**Passo a Passo — Banco de Dados SQL do DoctorLink**

**1. Definir os requisitos**

O sistema tem **dois tipos de usuários principais**:

* **Paciente**: pode se cadastrar, logar e marcar consultas.
* **Médico/Clínica**: pode se cadastrar, logar e oferecer serviços de saúde.

Outros pontos:

* Cada consulta liga **um paciente a um médico**, em uma data e horário.
* No futuro pode haver **receitas, exames, pagamentos** etc. (mas para o MVP basta cadastro/login + consultas).

**2. Modelagem Conceitual (DER Simplificado)**

Entidades principais:

* **Paciente (PacienteID, Nome, Email, Senha, DataNascimento)**
* **Medico (MedicoID, Nome, Email, Senha, Especialidade, CRM/Registro)**
* **Consulta (ConsultaID, PacienteID, MedicoID, DataHora, Status)**

Relacionamentos:

* Paciente **marca** Consulta.
* Médico/Clínica **atende** Consulta.

**3. Criar o Banco de Dados**

No **MySQL/MariaDB** (ou PostgreSQL, se preferirem).

**3. Criar o Banco de Dados**

No **MySQL/MariaDB** (ou PostgreSQL, se preferirem).

CREATE DATABASE doctorlink;

USE doctorlink;

**4. Criar Tabelas**

**🧑‍🦱 Paciente**

CREATE TABLE paciente (

paciente\_id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(150) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

senha VARCHAR(255) NOT NULL,

data\_nascimento DATE NOT NULL,

criado\_em TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**🩺 Médico / Clínica**

CREATE TABLE medico (

medico\_id INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(150) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

senha VARCHAR(255) NOT NULL,

especialidade VARCHAR(100) NOT NULL,

registro\_profissional VARCHAR(50), -- CRM, CRO, etc.

criado\_em TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**📅 Consulta**

CREATE TABLE consulta (

consulta\_id INT AUTO\_INCREMENT PRIMARY KEY,

paciente\_id INT NOT NULL,

medico\_id INT NOT NULL,

data\_hora DATETIME NOT NULL,

status ENUM('Agendada', 'Concluída', 'Cancelada') DEFAULT 'Agendada',

criado\_em TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (paciente\_id) REFERENCES paciente(paciente\_id),

FOREIGN KEY (medico\_id) REFERENCES medico(medico\_id)

);

**5. Inserir Dados de Teste**

-- Pacientes

INSERT INTO paciente (nome, email, senha, data\_nascimento) VALUES

('João Silva', 'joao@email.com', '123456', '1990-05-12'),

('Maria Souza', 'maria@email.com', '123456', '1985-09-23');

-- Médicos

INSERT INTO medico (nome, email, senha, especialidade, registro\_profissional) VALUES

('Dr. Carlos Lima', 'carlos@clinica.com', '123456', 'Cardiologia', 'CRM12345'),

('Clínica Bem Estar', 'contato@bemestar.com', '123456', 'Clínica Geral', 'CNPJ112233');

-- Consultas

INSERT INTO consulta (paciente\_id, medico\_id, data\_hora, status) VALUES

(1, 1, '2025-09-05 10:00:00', 'Agendada'),

(2, 2, '2025-09-06 14:30:00', 'Agendada');

**6. Consultas SQL Úteis para Testes**

**Listar pacientes cadastrados**

SELECT \* FROM paciente;

**Listar médicos e suas especialidades**

SELECT nome, especialidade FROM medico;

**Ver consultas agendadas (com nomes de paciente e médico)**

SELECT c.consulta\_id, p.nome AS paciente, m.nome AS medico, c.data\_hora, c.status

FROM consulta c

JOIN paciente p ON c.paciente\_id = p.paciente\_id

JOIN medico m ON c.medico\_id = m.medico\_id;

**7. Próximos Passos para os alunos**

* Criar **usuário no MySQL** específico para a aplicação (segurança).
* Implementar **processa\_cadastro.php** para salvar pacientes/médicos na tabela correta.
* Implementar **processa\_login.php** para validar email + senha.
* Conectar via **PDO (PHP Data Objects)** para evitar SQL Injection.