

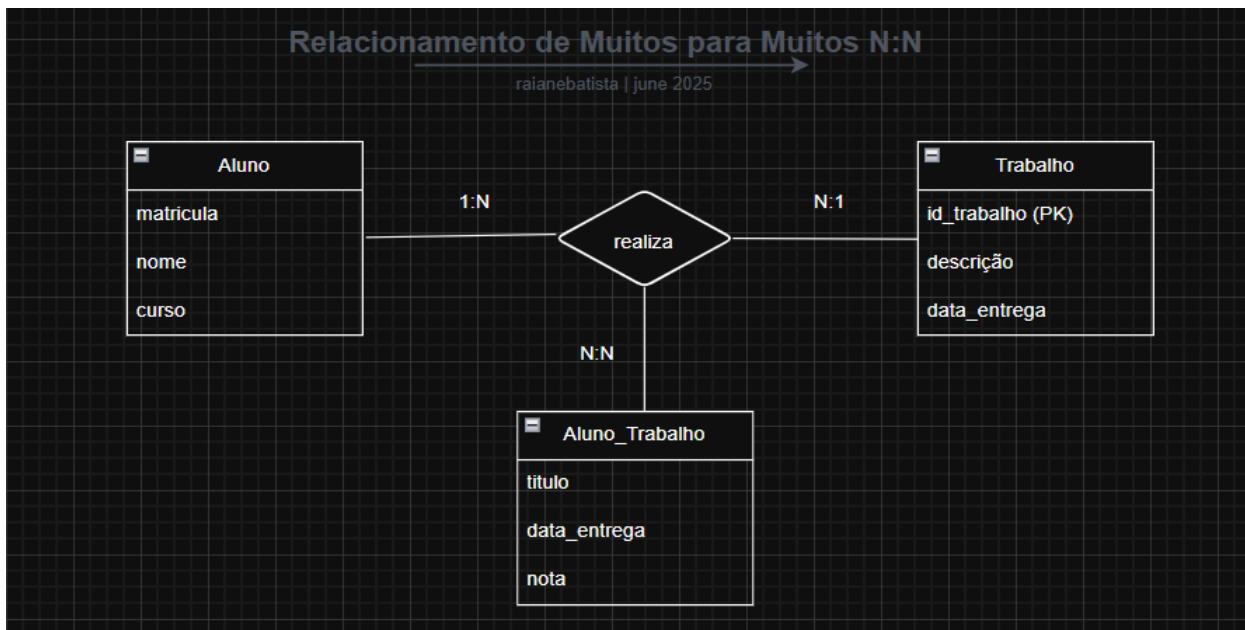
CURSO: DESENVOLVIMENTO FULL STACK BÁSICO

Atividade 01 - Banco de Dados

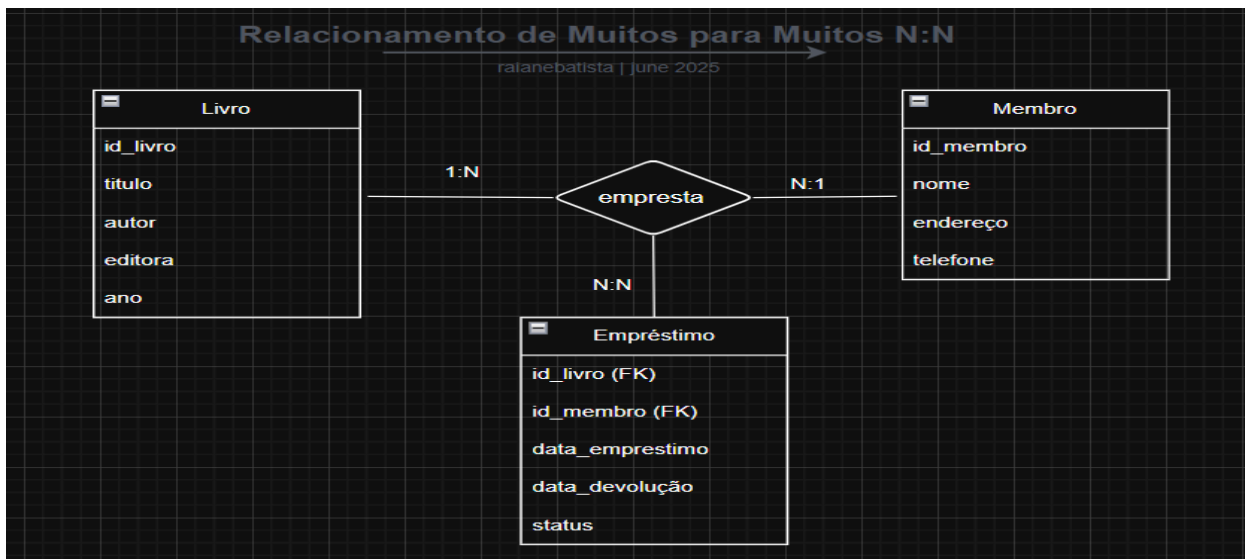
Aluna: Raiane Batista Silva Chaves

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.



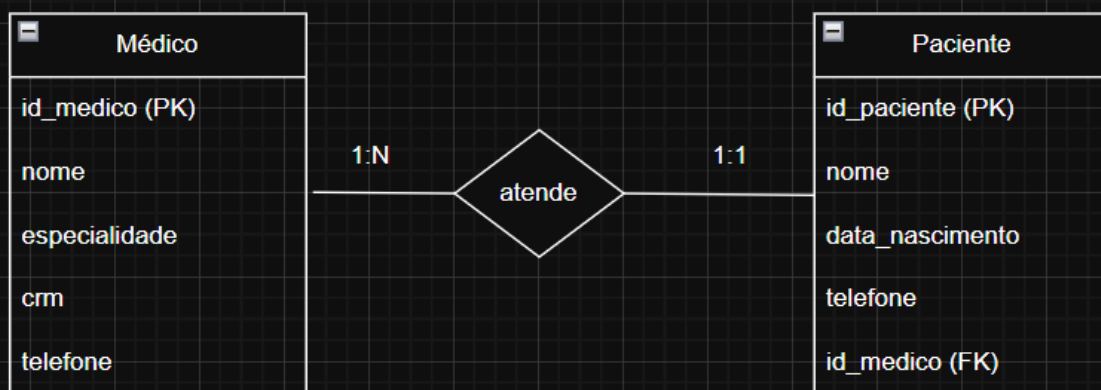
1.2 Um livro pode ser emprestado para vários membros, e um membro pode ter vários livros emprestados.



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

Relacionamento de Um para Muitos 1:N

raianebatista | june 2025



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 identificadas:

Bebê

Atributos: nome, data_nascimento, peso_nascimento, altura

Mãe:

Atributos: nome, endereço, telefone, data_nascimento

Médico:

Atributos: crm(PK), nome, telefone_celular, especialidade

2.2 Liste os relacionamentos entre as entidades

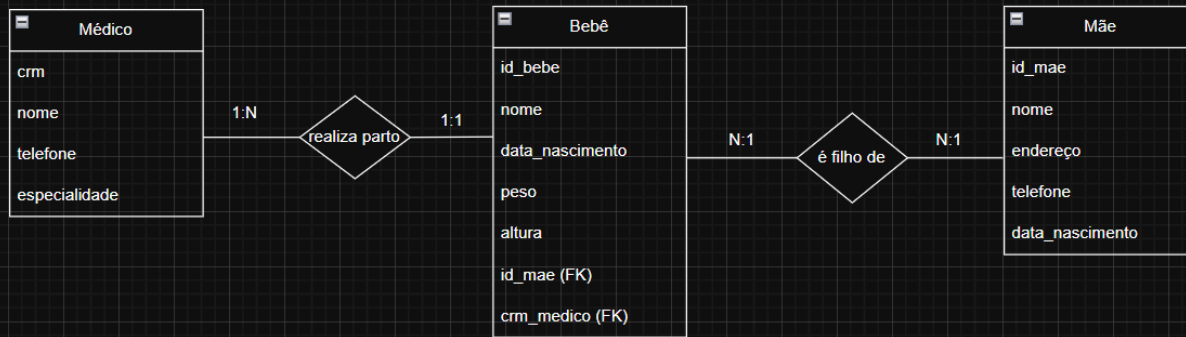
Relacionamentos identificados:

- Entre bebê e mamãe => N:1 (vários bebês podem ter a mesma mãe, mas cada bebê tem uma única mãe)
- Entre médico e bebê => 1:N (um médico pode realizar o parto de vários bebês, mas cada bebê tem seu parto realizado por um único médico)

2.3 Crie o modelo Entidade Relacionamento.

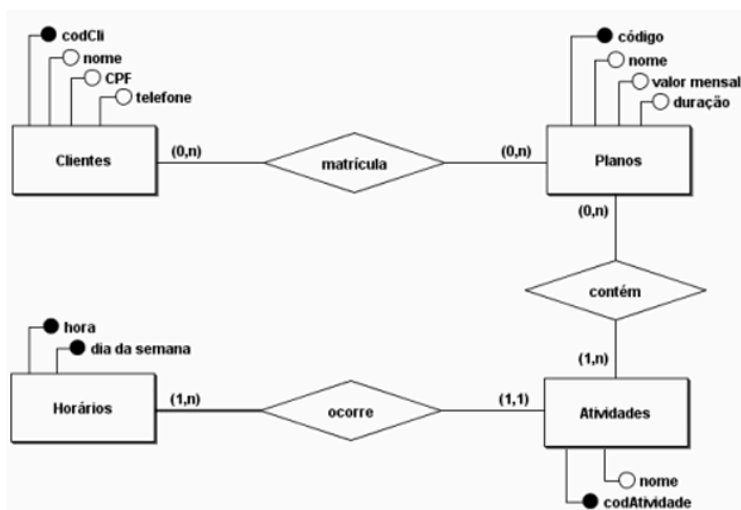
Relacionamento de Um para Muitos 1:N

raianebatista | June 2025



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.



No diagrama apresentado, podemos identificar entidades e seus atributos, sendo eles:

- **Clientes:**
Atributos: `codCli` (chave primária), `nome`, `cpf`, `telefone`
- **Planos:**
Atributos: `código` (chave primária), `nome`, `valor mensal`, `duração`
- **Horários:**
Atributos: `hora` (chave primária), `dia da semana`
- **Atividades:**
Atributos: `codAtividade` (chave primária), `nome`

Podemos observar os relacionamentos e as cardinalidades entre as entidades, nesse caso temos:

Matrícula - entre clientes e planos, sendo que:

Um cliente pode se matricular em nenhum ou vários planos (0,n)

Um plano pode ter nenhum ou vários clientes matriculados (0,n)

Aqui temos um relacionamento N:N (muitos para muitos), com essa associação, podemos ter uma tabela associativa na implementação do banco de dados.

Contém - entre planos e atividades, aqui temos:

Um plano pode conter uma ou mais atividades (1,n)

Uma atividade está associada a um e somente um plano (1,1)

Relacionamento 1:N (um para muitos), sendo:

Plano é o lado 1

Atividade é o lado N

Ocorre - entre Atividades e Horários

Uma atividade ocorre em um e somente um horário (1,1)

Um horário pode estar associado a uma ou mais atividades (1,n)

Relacionamento 1:N (um para muitos)

Horário é o lado 1

Atividade é o lado N

O relacionamento **matrícula** é opcional tanto para clientes quanto para planos, significa que é possível ter um cliente sem estar matriculado e um plano sem clientes, já o relacionamento **contém**, é obrigatório no lado da atividade, ou seja, toda atividade deve pertencer exatamente a um plano, o relacionamento **ocorre** também é obrigatório no lado da atividade, sendo assim, toda atividade precisa ocorrer em um horário específico.

Conclui-se que:

Clientes se matriculam em planos - relacionamento muitos para muitos => N,N

Planos, são compostos por uma ou mais atividades => relacionamento Um para muito 1,N

Atividade ocorre em um horário específico => relacionamento Um para muito 1,N

Horários podem ter várias atividades associadas.

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)

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)

MEMBROS:

- (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)

Crie queries SQL que respondam os cenários abaixo:

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

```
SELECT DISTINCT l.titulo
FROM emprestimos e
JOIN membros m ON e.id_membro = m.id_membro
JOIN livros l ON e.isbn = l.isbn
WHERE m.cidade = 'SÃO PAULO';
```

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

```
SELECT m.nome, COUNT(e.id_emprestimo) AS total_emprestimos
FROM membros m
LEFT JOIN emprestimos e ON m.id_membro = e.id_membro
GROUP BY m.nome
ORDER BY total_emprestimos DESC;
```

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

```
SELECT m.nome, l.titulo, e.data_emprestimo
FROM emprestimos e
JOIN membros m ON e.id_membro = m.id_membro
JOIN livros l ON e.isbn = l.isbn
WHERE e.data_devolucao IS NULL;
```

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

```
SELECT COUNT(*) AS total_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 emprestimos 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, COUNT(e.id_emprestimo) AS total_emprestimos
FROM emprestimos e
JOIN livros l ON e.isbn = l.isbn
GROUP BY l.titulo
ORDER BY total_emprestimos DESC
LIMIT 1;
```