

Banco de dados

Nome: Brenda.Mônica Gonzaga de Andrade Nunes

Atividade 08: Implementar os CRUDs usando o Python (Aula do dia 10/02/2024) das tabelas do banco de dados criado na atividade-07.

em uma primeira instância cria-se um container docker e entra no container MySQL para a criação das tabelas

```
[node2] (local) root@192.168.0.22 ~
$ docker exec -it mysql-container mysql -uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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>
```

```
$ docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=BD_AVIOES -p 3306:3306 -d mysql
:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
43759093d4f6: Pull complete
d255dceb9ed5: Pull complete
23d22e42ea50: Pull complete
431b106548a3: Pull complete
2be0d473cadf: Pull complete
f56a22f949f9: Pull complete
277ab5f6ddde: Pull complete
df1ba1ac457a: Pull complete
cc9646b08259: Pull complete
893b018337e2: Pull complete
Digest: sha256:146682692a3aa409eae7b7dc6a30f637c6cb49b6ca901c2cd160becc81127d3b
Status: Downloaded newer image for mysql:latest
dcc58b43ec6752359c316e3672f7ded9a1d5bf0eab35898a09d19cb7f44a8a53
[node2] (local) root@192.168.0.22 ~
$
```

- Criação tabela TB_ESCRITORIOS

```
mysql> USE BD_AVIOES;
Database changed
mysql> CREATE TABLE TB_ESCRITORIOS (
  -> id int AUTO_INCREMENT PRIMARY KEY,
  -> cidade text,
  -> telefone text,
  -> endereco_l1 text,
  -> endereco_l2 text,
  -> estado text,
  -> pais text,
  -> territorio text
  -> );
Query OK, 0 rows affected (0.03 sec)
```

- Criação tabela TB_FUNCIONARIOS

```
mysql> CREATE TABLE TB_FUNCIONARIOS (
  -> id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> ultimo_nome varchar(150),
  -> primeiro_nome varchar(150),
  -> extensao varchar(40),
  -> email varchar(150),
  -> id_escritorio int,
  -> reportar_para int,
  -> titulo_trabalho varchar(100),
  -> FOREIGN KEY(id_escritorio) REFERENCES TB_ESCRITORIOS(id)
  -> );
Query OK, 0 rows affected (0.03 sec)
```

- Criação tabela TB_CLIENTES

```
mysql> CREATE TABLE TB_CLIENTES (
  -> id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> nome varchar(150),
  -> ultimo_nome varchar(150),
  -> primeiro_nome varchar(150),
  -> telefone text,
  -> endereco_l1 text,
  -> endereco_l2 text,
  -> cidade varchar(40),
  -> estado varchar(40),
  -> codigo_postal varchar(20),
  -> pais varchar(40),
  -> id_funcionario int,
  -> limite_credito int,
  -> FOREIGN KEY(id_funcionario) REFERENCES TB_FUNCIONARIOS(id)
  -> );
Query OK, 0 rows affected (0.02 sec)
```

- Criação tabela TB_LINHA_PRODUTOS

```
mysql> CREATE TABLE TB_LINHA_PRODUTOS (
  -> id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> descricao text,
  -> descricao_html text,
  -> image text
  -> );
Query OK, 0 rows affected (0.02 sec)
```

- Criação tabela TB_PEDIDOS

```
mysql> CREATE TABLE TB_PEDIDOS (
  -> id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> dataPedido date,
  -> dataEntrega date,
  -> dataEnviado date,
  -> status text,
  -> coments text,
  -> cliente_id int,
  -> FOREIGN KEY(cliente_id) REFERENCES TB_CLIENTES(id)
  -> );
Query OK, 0 rows affected (0.02 sec)
```

- Criação tabela TB_PAGAMENTOS

```
mysql> CREATE TABLE TB_PAGAMENTOS (
  -> id int NOT NULL AUTO_INCREMENT,
  -> cliente_id int NOT NULL,
  -> pagamento_data date,
  -> valor int,
  -> PRIMARY KEY(id,cliente_id),
  -> FOREIGN KEY(cliente_id) REFERENCES TB_CLIENTES(id)
  -> );
Query OK, 0 rows affected (0.03 sec)
```

- Criação tabela TB_PRODUTOS

```
mysql> CREATE TABLE TB_PRODUTOS (
  -> id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  -> nome_produto text,
  -> linhaProduto_id int,
  -> escala text,
  -> fornecedor text,
  -> descricao text,
  -> qtd_estoque int,
  -> preco int,
  -> msrp int,
  -> FOREIGN KEY(linhaProduto_id) REFERENCES TB_LINHA_PRODUTOS(id)
  -> );
Query OK, 0 rows affected (0.02 sec)
```

- Criação tabela TB_DETALHES_PEDIDOS

```
mysql> CREATE TABLE TB_DETALHES_PEDIDOS (
->  pedido_id int NOT NULL,
->  produto_id int NOT NULL,
->  qtd_pedido int,
->  preco_unitario int,
->  numero_linha_pedido int,
->  PRIMARY KEY(pedido_id,produto_id),
->  FOREIGN KEY(pedido_id) REFERENCES TB_PEDIDOS(id),
->  FOREIGN KEY(produto_id) REFERENCES TB_PRODUTOS(id)
-> );
Query OK, 0 rows affected (0.02 sec)
```

- TABELAS CRIADAS

```
mysql> SHOW TABLES
-> ;
+-----+
| Tables_in_BD_AVIOES |
+-----+
| TB_CLIENTES          |
| TB_DETALHES_PEDIDOS  |
| TB_ESCRITORIOS       |
| TB_FUNCIONARIOS      |
| TB_LINHA_PRODUTOS    |
| TB_PAGAMENTOS        |
| TB_PEDIDOS           |
| TB_PRODUTOS          |
+-----+
8 rows in set (0.00 sec)
```

Em uma segunda instância criamos um ambiente virtual python, conecta a outra instância onde tem as tabelas e cria um arquivo python

```
[node1] (local) root@192.168.0.18 ~
$ python -m venv myenv
[node1] (local) root@192.168.0.18 ~
$ source myenv/bin/activate
(myenv) [node1] (local) root@192.168.0.18 ~
$ pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.2.0-py2.py3-none-any.whl.metadata (6.0 kB)
  Downloading mysql_connector_python-9.2.0-py2.py3-none-any.whl (398 kB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.2.0

[notice] A new release of pip is available: 24.2 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
(myenv) [node1] (local) root@192.168.0.18 ~

(myenv) [node1] (local) root@192.168.0.18 ~
$ vi app.py
```

Com o comando python app.py

```
$ python app.py
Conexão com o MySQL bem-sucedida
Cliente inserido com sucesso
Cliente inserido com sucesso
Clientes:
(1, 'Maria', 'Silva', 'Maria', '123456789', 'maria@email.com', 'Rua A, 123', 'Apto 2', 'Centro', 'SP', 'Brasil')
(2, 'João', 'Pereira', 'João', '987654321', 'joao@email.com', 'Rua B, 456', 'Casa', 'Bairro X', 'RJ', 'Brasil')

Cliente atualizado com sucesso
Cliente deletado com sucesso
Detalhe do pedido inserido com sucesso
Detalhe do pedido inserido com sucesso
Detalhes de Pedidos:
(1, 1, 1, 10, 5.50)
(2, 2, 3, 2, 12.90) []
```

```
Detalhe do pedido atualizado com sucesso
Detalhe do pedido deletado com sucesso
Funcionário inserido com sucesso
Funcionário inserido com sucesso
Funcionários:
(1, 'Silva', 'Carlos', '1234', 'Gerente', 'SP', '987654321')
(2, 'Santos', 'Ana', '5678', 'Vendedora', 'RJ', '123456789')
Funcionário atualizado com sucesso
Funcionário deletado com sucesso
Escritório inserido com sucesso
Escritório inserido com sucesso
Escritórios:
(1, 'São Paulo', '18799090', 'Avenida Paulista, 1000', 'SP', 'Brasil')
(2, 'Ceará', '987659321', 'Rua das Flores, 200', 'CE', 'Brasil')
Escritório atualizado com sucesso
Escritório deletado com sucesso
Linha de produto inserida com sucesso
Linha de produto inserida com sucesso
Linha de produto inserida com sucesso
```

```
Linha de produto inserida com sucesso
Linhas de Produtos:
(1, 'Canetas', '<p>Canetas de diversos tipos e marcas</p>', 'imagem_canetas.jpg')
(2, 'Utensílios de cozinha', '<p>Produtos de cozinha variados</p>', 'imagem_cozinha.jpg')
Linha de produto atualizada com sucesso
Linha de produto deletada com sucesso
Produto inserido com sucesso
Produto inserido com sucesso
Produtos:
(1, 'Caneta azul', 'Uma caneta esferográfica azul', 10, 5.50, 'imagem_caneta.jpg', 1)
(2, 'Faqueiro inox', 'Conjunto de talheres em inox', 5, 99.90, 'imagem_faqueiro.jpg', 2)
Produto atualizado com sucesso
Produto deletado com sucesso
Pedido inserido com sucesso
Pedido inserido com sucesso
Pedidos:
(1, 1, '2025-02-24', '2025-02-28', 'Entregue')
(2, 2, '2025-02-20', '2025-02-25', 'Em andamento')
```

```
Pedido atualizado com sucesso
Pedido deletado com sucesso
Pagamento inserido com sucesso
Pagamento inserido com sucesso
Pagamentos:
(1, 1, '2025-02-24', 55.00)
(2, 2, '2025-02-20', 199.80)
Pagamento atualizado com sucesso
Pagamento deletado com sucesso
```

- criação e edição do arquivo app.py

```

1 import mysql.connector
2 from mysql.connector import Error
3
4 def create_connection():
5     """Cria uma conexão com o banco de dados MySQL."""
6     connection = None
7     try:
8         connection = mysql.connector.connect(
9             host='192.168.0.28',
10            user='root',
11            password='root',
12            database='BD_AVIOES'
13        )
14        print("Conexão com o MySQL bem-sucedida")
15    except Error as e:
16        print(f"Erro '{e}' ocorreu")
17    return connection
18

```

```

20 def insert_cliente(connection, nome, ultimo_nome, primeiro_nome, telefone, endereco_l1, endereco_l2, cidade,
21 estado, codigo_postal, pais, id_funcionario, limite_credito):
22     cursor = connection.cursor()
23     query = """
24     INSERT INTO TB_CLIENTES (nome, ultimo_nome, primeiro_nome, telefone, endereco_l1, endereco_l2, cidade, es
25 tado, codigo_postal, pais, id_funcionario, limite_credito)
26     VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
27 """
28     cursor.execute(query, (nome, ultimo_nome, primeiro_nome, telefone, endereco_l1, endereco_l2, cidade, esta
29 do, codigo_postal, pais, id_funcionario, limite_credito))
30     connection.commit()
31     print("Cliente inserido com sucesso")
32
33 def read_clientes(connection):
34     cursor = connection.cursor()
35     cursor.execute("SELECT * FROM TB_CLIENTES")
36     clientes = cursor.fetchall()
37     for cliente in clientes:

```

34, 28 48

```

37 def update_cliente(connection, cliente_id, nome, ultimo_nome, primeiro_nome, telefone, endereco_l1, endereco
38 l2, cidade, estado, codigo_postal, pais, id_funcionario, limite_credito):
39     cursor = connection.cursor()
40     query = """
41     UPDATE TB_CLIENTES
42     SET nome = %s, ultimo_nome = %s, primeiro_nome = %s, telefone = %s, endereco_l1 = %s, endereco_l2 = %s,
43 idade = %s, estado = %s, codigo_postal = %s, pais = %s, id_funcionario = %s, limite_credito = %s
44     WHERE id = %s
45 """
46     cursor.execute(query, (nome, ultimo_nome, primeiro_nome, telefone, endereco_l1, endereco_l2, cidade, est
47 do, codigo_postal, pais, id_funcionario, limite_credito, cliente_id))
48     connection.commit()
49     print("Cliente atualizado com sucesso")
50
51 def delete_cliente(connection, cliente_id):
52     cursor = connection.cursor()
53     query = "DELETE FROM TB_CLIENTES WHERE id = %s"
54     cursor.execute(query, (cliente_id,))

```

51, 30 88

```

50 query = "DELETE FROM TB_CLIENTES WHERE id = %s"
51 cursor.execute(query, (cliente_id,))
52 connection.commit()
53 print("Cliente deletado com sucesso")
54
55 # Funções CRUD para TB_DETALHES_PEDIDOS
56 def create_detalhe_pedido(connection, pedido_id, produto_id, qtd_pedido, preco_unitario, numero_linha_pedido)
57 :
58     cursor = connection.cursor()
59     query = """
60     INSERT INTO TB_DETALHES_PEDIDOS (pedido_id, produto_id, qtd_pedido, preco_unitario, numero_linha_pedido)
61     VALUES (%s, %s, %s, %s, %s)
62     """
63     cursor.execute(query, (pedido_id, produto_id, qtd_pedido, preco_unitario, numero_linha_pedido))
64     connection.commit()
65     print("Detalhe do pedido inserido com sucesso")
66
67 def read_detalhe_pedido(connection):

```

```

68     cursor.execute("SELECT * FROM TB_DETALHES_PEDIDOS")
69     detalhes = cursor.fetchall()
70     for detalhe in detalhes:
71         print(detalhe)
72
73 def update_detalhe_pedido(connection, pedido_id, produto_id, qtd_pedido):
74     cursor = connection.cursor()
75     query = """
76     UPDATE TB_DETALHES_PEDIDOS
77     SET qtd_pedido = %s
78     WHERE pedido_id = %s AND produto_id = %s
79     """
80     cursor.execute(query, (qtd_pedido, pedido_id, produto_id))
81     connection.commit()
82     print("Detalhe do pedido atualizado com sucesso")
83
84 def delete_detalhe_pedido(connection, pedido_id, produto_id):
85     cursor = connection.cursor()

```

```

95     INSERT INTO TB_FUNCIONARIOS (ultimo_nome, primeiro_nome, extensao, email, id_escritorio, reportar_para, t
96     itulo_trabalho)
97     VALUES (%s, %s, %s, %s, %s, %s, %s)
98     """
99     cursor.execute(query, (ultimo_nome, primeiro_nome, extensao, email, id_escritorio, reportar_para, titulo
100     trabalho))
101     connection.commit()
102     print("Funcionário inserido com sucesso")
103
104 def read_funcionarios(connection):
105     cursor = connection.cursor()
106     cursor.execute("SELECT * FROM TB_FUNCIONARIOS")
107     funcionarios = cursor.fetchall()
108     for funcionario in funcionarios:
109         print(funcionario)
110
111 def update_funcionario(connection, funcionario_id, ultimo_nome, primeiro_nome, extensao, email, id_escritorio
112 , reportar_para, titulo_trabalho):

```

```

113     SET ultimo_nome = %s, primeiro_nome = %s, extensao = %s, email = %s, id_escritorio = %s, reportar_para =
114     %s, titulo_trabalho = %s
115     WHERE id = %s
116     """
117     cursor.execute(query, (ultimo_nome, primeiro_nome, extensao, email, id_escritorio, reportar_para, titulo
118     trabalho, funcionario_id))
119     connection.commit()
120     print("Funcionário atualizado com sucesso")
121
122 def delete_funcionario(connection, funcionario_id):
123     cursor = connection.cursor()
124     query = "DELETE FROM TB_FUNCIONARIOS WHERE id = %s"
125     cursor.execute(query, (funcionario_id,))
126     connection.commit()
127     print("Funcionário deletado com sucesso")
128
129 # Funções CRUD para TB_ESCRITORIOS
130 def create_escritorio(connection, cidade, telefone, endereco_11, endereco_12, estado, pais, territorio):

```

```

128 def create_escritorio(connection, cidade, telefone, endereco_l1, endereco_l2, estado, pais, territorio):
129     cursor = connection.cursor()
130     query = """
131     INSERT INTO TB_ESCRITORIOS (cidade, telefone, endereco_l1, endereco_l2, estado, pais, territorio)
132     VALUES (%s, %s, %s, %s, %s, %s, %s)
133     """
134     cursor.execute(query, (cidade, telefone, endereco_l1, endereco_l2, estado, pais, territorio))
135     connection.commit()
136     print("Escritório inserido com sucesso")
137
138 def read_escritorios(connection):
139     cursor = connection.cursor()
140     cursor.execute("SELECT * FROM TB_ESCRITORIOS")
141     escritorios = cursor.fetchall()
142     for escritorio in escritorios:
143         print(escritorio)
144
145 def update_escritorio(connection, escritorio_id, cidade, telefone, endereco_l1, endereco_l2, estado, pais, te
rritorio):
146     cursor = connection.cursor()
147     query = """
148     UPDATE TB_ESCRITORIOS
149     SET cidade = %s, telefone = %s, endereco_l1 = %s, endereco_l2 = %s, estado = %s, pais = %s, territorio =
    %s
150     WHERE id = %s
151     """
152     cursor.execute(query, (cidade, telefone, endereco_l1, endereco_l2, estado, pais, territorio, escritorio_id))
153     connection.commit()
154     print("Escritório atualizado com sucesso")
155
156 def delete_escritorio(connection, escritorio_id):
157     cursor = connection.cursor()
158     query = "DELETE FROM TB_ESCRITORIOS WHERE id = %s"
159     cursor.execute(query, (escritorio_id,))

```

```

164 def create_linha_produto(connection, descricao, descricao_html, image):
165     cursor = connection.cursor()
166     query = """
167     INSERT INTO TB_LINHA_PRODUTOS (descricao, descricao_html, image)
168     VALUES (%s, %s, %s)
169     """
170     cursor.execute(query, (descricao, descricao_html, image))
171     connection.commit()
172     print("Linha de produto inserida com sucesso")
173
174 def read_linhas_produto(connection):
175     cursor = connection.cursor()
176     cursor.execute("SELECT * FROM TB_LINHA_PRODUTOS")
177     linhas_produto = cursor.fetchall()
178     for linha in linhas_produto:
179         print(linha)
180
181 def update_linha_produto(connection, linha_produto_id, descricao_html, image):

```



```

283     cursor = connection.cursor()
284     cursor.execute("SELECT * FROM TB_PAGAMENTOS")
285     pagamentos = cursor.fetchall()
286     for pagamento in pagamentos:
287         print(pagamento)
288
289 def update_pagamento(connection, pagamento_id, pagamento_data, valor):
290     cursor = connection.cursor()
291     query = """
292     UPDATE TB_PAGAMENTOS
293     SET pagamento_data = %s, valor = %s
294     WHERE id = %s
295     """
296     cursor.execute(query, (pagamento_data, valor, pagamento_id))
297     connection.commit()
298     print("Pagamento atualizado com sucesso")
299
300 def delete_pagamento(connection, pagamento_id):

```

```

264 def delete_pedido(connection, pedido_id):
265     cursor = connection.cursor()
266     query = "DELETE FROM TB_PEDIDOS WHERE id = %s"
267     cursor.execute(query, (pedido_id,))
268     connection.commit()
269     print("Pedido deletado com sucesso")
270
271 # Funções CRUD para TB_PAGAMENTOS
272 def create_pagamento(connection, cliente_id, pagamento_data, valor):
273     cursor = connection.cursor()
274     query = """
275     INSERT INTO TB_PAGAMENTOS (cliente_id, pagamento_data, valor)
276     VALUES (%s, %s, %s)
277     """
278     cursor.execute(query, (cliente_id, pagamento_data, valor))
279     connection.commit()
280     print("Pagamento inserido com sucesso")
281

```

```

246 def read_pedidos(connection):
247     cursor = connection.cursor()
248     cursor.execute("SELECT * FROM TB_PEDIDOS")
249     pedidos = cursor.fetchall()
250     for pedido in pedidos:
251         print(pedido)
252
253 def update_pedido(connection, pedido_id, status, comentarios):
254     cursor = connection.cursor()
255     query = """
256     UPDATE TB_PEDIDOS
257     SET status = %s, comentarios = %s
258     WHERE id = %s
259     """
260     cursor.execute(query, (status, comentarios, pedido_id))
261     connection.commit()
262     print("Pedido atualizado com sucesso")
263
264 def delete_produto(connection, produto_id):
265     cursor = connection.cursor()
266     query = "DELETE FROM TB_PRODUTOS WHERE id = %s"
267     cursor.execute(query, (produto_id,))
268     connection.commit()
269     print("Produto deletado com sucesso")
270
271 # Funções CRUD para TB_PEDIDOS
272 def create_pedido(connection, cliente_id, dataPedido, dataEntrega, dataEnviado, status, comentarios):
273     cursor = connection.cursor()
274     query = """
275     INSERT INTO TB_PEDIDOS (cliente_id, dataPedido, dataEntrega, dataEnviado, status, comentarios)
276     VALUES (%s, %s, %s, %s, %s, %s)
277     """
278     cursor.execute(query, (cliente_id, dataPedido, dataEntrega, dataEnviado, status, comentarios))
279     connection.commit()
280     print("Pedido inserido com sucesso")

```

- Uso das funções CRUD em todas as tabelas

```

313 # Clientes
314 insert_cliente(connection, 'Maria', 'Silva', 'Maria', '123456789', 'Rua A', 'Apt 1', 'São Paulo', 'SP', '01234-567', 'Brasil', 1, 5000)
315 insert_cliente(connection, 'João', 'Pereira', 'João', '987654321', 'Rua B', 'Apt 2', 'Rio de Janeiro', 'RJ', '98765-432', 'Brasil', 2, 3000)
316 print("Clientes:")
317 read_clientes(connection)
318 update_cliente(connection, 1, 'Maria', 'Silva', 'Maria', '123456789', 'Rua A', 'Apt 1', 'São Paulo', 'SP', '01234-567', 'Brasil', 1, 5500)
319 delete_cliente(connection, 2)
320
321 # Detalhes de Pedido
322 create_detalhe_pedido(connection, 1, 198, 5, 100.0, 1)
323 create_detalhe_pedido(connection, 1, 102, 3, 150.0, 2)
324 print("Detalhes de Pedidos:")
325 read_detalhe_pedido(connection)
326 update_detalhe_pedido(connection, 1, 102, 4)
327 delete_detalhe_pedido(connection, 1, 198)

```

```

320
321     # Detalhes de Pedido
322     create_detalhe_pedido(connection, 1, 198, 5, 100.0, 1)
323     create_detalhe_pedido(connection, 1, 102, 3, 150.0, 2)
324     print("Detalhes de Pedidos:")
325     read_detalhe_pedido(connection)
326     update_detalhe_pedido(connection, 1, 102, 4)
327     delete_detalhe_pedido(connection, 1, 198)
328
329     # Funcionários
330     create_funcionario(connection, 'Silva', 'Carlos', '1234', 'carlos@empresa.com', 1, 2, 'Gerente')
331     create_funcionario(connection, 'Santos', 'Ana', '5678', 'ana@empresa.com', 2, 3, 'Assistente de gerente')
332     print("Funcionários:")
333     read_funcionarios(connection)
334     update_funcionario(connection, 1, 'Silva', 'Carlos', '2345', 'carlos@empresa.com', 1, 3, 'Diretor de vend
as')
335     delete_funcionario(connection, 2)
336

```

```

337     # Escritórios
338     create_escritorio(connection, 'São Paulo', '18799090', 'Av. Paulista, 1000', 'Bloco A', 'SP', 'Brasil', '
SP')
339     create_escritorio(connection, 'Ceará', '987659321', 'Av. Rio Branco, 2000', 'Bloco B', 'CE', 'Brasil', 'C
E')
340     print("Escritórios:")
341     read_escritorios(connection)
342     update_escritorio(connection, 1, 'São Paulo', '1102339455', 'Av. Paulista, 1100', 'Bloco C', 'SP', 'Brasi
l', 'SP')
343     delete_escritorio(connection, 2)
344
345     # Linha de Produtos
346     create_linha_produto(connection, 'Canetas', '<p>Canetas de diversos tipos e marcas</p>', 'canetasTM.jpg')
347     create_linha_produto(connection, 'Utensilios de cozinha', '<p>Utensilios de cozinha</p>', 'cozinha.jpg')
348     print("Linhas de Produtos:")
349     read_linhas_produto(connection)
350     update_linha_produto(connection, 1, '<p>Canetas de diversos tipos, marcas e cores</p>', 'canetasTMC.jpg')
351     delete_linha_produto(connection, 2)

```

```

344
345     # Linha de Produtos
346     create_linha_produto(connection, 'Canetas', '<p>Canetas de diversos tipos e marcas</p>', 'canetasTM.jpg')
347     create_linha_produto(connection, 'Utensilios de cozinha', '<p>Utensilios de cozinha</p>', 'cozinha.jpg')
348     print("Linhas de Produtos:")
349     read_linhas_produto(connection)
350     update_linha_produto(connection, 1, '<p>Canetas de diversos tipos, marcas e cores</p>', 'canetasTMC.jpg')
351     delete_linha_produto(connection, 2)
352
353     # Produtos
354     create_produto(connection, 'Caneta azul', 1, '1:10', 'Bic', 'Caneta azul ponta 0.7', 50, 6.5, 4.0)
355     create_produto(connection, 'faqueiro inox', 2, '1:10', 'Itatiaia', 'kit faqueiro inox', 100, 150.0, 200.0)
356     print("Produtos:")
357     read_produtos(connection)
358     update_produto(connection, 1, 'Caneta em gel azul 0.7', 15.0)
359     delete_produto(connection, 2)
360

```

```

353     # Produtos
354     create_produto(connection, 'Caneta azul', 1, '1:10', 'Bic', 'Caneta azul ponta 0.7', 50, 6.5, 4.0)
355     create_produto(connection, 'faqueiro inox', 2, '1:10', 'Itatiaia', 'kit faqueiro inox', 100, 150.0, 200.0)
356     print("Produtos:")
357     read_produtos(connection)
358     update_produto(connection, 1, 'Caneta em gel azul 0.7', 15.0)
359     delete_produto(connection, 2)
360
361     # Pedidos
362     create_pedido(connection, 1, '2025-01-20', '2025-02-20', '2025-01-21', 'Enviado', 'Pedido frágil')
363     create_pedido(connection, 2, '2025-01-21', '2025-02-21', '2025-01-22', 'Pagamento Pendente', 'Aguardando
pagamento')
364     print("Pedidos:")
365     read_pedidos(connection)
366     update_pedido(connection, 1, 'Enviado', 'Pedido enviado com sucesso')
367     delete_pedido(connection, 2)
368

```

```

369     # Pagamentos
370     create_pagamento(connection, 1, '2025-01-22', 50.0)
371     create_pagamento(connection, 2, '2025-02-17', 30.0)
372     print("Pagamentos:")
373     read_pagamentos(connection)
374     update_pagamento(connection, 1, '2025-01-23', 55.0)
375     delete_pagamento(connection, 2)
376
377     if connection:
378         connection.close()
379         print("Conexão com o MySQL fechada")
380
381 if __name__ == "__main__":
382     main()

```