

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, criamos o container do Docker e entramos no Container MySQL para executar os comandos de criação de tabelas:

```
#####
#                               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.                                                           #
#####
[node2] (local) root@192.168.0.12 ~
$ 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
893b018337e2: Pull complete
Digest: sha256:146682692a3aa409eae7b7dc6a30f637c6cb49b6ca901c2cd160becc81127d3b
Status: Downloaded newer image for mysql:latest
46bf1e7806d67ec6e55fff16f045f449eaec16848f12bf7a38d6b47ab10070f6
[node2] (local) root@192.168.0.12 ~
$ 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>
```

```
mysql> USE BD_AVIOES
Database changed
mysql>
mysql> CREATE TABLE TB_CLIENTES (
-> id INT NOT NULL PRIMARY KEY,
-> customerName VARCHAR(200) DEFAULT NULL,
-> contactLastName VARCHAR(100) DEFAULT NULL,
-> contactFirstName VARCHAR(100) DEFAULT NULL,
-> phone VARCHAR(20) DEFAULT NULL,
-> addressLine1 VARCHAR(40) DEFAULT NULL,
-> addressLine2 VARCHAR(40) DEFAULT NULL,
-> city VARCHAR(80) DEFAULT NULL,
-> state VARCHAR(80) DEFAULT NULL,
-> country VARCHAR(80) DEFAULT NULL,
-> postalCode VARCHAR(80) DEFAULT NULL,
-> salesRepEmployeeNumber VARCHAR(40) DEFAULT NULL,
-> creditLimit INT DEFAULT NULL);
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE TB_DETALHES_PEDIDOS (
-> id_orderNumber INT NOT NULL,
-> id_productCode INT NOT NULL,
-> quantityOrdered INT DEFAULT NULL,
-> priceEach INT DEFAULT NULL,
-> orderLineNumber VARCHAR(20) DEFAULT NULL);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql>
mysql> CREATE TABLE TB_EMPREGADOS (
-> id INT NOT NULL,
-> lastName VARCHAR(80) DEFAULT NULL,
-> firstName VARCHAR(20) DEFAULT NULL,
-> extension VARCHAR(40) DEFAULT NULL,
-> email VARCHAR(80) DEFAULT NULL,
-> officeCode VARCHAR(20) DEFAULT NULL,
-> reportsTo VARCHAR(20) DEFAULT NULL,
-> jobTitle VARCHAR(40) DEFAULT NULL );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE TB_ESCRITORIOS (
-> id INT NOT NULL,
-> city VARCHAR(100) DEFAULT NULL,
-> state VARCHAR(100) DEFAULT NULL,
-> country VARCHAR(100) DEFAULT NULL,
-> phone VARCHAR(20) DEFAULT NULL,
-> addressLine1 VARCHAR(40) DEFAULT NULL,
-> addressLine2 VARCHAR(40) DEFAULT NULL,
-> postalCode VARCHAR(40) DEFAULT NULL,
-> territory VARCHAR(60) DEFAULT NULL );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
mysql>
mysql> CREATE TABLE TB_LINHAS_DE_PRODUTOS (
-> id INT NOT NULL,
-> htmlDescription text,
-> image VARCHAR(100) DEFAULT NULL,
-> textDescription VARCHAR(100) DEFAULT NULL );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE TB_PAGAMENTOS (
-> id INT NOT NULL,
-> id_customerNumber VARCHAR(100) DEFAULT NULL,
-> paymentDate VARCHAR(20) DEFAULT NULL,
-> amount INT DEFAULT NULL );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
mysql> CREATE TABLE TB_PEDIDOS (
-> id INT NOT NULL,
-> id_customerNumber VARCHAR(100) DEFAULT NULL,
-> orderDate VARCHAR(20) DEFAULT NULL,
-> requiredDate VARCHAR(20) DEFAULT NULL,
-> shippedDate VARCHAR(20) DEFAULT NULL,
-> status VARCHAR(40) DEFAULT NULL,
-> comments text );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE TB_PRODUTOS (
-> id INT NOT NULL,
-> productName VARCHAR(100) DEFAULT NULL,
-> id_productLine INT DEFAULT NULL,
-> productScale VARCHAR(100) DEFAULT NULL,
-> productVendor VARCHAR(100) DEFAULT NULL,
-> productDescription text,
-> quantityInStock INT DEFAULT NULL,
-> buyPrice INT DEFAULT NULL,
-> MSRP INT DEFAULT NULL );
Query OK, 0 rows affected (0.01 sec)
```

```
# 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@192.168.0.13 ~
$ python -m venv myenv
[node1] (local) root@192.168.0.13 ~
$ source myenv/bin/activate
(myenv) [node1] (local) root@192.168.0.13 ~
$ 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@192.168.0.13 ~
$
```

```
1 import mysql.connector
2 from mysql.connector import Error
3
4 def create_connection():
5     connection = None
6     try:
7         connection = mysql.connector.connect(
8             host='192.168.0.13',
9             port='3306',
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
19 def execute_query(connection, query, data=None):
20     cursor = connection.cursor()
21     cursor.execute(query, data or ())
```

```
19 def execute_query(connection, query, data=None):
20     cursor = connection.cursor()
21     cursor.execute(query, data or ())
22     connection.commit()
23
24 def fetch_query(connection, query, data=None):
25     cursor = connection.cursor()
26     cursor.execute(query, data or ())
27     return cursor.fetchall()
28
29 def create_cliente(connection, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode, salesRepEmployeeNumber, creditLimit):
30     query = """
31         INSERT INTO TB_CLIENTES (customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode, salesRepEmployeeNumber, creditLimit)
32         VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
33     """
34     execute_query(connection, query, (customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, country, postalCode, salesRepEmployeeNumber, creditLimit))
35     print("Cliente adicionado com sucesso")
36
```

```

37 def create_pedido(connection, id_customerNumber, orderDate, requiredDate, shippedDate, status, comments):
38     query = """
39         INSERT INTO TB_PEDIDOS (id_customerNumber, orderDate, requiredDate, shippedDate, status, comments)
40         VALUES (%s, %s, %s, %s, %s, %s)
41     """
42     execute_query(connection, query, (id_customerNumber, orderDate, requiredDate, shippedDate, status, comments))
43     print("Pedido adicionado com sucesso")
44
45 def create_pagamento(connection, id_customerNumber, paymentDate, amount):
46     query = """
47         INSERT INTO TB_PAGAMENTOS (id_customerNumber, paymentDate, amount)
48         VALUES (%s, %s, %s)
49     """
50     execute_query(connection, query, (id_customerNumber, paymentDate, amount))
51     print("Pagamento registrado com sucesso")
52
53 def create_produto(connection, productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, buyP
rice, MSRP):
54     query = """
55         INSERT INTO TB_PRODUTOS (productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, bu
yPrice, MSRP)

```

```

55         INSERT INTO TB_PRODUTOS (productName, id_productLine, productScale, productVendor, productDescription, quantityInStock, bu
yPrice, MSRP)
56         VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
57     """
58     execute_query(connection, query, (productName, id_productLine, productScale, productVendor, productDescription, quantityInStoc
k, buyPrice, MSRP))
59     print("Produto adicionado com sucesso")
60
61 def read_table(connection, table_name):
62     query = f"SELECT * FROM {table_name}"
63     results = fetch_query(connection, query)
64     for row in results:
65         print(row)
66
67 def update_clientes(connection, cliente_id, customerName, contactLastName, contactFirstName):
68     query = """
69         UPDATE TB_CLIENTES SET customerName = %s, contactLastName = %s, contactFirstName = %s WHERE id = %s
70     """
71     execute_query(connection, query, (customerName, contactLastName, contactFirstName, cliente_id))
72     print("Cliente atualizado com sucesso")

```

```

74 def delete_from_table(connection, table_name, record_id):
75     query = f"DELETE FROM {table_name} WHERE id = %s"
76     execute_query(connection, query, (record_id,))
77     print(f"Registro removido da tabela {table_name}")
78
79 def main():
80     connection = create_connection()
81     if connection is None:
82         return
83
84     # Exemplo de uso das funções CRUD
85     create_cliente(connection, 'Empresa X', 'Silva', 'João', '99999-9999', 'Rua A', 'Apto 101', 'Cidade X', 'Estado Y', 'País Z',
'12345-678', 'EMP001', 5000)
86     create_pedido(connection, 1, '2025-02-26', '2025-03-01', '2025-03-05', 'Enviado', 'Pedido entregue com sucesso')
87     create_pagamento(connection, 1, '2025-02-26', 1500)
88     create_produto(connection, 'Produto A', 1, '1:24', 'Fabricante X', 'Descrição do produto', 100, 50, 100)
89     read_table(connection, 'TB_CLIENTES')
90     read_table(connection, 'TB_PEDIDOS')
91     read_table(connection, 'TB_PAGAMENTOS')
92     read_table(connection, 'TB_PRODUTOS')
93     update_cliente(connection, 1, 'Empresa X Modificada', 'Silva', 'João')

```

```

83
84     # Exemplo de uso das funções CRUD
85     create_cliente(connection, 'Empresa X', 'Silva', 'João', '99999-9999', 'Rua A', 'Apto 101', 'Cidade X', 'Estado Y', 'País Z',
'12345-678', 'EMP001', 5000)
86     create_pedido(connection, 1, '2025-02-26', '2025-03-01', '2025-03-05', 'Enviado', 'Pedido entregue com sucesso')
87     create_pagamento(connection, 1, '2025-02-26', 1500)
88     create_produto(connection, 'Produto A', 1, '1:24', 'Fabricante X', 'Descrição do produto', 100, 50, 100)
89     read_table(connection, 'TB_CLIENTES')
90     read_table(connection, 'TB_PEDIDOS')
91     read_table(connection, 'TB_PAGAMENTOS')
92     read_table(connection, 'TB_PRODUTOS')
93     update_cliente(connection, 1, 'Empresa X Modificada', 'Silva', 'João')
94     read_table(connection, 'TB_CLIENTES')
95     delete_from_table(connection, 'TB_CLIENTES', 1)
96     read_table(connection, 'TB_CLIENTES')
97
98     connection.close()
99
100 if __name__ == "__main__":
101     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
```