### **EXERCÍCIOS DE SQLite3 5**

## **Exercício 1 – Sistema de Biblioteca**

**Banco**: biblioteca.db  
**Tabela**: livros

### **Passos:**

1. Crie o banco biblioteca.db e conecte.
2. Crie uma tabela livros com os campos:  
   * id (inteiro, chave primária)
   * titulo (texto)
   * autor (texto)
   * ano (inteiro)
   * disponivel (booleano, 0 ou 1)
3. Insira 3 livros manualmente no código.
4. Permita que o usuário insira um novo livro via input().
5. Liste todos os livros.
6. Liste apenas os livros disponíveis.
7. Liste todos os livros publicados antes do ano 2000.
8. Liste os livros cujo autor começa com a letra "J".
9. Atualize um livro para ficar indisponível.
10. Delete um livro com base no título.

## **Exercício 2 – Controle de Vendas**

**Banco**: loja.db  
**Tabelas**: produtos, vendas

### **Passos:**

1. Crie o banco loja.db e conecte.
2. Crie a tabela produtos com:  
   * id, nome, preco (real)
3. Crie a tabela vendas com:  
   * id, produto\_id (chave estrangeira), quantidade, data
4. Insira 3 produtos fixos.
5. Permita que o usuário registre uma nova venda via input().
6. Mostre todos os produtos.
7. Mostre todas as vendas com nome do produto (JOIN).
8. Liste as vendas feitas em uma data específica (use LIKE '2025-05%').
9. Mostre o total vendido (em valor R$) por produto (agrupamento + SUM()).
10. Mostre o produto mais vendido (em quantidade).

## **Exercício 3 – Sistema de Cursos Online**

**Banco**: cursos.db  
**Tabelas**: cursos, alunos, inscricoes

### **Passos:**

1. Crie o banco cursos.db e conecte.
2. Crie as tabelas:  
   * cursos(id, nome, carga\_horaria)
   * alunos(id, nome, email)
   * inscricoes(id, curso\_id, aluno\_id)
3. Insira 2 cursos e 2 alunos.
4. Permita que o usuário inscreva um aluno em um curso via input().
5. Liste todos os cursos.
6. Liste os alunos e os cursos em que estão inscritos.
7. Liste todos os alunos que não estão inscritos em nenhum curso.
8. Liste todos os cursos com mais de 1 inscrito.
9. Mostre a quantidade de alunos por curso.
10. Permita remover uma inscrição via input() (informando nome do aluno e curso).

## 

## 

## 

## 

## **Exercício 4 – Cadastro de Filmes**

**Banco**: cinema.db  
**Tabela**: filmes

### **Passos:**

1. Crie o banco cinema.db e conecte.
2. Crie a tabela filmes(id, titulo, genero, duracao, classificacao)  
   * duracao em minutos, classificacao é a faixa etária (ex: 12, 14, 18).
3. Insira 3 filmes.
4. Peça para o usuário inserir um novo filme via input().
5. Liste todos os filmes.
6. Liste apenas os filmes do gênero “Ação”.
7. Liste os filmes com duração maior que 120 min.
8. Liste os filmes com classificação menor ou igual a 14 (para crianças).
9. Liste os filmes ordenados por duração decrescente.
10. Mostre a média de duração dos filmes cadastrados.

## **Exercício 5 – Sistema de Tarefas**

**Banco**: tarefas.db  
**Tabela**: tarefas

### **Passos:**

1. Crie o banco tarefas.db e conecte.
2. Crie a tabela tarefas(id, titulo, prioridade, concluida)  
   * prioridade: texto ("alta", "media", "baixa")
   * concluida: booleano 0 ou 1
3. Insira 3 tarefas com prioridade e status diferentes.
4. Peça para o usuário adicionar uma nova tarefa via input().
5. Liste todas as tarefas pendentes (concluida = 0).
6. Liste as tarefas ordenadas por prioridade (ordenar por CASE: alta > media > baixa).
7. Atualize uma tarefa para marcar como concluída (via input()).
8. Liste quantas tarefas estão pendentes.
9. Remova uma tarefa com base no título.
10. Mostre a porcentagem de tarefas concluídas (use COUNT() e cálculo).

## 

## **Exercício 6 – Sistema de Reservas de Hotel**

**Banco**: hotel.db  
**Tabelas**: quartos, reservas

### **Passos:**

1. Crie a tabela quartos(id, numero, tipo, preco\_diaria)  
   * tipo: "simples", "duplo", "luxo"
2. Crie a tabela reservas(id, quarto\_id, cliente, dias, data\_reserva)
3. Insira 3 quartos fixos.
4. Permita que o usuário registre uma nova reserva via input().
5. Liste todos os quartos disponíveis.
6. Liste as reservas com nome do cliente e tipo do quarto (JOIN).
7. Liste as reservas com mais de 5 dias.
8. Mostre o total arrecadado em reservas por tipo de quarto.
9. Liste o quarto mais reservado.
10. Liste as reservas feitas no mês atual.
11. .

## **Exercício 7 – Controle de Estoque**

**Banco**: estoque.db  
**Tabela**: produtos

### **Passos:**

1. Crie a tabela produtos(id, nome, quantidade, categoria, preco\_unitario)
2. Insira 4 produtos variados.
3. Peça para o usuário cadastrar um novo produto via input().
4. Liste todos os produtos com estoque abaixo de 10 unidades.
5. Liste os produtos da categoria “Limpeza”.
6. Liste os produtos ordenados por quantidade ascendente.
7. Atualize o estoque de um produto específico via input().
8. Calcule o valor total em estoque de cada produto (quantidade \* preco).
9. Liste os produtos mais caros (preco > 100).
10. Remova um produto com base no nome.

## 

## 

## 

## 

## 

## 

## **Exercício 8 – Catálogo de Músicas**

**Banco**: musicas.db  
**Tabelas**: musicas, artistas

### **Passos:**

1. Crie a tabela artistas(id, nome, nacionalidade)
2. Crie a tabela musicas(id, titulo, artista\_id, duracao)
3. Insira 3 artistas e 5 músicas.
4. Peça para o usuário cadastrar uma nova música.
5. Liste todas as músicas com nome do artista (JOIN).
6. Liste as músicas com mais de 4 minutos.
7. Liste as músicas de artistas brasileiros.
8. Calcule a duração total de músicas por artista.
9. Liste os artistas que têm mais de 2 músicas cadastradas.
10. Liste as 3 músicas mais curtas.

## 

## 

## 

## 

## 

## 

## 

## **Exercício 9 – Sistema de Atendimento Médico**

**Banco**: clinica.db  
**Tabelas**: pacientes, consultas

### **Passos:**

1. Crie pacientes(id, nome, idade, plano\_saude)
2. Crie consultas(id, paciente\_id, data, especialidade)
3. Insira 3 pacientes e 2 consultas.
4. Peça para o usuário cadastrar uma nova consulta.
5. Liste todas as consultas com nome do paciente.
6. Liste as consultas da especialidade “Cardiologia”.
7. Liste os pacientes que têm plano de saúde.
8. Liste os pacientes com mais de uma consulta marcada.
9. Liste as consultas do dia atual.
10. Remova uma consulta via input().

## 

## 

## 

## 

## 

## 

## 

## **Exercício 10 – Sistema de Eventos**

**Banco**: eventos.db  
**Tabelas**: eventos, participantes

### **Passos:**

1. Crie eventos(id, nome, local, data)
2. Crie participantes(id, nome, email, evento\_id)
3. Insira 2 eventos e 3 participantes.
4. Peça para o usuário cadastrar um novo participante.
5. Liste todos os eventos.
6. Liste todos os participantes com o nome do evento (JOIN).
7. Liste os participantes do evento “Python Summit”.
8. Liste os eventos com mais de 2 participantes.
9. Liste os participantes com email de domínio @gmail.com.
10. Permita editar o nome de um evento.

## 

## 

## 

## 

## 

## 

## 

## **Exercício 11 – Controle de Clientes e Pedidos**

**Banco**: comercio.db  
**Tabelas**: clientes, pedidos

### **Passos:**

1. Crie clientes(id, nome, cidade)
2. Crie pedidos(id, cliente\_id, produto, valor, data)
3. Insira 2 clientes e 3 pedidos.
4. Peça para o usuário registrar um novo pedido.
5. Liste todos os pedidos com nome do cliente.
6. Liste os pedidos da cidade “São Paulo”.
7. Liste os clientes que nunca fizeram pedidos.
8. Calcule o total gasto por cliente.
9. Liste os pedidos com valor acima de R$ 500.
10. Liste os pedidos realizados nos últimos 7 dias.

## 

## 

## 

## 

## 

## 

## 

## **Exercício 12 – Registro de Despesas Pessoais**

**Banco**: financas.db  
 **Tabela**: despesas

### **Passos:**

1. Crie despesas(id, descricao, categoria, valor, data)
2. Insira 4 despesas com diferentes categorias.
3. Peça para o usuário registrar uma nova despesa.
4. Liste todas as despesas do mês atual.
5. Liste as despesas da categoria “Alimentação”.
6. Liste as 3 despesas mais caras.
7. Calcule o total de gastos por categoria.
8. Calcule o total de despesas do mês.
9. Liste as despesas ordenadas por valor.
10. Permita remover uma despesa via input().

## **Exercício 13 – Sistema de Avaliações de Produtos**

**Banco**: avaliacoes.db  
**Tabelas**: produtos, avaliacoes

### **Passos:**

1. Crie produtos(id, nome, categoria)
2. Crie avaliacoes(id, produto\_id, nota, comentario)
3. Insira 3 produtos e 5 avaliações.
4. Peça para o usuário inserir uma nova avaliação.
5. Liste todas as avaliações com nome do produto.
6. Liste as avaliações com nota menor que 3.
7. Calcule a média de notas por produto.
8. Liste os produtos com média maior que 4.
9. Liste os comentários que contenham a palavra “excelente”.
10. Atualize um comentário específico via input().