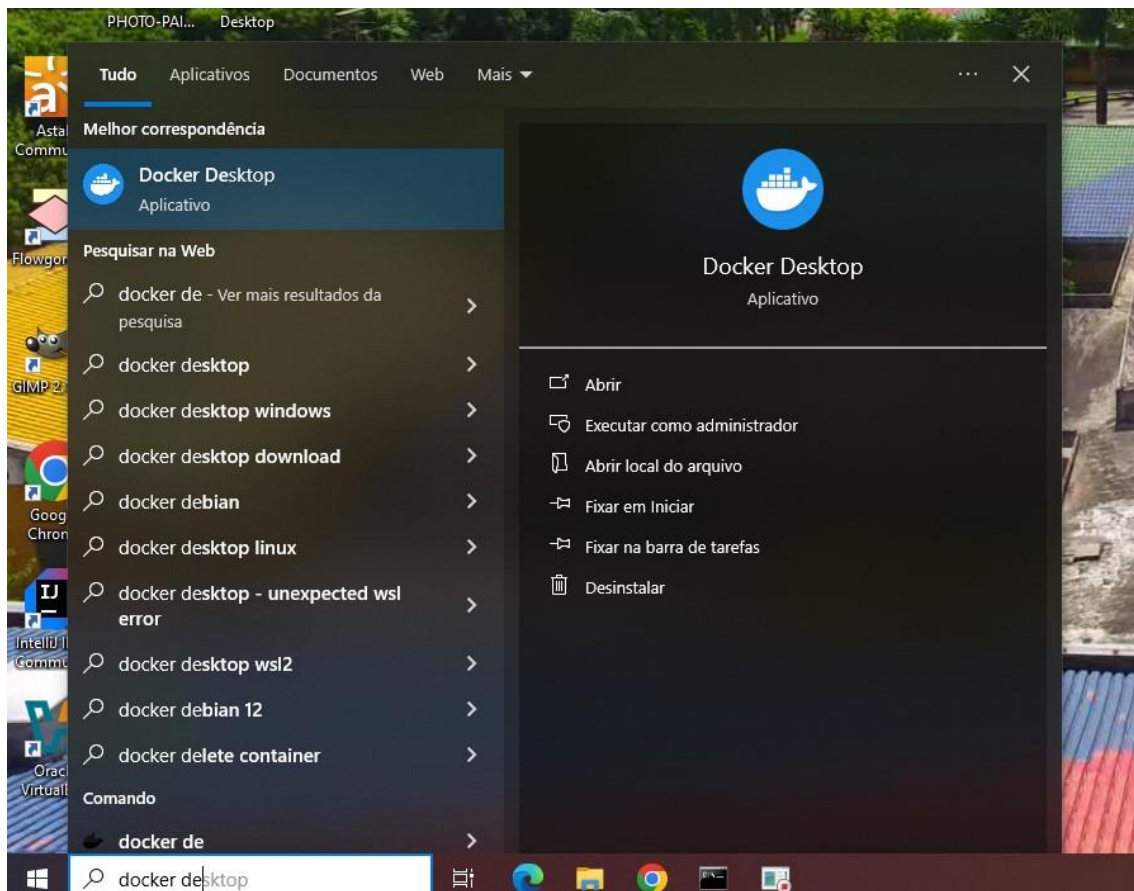
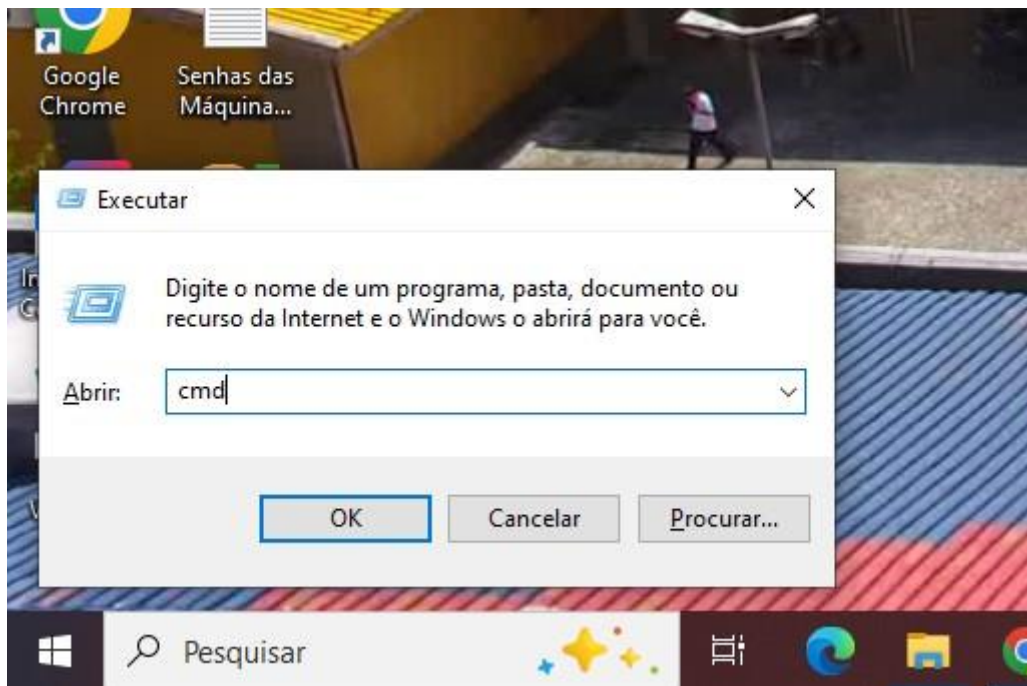


Abram o Docker Desktop, no menu iniciar

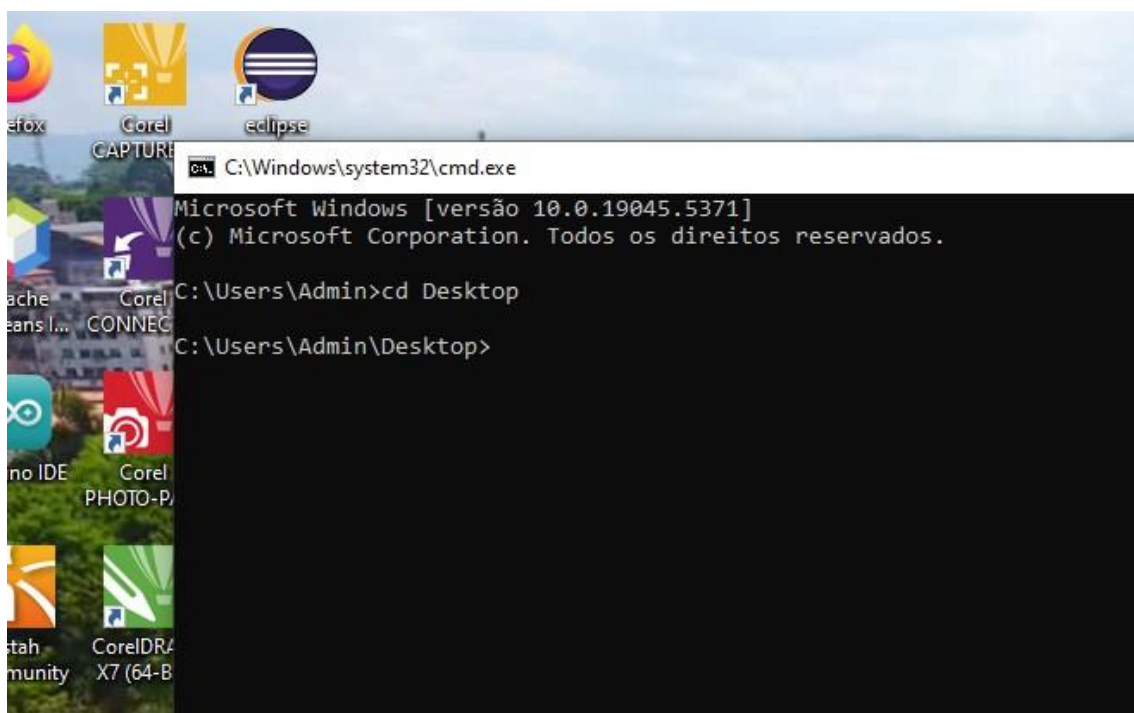


Em seguida abram o terminal (CMD)



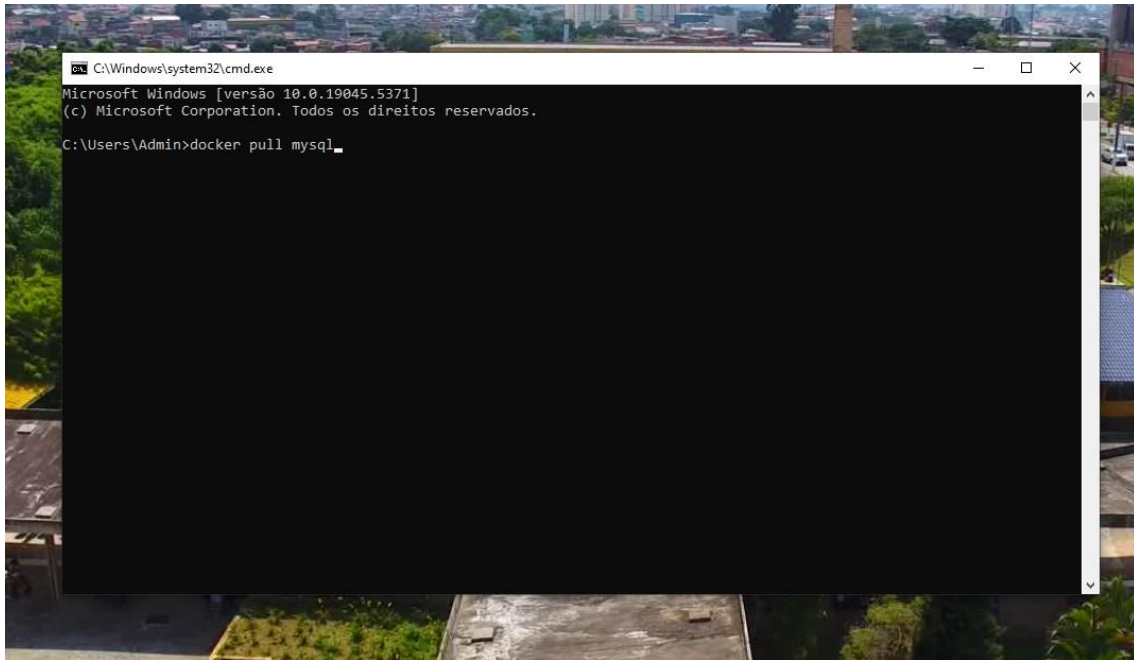


Vá para a área de trabalho (Desktop)



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  
C:\Users\Admin\Desktop\Link:/var/lib/mysql -p 3306:3306 -d  
mysql:latest
```



Explicação do código:

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

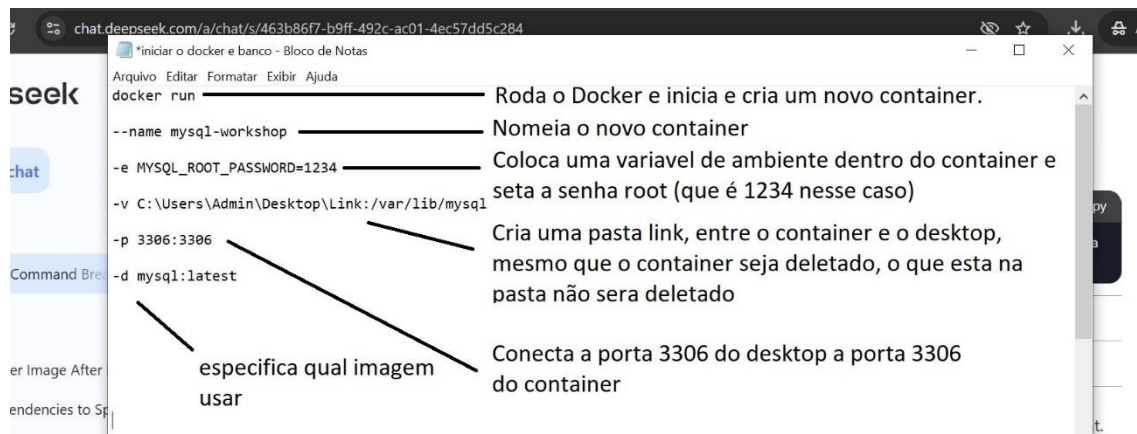
**--name mysql-workshop:** Nomeia o novo container;

**-e MYSQL\_ROOT\_PASSWORD=1234:** Coloca uma variável de ambiente dentro do container e seta a senha root para 1234;

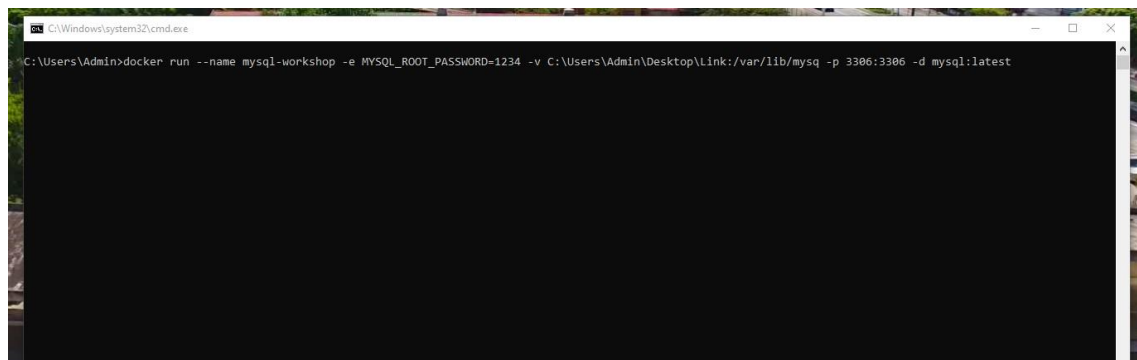
**-v C:\Users\Admin\Desktop\Link:/var/lib/mysql:** Cria uma pasta volume (link) entre o desktop e o container, ambos vão compartilhar o conteúdo da pasta;

**-p 3306:3006:** Conecta a porta 3306 do desktop com a porta 3306 do container;

**-d mysql:latest:** Especifica qual imagem usar;



Executando no terminal

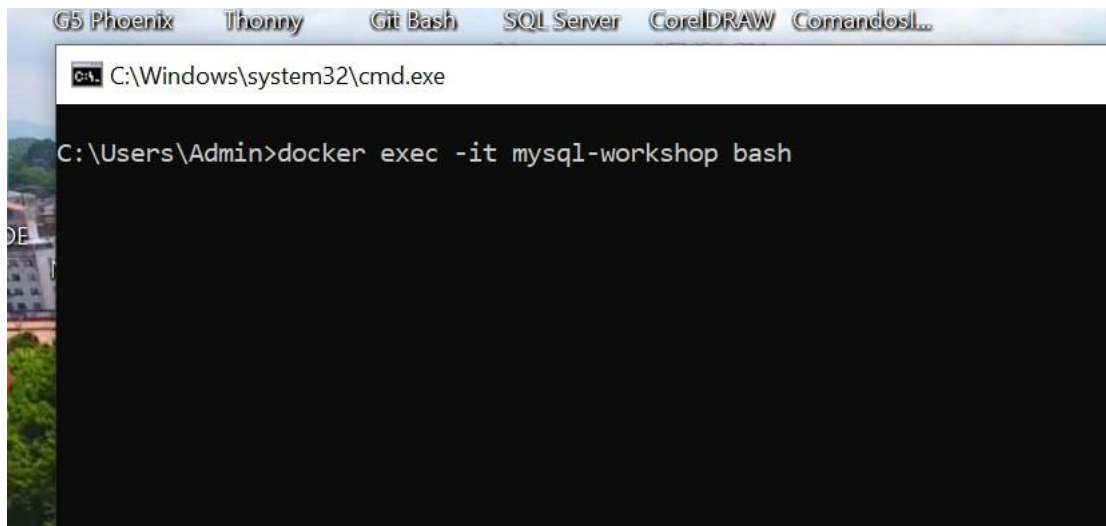


Agora entramos no bash do container com o código:

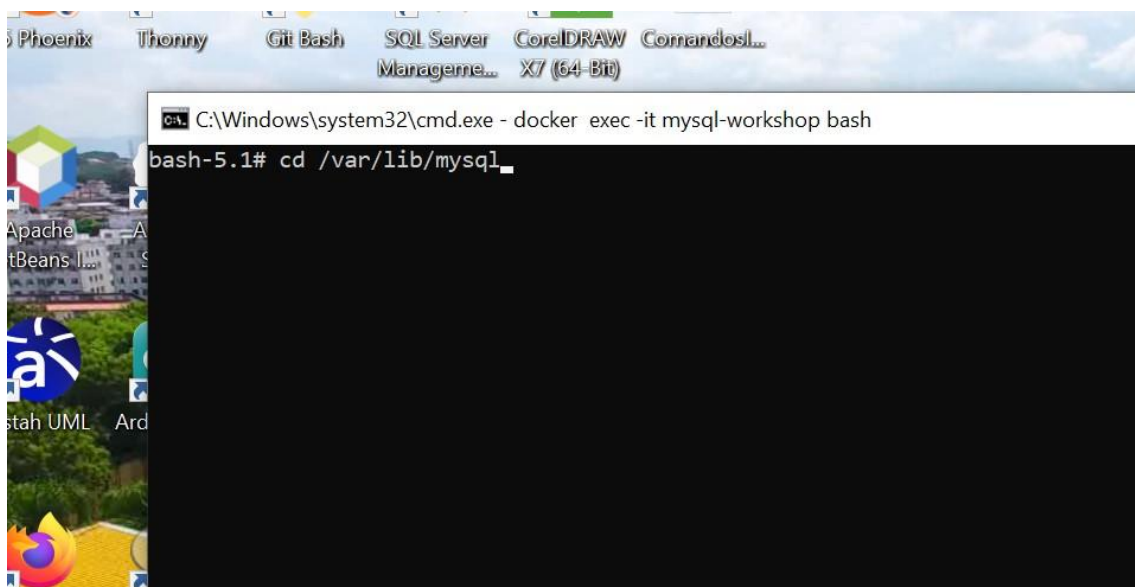
**docker exec -it mysql-workshop bash**

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

**cd /var/lib/mysql**



A screenshot of a Windows command prompt window. The title bar shows several open applications: G5 Phoenix, Thonny, Git Bash, SQL Server, CorelDRAW, and Comandosl... The command prompt shows the path C:\Windows\system32\cmd.exe. The user has entered the command `C:\Users\Admin>docker exec -it mysql-workshop bash`.



A screenshot of a Docker container terminal window. The title bar shows the same applications as the first screenshot. The terminal title is `C:\Windows\system32\cmd.exe - docker exec -it mysql-workshop bash`. The prompt is `bash-5.1#` and the user has entered `cd /var/lib/mysql_`. The terminal background is black.

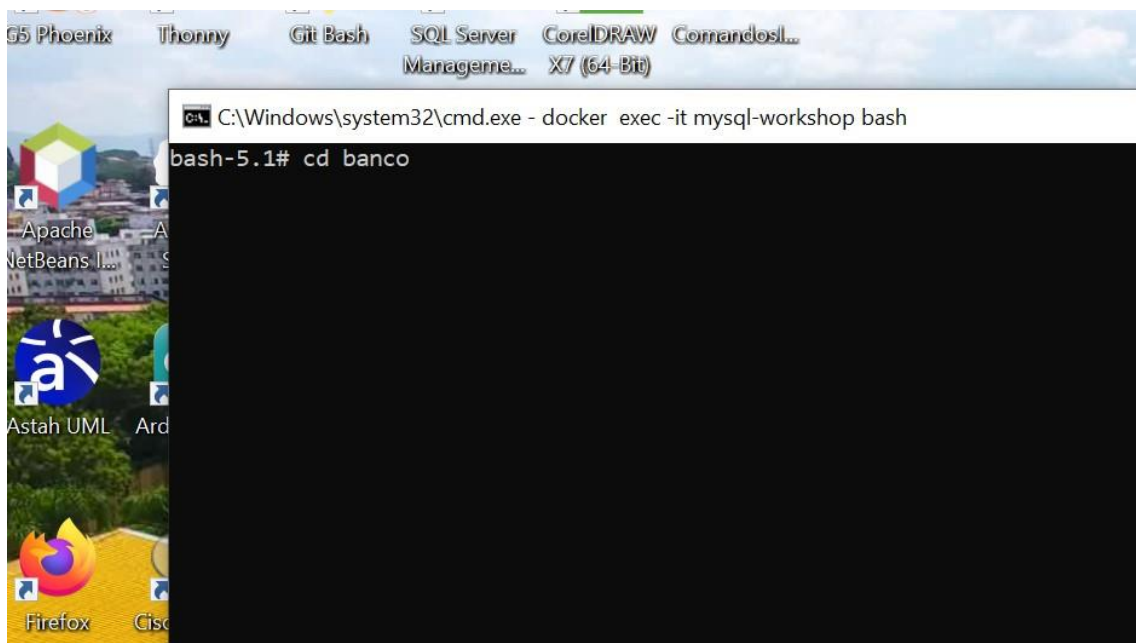
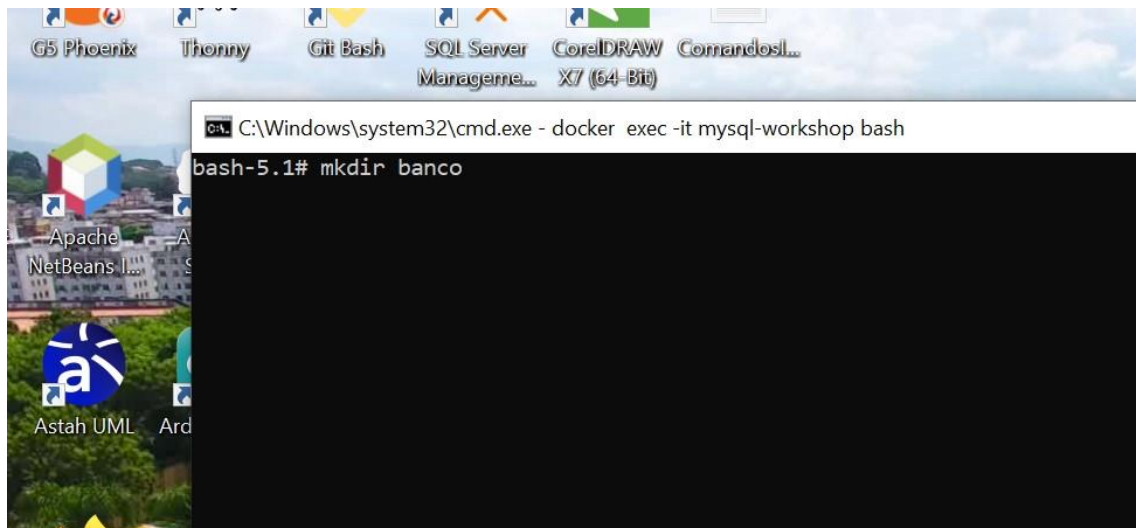
Crie uma pasta chamada “banco” com o código:

**mkdir banco**

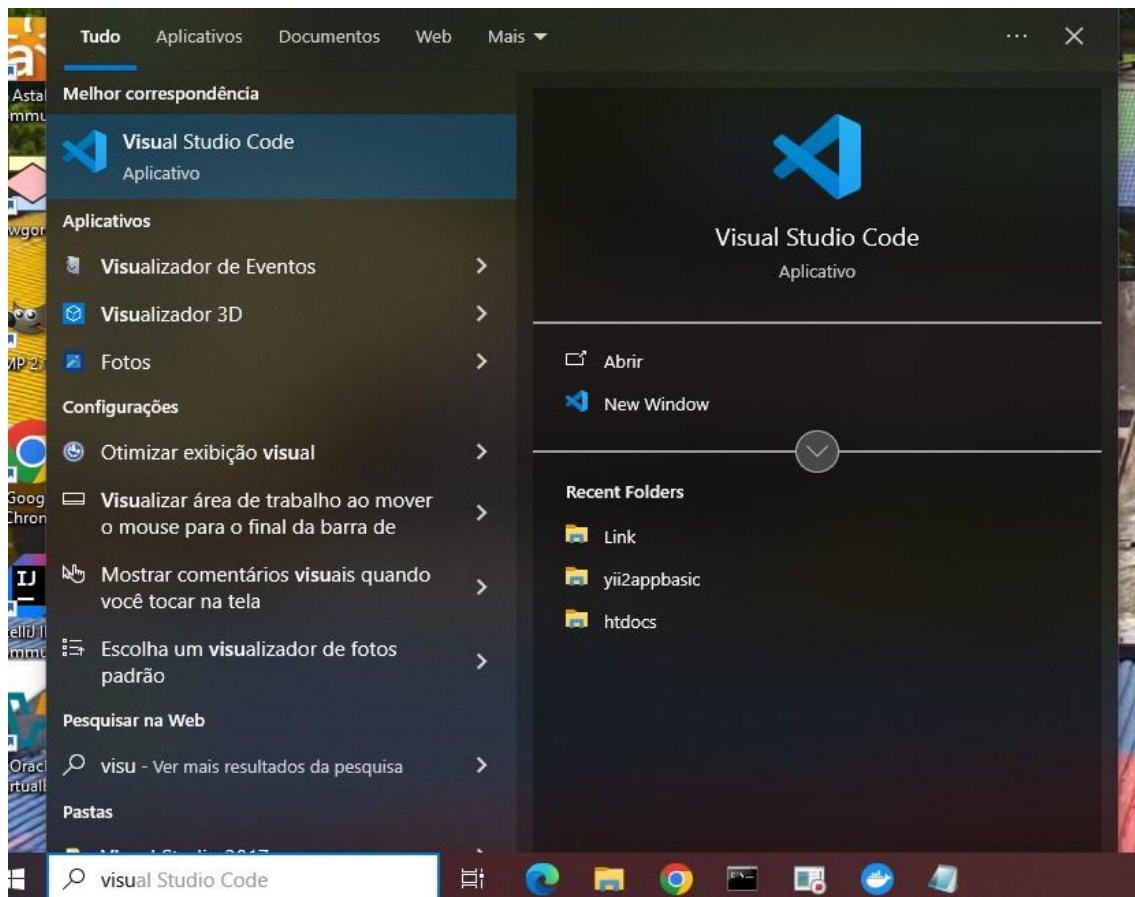
e entre na pasta com o código:

**cd banco**

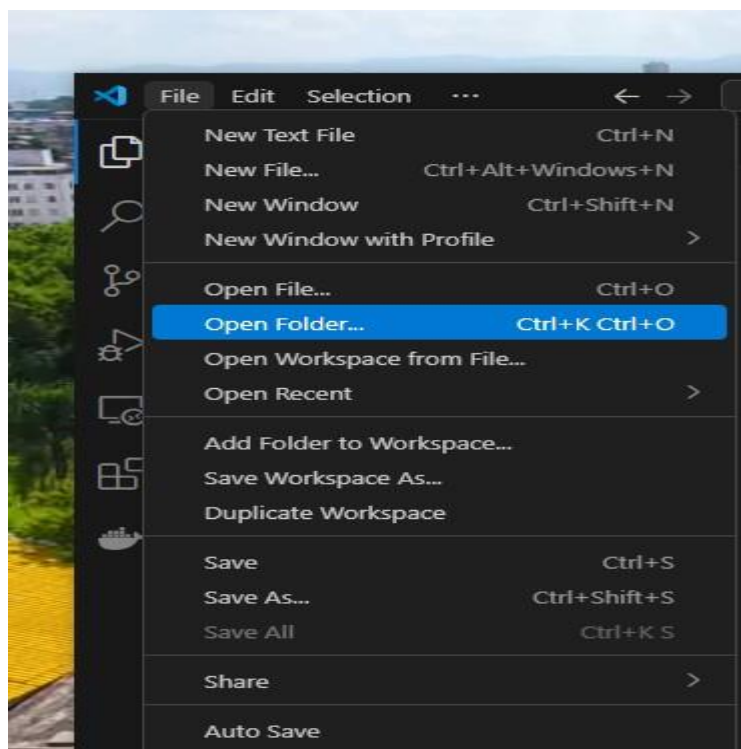


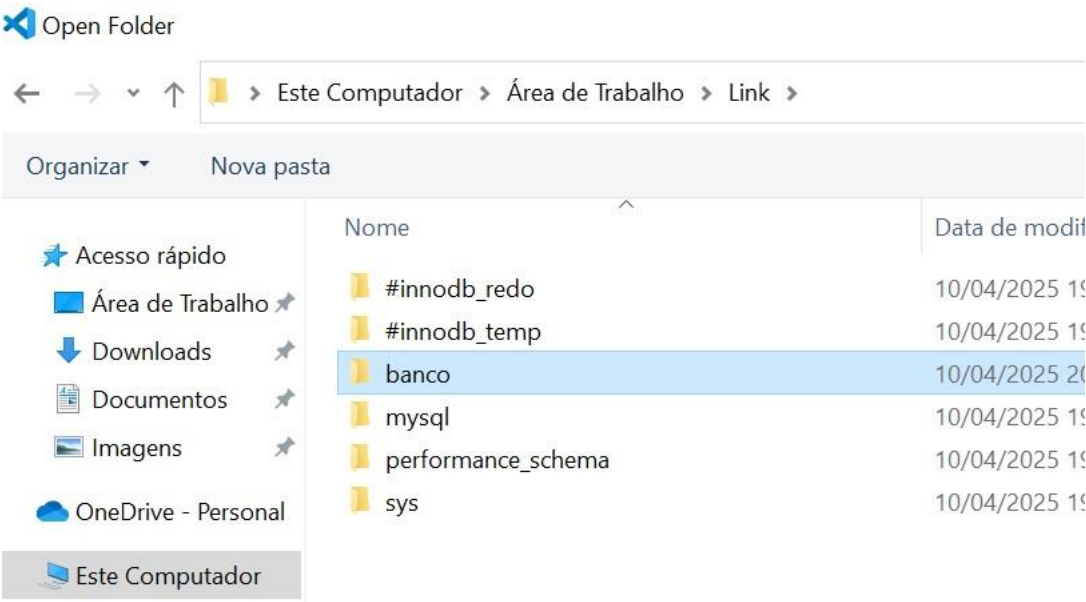
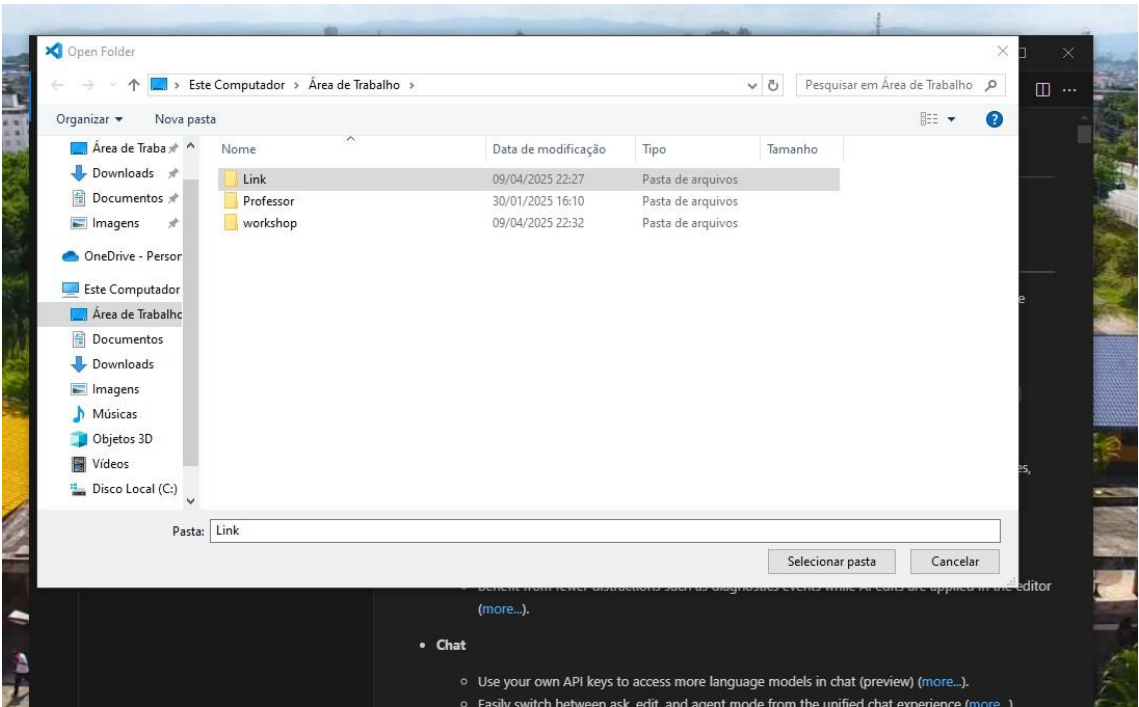


De volta no desktop, abra o visual studio code

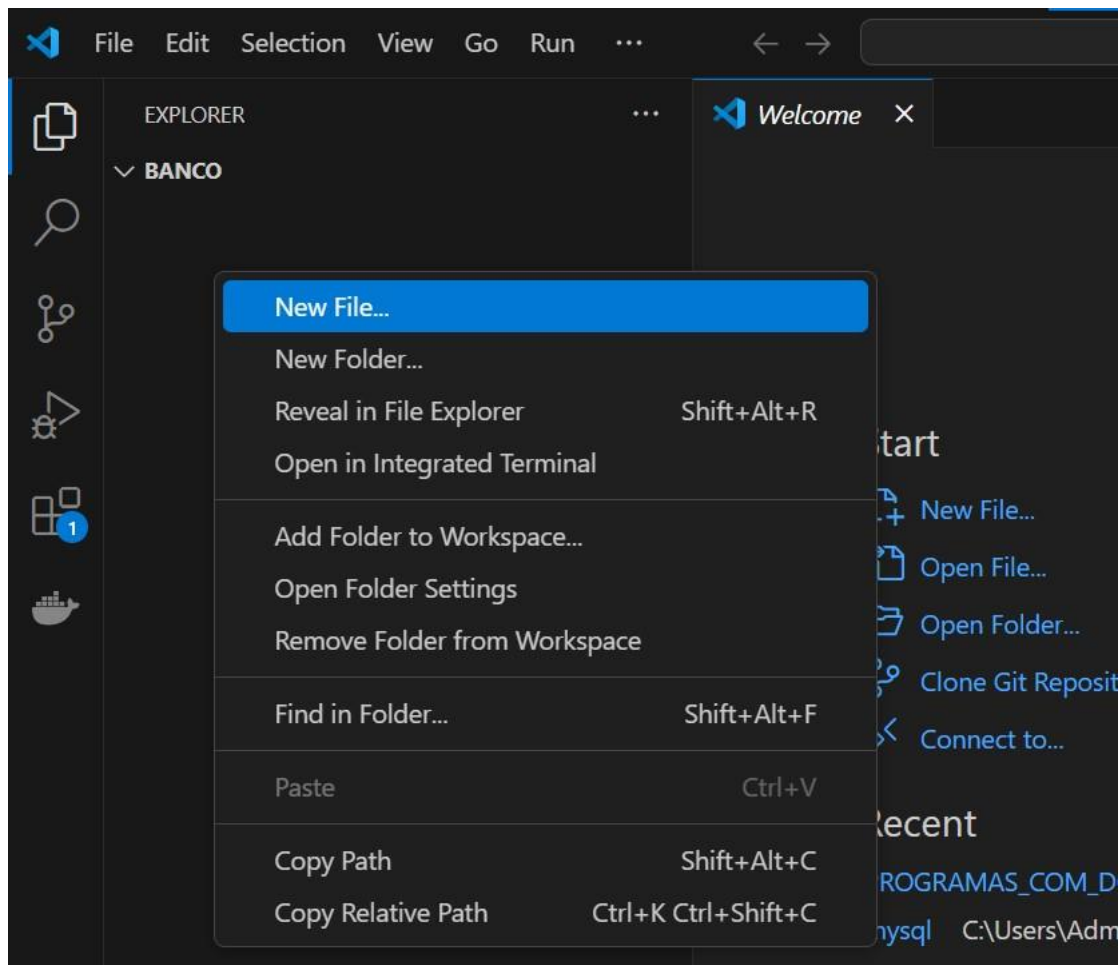


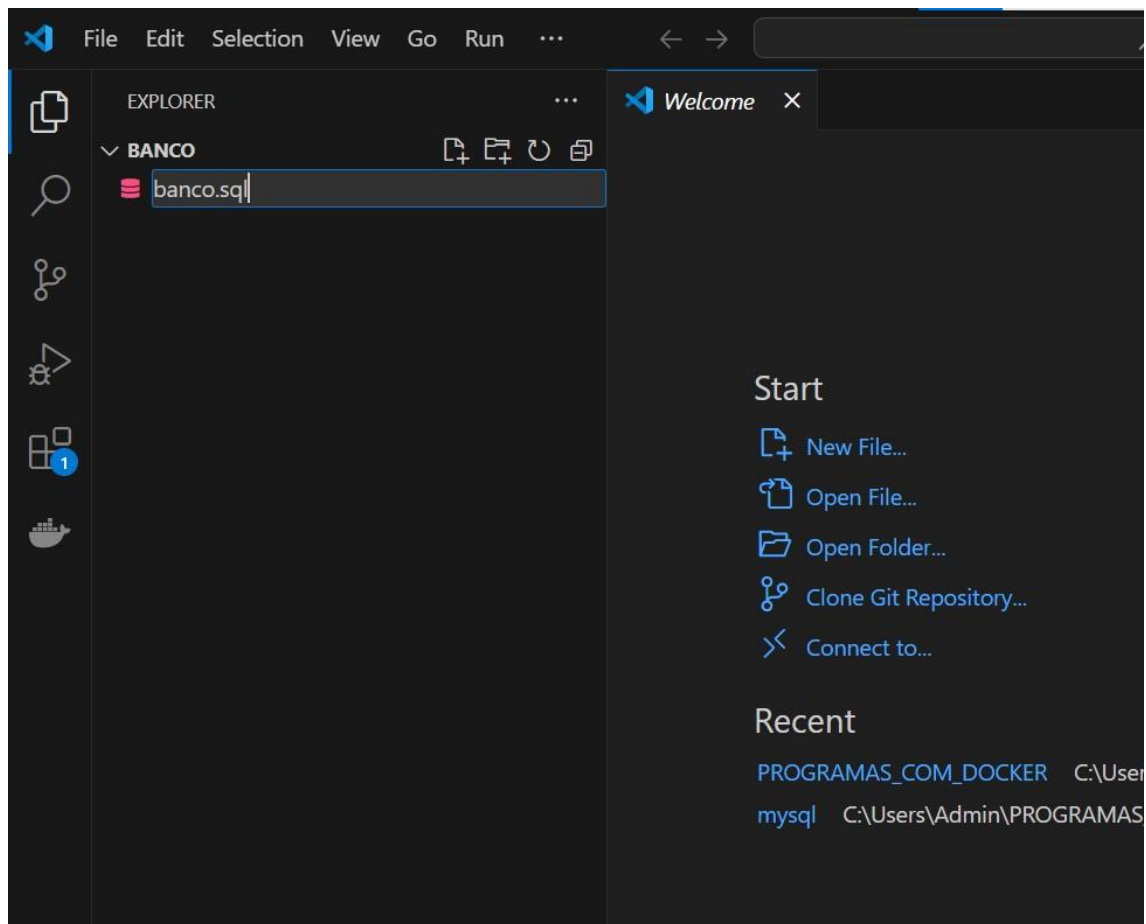
E abrir a pasta link/banco na área de trabalho e criar um novo arquivo “banco.sql”;











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,
```

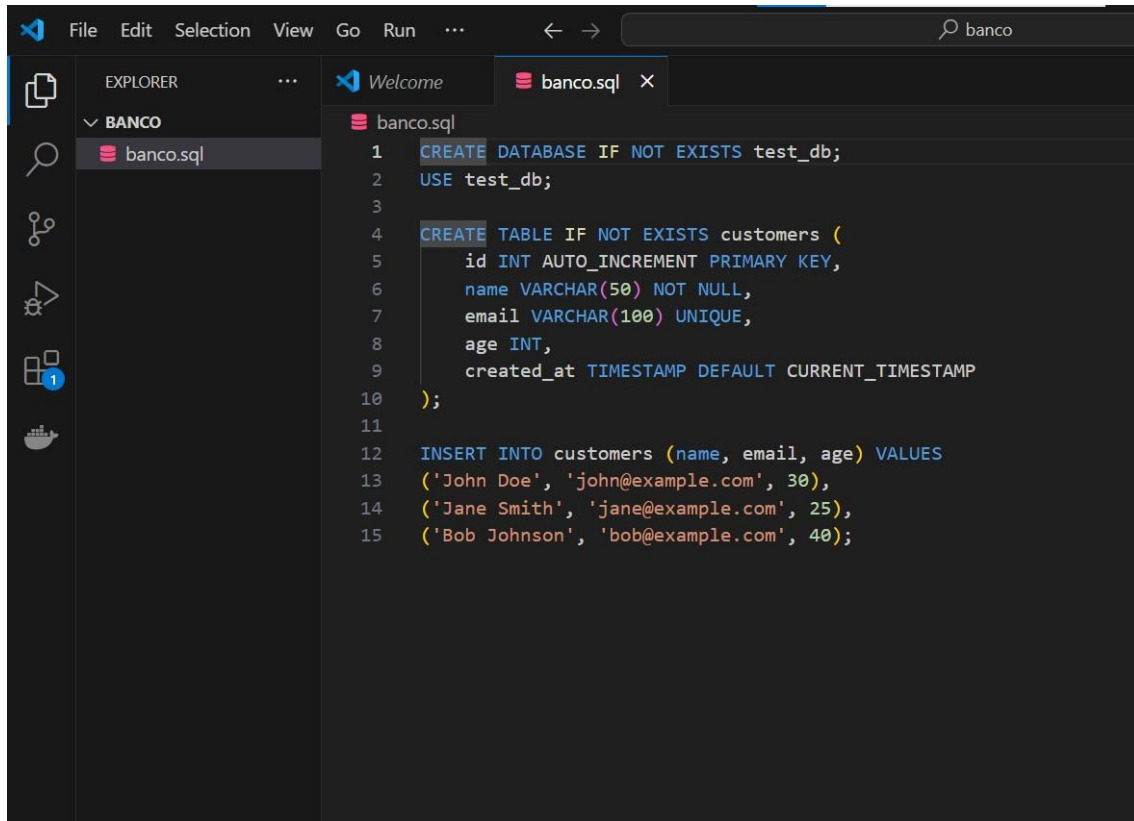
```
    idade INT,
```

```
    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);
```

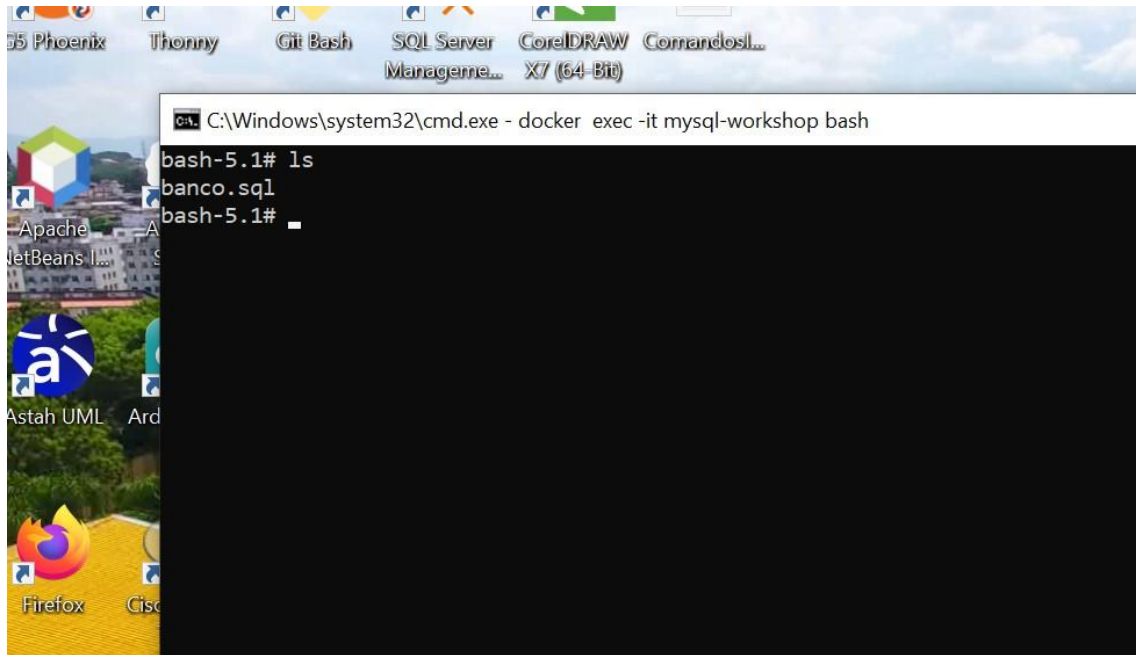


The screenshot shows a code editor with a dark theme. The Explorer panel on the left shows a folder named 'BANCO' containing a file 'banco.sql'. The main editor area displays the contents of 'banco.sql', which is a SQL script. The script starts with creating a database 'test\_db' and using it. Then, it creates a table 'customers' with columns 'id' (auto-increment primary key), 'name' (varchar 50, not null), 'email' (varchar 100, unique), 'age' (int), and 'created\_at' (timestamp default current\_timestamp). Finally, it inserts three rows of customer data.

```
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**



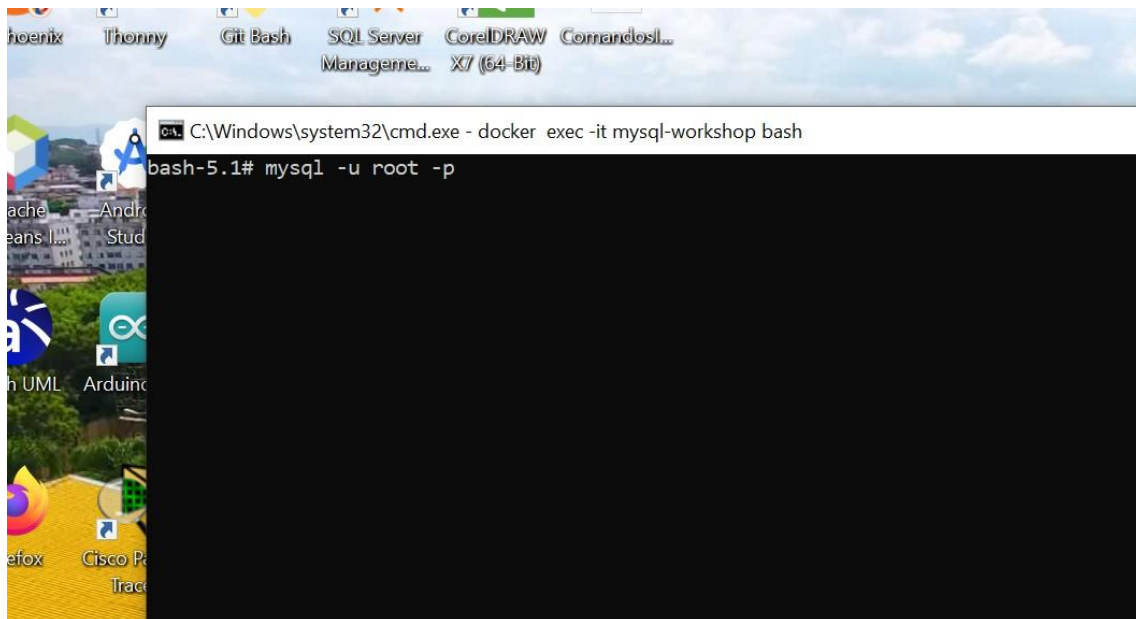
Inicie o mysql dentro do container com o código:

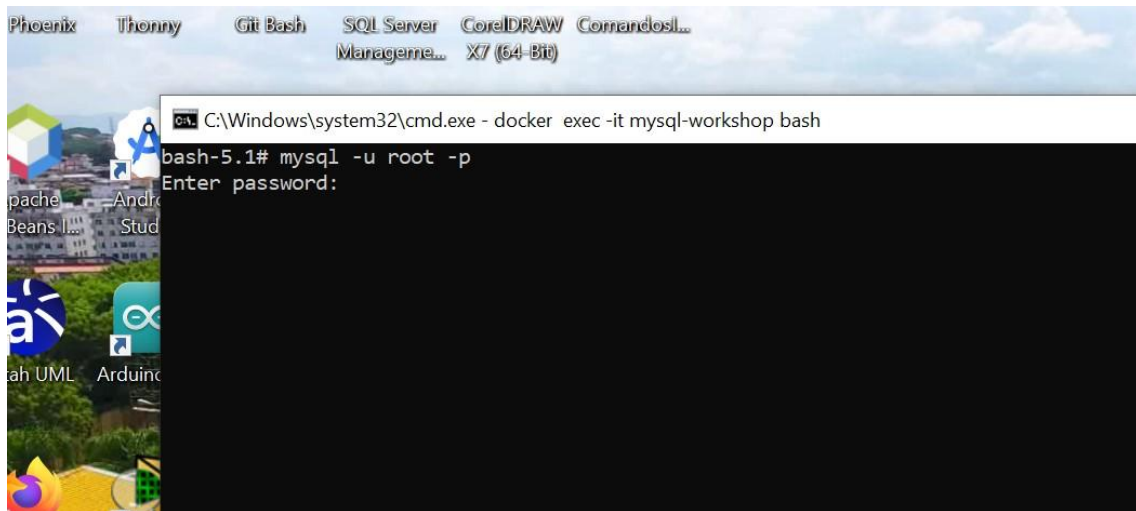
**mysql -u root -p**

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

**-p:** indica a senha (no caso é 1234).

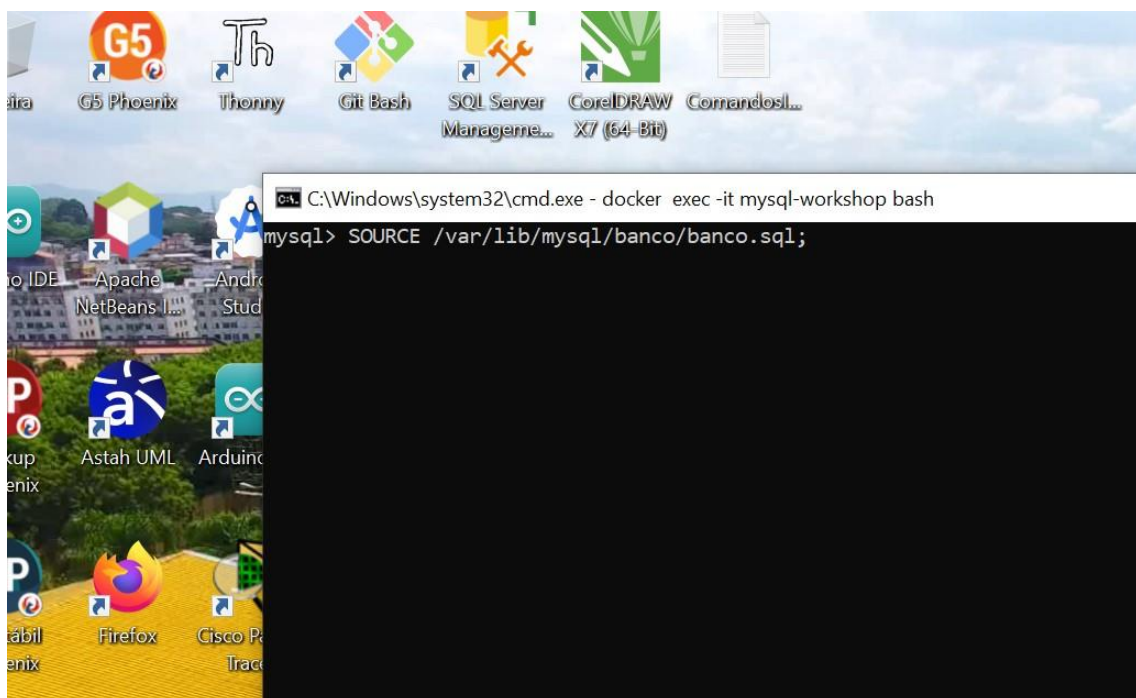
Após dar enter, o mysql ira pedir a senha





Execute o arquivo banco.sql com o código :

**SOURCE /var/lib/mysql/banco/banco.sql;**



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

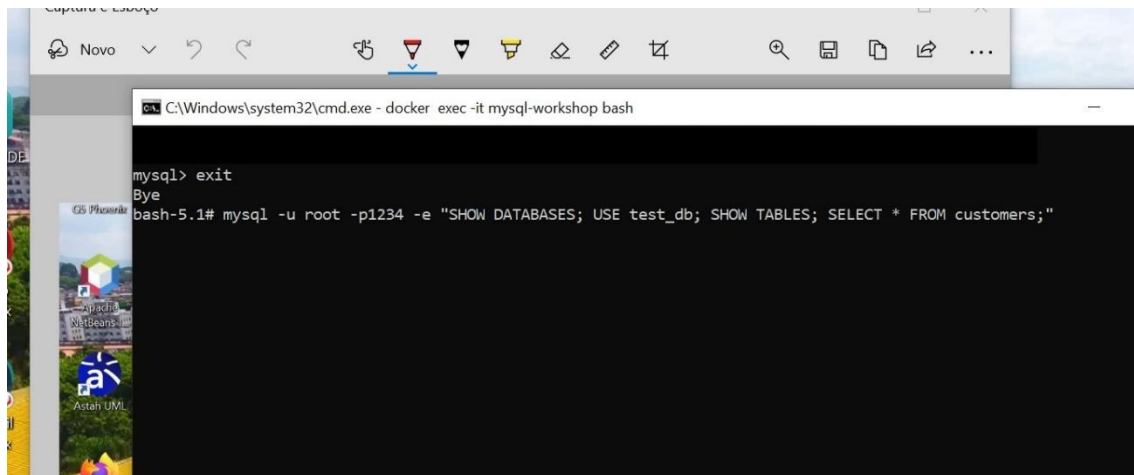
**Exit**

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

**mysql -u root -p1234 -e "SHOW DATABASES; USE test\_db; SHOW TABLES; SELECT \* FROM clientes;"**

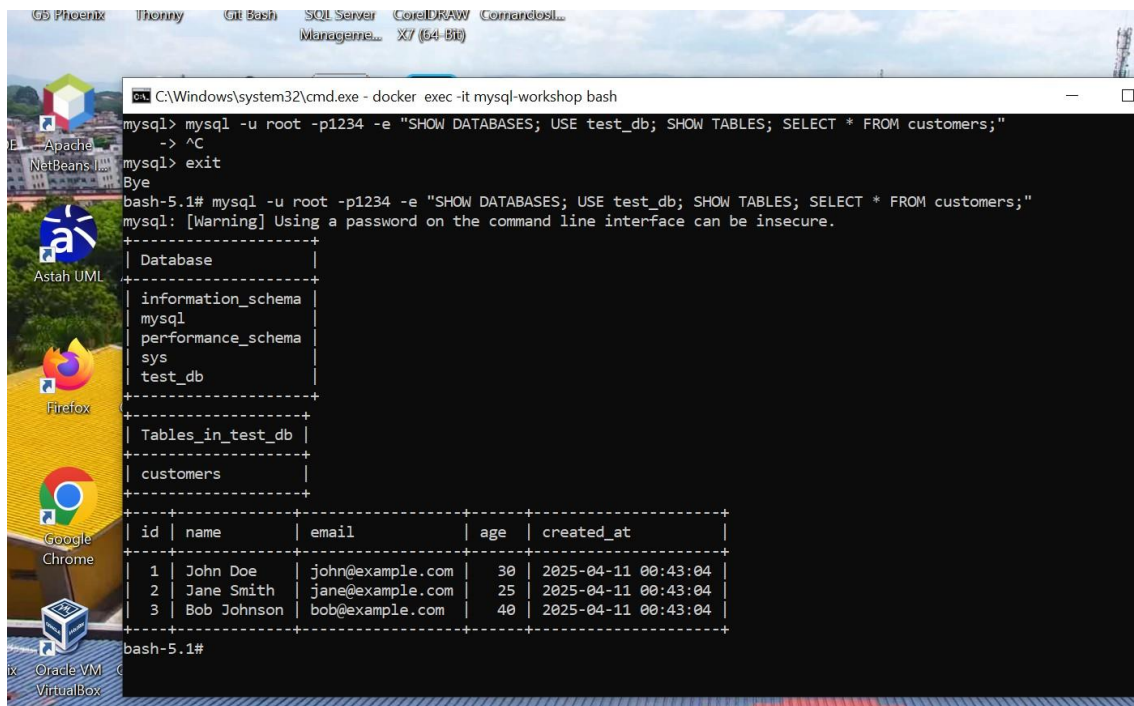
Esse deve ser o resultado:





A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - docker exec -it mysql-workshop bash". The terminal shows the following sequence of commands and output:

```
mysql> exit
Bye
bash-5.1# mysql -u root -p1234 -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM customers;"
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - docker exec -it mysql-workshop bash". The terminal shows the following sequence of commands and output:

```
mysql> mysql -u root -p1234 -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM customers;"
-> ^C
mysql> exit
Bye
bash-5.1# mysql -u root -p1234 -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM customers;"
mysql: [Warning] Using a password on the command line interface can be insecure.
```

The output of the query is displayed in a formatted table:

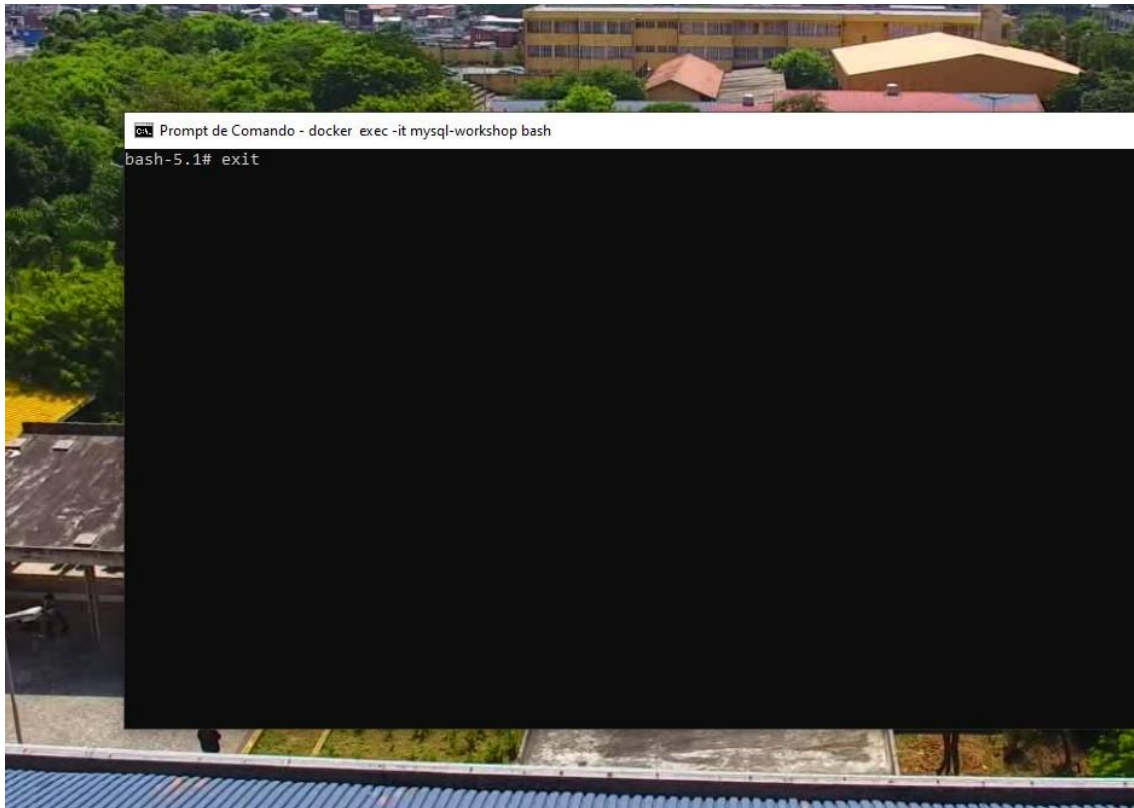
Database					
information_schema	mysql	performance_schema	sys	test_db	
Tables_in_test_db					
customers					

id	name	email	age	created_at
1	John Doe	john@example.com	30	2025-04-11 00:43:04
2	Jane Smith	jane@example.com	25	2025-04-11 00:43:04
3	Bob Johnson	bob@example.com	40	2025-04-11 00:43:04

```
bash-5.1#
```

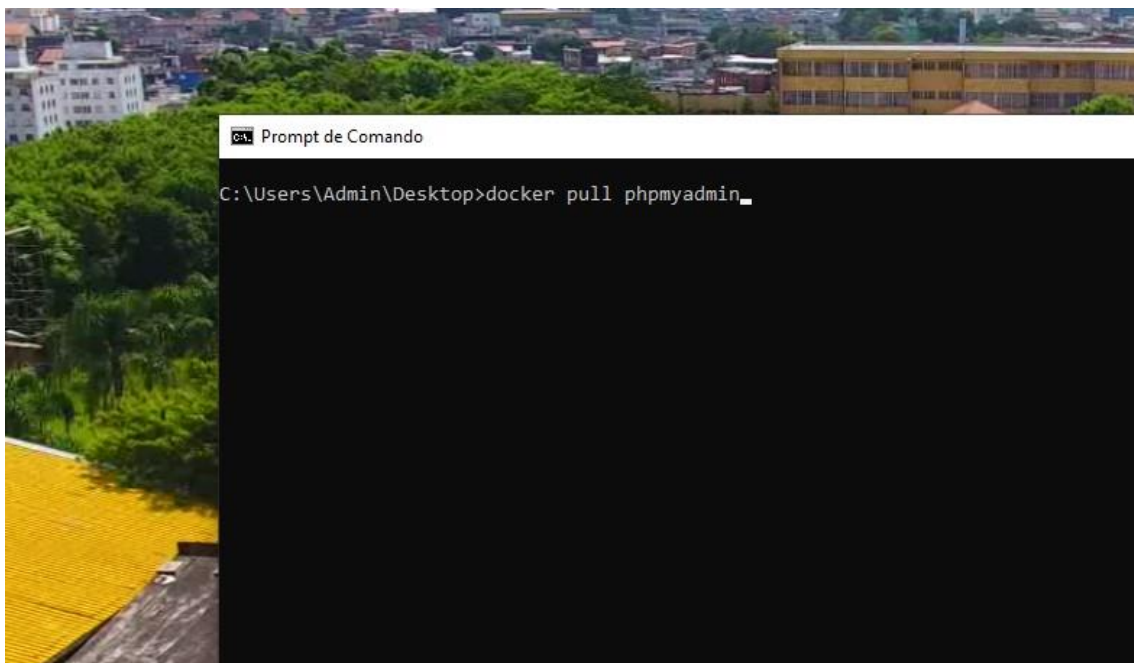
Digite **exit** no bash para sair do container



Agora digite o seguinte código:

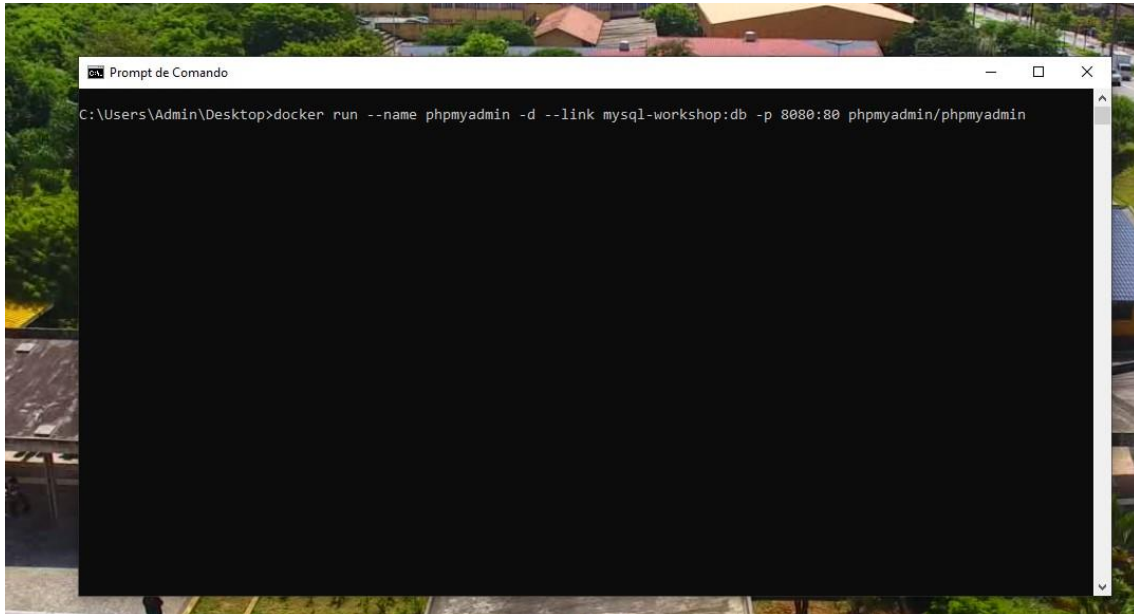
**docker pull phpmyadmin**

para baixar a imagem do Docker phpmyadmin

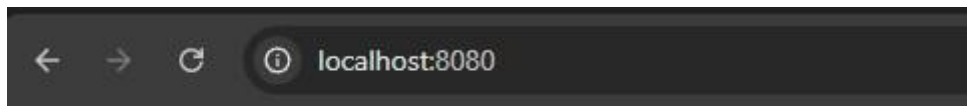


Crie o container phpmyadmin e link ele ao container mysql com o código:

**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



Bem-vindo ao phpMyAdmin

Idioma (*Language*)

Português (Brasil) - Portuguese (Brazil)



Entrar 

Usuário:

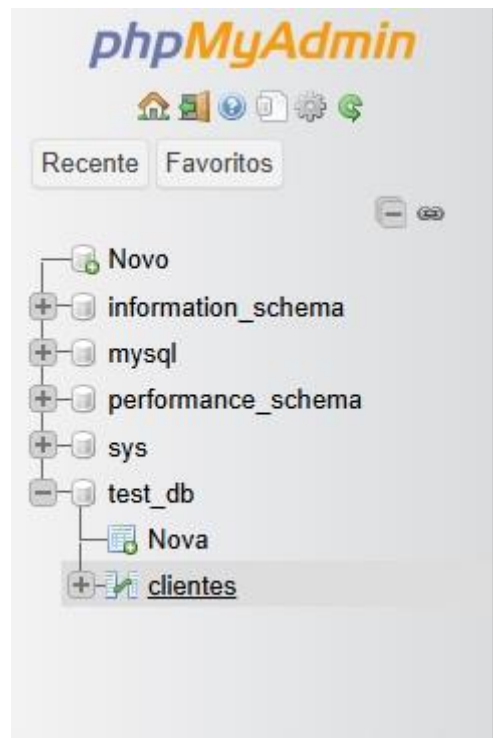
root

Senha:

....

Entrar

Vá para a tabela clientes



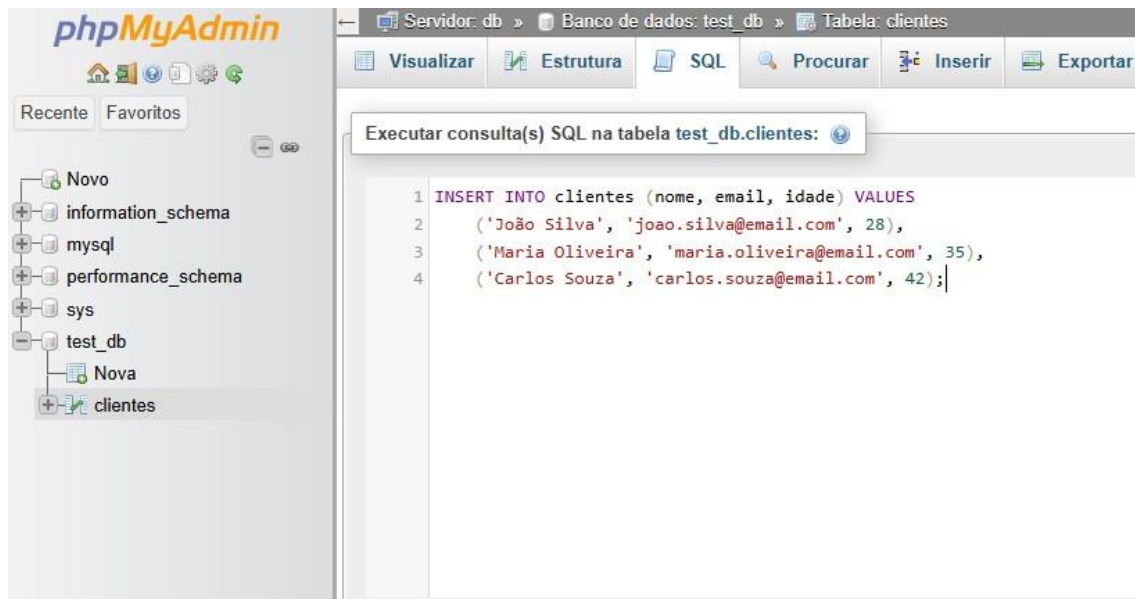
Va para a aba SQL



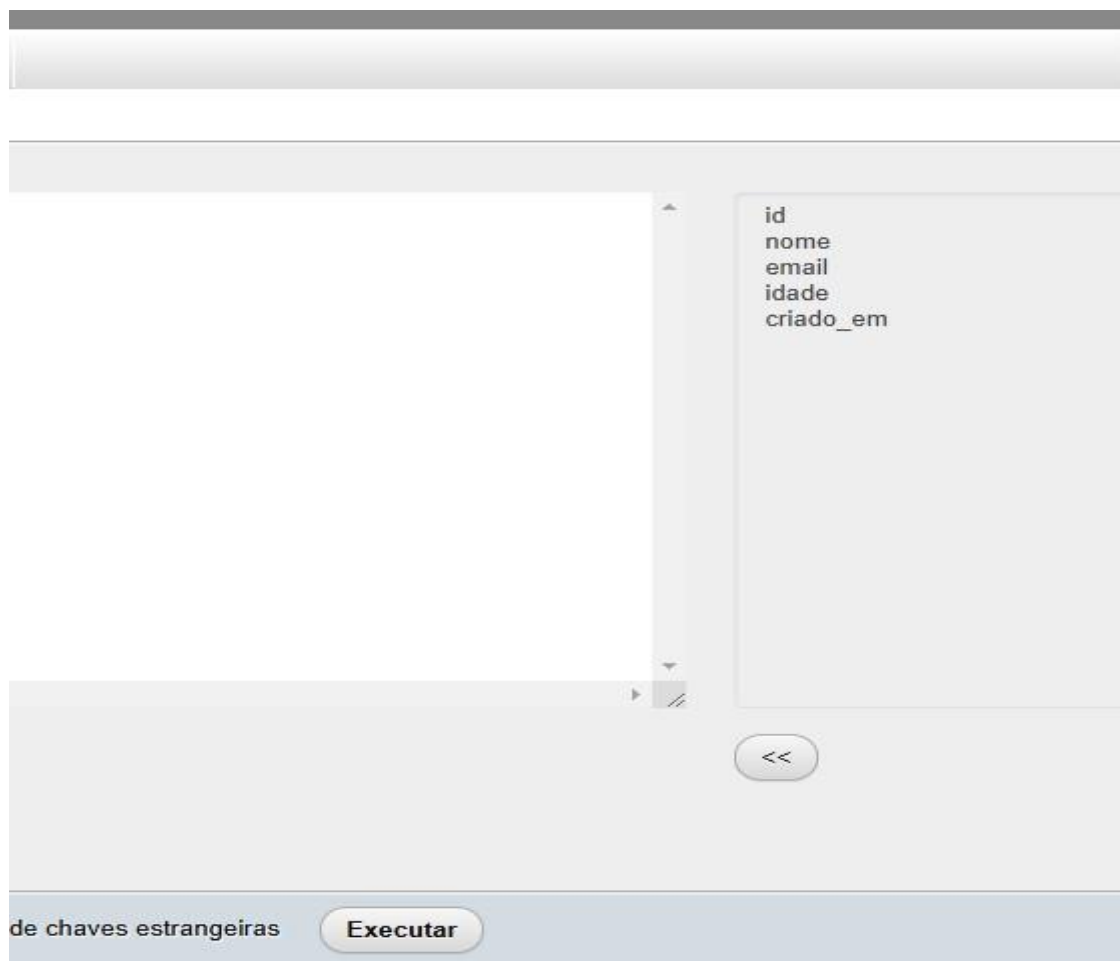
E digite adicione alguns clientes:

```
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);
```



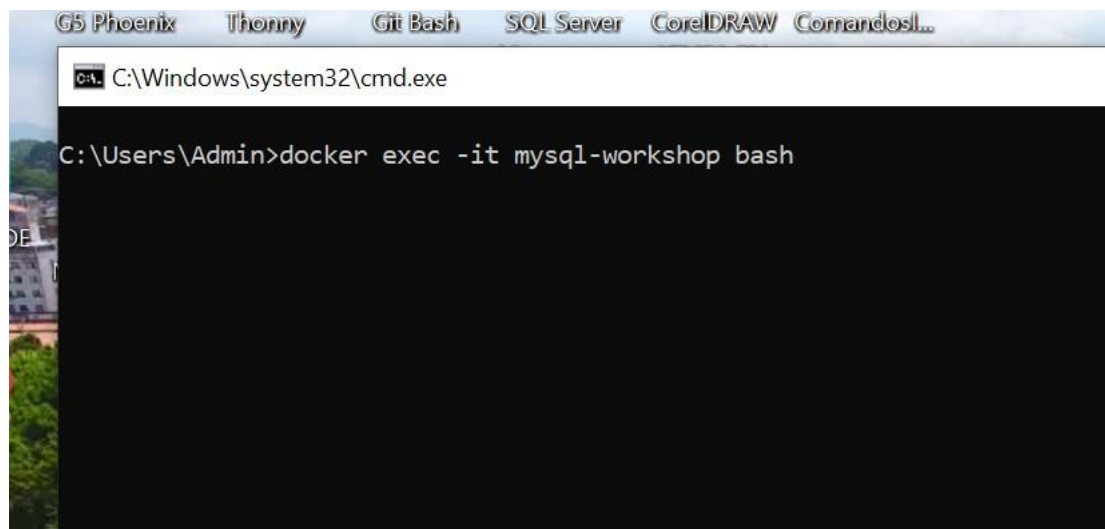


Execute o código



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

**docker exec -it mysql-workshop bash**

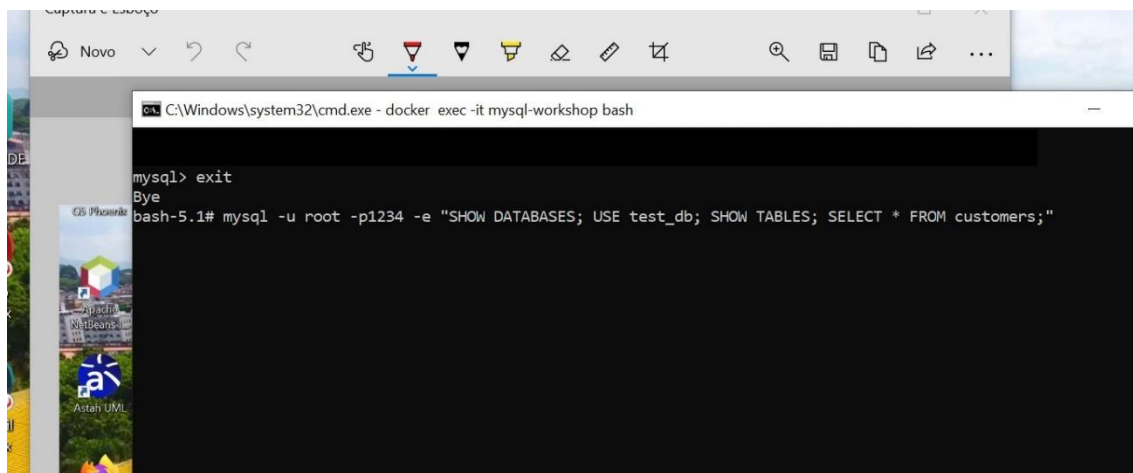


```
C:\Windows\system32\cmd.exe

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

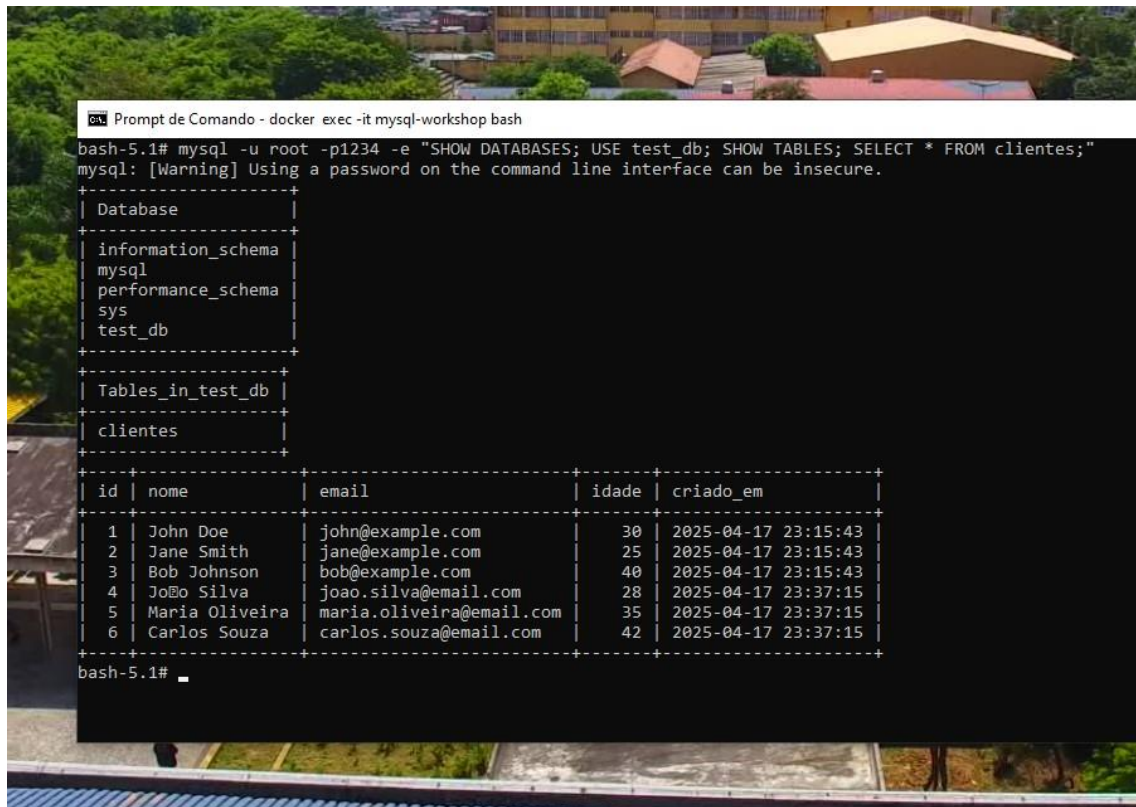
E verifique se os inserts foram feitos com sucesso com o código:

**mysql -u root -p1234 -e "SHOW DATABASES; USE test\_db;  
SHOW TABLES; SELECT \* FROM clientes;"**



```
C:\Windows\system32\cmd.exe - docker exec -it mysql-workshop bash

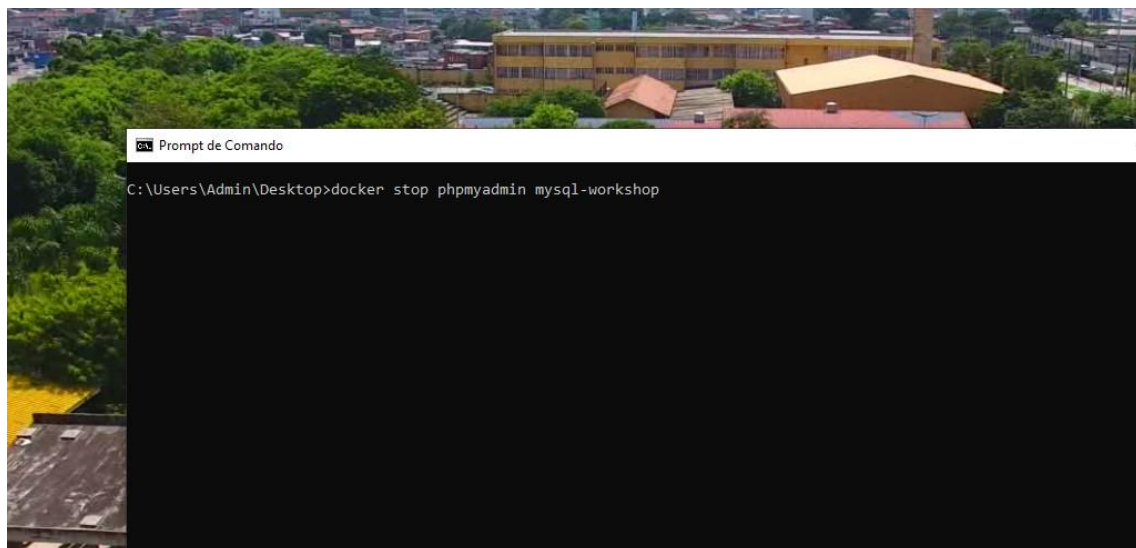
mysql> exit
Bye
bash-5.1# mysql -u root -p1234 -e "SHOW DATABASES; USE test_db; SHOW TABLES; SELECT * FROM customers;"
```

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 command being executed: 'bash-5.1# mysql -u root -p1234 -e "SHOW DATABASES; USE test\_db; SHOW TABLES; SELECT \* FROM clientes;"'. The output shows the databases, the tables in the test\_db database, and the contents of the clientes table.

```
Prompt de Comando - docker exec -it mysql-workshop bash
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 | João 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. It shows the command 'C:\Users\Admin\Desktop>docker stop phpmyadmin mysql-workshop' being entered.

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

E delete os containers com o código:

**docker stop phpmyadmin mysql-container**

