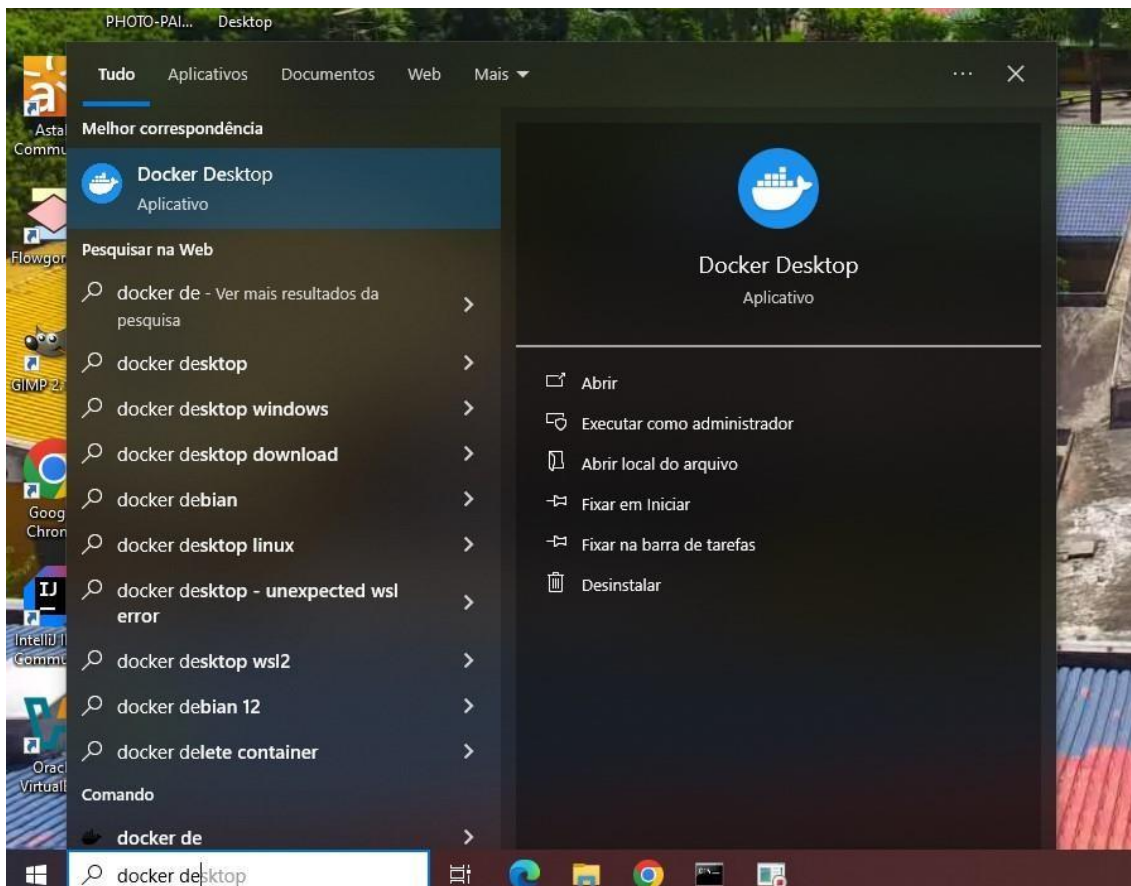
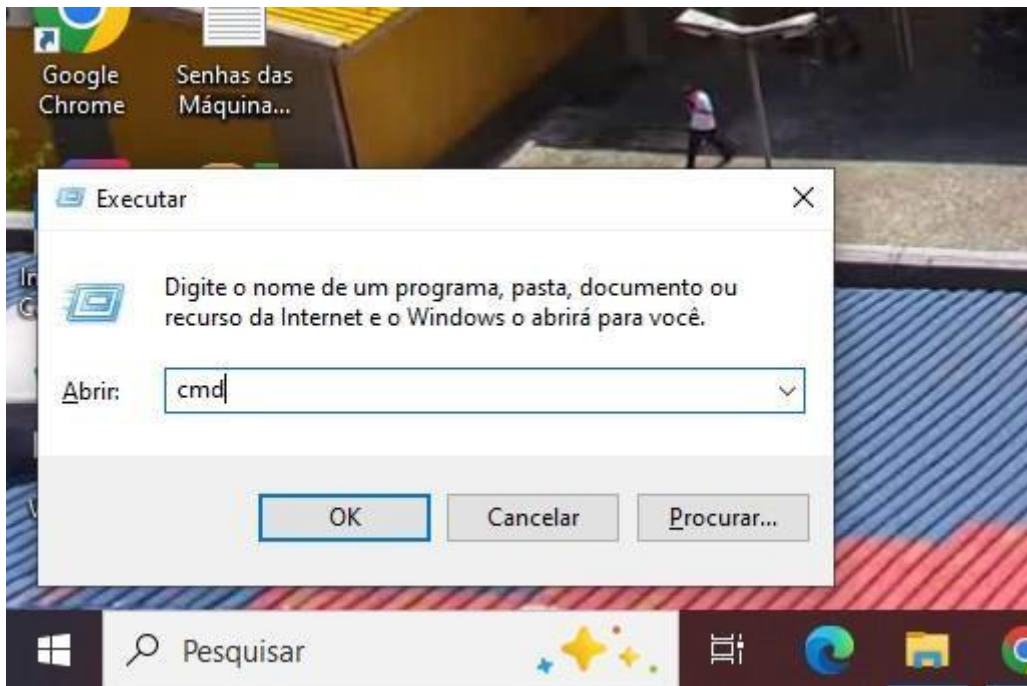


Abram o Docker Desktop, no menu iniciar

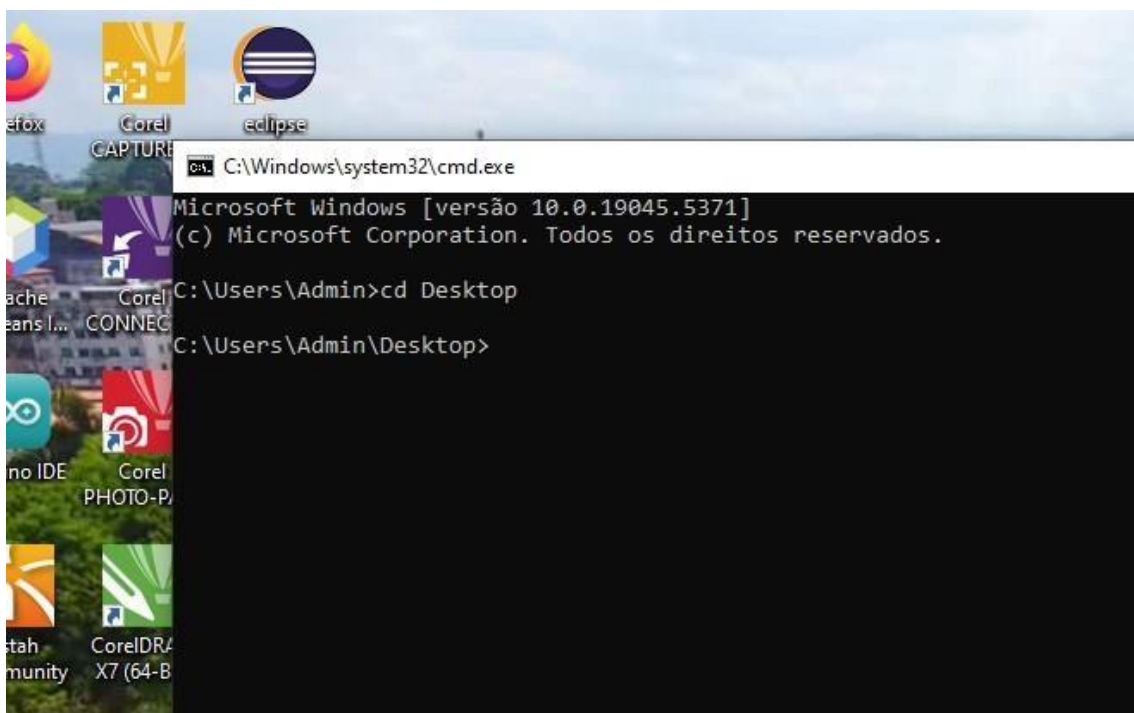


Em seguida abram o terminal (CMD)



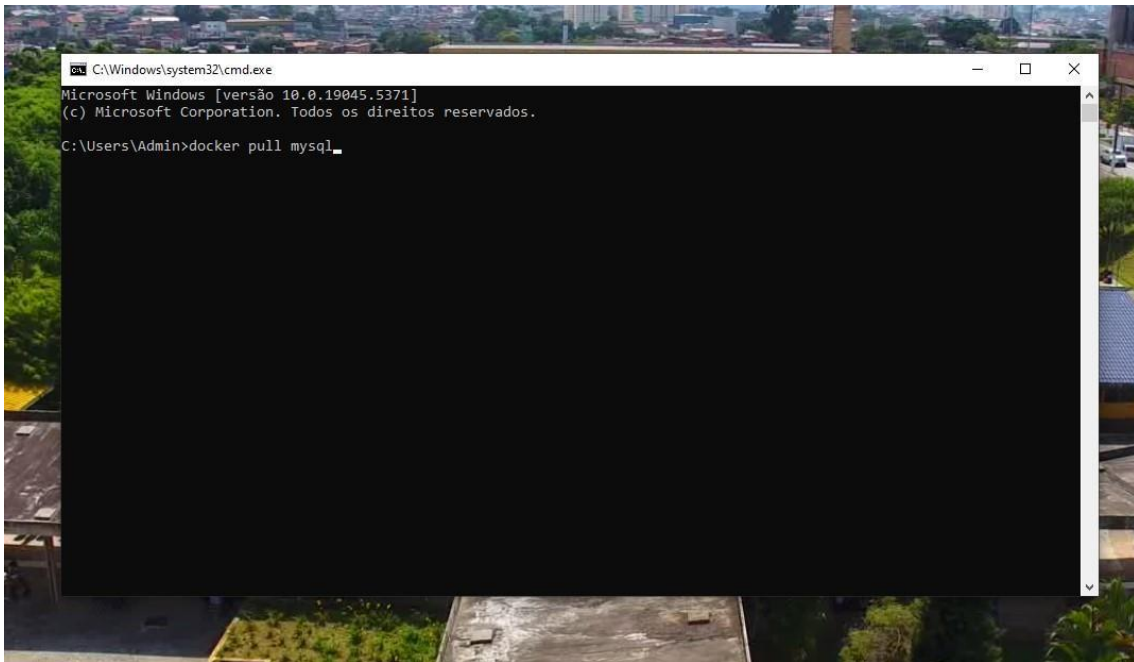


Vá para a área de trabalho (Desktop)



Faça o pull na imagem mysql

O pull é o download da imagem docker, existem diversas imagens diferentes com propósitos diferentes, no caso, a imagem mysql é focada em banco de dados mysql



Depois do download da imagem, use o código:

```
docker run --name mysql-workshop -e
MYSQL_ROOT_PASSWORD=1234 -v dados_mysql:/var/lib/mysql -
v C:\Users\Admin\Desktop\scripts:/var/scripts -p 3306:3306 -d
mysql:latest
```



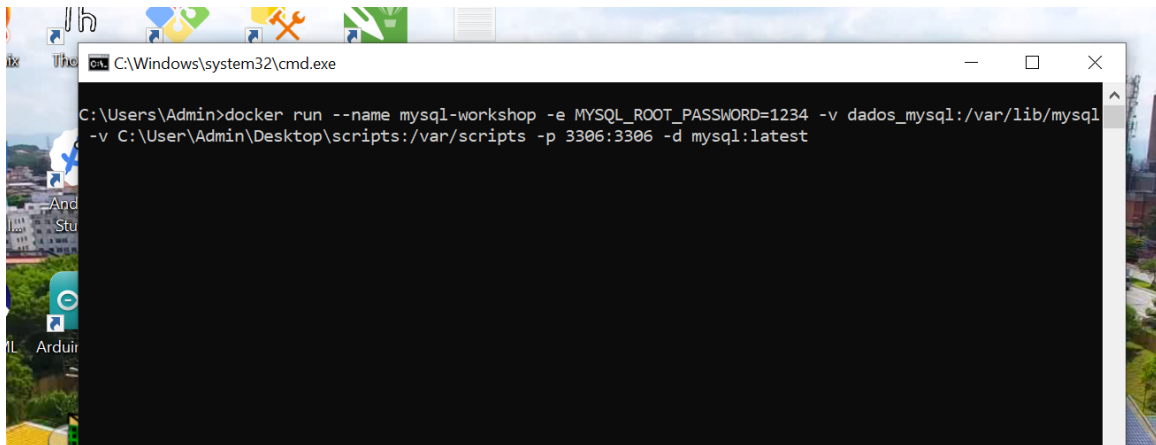
Explicação do código:

Docker Run: Roda o Docker e cria e inicia um novo container;

--name mysql-workshop: Nomeia o novo container;

- e **MYSQL_ROOT_PASSWORD=1234**: Coloca uma variável de ambiente dentro do container e seta a senha root para 1234;
- v **dados_mysql:/var/lib/mysql**: Cria uma pasta volume que é controlada e gerenciada pelo próprio docker
- v **C:\Users\Admin\Desktop\scripts:/var/scripts**: Cria uma pasta volume (scripts) entre o desktop e o container, ambos vão compartilhar o conteúdo da pasta;
- p **3306:3006**: Conecta a porta 3306 do desktop com a porta 3006 do container;
- d **mysql:latest**: Especifica qual imagem usar;

Executando no terminal

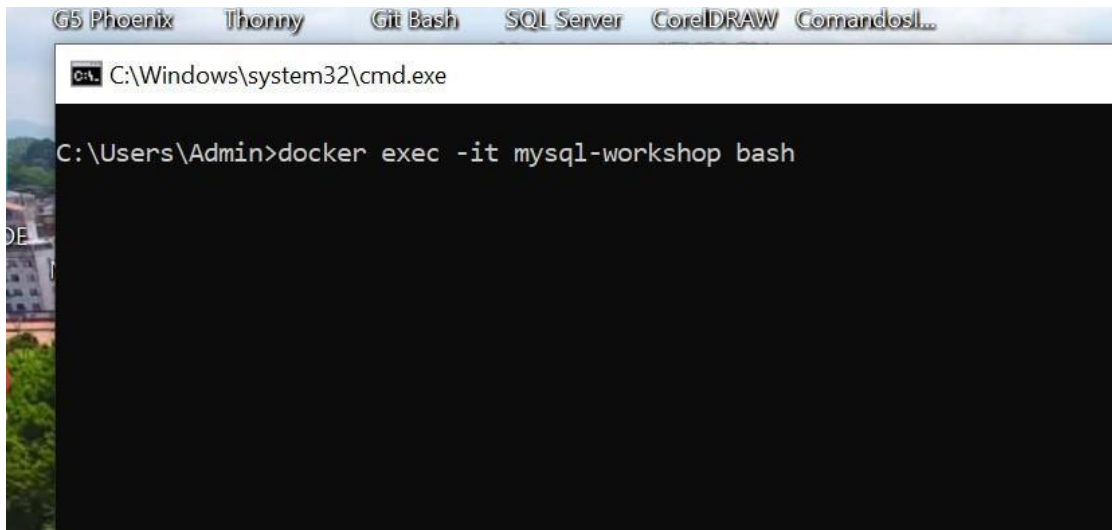


Agora entramos no bash do container com o código:

docker exec -it mysql-workshop bash

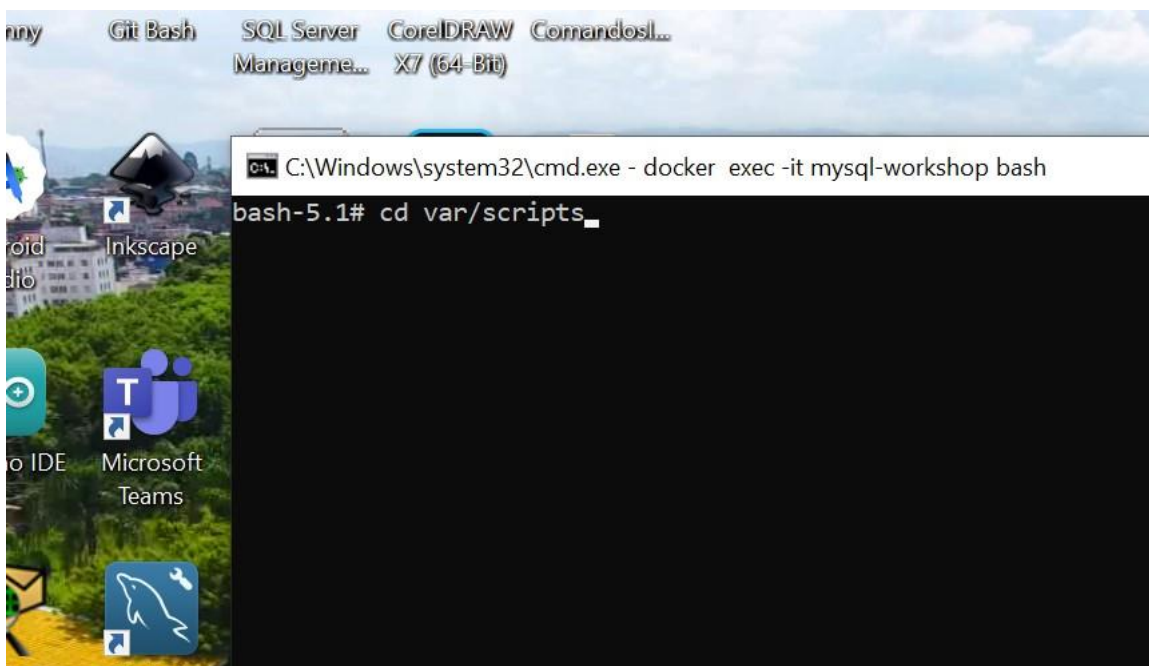
e vá para a pasta mysql (a pasta volume) com o código:

cd /var/scripts



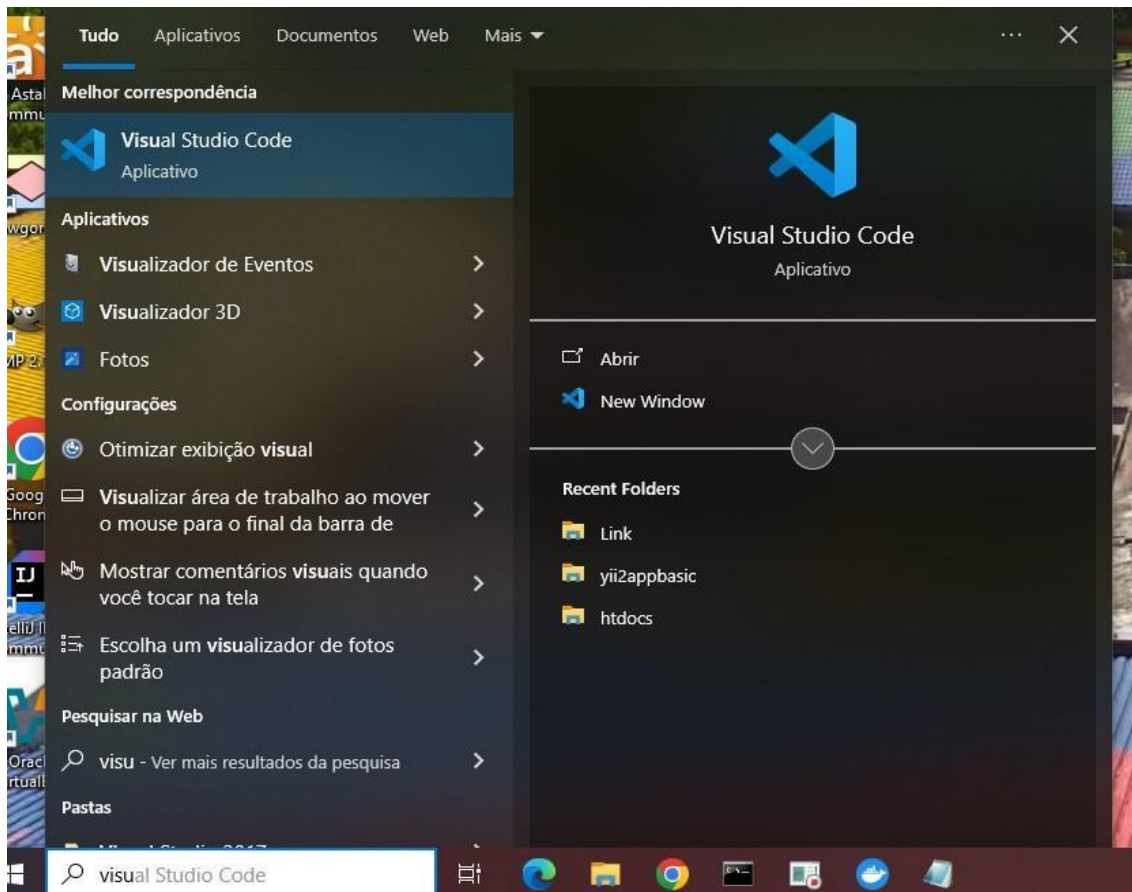
```
C:\Windows\system32\cmd.exe

C:\Users\Admin>docker exec -it mysql-workshop bash
```

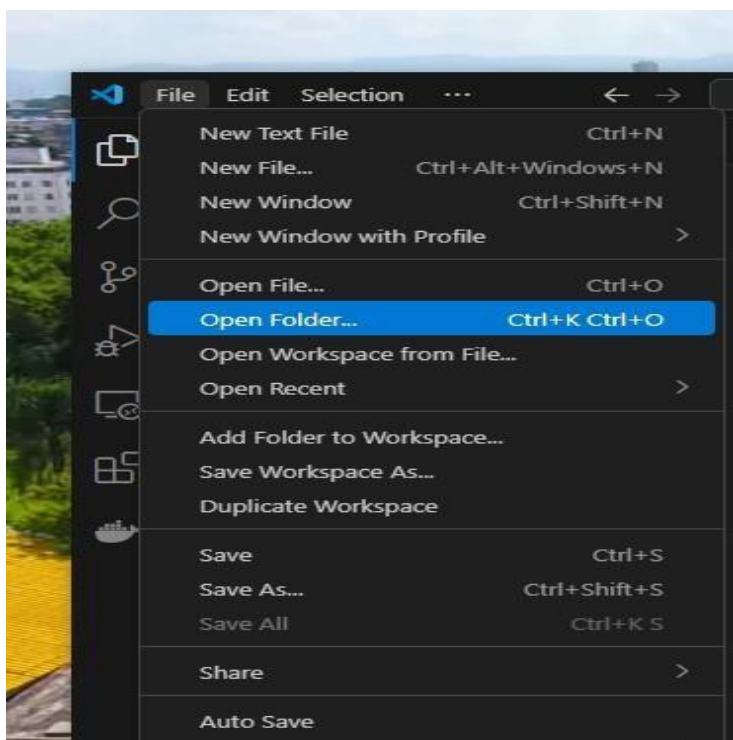


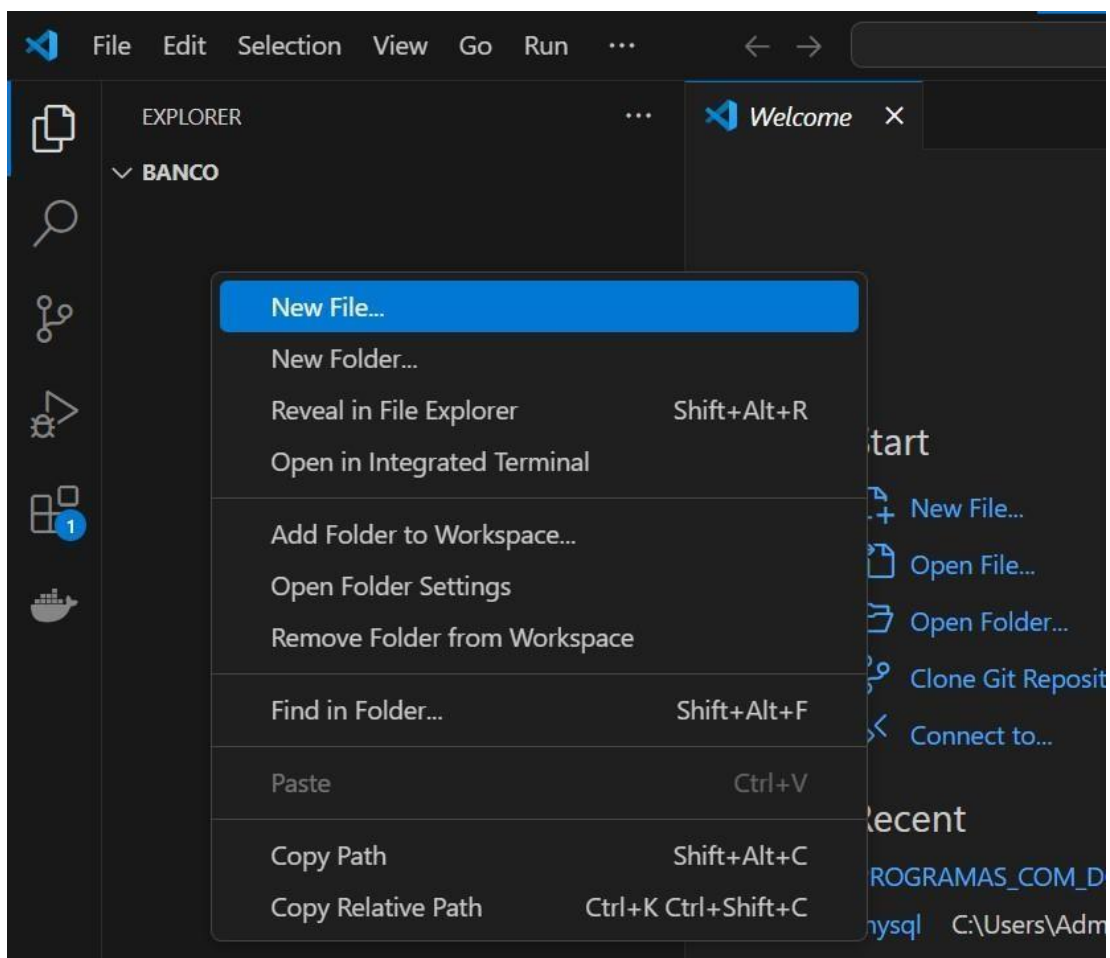
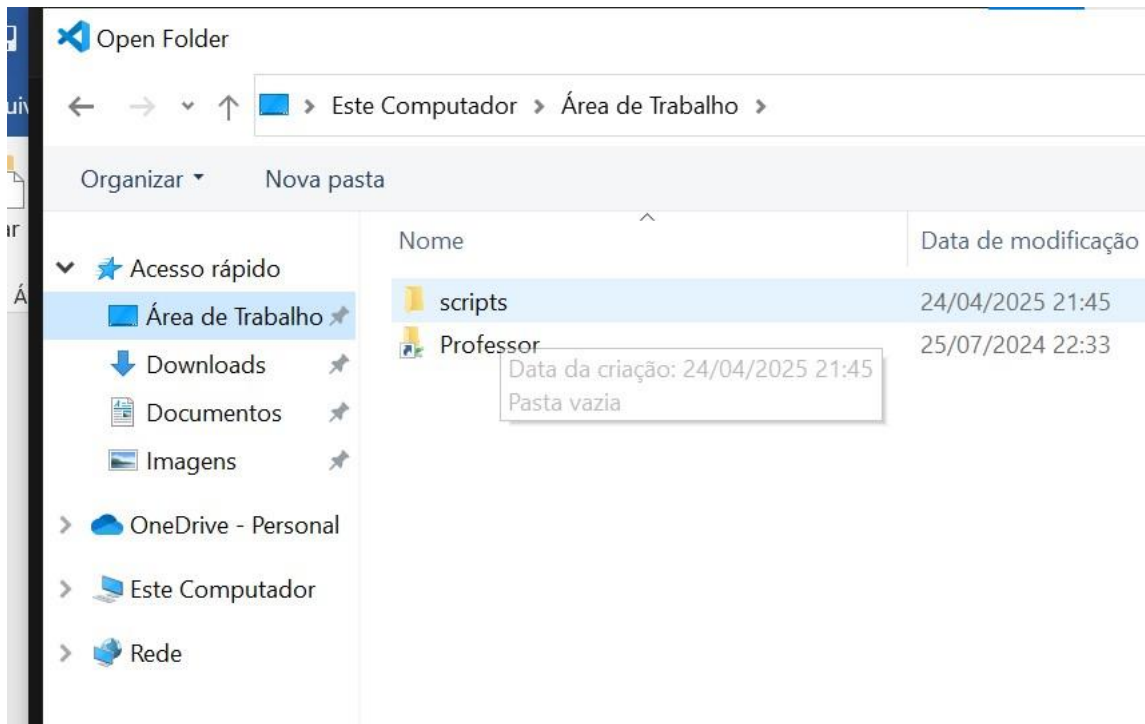
```
C:\Windows\system32\cmd.exe - docker exec -it mysql-workshop bash
bash-5.1# cd var/scripts_
```

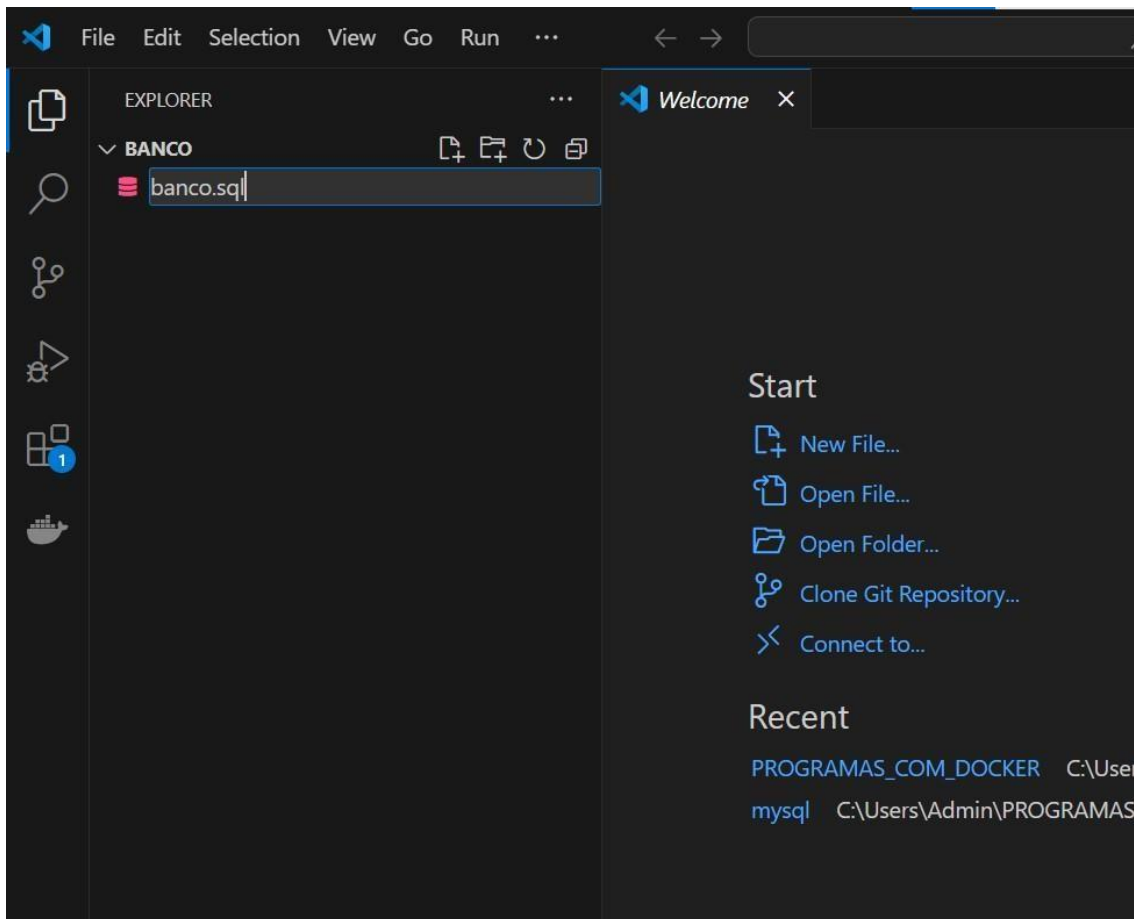
De volta no desktop, abra o visual studio code



E abrir a pasta scripts na área de trabalho e criar um novo arquivo “banco.sql”;







Cole o código abaixo no arquivo:

```
CREATE DATABASE IF NOT EXISTS test_db;
```

```
USE test_db;
```

```
CREATE TABLE IF NOT EXISTS clientes (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    nome VARCHAR(50) NOT NULL,
```

```
    email VARCHAR(100) UNIQUE,
```

```
    idade INT,
```

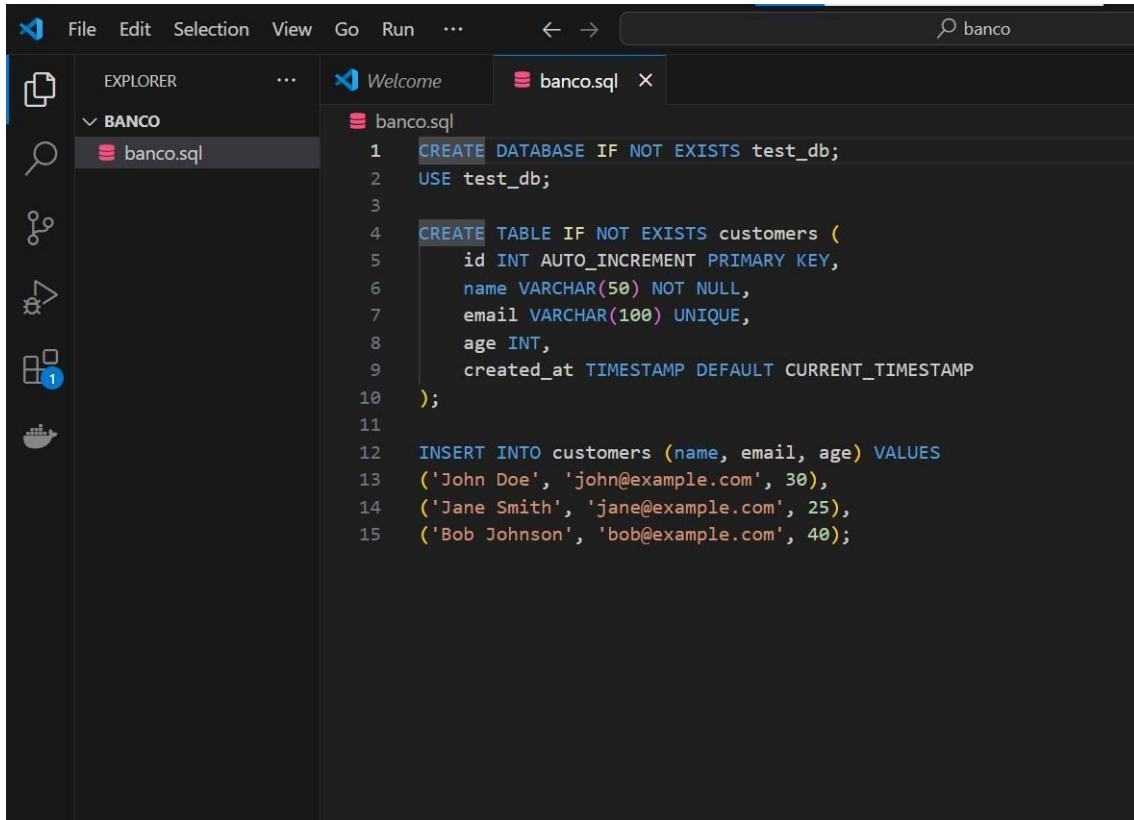
```
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

```
);
```

```
INSERT INTO clientes (nome, email, idade) VALUES
```



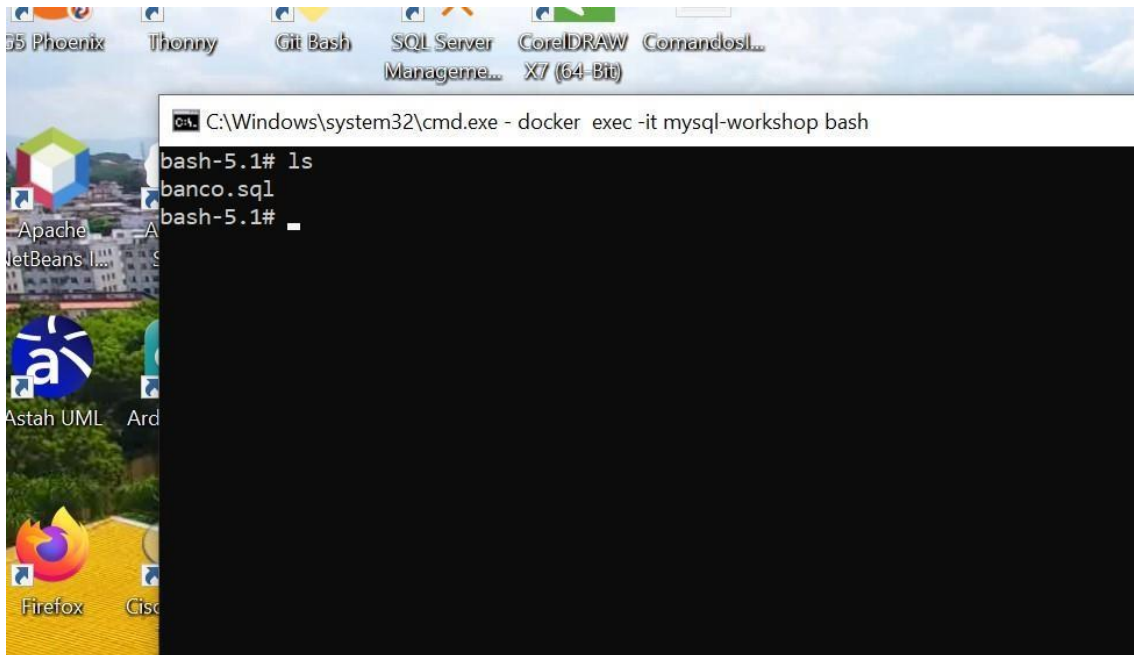
```
('John Doe', 'john@example.com', 30),  
('Jane Smith', 'jane@example.com', 25),  
('Bob Johnson', 'bob@example.com', 40);
```

A screenshot of a code editor interface. The Explorer panel on the left shows a folder named 'BANCO' containing a file 'banco.sql'. The main editor area displays the contents of 'banco.sql', which is a SQL script. The script starts with creating a database 'test_db' and using it. Then, it creates a table 'customers' with columns 'id' (auto-increment primary key), 'name' (varchar 50), 'email' (varchar 100, unique), 'age' (int), and 'created_at' (timestamp default current timestamp). Finally, it inserts three rows of customer data: John Doe, Jane Smith, and Bob Johnson.

```
1 CREATE DATABASE IF NOT EXISTS test_db;  
2 USE test_db;  
3  
4 CREATE TABLE IF NOT EXISTS customers (  
5     id INT AUTO_INCREMENT PRIMARY KEY,  
6     name VARCHAR(50) NOT NULL,  
7     email VARCHAR(100) UNIQUE,  
8     age INT,  
9     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
10 );  
11  
12 INSERT INTO customers (name, email, age) VALUES  
13 ('John Doe', 'john@example.com', 30),  
14 ('Jane Smith', 'jane@example.com', 25),  
15 ('Bob Johnson', 'bob@example.com', 40);
```

De volta para o bash, verifique se o arquivo banco.sql esta na pasta com o comando:

ls



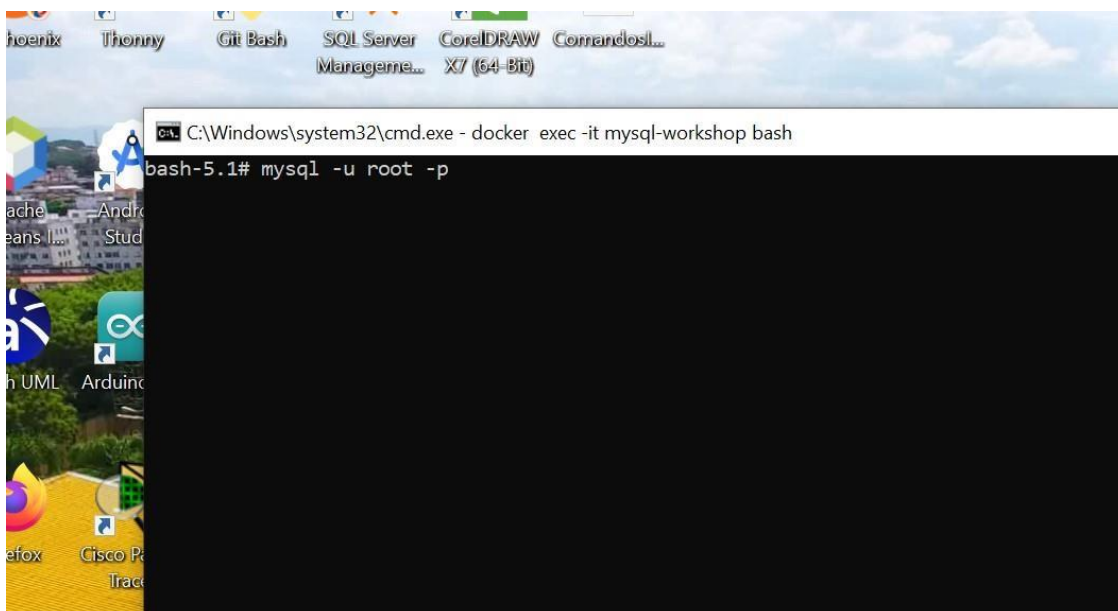
Inicie o mysql dentro do container com o código:

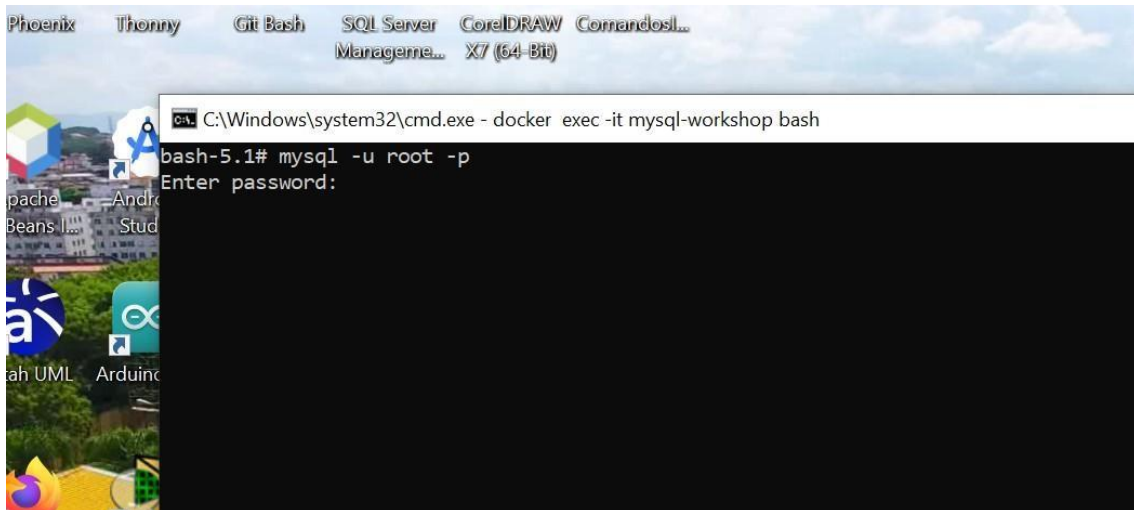
mysql -u root -p

-u: indica o usuário que esta utilizando o mysql;

-p: indica a senha (no caso é 1234).

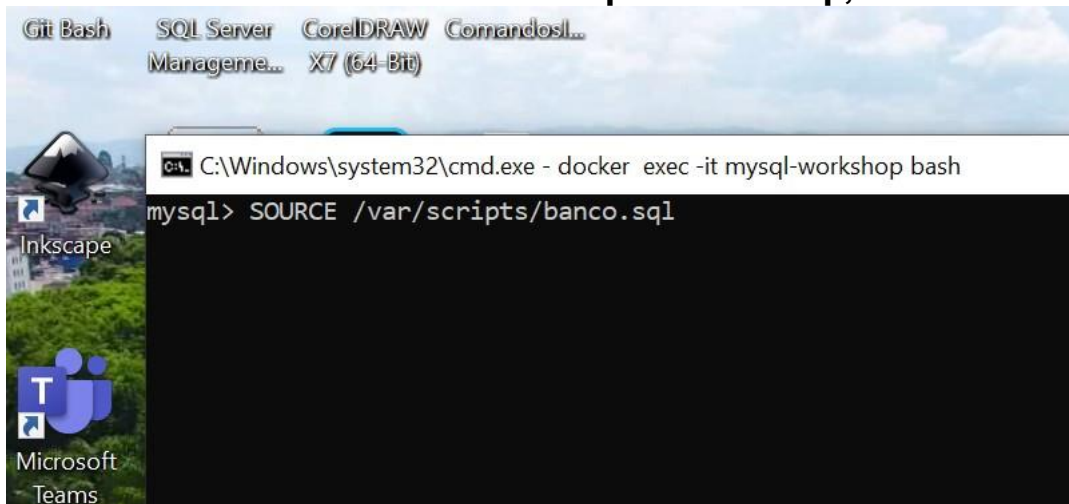
Após dar enter, o mysql ira pedir a senha





Execute o arquivo banco.sql com o código :

SOURCE /var/scripts/banco.sql;



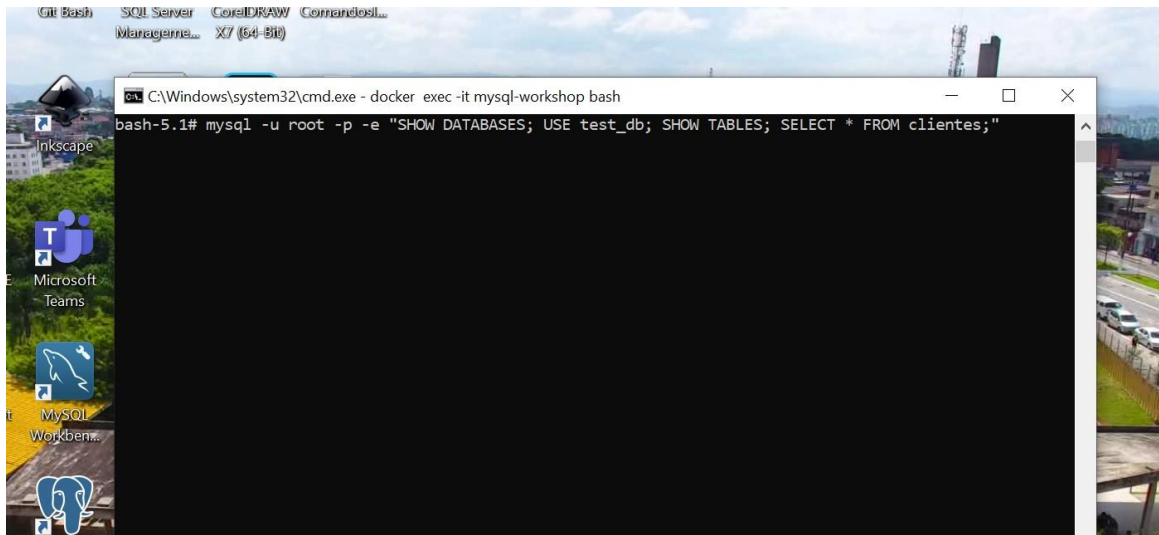
Após executar o arquivo, saia do mysql com o código:

Exit

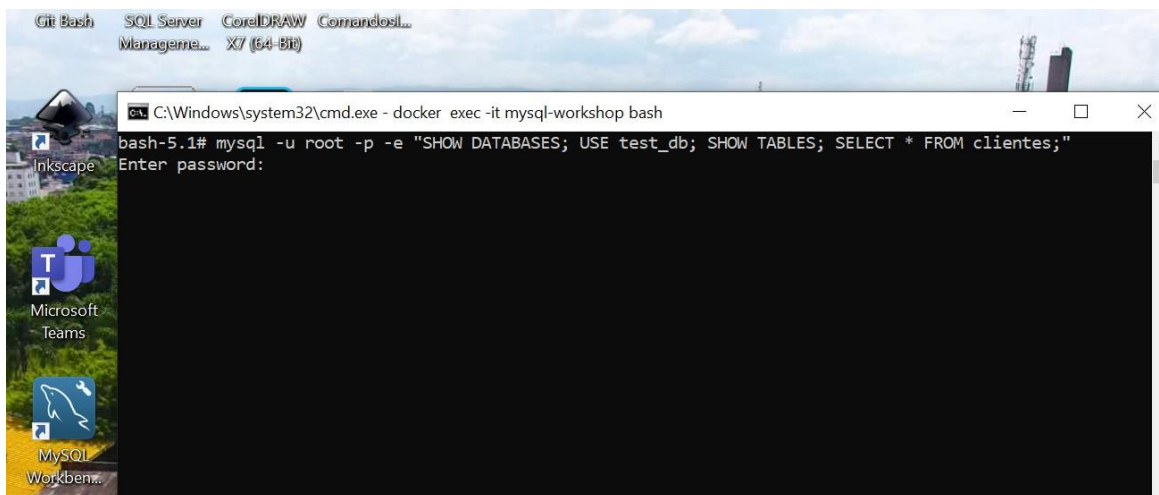
E teste para ver se o banco foi executado com sucesso com esse código:

mysql -u root -p -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM clientes;"

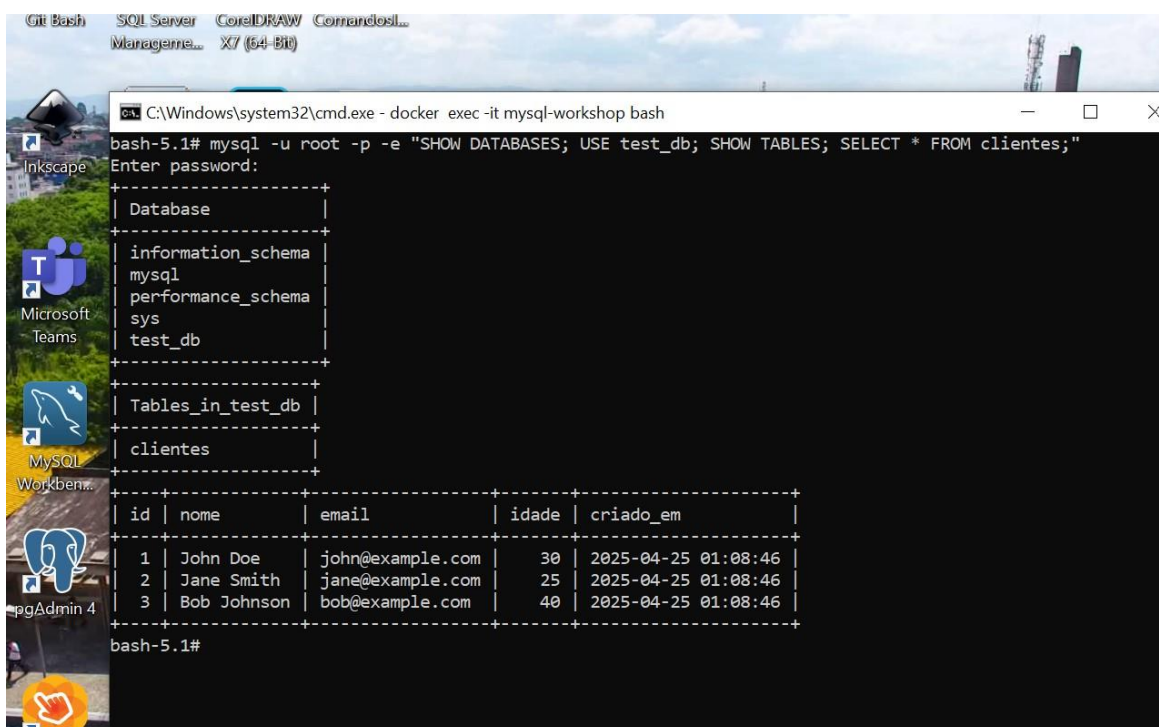
Esse deve ser o resultado:



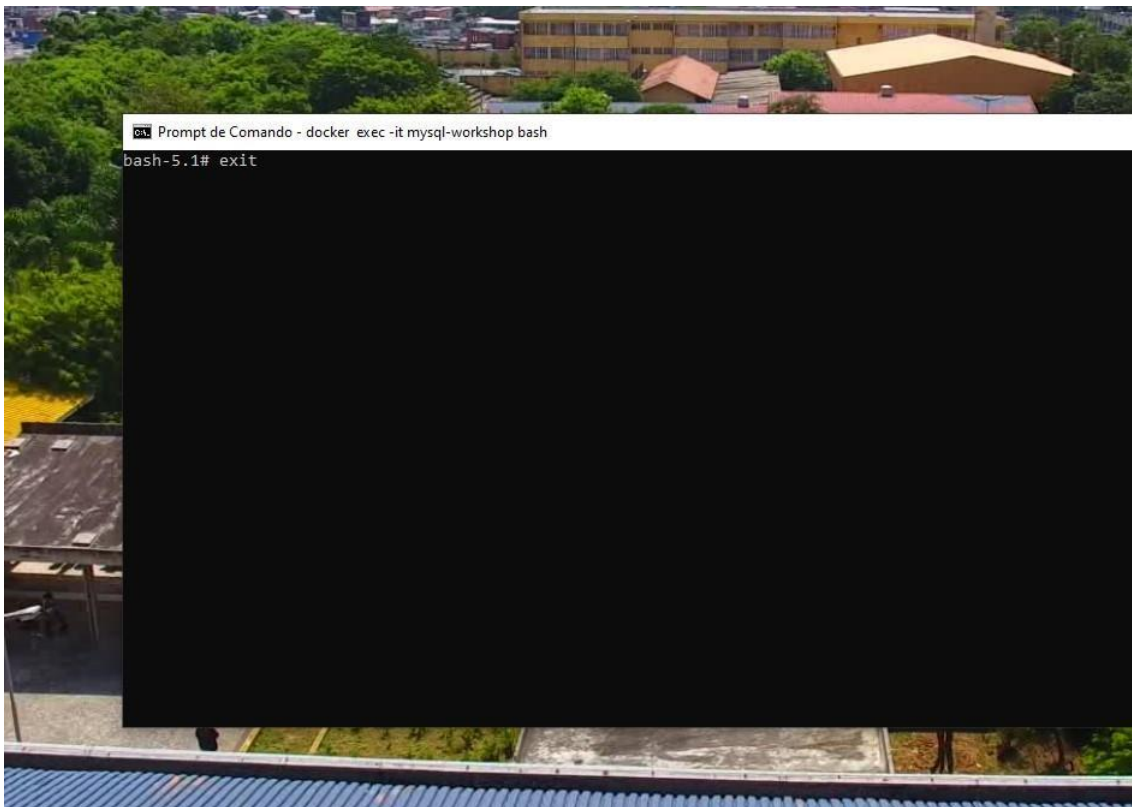
Coloque a senha: **1234**



O resultado esperado é esse:



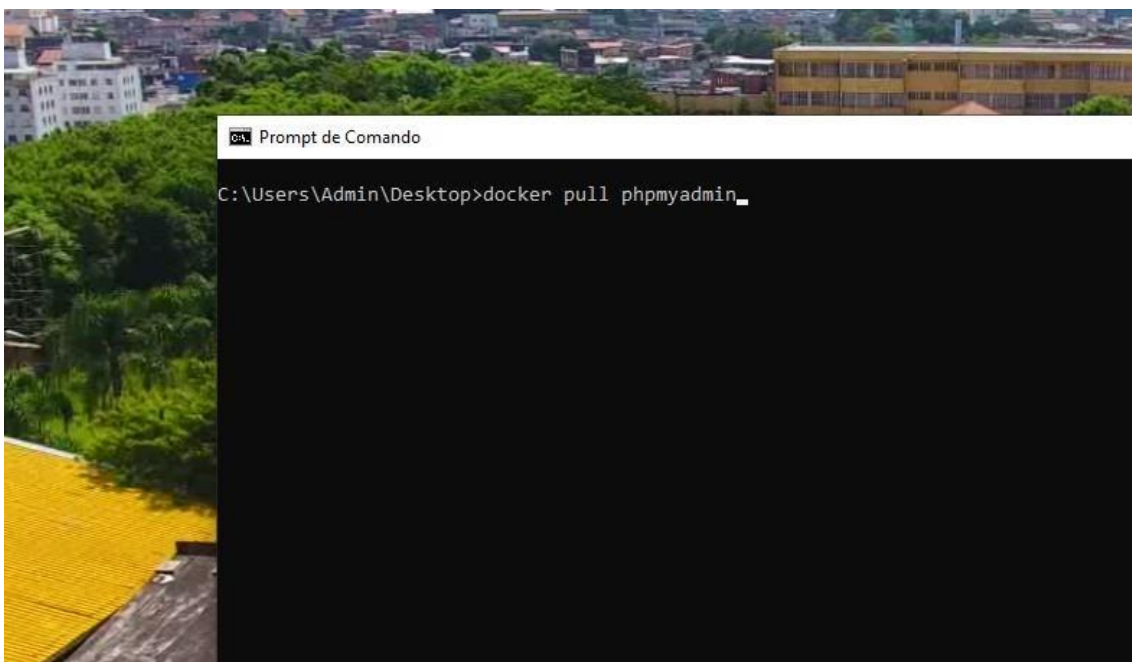
Digite **exit** no bash para sair do container



Agora digite o seguinte código:

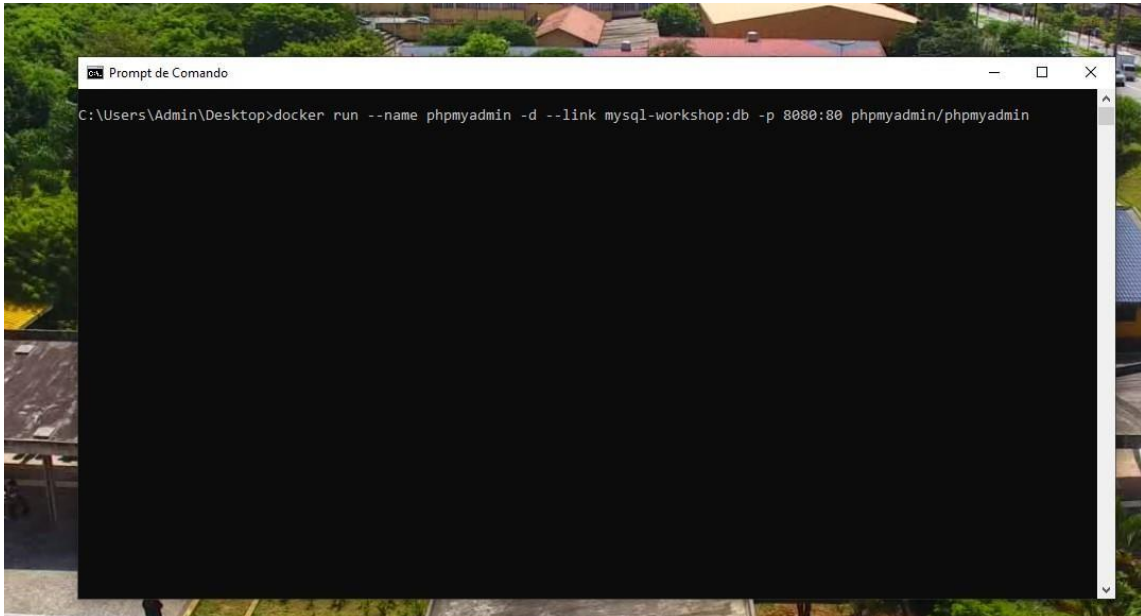
docker pull phpmyadmin

para baixar a imagem do Docker phpmyadmin

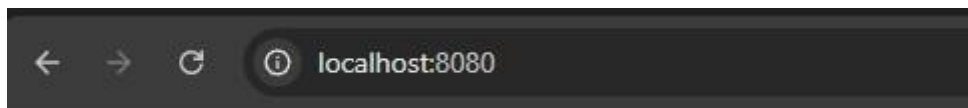


Crie o container phpmyadmin e link ele ao container mysql com o código:

docker run --name phpmyadmin -d --link mysql-container:db -p 8080:80 phpmyadmin/phpmyadmin



No navegador entre em: **Localhost:8080**



Faça o login no phpmyadmin com:

Usuário: root

Senha: 1234



Idioma (Language)

Português (Brasil) - Portuguese (Brazil) ▼

Entrar ⓘ

Usuário:

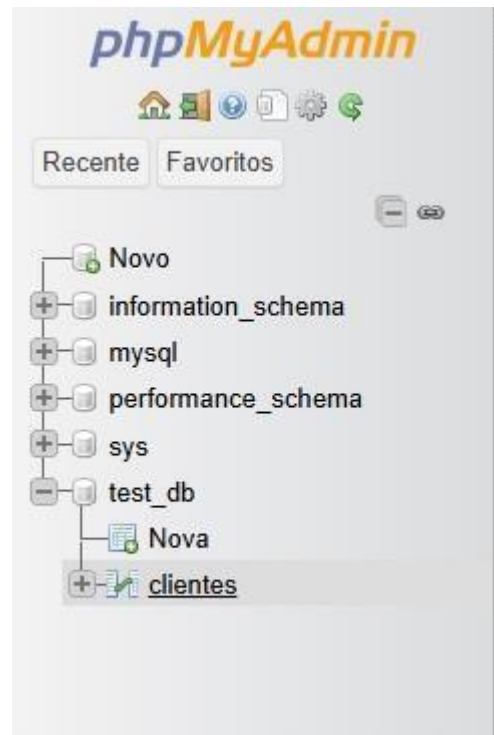
root

Senha:

....

Entrar

Vá para a tabela clientes

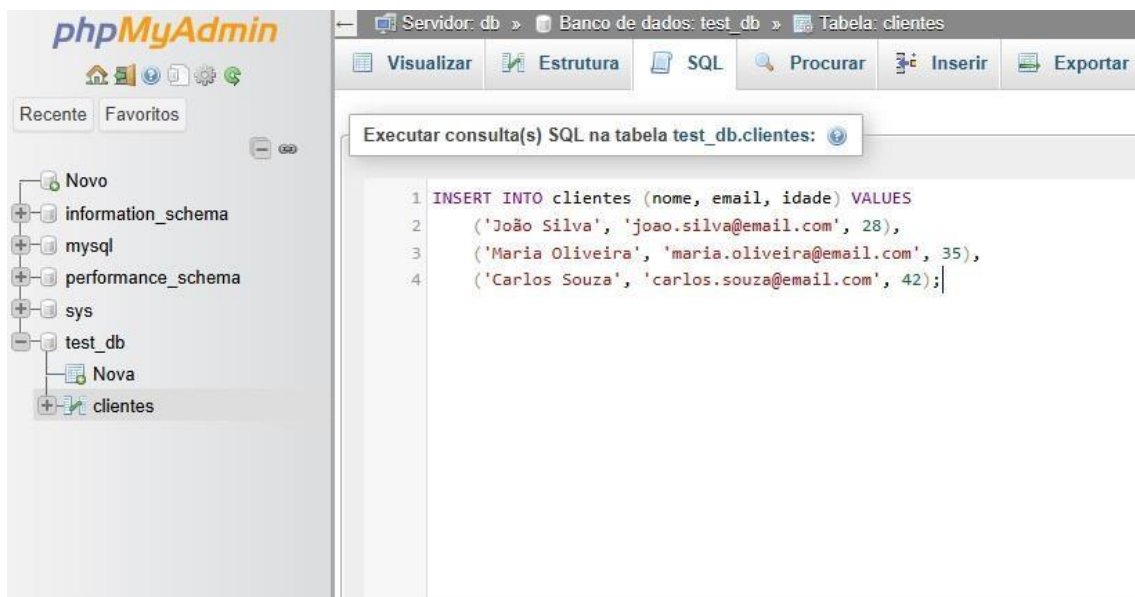


Va para a aba SQL

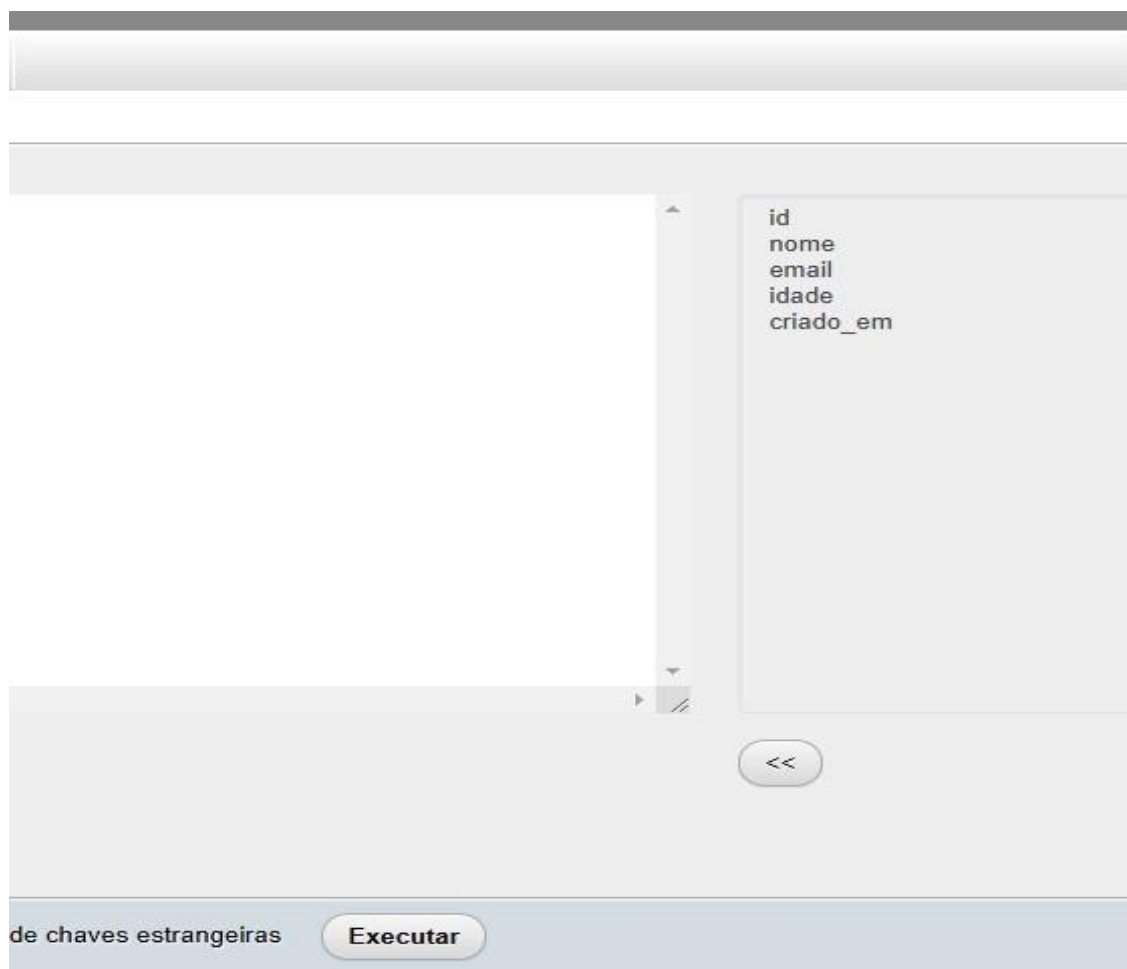


E digite adicione alguns clientes:

```
INSERT INTO clientes (nome, email, idade) VALUES  
('João Silva', 'joao.silva@email.com', 28),  
('Maria Oliveira', 'maria.oliveira@email.com', 35),  
('Carlos Souza', 'carlos.souza@email.com', 42);
```

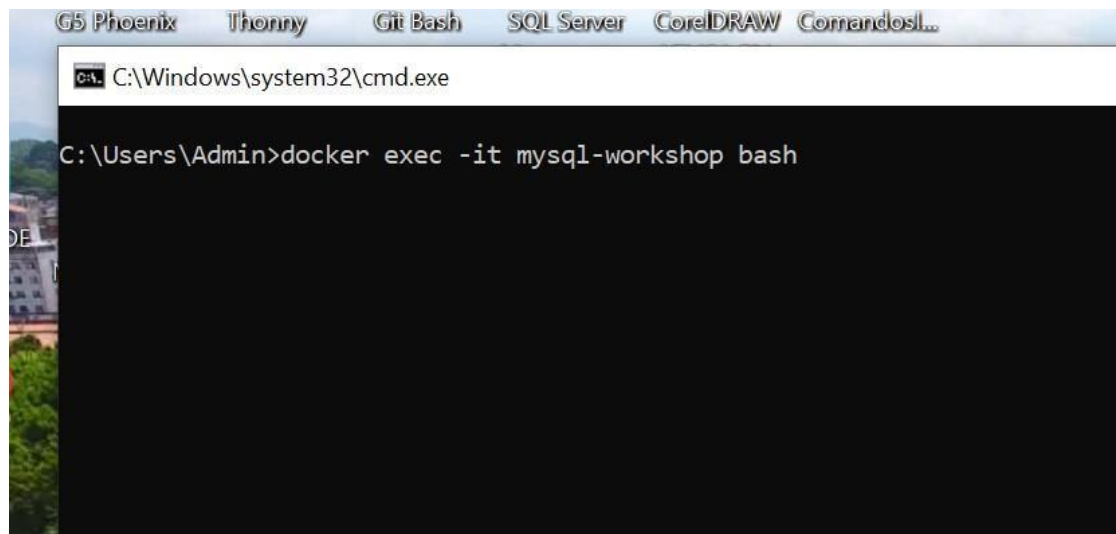


Execute o código

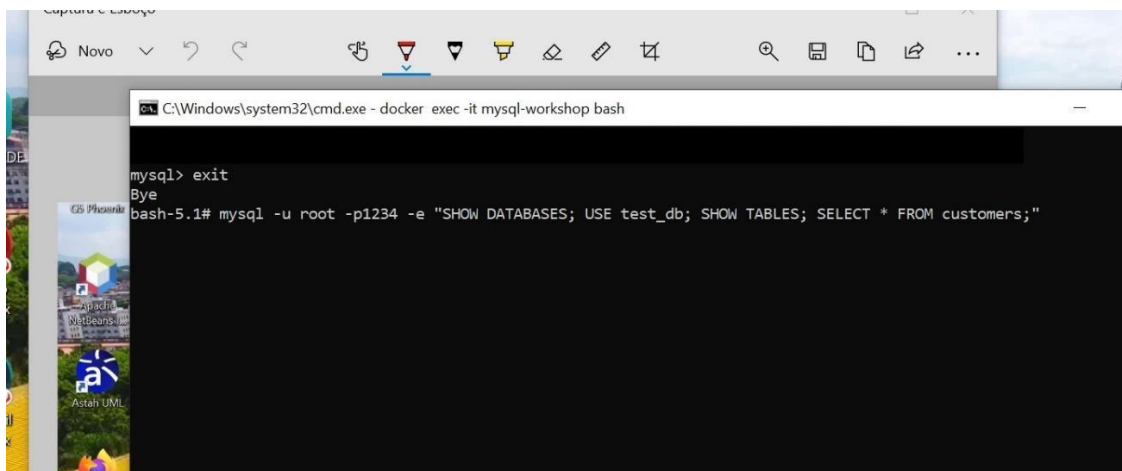


Verifique no container se os inserts foram colocados com o código:

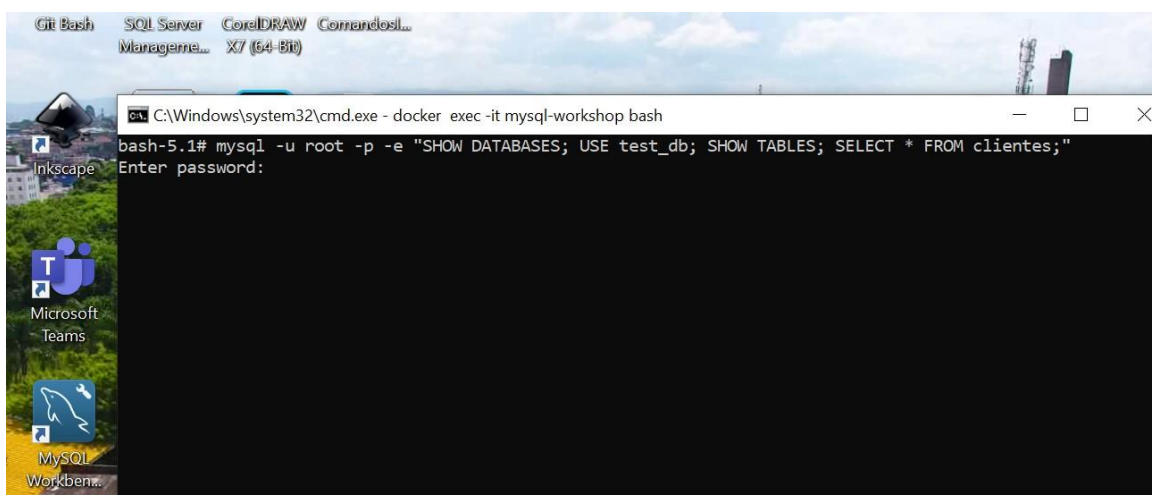
docker exec -it mysql-workshop bash

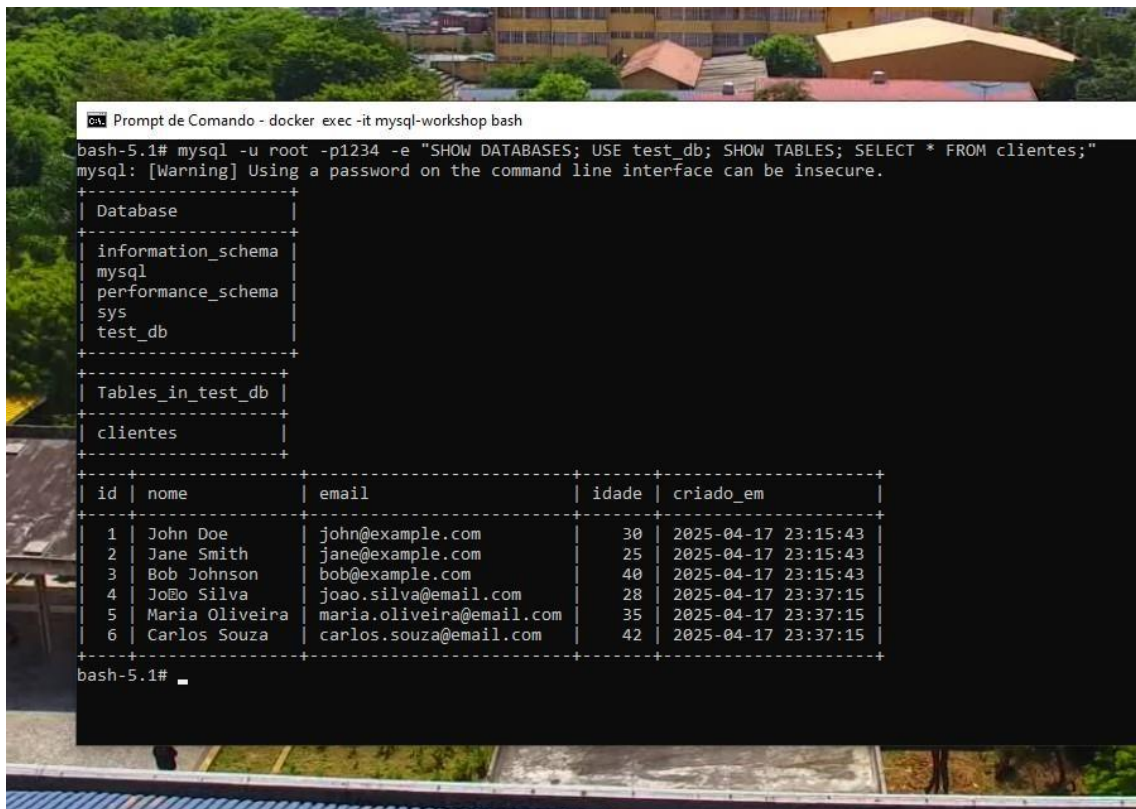


E verifique se os inserts foram feitos com sucesso com o código:
mysql -u root -p -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM clientes;"



Coloque a senha: **1234**

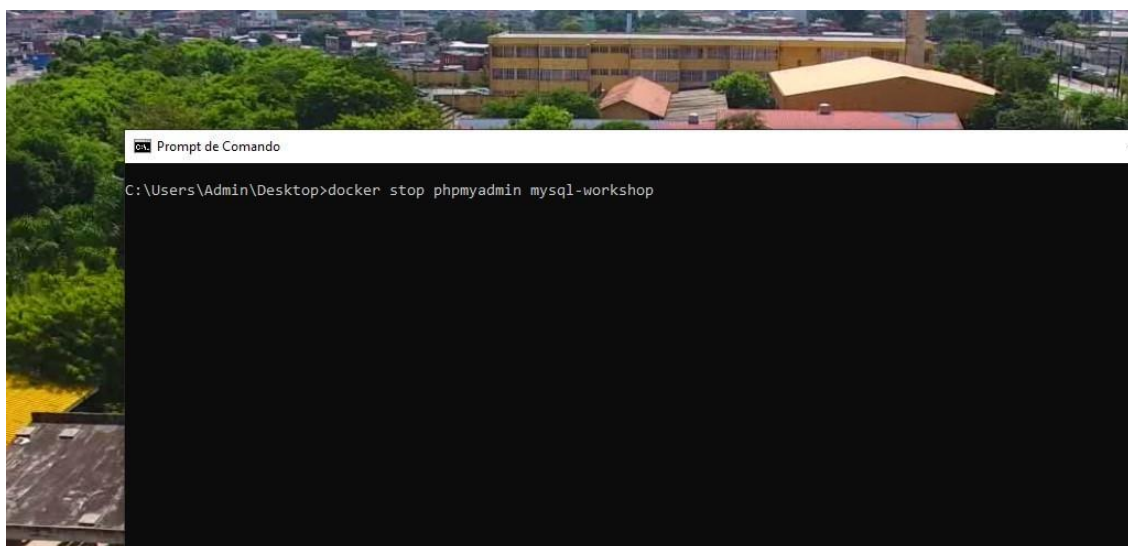


A terminal window titled 'Prompt de Comando - docker exec -it mysql-workshop bash' is overlaid on a background image of a residential area with houses and trees. The terminal shows a MySQL session where the user runs 'SHOW DATABASES;', 'USE test_db;', 'SHOW TABLES;', and 'SELECT * FROM clientes;'. The output shows the 'test_db' database with a 'clientes' table. The table has columns 'id', 'nome', 'email', 'idade', and 'criado_em', and contains 6 rows of data.

```
bash-5.1# mysql -u root -p1234 -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM clientes;"
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_db |
+-----+
+-----+
| Tables_in_test_db |
+-----+
| clientes |
+-----+
+-----+
| id | nome | email | idade | criado_em |
+-----+
| 1 | John Doe | john@example.com | 30 | 2025-04-17 23:15:43 |
| 2 | Jane Smith | jane@example.com | 25 | 2025-04-17 23:15:43 |
| 3 | Bob Johnson | bob@example.com | 40 | 2025-04-17 23:15:43 |
| 4 | JoBo Silva | joao.silva@email.com | 28 | 2025-04-17 23:37:15 |
| 5 | Maria Oliveira | maria.oliveira@email.com | 35 | 2025-04-17 23:37:15 |
| 6 | Carlos Souza | carlos.souza@email.com | 42 | 2025-04-17 23:37:15 |
+-----+
bash-5.1#
```

Pare o container com o código:

docker stop phpmyadmin mysql-container

A terminal window titled 'Prompt de Comando' is overlaid on the same background image. The terminal shows the command 'C:\Users\Admin\Desktop>docker stop phpmyadmin mysql-workshop' being entered.

```
C:\Users\Admin\Desktop>docker stop phpmyadmin mysql-workshop
```

E delete os containers com o código:

docker stop phpmyadmin mysql-container

