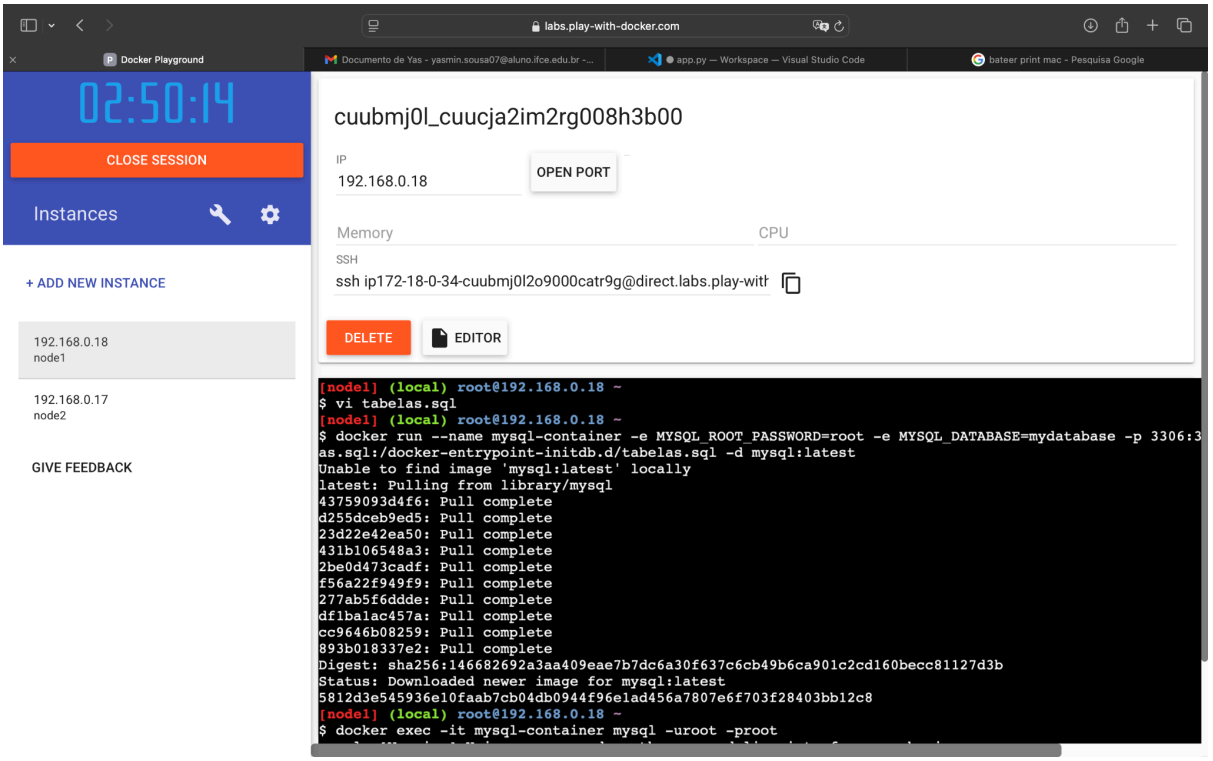
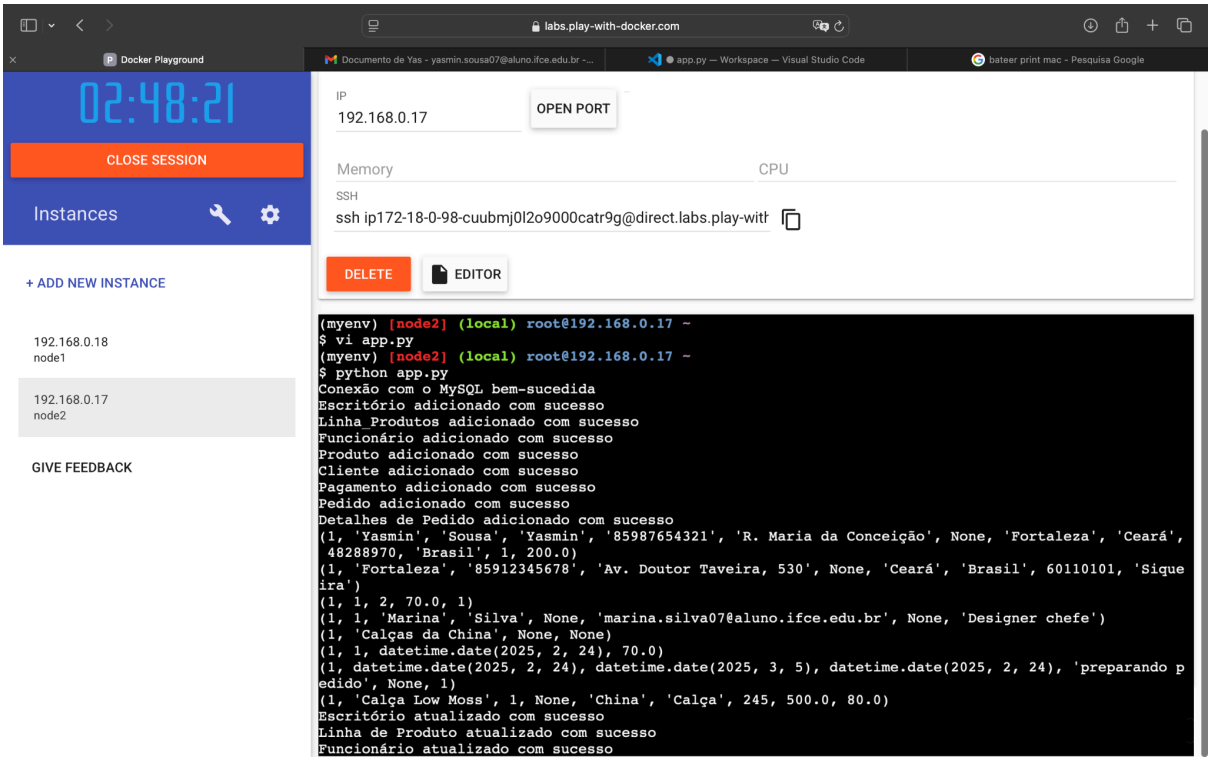
Atividade-08

Participantes:
Kauã Sousa de Oliveira
Yasmin Sousa Oliveira
Curso: Informática
Período: 4

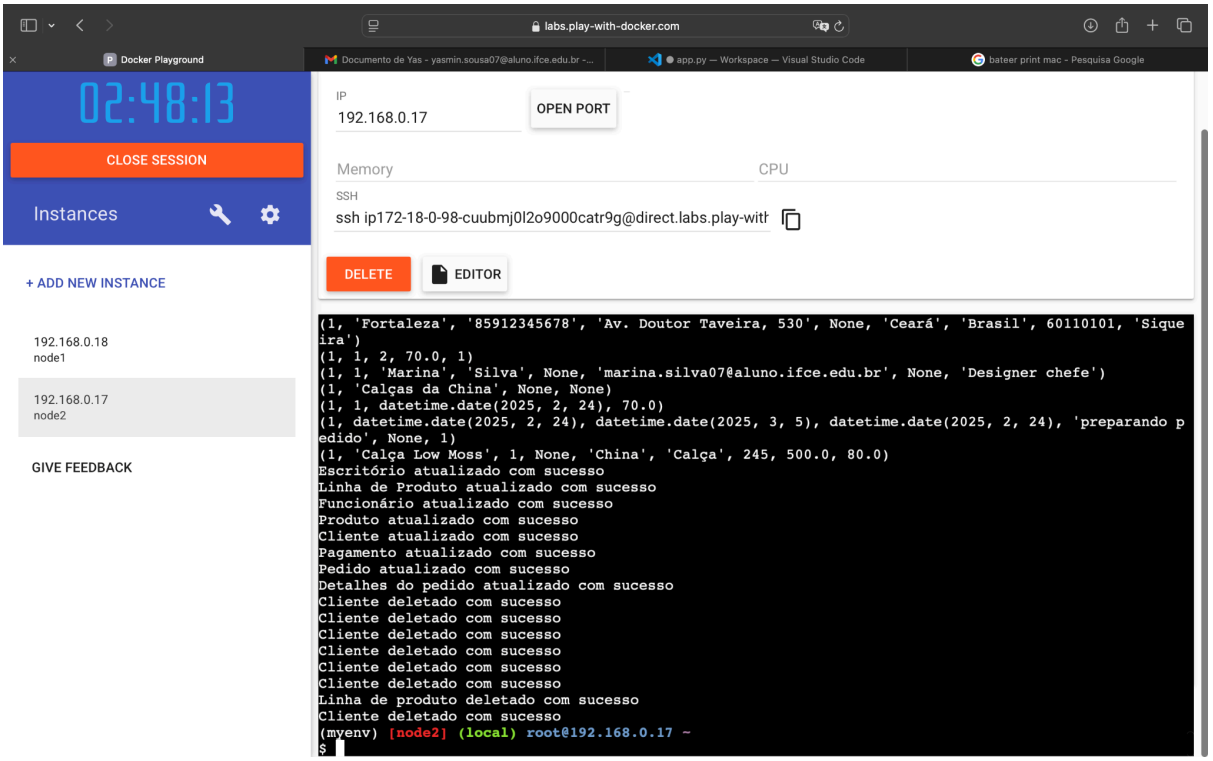
Implementar os CRUDs usando o Python (Aula do dia 10/02/2024) das tabelas do banco de dados criado na atividade-07.
Tarefa em dupla postado no GitHub de cada participante com o link do Github referenciado no Google Classroom.



Criação do container MySQL



Executando o código python / parte 1



Executando o código python / parte 2

CÓDIGO PYTHON

```
import mysql.connector
from mysql.connector import Error
from datetime import date

def create_connection():
    """Cria uma conexão com o banco de dados MySQL."""
    connection = None
    try:
        connection = mysql.connector.connect(
            host='192.168.0.18',
            port='3306',
            user='root',
            password='root',
            database='mydatabase'
        )
        print("Conexão com o MySQL bem-sucedida")
    except Error as e:
        print(f"Erro '{e}' ocorreu")

    return connection

def create_cliente(connection, id, nomeCliente, contatoSobrenome, contatoPrimeiroNome, telefone,
LinhaEndereco1, LinhaEndereco2, cidade, estado, codigoPostal, pais, numeroFuncionarioVendas_id,
limiteCredito):
    """Insere um novo cliente na tabela TB_CLIENTES."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_CLIENTES (id, nomeCliente, contatoSobrenome, contatoPrimeiroNome,
telefone, LinhaEndereço1, LinhaEndereço2, cidade, estado, codigoPostal, país, numeroFuncionarioVendas_id,
limiteCredito) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
    cursor.execute(query, (id, nomeCliente, contatoSobrenome, contatoPrimeiroNome, telefone, LinhaEndereco1,
LinhaEndereco2, cidade, estado, codigoPostal, pais, numeroFuncionarioVendas_id, limiteCredito))
    connection.commit()
    print("Cliente adicionado com sucesso")

def create_detalhes_pedidos(connection, id, codigoProduto_id, quantidadePedida, cadaPreco,
numeroLinhaPedido):
    """Insere um novo detalhes de pedido na tabela TB_DETALHES_PEDIDOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_DETALHES_PEDIDOS (id, codigoProduto_id, quantidadePedida, cadaPreço,
numeroLinhaPedido) VALUES (%s, %s, %s, %s, %s)"
    cursor.execute(query, (id, codigoProduto_id, quantidadePedida, cadaPreco, numeroLinhaPedido))
    connection.commit()
    print("Detalhes de Pedido adicionado com sucesso")

def create_escritorio(connection, id, cidade, telefone, LinhaEndereco1, LinhaEndereco2, estado, pais,
codigoPostal, territorio):
    """Insere um novo escritório na tabela TB_ESCRITORIOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_ESCRITORIOS (id, cidade, telefone, linhaEndereço1, linhaEndereço2, estado,
país, codigoPostal, territorio) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
```

```
    cursor.execute(query, (id, cidade, telefone, LinhaEndereco1, LinhaEndereco2, estado, pais, codigoPostal,
territorio))
    connection.commit()
    print("Escritório adicionado com sucesso")

def create_funcionario(connection, id, codigoEscritorio_id, nome, sobrenome, extenção, email, relatarPara_id,
cargo):
    """Insere um novo funcionário na tabela TB_FUNCIONARIOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_FUNCIONARIOS (id, codigoEscritorio_id, nome, sobrenome, extenção, email,
relatarPara_id, cargo) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
    cursor.execute(query, (id, codigoEscritorio_id, nome, sobrenome, extenção, email, relatarPara_id, cargo))
    connection.commit()
    print("Funcionário adicionado com sucesso")

def create_linha_produtos(connection, id, descriçãoTexto, descriçãoHTML, image):
    """Insere uma nova linha de produto na tabela TB_LINHA_PRODUTOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_LINHA_PRODUTOS (id, descriçãoTexto, descriçãoHTML, image) VALUES
(%s, %s, %s, %s)"
    cursor.execute(query, (id, descriçãoTexto, descriçãoHTML, image))
    connection.commit()
    print("Linha_Produtos adicionado com sucesso")

def create_pagamento(connection, id, numeroCliente_id, dataPagamento, quantia):
    """Insere um novo pagamento na tabela TB_PAGAMENTOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_PAGAMENTOS (id, numeroCliente_id, dataPagamento, quantia) VALUES
(%s, %s, %s, %s)"
    cursor.execute(query, (id, numeroCliente_id, dataPagamento, quantia))
    connection.commit()
    print("Pagamento adicionado com sucesso")

def create_pedido(connection, id, dataPedido, dataRequerida, dataEnvio, status, comentarios,
numeroCliente_id):
    """Insere um novo pedido na tabela TB_PEDIDOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_PEDIDOS (id, dataPedido, dataRequerida, dataEnvio, status, comentarios,
numeroCliente_id) VALUES (%s, %s, %s, %s, %s, %s, %s)"
    cursor.execute(query, (id, dataPedido, dataRequerida, dataEnvio, status, comentarios, numeroCliente_id))
    connection.commit()
    print("Pedido adicionado com sucesso")

def create_produto(connection, id, nomeProduto, linhaProduto_id, escalaProduto, fornecedorProduto,
descricaoProduto, quantidadeEstoque, precoCompra, MSRP):
    """Insere um novo produto na tabela TB_PRODUTOS."""
    cursor = connection.cursor()
    query = "INSERT INTO TB_PRODUTOS (id, nomeProduto, linhaProduto_id, escalaProduto,
fornecedorProduto, descriçãoProduto, quantidadeEstoque, preçoCompra, MSRP) VALUES (%s, %s, %s, %s,
%s, %s, %s, %s, %s)"
    cursor.execute(query, (id, nomeProduto, linhaProduto_id, escalaProduto, fornecedorProduto,
descricaoProduto, quantidadeEstoque, precoCompra, MSRP))
```

```
    connection.commit()
    print("Produto adicionado com sucesso")

def read_clientes(connection):
    """Lê todos os clientes da tabela TB_CLIENTES."""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_CLIENTES")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_produtos(connection):
    """Lê todos os clientes da tabela TB_PRODUTOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_PRODUTOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_pedidos(connection):
    """Lê todos os clientes da tabela TB_PEDIDOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_PEDIDOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_pagamentos(connection):
    """Lê todos os clientes da tabela TB_PAGAMENTOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_PAGAMENTOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_linha_produtos(connection):
    """Lê todos os clientes da tabela TB_LINHA_PRODUTOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_LINHA_PRODUTOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_funcionarios(connection):
    """Lê todos os clientes da tabela TB_FUNCIONARIOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_FUNCIONARIOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_detalhes_pedidos(connection):
```

```
    """Lê todos os clientes da tabela TB_DETALHES_PEDIDOS"""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_DETALHES_PEDIDOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def read_escritorios(connection):
    """Lê todos os clientes da tabela TB_ESCRITORIOS."""
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM TB_ESCRITORIOS")
    clientes = cursor.fetchall()
    for cliente in clientes:
        print(cliente)

def update_cliente(connection, id, telefone):
    """Atualiza um cliente existente na tabela TB_CLIENTES."""
    cursor = connection.cursor()
    query = "UPDATE TB_CLIENTES SET telefone = %s WHERE id = %s"
    cursor.execute(query, (telefone, id))
    connection.commit()
    print("Cliente atualizado com sucesso")

def update_detalhes_pedido(connection, id, quantidadePedida):
    """Atualiza um detalhe de pedido existente na tabela TB_DETALHES_PEDIDOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_DETALHES_PEDIDOS SET quantidadePedida = %s WHERE id = %s"
    cursor.execute(query, (quantidadePedida, id))
    connection.commit()
    print("Detalhes do pedido atualizado com sucesso")

def update_escritorio(connection, id, telefone):
    """Atualiza um escritório existente na tabela TB_ESCRITORIOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_ESCRITORIOS SET telefone = %s WHERE id = %s"
    cursor.execute(query, (telefone, id))
    connection.commit()
    print("Escritório atualizado com sucesso")

def update_funcionario(connection, id, email):
    """Atualiza um funcionário existente na tabela TB_FUNCIONARIOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_FUNCIONARIOS SET email = %s WHERE id = %s"
    cursor.execute(query, (email, id))
    connection.commit()
    print("Funcionário atualizado com sucesso")

def update_linha_produto(connection, id, descricao):
    """Atualiza um linha de produto existente na tabela TB_LINHA_PRODUTOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_LINHA_PRODUTOS SET descriçãoTexto = %s WHERE id = %s"
    cursor.execute(query, (descricao, id))
```

```
    connection.commit()
    print("Linha de Produto atualizado com sucesso")

def update_pagamentos(connection, id, dataPagamento):
    """Atualiza um pagamento existente na tabela TB_PAGAMENTOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_PAGAMENTOS SET dataPagamento = %s WHERE id = %s"
    cursor.execute(query, (dataPagamento, id))
    connection.commit()
    print("Pagamento atualizado com sucesso")

def update_pedidos(connection, id, status):
    """Atualiza um pedido existente na tabela TB_PEDIDOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_PEDIDOS SET status = %s WHERE id = %s"
    cursor.execute(query, (status, id))
    connection.commit()
    print("Pedido atualizado com sucesso")

def update_produtos(connection, id, preco):
    """Atualiza um produto existente na tabela TB_PRODUTOS."""
    cursor = connection.cursor()
    query = "UPDATE TB_PRODUTOS SET preçoCompra = %s WHERE id = %s"
    cursor.execute(query, (preco, id))
    connection.commit()
    print("Produto atualizado com sucesso")

def delete_cliente(connection, cliente_id):
    """Deleta um cliente da tabela TB_CLIENTES."""
    cursor = connection.cursor()
    query = "DELETE FROM TB_CLIENTES WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_linha_produto(connection, cliente_id):
    """Deleta um cliente da tabela TB_LINHA_PRODUTOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_LINHA_PRODUTOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Linha de produto deletado com sucesso")

def delete_detalhes_pedidos(connection, cliente_id):
    """Deleta um cliente da tabela TB_DETALHES_PEDIDOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_DETALHES_PEDIDOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_escritorios(connection, cliente_id):
```

```
    """Deleta um cliente da tabela TB_ESCRITORIOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_ESCRITORIOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_funcionarios(connection, cliente_id):
    """Deleta um cliente da tabela TB_FUNCIONARIOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_FUNCIONARIOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_pagamentos(connection, cliente_id):
    """Deleta um cliente da tabela TB_PAGAMENTOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_PAGAMENTOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_pedidos(connection, cliente_id):
    """Deleta um cliente da tabela TB_PEDIDOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_PEDIDOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

def delete_produtos(connection, cliente_id):
    """Deleta um cliente da tabela TB_PRODUTOS"""
    cursor = connection.cursor()
    query = "DELETE FROM TB_PRODUTOS WHERE id = %s"
    cursor.execute(query, (cliente_id,))
    connection.commit()
    print("Cliente deletado com sucesso")

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

    # Exemplo de uso das funções CRUD
    create_escritorio(connection, 1, "Fortaleza", "85912345678", "Av. Doutor Taveira, 530", None, "Ceará",
"Brasil", 60110101, "Siqueira")
    create_linha_produtos(connection, 1, "Calças da China", None, None)
    create_funcionario(connection, 1, 1, "Marina", "Silva", None, "marina.silva07@aluno.ifce.edu.br", None,
"Designer chefe")
    create_produto(connection, 1, "Calça Low Moss", 1, None, "China", "Calça", 245, 500, 80,)
```

```
    create_cliente(connection, 1, "Yasmin", "Sousa", "Yasmin", "85987654321", "R. Maria da Conceição", None,
"Fortaleza", "Ceará", 48288970, "Brasil", 1, 200)
    create_pagamento(connection, 1,1, date(2025, 2, 24), 70)
    create_pedido(connection, 1, date(2025, 2, 24), date(2025, 3, 5), date(2025, 2, 24), "preparando pedido",
None, 1)
    create_detalhes_pedidos(connection, 1, 1, 2, 70, 1)

    read_clientes(connection)
    read_escritorios(connection)
    read_detalhes_pedidos(connection)
    read_funcionarios(connection)
    read_linha_produtos(connection)
    read_pagamentos(connection)
    read_pedidos(connection)
    read_produtos(connection)

    update_escritorio(connection, 1, "85912345677")
    update_linha_produto(connection, 1,"Calça de Taiwan")
    update_funcionario(connection, 1, "yasmin.silva07@aluno.ifce.edu.br")
    update_produtos(connection, 1, 70 )
    update_cliente(connection, 1, "85912345677")
    update_pagamentos(connection, 1, date(2025, 2, 24))
    update_pedidos(connection, 1, "entregue")
    update_detalhes_pedido(connection, 1, 1)

    delete_detalhes_pedidos(connection, 1)
    delete_pedidos(connection, 1)
    delete_pagamentos(connection, 1)
    delete_cliente(connection, 1)
    delete_produtos(connection, 1)
    delete_funcionarios(connection, 1)
    delete_linha_produto(connection, 1)
    delete_escritorios(connection, 1)



    connection.close()

if __name__ == "__main__":
    main()
```

SQL Utilizado para a Criação das Tabelas

```
-- phpMyAdmin SQL Dump
-- version 5.2.2
-- https://www.phpmyadmin.net/
--
-- Host: mysql:3306
-- Tempo de geração: 05/02/2025 às 17:56
-- Versão do servidor: 8.0.41
-- Versão do PHP: 8.2.27

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Banco de dados: `mydatabase`
--

USE mydatabase;

-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_CLIENTES`
--

CREATE TABLE `TB_CLIENTES` (
  `id` int NOT NULL,
  `nomeCliente` text,
  `contatoSobrenome` text,
  `contatoPrimeiroNome` text,
  `telefone` text,
  `LinhaEndereço1` text,
  `LinhaEndereço2` text,
  `cidade` text,
  `estado` text,
  `codigoPostal` int DEFAULT NULL,
  `país` text,
  `numeroFuncionarioVendas_id` int DEFAULT NULL,
  `limiteCredito` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_CLIENTES`
--
```

```
-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_DETALHES_PEDIDOS`
--

CREATE TABLE `TB_DETALHES_PEDIDOS` (
  `id` int NOT NULL,
  `codigoProduto_id` int NOT NULL,
  `quantidadePedida` int DEFAULT NULL,
  `cadaPreço` double DEFAULT NULL,
  `numeroLinhaPedido` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_DETALHES_PEDIDOS`
--

-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_ESCRITORIOS`
--

CREATE TABLE `TB_ESCRITORIOS` (
  `id` int NOT NULL,
  `cidade` text,
  `telefone` text,
  `linhaEndereço1` text,
  `linhaEndereço2` text,
  `estado` text,
  `país` text,
  `codigoPostal` int DEFAULT NULL,
  `territorio` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_ESCRITORIOS`
--

-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_FUNCIONARIOS`
--

CREATE TABLE `TB_FUNCIONARIOS` (
  `id` int NOT NULL,
  `codigoEscritorio_id` int NOT NULL,
  `nome` text,
  `sobrenome` text,
```

```
  `extenção` text,
  `email` text,
  `relatarPara_id` int,
  `cargo` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_FUNCIONARIOS`
--

-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_LINHA_PRODUTOS`
--

CREATE TABLE `TB_LINHA_PRODUTOS` (
  `id` int NOT NULL,
  `descriçãoTexto` text,
  `descriçãoHTML` longtext,
  `image` blob
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_LINHA_PRODUTOS`
--


-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_PAGAMENTOS`
--

CREATE TABLE `TB_PAGAMENTOS` (
  `id` int NOT NULL,
  `numeroCliente_id` int NOT NULL,
  `dataPagamento` date DEFAULT NULL,
  `quantia` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_PAGAMENTOS`
--

-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_PEDIDOS`
--

CREATE TABLE `TB_PEDIDOS` (
```

```
  `id` int NOT NULL,
  `dataPedido` date DEFAULT NULL,
  `dataRequerida` date DEFAULT NULL,
  `dataEnvio` date DEFAULT NULL,
  `status` text,
  `comentarios` text,
  `numeroCliente_id` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_PEDIDOS`
--


-- --------------------------------------------------------

--
-- Estrutura para tabela `TB_PRODUTOS`
--

CREATE TABLE `TB_PRODUTOS` (
  `id` int NOT NULL,
  `nomeProduto` text,
  `linhaProduto_id` int NOT NULL,
  `escalaProduto` text,
  `fornecedorProduto` text,
  `descriçãoProduto` text,
  `quantidadeEstoque` int DEFAULT NULL,
  `preçoCompra` double DEFAULT NULL,
  `MSRP` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Despejando dados para a tabela `TB_PRODUTOS`
--


--
-- Índices para tabelas despejadas
--

--
-- Índices de tabela `TB_CLIENTES`
--
ALTER TABLE `TB_CLIENTES`
  ADD PRIMARY KEY (`id`),
  ADD KEY `numeroFuncionarioVendas_id` (`numeroFuncionarioVendas_id`);

--
-- Índices de tabela `TB_DETALHES_PEDIDOS`
--
ALTER TABLE `TB_DETALHES_PEDIDOS`
```

```
  ADD PRIMARY KEY (`id`),
  ADD KEY `codigoProduto_id` (`codigoProduto_id`);

--
-- Índices de tabela `TB_ESCRITORIOS`
--
ALTER TABLE `TB_ESCRITORIOS`
  ADD PRIMARY KEY (`id`);

--
-- Índices de tabela `TB_FUNCIONARIOS`
--
ALTER TABLE `TB_FUNCIONARIOS`
  ADD PRIMARY KEY (`id`),
  ADD KEY `codigoEscritorio_id` (`codigoEscritorio_id`),
  ADD KEY `relatarPara_id` (`relatarPara_id`);

--
-- Índices de tabela `TB_LINHA_PRODUTOS`
--
ALTER TABLE `TB_LINHA_PRODUTOS`
  ADD PRIMARY KEY (`id`);

--
-- Índices de tabela `TB_PAGAMENTOS`
--
ALTER TABLE `TB_PAGAMENTOS`
  ADD PRIMARY KEY (`id`),
  ADD KEY `numeroCliente_id` (`numeroCliente_id`);

--
-- Índices de tabela `TB_PEDIDOS`
--
ALTER TABLE `TB_PEDIDOS`
  ADD PRIMARY KEY (`id`),
  ADD KEY `numeroCliente_id` (`numeroCliente_id`);

--
-- Índices de tabela `TB_PRODUTOS`
--
ALTER TABLE `TB_PRODUTOS`
  ADD PRIMARY KEY (`id`),
  ADD KEY `linhaProduto_id` (`linhaProduto_id`);

--
-- AUTO_INCREMENT para tabelas despejadas
--

--
-- AUTO_INCREMENT de tabela `TB_CLIENTES`
--
ALTER TABLE `TB_CLIENTES`
```

```
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT de tabela `TB_DETALHES_PEDIDOS`
--
ALTER TABLE `TB_DETALHES_PEDIDOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_ESCRITORIOS`
--
ALTER TABLE `TB_ESCRITORIOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_FUNCIONARIOS`
--
ALTER TABLE `TB_FUNCIONARIOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_LINHA_PRODUTOS`
--
ALTER TABLE `TB_LINHA_PRODUTOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_PAGAMENTOS`
--
ALTER TABLE `TB_PAGAMENTOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_PEDIDOS`
--
ALTER TABLE `TB_PEDIDOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de tabela `TB_PRODUTOS`
--
ALTER TABLE `TB_PRODUTOS`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- Restrições para tabelas despejadas
--

--
-- Restrições para tabelas `TB_CLIENTES`
--
ALTER TABLE `TB_CLIENTES`
```

```
  ADD CONSTRAINT `TB_CLIENTES_ibfk_1` FOREIGN KEY (`numeroFuncionarioVendas_id`)
REFERENCES `TB_FUNCIONARIOS` (`id`);

--
-- Restrições para tabelas `TB_DETALHES_PEDIDOS`
--
ALTER TABLE `TB_DETALHES_PEDIDOS`
  ADD CONSTRAINT `TB_DETALHES_PEDIDOS_ibfk_1` FOREIGN KEY (`codigoProduto_id`)
REFERENCES `TB_PRODUTOS` (`id`);

--
-- Restrições para tabelas `TB_FUNCIONARIOS`
--
ALTER TABLE `TB_FUNCIONARIOS`
  ADD CONSTRAINT `TB_FUNCIONARIOS_ibfk_1` FOREIGN KEY (`codigoEscritorio_id`)
REFERENCES `TB_ESCRITORIOS` (`id`),
  ADD CONSTRAINT `TB_FUNCIONARIOS_ibfk_2` FOREIGN KEY (`relatarPara_id`) REFERENCES
`TB_FUNCIONARIOS` (`id`);

--
-- Restrições para tabelas `TB_PAGAMENTOS`
--
ALTER TABLE `TB_PAGAMENTOS`
  ADD CONSTRAINT `TB_PAGAMENTOS_ibfk_1` FOREIGN KEY (`numeroCliente_id`) REFERENCES
`TB_CLIENTES` (`id`);

--
-- Restrições para tabelas `TB_PEDIDOS`
--
ALTER TABLE `TB_PEDIDOS`
  ADD CONSTRAINT `TB_PEDIDOS_ibfk_1` FOREIGN KEY (`numeroCliente_id`) REFERENCES
`TB_CLIENTES` (`id`);

--
-- Restrições para tabelas `TB_PRODUTOS`
--
ALTER TABLE `TB_PRODUTOS`
  ADD CONSTRAINT `TB_PRODUTOS_ibfk_1` FOREIGN KEY (`linhaProduto_id`) REFERENCES
`TB_LINHA_PRODUTOS` (`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```