

```
1 !curl ipecho.net/plain
```

```
34.138.42.127
```

```
1 # Instalação do pacote mysql-connector-python usando o gerenciador de pacotes pip
```

```
2 !pip install mysql-connector-python
```

```
Collecting mysql-connector-python
```

```
  Downloading mysql_connector_python-8.3.0-cp310-cp310-manylinux_2_17_x86_64.whl (21.5 MB)
    21.5/21.5 MB 35.6 MB/s eta 0:00:00
```

```
Installing collected packages: mysql-connector-python
```

```
Successfully installed mysql-connector-python-8.3.0
```

```
1 # Importação das bibliotecas necessárias
```

```
2 import mysql.connector # necessária para se conectar com o banco de dados MySQL
```

```
3 import pandas as pd # Será usada para visualização das tabelas extraídas do My SQL
```

```
1 # Definição das informações de conexão ao servidor (instância)
```

```
2 host = '34.172.110.76'
```

```
3 user = 'root'
```

```
4 password = ''
```

```
1 '''
```

```
2 Esta consulta acontece no nível instância, e serve para ver quais bancos
```

```
3 temos em nossa instância
```

```
4 '''
```

```
5
```

```
6 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
```

```
7 connection = mysql.connector.connect(
```

```
8     host=host,
```

```
9     user=user,
```

```
10    password=password,
```

```
11 )
```

```
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
```

```
13 query = 'show databases;' # Definição da consulta SQL a ser executada
```

```
14 cursor.execute(query) # Execução da consulta no banco de dados
```

```
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
```

```
16 tables = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
```

```
17 display(tables) # Exibição do DataFrame Pandas contendo os resultados da consulta
```

```
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
```

```
19 connection.close() # Fechamento da conexão após a consulta
```

	Database
0	douglas
1	information_schema
2	mysql
3	performance_schema
4	sys

```
1 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
```

```
2 connection = mysql.connector.connect(
```

```
3     host=host,
```

```
4     user=user,
```

```
5     password=password,
```

```
6 )
```

```
7 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
```

```
8 query = 'create database sextou;' # Definição da consulta SQL a ser executada
```

```
9 cursor.execute(query) # Execução da consulta no banco de dados
```

```
10 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
```

```
11 tables = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
```

```
12 display(tables) # Exibição do DataFrame Pandas contendo os resultados da consulta
```

```
13 cursor.close() # Fechamento do cursor após a conclusão da consulta
14 connection.close() # Fechamento da conexão após a consulta

—

1 # Definição das informações de conexão ao banco de dados
2 host = '34.172.110.76'
3 user = 'root'
4 password = ''
5 database = 'sextou'

1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 create table Cliente (
14     id_cliente INT PRIMARY KEY,
15     nome VARCHAR (50),
16     endereco VARCHAR (100)
17 );
18 ''' # Definição da consulta SQL a ser executada
19 cursor.execute(query) # Execução da consulta no banco de dados
20 cursor.close() # Fechamento do cursor após a conclusão da consulta
21 connection.close() # Fechamento da conexão após a consulta

1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 CREATE TABLE Pedido (
14     id_pedido INT PRIMARY KEY,
15     id_cliente INT,
16     descricao VARCHAR(100),
17     FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
18 );
19 ''' # Definição da consulta SQL a ser executada
20 cursor.execute(query) # Execução da consulta no banco de dados
21 cursor.close() # Fechamento do cursor após a conclusão da consulta
22 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 CREATE TABLE Produto (
14     id_produto INT PRIMARY KEY,
15     nome VARCHAR(50),
16     preco DECIMAL(10, 2)
17 );
18 ''' # Definição da consulta SQL a ser executada
19 cursor.execute(query) # Execução da consulta no banco de dados
20 cursor.close() # Fechamento do cursor após a conclusão da consulta
21 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 CREATE TABLE ItemPedido (
14     id_item INT PRIMARY KEY,
15     id_pedido INT,
16     id_produto INT,
17     quantidade INT,
18     FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
19     FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)
20 );
21 ''' # Definição da consulta SQL a ser executada
22 cursor.execute(query) # Execução da consulta no banco de dados
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 CREATE TABLE Categoria (
14     id_categoria INT PRIMARY KEY,
15     nome VARCHAR(50)
16 );
17 ''' # Definição da consulta SQL a ser executada
18 cursor.execute(query) # Execução da consulta no banco de dados
19 cursor.close() # Fechamento do cursor após a conclusão da consulta
20 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 CREATE TABLE ProdutoCategoria (
14     id_produto INT,
15     id_categoria INT,
16     PRIMARY KEY (id_produto, id_categoria),
17     FOREIGN KEY (id_produto) REFERENCES Produto(id_produto),
18     FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)
19 );
20 ''' # Definição da consulta SQL a ser executada
21 cursor.execute(query) # Execução da consulta no banco de dados
22 cursor.close() # Fechamento do cursor após a conclusão da consulta
23 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO Cliente (id_cliente, nome, endereco)
14 VALUES
15     (1, 'Douglas', 'SP'),
16     (2, 'Daniela', 'SP'),
17     (3, 'Ester', 'MG'),
18     (4, 'Maria', 'BA'),
19     (5, 'Douglas', 'PE'),
20     (6, 'Raimundo', 'CE'),
21     (7, 'Jucimara', 'GO'),
22     (8, 'Marlene', 'AC'),
23     (9, 'Douglas', 'MA'),
24     (10, 'Daniela', 'PI'),
25     (11, 'Pedro', 'PB'),
26     (12, 'Joaquim', 'RS'),
27     (13, 'Julio', 'TO'),
28     (14, 'Carla', 'DF'),
29     (15, 'Bianca', 'PA'),
30     (16, 'Sandra', 'SP'),
31     (17, 'Celso', 'SP'),
32     (18, 'Silvana', 'BA'),
33     (19, 'Prof Raquel', 'BA'),
34     (20, 'Pedro', 'SP'),
35     (21, 'Erick', 'BA'),
36     (22, 'Denise', 'DF'),
37     (23, 'Diana', 'SP'),
38     (24, 'Dagmar', 'DF'),
39     (25, 'Deiverson', 'SP');
40 ''' # Definição da consulta SQL a ser executada
41 cursor.execute(query) # Execução da consulta no banco de dados
42 connection.commit() # Confirmar as alterações
43 cursor.close() # Fechamento do cursor após a conclusão da consulta
44 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO Pedido (id_pedido, id_cliente, descricao)
14 VALUES
15     (1, '1', 'Caneca'),
16     (2, '1', 'Garfo'),
17     (3, '2', 'Faca'),
18     (4, '5', 'Copo'),
19     (5, '19', 'Copo'),
20     (6, '19', 'Colher'),
21     (7, '15', 'Colher'),
22     (8, '8', 'Colher'),
23     (9, '2', 'Copo'),
24     (10, '19', 'Colher'),
25     (11, '6', 'Copo'),
26     (12, '11', 'Colher'),
27     (13, '10', 'Garfo'),
28     (14, '4', 'Garfo'),
29     (15, '3', 'Colher'),
30     (16, '3', 'Garfo'),
31     (17, '2', 'Copo'),
32     (18, '17', 'Garfo'),
33     (19, '14', 'Caneca'),
34     (20, '13', 'Caneca');
35 ''' # Definição da consulta SQL a ser executada
36 cursor.execute(query) # Execução da consulta no banco de dados
37 connection.commit() # Confirmar as alterações
38 cursor.close() # Fechamento do cursor após a conclusão da consulta
39 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO Produto (id_produto, nome, preco)
14 VALUES
15     (1, 'Caneca', 15.99),
16     (2, 'Garfo', 5.99),
17     (3, 'Faca', 7.99),
18     (4, 'Colher', 3.99),
19     (5, 'Copo', 10.99);
20 ''' # Definição da consulta SQL a ser executada
21 cursor.execute(query) # Execução da consulta no banco de dados
22 connection.commit() # Confirmar as alterações
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO ItemPedido (id_item, id_pedido, id_produto, quantidade)
14 VALUES
15     (1, 1, 1, 2),
16     (2, 2, 2, 5),
17     (3, 3, 3, 7),
18     (4, 4, 5, 10),
19     (5, 5, 5, 11),
20     (6, 6, 4, 30),
21     (7, 7, 4, 32),
22     (8, 8, 4, 44),
23     (9, 9, 5, 13),
24     (10, 10, 4, 33),
25     (11, 11, 5, 34),
26     (12, 12, 4, 32),
27     (13, 13, 2, 33),
28     (14, 14, 2, 34),
29     (15, 15, 4, 27),
30     (16, 16, 2, 18),
31     (17, 17, 5, 17),
32     (18, 18, 2, 33),
33     (19, 19, 1, 50),
34     (20, 20, 1, 47);
35 ''' # Definição da consulta SQL a ser executada
36 cursor.execute(query) # Execução da consulta no banco de dados
37 connection.commit() # Confirmar as alterações
38 cursor.close() # Fechamento do cursor após a conclusão da consulta
39 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO Categoria (id_categoria, nome)
14 VALUES
15     (1, 'Comer'),
16     (2, 'Beber');
17 ''' # Definição da consulta SQL a ser executada
18 cursor.execute(query) # Execução da consulta no banco de dados
19 connection.commit() # Confirmar as alterações
20 cursor.close() # Fechamento do cursor após a conclusão da consulta
21 connection.close() # Fechamento da conexão após a consulta
```

```
1 '''
2 Objetivo: Criar uma tabela no banco de dados
3 '''
4 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
5 connection = mysql.connector.connect(
6     host=host,
7     database=database,
8     user=user,
9     password=password
10 )
11 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
12 query = '''
13 INSERT INTO ProdutoCategoria (id_produto, id_categoria)
14 VALUES
15     (1, 2),
16     (2, 1),
17     (3, 1),
18     (4, 1),
19     (5, 2);
20 ''' # Definição da consulta SQL a ser executada
21 cursor.execute(query) # Execução da consulta no banco de dados
22 connection.commit() # Confirmar as alterações
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta

1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela Cliente
4 '''
5 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM Cliente' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 clientes = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
17 display(clientes) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona todas as colunas da tabela: Cliente e armazena
22 os resultados da consulta em um dataframe Pandas chamado clientes
23 '''
```


	id_cliente	nome	endereco
0	1	Douglas	SP
1	2	Daniela	SP
2	3	Ester	MG
3	4	Maria	BA
4	5	Douglas	PE
5	6	Raimundo	CE
6	7	Jucimara	GO
7	8	Marlene	AC
8	9	Douglas	MA
9	10	Daniela	PI
10	11	Pedro	PB
11	12	Joaquim	RS
12	13	Julio	TO
13	14	Carla	DF
14	15	Bianca	PA
15	16	Sandra	SP
16	17	Celso	SP
17	18	Silvana	BA
18	19	Prof Raquel	BA
19	20	Pedro	SP
20	21	Erick	BA
21	22	Denise	DF
22	23	Diana	SP
23	24	Dagmar	DF
24	25	Deiverson	SP

'\nResumo do comando: Selecciona todas as colunas da tabela: Cliente e
armazena\nos resultados da consulta em um dataframe Pandas chamado cl

```
1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela Cliente
4 '''
5 # Estabelecimento da conexão com o banco de dados usando as informações fornecidas
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM Pedido' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 pedidos = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os result
17 display(pedidos) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona todas as colunas da tabela: Cliente e armazena
22 os resultados da consulta em um dataframe Pandas chamado clientes
23 '''
```

	id_pedido	id_cliente	descricao
0	1	1	Caneca
1	2	1	Garfo
2	3	2	Faca
3	4	5	Copo
4	5	19	Copo
5	6	19	Colher
6	7	15	Colher
7	8	8	Colher
8	9	2	Copo
9	10	19	Colher
10	11	6	Copo
11	12	11	Colher
12	13	10	Garfo
13	14	4	Garfo
14	15	3	Colher
15	16	3	Garfo
16	17	2	Copo
17	18	17	Garfo
18	19	14	Caneca
19	20	13	Caneca

\nResumo do comando: Seleciona todas as colunas da tabela: Cliente e armazena\nos resultados da consulta em um dataframe Pandas chamado cl

```

1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela Produto
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM Produto;' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 produtos = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
17 display(produtos) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona todas as colunas da tabela: Produto e armazena
22 os resultados da consulta em um dataframe Pandas chamado produtos
23 '''

```

	id_produto	nome	preco
0	1	Caneca	15.99
1	2	Garfo	5.99
2	3	Faca	7.99
3	4	Colher	3.99
4	5	Copo	10.99

\nResumo do comando: Seleciona todas as colunas da tabela: Produto e armazena\nos resultados da consulta em um dataframe Pandas chamado pr

```

1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela ItemPedido
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM ItemPedido;' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 itempedido = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os res
17 display(itempedido) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona todas as colunas da tabela: ItemPedido e armazena
22 os resultados da consulta em um dataframe Pandas chamado itempedido
23 '''

```

	id_item	id_pedido	id_produto	quantidade
0	1	1	1	2
1	2	2	2	5
2	3	3	3	7
3	4	4	5	10
4	5	5	5	11
5	6	6	4	30
6	7	7	4	32
7	8	8	4	44
8	9	9	5	13
9	10	10	4	33
10	11	11	5	34
11	12	12	4	32
12	13	13	2	33
13	14	14	2	34
14	15	15	4	27
15	16	16	2	18
16	17	17	5	17
17	18	18	2	33
18	19	19	1	50
19	20	20	1	47

\nResumo do comando: Selecciona todas as colunas da tabela: ItemPedid
o e armazena\nos resultados da consulta em um dataframe Pandas chamado

```
1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela Categoria
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM Categoria;' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 categoria = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resu
17 display(categoria) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Selecciona todas as colunas da tabela: Categoria e armazena
22 os resultados da consulta em um dataframe Pandas chamado categoria
23 '''
```

	id_categoria	nome
0	1	Comer
1	2	Beber

\nResumo do comando: Selecciona todas as colunas da tabela: Categoria
e armazena\nos resultados da consulta em um dataframe Pandas chamado

```

1 '''
2 Objetivo: Fazer uma consulta que retorne todos os campos e todos os registros
3 da tabela ProdutoCategoria
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor()# Criação de um cursor para executar consultas SQL no banco de dados
13 query = 'SELECT * FROM ProdutoCategoria;' # Definição da consulta SQL a ser executada
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 produtocategoria = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando
17 display(produtocategoria) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona todas as colunas da tabela: ProdutoCategoria e armazena
22 os resultados da consulta em um dataframe Pandas chamado produtocategoria
23 '''

```

	id_produto	id_categoria
0	2	1
1	3	1
2	4	1
3	1	2
4	5	2

'\nResumo do comando: Seleciona todas as colunas da tabela: ProdutoCategoria e armazena\nnos resultados da c
onsulta em um dataframe Pandas chamado produtocategoria\n'

```

1 # Abertura da conexão para uma nova consulta
2 connection = mysql.connector.connect(
3     host=host,
4     user=user,
5     password=password,
6     database=database,
7 )
8 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
9 # Definição da consulta SQL a ser executada
10 query = '''
11 SELECT id_produto, quantidade FROM ItemPedido
12 WHERE quantidade >= 40;
13 '''
14 cursor.execute(query) # Execução da consulta no banco de dados
15 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
16 where1 = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resulta
17 display(where1) # Exibição do DataFrame Pandas contendo os resultados da consulta
18 cursor.close() # Fechamento do cursor após a conclusão da consulta
19 connection.close() # Fechamento da conexão após a consulta
20 '''
21 Resumo do comando: Seleciona as colunas: id_produto e quantidade da tabela:
22 ItemPedido, quando a quantidade for maior ou igual a 40 e armazena
23 os resultados da consulta em um dataframe Pandas chamado where1
24 '''

```

	id_produto	quantidade
0	4	44
1	1	50
2	1	47

\nResumo do comando: Seleciona as colunas: id_produto e quantidade da tabela:\nItemPedido, quando a quanti
dade for maior ou igual a 40 e armazena\nnos resultados da consulta em um dataframe Pandas chamado where1

```
1 # Abertura da conexão para uma nova consulta
2 connection = mysql.connector.connect(
3     host=host,
4     user=user,
5     password=password,
6     database=database,
7 )
8 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
9 # Definição da consulta SQL a ser executada
10 query = '''SELECT * FROM Cliente
11 WHERE endereco = "SP";''' #campo de texto precisa estar em aspas na consulta
12 cursor.execute(query) # Execução da consulta no banco de dados
13 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
14 where2 = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resulta
15 display(where2) # Exibição do DataFrame Pandas contendo os resultados da consulta
16 cursor.close() # Fechamento do cursor após a conclusão da consulta
17 connection.close() # Fechamento da conexão após a consulta
18 '''
19 Resumo do comando: Seleciona todas as colunas, quando o endereço for igual a SP
20 e armazena os resultados da consulta em um dataframe Pandas chamado where2
21 '''
```

	id_cliente	nome	endereco
0	1	Douglas	SP
1	2	Daniela	SP
2	16	Sandra	SP
3	17	Celso	SP
4	20	Pedro	SP
5	23	Diana	SP
6	25	Deiverson	SP

\nResumo do comando: Seleciona todas as colunas, quando o endereço f
or igual a SP\ne armazena os resultados da consulta em um dataframe P

```

1 '''
2 Vamos agora fazer uma consulta que retorne distintamente os ids de produto
3 e quantidade onde a quantidade for igual a 33
4 '''
5
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT DISTINCT id_produto, quantidade FROM ItemPedido
17 WHERE quantidade = 33;
18 '''
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
22 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta

```

	id_produto	quantidade
0	2	33
1	4	33

```

1 '''
2 Vamos agora fazer uma consulta que retorne distintamente os nomes
3 presentes na tabela clientes
4 '''
5
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT DISTINCT nome FROM Cliente
17 '''
18 cursor.execute(query) # Execução da consulta no banco de dados
19 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
20 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
21 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
22 cursor.close() # Fechamento do cursor após a conclusão da consulta
23 connection.close() # Fechamento da conexão após a consulta

```

	nome
0	Douglas
1	Daniela
2	Ester
3	Maria
4	Raimundo
5	Jucimara
6	Marlene
7	Pedro
8	Joaquim
9	Julio
10	Carla
11	Bianca
12	Sandra
13	Celso
14	Silvana
15	Prof Raquel
16	Erick
17	Denise
18	Diana
19	Dagmar
20	Deiverson

```

1 '''
2 Agora imagine que você deseja consultar todas as colunas da tabela cliente,
3 no entanto, quer exibir apenas as primeiras 5 linhas
4 '''
5
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT * FROM Cliente
17 LIMIT 5;
18 '''
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 limit = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
22 display(limit) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
25 '''
26
27 # Resumo do comando: Lendo o comando: Seleciona todos os atributos da tabela
28 # Cliente e mostra apenas os cinco primeiros registros, e armazena os resultados
29 # da consulta em um dataframe Pandas chamado limit
30 '''

```


	id_cliente	nome	endereco
0	1	Douglas	SP
1	2	Daniela	SP
2	3	Ester	MG
3	4	Maria	BA
4	5	Douglas	PE

\nResumo do comando: Lendo o comando: Selecciona todos os atributos d
a tabela\ncliente e mostra apenas os cinco primeiros registros e arm

```
1 '''
2 Agora vamos fazer uma consulta que retorne todos os ids e nomes de clientes que
3 sejam da BA e tenha id 19
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT id_cliente, nome
16 FROM Cliente
17 WHERE endereco = 'BA'
18 AND id_cliente = 19;
19 '''
20 cursor.execute(query) # Execução da consulta no banco de dados
21 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
22 e = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados d
23 display(e) # Exibição do DataFrame Pandas contendo os resultados da consulta
24 cursor.close() # Fechamento do cursor após a conclusão da consulta
25 connection.close() # Fechamento da conexão após a consulta
26 '''
27 Resumo do comando: Selecciona as colunas: nome e endereco da tabela: Clientes,
28 apenas os registros com endereco igual a 'BA' e id_cliente igual a 19, e
29 armazena os resultados da consulta em um dataframe Pandas chamado e
30 '''
31
```

	id_cliente	nome
0	19	Prof Raquel

\nResumo do comando: Selecciona as colunas: nome e endereco da tabel
a: Clientes,\napenas os registros com endereco igual a 'BA' e id_clie
nte igual a 19. e\narmazena os resultados da consulta em um dataframe

```

1 '''
2 Agora vamos fazer uma consulta que retorne todos os nomes e endereços da tabela
3 cliente que sejam de SP ou MG
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT nome, endereco
16 FROM Cliente
17 WHERE endereco = 'SP'
18 OR endereco = 'MG';
19 '''
20 cursor.execute(query) # Execução da consulta no banco de dados
21 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
22 ou = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
23 display(ou) # Exibição do DataFrame Pandas contendo os resultados da consulta
24 cursor.close() # Fechamento do cursor após a conclusão da consulta
25 connection.close() # Fechamento da conexão após a consulta
26 '''
27 Resumo do comando: Seleciona as colunas: nome e endereco da tabela: Clientes,
28 apenas os registros com endereco igual a 'SP' ou endereco a 'MG', e
29 armazena os resultados da consulta em um dataframe Pandas chamado ou
30 '''
31

```

	nome	endereco
0	Douglas	SP
1	Daniela	SP
2	Ester	MG
3	Sandra	SP
4	Celso	SP
5	Pedro	SP
6	Diana	SP
7	Deiverson	SP

'\nResumo do comando: Seleciona as colunas: nome e endereco da tabel
 a: Clientes,\napenas os registros com endereco igual a 'SP' ou endere
 co a 'MG'. e\narmazena os resultados da consulta em um dataframe Pand

```

1 '''
2 agora vamos fazer uma consulta que retorne nome e endereço dos clientes que não
3 sejam de SP
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT nome, endereco
16 FROM Cliente
17 WHERE NOT endereco = 'SP';
18 '''
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 nao = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
22 display(nao) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
25 '''
26 Resumo do comando: Seleciona as colunas: nome e endereco da tabela: Clientes,
27 apenas os registros com endereco não seja igual a 'SP', e armazena os resultados
28 da consulta em um dataframe Pandas chamado nao
29 '''
30

```

	nome	endereco
0	Ester	MG
1	Maria	BA
2	Douglas	PE
3	Raimundo	CE
4	Jucimara	GO
5	Marlene	AC
6	Douglas	MA
7	Daniela	PI
8	Pedro	PB
9	Joaquim	RS
10	Julio	TO
11	Carla	DF
12	Bianca	PA
13	Silvana	BA
14	Prof Raquel	BA
15	Erick	BA
16	Denise	DF
17	Dagmar	DF

\nResumo do comando: Seleciona as colunas: nome e endereco da tabel
 a: Clientes.\nanenas os registros com endereco não seja igual a 'SP'.

```

1 '''
2 imagine que voce deseja fazer uma consulta que retorne a quantidade média
3 do atributo quantidade da tabela ItemPedido, arredondando a média para 2 casas
4 decimais
5 '''
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT AVG(quantidade) AS qtde_media,
17 round(AVG(quantidade),2) AS qtde_media_arredondada
18 FROM ItemPedido;
19 '''
20 cursor.execute(query) # Execução da consulta no banco de dados
21 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
22 media = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
23 display(media) # Exibição do DataFrame Pandas contendo os resultados da consulta
24 cursor.close() # Fechamento do cursor após a conclusão da consulta
25 connection.close() # Fechamento da conexão após a consulta
26 '''
27 Resumo do comando: Seleciona as colunas: quantidade da tabela ItemPedido,
28 retorna o valor arredondado (ROUND) médio da coluna (AVG), e armazena os
29 resultados da consulta em um dataframe Pandas chamado media
30 '''
31

```

	qtde_media	qtde_media_arredondada
0	25.6000	25.60

Resumo do comando: Seleciona as colunas: quantidade da tabela ItemPedido, retorna o valor arredondado (ROUND) médio da coluna (AVG), e armazena os resultados da consulta em um dataframe Pandas chamado media

```

1 '''
2 Nosso objetivo agora é fazer uma consulta sobre a tabela ItemPedido no
3 atributo quantidade que retorne a soma de todas as quantidades
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT SUM(quantidade) AS soma_qtde
16 FROM ItemPedido;
17 '''
18 cursor.execute(query) # Execução da consulta no banco de dados
19 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
20 soma = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
21 display(soma) # Exibição do DataFrame Pandas contendo os resultados da consulta
22 cursor.close() # Fechamento do cursor após a conclusão da consulta
23 connection.close() # Fechamento da conexão após a consulta
24 '''
25 Resumo do comando: Seleciona as colunas: quantidade da tabela ItemPedido,
26 retorna o somatório da coluna, e armazena os resultados da consulta em um
27 dataframe Pandas chamado soma
28 '''

```

soma_qtde	
0	512

\nResumo do comando: Selecciona as colunas: quantidade da tabela Item Pedido.\nretorna o somatório da coluna. e armazena os resultados da c

```

1 '''
2 Agora vamos fazer uma consulta sobre a tabela ItemPedido no atributo
3 quantidade e que retorne o maior valor encontrado nessa coluna
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT MAX(quantidade) AS maior_quantidade_vendida
16 FROM ItemPedido;
17 '''
18 cursor.execute(query) # Execução da consulta no banco de dados
19 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
20 max = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados c
21 display(max) # Exibição do DataFrame Pandas contendo os resultados da consulta
22 cursor.close() # Fechamento do cursor após a conclusão da consulta
23 connection.close() # Fechamento da conexão após a consulta
24 '''
25 Resumo do comando: Maior quantidade de produtos vendidos da tabela:ItemPedido,
26 e armazena os resultados da consulta em um dataframe Pandas chamado max
27 '''

```

maior_quantidade_vendida	
0	50

\nResumo do comando: Maior quantidade de produtos vendidos da tabel
a:ItemPedido.\ne armazena os resultados da consulta em um dataframe P

```

1 '''
2 Agora vamos fazer uma consulta sobre a tabela ItemPedido no atributo
3 quantidade e que retorne o menor valor encontrado nessa coluna
4 '''
5
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT MIN(quantidade) menor_quantidade_venda
17 FROM ItemPedido;
18 ''' # nesse faltou a palavra reserva AS, mesmo assim o MySQL entende e realiza a consulta
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 min = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
22 display(min) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
25 '''
26 Resumo do comando:Seleciona a menor quantidade de produtos vendidos da
27 tabela:ItemPedido, e armazena os resultados da consulta em um dataframe
28 Pandas chamado min
29 '''

```

menor_quantidade_venda	
0	2

\nResumo do comando:Seleciona a menor quantidade de produtos vendido
s da\tabela:ItemPedido. e armazena os resultados da consulta em um d

```

1 '''
2 Agora vamos fazer uma consulta sobre a tabela Produto no atributo
3 id produto que retorne a contagem de quantos id temos nessa tabela
4 '''
5
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT count(id_produto) as qtd_produtos
17 FROM Produto;
18 '''
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 contagem = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
22 display(contagem) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta
25 '''
26 Resumo do comando: Seleciona a coluna ide_produto e faz a contagem de itens a
27 partir da tabela:Produto, e armazena os resultados da consulta em um dataframe
28 Pandas chamado contagem
29 '''

```

qtd_produtos	
0	5

'\nResumo do comando: Selecciona a coluna ide_produto e faz a contagem de itens a partir da tabela Produto. e armazena os resultados da con

```

1 '''
2 Agora Imagine que você deseja fazer uma consulta na tabela Cliente que retorne
3 a frequência de cada endereço
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT endereco
16        , count(*) as freq
17 FROM Cliente
18 GROUP BY endereco;
19 '''
20 cursor.execute(query) # Execução da consulta no banco de dados
21 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
22 grupo = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultad
23 display(grupo) # Exibição do DataFrame Pandas contendo os resultados da consulta
24 cursor.close() # Fechamento do cursor após a conclusão da consulta
25 connection.close() # Fechamento da conexão após a consulta
26 '''
27 Resumo do comando: Comando que seleciona o atributo endereço, cria e seleciona a
28 coluna count renomeada com freq da tabela Cliente e agrupa por endereço,
29 e armazena os resultados da consulta em um dataframe Pandas chamado grupo
30 '''

```

	endereco	freq
0	SP	7
1	MG	1
2	BA	4
3	PE	1
4	CE	1
5	GO	1
6	AC	1
7	MA	1
8	PI	1
9	PB	1
10	RS	1
11	TO	1
12	DF	3
13	PA	1

'\nResumo do comando: Comando que seleciona o atributo endereço, cria e seleciona a\ncoluna count renomeada com freq da tabela Cliente e agrupa por endereço. e armazena os resultados da consulta em um datafr

```

1 '''
2 Agora Imagine que você deseja fazer uma consulta na tabela Cliente que retorne
3 a frequência de cada endereço, mas apenas para aqueles endereços que tiverem
4 frequência maior do 1
5 '''
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT endereco
17        , count(*)
18   FROM Cliente
19  GROUP BY endereco
20  HAVING count(*) > 1;
21 '''
22 cursor.execute(query) # Execução da consulta no banco de dados
23 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
24 tendo = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados
25 display(tendo) # Exibição do DataFrame Pandas contendo os resultados da consulta
26 cursor.close() # Fechamento do cursor após a conclusão da consulta
27 connection.close() # Fechamento da conexão após a consulta
28 '''
29 Resumo do comando: Consulta que seleciona endereco e contagem de endereco da
30 tabela Cliente, Agrupado por endereço tendo apenas (HAVING) os endereços que se
31 repetem pelo menos duas vezes. e armazena os resultados da consulta
32 em um dataframe Pandas chamado tendo
33 '''

```

	endereco	count(*)
0	SP	7
1	BA	4
2	DF	3

'\nResumo do comando: Consulta que seleciona endereco e contagem de endereco da\ntabela Cliente, Agrupado por endereço tendo apenas (HAVING) os endereços que se\nrepetem pelo menos duas vezes. e armazena os resultados da consulta\nem um dataframe Pandas chamado tendo\n'

```

1 '''
2 Agora nosso objetivo é fazer duas consultas diferentes na mesma tabela e exibir
3 os resultados das duas consultas na mesma tabela
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT id_pedido, id_produto
16   FROM ItemPedido
17  WHERE id_produto = 2
18 UNION
19 SELECT id_pedido, id_produto
20   FROM ItemPedido
21  WHERE id_produto = 3;
22 '''
23 cursor.execute(query) # Execução da consulta no banco de dados
24 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
25 uniao = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultados

```



```

26 display(uniao) # Exibição do DataFrame Pandas contendo os resultados da consulta
27 cursor.close() # Fechamento do cursor após a conclusão da consulta
28 connection.close() # Fechamento da conexão após a consulta
29 '''
30 Resumo do comando: Unindo duas consultas, sendo a primeira como a seleção de
31 id_pedido e id_produto da tabela ItemPedido onde o id_produto é 1, e a segunda
32 como a seleção de id_pedido e id_produto da tabela ItemPedido onde o
33 id_produto é 4 e armazena os resultados da consulta
34 em um dataframe Pandas chamado uniao
35 '''

```

	id_pedido	id_produto
0	2	2
1	13	2
2	14	2
3	16	2
4	18	2
5	3	3

'\nResumo do comando: Unindo duas consultas, sendo a primeira como a
 seleção de\nid_pedido e id_produto da tabela ItemPedido onde o id_pro
 duto é 1, e a segunda\ncomo a seleção de id_pedido e id_produto da ta

```

1 '''
2 esta consulta tem como objetivos selecionar as colunas id_pedido e id_produto
3 da tabela ItemPedido onde id_produto seja 1 ou seja 2 e o id_pedido esteja entre
4 1 e 9
5 '''
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT
17     id_pedido,
18     id_produto
19 FROM ItemPedido
20 WHERE id_produto IN (1, 2)
21     AND id_pedido BETWEEN 1 AND 9;
22 '''
23 cursor.execute(query) # Execução da consulta no banco de dados
24 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
25 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
26 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
27 cursor.close() # Fechamento do cursor após a conclusão da consulta
28 connection.close() # Fechamento da conexão após a consulta

```

	id_pedido	id_produto
0	1	1
1	2	2

```

1 '''
2 Agora vamos fazer uma consulta sobre a tabela Produto que retorne as colunas
3 id_produto e nomes apenas para os produtos cujo nome termine com a letra o
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT
16     id_produto,
17     nome
18 FROM Produto
19 WHERE nome LIKE '%o';
20 '''
21 cursor.execute(query) # Execução da consulta no banco de dados
22 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
23 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
24 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
25 cursor.close() # Fechamento do cursor após a conclusão da consulta
26 connection.close() # Fechamento da conexão após a consulta

```

	id_produto	nome
0	2	Garfo
1	5	Copo

```

1 '''
2 Agora vamos fazer uma consulta na mesma tabela da consulta anterior que retorne
3 os mesmos atributos apenas para os casos em que o nome do produto começa
4 com a letra c
5 '''
6 # Abertura da conexão para uma nova consulta
7 connection = mysql.connector.connect(
8     host=host,
9     user=user,
10    password=password,
11    database=database,
12 )
13 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
14 # Definição da consulta SQL a ser executada
15 query = '''
16 SELECT
17     id_produto,
18     nome
19 FROM Produto
20 WHERE nome LIKE 'C%';
21 '''
22 cursor.execute(query) # Execução da consulta no banco de dados
23 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
24 like = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultado
25 display(like) # Exibição do DataFrame Pandas contendo os resultados da consulta
26 cursor.close() # Fechamento do cursor após a conclusão da consulta
27 connection.close() # Fechamento da conexão após a consulta

```

	id_produto	nome
0	1	Caneca
1	4	Colher
2	5	Copo

```

1 '''
2 Agora vamos fazer uma consulta que retorne id_produto e nome para os produtos da
3 tabela Produto que contenham 'lh'
4 '''
5 '''
6 Agora vamos fazer uma consulta na mesma tabela da consulta anterior que retorne
7 os mesmos atributos apenas para os casos em que o nome do produto começa
8 com aletra c
9 '''
10 # Abertura da conexão para uma nova consulta
11 connection = mysql.connector.connect(
12     host=host,
13     user=user,
14     password=password,
15     database=database,
16 )
17 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
18 # Definição da consulta SQL a ser executada
19 query = '''
20 SELECT
21     id_produto,
22     nome
23 FROM Produto
24 WHERE nome LIKE '%lh%';
25 '''
26 cursor.execute(query) # Execução da consulta no banco de dados
27 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
28 like = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resultado
29 display(like) # Exibição do DataFrame Pandas contendo os resultados da consulta
30 cursor.close() # Fechamento do cursor após a conclusão da consulta
31 connection.close() # Fechamento da conexão após a consulta

```

	id_produto	nome
0	4	Colher

```

1 '''
2 Agora vamos fazer uma consulta na tabela Produto que retorne todos os valores
3 nulos da coluna nome
4 '''
5 '''
6 Verificar a existência de valores nulos para o atributo nome na tabela Produto
7 '''
8 # Abertura da conexão para uma nova consulta
9 connection = mysql.connector.connect(
10     host=host,
11     user=user,
12     password=password,
13     database=database,
14 )
15 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
16 # Definição da consulta SQL a ser executada
17 query = '''
18 Select nome from Produto
19 WHERE nome IS NULL;
20 '''
21 cursor.execute(query) # Execução da consulta no banco de dados
22 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
23 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
24 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
25 cursor.close() # Fechamento do cursor após a conclusão da consulta
26 connection.close() # Fechamento da conexão após a consulta
27

```

nome

```

1 '''
2 Agora vamos fazer uma consulta na tabela Produto que retorne todos os valores
3 não nulos da coluna nome
4 '''
5 '''
6 Verificar a existência de valores nulos para o atributo nome na tabela Produto
7 '''
8 # Abertura da conexão para uma nova consulta
9 connection = mysql.connector.connect(
10     host=host,
11     user=user,
12     password=password,
13     database=database,
14 )
15 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
16 # Definição da consulta SQL a ser executada
17 query = '''
18 Select nome from Produto
19 WHERE nome IS NOT NULL;
20 '''
21 cursor.execute(query) # Execução da consulta no banco de dados
22 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
23 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
24 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
25 cursor.close() # Fechamento do cursor após a conclusão da consulta
26 connection.close() # Fechamento da conexão após a consulta

```

	nome
0	Caneca
1	Garfo
2	Faca
3	Colher
4	Copo

```

1 '''
2 Vamos selecionar todos os atributos da tabela ItemPedido, ordenando de forma
3 Crescente por quantidade
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT *
16 FROM ItemPedido
17 ORDER BY quantidade ASC;
18 ''' # se não usarmos asc ou desc, por padrão ele usará asc
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
22 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta

```

	id_item	id_pedido	id_produto	quantidade
0	1	1	1	2
1	2	2	2	5
2	3	3	3	7
3	4	4	5	10
4	5	5	5	11
5	9	9	5	13
6	17	17	5	17
7	16	16	2	18
8	15	15	4	27
9	6	6	4	30
10	7	7	4	32
11	12	12	4	32
12	10	10	4	33
13	13	13	2	33
14	18	18	2	33
15	11	11	5	34
16	14	14	2	34
17	8	8	4	44
18	20	20	1	47
19	19	19	1	50

```

1 '''
2 Vamos selecionar todos os atributos da tabela ItemPedido, ordenando de forma
3 Decrescente por quantidade
4 '''
5 # Abertura da conexão para uma nova consulta
6 connection = mysql.connector.connect(
7     host=host,
8     user=user,
9     password=password,
10    database=database,
11 )
12 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
13 # Definição da consulta SQL a ser executada
14 query = '''
15 SELECT *
16 FROM ItemPedido
17 ORDER BY quantidade DESC;
18 ''' # se não usarmos asc ou desc, por padrão ele usará asc
19 cursor.execute(query) # Execução da consulta no banco de dados
20 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
21 distinct = pd.DataFrame(result, columns=cursor.column_names) # Criação de um DataFrame Pandas usando os resul
22 display(distinct) # Exibição do DataFrame Pandas contendo os resultados da consulta
23 cursor.close() # Fechamento do cursor após a conclusão da consulta
24 connection.close() # Fechamento da conexão após a consulta

```

	id_item	id_pedido	id_produto	quantidade
0	19	19	1	50
1	20	20	1	47
2	8	8	4	44
3	11	11	5	34
4	14	14	2	34
5	10	10	4	33
6	13	13	2	33
7	18	18	2	33
8	7	7	4	32
9	12	12	4	32
10	6	6	4	30
11	15	15	4	27
12	16	16	2	18
13	17	17	5	17
14	9	9	5	13
15	5	5	5	11
16	4	4	5	10
17	3	3	3	7
18	2	2	2	5
19	1	1	1	2

```
1 # Abertura da conexão para uma nova consulta
2 connection = mysql.connector.connect(
3     host=host,
4     user=user,
5     password=password,
6     database=database,
7 )
8 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
9 # Definição da consulta SQL a ser executada
10 query = '''
11 SELECT nome, id_produto,
12         TRIM(nome) AS trim,
13         REPLACE(nome, 'o', 'oo') AS rep,
14         LPAD(id_produto, 5, "0") AS Lpad,
15         SUBSTRING(nome, 1, 2) AS sub,
16         UPPER(nome) AS UPPER,
17         LOWER(nome) AS LOWER.
18 '''
19
20 # Nesse bloco vamos selecionar o atributo id_cliente da tabela pedido e vamos
21 # fazer uma consulta que retorne uma coluna nova chamada id_cliente_texto com
22 # a concatenação (CONCAT) do texto "Cliente - " com o id do cliente
23 # convertido (CAST) em texto
24 query = '''
25 7 # Abertura da conexão para uma nova consulta
26 8 connection = mysql.connector.connect(
27 9     host=host,
28 10    user=user,
29 11    password=password,
30 12    database=database,
31 13 )
32 14 cursor = connection.cursor() # Criação de um cursor para executar consultas SQL no banco de dados
33 15 # Definição da consulta SQL a ser executada
34 16 query = '''
35 17 SELECT id_cliente,
36 18        concat('Cliente - ', CAST(id_cliente AS CHAR)) AS id_cliente_texto
37 19 FROM Pedido;
38 20 '''
39 21 cursor.execute(query) # Execução da consulta no banco de dados
40 22 result = cursor.fetchall() # Recuperação de todos os resultados da consulta
41 23 distinct = pd.DataFrame(result, columns=cursor.column names) # Criação de um DataFrame Pandas usando os resul
```