### **EXERCÍCIOS DE SQL 5**

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

**EXERCÍCIO 1**

CREATE TABLE ClientesOnline (

id INTEGER PRIMARY KEY,

nome TEXT

);

CREATE TABLE ClientesLojaFisica (

id INTEGER PRIMARY KEY,

nome TEXT

);

INSERT INTO ClientesOnline (id, nome) VALUES

(1, 'Ana'),

(2, 'Bruno'),

(3, 'Carla'),

(4, 'Diego');

INSERT INTO ClientesLojaFisica (id, nome) VALUES

(1, 'Bruno'),

(2, 'Diego'),

(3, 'Eduarda'),

(4, 'Fernanda');

**Perguntas:**

1. Liste todos os nomes de clientes únicos (sem duplicatas) considerando os dois canais.

2. Liste todos os nomes de clientes, incluindo duplicatas se houver.

3. Liste os nomes dos clientes que compraram tanto online quanto na loja física.

4. Liste os nomes dos clientes que compraram apenas online.

5. Liste os nomes dos clientes que compraram apenas na loja física.

**EXERCÍCIO 2**

CREATE TABLE Candidatos2024 (

id INTEGER PRIMARY KEY,

nome TEXT

);

CREATE TABLE Candidatos2025 (

id INTEGER PRIMARY KEY,

nome TEXT

);

INSERT INTO Candidatos2024 (id, nome) VALUES

(1, 'Joao'),

(2, 'Lucas'),

(3, 'Marina'),

(4, 'Paula');

INSERT INTO Candidatos2025 (id, nome) VALUES

(1, 'Lucas'),

(2, 'Paula'),

(3, 'Rafael'),

(4, 'Tatiane');

**Perguntas:**

1. Liste todos os candidatos únicos que participaram em qualquer um dos anos.

2. Liste os candidatos que participaram dos dois anos.

3. Liste os candidatos que participaram somente em 2024.

4. Liste os candidatos que participaram somente em 2025.

5. Liste todos os nomes com repetições (se tiverem participado em ambos).

**EXERCÍCIO 3**

CREATE TABLE ListaEmailPromocoes (

id INTEGER PRIMARY KEY,

email TEXT

);

CREATE TABLE ListaEmailNoticias (

id INTEGER PRIMARY KEY,

email TEXT

);

INSERT INTO ListaEmailPromocoes (id, email) VALUES

(1, 'ana@email.com'),

(2, 'bruno@email.com'),

(3, 'carla@email.com');

INSERT INTO ListaEmailNoticias (id, email) VALUES

(1, 'bruno@email.com'),

(2, 'diego@email.com'),

(3, 'carla@email.com');

**Perguntas:**

1. Mostre todos os e-mails cadastrados em pelo menos uma lista, sem duplicatas.

2. Mostre todos os e-mails cadastrados, inclusive duplicatas se houver.

3. Mostre os e-mails que estão em ambas as listas.

4. Mostre os e-mails que estão somente na lista de promoções.

5. Mostre os e-mails que estão somente na lista de notícias.

**EXERCÍCIO 4**

CREATE TABLE ProfessoresCursoA (

id INTEGER PRIMARY KEY,

nome TEXT

);

CREATE TABLE ProfessoresCursoB (

id INTEGER PRIMARY KEY,

nome TEXT

);

INSERT INTO ProfessoresCursoA (id, nome) VALUES

(1, 'Carlos'),

(2, 'Fernanda'),

(3, 'Juliana');

INSERT INTO ProfessoresCursoB (id, nome) VALUES

(1, 'Carlos'),

(2, 'Tatiane'),

(3, 'Pedro');

**Perguntas:**

1. Liste os nomes de todos os professores únicos que atuam em pelo menos um curso.

2. Liste os professores que atuam em ambos os cursos.

3. Liste os professores que atuam apenas no Curso A.

4. Liste os professores que atuam apenas no Curso B.

5. Liste todos os professores com repetições, se houver.

**EXERCÍCIO 5**

CREATE TABLE FilmesAssistidos2023 (

id INTEGER PRIMARY KEY,

titulo TEXT

);

CREATE TABLE FilmesAssistidos2024 (

id INTEGER PRIMARY KEY,

titulo TEXT

);

INSERT INTO FilmesAssistidos2023 (id, titulo) VALUES

(1, 'Matrix'),

(2, 'Titanic'),

(3, 'Avatar');

INSERT INTO FilmesAssistidos2024 (id, titulo) VALUES

(1, 'Matrix'),

(2, 'Oppenheimer'),

(3, 'Barbie');

**Perguntas:**

1. Liste todos os filmes únicos assistidos em qualquer um dos anos.

2. Liste os filmes assistidos nos dois anos.

3. Liste os filmes assistidos apenas em 2023.

4. Liste os filmes assistidos apenas em 2024.

5. Liste todos os filmes com repetições, se tiverem sido vistos mais de uma vez.