



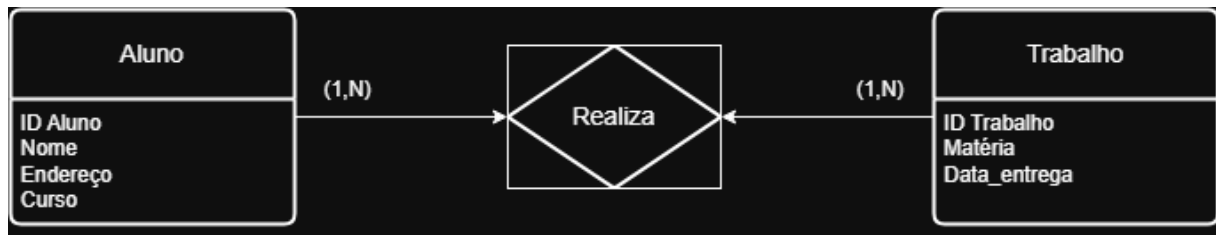
## CURSO: DESENVOLVIMENTO FULL STACK BÁSICO

### Atividade 01 - Banco de Dados

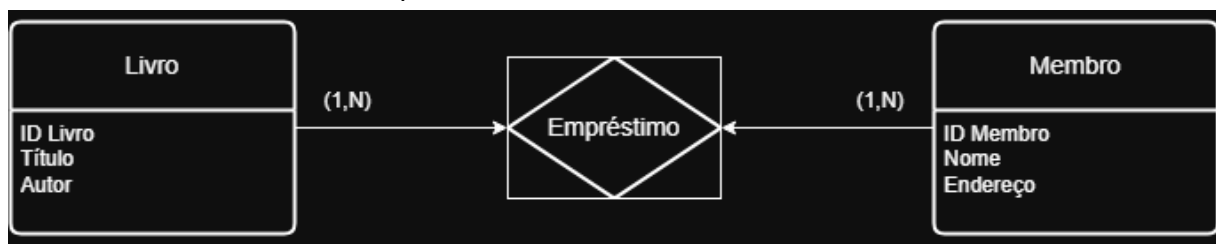
Aluna: Lara Corsini de Miranda

#### 1. Desenvolva o Diagrama Entidade Relacionamento para as seguintes situações:

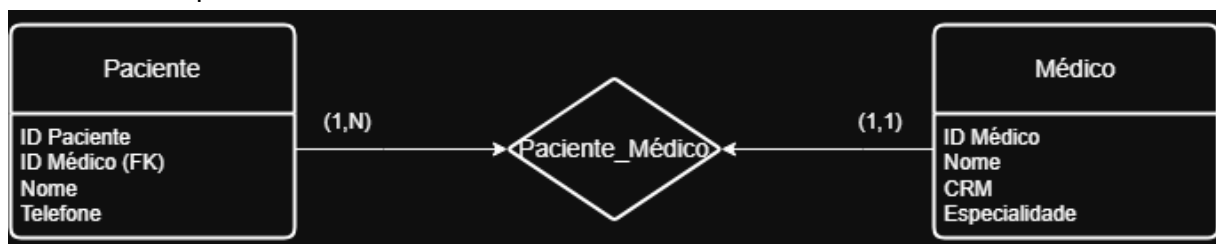
- 1.1. Um aluno realiza vários trabalhos. Um trabalho é realizado por um ou mais alunos.



- 1.2. Um livro pode ser emprestado para vários membros, e um membro pode ter vários livros emprestados.



- 1.3. Um médico atende a vários pacientes, enquanto cada paciente é atendido por um único médico.



2. Um berçário deseja informatizar suas operações. Quando um bebê nasce, algumas informações são armazenadas sobre ele, tais como: nome, data do nascimento, peso do nascimento, altura, a mãe deste bebê e o médico que fez seu parto. Para as mães, o berçário também deseja manter um controle, guardando informações como: nome, endereço, telefone e data de nascimento. Para os médicos, é importante saber: CRM, nome, telefone celular e especialidade.

- 2.1. Liste as entidades que você identificar no texto acima.

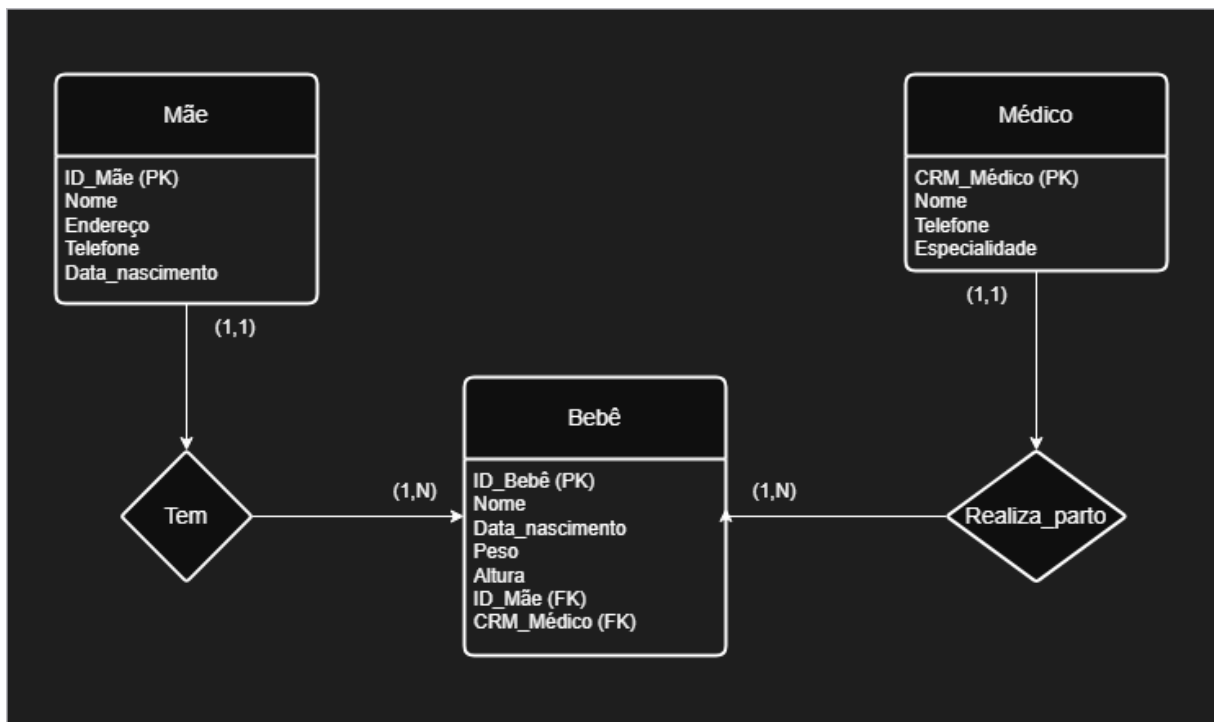
**Entidades:** Bebê, Mãe e Médico

- 2.2. Liste os relacionamentos entre as entidades

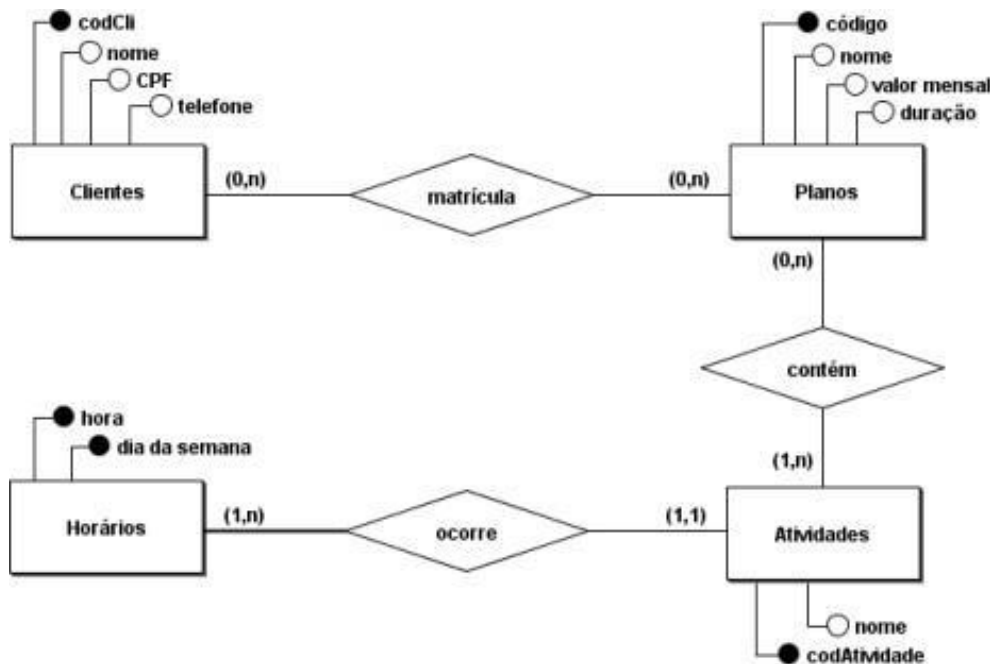
**Bebê -> Mãe:** Este será um relacionamento de um para muitos (1:N). Uma mãe pode ter vários bebês, mas um bebê só pode ter uma mãe.

**Bebê -> Médico:** Também será um relacionamento de um para muitos (1:N). Um médico pode fazer o parto de vários bebês, mas um bebê só será concebido por um médico.

- 2.3. Crie o modelo Entidade Relacionamento.



3. Elabore uma descrição em texto, em que você explica com todos os detalhes que conseguir observar no diagrama presente na imagem abaixo.



Descreva as entidades, os relacionamentos, os atributos e todas as restrições que estão representados neste diagrama.

#### Entidades:

- **Clientes - Atributos:** CodCli (chave primária), nome, CPF e telefone
- **Planos - Atributos:** código (chave primária), nome, valor mensal e duração
- **Horários - Atributos:** hora e dia da semana (chaves primárias compostas)
- **Atividades - Atributos:** codAtividade (chave primária) e nome

#### Relacionamentos:

- **Matrícula:** Relação entre Clientes e Planos. É um relacionamento de muitos para muitos, onde um cliente pode ter *nenhum* ou vários planos e um plano pode conter *nenhum* ou vários clientes. Nesse caso **não há restrições**.
- **Contém:** Relação entre Planos e Atividades. É um relacionamento de um para muitos com **restrição** na entidade Atividade, pois um plano pode conter nenhuma ou várias atividades, porém uma atividade precisa conter pelo menos um plano, também podendo estar em vários.
- **Ocorre:** Relacionamento entre Atividade e Horários. É um relacionamento de um para muitos com **restrição** na entidade Horários, pois uma atividade pode ocorrer em vários horários, porém um horário pertence somente à uma atividade.



4. Crie uma base de dados chamada **Biblioteca** com as tabelas a seguir:

- Livros (ISBN, título, autor, ano\_publicacao)
- Membros (ID\_MEMBRO, nome, endereço, cidade)
- Empréstimos (ID\_EMPRÉSTIMO, ISBN, ID\_MEMBRO, data\_empréstimo, data\_devolucao)

---

```
CREATE DATABASE Biblioteca;  
USE Biblioteca;
```

```
CREATE TABLE Livros (  
    ISBN VARCHAR(13) PRIMARY KEY,  
    título VARCHAR(255),  
    autor VARCHAR(255),  
    ano_publicacao INT  
);
```

```
CREATE TABLE Membros (  
    ID_MEMBRO INT PRIMARY KEY,  
    nome VARCHAR(255),  
    endereço VARCHAR(255),  
    cidade VARCHAR(100)  
);
```

```
CREATE TABLE Empréstimos (  
    ID_EMPRESTIMO INT PRIMARY KEY,  
    ISBN VARCHAR(13),  
    ID_MEMBRO INT,  
    data_emprestimo DATE,  
    data_devolucao DATE,  
    FOREIGN KEY (ISBN) REFERENCES Livros(ISBN),  
    FOREIGN KEY (ID_MEMBRO) REFERENCES Membros(ID_MEMBRO)  
);
```

OBS: Pra determinar o tamanho de caracteres pro ISBN, utilizei a quantidade informada pelo google.

O ISBN (International Standard Book Number) possui 13 dígitos desde 2007. Antes dessa data, o sistema utilizava 10 dígitos. [🔗](#)

Em um banco de dados real, seria interessante acrescentar o NOT NULL para alguns valores que devem ser obrigatórios de serem preenchidos para manter o registro de informações completo, como título e autor do livro por exemplo.

---



Insira os seguintes registros:

#### LIVROS:

- (978-3-16-148410-0, "O Senhor dos Anéis", "J.R.R. Tolkien", 1954)
- (978-0-262-54143-2, "A Arte da Guerra", "Sun Tzu", 500 a.C.)
- (978-0-06-112008-4, "O Sol é Para Todos", "Harper Lee", 1960)

---

**INSERT INTO** Livros (ISBN, titulo, autor, ano\_publicacao) **VALUES**  
(978-3-16-148410-0, 'O Senhor dos Anéis', 'J.R.R. Tolkien', 1954),  
(978-0-262-54143-2, 'A Arte da Guerra', 'Sun Tzu', -500),  
(978-0-06-112008-4, 'O Sol é Para Todos', 'Harper Lee', 1960);

---

#### MEMBROS:

- (1, "ANA SILVA", "RUA DOS SONHOS", "SÃO PAULO")
- (2, "CARLOS PEREIRA", "AVENIDA LIBERDADE", "RIO DE JANEIRO")
- (3, "MARIA OLIVEIRA", "RUA DA PAZ", "NATAL")

---

**INSERT INTO** Membros (ID\_MEMBRO, nome, endereço, cidade) **VALUES**  
(1, 'ANA SILVA', 'RUA DOS SONHOS', 'SÃO PAULO'),  
(2, 'CARLOS PEREIRA', 'AVENIDA LIBERDADE', 'RIO DE JANEIRO'),  
(3, 'MARIA OLIVEIRA', 'RUA DA PAZ', 'NATAL');

---

#### EMPRÉSTIMOS:

- (1, 978-3-16-148410-0, 1, '2023-01-10', '2023-01-17')
- (2, 978-0-262-54143-2, 2, '2023-01-12', '2023-01-19')
- (3, 978-0-06-112008-4, 3, '2023-01-15', NULL)

---

**INSERT INTO** Empréstimos (ID\_EMPRESTIMO, ISBN, ID\_MEMBRO, data\_emprestimo, data\_devolucao) **VALUES**  
(1, '978-3-16-148410-0', 1, '2023-01-10', '2023-01-17'),  
(2, '978-0-262-54143-2', 2, '2023-01-12', '2023-01-19'),  
(3, '978-0-06-112008-4', 3, '2023-01-15', NULL);

---



Crie queries SQL que respondam os cenários abaixo:

a) Encontre o título dos livros que foram emprestados a membros de São Paulo.

---

```
SELECT L.titulo FROM Livros L
JOIN Empréstimos E ON L.ISBN = E.ISBN
JOIN Membros M ON E.ID_MEMBRO = M.ID_MEMBRO
WHERE M.cidade = 'SÃO PAULO';
```

---

b) Encontre o número de empréstimos realizados por cada membro.

---

```
SELECT M.nome, COUNT(E.ID_EMPRESTIMO) AS n_emprestimos
FROM Membros M
LEFT JOIN Empréstimos E ON M.ID_MEMBRO = E.ID_MEMBRO
GROUP BY M.nome;
ORDER BY M.nome;
```

*LEFT JOIN usado para mostrar todos os membros, mesmo os que não têm empréstimos*  
*ORDER BY para que fique com resultado mais organizado*

---

c) Liste os membros que não devolveram livros até a data atual.

---

```
SELECT M.nome FROM Membros M
JOIN Empréstimos E ON M.ID_MEMBRO = E.ID_MEMBRO
WHERE E.data_devolucao IS NULL;
```

---

d) Encontre quantos livros foram publicados antes do ano 2000.

---

```
SELECT COUNT(*) AS livros_antes_2000 FROM Livros
WHERE ano_publicacao < 2000;
```

---

e) Liste todos os livros que ainda não foram emprestados.

---

```
SELECT L.titulo FROM Livros L
LEFT JOIN Empréstimos E ON L.ISBN = E.ISBN
WHERE E.ISBN IS NULL;
```

---



- f) Encontre o título do livro mais emprestado, ou seja, o que possui o maior número de empréstimos.

---

```
SELECT L.titulo FROM Livros L
JOIN Empréstimos E ON L.ISBN = E.ISBN
GROUP BY L.titulo
ORDER BY COUNT(E.ID_EMPRESTIMO) DESC
LIMIT 1;
```

---