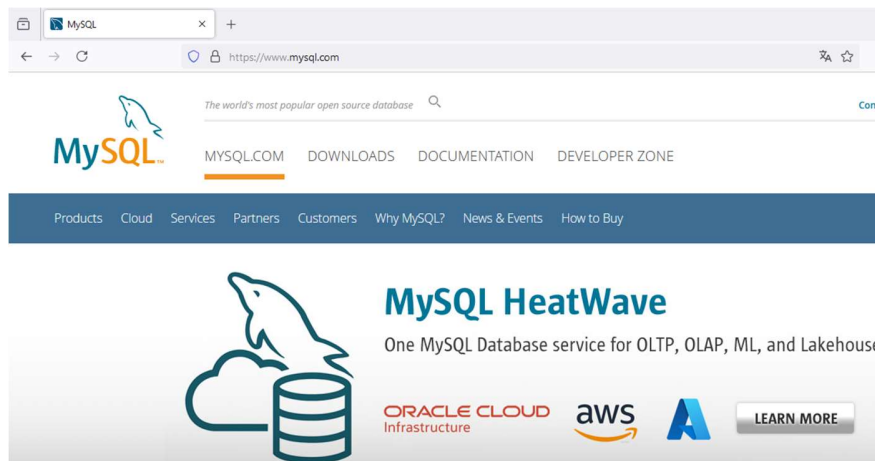


# MySQL CRUD

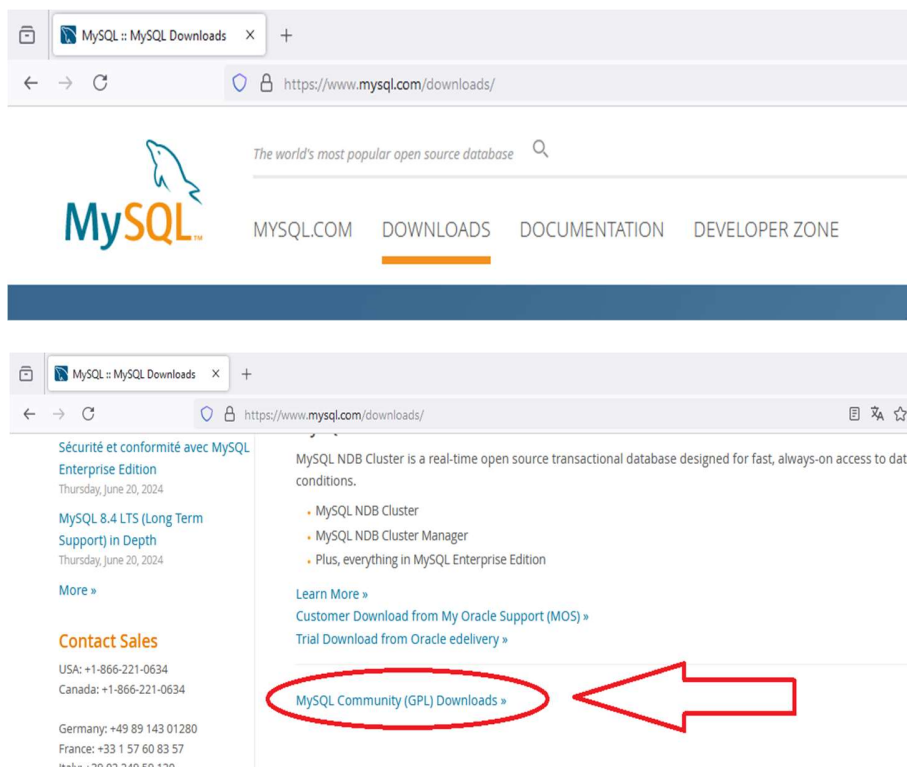
## CONFIGURAÇÃO INICIAL

### Baixar o MySQL:

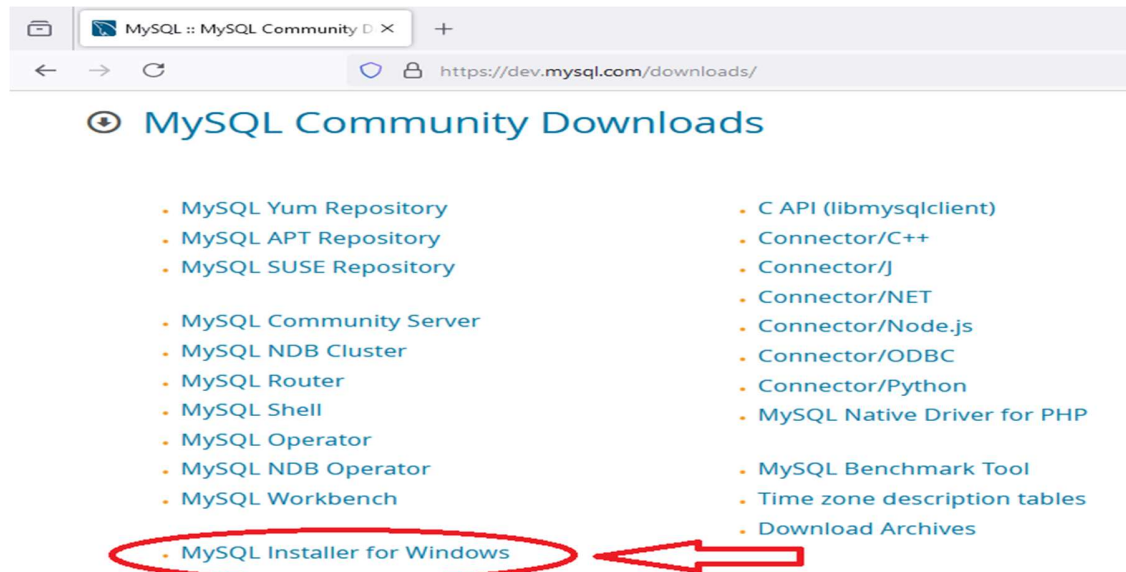
1. Baixe e instale o MySQL do site oficial (<https://www.mysql.com/>).



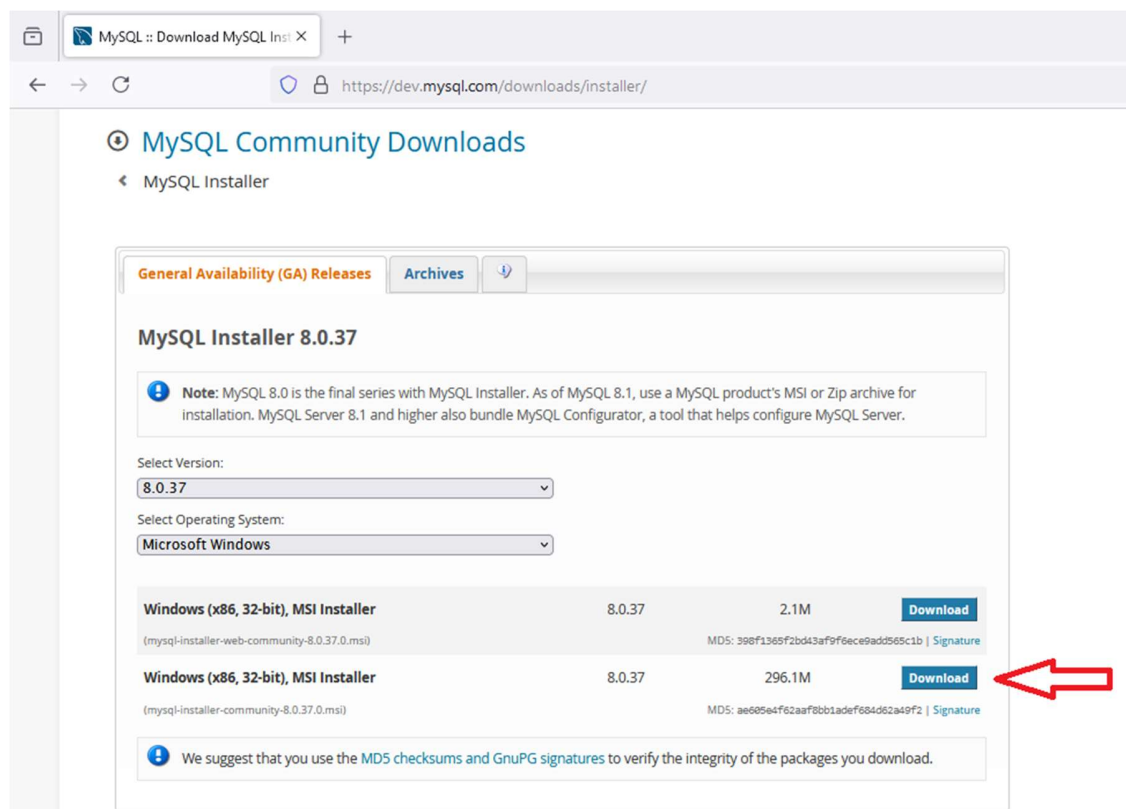
2. Click em Downloads e procure => **MySQL Community (GPL) Downloads**.



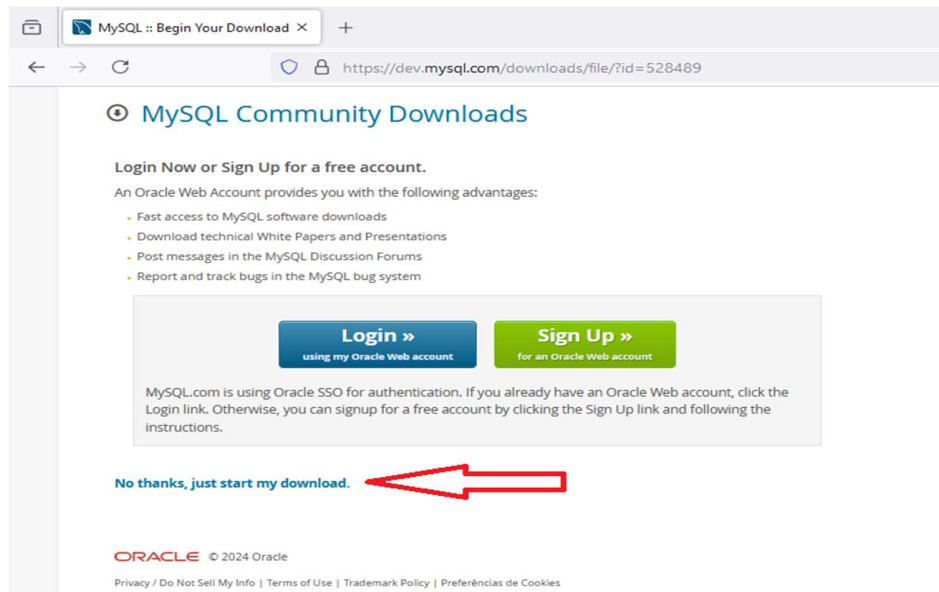
3. Click em => **MySQL Installer for Windows.**



4. Click em => **Windows (x86, 32-bit), MSI Installer 8.0.37 296.1M Download.**

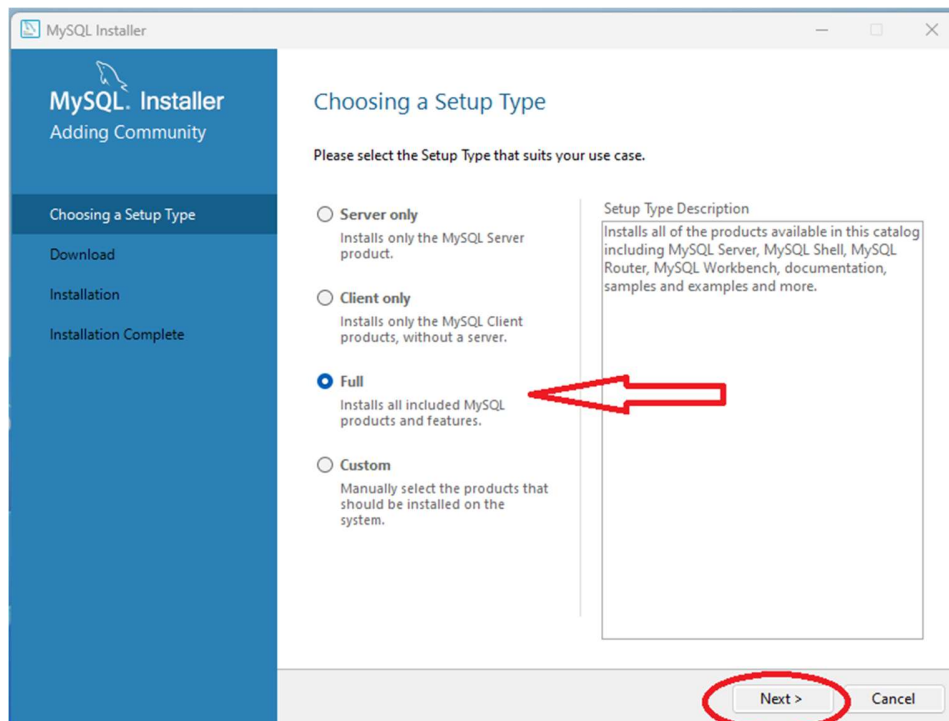


5. Click em => **No thanks, just start my download.**

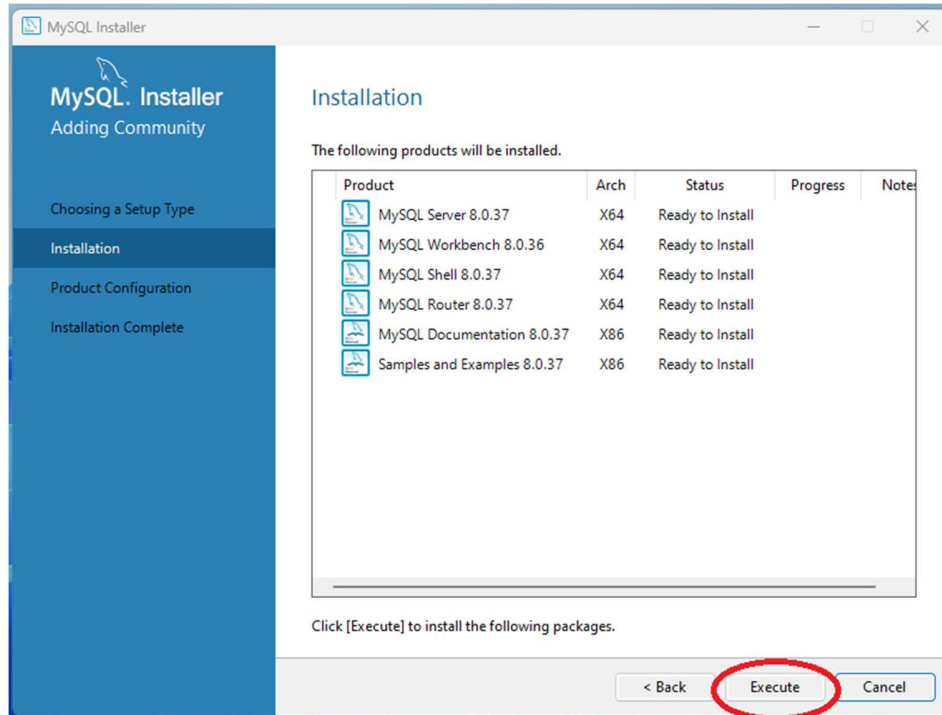


## Instalação do MySQL:

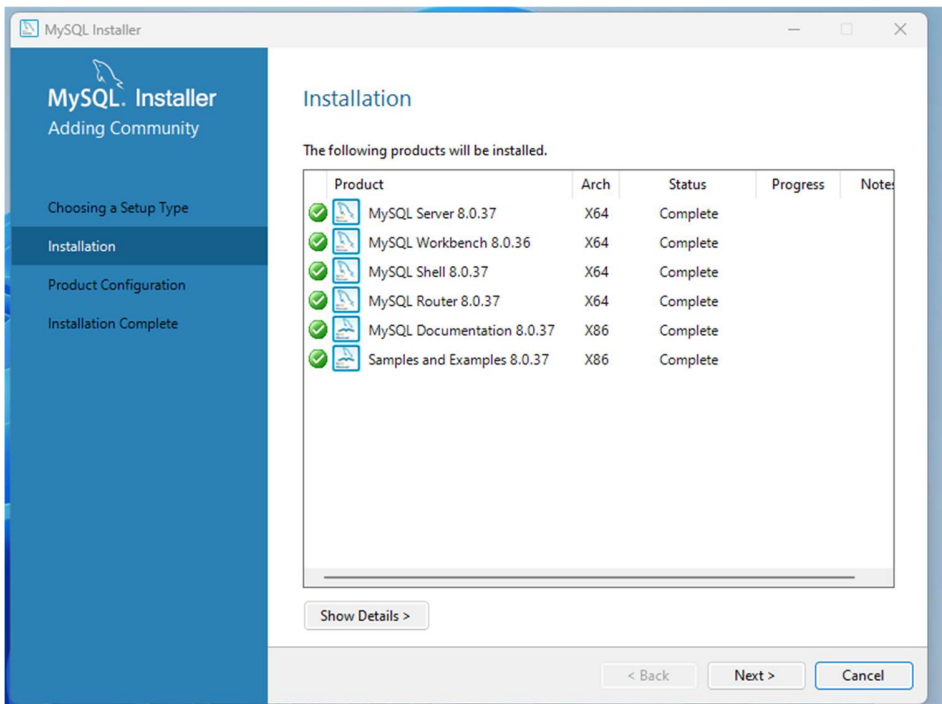
1. Selecione o **Full** e clique em **Next**.



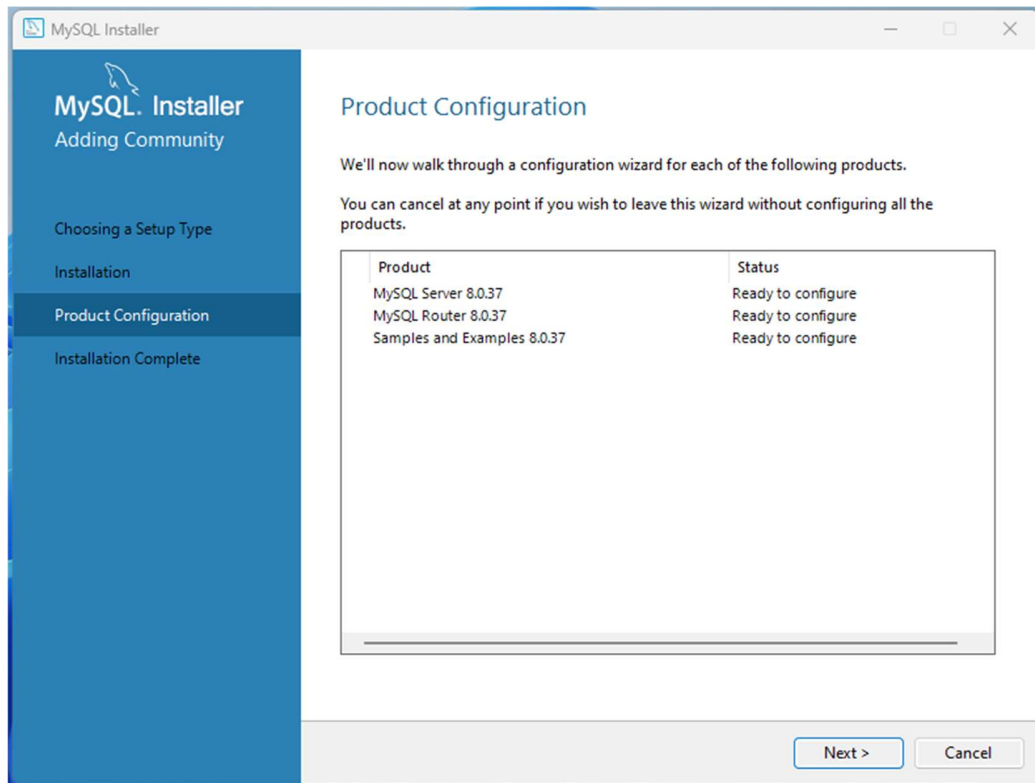
## 2. Selezione o **Execute**.



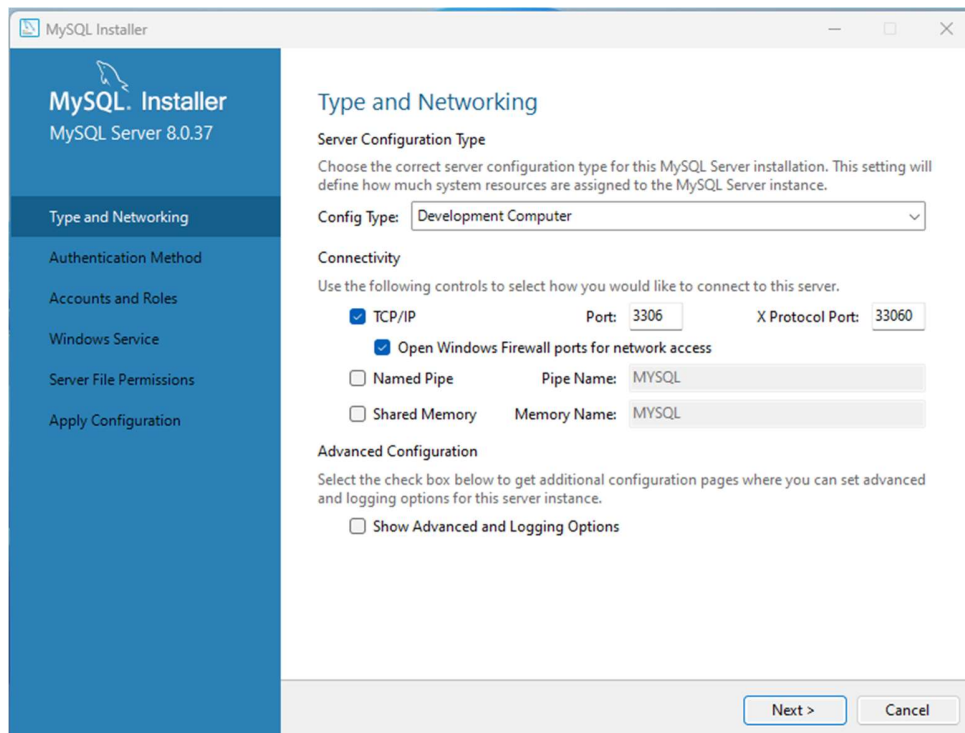
## 3. Clique em **Next**.



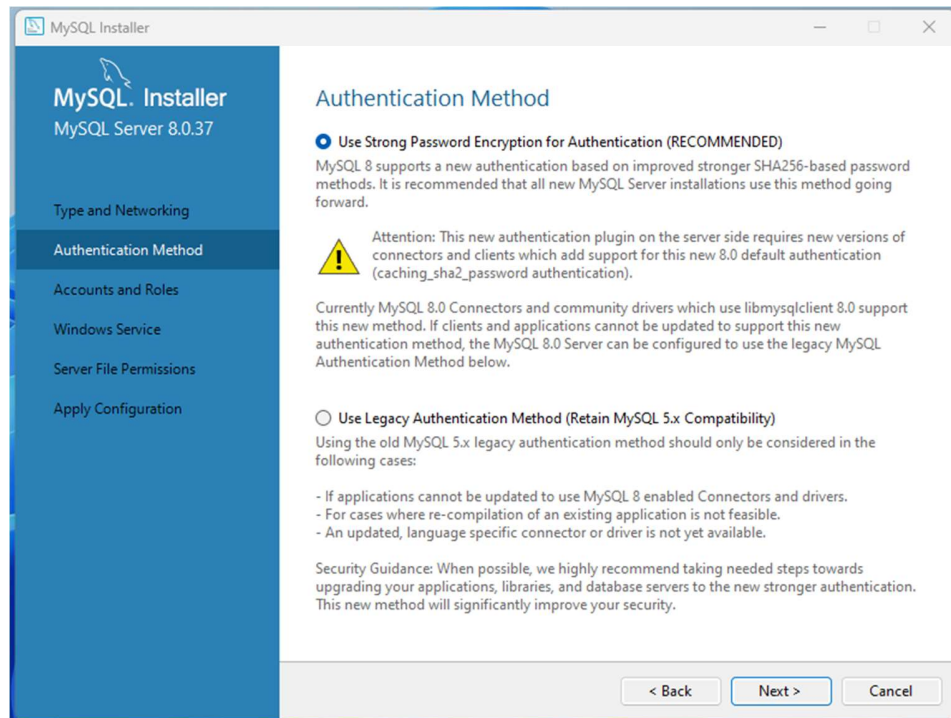
4. Clique em **Next**.



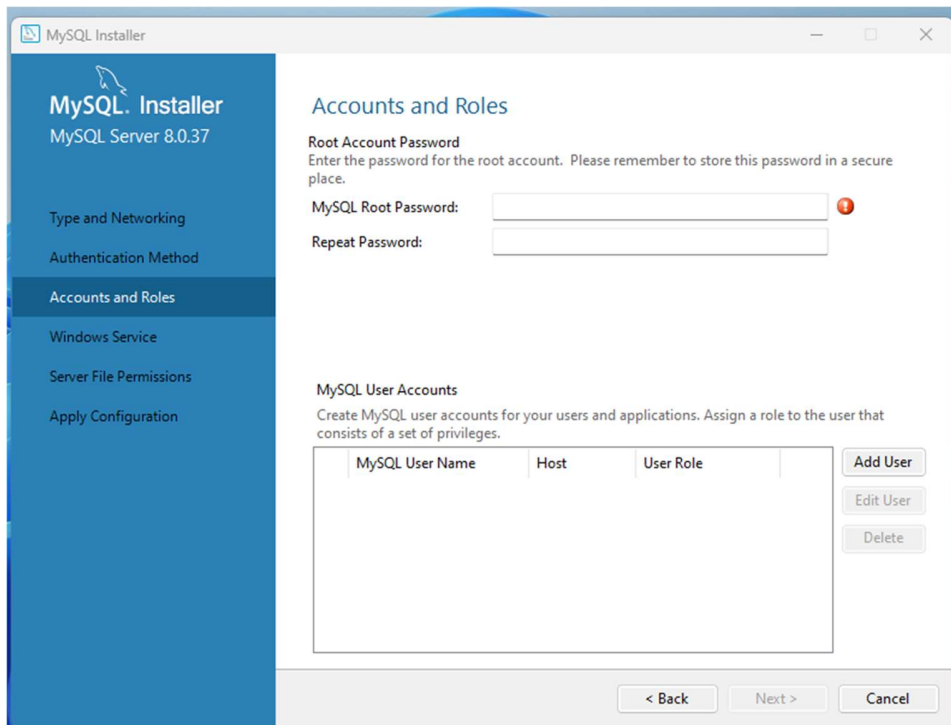
5. Clique em **Next**.



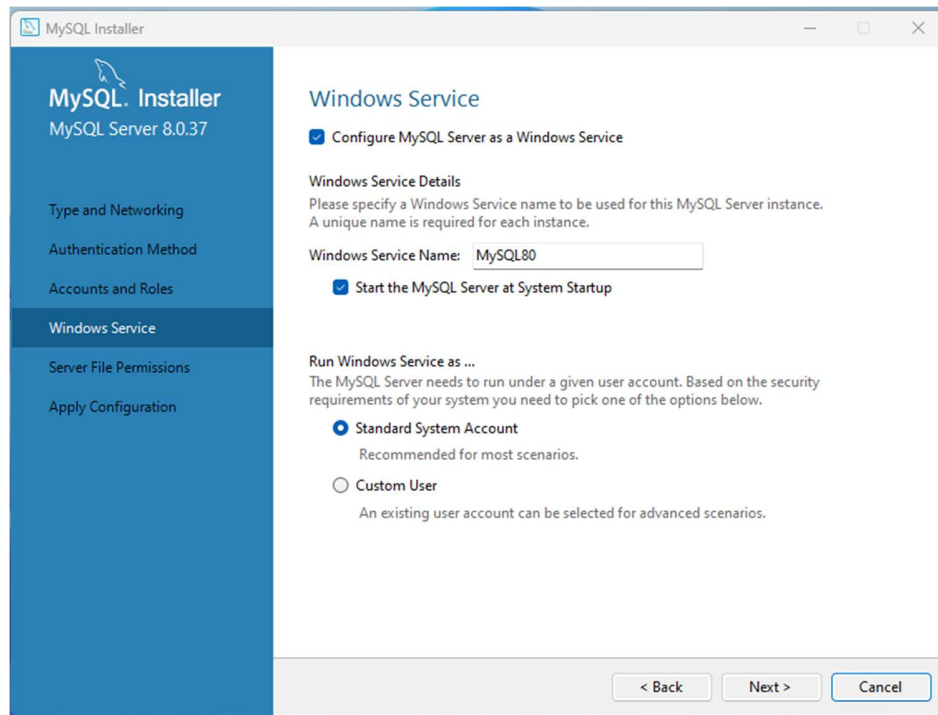
6. Clique em **Next**.



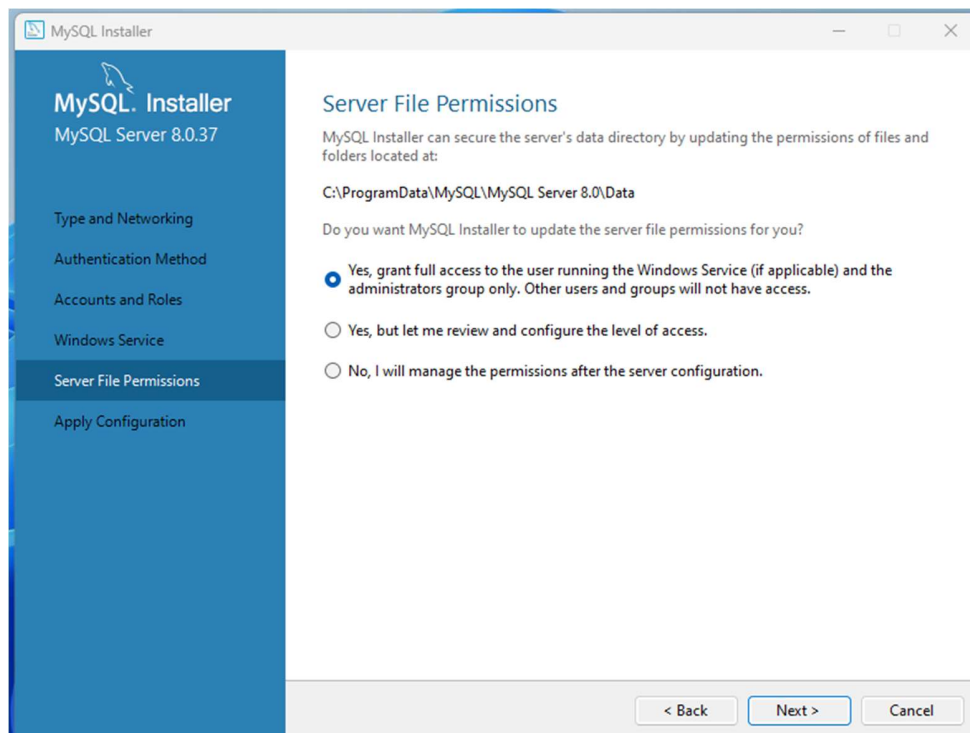
7. Crieu ma senha, que será a senha de conexão com o MySQL, depois clique em **Next**.



8. Clique em **Next**.

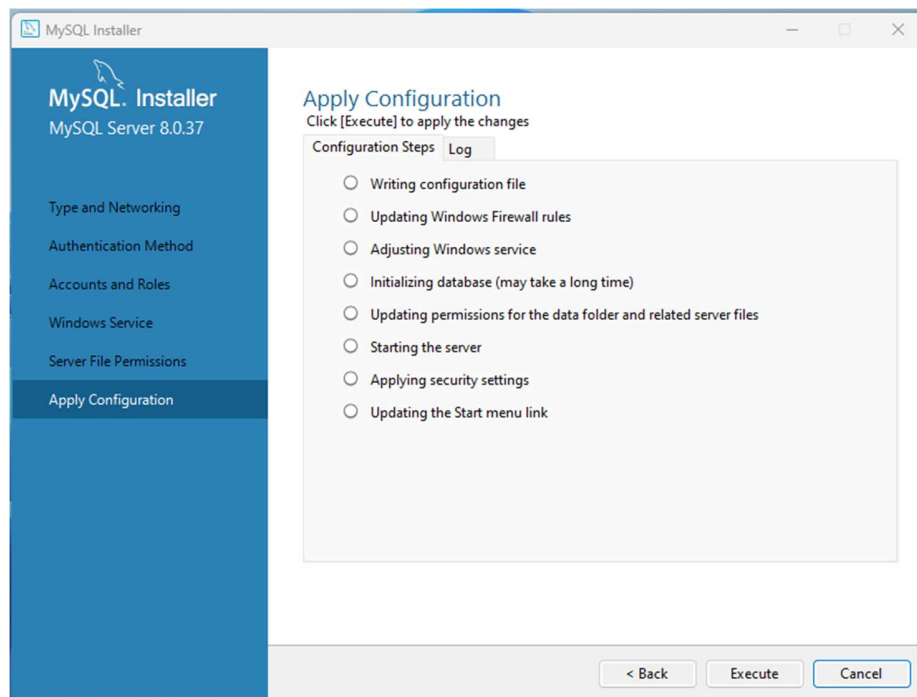


9. Clique em **Next**.

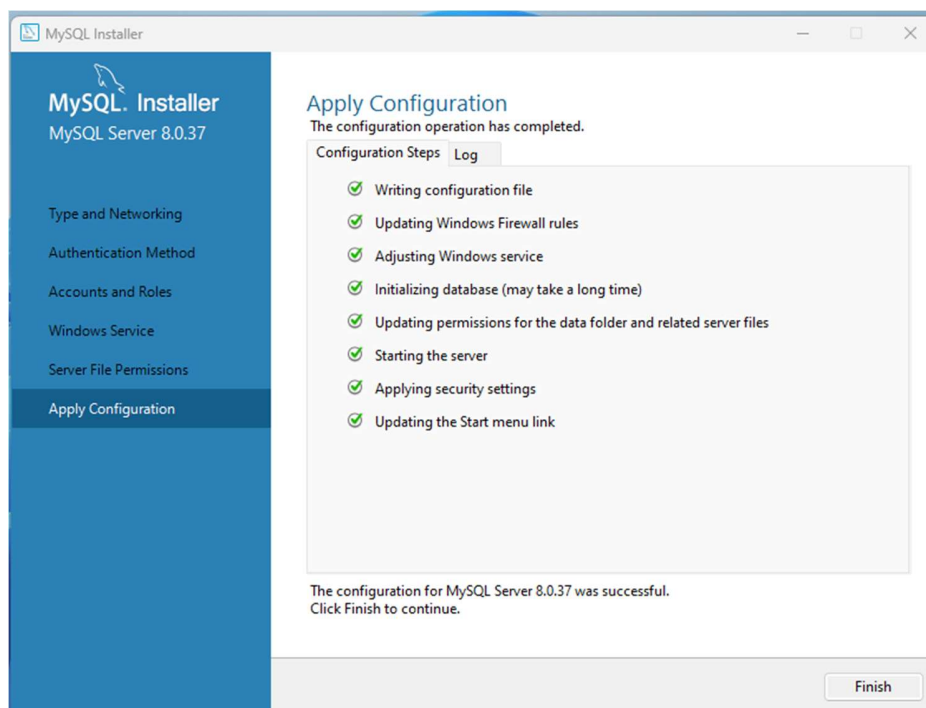




10. Clique em **Execute**.

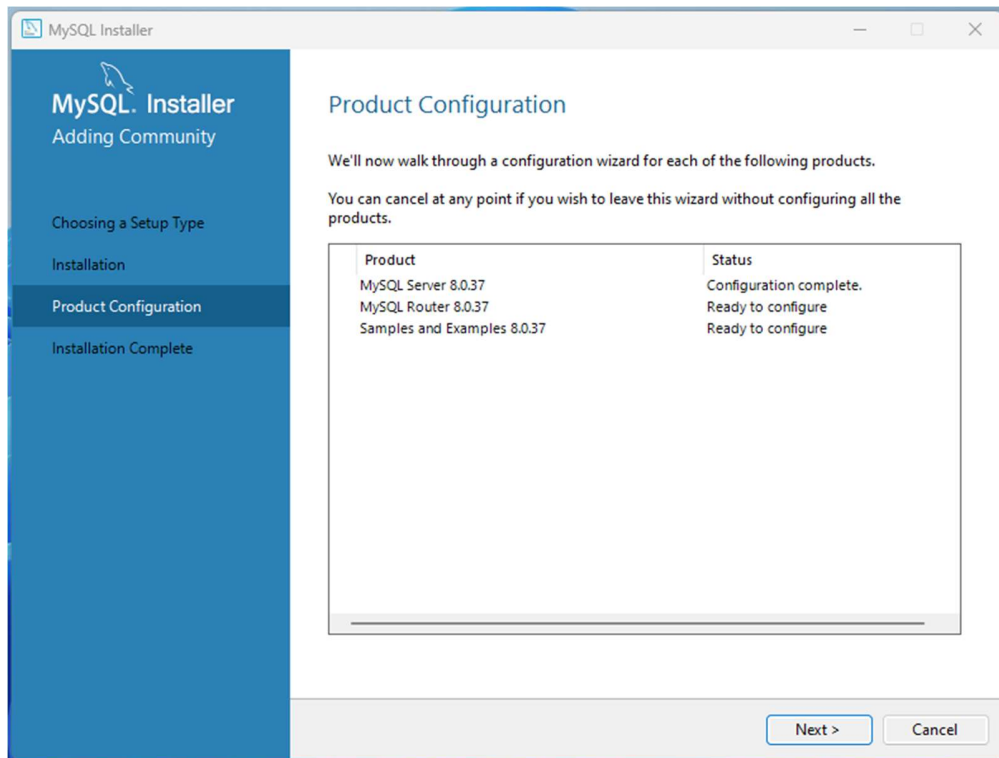


11. Clique em **Finish**.

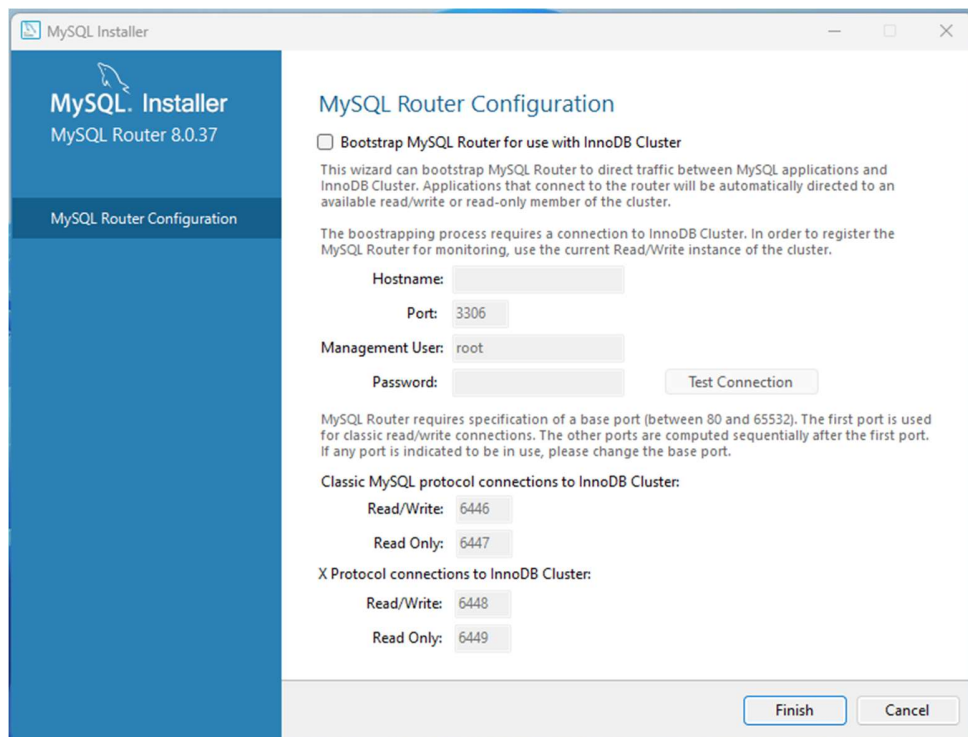




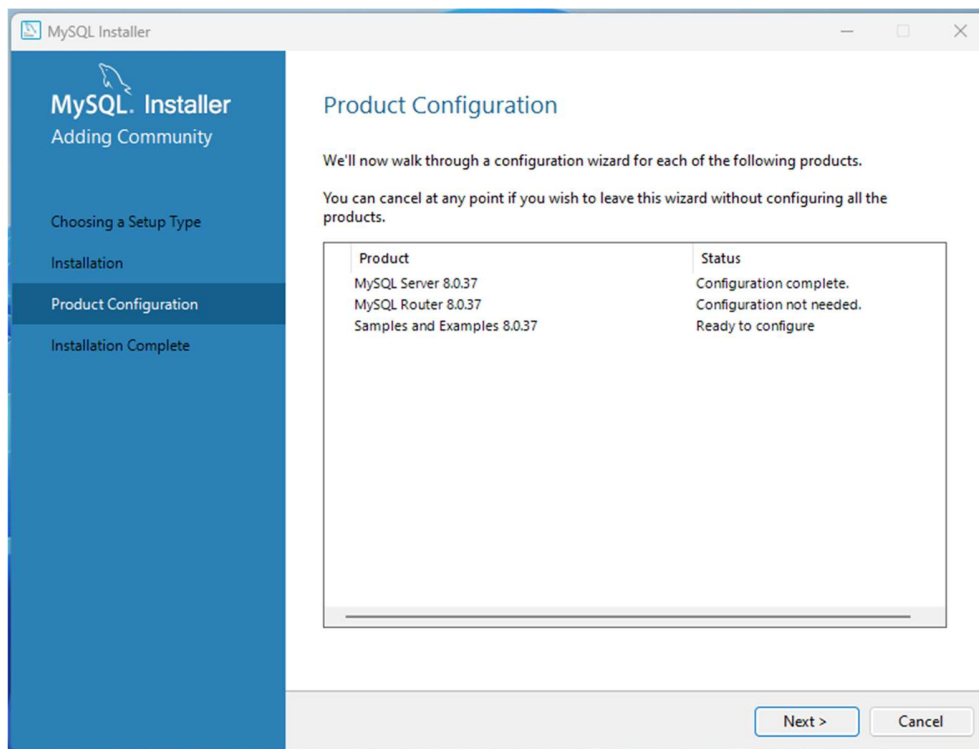
12. Clique em **Next**.



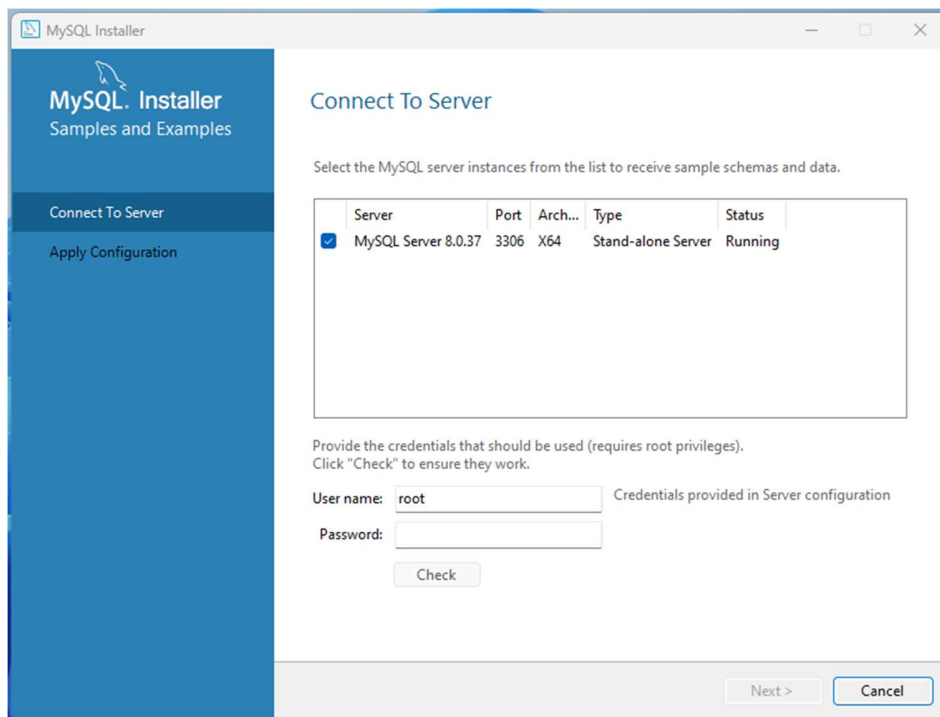
13. Clique em **Finish**.



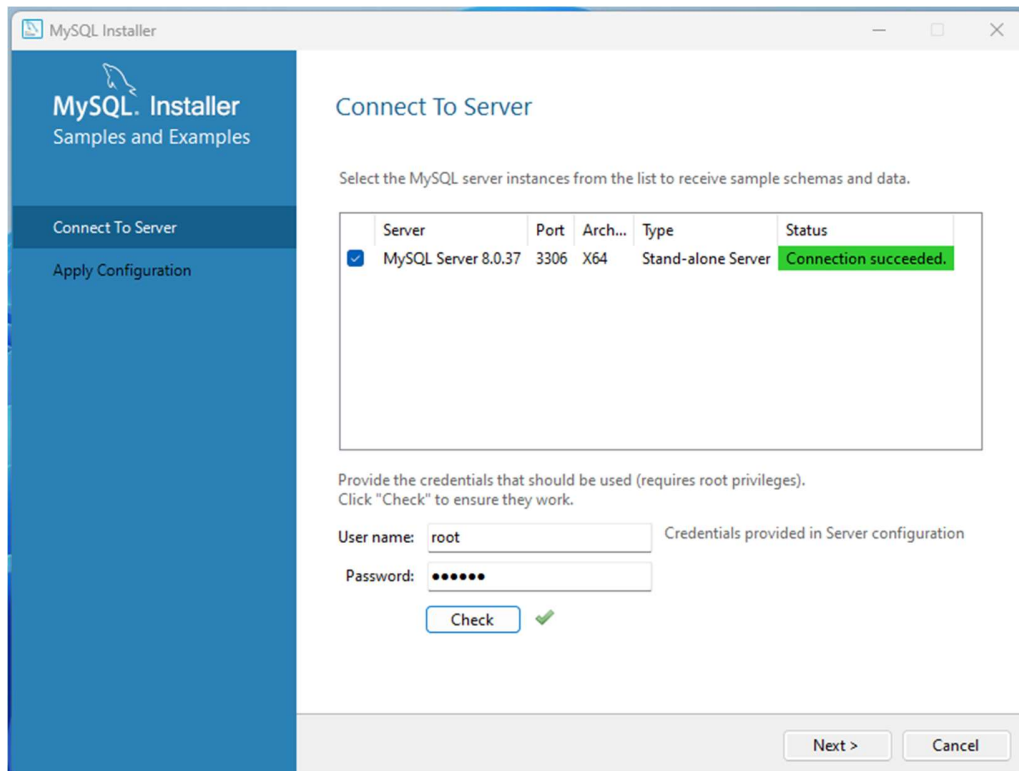
14. Clique em **Next**.



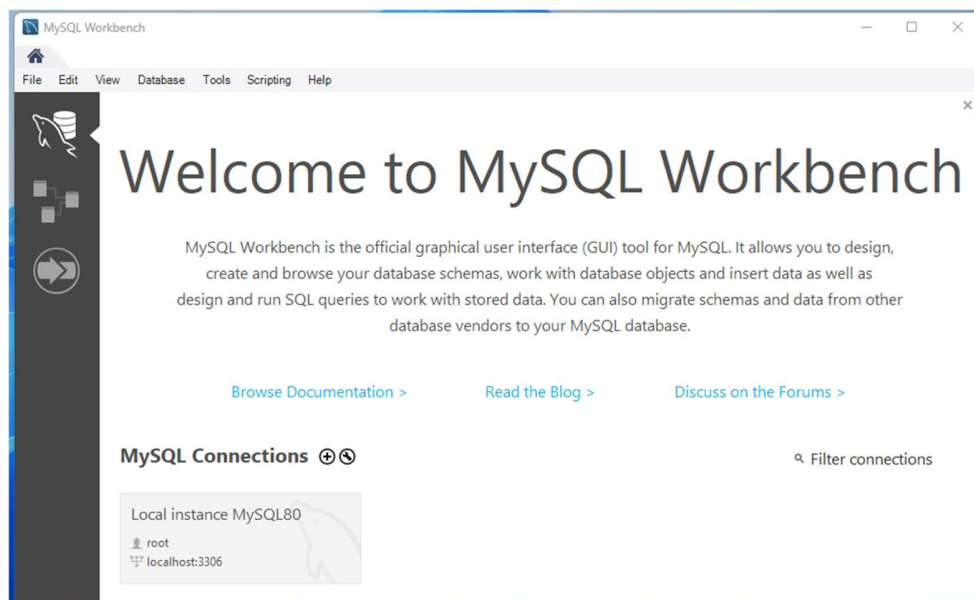
15. Digite a senha que você criou, clique em **Check**.



16. Aguarde a mensagem de sucesso e clique em **Next**.



17. Clique em **Next** até Finalizar, será exibido o **WorkBench**.

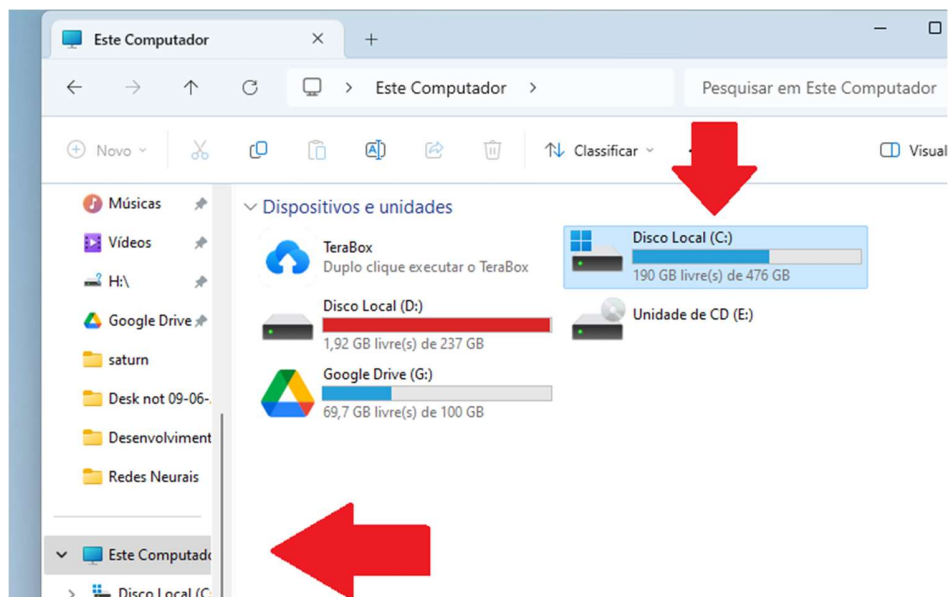


## Configurar a Variável de Ambiente:

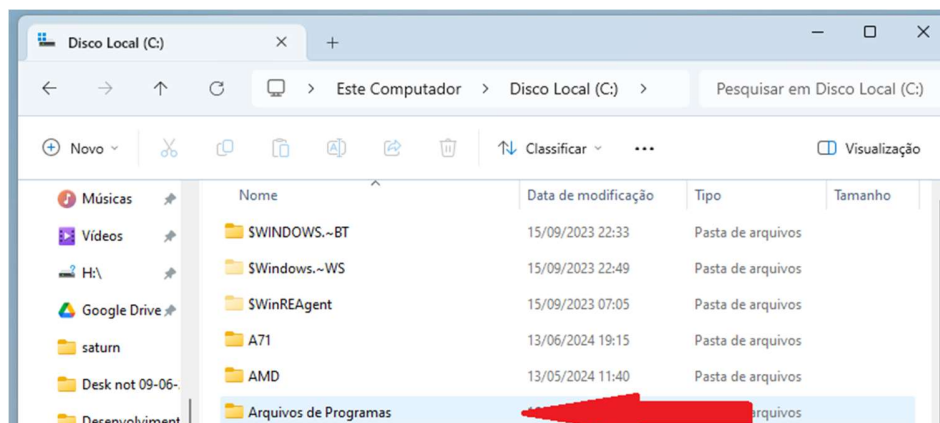
1. Abra Explorador de Arquivos.



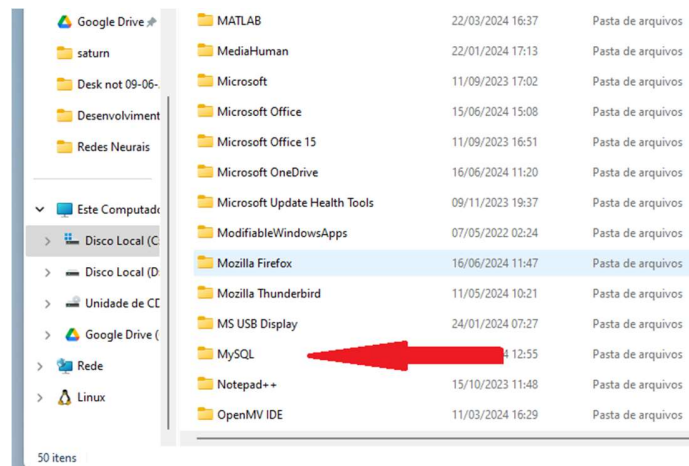
2. Clique em **Este Computador**, depois de dois Cliques em **Disco Local (C:)**:



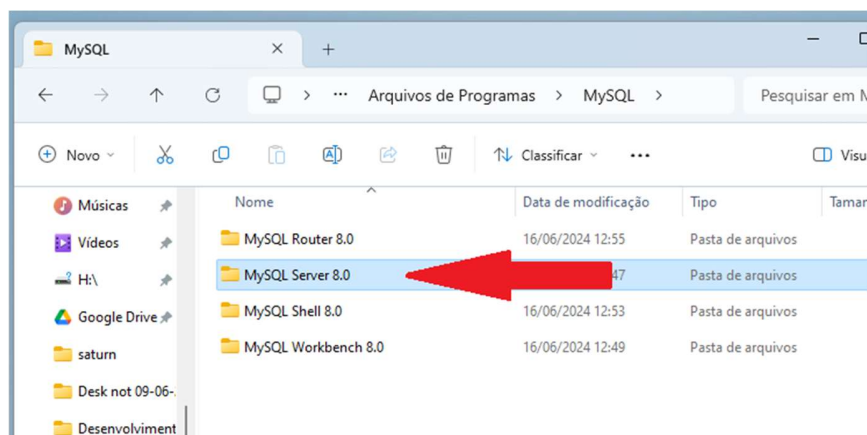
3. De dois cliques em **Arquivos de Programas**.



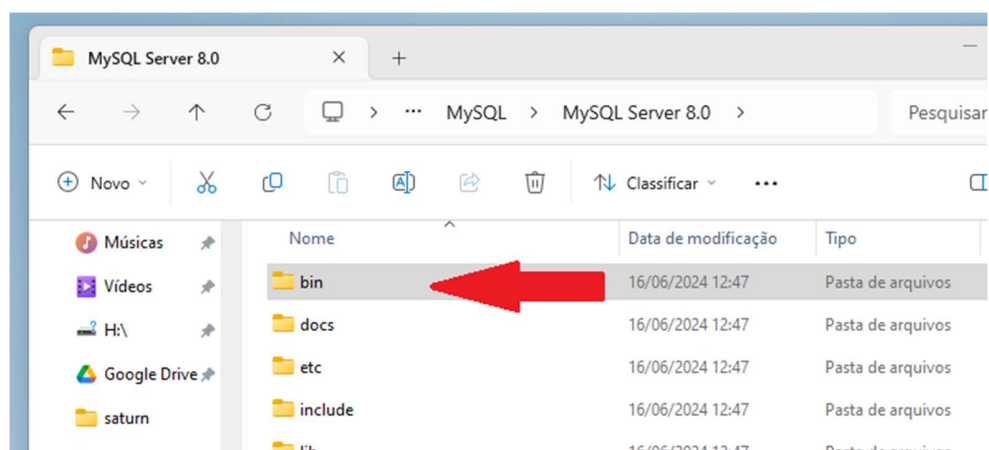
4. De dois cliques em **MySQL**.



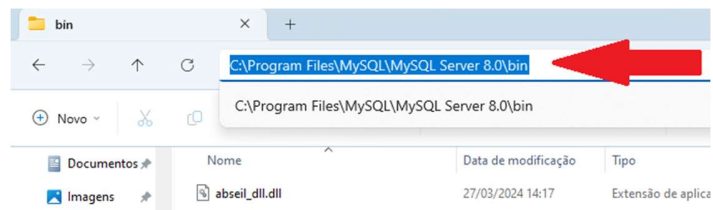
5. De dois cliques em **MySQL Server 8.0**.



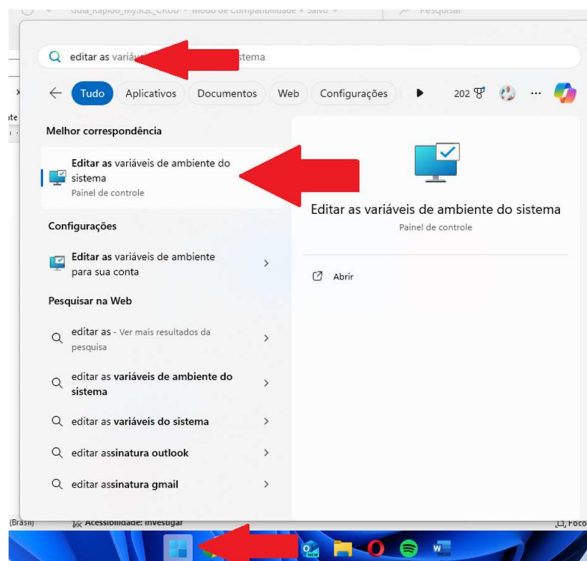
6. De dois cliques em **Bin**.



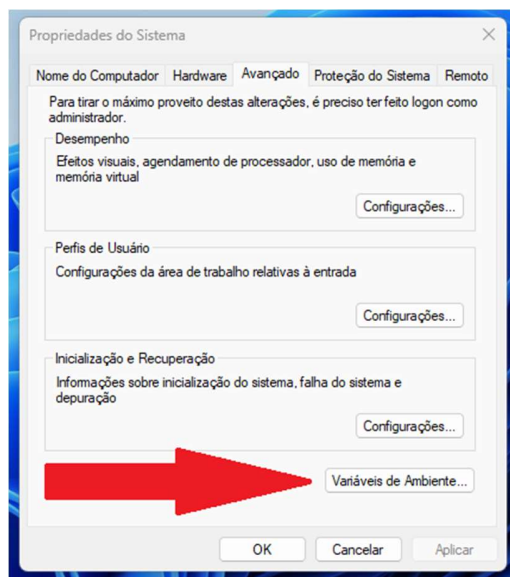
6. Clique na barra de endereço após o bin e copie o caminho.



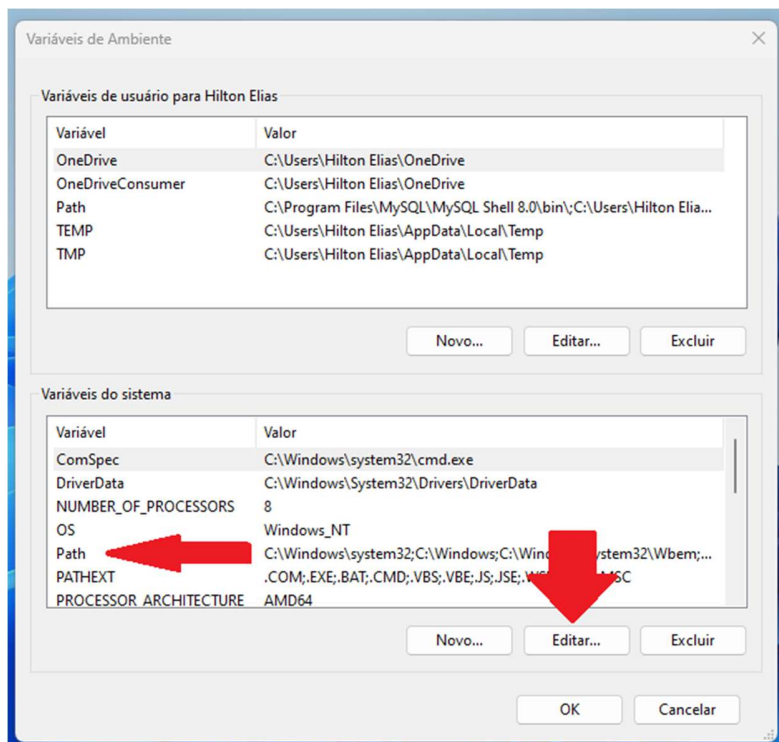
7. Clique na bandeira do Windows, digite **Editar as variáveis de ambiente do sistema** e abra.



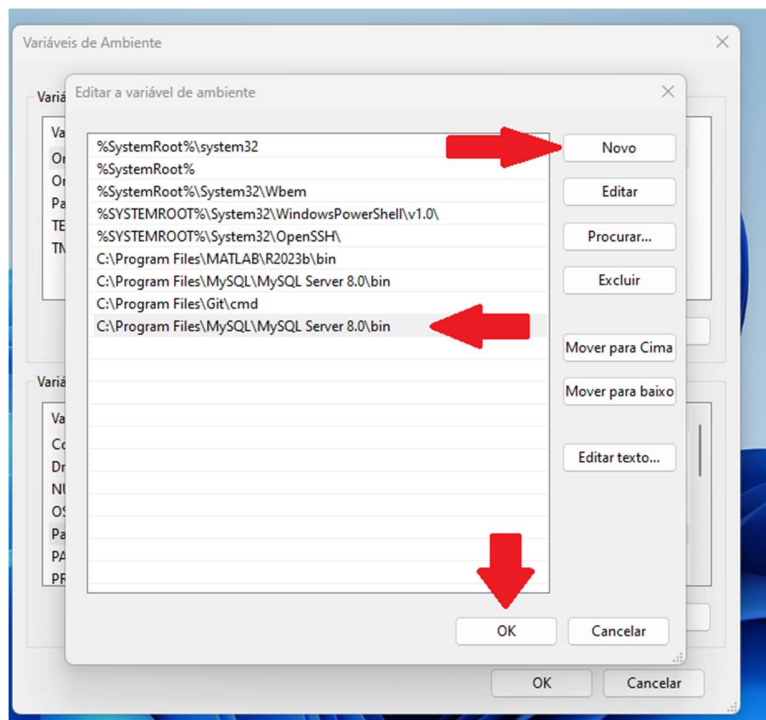
8. Clique em **variáveis de ambiente...**



9. Clique em **Path**, depois em **Editar...**



10. Clique em **Novo**, depois cole o endereço copiado do MySQL Server e depois em **OK** e **OK** para fechar as Variáveis de Ambiente.

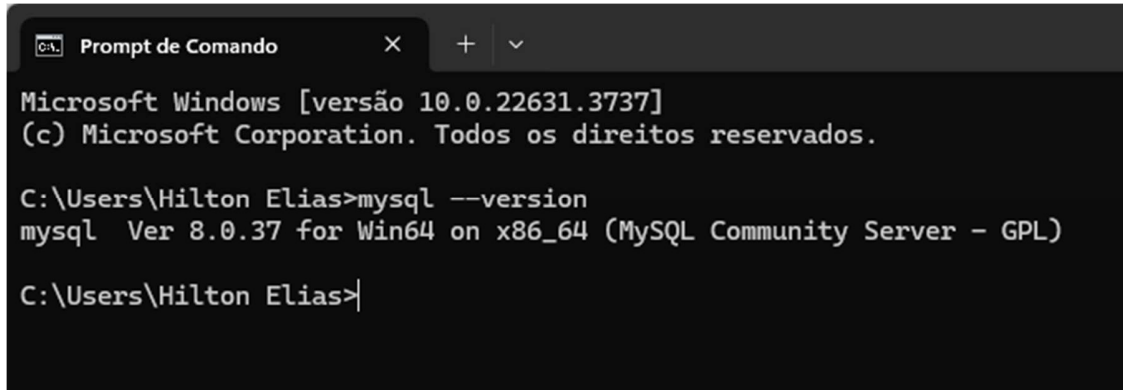




### Acesso ao MySQL:

1. Abra o terminal ou o prompt de comando e teste digitando o comando:

**mysql --version**



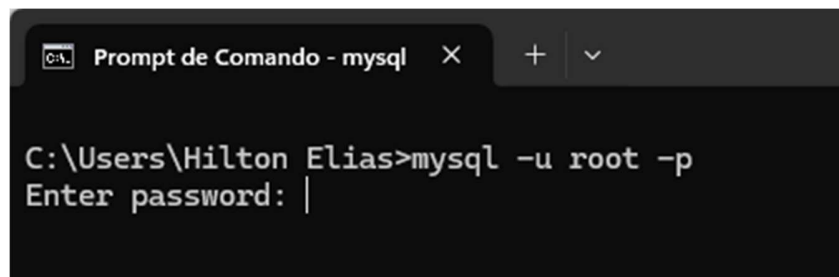
```
Prompt de Comando X + v
Microsoft Windows [versão 10.0.22631.3737]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Hilton Elias>mysql --version
mysql Ver 8.0.37 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\Hilton Elias>
```

2. Conecte-se ao MySQL usando o comando:

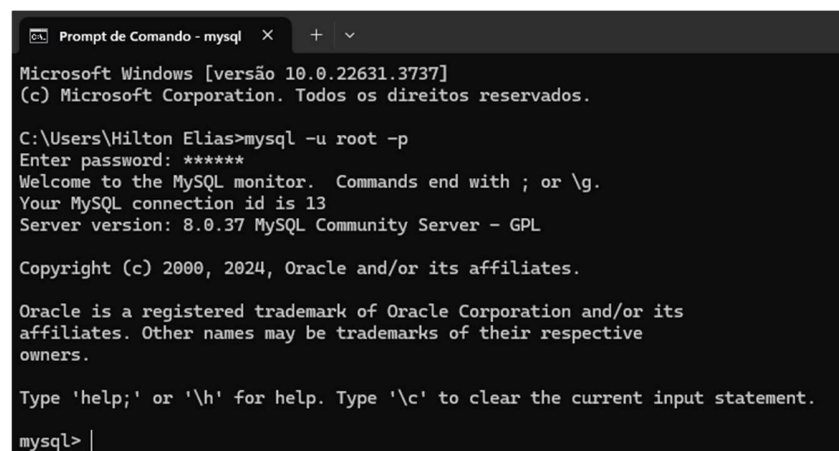
**mysql -u root -p**



```
Prompt de Comando - mysql X + v

C:\Users\Hilton Elias>mysql -u root -p
Enter password: |
```

3. Digite a senha que você configurou.



```
Prompt de Comando - mysql X + v
Microsoft Windows [versão 10.0.22631.3737]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Hilton Elias>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

4. Visualizando as Bases de Dados.

**show databases;**

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sakila     |
| sys       |
| world     |
+-----+
6 rows in set (0.00 sec)

mysql>
```

5. Acessando uma Base de Dados.

**use world**

```
mysql> use world
Database changed
mysql> |
```

6. Visualizando as Tabelas.

**show tables;**

```
mysql> show tables;
+-----+
| Tables_in_world |
+-----+
| city            |
| country         |
| countrylanguage |
+-----+
3 rows in set (0.02 sec)

mysql> |
```

7. Visualizando as Colunas de uma Tabela.

**describe city;**

```
mysql> describe city;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| ID         | int       | NO   | PRI | NULL    | auto_increment |
| Name       | char(35)  | NO   |     |          |                 |
| CountryCode | char(3)   | NO   | MUL |          |                 |
| District   | char(20)  | NO   |     |          |                 |
| Population | int       | NO   |     | 0        |                 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> |
```

## Criando um Banco de Dados e uma Tabela

**Criar um banco de dados:**

```
CREATE DATABASE meu_banco;
```

**Corrigindo o nome do banco de dados (renomeando):**

Se você cometeu um erro ao nomear o banco de dados, você pode renomeá-lo assim:

```
ALTER DATABASE nome_antigo MODIFY NAME nome_novo;
```

- **Nota: MySQL não permite renomear bancos de dados diretamente. É necessário criar um novo banco de dados e mover as tabelas do banco antigo para o novo.**

**Deletando um banco de dados:**

```
DROP DATABASE meu_banco;
```

**Usar o banco de dados criado:**

```
USE meu_banco;
```

**Criar uma tabela chamada `usuarios`:**

```
CREATE TABLE usuarios (ID INT AUTO_INCREMENT PRIMARY KEY, Nome  
VARCHAR(100), Email VARCHAR(100), Idade INT);
```

**Corrigindo o nome da tabela (renomeando):**

Se você cometeu um erro ao nomear a tabela, você pode renomeá-la assim:

```
RENAME TABLE nome_antigo TO nome_novo;
```

Exemplo:

**RENAME TABLE** usuarios **TO** usuarios; -- Corrige o nome da tabela

## Operações CRUD

### 1. Create (Criar)

Para inserir dados na tabela:

**INSERT INTO** usuarios (nome, email, idade) **VALUES** ('João', 'joao@example.com', 25);

### 2. Read (Ler)

Para buscar dados na tabela:

**SELECT** \* **FROM** usuarios;

### 3. Update (Atualizar)

Para atualizar dados na tabela:

**UPDATE** usuarios **SET** idade = 26 **WHERE** nome = 'João';

### 4. Delete (Deletar)

Para deletar dados da tabela:

**DELETE FROM** usuarios **WHERE** nome = 'João';

## Exemplos Práticos

Inserir múltiplos usuários:

**INSERT INTO** usuarios (nome, email, idade) **VALUES** ('Maria', 'maria@example.com', 30),  
('Pedro', 'pedro@example.com', 22);

**Buscar usuários acima de uma certa idade:**

```
SELECT * FROM usuarios WHERE idade > 25;
```

**Atualizar email de um usuário específico:**

```
UPDATE usuarios SET email = 'novo_email@example.com' WHERE nome = 'Pedro';
```

**Deletar todos os usuários com idade abaixo de 25 anos:**

```
DELETE FROM usuarios WHERE idade < 25;
```

### **Fechando a Conexão**

Para sair do MySQL:

```
exit
```

## **Introdução aos Bancos de Dados**

### **O Que São Bancos de Dados?**

Bancos de dados são coleções organizadas de dados que podem ser facilmente acessadas, gerenciadas e atualizadas. Eles são fundamentais para armazenar informações de forma estruturada e eficiente, permitindo a recuperação e manipulação de dados de maneira rápida e segura.

### **História dos Bancos de Dados**

Os primeiros bancos de dados surgiram na década de 1960 com sistemas hierárquicos e de rede. Na década de 1970, o modelo relacional foi introduzido por Edgar F. Codd, revolucionando a maneira como os dados eram armazenados e manipulados. A partir daí, surgiram vários tipos de bancos de dados, cada um adequado a diferentes necessidades e aplicações.

## Importância dos Bancos de Dados

Bancos de dados são essenciais em praticamente todas as áreas que envolvem o uso de informações. Eles suportam operações comerciais, científicas, educacionais e muitas outras, fornecendo uma base sólida para a tomada de decisões baseada em dados.

## Exemplificação

Imagine uma loja online que precisa armazenar informações sobre seus produtos, clientes, pedidos e pagamentos. Sem um banco de dados, essa tarefa seria caótica e propensa a erros. Um banco de dados permite que essas informações sejam organizadas em tabelas, facilitando o acesso e a manipulação dos dados de forma eficiente e segura.

## Exercícios

1. Explique, com suas palavras, o que é um banco de dados e por que ele é importante.
  2. Pesquise e liste três tipos de bancos de dados que surgiram antes do modelo relacional e suas principais características.
- 

## Tipos de Dados

### Introdução aos Tipos de Dados

Os tipos de dados são fundamentais na criação de um banco de dados, pois definem o tipo de informação que cada coluna de uma tabela pode armazenar. Eles garantem a integridade dos dados e otimizam o uso de espaço e desempenho.

### Tipos de Dados Comuns no MySQL

1. **Numéricos**
  - **INT**: Inteiros, por exemplo, a idade de uma pessoa.
  - **FLOAT e DOUBLE**: Números com ponto flutuante, como valores de temperatura ou medições científicas.
  - **DECIMAL**: Números decimais precisos, como preços de produtos.
2. **Caractere**
  - **CHAR**: Cadeias de caracteres de comprimento fixo, útil para códigos de países ou estados.
  - **VARCHAR**: Cadeias de caracteres de comprimento variável, ideal para nomes e endereços.

### 3. Datas e Horas

- **DATE:** Datas no formato AAAA-MM-DD, como datas de nascimento.
- **TIME:** Horas no formato HH:MM

, como horários de abertura de uma loja.

- **DATETIME e TIMESTAMP:** Combinações de data e hora, úteis para registros de transações.

### 4. Tipos de Dados Textuais

- **TEXT:** Texto de até 65.535 caracteres, como descrições detalhadas de produtos.
- **BLOB:** Armazenamento de dados binários grandes, como imagens ou arquivos PDF.

### Exemplos

**CREATE TABLE** Cliente (ID **INT PRIMARY KEY**, Nome **VARCHAR(100)**, DataNascimento **DATE**, Saldo **DECIMAL(10, 2)**);

Neste exemplo, a tabela Cliente tem colunas para armazenar um identificador único (ID), um nome (Nome), uma data de nascimento (DataNascimento) e um saldo financeiro (Saldo).

### Exercícios

1. Defina um tipo de dado apropriado para cada uma das seguintes colunas:
  - Nome do produto
  - Quantidade em estoque
  - Preço do produto
  - Data de fabricação
2. Crie uma tabela chamada Funcionario com as seguintes colunas: ID, Nome, Salario e DataContratacao. Escolha os tipos de dados mais apropriados para cada coluna.

---

## Tipos de Banco de Dados

### Bancos de Dados Relacionais (RDBMS)

- Estruturados em tabelas.



- Utilizam SQL (Structured Query Language).
- Exemplo: MySQL, PostgreSQL, Oracle.

### **Bancos de Dados Não-Relacionais (NoSQL)**

- Armazenamento de dados em documentos, gráficos, chave-valor, entre outros.
- Flexíveis e escaláveis.
- Exemplo: MongoDB, Cassandra, Redis.

### **Exemplificação**

Um banco de dados relacional pode ser comparado a uma planilha de Excel com várias tabelas relacionadas entre si por meio de chaves primárias e estrangeiras. Por outro lado, um banco de dados NoSQL pode ser comparado a uma coleção de arquivos JSON onde cada arquivo representa um documento com estrutura flexível.

### **Exercícios**

1. Compare bancos de dados relacionais e não-relacionais, destacando suas principais diferenças e exemplos de uso.
2. Pesquise sobre um banco de dados NoSQL e descreva suas principais características e vantagens.

---

## **Sistemas de Gerenciamento de Banco de Dados (SGBD)**

### **Funções de um SGBD**

- Criação, atualização e manutenção de bancos de dados.
- Controle de acesso e segurança.
- Backup e recuperação de dados.
- Suporte a transações.

### **Exemplos de SGBDs**

- **MySQL:** Amplamente utilizado para aplicações web.
- **PostgreSQL:** Conhecido por sua robustez e conformidade com padrões SQL.
- **Oracle:** Usado em grandes corporações com requisitos avançados de processamento de transações.

### **Exemplificação**

Considere um site de comércio eletrônico com milhões de usuários e transações diárias. Um SGBD como o MySQL gerencia todos os dados de usuários, produtos e pedidos, garantindo que as transações sejam realizadas de forma segura e eficiente.

### **Exercícios**

1. Explique a função de um SGBD e cite três exemplos de SGBDs populares.
2. Pesquise sobre as diferenças entre MySQL e PostgreSQL e liste suas vantagens e desvantagens.