

```
In [1]: # Versão linguagem Python utilizada

from platform import python_version
python_version()
```

```
Out[1]: '3.9.13'
```

```
In [2]: # Importando biblioteca SQLite

import sqlite3

sqlite3.sqlite_version
```

```
Out[2]: '3.39.3'
```

```
In [3]: # Importando biblioteca Pandas

import pandas as pd

pd.__version__
```

```
Out[3]: '2.1.4'
```

```
In [4]: # Carregando as tabelas

orders = pd.read_csv('./Warehouse-Orders---Orders.csv')
warehouse = pd.read_csv('./Warehouse-Orders---Warehouse.csv')
```

```
In [5]: # Criando um banco de dados

conexao = sqlite3.connect('data-orders.db')
```

```
In [10]: # Adicionando as tabelas carregadas ao banco de dados criado

orders.to_sql('table_orders', conexao, index=False, if_exists='replace')
warehouse.to_sql('table_warehouse', conexao, index=False, if_exists='replace')
```

```
Out[10]: 10
```


Out[70]:

	id_pedido	id_cliente	id_deposito	data_pedido	data_entrega
0	789	3731	8118	01/01/2019	01/04/2019
1	790	3486	8118	1/1/2019	1/4/2019
2	791	2623	8118	1/1/2019	1/4/2019
3	792	9869	8118	1/1/2019	1/4/2019
4	793	6866	8118	1/1/2019	1/4/2019
...
9994	10783	8554	9080	12/30/2019	1/2/2020
9995	10784	6439	6509	12/30/2019	1/2/2020
9996	10785	4379	6509	12/30/2019	1/2/2020
9997	10786	1799	6509	12/30/2019	1/2/2020
9998	10787	8469	4338	12/30/2019	1/2/2020

9999 rows × 5 columns

A tabela "orders" contém dados sobre compras/pedidos realizados, como id dos clientes, data do pedido, data de entrega e de qual depósito saiu o produto comprado.

In [71]:

```
# Verificando conteúdo da tabela "warehouse"

query2 = 'SELECT * FROM warehouse'

cursor.execute(query2)

# Visualizando os nomes das colunas

nomes_colunas = [description[0] for description in cursor.description]
print(nomes_colunas)

['warehouse_id', 'warehouse_alias', 'maximum_capacity', 'employee_total', 'state']
```

In [72]:

```
# Dados da query2

tabela2 = cursor.fetchall()
```

```
# Carregando os dados como dataframe do Pandas

df_warehouse = pd.DataFrame(tabela2, columns = ['id_deposito',
                                                'nome_deposito',
                                                'capacidade_max',
                                                'total_employees',
                                                'estado'])

df_warehouse
```

```
Out[72]:
```

	id_deposito	nome_deposito	capacidade_max	total_employees	estado
0	1543	Somerset Fulfillment Center	210	14	KY
1	2270	Bowling Green Warehouse	280	13	KY
2	2666	Lansing Fulfillment Center	290	16	MI
3	3417	Gatlinburg Warehouse	620	6	TN
4	3961	Lansing Storage Warehouse	740	22	MI
5	4338	Knoxville Fulfillment Center	215	13	TN
6	6509	Memphis Fulfillment Center	755	22	TN
7	8118	Ann Arbor Fulfillment Center	780	17	MI
8	9080	Frankfort Fulfillment Center	235	5	KY
9	9831	Clarsvill Warehouse	400	16	TN

A tabela "warehouse" contém dados sobre os depósitos, como nome do depósito, capacidade, quantidade de trabalhadores e estado.

É importante notar que 'id_deposito' é uma PRIMARY KEY nessa tabela e uma FOREIGN KEY na tabela de pedidos "orders".

```
In [73]: # Utilizando INNER JOIN para contar o número de estados incluídos nos dados de pedidos

query3 = """SELECT
            COUNT(DISTINCT warehouse.state) AS num_de_estados
        FROM
            orders
```


Out[74]:

	id_pedido	id_cliente	id_deposito	data_pedido	data_entrega	nome_do_deposito	estado
0	789	3731	8118	01/01/2019	01/04/2019	Ann Arbor Fulfillment Center	MI
1	790	3486	8118	1/1/2019	1/4/2019	Ann Arbor Fulfillment Center	MI
2	791	2623	8118	1/1/2019	1/4/2019	Ann Arbor Fulfillment Center	MI
3	792	9869	8118	1/1/2019	1/4/2019	Ann Arbor Fulfillment Center	MI
4	793	6866	8118	1/1/2019	1/4/2019	Ann Arbor Fulfillment Center	MI
...
9994	10783	8554	9080	12/30/2019	1/2/2020	Frankfort Fulfillment Center	KY
9995	10784	6439	6509	12/30/2019	1/2/2020	Memphis Fulfillment Center	TN
9996	10785	4379	6509	12/30/2019	1/2/2020	Memphis Fulfillment Center	TN
9997	10786	1799	6509	12/30/2019	1/2/2020	Memphis Fulfillment Center	TN
9998	10787	8469	4338	12/30/2019	1/2/2020	Knoxville Fulfillment Center	TN

9999 rows × 7 columns

In [76]:

```
cursor.close()
```