

Criação do Banco de Dados

Passo 1: Instalar o MySQL Workbench

Faça o download do MySQL Workbench a partir do site oficial do MySQL e instale-o em seu computador. Após a instalação, abra o MySQL Workbench

Passo 2: Configurar a Conexão com o MySQL

No MySQL Workbench, clique no sinal de mais (+) ao lado de "MySQL Connections" para configurar uma nova conexão. Na janela que aparece, insira as informações necessárias, como o nome da conexão, o hostname (geralmente "localhost" para uma instalação local do MySQL), a porta (geralmente 3306 para MySQL), o usuário e a senha. Clique em "Test Connection" para verificar se a conexão foi bem-sucedida. Se foi, clique em "OK" para salvar a conexão

Passo 3: Criar um Novo Usuário

Para criar um novo usuário, primeiro conecte-se ao MySQL como o usuário root. No shell do MySQL, use o comando `CREATE USER` para criar um novo usuário. Por exemplo:

```
---  
  
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';  
  
---
```

Neste comando, 'newuser' é o nome do novo usuário e 'password' é a senha do novo usuário. 'localhost' indica que o novo usuário pode se conectar ao MySQL apenas a partir do mesmo computador onde o MySQL está instalado [Fonte 3](#).

Passo 4: Conceder Permissões ao Novo Usuário

Depois de criar o novo usuário, você precisa conceder permissões a ele. Use o comando `GRANT` para fazer isso. Por exemplo:

```
GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Este comando concede ao novo usuário todos os privilégios em todos os bancos de dados e tabelas

Passo 5: Criar um Novo Banco de Dados

Para criar um novo banco de dados, use o comando `CREATE DATABASE`. Por exemplo:

```
CREATE DATABASE new_database;
```

Este comando cria um novo banco de dados chamado 'new_database'

Passo 6: Criar Tabelas no Novo Banco de Dados

Para criar tabelas no novo banco de dados, primeiro selecione o banco de dados com o comando `USE`. Em seguida, use o comando `CREATE TABLE` para criar uma nova tabela. Por exemplo:

```
USE new_database; CREATE TABLE new_table ( column1 datatype, column2 datatype, ... );
```

Este comando cria uma nova tabela chamada 'new_table' no banco de dados 'new_database'. 'column1', 'column2', etc. são os nomes das colunas na nova tabela, e 'datatype' é o tipo de dados de cada coluna

Por fim, lembre-se de que é importante planejar a estrutura do seu banco de dados antes de começar a criar bancos de dados e tabelas. Isso inclui decidir quais informações precisam ser armazenadas, quais tabelas precisam ser criadas e quais informações vão em cada tabela

Criação das Tabelas

Passo 1: Crie a tabela Chave

A classe Chave tem três atributos: `chave_id`, `descricao` e `status`. Assim, a tabela Chave no banco de dados MySQL deve ter colunas correspondentes a esses atributos.

Aqui está um exemplo de como criar a tabela Chave:

```
CREATE TABLE Chave ( chave_id INT PRIMARY KEY, descricao VARCHAR(255), status VARCHAR(255) );
```

Passo 2: Crie a tabela Servidor

Da mesma forma, a classe Servidor tem três atributos: `servidor_id`, `nome` e `departamento`. Portanto, a tabela Servidor deve ter colunas correspondentes a esses atributos.

Aqui está um exemplo de como criar a tabela Servidor:

```
CREATE TABLE Servidor ( servidor_id INT PRIMARY KEY, nome VARCHAR(255), departamento VARCHAR(255) );
```

Passo 3: Crie a tabela Empréstimo

A classe Empréstimo tem quatro

atributos: `chave`, `servidor`, `data_emprestimo` e `data_devolucao`. A tabela Empréstimo deve ter colunas correspondentes a esses atributos.

Aqui está um exemplo de como criar a tabela Empréstimo:

```
CREATE TABLE Emprestimo ( emprestimo_id INT PRIMARY KEY, chave_id INT, servidor_id INT, data_emprestimo DATE, data_devolucao DATE, FOREIGN KEY (chave_id) REFERENCES Chave(chave_id), FOREIGN KEY (servidor_id) REFERENCES Servidor(servidor_id) );
```

Observe que `chave_id` e `servidor_id` na tabela Empréstimo são chaves estrangeiras que referenciam as chaves primárias nas tabelas Chave e Servidor, respectivamente.

Passo 4: Conectando ao Banco de Dados

Para conectar-se ao banco de dados MySQL a partir de um programa Python, você pode usar a biblioteca `mysql.connector`, conforme mostrado abaixo :

```
import mysql.connector db = mysql.connector.connect( host="localhost", user="your_username", password="your_password", database="your_database" )
```

Passo 5: Executando as Consultas SQL

Depois de estabelecer a conexão com o banco de dados, você pode executar as consultas SQL para criar as tabelas. Aqui está um exemplo de como fazer isso [Fonte 1](#), [Fonte 6](#):

```
cursor = db.cursor() # Crie a tabela Chave cursor.execute(""" CREATE TABLE Chave ( chave_id INT PRIMARY KEY, descricao VARCHAR(255), status VARCHAR(255) ) """) # Crie a tabela Servidor cursor.execute(""" CREATE TABLE Servidor ( servidor_id INT PRIMARY KEY, nome VARCHAR(255), departamento VARCHAR(255) ) """) # Crie a tabela Empréstimo cursor.execute(""" CREATE TABLE Emprestimo ( emprestimo_id INT PRIMARY KEY, chave_id INT, servidor_id INT, data_emprestimo DATE, data_devolucao DATE, FOREIGN KEY (chave_id) REFERENCES Chave(chave_id), FOREIGN KEY (servidor_id) REFERENCES Servidor(servidor_id) ) """)
```