

# Trabalho de Banco de Dados

---

## *Sistema de Gestão de Clínicas Médicas (Saúde+)*

---

Autor: Gustavo Camerino de Carvalho

Data: Maio 2025

## 1 Introdução

A rede de clínicas médicas Saúde+ deseja informatizar seus processos de agendamento de consultas, controle de pacientes, procedimentos e pagamentos. Este documento apresenta a modelagem de um banco de dados relacional, incluindo a identificação de entidades, definição de atributos, relacionamentos, diagrama entidade-relacionamento (DER), modelo relacional, regras de integridade, inserções de dados e consultas SQL.

## 2 Modelagem Conceitual

### 2.1 Identificação das Entidades

As entidades do sistema são:

- Paciente
- Médico
- Consulta
- Procedimento
- Consulta\_Procedimento (tabela associativa)
- Pagamento
- Convênio
- Recepcionista

### 2.2 Definição dos Atributos

- Entidades e Atributos
  - Paciente: CPF (PK), Nome varchar(30), Data\_Nascimento DATE, Telefone varchar(30), Endereço varchar(30), Cod\_Convenio (FK, NULLABLE)
  - Médico: CRM (PK), Nome varchar(30), Especialidade varchar(30), Telefone varchar(30)

- Consulta: ID\_Consulta (PK), Data DATE, Hora TIME, Status varchar(20), CRM (FK), CPF\_Paciente (FK), CPF\_Recepcionista (FK)
- Procedimento:Codigo\_Procedimento (PK), Nome varchar(50), Valor INT, Tempo\_Estimado INT
- Consulta\_Procedimento: ID\_Consulta (PK, FK), Codigo\_Procedimento (PK, FK), Observacoes TEXT
- Pagamento: ID\_Pagamento (PK), ID\_Consulta (FK), Valor DECIMAL(10,2), Forma\_Pagamento varchar(20), Status varchar(20), Pagador varchar(30)
- Convênio: Cod\_Convenio (PK), Nome varchar(20), Percentual\_Cobertura DECIMAL(5,2)
- Recepcionista: CPF\_Recepcionista (PK), Nome varchar(20), Telefone varchar(20)

## 2.3 Relacionamentos e Cardinalidade

Relacionamento	Entidade A	Entidade B	Cardinalidade	Descrição
<b>Realiza</b>	Paciente (1)	Consulta (N)	1:N	Um paciente pode ter várias consultas.
<b>Atende</b>	Médico (1)	Consulta (N)	1:N	Um médico realiza várias consultas.
<b>Agenda</b>	Recepcionista (1)	Consulta (N)	1:N	Uma recepcionista agenda várias consultas.
<b>Inclui</b>	Consulta (N)	Procedimento (N)	N:M	Uma consulta pode incluir vários procedimentos, via

Relacionamento	Entidade A	Entidade B	Cardinalidade	Descrição
				Consulta_Procedimento.
<b>Paga</b>	Consulta (1)	Pagamento (N)	1:N	Uma consulta pode ter vários pagamentos.
<b>Possui</b>	Convênio (1)	Paciente (0..N)	1:N (opcional)	Um convênio cobre vários pacientes (de forma opcional).

• Relacionamentos:

- Paciente – (1:N) – Consulta
- Médico – (1:N) – Consulta
- Recepcionista – (1:N) – Consulta
- Consulta – (N:M) – Procedimento (via Consulta\_Procedimento)
- Consulta – (1:N) – Pagamento
- Convênio – (1:N) – Paciente (opcional)

**Chaves Estrangeiras:**

- Consulta: CPF\_Paciente (Paciente.CPF), CRM (Médico.CRM), CPF\_Recepcionista (Recepcionista.CPF\_Recepcionista)
- Paciente: Cod\_Convenio (Convênio.Cod\_Convenio, NULLABLE)
- Consulta\_Procedimento: ID\_Consulta (Consulta.ID\_Consulta),Codigo\_Procedimento (Procedimento.Codigo\_Procedimento)
- Pagamento: ID\_Consulta (Consulta.ID\_Consulta)

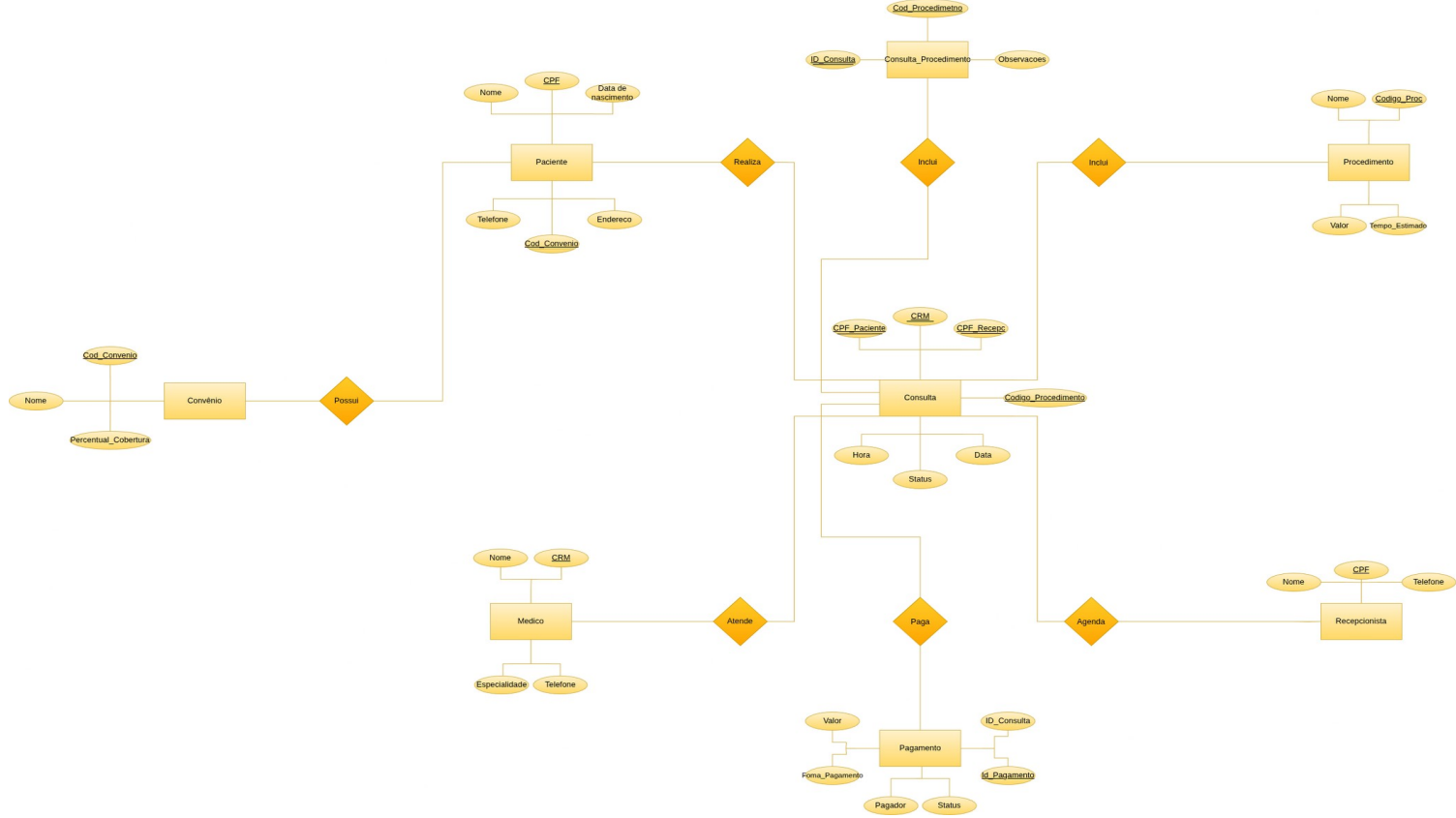


Figure 1: Diagrama Entidade-Relacionamento

### 3 Modelagem Relacional

As tabelas são normalizadas até a 3a Forma Normal (3NF):

- **Paciente** (CPF PK, Nome, Data\_Nascimento, Telefone, Endereço, Cod\_Convenio FK NULLABLE)
- **Convênio** (Cod\_Convenio PK, Nome, Percentual\_Cobertura)
- **Médico** (CRM PK, Nome, Especialidade, Telefone)
- **Recepcionista** (CPF\_Recepcionista PK, Nome, Telefone)
- **Consulta** (ID\_Consulta PK, Data, Hora, Status, CRM FK, CPF\_Paciente FK, CPF\_Recepcionista FK)
- **Procedimento** (Codigo\_Procedimento PK, Nome, Valor, Tempo\_Estimado)
- **Consulta\_Procedimento** (ID\_Consulta PK/FK, Codigo\_Procedimento PK/FK, Observacoes)
- **Pagamento** (ID\_Pagamento PK, ID\_Consulta FK, Valor, Forma\_Pagamento, Status, Pagador)

#### 3.1 Normalização

- 1NF: Atributos atômicos, sem grupos repetitivos.
- 2NF: Sem dependências parciais, pois as chaves primárias determinam todos os atributos.
- 3NF: Sem dependências transitivas; atributos não-chaves dependem apenas da chave primária

#### 3.2 Regras de Integridade

##### 1. Integridade de Entidade:

- Cada tabela possui uma PK única (CPF, CRM, ID\_Consulta, etc.).

##### 2. Integridade Referencial:

- Consulta.CPF\_Paciente → Paciente.CPF (NOT NULL)

- Consulta.CRM → Médico.CRM (NOT NULL)
- Consulta.CPF\_Recepcionista → Recepcionista.CPF\_Recepcionista (NOT NULL)
- Paciente.Cod\_Convenio → Convênio.Cod\_Convenio (NULLABLE)
- Consulta\_Procedimento.ID\_Consulta → Consulta.ID\_Consulta (NOT NULL)
- Consulta\_Procedimento.Codigo\_Procedimento → Procedimento.Codigo\_Procedimento (NOT NULL)
- Pagamento.ID\_Consulta → Consulta.ID\_Consulta (NOT NULL)

### **3. Integridade de Domínio:**

- Consulta.Status: {'agendada', 'concluída', 'cancelada'}
- Pagamento.Forma\_Pagamento: {'cartão', 'dinheiro', 'boleto'}
- Pagamento.Status: {'pago', 'pendente', 'cancelado'}
- Pagamento.Pagador: {'paciente', 'convênio'}

### **4. Regras de Negócio:**

- Consultas requerem paciente, médico e recepcionista (FKs NOT NULL).
- Pacientes podem não ter convênio (Cod\_Convenio NULLABLE).
- Procedimentos só podem ser vinculados a consultas/procedimentos válidos.

## 4 Inserções de Dados

Exemplos de inserções (mínimo de 10 registros):

### ◆ Inserções de Dados (mínimo de 10 registros)

#### Convenio

```
INSERT INTO Convenio (Cod_Convenio, Nome, Percentual_Cobertura) VALUES
('C001', 'Amil', 80.0),
('C002', 'Unimed', 70.0);
```

#### Paciente

```
INSERT INTO Paciente (CPF, Nome, Data_Nascimento, Telefone, Endereco,
Cod_Convenio) VALUES
('12345678901', 'Joao Silva', '1990-05-15', '11987654321', 'Rua A, 123',
'C001'),
('98765432109', 'Maria Oliveira', '1985-10-20', '11912345678', 'Rua B,
456', NULL),
('45678912345', 'Pedro Santos', '1978-03-12', '11955556666', 'Rua C,
789', 'C002');
```

#### Médico

```
INSERT INTO Medico (CRM, Nome, Especialidade, Telefone) VALUES
('CRM1234', 'Dr. Carlos Almeida', 'Cardiologia', '11977778888'),
('CRM5678', 'Dra. Ana Costa', 'Pediatria', '11999990000');
```

#### Recepcionista

```
INSERT INTO Recepcionista (CPF_Recepcionista, Nome, Telefone) VALUES
('11122233344', 'Clara Mendes', '11911112222');
```

#### Procedimento

```
INSERT INTO Procedimento (Codigo_Procedimento, Nome, Valor,
Tempo_Estimado) VALUES
('P001', 'Raio-X', 150.00, 30),
('P002', 'Exame de Sangue', 100.00, 15),
('P003', 'Consulta Geral', 200.00, 45);
```

### Consulta

```
INSERT INTO Consulta (ID_Consulta, Data, Hora, Status, CRM,
CPF_Paciente, CPF_Recepcionista) VALUES
(1, '2025-05-01', '10:00:00', 'concluida', 'CRM1234', '12345678901',
'11122233344'),
(2, '2025-05-01', '11:00:00', 'agendada', 'CRM5678', '98765432109',
'11122233344'),
(3, '2025-05-02', '14:00:00', 'concluida', 'CRM1234', '45678912345',
'11122233344');
```

### Consulta\_Procedimento

```
INSERT INTO Consulta_Procedimento (ID_Consulta, Codigo_Procedimento,
Observacoes) VALUES
(1, 'P001', 'Raio-X toracico'),
(1, 'P003', 'Consulta inicial'),
(3, 'P002', 'Hemograma completo'),
(3, 'P003', 'Consulta de acompanhamento');
```

### Pagamento

```
INSERT INTO Pagamento (ID_Pagamento, ID_Consulta, Valor,
Forma_Pagamento, Status, Pagador, CPF_Paciente) VALUES
(1, 1, 40.00, 'cartao', 'pago', 'paciente', '12345678901'),
(2, 1, 160.00, 'boleto', 'pago', 'convenio', NULL),
(3, 3, 300.00, 'dinheiro', 'pago', 'paciente', '45678912345');
```

## 5 – DDL

```
CREATE TABLE Convenio (
    Cod_Convenio VARCHAR(10) PRIMARY KEY,
    Nome VARCHAR(20) NOT NULL,
    Percentual_Cobertura DECIMAL(5,2) NOT NULL
);
```

```
CREATE TABLE Paciente (
    CPF VARCHAR(11) PRIMARY KEY,
    Nome VARCHAR(30) NOT NULL,
    Data_Nascimento DATE NOT NULL,
    Telefone VARCHAR(30) NOT NULL,
    Endereco VARCHAR(30) NOT NULL,
    Cod_Convenio VARCHAR(10),
    FOREIGN KEY (Cod_Convenio) REFERENCES Convenio(Cod_Convenio)
    ON DELETE SET NULL ON UPDATE CASCADE
);
```

```
CREATE TABLE Medico (
    CRM VARCHAR(10) PRIMARY KEY,
```



```

        Nome VARCHAR(30) NOT NULL,
        Especialidade VARCHAR(30) NOT NULL,
        Telefone VARCHAR(30) NOT NULL
    );

CREATE TABLE Recepcionista (
    CPF_Recepcionista VARCHAR(11) PRIMARY KEY,
    Nome VARCHAR(20) NOT NULL,
    Telefone VARCHAR(20) NOT NULL
);

CREATE TABLE Consulta (
    ID_Consulta INT PRIMARY KEY,
    Data DATE NOT NULL,
    Hora TIME NOT NULL,
    Status VARCHAR(20) CHECK (Status IN ('agendada', 'concluida',
'cancelada')) NOT NULL,
    CRM VARCHAR(10) NOT NULL,
    CPF_Paciente VARCHAR(11) NOT NULL,
    CPF_Recepcionista VARCHAR(11) NOT NULL,
    FOREIGN KEY (CRM) REFERENCES Medico(CRM)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CPF_Paciente) REFERENCES Paciente(CPF)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CPF_Recepcionista) REFERENCES
Recepcionista(CPF_Recepcionista)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Procedimento (
    Codigo_Procedimento VARCHAR(10) PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    Valor INT NOT NULL,
    Tempo_Estimado INT NOT NULL
);

CREATE TABLE Consulta_Procedimento (
    ID_Consulta INT,
    Codigo_Procedimento VARCHAR(10),
    Observacoes TEXT,
    PRIMARY KEY (ID_Consulta, Codigo_Procedimento),
    FOREIGN KEY (ID_Consulta) REFERENCES Consulta(ID_Consulta)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Codigo_Procedimento) REFERENCES
Procedimento(Codigo_Procedimento)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Pagamento (

```

```

        ID_Pagamento INT PRIMARY KEY,
        ID_Consulta INT NOT NULL,
        Valor DECIMAL(10,2) NOT NULL,
        Forma_Pagamento VARCHAR(20) CHECK (Forma_Pagamento IN ('cartao',
'dinheiro', 'boleto')) NOT NULL,
        Status VARCHAR(20) CHECK (Status IN ('pago', 'pendente',
'cancelado')) NOT NULL,
        Pagador VARCHAR(30) CHECK (Pagador IN ('paciente', 'convenio')) NOT
NULL,
        FOREIGN KEY (ID_Consulta) REFERENCES Consulta(ID_Consulta)
        ON DELETE CASCADE ON UPDATE CASCADE
    );

```

## 4 Consultas de Dados

### 1. Consultas agendadas para um médico em um dia específico

```

SELECT c.ID_Consulta, c.Data, c.Hora, c.Status, p.Nome AS Paciente
FROM Consulta c
JOIN Paciente p ON c.CPF_Paciente = p.CPF
WHERE c.CRM = 'CRM1234' AND c.Data = '2025-05-01' AND c.Status =
'agendada';

```

### 2. Consultas realizadas por médico no último mês

```

SELECT m.CRM, m.Nome, COUNT(*) AS Total_Consultas
FROM Consulta c
JOIN Medico m ON c.CRM = m.CRM
WHERE c.Data >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH) AND c.Status
= 'concluida'
GROUP BY m.CRM, m.Nome;

```

### 3. Pacientes sem consultas agendadas

```
SELECT p.CPF, p.Nome
FROM Paciente p
LEFT JOIN Consulta c ON p.CPF = c.CPF_Paciente
WHERE c.ID_Consulta IS NULL;
```

### 4. Procedimentos de uma consulta específica

```
SELECT p.Codigo_Procedimento, p.Nome, p.Valor, cp.Observacoes
FROM Consulta_Procedimento cp
JOIN Procedimento p ON cp.Codigo_Procedimento =
p.Codigo_Procedimento
WHERE cp.ID_Consulta = 1;
```

### 5. Total pago por paciente nos últimos 3 meses

```
SELECT p.CPF, p.Nome, SUM(pg.Valor) AS Total_Pago
FROM Paciente p
JOIN Pagamento pg ON pg.Pagador = 'paciente' AND pg.CPF_Paciente =
p.CPF
JOIN Consulta c ON pg.ID_Consulta = c.ID_Consulta
WHERE pg.Status = 'pago' AND c.Data >= DATE_SUB(CURDATE(),
INTERVAL 3 MONTH)
GROUP BY p.CPF, p.Nome;
```

### 6. Convênios com cobertura superior a 50%

```
SELECT Cod_Convenio, Nome, Percentual_Cobertura
FROM Convenio
WHERE Percentual_Cobertura > 50.0;
```

### 7. Pacientes atendidos por um médico

```
SELECT DISTINCT p.CPF, p.Nome
FROM Paciente p
JOIN Consulta c ON p.CPF = c.CPF_Paciente
WHERE c.CRM = 'CRM1234';
```

### 8. Faturamento total do último mês

```
SELECT SUM(pg.Valor) AS Faturamento_Total
FROM Pagamento pg
JOIN Consulta c ON pg.ID_Consulta = c.ID_Consulta
WHERE pg.Status = 'pago' AND c.Data >= DATE_SUB(CURDATE(),
INTERVAL 1 MONTH);
```

9. Médicos com mais de 5 consultas na próxima semana

```
SELECT m.CRM, m.Nome, COUNT(*) AS Total_Consultas
FROM Consulta c
JOIN Medico m ON c.CRM = m.CRM
WHERE c.Data BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 7
DAY)
    AND c.Status = 'agendada'
GROUP BY m.CRM, m.Nome
HAVING COUNT(*) > 5;
```

10. Pacientes que mais gastaram nos últimos 6 meses

```
SELECT p.CPF, p.Nome, SUM(pg.Valor) AS Total_Gasto
FROM Paciente p
JOIN Pagamento pg ON pg.Pagador = 'paciente' AND pg.CPF_Paciente =
p.CPF
JOIN Consulta c ON pg.ID_Consulta = c.ID_Consulta
WHERE pg.Status = 'pago' AND c.Data >= DATE_SUB(CURDATE(),
INTERVAL 6 MONTH)
GROUP BY p.CPF, p.Nome
ORDER BY Total_Gasto DESC
LIMIT 10;
```