

Banco de Dados

Felipe Oliveira

Análise e Desenvolvimento de Sistemas
Faculdade SENAC

18 de fevereiro de 2025

Overview

1. Modelos de Dados

2. Modelo Relacional

3. Comando CREATE

4. Tipos de dados no MySQL

4.1 Tipos Numéricos

4.2 Tipos de Texto

4.3 Tipos de Data e Hora

4.4 Outros Tipos

5. Exercício

6. Comando INSERT

Modelos de Dados

Introdução aos Modelos de dados:

- **Modelo relacional:** Dados organizados em tabelas, base do MySQL.

Modelos de Dados

The diagram illustrates two database tables: **Pedidos** (Orders) and **Clientes** (Clients). The **Pedidos** table has a primary key on **id_pedido** and a foreign key on **id_cliente** that references the **id_cliente** primary key in the **Clientes** table.

Pedidos			
id_pedido	data_pedido	status	id_cliente
101	2024-02-10	Enviado	1
102	2024-02-12	Processando	2
103	2024-02-15	Cancelado	3
104	2024-02-18	Entregue	4

Clientes			
id_cliente	nome	email	telefone
1	João Silva	joao@email.com	(11) 99999-0001
2	Maria Souza	maria@email.com	(21) 98888-0002
3	Pedro Santos	pedro@email.com	(31) 97777-0003
4	Ana Oliveira	ana@email.com	(41) 96666-0004

Figura: As tabelas possuem chaves primárias (PK) e chaves estrangeiras (FK).

O que é o Modelo Relacional?

O modelo relacional é uma forma de organizar dados em tabelas.

- **Tabelas (Relacionamentos):** Estruturas que armazenam dados em linhas e colunas.
- **Chaves Primárias:** Identificadores únicos para cada registro em uma tabela.
- **Chaves Estrangeiras:** Referências que estabelecem relacionamentos entre tabelas.
- **Integridade Referencial:** Garante a consistência dos dados entre tabelas relacionadas.

Componentes do Modelo Relacional

1. Tabelas (Entidades):

- Representam entidades do mundo real (ex: clientes, produtos).
- Compostas por linhas (registros) e colunas (atributos).

2. Chave Primária:

- Identificador único de cada registro em uma tabela.
- Exemplo: `id` na tabela `clientes`.

3. Chave Estrangeira:

- Referência a uma chave primária em outra tabela.
- Exemplo: `cliente_id` na tabela `pedidos`.

Exemplo de Modelo Relacional

Tabela clientes:

- id (Chave Primária)
- nome
- email

Tabela pedidos:

- id (Chave Primária)
- cliente_id (Chave Estrangeira para clientes)
- data_pedido
- valor

Relacionamento:

- Um cliente pode ter vários pedidos.
- O campo cliente_id em pedidos referencia id em clientes.

Exemplo de Criação no MySQL

```
CREATE TABLE clientes (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL UNIQUE  
);  
  
CREATE TABLE pedidos (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  cliente_id INT,  
  data_pedido DATE NOT NULL,  
  valor DECIMAL(10, 2) NOT NULL,  
  FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```


Vantagens do Modelo Relacional

- **Simplicidade:** Estrutura de dados fácil de entender e usar.
- **Flexibilidade:** Permite consultas complexas com SQL.
- **Integridade:** Garante a consistência dos dados através de restrições.
- **Escalabilidade:** Adequado para aplicações de pequeno a grande porte.

O que é o comando CREATE?

Definição

O comando `CREATE` no MySQL é usado para criar novos objetos no banco de dados, como bancos de dados e tabelas. Ele é essencial para definir a estrutura do banco de dados.

Principais usos:

- Criar um banco de dados: `CREATE DATABASE`.
- Criar uma tabela: `CREATE TABLE`.

CREATE DATABASE

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal is the SQL command to create a database.

```
CREATE DATABASE nome_do_banco;
```

Figura: Sintaxe de criação de uma estrutura de banco de dados

CREATE TABLE

```
CREATE TABLE nome_da_tabela (  
    coluna1 tipo_de_dado restricoes,  
    coluna2 tipo_de_dado restricoes,  
    ...  
);
```

Figura: Sintaxe de criação de uma tabela

Exemplo

```

-- Cria o bd empresa
CREATE DATABASE sistema_empresa;

-- Comando para indicar ao MySQL que iremos utilizar o bd sistema_empresa
USE sistema_empresa;

-- Cria a tabela clientes com seus campos e restrições
CREATE TABLE departamentos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL
);
```

Explicação do Exemplo de Tabela

- `id INT AUTO_INCREMENT PRIMARY KEY`: Cria uma coluna de números inteiros que é incrementada automaticamente e serve como chave primária.
- `nome VARCHAR(100) NOT NULL`: Cria uma coluna de texto com até 100 caracteres que não pode ser nula.

Tipos de dados

Os tipos de dados no MySQL definem o tipo de valor que pode ser armazenado em uma coluna de uma tabela. Escolher o tipo de dados correto é essencial para:

- Garantir a integridade dos dados.
- Otimizar o armazenamento.
- Melhorar o desempenho das consultas.

Tipos Numéricos

- INT: Números inteiros (ex: 1, 2, 3).
- FLOAT: Números de ponto flutuante (ex: 3.14).
- DOUBLE: Números de ponto flutuante com maior precisão.
- DECIMAL: Números decimais de precisão fixa (ex: 10.99).

Exemplo



```
CREATE TABLE produtos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    preco DECIMAL(10, 2) NOT NULL  
);
```

Tipos de Texto

- `CHAR(n)`: Texto de tamanho fixo, onde `n` é o número exato de caracteres.
- `VARCHAR(n)`: Texto de tamanho variável, onde `n` é o número máximo de caracteres (ex: `VARCHAR(100)`).
- `TEXT`: Texto de tamanho variável com capacidade para armazenar grandes quantidades de texto.

Exemplo

```
CREATE TABLE usuarios (  
    nome VARCHAR(100) NOT NULL,  
    descricao TEXT  
);
```

Tipos de Data e Hora

- DATE: Armazena uma data no formato YYYY-MM-DD.
- DATETIME: Armazena data e hora no formato YYYY-MM-DD HH:MM:SS.
- TIMESTAMP: Armazena data e hora, mas com fuso horário automático.
- TIME: Armazena apenas a hora no formato HH:MM:SS.

Exemplo



```
CREATE TABLE eventos (  
    data_evento DATE NOT NULL,  
    hora_inicio TIME NOT NULL  
);
```

Outros Tipos

- **BOOLEAN**: Armazena valores verdadeiros ou falsos).
- **ENUM**: Permite definir uma lista de valores possíveis para uma coluna (ex: `ENUM('ativo', 'inativo')`).

Exemplo

```
CREATE TABLE pedidos (  
    status ENUM('pendente', 'processando', 'concluido') DEFAULT 'pendente',  
    foi_entregue BOOLEAN  
);
```

Descrição do Problema

Você foi contratado para desenvolver um sistema de gerenciamento de uma biblioteca. Sua primeira tarefa é criar uma tabela no MySQL para armazenar informações sobre os livros.

Requisitos:

- A tabela deve se chamar `livros`.
- Cada livro deve ter um ID único, gerado automaticamente.
- Cada livro deve ter um título, um autor, um ano de publicação e um gênero.
- O título e o autor são obrigatórios.
- O gênero deve ser um dos seguintes: Aventura, Ficção, Romance, Técnico.
- O ano de publicação deve ser um número inteiro.

Tarefa

Com base nos requisitos, crie a tabela `livros` no MySQL. Use o comando `CREATE TABLE` e defina as colunas com os tipos de dados apropriados.

Dicas:

- Use `INT` para o ID e o ano de publicação.
- Use `VARCHAR` para o título e o autor.
- Use `ENUM` para o gênero.
- Defina o ID como chave primária e auto-incremento.

Ao fim do exercício...

Com este exercício, você praticou:

- A criação de tabelas no MySQL.
- A escolha de tipos de dados apropriados.
- A definição de restrições e valores padrão.

O que é o comando INSERT?

Definição

O comando `INSERT` é usado para adicionar novos registros (linhas) em uma tabela do banco de dados MySQL.

Sintaxe básica



```
INSERT INTO nome_da_tabela (coluna1, coluna2, coluna3, ...)
VALUES (valor1, valor2, valor3, ...);
```

Exemplo



```
INSERT INTO clientes (nome, email, telefone)
VALUES ('João Silva', 'joao.silva@email.com', '11987654321');
```

Casos de uso

- Inserir um único registro.
- Inserir múltiplos registros de uma vez:

```
INSERT INTO clientes (nome, email, telefone)
VALUES
  ('Maria Oliveira', 'maria.oliveira@email.com', '11912345678'),
  ('Carlos Souza', 'carlos.souza@email.com', '11987651234');
```

Dicas importantes

- Certifique-se de que os valores correspondam ao tipo de dados das colunas.
- Colunas com restrições como NOT NULL ou UNIQUE devem ser preenchidas corretamente.

Exercício

1. Insira 10 registros na tabela do seu sistema de biblioteca.