FAP UFRN Softex 2024 – Módulo 3 – Introdução a Bancos de Dados

Prof. José Alfredo F. Costa

**MySQL - exemplos:**

**1: Introdução ao MySQL**

* Explicação: O MySQL é um sistema de gerenciamento de banco de dados relacional amplamente utilizado para armazenar e gerenciar dados.

**2: Por que Usar o MySQL?**

* Explicação: Destaque as vantagens do MySQL, como escalabilidade, confiabilidade e suporte a SQL.

**3: Instalação do MySQL**

* Passos:
  1. Visite o site oficial do MySQL.
  2. Faça o download da versão apropriada para o seu sistema operacional.
  3. Execute o instalador e siga as instruções na tela para configurar o MySQL.

**4: Conectar-se ao MySQL**

* Explicação: Mostre como usar o comando **mysql** ou uma biblioteca de conexão para se conectar a um servidor MySQL.

**5: Criar um Banco de Dados**

* Exemplo de Código:

CREATE DATABASE meu\_banco;

**6: Selecionar um Banco de Dados**

* Exemplo de Código:

USE meu\_banco;

**7: Criar uma Tabela**

* Exemplo de Código:

CREATE TABLE usuarios ( id INT AUTO\_INCREMENT PRIMARY KEY, nome VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL );

**8: Tipos de Dados no MySQL**

* Tipos de dados comuns, como INT, VARCHAR, DATE, e outros.

**9: Inserir Registros**

* Exemplo de Código:

INSERT INTO usuarios (nome, email) VALUES ('João', 'joao@email.com');

**10: Consultar Dados**

* Exemplo de Código:

SELECT \* FROM usuarios;

**11: Atualizar Registros**

* Exemplo de Código:

UPDATE usuarios SET nome = 'Maria' WHERE id = 1;

**12: Excluir Registros**

* Exemplo de Código:

DELETE FROM usuarios WHERE id = 2;

**13: Chaves Primárias e Índices**

* Uso de chaves primárias para identificar registros exclusivos e índices para otimizar consultas.

**14: Consultas com Cláusulas WHERE**

* Exemplo de Código:

SELECT \* FROM usuarios WHERE nome = 'João';

**15: Ordenação e Limitação de Resultados**

* Exemplo de Código:

SELECT \* FROM usuarios ORDER BY nome ASC LIMIT 10;

**16: Junção de Tabelas (JOINs)**

* Explicação: Demonstre como usar INNER JOIN e LEFT JOIN para combinar dados de várias tabelas.

**17: Funções Agregadas**

* Exemplo de Código:

SELECT COUNT(\*) AS total\_usuarios FROM usuarios;

**18: Transações no MySQL**

* Importância das transações para garantir a consistência dos dados.

**19: Backup e Restauração de Banco de Dados**

* Como fazer backup e restaurar bancos de dados MySQL?

**20: Segurança e Práticas Recomendadas**

* Dicas sobre como manter seu banco de dados seguro, como definir permissões e evitar injeção SQL.

**1: Introdução ao MySQL**

* Explicação: MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) amplamente utilizado.

**2: Por que Usar o MySQL?**

* Explicação: O MySQL é popular devido à sua escalabilidade, confiabilidade, suporte à linguagem SQL e comunidade ativa.

**3: Instalação do MySQL**

* Passos:
  1. Visite o site oficial do MySQL.
  2. Baixe a versão apropriada para seu sistema operacional.
  3. Execute o instalador e siga as instruções na tela.

**4: Conectar-se ao MySQL**

* Explicação: Use o comando **mysql** ou uma biblioteca de conexão (por exemplo, MySQL Node.js) para estabelecer conexão com o servidor MySQL.
* Exemplo de Código (via linha de comando):

mysql -u seu\_usuario -p

Exemplo: mysql -u jafcosta -p

**5: Criar um Banco de Dados**

* Exemplo de Código:

CREATE DATABASE meu\_banco;

* Uso: Cria um novo banco de dados chamado **meu\_banco**.

**6: Selecionar um Banco de Dados**

* Exemplo de Código:

USE meu\_banco;

* Uso: Seleciona o banco de dados **meu\_banco** para realizar operações subsequentes.

**7: Criar uma Tabela**

* Exemplo de Código:

CREATE TABLE usuarios ( id INT AUTO\_INCREMENT PRIMARY KEY, nome VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL );

* Uso: Cria uma tabela chamada **usuarios** com três colunas (id, nome e email).

**8: Tipos de Dados no MySQL**

* Explicação: O MySQL suporta vários tipos de dados, como INT, VARCHAR, DATE, TIMESTAMP, entre outros.

**9: Inserir Registros**

* Exemplo de Código:

INSERT INTO usuarios (nome, email) VALUES ('João', 'joao@email.com');

* Uso: Insere um novo registro na tabela **usuarios**.

**10: Consultar Dados**

* Exemplo de Código (selecionando todos os registros):

SELECT \* FROM usuarios;

* Uso: Recupera todos os registros da tabela **usuarios**.

**11: Atualizar Registros**

* Exemplo de Código:

UPDATE usuarios SET nome = 'Maria' WHERE id = 1;

* Uso: Atualiza o nome de um usuário específico na tabela **usuarios**.

**12: Excluir Registros**

* Exemplo de Código:

DELETE FROM usuarios WHERE id = 2;

* Uso: Exclui um registro da tabela **usuarios** com base no ID.

**13: Chaves Primárias e Índices**

* Explicação: As chaves primárias garantem a unicidade dos registros, enquanto os índices melhoram o desempenho de consultas.
* Exemplo de Código (adicionando uma chave primária):

ALTER TABLE usuarios ADD PRIMARY KEY (id);

**14: Consultas com Cláusulas WHERE**

* Exemplo de Código:

SELECT \* FROM usuarios WHERE nome = 'João';

* Uso: Filtra registros com base em uma condição específica (neste caso, nome igual a 'João').

**15: Ordenação e Limitação de Resultados**

* Exemplo de Código (ordenando e limitando resultados):

SELECT \* FROM usuarios ORDER BY nome ASC LIMIT 10;

* Uso: Ordena os registros por nome em ordem crescente e limita o resultado a 10 registros.

**16: Junção de Tabelas (JOINs)**

* Explicação: JOINs são usados para combinar dados de várias tabelas.
* Exemplo de Código (INNER JOIN entre duas tabelas):

SELECT pedidos.id, usuarios.nome FROM pedidos INNER JOIN usuarios ON pedidos.usuario

criação de uma base de dados em MySQL de uma coleção de livros com tabela livros, com atributos id, título, autores, estado do livro, ano e isbn

Criar uma base de dados MySQL para uma coleção de livros com uma tabela chamada "livros" e os atributos id, título, autores, estado do livro, ano e ISBN.

Aqui estão os passos para criar essa base de dados e tabela:

**Passo 1: Conectar-se ao MySQL** Primeiro, conecte-se ao seu servidor MySQL usando um cliente SQL, como o MySQL Workbench ou o terminal, usando o seguinte comando:

mysql -u seu\_usuario -p

Você será solicitado a inserir sua senha.

**Passo 2: Criar uma nova base de dados** Agora, você pode criar uma nova base de dados para sua coleção de livros. Vamos chamá-la de "biblioteca" neste exemplo:

CREATE DATABASE biblioteca;

Certifique-se de que a base de dados foi criada com sucesso.

**Passo 3: Selecionar a Base de Dados** Para começar a trabalhar com a nova base de dados, selecione-a usando o comando "USE":

USE biblioteca;

**Passo 4: Criar a Tabela "livros"** Agora, você pode criar a tabela "livros" com os atributos mencionados (id, título, autores, estado do livro, ano e ISBN):

CREATE TABLE livros ( id INT AUTO\_INCREMENT PRIMARY KEY, titulo VARCHAR(255) NOT NULL, autores VARCHAR(255), estado\_do\_livro ENUM('Disponível', 'Emprestado', 'Em Reparo') NOT NULL, ano INT, isbn VARCHAR(13) );

Neste comando SQL:

* **id** é uma coluna de identificação única para cada livro, com incremento automático.
* **titulo** armazena o título do livro, que não pode ser nulo.
* **autores** é uma coluna para armazenar os nomes dos autores.
* **estado\_do\_livro** é uma coluna ENUM que pode ter três valores: Disponível, Emprestado ou Em Reparo.
* **ano** é um campo opcional para o ano de publicação.
* **isbn** é uma coluna para o código ISBN do livro, com um máximo de 13 caracteres.

A tabela "livros" agora está criada com os atributos desejados.

**Passo 5: Verificar a Tabela** Você pode verificar se a tabela foi criada corretamente usando o seguinte comando:

DESCRIBE livros;

Isso exibirá uma descrição da estrutura da tabela "livros", confirmando que os atributos foram definidos corretamente.

Agora você tem uma base de dados MySQL chamada "biblioteca" com uma tabela "livros" que pode ser usada para armazenar informações sobre livros, incluindo título, autores, estado do livro, ano e ISBN. Você pode começar a inserir registros de livros nesta tabela conforme necessário.

Dicionário de dados para a tabela simples de livros e autores, incluindo informações, exemplos e uma representação de tabela:

**Tabela: livros**

| **Campo** | **Tipo de Dados** | **Restrições** | **Descrição** | **Exemplo** |
| --- | --- | --- | --- | --- |
| id | INT | Chave Primária, Auto-Incr. | Identificador único para cada livro | 1 |
| titulo | VARCHAR(255) | NOT NULL | Título do livro | "Aventuras de Aprendizado" |
| autores | VARCHAR(255) |  | Nomes dos autores do livro | "Alice Johnson, Bob Smith" |
| estado\_do\_livro | ENUM ('Disponível', 'Emprestado', 'Em Reparo') |  | Estado atual do livro | "Disponível" |
| ano | INT |  | Ano de publicação do livro (opcional) | 2023 |
| isbn | VARCHAR(13) |  | Código ISBN do livro (até 13 caracteres) | "978-9876543210" |

**Exemplo de Registro:**

| **id** | **titulo** | **autores** | **estado\_do\_livro** | **ano** | **isbn** |
| --- | --- | --- | --- | --- | --- |
| 1 | Aventuras de Aprendizado | Alice Johnson, Bob Smith | Disponível | 2023 | 978-9876543210 |

**Descrição:**

* **id**: É uma coluna de identificação única para cada livro. Ela é incrementada automaticamente a cada novo registro, garantindo que cada livro tenha um identificador único.
* **titulo**: Armazena o título do livro. É uma coluna obrigatória (NOT NULL) para garantir que todos os livros tenham um título.
* **autores**: Esta coluna armazena os nomes dos autores do livro. Pode conter vários nomes separados por vírgulas para livros com vários autores.
* **estado\_do\_livro**: É uma coluna ENUM que pode ter três valores: 'Disponível', 'Emprestado' ou 'Em Reparo'. Essa coluna registra o estado atual do livro.
* **ano**: É um campo opcional que armazena o ano de publicação do livro.
* **isbn**: Esta coluna armazena o código ISBN do livro, que é uma sequência de até 13 caracteres.

O exemplo de registro mostra um livro com todos os campos preenchidos, incluindo título, autores, estado do livro, ano de publicação e ISBN. Esses campos são representativos de como os dados seriam armazenados na tabela "livros".

Entrada de dados para um novo livro com dois autores e popular as bases de dados "livros" e "autores" usando as tabelas "livros\_autores" para estabelecer as relações.

Suponhamos que o novo livro seja "Aventuras de Aprendizado" com os autores "Alice Johnson" e "Bob Smith".

**Passo 1: Inserir os Autores na Tabela "autores"**

Primeiro, insira os dois autores na tabela "autores":

-- Inserir o autor Alice Johnson

INSERT INTO autores (nome) VALUES ('Alice Johnson');

-- Inserir o autor Bob Smith

INSERT INTO autores (nome) VALUES ('Bob Smith');

Isso criará dois registros de autores com IDs únicos gerados automaticamente.

**Passo 2: Inserir o Livro na Tabela "livros"**

Agora, insira o novo livro na tabela "livros". Suponhamos que o novo livro tenha o título "Aventuras de Aprendizado", seja do ano de 2023, tenha ISBN "978-9876543210" e esteja disponível:

-- Inserir o novo livro

INSERT INTO livros (titulo, autores, estado\_do\_livro, ano, isbn)

VALUES ('Aventuras de Aprendizado', 'Alice Johnson e Bob Smith', 'Disponível', 2023, '978-9876543210');

**Passo 3: Registrar as Associações na Tabela "livros\_autores"**

Finalmente, registre as associações entre o novo livro e seus autores na tabela "livros\_autores". Para fazer isso, você precisará dos IDs dos autores e do ID do livro recém-inserido. Suponha que o ID do livro seja 2 (você pode obter esse valor usando uma consulta SQL para recuperar o último ID inserido):

-- Obter o ID do livro recém-inserido

SELECT LAST\_INSERT\_ID() INTO @livro\_id;

-- Obter os IDs dos autores

SELECT id INTO @autor1\_id FROM autores WHERE nome = 'Alice Johnson';

SELECT id INTO @autor2\_id FROM autores WHERE nome = 'Bob Smith';

-- Registrar as associações

INSERT INTO livros\_autores (livro\_id, autor\_id) VALUES (@livro\_id, @autor1\_id);

INSERT INTO livros\_autores (livro\_id, autor\_id) VALUES (@livro\_id, @autor2\_id);

Isso irá estabelecer as relações entre o livro "Aventuras de Aprendizado" e seus autores, "Alice Johnson" e "Bob Smith", na tabela "livros\_autores".

Agora, a base de dados foi populada com o novo livro e seus autores, mantendo a integridade das relações por meio da tabela "livros\_autores". Você pode continuar adicionando mais livros e autores da mesma maneira conforme necessário.

**Dicionários de dados após a normalização dos dados, incluindo informações, exemplos e representações de tabelas:**

**Tabela: livros**

| **Campo** | **Tipo de Dados** | **Restrições** | **Descrição** | **Exemplo** |
| --- | --- | --- | --- | --- |
| id | INT | Chave Primária, Auto-Incr. | Identificador único para cada livro | 1 |
| titulo | VARCHAR(255) | NOT NULL | Título do livro | "Aventuras de Aprendizado" |
| estado\_do\_livro | ENUM ('Disponível', 'Emprestado', 'Em Reparo') |  | Estado atual do livro | "Disponível" |
| ano | INT |  | Ano de publicação do livro (opcional) | 2023 |

**Exemplo de Registro (livros):**

| **id** | **titulo** | **estado\_do\_livro** | **ano** |
| --- | --- | --- | --- |
| 1 | Aventuras de Aprendizado | Disponível | 2023 |

**Tabela: autores**

| **Campo** | **Tipo de Dados** | **Restrições** | **Descrição** | **Exemplo** |
| --- | --- | --- | --- | --- |
| id | INT | Chave Primária, Auto-Incr. | Identificador único para cada autor | 1 |
| nome | VARCHAR(255) | NOT NULL | Nome do autor | "Alice Johnson" |
|  |  |  |  | "Bob Smith" |

**Exemplo de Registro (autores):**

| **id** | **nome** |
| --- | --- |
| 1 | Alice Johnson |
| 2 | Bob Smith |

**Tabela: livros\_autores (Tabela de Junção)**

| **Campo** | **Tipo de Dados** | **Restrições** | **Descrição** | **Exemplo** |
| --- | --- | --- | --- | --- |
| livro\_id | INT | Chave Estrangeira para livros ([livros.id](http://livros.id/)) | ID do livro associado ao autor | 1 |
| autor\_id | INT | Chave Estrangeira para autores ([autores.id](http://autores.id/)) | ID do autor associado ao livro | 1 |

**Exemplo de Registro (livros\_autores):**

| **livro\_id** | **autor\_id** |
| --- | --- |
| 1 | 1 |
| 1 | 2 |

**Descrição:**

* **Tabela "livros"**: Esta tabela armazena informações sobre os livros. Ela contém um identificador único (id), título do livro, estado atual do livro e o ano de publicação (opcional). A coluna "estado\_do\_livro" ainda é uma coluna ENUM com os valores 'Disponível', 'Emprestado' ou 'Em Reparo'.
* **Tabela "autores"**: Esta tabela armazena informações sobre os autores. Ela contém um identificador único (id) e o nome do autor. Cada autor é identificado por seu nome.
* **Tabela "livros\_autores"**: Esta é uma tabela de junção que associa livros a autores. Ela contém duas colunas: "livro\_id" e "autor\_id", que são chaves estrangeiras referenciando as tabelas "livros" e "autores". Isso permite a associação de vários autores a um livro e vice-versa.

Os exemplos de registro mostram como os dados seriam armazenados nas tabelas normalizadas. Por exemplo, na tabela "livros\_autores", o registro indica que o livro com ID 1 está associado aos autores com IDs 1 e 2, o que significa que o livro tem dois autores: Alice Johnson e Bob Smith.

Esse design de banco de dados facilita a manutenção e evita a duplicação de dados.