título do livro, quantidade pedida e o valor total (preço \* quantidade).

3. Calcule o total vendido por gênero.

4. Classifique os livros como 'Pouco', 'Médio' ou 'Muito' estocados: estoque < 20 = Pouco, entre 20 e 50 = Médio, > 50 = Muito

5. Liste os livros pedidos no dia '2023-08-13'.

6. Liste os livros cujo total de vendas (preço \* quantidade) ultrapassa 100.

7. Calcule a soma total de pedidos (em reais) por livro.

8. Liste os livros cujo título começa com a letra 'C'.

9. Liste os livros cujo preço está entre 30 e 90.

**EXERCÍCIO 6**

-- Criação da tabela de planos

CREATE TABLE planos (

id INTEGER PRIMARY KEY,

nome TEXT,

tipo TEXT,

duracao\_meses INTEGER,

preco\_mensal REAL

);

-- Criação da tabela de assinaturas

CREATE TABLE assinaturas (

id INTEGER PRIMARY KEY,

plano\_id INTEGER,

cliente TEXT,

data\_assinatura TEXT,

FOREIGN KEY (plano\_id) REFERENCES planos(id)

);

-- Inserção de dados na tabela de planos

INSERT INTO planos (id, nome, tipo, duracao\_meses, preco\_mensal) VALUES

(1, 'Basico', 'Mensal', 1, 100.00),

(2, 'Intermediario', 'Trimestral', 3, 90.00),

(3, 'Avancado', 'Semestral', 6, 80.00),

(4, 'Premium', 'Anual', 12, 70.00),

(5, 'Personalizado', 'Mensal', 1, 150.00);

-- Inserção de dados na tabela de assinaturas

INSERT INTO assinaturas (id, plano\_id, cliente, data\_assinatura) VALUES

(1, 1, 'Joao', '2023-06-01'),

(2, 2, 'Mariana', '2023-06-02'),

(3, 3, 'Paulo', '2023-06-03'),

(4, 4, 'Juliana', '2023-06-04'),

(5, 5, 'Lucas', '2023-06-05'),

(6, 3, 'Bruna', '2023-06-06');

**EXERCÍCIOS:**

1. Liste os planos do tipo 'Mensal' com preço acima de 100.

2. Mostre o nome do plano e o nome do cliente que o assinou.

3. Calcule o valor total de cada assinatura (preco\_mensal \* duracao).

4. Classifique os planos como 'Curto', 'Médio' ou 'Longo': até 3 meses = Curto, 4 a 6 = Médio, acima de 6 = Longo

5. Liste os nomes dos clientes que assinaram o plano 'Avançado'.

6. Calcule a média de preço mensal por tipo de plano.

7. Liste os planos cujo nome começa com 'P'.

8. Liste os planos cujo preço mensal está entre 80 e 120.

**EXERCÍCIO 7**

-- Criação da tabela de livros

CREATE TABLE livros (

id INTEGER PRIMARY KEY,

titulo TEXT,

autor TEXT,

preco REAL,

estoque INTEGER,

categoria TEXT

);

-- Criação da tabela de vendas\_livros

CREATE TABLE vendas\_livros (

id INTEGER PRIMARY KEY,

livro\_id INTEGER,

quantidade INTEGER,

data\_venda TEXT,

FOREIGN KEY (livro\_id) REFERENCES livros(id)

);

-- Inserção de dados

INSERT INTO livros (id, titulo, autor, preco, estoque, categoria) VALUES

(1, 'Dom Casmurro', 'Machado de Assis', 30.00, 40, 'Romance'),

(2, '1984', 'George Orwell', 45.00, 20, 'Ficção'),

(3, 'Clean Code', 'Robert C. Martin', 120.00, 10, 'Tecnologia'),

(4, 'A Arte da Guerra', 'Sun Tzu', 25.00, 35, 'Filosofia'),

(5, 'O Hobbit', 'J.R.R. Tolkien', 60.00, 15, 'Fantasia');

INSERT INTO vendas\_livros (id, livro\_id, quantidade, data\_venda) VALUES

(1, 1, 3, '2023-06-01'),

(2, 2, 2, '2023-06-02'),

(3, 3, 1, '2023-06-03'),

(4, 5, 4, '2023-06-04'),

(5, 4, 2, '2023-06-05');

**EXERCÍCIOS:**

1. Liste os livros da categoria "Ficção" com preço entre 40 e 70.

2. Mostre o título, quantidade vendida e o total por venda.

3. Mostre a soma total de vendas por autor.

4. Classifique o estoque em 'Baixo', 'Médio' e 'Alto'.

5. Liste os livros vendidos no dia '2023-06-01'.

6. Liste os livros cujo total de vendas ultrapassou R$ 100.

7. Mostre a soma total de vendas por livro.

8. Liste os livros cujo título começa com a letra 'C'.

9. Liste os livros com preço abaixo de 50.

**EXERCÍCIO 8**

-- Criação da tabela de softwares

CREATE TABLE softwares (

id INTEGER PRIMARY KEY,

nome TEXT,

tipo TEXT,

preco REAL,

licencas\_disponiveis INTEGER

);

-- Criação da tabela de compras

CREATE TABLE compras (

id INTEGER PRIMARY KEY,

software\_id INTEGER,

quantidade INTEGER,

data\_compra TEXT,

FOREIGN KEY (software\_id) REFERENCES softwares(id)

);

-- Inserção de dados

INSERT INTO softwares (id, nome, tipo, preco, licencas\_disponiveis) VALUES

(1, 'Photoshop', 'Design', 500.00, 10),

(2, 'Excel Pro', 'Produtividade', 200.00, 15),

(3, 'Antivirus Plus', 'Seguranca', 100.00, 30),

(4, 'Editor de Vídeo', 'Multimidia', 350.00, 5),

(5, 'IDE Java', 'Desenvolvimento', 0.00, 50);

INSERT INTO compras (id, software\_id, quantidade, data\_compra) VALUES

(1, 1, 2, '2023-05-01'),

(2, 2, 3, '2023-05-02'),

(3, 3, 4, '2023-05-03'),

(4, 4, 1, '2023-05-04'),

(5, 2, 2, '2023-05-05');

**EXERCÍCIOS:**

1. Liste os softwares com tipo "Produtividade" e preço acima de 150.

2. Mostre o nome do software, quantidade comprada e o total gasto.

3. Mostre o total de compras por tipo.

4. Classifique o número de licenças como 'Poucas', 'Normais' ou 'Muitas'.

5. Liste os softwares comprados no dia '2023-05-03'.

6. Liste os softwares com total de compras superior a R$ 500.

7. Mostre o total por software comprado.

8. Liste os softwares com nome começando com 'E'.

9. Liste os softwares gratuitos.

**EXERCÍCIO 9**

-- Criação da tabela de professores

CREATE TABLE professores (

id INTEGER PRIMARY KEY,

nome TEXT,

materia TEXT,

valor\_hora REAL

);

-- Criação da tabela de aulas

CREATE TABLE aulas (

id INTEGER PRIMARY KEY,

professor\_id INTEGER,

aluno TEXT,

horas INTEGER,

data\_aula TEXT,

FOREIGN KEY (professor\_id) REFERENCES professores(id)

);

-- Inserção de dados

INSERT INTO professores (id, nome, materia, valor\_hora) VALUES

(1, 'Lucas', 'Matematica', 80.00),

(2, 'Juliana', 'Ingles', 70.00),

(3, 'Marcos', 'Historia', 60.00),

(4, 'Aline', 'Portugues', 90.00),

(5, 'Paulo', 'Fisica', 100.00);

INSERT INTO aulas (id, professor\_id, aluno, horas, data\_aula) VALUES

(1, 1, 'Ana', 2, '2023-04-10'),

(2, 2, 'Bruno', 3, '2023-04-11'),

(3, 3, 'Carlos', 1, '2023-04-12'),

(4, 4, 'Ana', 2, '2023-04-13'),

(5, 1, 'Daniel', 3, '2023-04-14');

**EXERCÍCIOS:**

1. Liste os professores de "Física" com valor por hora acima de 90.

2. Mostre o nome do professor, aluno e o total da aula (horas \* valor).

3. Mostre o total arrecadado por professor.

4. Classifique o valor/hora como 'Barato', 'Médio' ou 'Caro'.

5. Liste os alunos com aula no dia '2023-04-13'.

6. Liste os professores com mais de R$ 200 arrecadados.

7. Mostre a soma total por professor.

8. Liste os professores cujo nome começa com 'J'.

9. Liste os professores com valor/hora até R$ 70.