

Introdução a Banco de dados

Atributos:

Os atributos são características ou opções adicionais que podem ser definidas ao criar colunas em uma tabela no MySQL. Eles permitem especificar regras, comportamentos e restrições para os dados armazenados nas colunas, garantindo integridade, validade e eficiência no banco de dados.

Os atributos no MySQL influenciam como os dados são armazenados, manipulados e acessados, oferecendo controle sobre aspectos como o tipo de dado, a obrigatoriedade de preenchimento, os valores padrão, os limites de tamanho e muito mais.

Atributos para Gerenciamento de Identificadores Únicos:

Esses atributos são usados para assegurar que cada registro tenha identificadores únicos e funcionem corretamente como chave primária ou em relacionamentos.

Atributos para Gerenciamento de Identificadores Únicos:

AUTO_INCREMENT

Descrição: Utilizado em campos numéricos para gerar automaticamente um número sequencial único a cada novo registro.

Característica: Geralmente usado em colunas de chave primária para garantir que cada linha tenha um identificador único sem precisar ser especificado manualmente.

Atributos para Gerenciamento de Identificadores Únicos:

AUTO_INCREMENT

Exemplo:

```
CREATE TABLE exemplo_auto_increment (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL  
);
```

Atributos para Gerenciamento de Identificadores Únicos:

PRIMARY KEY

Descrição: Define a chave primária da tabela, garantindo que cada linha seja identificável de forma única.

Características: Combina restrições de NOT NULL e UNIQUE.

Atributos para Gerenciamento de Identificadores Únicos:

PRIMARY KEY

Exemplo:

```
CREATE TABLE exemplo_primary_key (  
    id INT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL  
);
```

Atributos para Gerenciamento de Identificadores Únicos:

UNIQUE

Descrição: Garante que todos os valores da coluna sejam únicos no banco de dados.

Características: Para campos como CPF, emails ou qualquer identificador que não pode se repetir.

Atributos para Gerenciamento de Identificadores Únicos:

UNIQUE

Exemplo:

```
CREATE TABLE exemplo_unique (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    email VARCHAR(100) UNIQUE NOT NULL  
);
```


Atributos no MySQL

Atributos para Restrições de Dados:

Esses atributos são usados para controlar as condições e limitações dos dados armazenados nas colunas.

Atributos para Restrições de Dados:

NOT NULL

Descrição: Garante que a coluna não poderá conter valores nulos.

Características: Usado para campos essenciais onde você precisa garantir que sempre haja um valor válido

Atributos para Gerenciamento de Identificadores Únicos:

NOT NULL

Exemplo:

```
CREATE TABLE exemplo_not_null (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL  
);
```

Atributos para Restrições de Dados:

CHECK

Descrição: Define condições para restringir os dados que podem ser armazenados em uma coluna.

Características: Garantir que os dados atendam a regras específicas.

Atributos para Restrições de Dados:

CHECK

Exemplo:

```
CREATE TABLE exemplo_check (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idade INT,  
    CHECK (idade >= 18)  
);
```

Atributos para Restrições de Dados:

DEFAULT

Descrição: Define um valor padrão para uma coluna caso nenhum valor seja fornecido durante a inserção.

Características: Útil para fornecer valores padrão em casos onde a informação não for obrigatória.

Atributos para Restrições de Dados:

DEFAULT

Exemplo:

```
CREATE TABLE exemplo_default (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    status VARCHAR(20) DEFAULT 'ativo'  
);
```

Atributos no MySQL

Relacionamentos entre Tabelas:

Esses atributos facilitam a definição de chaves estrangeiras e relacionamentos entre tabelas.

Relacionamentos entre Tabelas:

FOREIGN KEY

Descrição: Define uma chave estrangeira para estabelecer um relacionamento entre tabelas.

Características: Usado para garantir a integridade referencial entre tabelas relacionadas.

Relacionamentos entre Tabelas:

FOREIGN KEY

Exemplo:

```
CREATE TABLE tabela_pai (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL  
);
```

Relacionamentos entre Tabelas:

FOREIGN KEY

Exemplo:

```
CREATE TABLE tabela_filha (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    pai_id INT,  
    FOREIGN KEY (pai_id) REFERENCES tabela_pai(id)  
);
```

Regras de Conversão

DER para o Modelo Relacional

DER para o Modelo Relacional

Entidades = Tabelas:

Cada entidade do Diagrama Entidade-Relacionamento (DER) é transformada em uma tabela no modelo relacional.

DER para o Modelo Relacional

Atributos = Colunas:

Os atributos de uma entidade no DER correspondem às colunas da tabela no modelo relacional. Por exemplo:

Se uma entidade "Cliente" possui atributos como Nome, CPF e Endereço, no modelo relacional esses atributos serão representados como colunas na tabela Cliente.

DER para o Modelo Relacional

Relacionamentos entre entidades:

Relacionamentos no DER podem ser representados de três maneiras no modelo relacional, dependendo da sua cardinalidade:

1:1 (um para um): Geralmente adicionando uma chave estrangeira em uma das tabelas.

1:N (um para muitos): Inserindo uma chave estrangeira na tabela que representa o lado da relação "muitos".

N:N (muitos para muitos): Criando uma tabela associativa (ou tabela de junção) que contém as chaves estrangeiras das tabelas envolvidas no relacionamento.

DER para o Modelo Relacional

Relacionamentos entre entidades:

Relacionamentos no DER podem ser representados de três maneiras no modelo relacional, dependendo da sua cardinalidade:

1:1 (um para um): Geralmente adicionando uma chave estrangeira em uma das tabelas.

1:N (um para muitos): Inserindo uma chave estrangeira na tabela que representa o lado da relação "muitos".

N:N (muitos para muitos): Criando uma tabela associativa (ou tabela de junção) que contém as chaves estrangeiras das tabelas envolvidas no relacionamento.

DER para o Modelo Relacional

1:1 (um para um):

Suponha que temos as entidades Pessoa e Passaporte. Cada pessoa possui apenas um passaporte, e cada passaporte pertence a apenas uma pessoa.

DER:

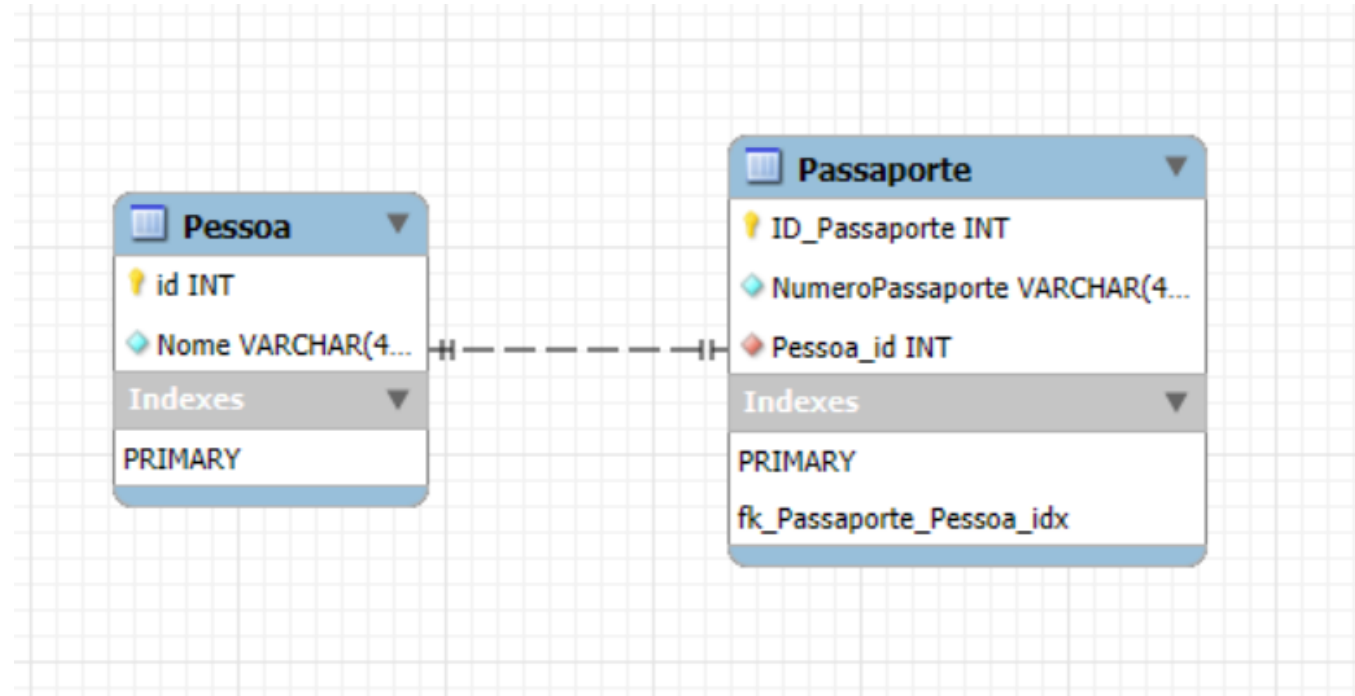
Pessoa (ID_Pessoa, Nome)

Passaporte (ID_Passaporte, NumeroPassaporte, ID_Pessoa)

Relacionamento 1:1 entre Pessoa e Passaporte.

DER para o Modelo Relacional

1:1 (um para um):



Aqui, ID_Pessoa é a chave estrangeira em Passaporte, indicando o relacionamento entre as tabelas.

DER para o Modelo Relacional

1:N (Um para Muitos):

Considere as entidades Autor e Livro. Um autor pode escrever vários livros, mas cada livro é associado a apenas um autor.

DER:

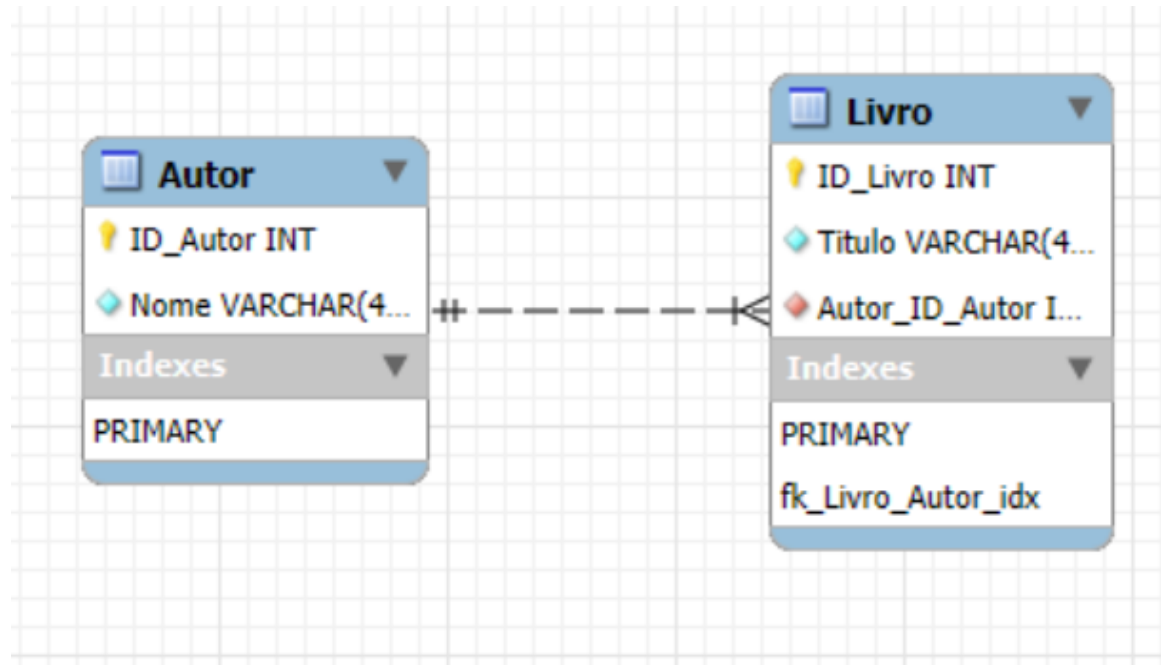
Autor (ID_Autor, Nome)

Livro (ID_Livro, Titulo, ID_Autor)

Relacionamento 1:N entre Autor e Livro.

DER para o Modelo Relacional

1:N (Um para Muitos):



Aqui, ID_Autor em Livro é a chave estrangeira, relacionando múltiplos livros a um único autor.

DER para o Modelo Relacional

N:N (Muitos para Muitos):

Suponha que temos as entidades Estudante e Curso. Um estudante pode estar matriculado em vários cursos, e um curso pode conter vários estudantes.

DER:

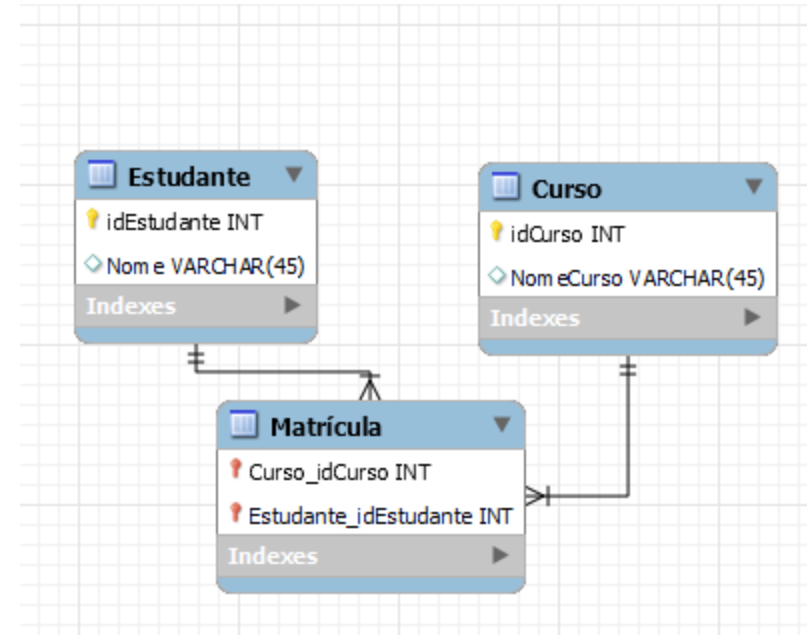
Estudante (ID_Estudante, Nome)

Curso (ID_Curso, NomeCurso)

Relacionamento N:N entre Estudante e Curso.

DER para o Modelo Relacional

N:N (Muitos para Muitos):



A tabela Matrícula é a tabela associativa que contém as chaves estrangeiras ID_Estudante e ID_Curso.

Isso permite mapear múltiplos estudantes a múltiplos cursos de maneira flexível.

Referências Bibliográficas

**Elmasri, R.; Navathe, S. B. Sistemas de Banco de Dados. 7. ed.
São Paulo: Pearson, 2019.**

**NIELD, Thomas. Introdução à Linguagem SQL: Abordagem Prática
Para Iniciantes. 1. ed. São Paulo: Novatec Editora, 26 abr. 2016.**

**KLINE, Kevin E.; KLINE, Daniel. SQL: O Guia Essencial - Manual
de Referência Profissional. 3. ed. Rio de Janeiro: Alta Books, 15
set. 2010.**

Referências Bibliográficas

MYSQL. MySQL Documentation: Data Types. Disponível em:
<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>