

Atividade - 08

Alunos: Pedro Henrique Alves Barbosa & Gustavo Fontenele Coêlho

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

Tarefa em dupla postado no GitHub de cada participante com o link do Github referenciado no Google Classroom.

1ª Instância)

```
#####
#                                     #
#      WARNING!!!!                   #
# This is a sandbox environment. Using personal credentials   #
# is HIGHLY! discouraged. Any consequences of doing so are   #
# completely the user's responsibilities.                       #
# The PWD team.                                               #
#####
[node1] (local) root@192.168.0.28 ~
$ docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=BD_LOJA -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
2be0d473cacf: Pull complete
f56a22f949f9: Pull complete
277ab5f6ddde: Pull complete
df1balac457a: Pull complete
cc9646b08259: Pull complete
893b018337e2: Pull complete
Digest: sha256:146682692a3aa409eae7b7dc6a30f637c6cb49b6ca901c2cd160becc81127d3b
Status: Downloaded newer image for mysql:latest
bebe9a13d620c0cd4bd033170f6bae31d3e1caae735576af649ca25a3449b853
[node1] (local) root@192.168.0.28 ~
$ 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
```

```
mysql> USE BD_LOJA
Database changed
mysql> USE BD_LOJA
Database changed
mysql> CREATE TABLE TB_ESCRITORIOS (
->   id int NOT NULL AUTO_INCREMENT,
->   cidade text,
->   phone text,
->   endereco_pt1 text,
->   endereco_pt2 text,
->   estado text,
->   pais text,
->   codigo_postal text,
->   territorio text,
->   PRIMARY KEY (id)
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE TB_FUNCIONARIOS (
->   id int NOT NULL AUTO_INCREMENT,
->   nome_ultimo varchar(100),
->   nome_primeiro varchar(100),
->   extensao varchar(10),
->   email varchar(255),
->   escritorio_id int,
->   relatorios_para_funcionario_id int,
->   trabalho varchar(100),
->   PRIMARY KEY (id),
->   FOREIGN KEY (escritorio_id) REFERENCES TB_ESCRITORIOS (id),
->   FOREIGN KEY (relatorios_para_funcionario_id) REFERENCES TB_FUNCIONARIOS (id)
-> );
```

```
mysql> CREATE TABLE TB_CLIENTES (  
-> id int NOT NULL AUTO_INCREMENT,  
-> nome text,  
-> nome_ultimo varchar(100),  
-> nome_primeiro varchar(100),  
-> telefone text,  
-> endereco_pt1 varchar(255),  
-> endereco_pt2 varchar(255),  
-> cidade varchar(50),  
-> estado varchar(50),  
-> codigo_postal varchar(20),  
-> pais varchar(50),  
-> funcionario_id int,  
-> limite_credito double,  
-> PRIMARY KEY (id),  
-> FOREIGN KEY (funcionario_id) REFERENCES TB_FUNCIONARIOS (id)  
-> );
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> CREATE TABLE TB_LINHAS_PRODUTOS (  
-> id int NOT NULL AUTO_INCREMENT,  
-> descricao text,  
-> descricao_html longtext,  
-> image text,  
-> PRIMARY KEY (id)  
-> );
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> CREATE TABLE TB_PRODUTOS (  
-> id int NOT NULL AUTO_INCREMENT,  
-> nome text,  
-> linha_produto_id int,  
-> escala text,  
-> fornecedor text,
```

```
-> fornecedor text,  
-> descricao text,  
-> quantidade_estoque int,  
-> preco double,  
-> msrp double,  
-> PRIMARY KEY (id),  
-> FOREIGN KEY (linha_produto_id) REFERENCES TB_LINHAS_PRODUTOS (id)  
-> );  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> CREATE TABLE TB_PEDIDOS (  
-> id int NOT NULL AUTO_INCREMENT,  
-> data_pedido date,  
-> data_entrega date,  
-> data_envio date,  
-> status text,  
-> comentarios text,  
-> cliente_id int,  
-> PRIMARY KEY (id),  
-> FOREIGN KEY (cliente_id) REFERENCES TB_CLIENTES (id)  
-> );  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> CREATE TABLE TB_DETALHES_PEDIDO (  
-> pedido_id int NOT NULL,  
-> produto_id int NOT NULL,  
-> quantidade_pedida int,  
-> preco_unitario double,  
-> 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 (0.01 sec)  
  
mysql> CREATE TABLE TB_PAGAMENTOS (  
-> id int NOT NULL AUTO_INCREMENT,  
-> data_pagamento date,  
-> valor double,  
-> PRIMARY KEY (id),  
-> FOREIGN KEY (pedido_id) REFERENCES TB_PEDIDOS (id),  
-> FOREIGN KEY (cliente_id) REFERENCES TB_CLIENTES (id)  
-> );  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE TB_DETALHES_PEDIDO (
->     pedido_id int NOT NULL,
->     produto_id int NOT NULL,
->     quantidade_pedida int,
->     preco_unitario double,
->     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 (0.01 sec)

mysql> CREATE TABLE TB_PAGAMENTOS (
->     id int NOT NULL AUTO_INCREMENT,
->     cliente_id int NOT NULL,
->     data_pagamento date,
->     valor double,
->     PRIMARY KEY (id,cliente_id),
->     FOREIGN KEY (cliente_id) REFERENCES TB_CLIENTES (id)
-> );
Query OK, 0 rows affected (0.01 sec)
```

2° instância)

```
1 import mysql.connector
2 from mysql.connector import Error
3 from datetime import date
4
5 def create_connection():
6     """Cria uma conexão com o banco de dados MySQL."""
7     connection = None
8     try:
9         connection = mysql.connector.connect(
10             host='192.168.0.28',
11             port='3306',
12             user='root',
13             password='root',
14             database='BD_LOJA'
15         )
16         print("Conexão com o MySQL bem-sucedida")
17     except Error as e:
18         print(f"Erro '{e}' ocorreu")
19     return connection
20
21 def create_cliente(connection, nome, nome_ultimo, nome_primeiro, telefone, endereco_pt1, endereco_pt2, cidade, estado, codigo_postal, pais, funcionario_id, limite_credito):
22     cursor = connection.cursor()
23     query = """INSERT INTO TB_CLIENTES (nome, nome_ultimo, nome_primeiro, telefone, endereco_pt1, endereco_pt2, cidade, estado, codigo_postal, pais, funcionario_id, limite_credito)
24     VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"""
25     values = (nome, nome_ultimo, nome_primeiro, telefone, endereco_pt1, endereco_pt2, cidade, estado, codigo_postal, pais, funcionario_id, limite_credito)
26     cursor.execute(query, values)
27     connection.commit()
28     print("Cliente adicionado com sucesso")
29
30 def read_clientes(connection):
31     cursor = connection.cursor()
32     cursor.execute("SELECT * FROM TB_CLIENTES")
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```

31 def read_clientes(connection):
32     cursor = connection.cursor()
33     cursor.execute("SELECT * FROM TB_CLIENTES")
34     return cursor.fetchall()
35
36 def update_cliente(connection, id, nome, nome_ultimo, nome_primeiro, telefone, endereco_pt1,
37     endereco_pt2, cidade, estado, codigo_postal, pais, funcionario_id, limite_credito):
38     cursor = connection.cursor()
39     query = """UPDATE TB_CLIENTES SET nome=%s, nome_ultimo=%s, nome_primeiro=%s,
40         telefone=%s, endereco_pt1=%s, endereco_pt2=%s, cidade=%s, estado=%s,
41         codigo_postal=%s, pais=%s, funcionario_id=%s, limite_credito=%s WHERE id=%s"""
42     values = (nome, nome_ultimo, nome_primeiro, telefone, endereco_pt1, endereco_pt2,
43         cidade, estado, codigo_postal, pais, funcionario_id, limite_credito, id)
44     cursor.execute(query, values)
45     connection.commit()
46     print("Cliente atualizado com sucesso")
47
48 def delete_cliente(connection, id):
49     cursor = connection.cursor()
50     cursor.execute("DELETE FROM TB_CLIENTES WHERE id=%s", (id,))
51     connection.commit()
52     print("Cliente deletado com sucesso")
53
54 def create_produto(connection, nome, linha_produto_id, escala, fornecedor, descricao,
55     quantidade_estoque, preco, msrp):
56     cursor = connection.cursor()
57     query = """INSERT INTO TB_PRODUTOS (nome, linha_produto_id, escala, fornecedor,
58         descricao, quantidade_estoque, preco, msrp)
59         VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"""
60     values = (nome, linha_produto_id, escala, fornecedor, descricao,
61         quantidade_estoque, preco, msrp)
62     cursor.execute(query, values)
63     connection.commit()
64     print("Produto adicionado com sucesso")

```

```

60     values = (nome, linha_produto_id, escala, fornecedor, descricao,
61         quantidade_estoque, preco, msrp)
62     cursor.execute(query, values)
63     connection.commit()
64     print("Produto adicionado com sucesso")
65
66 def read_produtos(connection):
67     cursor = connection.cursor()
68     cursor.execute("SELECT * FROM TB_PRODUTOS")
69     return cursor.fetchall()
70
71 def update_produto(connection, id, nome, linha_produto_id, escala, fornecedor,
72     descricao, quantidade_estoque, preco, msrp):
73     cursor = connection.cursor()
74     query = """UPDATE TB_PRODUTOS SET nome=%s, linha_produto_id=%s, escala=%s,
75         fornecedor=%s, descricao=%s, quantidade_estoque=%s, preco=%s, msrp=%s
76         WHERE id=%s"""
77     values = (nome, linha_produto_id, escala, fornecedor, descricao,
78         quantidade_estoque, preco, msrp, id)
79     cursor.execute(query, values)
80     connection.commit()
81     print("Produto atualizado com sucesso")
82
83 def delete_produto(connection, id):
84     cursor = connection.cursor()
85     cursor.execute("DELETE FROM TB_PRODUTOS WHERE id=%s", (id,))
86     connection.commit()
87     print("Produto deletado com sucesso")
88
89 def create_pedido(connection, data_pedido, data_entrega, data_envio, status,
90     comentarios, cliente_id):
91     cursor = connection.cursor()
92     query = """INSERT INTO TB_PEDIDOS (data_pedido, data_entrega, data_envio,
93         status, comentarios, cliente_id)
94     VALUES (%s, %s, %s, %s, %s, %s)"""

```

```

89 def create_pedido(connection, data_pedido, data_entrega, data_envio, status,
90                    comentarios, cliente_id):
91     cursor = connection.cursor()
92     query = """INSERT INTO TB_PEDIDOS (data_pedido, data_entrega, data_envio,
93                                     status, comentarios, cliente_id)
94             VALUES (%s, %s, %s, %s, %s, %s)"""
95     values = (data_pedido, data_entrega, data_envio, status, comentarios, cliente_id)
96     cursor.execute(query, values)
97     connection.commit()
98     print("Pedido adicionado com sucesso")
99
100 def update_pedido(connection, id, data_pedido, data_entrega, data_envio,
101                   status, comentarios, cliente_id):
102     cursor = connection.cursor()
103     query = """UPDATE TB_PEDIDOS SET data_pedido=%s, data_entrega=%s, data_envio=%s,
104                                     status=%s, comentarios=%s, cliente_id=%s WHERE id=%s"""
105     values = (data_pedido, data_entrega, data_envio, status, comentarios,
106             cliente_id, id)
107     cursor.execute(query, values)
108     connection.commit()
109     print("Pedido atualizado com sucesso")
110
111 def read_pedidos(connection):
112     cursor = connection.cursor()
113     cursor.execute("SELECT * FROM TB_PEDIDOS")
114     return cursor.fetchall()
115
116 def delete_pedido(connection, id):
117     cursor = connection.cursor()
118     cursor.execute("DELETE FROM TB_PEDIDOS WHERE id=%s", (id,))
119     connection.commit()
120     print("Pedido deletado com sucesso")
121
122 def create_detalhe_pedido(connection, pedido_id, produto_id, quantidade_pedida,
123                           preco_unitario, numero_linha_pedido):

```

```

119 connection.commit()
120 print("Pedido deletado com sucesso")
121
122 def create_detalhe_pedido(connection, pedido_id, produto_id, quantidade_pedida,
123                             preco_unitario, numero_linha_pedido):
124     cursor = connection.cursor()
125     query = """INSERT INTO TB_DETALHES_PEDIDO (pedido_id, produto_id,
126         quantidade_pedida, preco_unitario, numero_linha_pedido)
127         VALUES (%s, %s, %s, %s, %s)"""
128     values = (pedido_id, produto_id, quantidade_pedida, preco_unitario,
129         numero_linha_pedido)
130     cursor.execute(query, values)
131     connection.commit()
132     print("Detalhe do pedido adicionado com sucesso")
133
134 def read_detalhes_pedido(connection):
135     cursor = connection.cursor()
136     cursor.execute("SELECT * FROM TB_DETALHES_PEDIDO")
137     return cursor.fetchall()
138
139 def update_detalhe_pedido(connection, pedido_id, produto_id, quantidade_pedida,
140                             preco_unitario, numero_linha_pedido):
141     cursor = connection.cursor()
142     query = """UPDATE TB_DETALHES_PEDIDO SET quantidade_pedida=%s,
143         preco_unitario=%s, numero_linha_pedido=%s
144         WHERE pedido_id=%s AND produto_id=%s"""
145     values = (quantidade_pedida, preco_unitario, numero_linha_pedido,
146         pedido_id, produto_id)
147     cursor.execute(query, values)
148     connection.commit()
149     print("Detalhe do pedido atualizado com sucesso")
150
151 def delete_detalhes_pedido(connection, pedido_id, produto_id):
152     cursor = connection.cursor()
153     cursor.execute("""DELETE FROM TB_DETALHES_PEDIDO

```



```

154         WHERE pedido_id=%s AND produto_id=%s""", (pedido_id, produto_id))
155     connection.commit()
156     print("Detalhe do pedido deletado com sucesso")
157
158 def create_pagamento(connection, cliente_id, data_pagamento, valor):
159     cursor = connection.cursor()
160     query = """INSERT INTO TB_PAGAMENTOS (cliente_id, data_pagamento, valor)
161             VALUES (%s, %s, %s)"""
162     cursor.execute(query, (cliente_id, data_pagamento, valor))
163     connection.commit()
164     print("Pagamento adicionado com sucesso")
165
166 def read_pagamentos(connection):
167     cursor = connection.cursor()
168     cursor.execute("SELECT * FROM TB_PAGAMENTOS")
169     return cursor.fetchall()
170
171 def update_pagamento(connection, id, cliente_id, data_pagamento, valor):
172     cursor = connection.cursor()
173     query = """UPDATE TB_PAGAMENTOS SET data_pagamento=%s, valor=%s
174             WHERE id=%s AND cliente_id=%s"""
175     cursor.execute(query, (data_pagamento, valor, id, cliente_id))
176     connection.commit()
177     print("Pagamento atualizado com sucesso")
178
179 def delete_pagamento(connection, id, cliente_id):
180     cursor = connection.cursor()
181     cursor.execute("""DELETE FROM TB_PAGAMENTOS
182             WHERE id=%s AND cliente_id=%s""", (id, cliente_id))
183     connection.commit()
184     print("Pagamento deletado com sucesso")
185
186 def create_funcionario(connection, nome_ultimo, nome_primeiro, extensao, email,
187     escritorio_id, relatorios_para_funcionario_id, trabalho):
188     cursor = connection.cursor()

```

```

183     connection.commit()
184     print("Pagamento deletado com sucesso")
185
186 def create_funcionario(connection, nome_ultimo, nome_primeiro, extensao, email,
187     escritorio_id, relatorios_para_funcionario_id, trabalho):
188     cursor = connection.cursor()
189     query = """INSERT INTO TB_FUNCIONARIOS (nome_ultimo, nome_primeiro, extensao,
190     email, escritorio_id, relatorios_para_funcionario_id, trabalho)
191     VALUES (%s, %s, %s, %s, %s, %s, %s)"""
192     values = (nome_ultimo, nome_primeiro, extensao, email, escritorio_id,
193     relatorios_para_funcionario_id, trabalho)
194     cursor.execute(query, values)
195     connection.commit()
196     print("Funcionário adicionado com sucesso")
197
198 def read_funcionarios(connection):
199     cursor = connection.cursor()
200     cursor.execute("SELECT * FROM TB_FUNCIONARIOS")
201     return cursor.fetchall()
202
203 def update_funcionario(connection, id, nome_ultimo, nome_primeiro, extensao,
204     email, escritorio_id, relatorios_para_funcionario_id, trabalho):
205     cursor = connection.cursor()
206     query = """UPDATE TB_FUNCIONARIOS SET nome_ultimo=%s, nome_primeiro=%s,
207     extensao=%s, email=%s, escritorio_id=%s,
208     relatorios_para_funcionario_id=%s, trabalho=%s WHERE id=%s"""
209     values = (nome_ultimo, nome_primeiro, extensao, email, escritorio_id,
210     relatorios_para_funcionario_id, trabalho, id)
211     cursor.execute(query, values)
212     connection.commit()
213     print("Funcionário atualizado com sucesso")
214
215 def delete_funcionario(connection, id):
216     cursor = connection.cursor()
217     cursor.execute("DELETE FROM TB_FUNCIONARIOS WHERE id=%s", (id,))

```

```

213     print("Funcionário atualizado com sucesso")
214
215 def delete_funcionario(connection, id):
216     cursor = connection.cursor()
217     cursor.execute("DELETE FROM TB_FUNCIONARIOS WHERE id=%s", (id,))
218     connection.commit()
219     print("Funcionário deletado com sucesso")
220
221 def create_escritorio(connection, cidade, phone, endereco_pt1, endereco_pt2,
222                       estado, pais, codigo_postal, territorio):
223     cursor = connection.cursor()
224     query = """INSERT INTO TB_ESCRITORIOS (cidade, phone, endereco_pt1,
225                                           endereco_pt2, estado, pais, codigo_postal, territorio)
226             VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"""
227     values = (cidade, phone, endereco_pt1, endereco_pt2, estado, pais,
228              codigo_postal, territorio)
229     cursor.execute(query, values)
230     connection.commit()
231     print("Escritório adicionado com sucesso")
232
233 def read_escritorios(connection):
234     cursor = connection.cursor()
235     cursor.execute("SELECT * FROM TB_ESCRITORIOS")
236     return cursor.fetchall()
237
238 def update_escritorio(connection, id, cidade, phone, endereco_pt1, endereco_pt2,
239                      estado, pais, codigo_postal, territorio):
240     cursor = connection.cursor()
241     query = """UPDATE TB_ESCRITORIOS SET cidade=%s, phone=%s, endereco_pt1=%s,
242                                           endereco_pt2=%s, estado=%s, pais=%s, codigo_postal=%s,
243                                           territorio=%s WHERE id=%s"""
244     values = (cidade, phone, endereco_pt1, endereco_pt2, estado, pais,
245              codigo_postal, territorio, id)
246     cursor.execute(query, values)
247     connection.commit()

```

```

254     print("Escritório deletado com sucesso")
255
256 def create_linha_produto(connection, descricao, descricao_html, image):
257     cursor = connection.cursor()
258     query = """INSERT INTO TB_LINHAS_PRODUTOS (descricao, descricao_html, image)
259             VALUES (%s, %s, %s)"""
260     cursor.execute(query, (descricao, descricao_html, image))
261     connection.commit()
262     print("Linha de produto adicionada com sucesso")
263
264 def read_linhas_produtos(connection):
265     cursor = connection.cursor()
266     cursor.execute("SELECT * FROM TB_LINHAS_PRODUTOS")
267     return cursor.fetchall()
268
269 def update_linha_produto(connection, id, descricao, descricao_html, image):
270     cursor = connection.cursor()
271     query = """UPDATE TB_LINHAS_PRODUTOS SET descricao=%s, descricao_html=%s,
272             image=%s WHERE id=%s"""
273     cursor.execute(query, (descricao, descricao_html, image, id))
274     connection.commit()
275     print("Linha de produto atualizada com sucesso")
276
277 def delete_linha_produto(connection, id):
278     cursor = connection.cursor()
279     cursor.execute("DELETE FROM TB_LINHAS_PRODUTOS WHERE id=%s", (id,))
280     connection.commit()
281     print("Linha de produto deletada com sucesso")
282
283
284
285 def main():
286     connection = create_connection()
287     if connection is None:
288         return

```

```

def main():
    connection = create_connection()
    if connection is None:
        return

    create_escritorio(connection, "Fortaleza", "0123456789", "Santos Dumont", "1234", "CE", "Brasil", "012345678", "Nordeste")
    create_escritorio(connection, "Fortaleza", "218888888888", "Av Atlântica", "2000", "CE", "Brasil", "020000000", "Nordeste")

    create_funcionario(connection, "Silveira", "Jonas", "101", "jonas@email.com", 1, None, "Gerente")
    create_funcionario(connection, "Santola", "Mariana", "103", "mariana@email.com", 2, 1, "Vendedor")

    create_cliente(connection, "José Pereira", "Pereira", "José", "11999999999", "Rua A", "123", "São Paulo", "SP", "010000000", "Brasil", 1, 5000.00)
    create_cliente(connection, "Ana Santos", "Santos", "Ana", "21888888888", "Rua B", "456", "Rio de Janeiro", "RJ", "020000000", "Brasil", 2, 3000.00)

    create_linha_produto(connection, "Eletrônicos", "<p>Produtos eletrônicos</p>", "eletronicos.jpg")
    create_linha_produto(connection, "Móveis", "<p>Móveis para casa</p>", "moveis.jpg")

    create_produto(connection, "Smartphone X", 1, "1:1", "TechCorp", "Smartphone avançado", 100, 1999.99, 2499.99)
    create_produto(connection, "Sofá Confort", 2, "1:1", "MoveisLux", "Sofá 3 lugares", 50, 2999.99, 3499.99)

    create_pedido(connection, date(2024, 2, 1), date(2024, 2, 10), date(2024, 2, 5), "Entregue", "Entrega normal", 1)
    create_pedido(connection, date(2024, 2, 2), date(2024, 2, 11), date(2024, 2, 6), "Entregue", "Entrega expressa", 2)

    create_detalhe_pedido(connection, 1, 1, 2, 1999.99, 1)
    create_detalhe_pedido(connection, 2, 2, 1, 2999.99, 1)

    create_pagamento(connection, 1, date(2024, 2, 5), 3999.98)
    create_pagamento(connection, 2, date(2024, 2, 6), 2999.99)

    print("\nLeitura de todas as tabelas:")
    print("\nEscritórios:", read_escritorios(connection))

```

```

314 print("\nLeitura de todas as tabelas:")
315 print("\nEscritórios:", read_escritorios(connection))
316 print("\nFuncionários:", read_funcionarios(connection))
317 print("\nLinhas de Produtos:", read_linhas_produtos(connection))
318 print("\nProdutos:", read_produtos(connection))
319 print("\nClientes:", read_clientes(connection))
320 print("\nPedidos:", read_pedidos(connection))
321 print("\nDetalhes de Pedido:", read_detalhes_pedido(connection))
322 print("\nPagamentos:", read_pagamentos(connection))
323
324
325
326 update_escritorio(connection, 1, "São Paulo", "11999999999", "Av Paulista", "2000", "SP", "Brasil", "01000000", "Sudeste")
327 update_funcionario(connection, 1, "Silva", "João Paulo", "101", "joao.paulo@email.com", 1, None, "Gerente Senior")
328 update_linha_produto(connection, 1, "Eletrônicos Premium", "<p>Produtos eletrônicos de alta qualidade</p>", "eletronicos_premium.jpg")
329 update_produto(connection, 1, "Smartphone X Pro", 1, "1:1", "TechCorp", "Smartphone premium", 100, 2499.99, 2999.99)
330 update_cliente(connection, 1, "José Paulo Pereira", "Pereira", "José Paulo", "11999999999", "Rua A", "123", "São Paulo", "SP", "01000000", "Brasil", 1, 6000.00)
331 update_pedido(connection, 1, date(2024, 2, 1), date(2024, 2, 9), date(2024, 2, 4), "Entregue", "Entrega antecipada", 1)
332 update_detalhe_pedido(connection, 1, 1, 3, 2499.99, 1)
333 update_pagamento(connection, 1, 1, date(2024, 2, 4), 7499.97)
334
335
336 delete_pagamento(connection, 1, 1)
337
338 delete_detalhe_pedido(connection, 1, 1)
339
340
341 delete_pedido(connection, 1)
342
343
344 delete_produto(connection, 1)
345
346 delete_linha_produto(connection, 1)
347

```

347,0-1

```

342
343
344     delete_produto(connection, 1)
345
346     delete_linha_produto(connection, 1)
347
348
349     delete_cliente(connection, 1)
350
351     delete_funcionario(connection, 2)
352
353
354     delete_escritorio(connection, 1)
355
356     connection.close()
357
358
359 if __name__ == "__main__":
360     main()

```

```
$ python app.py
Conexão com o MySQL bem-sucedida
Escritório adicionado com sucesso
Escritório adicionado com sucesso
Escritório adicionado com sucesso
Funcionário adicionado com sucesso
Funcionário adicionado com sucesso
Funcionário adicionado com sucesso
Cliente adicionado com sucesso
Cliente adicionado com sucesso
Cliente adicionado com sucesso
Linha de produto adicionada com sucesso
Linha de produto adicionada com sucesso
Linha de produto adicionada com sucesso
Produto adicionado com sucesso
Produto adicionado com sucesso
Produto adicionado com sucesso
Pedido adicionado com sucesso
Pedido adicionado com sucesso
Pedido adicionado com sucesso
Detalhe do pedido adicionado com sucesso
Detalhe do pedido adicionado com sucesso
Detalhe do pedido adicionado com sucesso
Pagamento adicionado com sucesso
Pagamento adicionado com sucesso
Pagamento adicionado com sucesso
```

```
Leitura de todas as tabelas:

Escritórios: [(1, 'São Paulo', '119999999999', 'Av Paulista', '1000', 'SP', 'Brasil', '010000000', 'Sudeste'), (2, 'Rio de Janeiro', '218888888888', 'Av Atlântica', '2000', 'RJ', 'Brasil', '020000000', 'Sudeste'), (3, 'Belo Horizonte', '317777777777', 'Av Afonso Pena', '3000', 'MG', 'Brasil', '030000000', 'Sudeste')]

Funcionários: [(1, 'Silva', 'João', '101', 'joao@email.com', 1, None, 'Gerente'), (2, 'Santos', 'Maria', '102', 'maria@email.com', 2, 1, 'Vendedor'), (3, 'Oliveira', 'Pedro', '103', 'pedro@email.com', 3, 1, 'Vendedor')]

Linhas de Produtos: [(1, 'Eletrônicos', '<p>Produtos eletrônicos</p>', 'eletronicos.jpg'), (2, 'Móveis', '<p>Móveis para casa</p>', 'moveis.jpg'), (3, 'Roupas', '<p>Vestuário', 'roupas.jpg')]

Produtos: [(1, 'Smartphone X', 1, '1:1', 'TechCorp', 'Smartphone avançado', 100, 1999.99, 2499.99), (2, 'Sofá Confort', 2, '1:1', 'MoveisLux', 'Sofá 3 lugares', 50, 2999.99, 3499.99), (3, 'Camisa Casual', 3, '1:1', 'FashionStyle', 'Camisa manga curta', 200, 99.99, 149.99)]

Clientes: [(1, 'José Pereira', 'Pereira', 'José', '119999999999', 'Rua A', '123', 'São Paulo', 'SP', '010000000', 'Brasil', 1, 5000.0), (2, 'Ana Santos', 'Santos', 'Ana', '218888888888', 'Rua B', '456', 'Rio de Janeiro', 'RJ', '020000000', 'Brasil', 2, 3000.0), (3, 'Carlos Lima', 'Lima', 'Carlos', '317777777777', 'Rua C', '789', 'Belo Horizonte', 'MG', '030000000', 'Brasil', 3, 4000.0)]

Pedidos: [(1, datetime.date(2024, 2, 1), datetime.date(2024, 2, 10), datetime.date(2024, 2, 5), 'Entregue', 'Entrega normal', 1), (2, datetime.date(2024, 2, 2), datetime.date(2024, 2, 11), datetime.date(2024, 2, 6), 'Entregue', 'Entrega expressa', 2), (3, datetime.date(2024, 2, 3), datetime.date(2024, 2, 12), datetime.date(2024, 2, 7), 'Em processamento', 'Aguardando pagamento', 3)]

Detalhes de Pedido: [(1, 1, 2, 1999.99, 1), (2, 2, 1, 2999.99, 1), (3, 3, 3, 99.99, 1)]

Pagamentos: [(1, 1, datetime.date(2024, 2, 5), 3999.98), (2, 2, datetime.date(2024, 2, 6), 2999.99), (3, 3, datetime.date(2024, 2, 7), 299.97)]

Escritório atualizado com sucesso
Funcionário atualizado com sucesso
Linha de produto atualizada com sucesso
Produto atualizado com sucesso
Cliente atualizado com sucesso
Pedido atualizado com sucesso
```