**Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Banco de dados: PetShop**

* **Cliente**: nome, telefone, cpf
* **Animal**: nome, espécie, raça, dono (cliente)
* **Serviço**: tipo de serviço (banho, tosa, vacina), valor base
* **Agendamento**: animal, serviço, data, status (realizado, cancelado, pendente), valor final

CREATE DATABASE PetShop;

USE PetShop;

-- Cliente

CREATE TABLE Cliente (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100),

telefone VARCHAR(20),

cpf VARCHAR(15));

-- Animal

CREATE TABLE Animal (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100),

especie VARCHAR(50),

raca VARCHAR(50),

cliente\_id INT,

FOREIGN KEY (cliente\_id) REFERENCES Cliente(id));

-- Serviço

CREATE TABLE Servico (

id INT AUTO\_INCREMENT PRIMARY KEY,

tipo VARCHAR(50),

valor\_base DECIMAL(10, 2)

);

-- Agendamento

CREATE TABLE Agendamento (

id INT AUTO\_INCREMENT PRIMARY KEY,

animal\_id INT,

servico\_id INT,

data\_agendamento DATETIME,

status VARCHAR(20),

valor\_final DECIMAL(10, 2),

FOREIGN KEY (animal\_id) REFERENCES Animal(id),

FOREIGN KEY (servico\_id) REFERENCES Servico(id));

-- Clientes

INSERT INTO Cliente (nome, telefone, cpf) VALUES

('João Silva', '11999999999', '111.111.111-11'),

('Maria Souza', '11888888888', '222.222.222-22'),

('Carlos Lima', '11777777777', '333.333.333-33');

-- Animais

INSERT INTO Animal (nome, especie, raca, cliente\_id) VALUES

('Rex', 'Cachorro', 'Labrador', 1),

('Luna', 'Gato', 'Persa', 2),

('Thor', 'Cachorro', 'Poodle', 1),

('Mia', 'Gato', 'Siamês', 3);

-- Serviços

INSERT INTO Servico (tipo, valor\_base) VALUES

('Banho', 50.00),

('Tosa', 40.00),

('Vacina', 90.00);

-- Agendamentos

INSERT INTO Agendamento (animal\_id, servico\_id, data\_agendamento, status, valor\_final) VALUES

(1, 1, '2025-04-01 10:00:00', 'realizado', 50.00),

(1, 2, '2025-04-03 14:00:00', 'realizado', 40.00),

(2, 1, '2025-04-02 09:00:00', 'cancelado', 0.00),

(3, 3, '2025-04-05 11:30:00', 'pendente', 90.00),

(4, 1, '2025-04-06 08:30:00', 'realizado', 50.00);

**QUESTÕES**

Resolver as questões abaixo utilizando comandos SQL

1. Exibir o nome e a espécie dos animais do cliente João Silva
2. Exibir os serviços *realizados* pelos animais 1 e 3
3. Exibir o nome do animal e a quantidade total de serviços efetuados por cada animal (apenas os serviços *realizados*)
4. Exibir os dados dos agendamentos *realizados* ou *pendentes*
5. Exibir o nome dos clientes que têm animais com serviços *pendentes* (consulta aninhada)
6. Exibir os dados dos serviços com valor maior que *qualquer* valor final já pago. (Consulta aninhada)   
   Obs. Tem serviço com valor final zerado.
7. Exibir o nome dos animais que já realizaram o serviço mais caro (consulta aninhada)
8. Exibir o nome dos clientes que possuem mais de um animal com agendamento realizado (consulta aninhada)
9. Criação da view de histórico de agendamentos com o nome do dono, nome do animal, tipo de serviço, data do agendamento, status e valor final. Exibir o comando para executar a view.

**Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Banco de dados : Biblioteca**

* **Leitor**: nome, telefone, cpf
* **Livro**: titulo, genero, ano\_publicacao
* **Autor:** nome
* **Livro\_Autor:** livro\_id, autor\_id
* **Emprestimo:** leitor\_id, livro\_id, data\_emprestimo, data\_devolucao, status

CREATE DATABASE BibliotecaDB;

USE BibliotecaDB;

-- Leitor

CREATE TABLE Leitor (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100),

telefone VARCHAR(20),

cpf VARCHAR(14));

-- Livro

CREATE TABLE Livro (

id INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(150),

genero VARCHAR(50),

ano\_publicacao INT);

-- Autor

CREATE TABLE Autor (

id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100));

-- Livro\_Autor

CREATE TABLE Livro\_Autor (

livro\_id INT,

autor\_id INT,

PRIMARY KEY (livro\_id, autor\_id),

FOREIGN KEY (livro\_id) REFERENCES Livro(id),

FOREIGN KEY (autor\_id) REFERENCES Autor(id));

-- Empréstimo

CREATE TABLE Emprestimo (

id INT AUTO\_INCREMENT PRIMARY KEY,

leitor\_id INT,

livro\_id INT,

data\_emprestimo DATE,

data\_devolucao DATE,

status VARCHAR(20),

FOREIGN KEY (leitor\_id) REFERENCES Leitor(id),

FOREIGN KEY (livro\_id) REFERENCES Livro(id));

-- Leitores

INSERT INTO Leitor (nome, telefone, cpf) VALUES

('Ana Lima', '11999999999', '111.111.111-11'), ('Bruno Silva', '11888888888', '222.222.222-22'),

('Clara Rocha', '11777777777', '333.333.333-33');

-- Livros

INSERT INTO Livro (titulo, genero, ano\_publicacao) VALUES

('A Menina que Roubava Livros', 'Romance', 2007), ('O Hobbit', 'Fantasia', 1937),

('Dom Casmurro', 'Romance', 1899), ('1984', 'Ficção Científica', 1949);

-- Autores

INSERT INTO Autor (nome) VALUES

('Markus Zusak'), ('J.R.R. Tolkien'), ('Machado de Assis'), ('George Orwell');

-- Livro\_Autor

INSERT INTO Livro\_Autor (livro\_id, autor\_id) VALUES

(1, 1),(2, 2),(3, 3),(4, 4);

-- Empréstimos

INSERT INTO Emprestimo (leitor\_id, livro\_id, data\_emprestimo, data\_devolucao, status) VALUES

(1, 1, '2025-04-01', '2025-04-10', 'devolvido'),

(1, 3, '2025-04-03', NULL, 'pendente'),

(2, 2, '2025-04-05', '2025-04-12', 'devolvido'),

(3, 4, '2025-04-06', NULL, 'pendente'),

(3, 1, '2025-04-07', NULL, 'pendente');

**QUESTÕES**

Resolver as questões abaixo utilizando comandos SQL

1. Exibir o nome dos leitores que pegaram livros de romance
2. Exibir a quantidade de livros emprestados por leitor
3. Exibir títulos dos livros com empréstimos pendentes ou devolvidos
4. Exibir os dados dos Livros com ano de publicação maior que qualquer livro emprestado pelo leitor 1 (consultas aninhadas)
5. Exibir o nome dos autores cujos livros foram publicados antes de todos os livros do gênero 'Ficção Científica' (consultas aninhadas)
6. Listar os leitores que pegaram livros do mesmo gênero que o livro mais antigo (consultas aninhadas)
7. Listar o título dos livros emprestados por leitores que pegaram também o livro “1984” (consultas aninhadas)
8. Mostrar o nome dos autores cujos livros foram emprestados por mais de um leitor (consultas aninhadas)
9. Crie uma *view* chamada vw\_HistoricoEmprestimos com o objetivo de facilitar a visualização do histórico completo de empréstimos da biblioteca. Essa *view* deve consolidar informações de diferentes tabelas do banco de dados, apresentando de forma unificada os seguintes dados:

* Nome do leitor que realizou o empréstimo;
* Título e gênero do livro emprestado;
* Data de realização do empréstimo;
* Data de devolução (caso tenha sido devolvido);
* Status do empréstimo (por exemplo: "pendente", "devolvido").

Exibir o comando que consulta a view