Import software libraries

In [1]:
```
# Import required libraries
import sys             # Read system parameters.
import pandas as pd    # Manipulate and analyze data.
import sqlite3 as sql3 # Manage SQL database.

# Summarize software libraries used.
print('Libraries used in this project:')
print('- Python {}'.format(sys.version))
print('- pandas {}'.format(pd.__version__))
print('- sqlite3 {}'.format(sql3.sqlite_version))

'''
Libraries used in this project:
- Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
- pandas 1.3.4
- sqlite3 3.36.0
'''
```

Examine the database

In [2]:
```
# Connect to SQLite database.
db = sql3.connect("prod_sample.db")
db
```

Out[2]:
```
<sqlite3.Connection at 0x278c5cb4990>
```

In [3]:
```
# List all the tables in the database.

cursor = db.cursor()

cursor.execute("SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;")

available_table=(cursor.fetchall())
```

In [4]:
```
available_table
```

Out[4]:
```
[('online_retail_history',), ('stock_description',)]
```

Read data from the `online_retail_history` table

In [5]:
```
# Write the query to be executed that selects everything from the online_retail_history table.

table1 = pd.read_sql_query('SELECT * FROM online_retail_history', db)
table1
```

```
# Use the read_sql function in pandas to read a query into a DataFrame.


# Preview the first five rows of the data.

table1.head()
```

Out[5]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30
1	536367	84879	32	2010-12-01 08:34:00	1.69	13047	United Kingdom	54.08
2	536373	85123A	6	2010-12-01 09:02:00	2.55	17850	United Kingdom	15.30
3	536375	85123A	6	2010-12-01 09:32:00	2.55	17850	United Kingdom	15.30
4	536378	20725	10	2010-12-01 09:37:00	1.65	14688	United Kingdom	16.50

In [6]:
```
# Get the shape of the data.

table1.shape
```

Out[6]:
```
(13532, 8)
```

In [7]:
```
#table1.to_csv("online.csv", index=False)
```

Read data from the `stock_description` table

In [8]:
```
# Write the query to be executed that selects everything from the online_retail_history table.

table2 = pd.read_sql_query('SELECT * FROM stock_description', db)
table2
```

```
# Use the read_sql function in pandas to read a query into a DataFrame.


# Preview the first five rows of the data.

table2.head()
```

Out[8]:

	StockCode	Description
0	10002	INFLATABLE POLITICAL GLOBE
1	10080	GROOVY CACTUS INFLATABLE
2	10120	DOGGY RUBBER
3	10123C	HEARTS WRAPPING TAPE
4	10124A	SPOTS ON RED BOOKCOVER TAPE

In [9]:
```
# Get the shape of the data.

table2.head()
```

Out[9]:

	StockCode	Description
0	10002	INFLATABLE POLITICAL GLOBE
1	10080	GROOVY CACTUS INFLATABLE
2	10120	DOGGY RUBBER
3	10123C	HEARTS WRAPPING TAPE
4	10124A	SPOTS ON RED BOOKCOVER TAPE

In [10]:
```
#table2.to_csv("desc.csv", index=False)
```

Aggregate the `online_retail_history` and `stock_description` datasets

In [11]:
```
# Write a query to aggregate the two datasets so that you have the stock descriptions as well as the stock code

transactions_agg = pd.merge(left=table1, right=table2, how='inner', on='StockCode')

# Use the read_sql function in pandas to read a query into a DataFrame.


# Preview the first five rows of the data.

transactions_agg.head()
```

Out[11]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
1	536373	85123A	6	2010-12-01 09:02:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
2	536375	85123A	6	2010-12-01 09:32:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
3	536390	85123A	64	2010-12-01 10:19:00	2.55	17511	United Kingdom	163.2	CREAM HANGING HEART T-LIGHT HOLDER
4	536394	85123A	32	2010-12-01 10:39:00	2.55	13408	United Kingdom	81.6	CREAM HANGING HEART T-LIGHT HOLDER

In [12]:
```
# Get the shape of the data.

transactions_agg.shape
```

Out[12]:
```
(13532, 9)
```

In [13]:
```
#transactions_agg.to_csv("combined.csv", index=False)
```

Identify and fix corrupt or unusable data

In [14]:
```
# Check the value counts of the "Description" field.

transactions_agg["Description"].value_counts()
```

Out[14]:
```
CREAM HANGING HEART T-LIGHT HOLDER    2174
JUMBO BAG RED RETROSPOT               1960
                                      1761
REGENCY CAKESTAND 3 TIER              1711
PARTY BUNTING                         1615
LUNCH BAG RED RETROSPOT               1621
ASSORTED COLOUR BIRD ORNAMENT         1405
SET OF 3 CAKE TINS PANTRY DESIGN      1297
LUNCH BAG  BLACK SKULL.               1271
SET OF 3 CAKE TINS PANTRY DESIGN      1257
PACK OF 72 RETROSPOT CAKE CASES       1178
Name: Description, dtype: int64
```

In [15]:
```
# Remove rows where "Description" is just a question mark (?).

table3 = transactions_agg[transactions_agg["Description"] != "?"]

table3["Description"].value_counts()
```

```
# Preview the first five rows of the data.


table3.head()
```

Out[15]:
```
CREAM HANGING HEART T-LIGHT HOLDER    2174
JUMBO BAG RED RETROSPOT               1960
REGENCY CAKESTAND 3 TIER              1711
PARTY BUNTING                         1615
LUNCH BAG RED RETROSPOT               1621
ASSORTED COLOUR BIRD ORNAMENT         1405
POSTAGE                               1329
LUNCH BAG  BLACK SKULL.               1271
SET OF 3 CAKE TINS PANTRY DESIGN      1257
PACK OF 72 RETROSPOT CAKE CASES       1178
Name: Description, dtype: int64
```

In [16]:
```
table3.head()
```

Out[16]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
1	536373	85123A	6	2010-12-01 09:02:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
2	536375	85123A	6	2010-12-01 09:32:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
3	536390	85123A	64	2010-12-01 10:19:00	2.55	17511	United Kingdom	163.2	CREAM HANGING HEART T-LIGHT HOLDER
4	536394	85123A	32	2010-12-01 10:39:00	2.55	13408	United Kingdom	81.6	CREAM HANGING HEART T-LIGHT HOLDER

Identify and remove duplicates

In [17]:
```
# Identify all duplicated data.

duplicated_data = table3.duplicated(keep=False)

duplicated_data.shape[0]
```

Out[17]:
```
223
```

In [18]:
```
# Print the duplicated data.

print(duplicated_data)
```

```
      Invoice StockCode  Quantity         InvoiceDate  Price CustomerID  \
288    540953    85123A         3  2011-01-12 13:16:00   2.95      14587
290    540953    85123A         1  2011-01-12 13:16:00   2.95      14587
309    541160    85123A         3  2011-01-13 12:23:00   2.95      17787
331    541660    85123A         3  2011-01-20 12:20:00   2.95      17787
358    542239    85123A         2  2011-01-26 14:35:00   2.95      17784
...       ...       ...       ...                  ...    ...        ...
13929  576778     22720         3  2011-11-16 13:25:00   4.95      14554
14948  577232     22720         3  2011-11-21 11:28:00   4.95      13503
14949  577473     22720         1  2011-11-23 11:28:00   4.95      13503
14950  577596     22720         2  2011-11-23 12:36:00   4.95      14159
14951  577596     22720         2  2011-11-23 12:36:00   4.95      14159

                Country  TotalAmount                        Description
288      United Kingdom         8.85  CREAM HANGING HEART T-LIGHT HOLDER
290      United Kingdom         2.95  CREAM HANGING HEART T-LIGHT HOLDER
330      United Kingdom         8.85  CREAM HANGING HEART T-LIGHT HOLDER
331      United Kingdom         8.85  CREAM HANGING HEART T-LIGHT HOLDER
358      United Kingdom         5.90  CREAM HANGING HEART T-LIGHT HOLDER
...                 ...          ...                                 ...
13929    United Kingdom        14.85    SET OF 3 CAKE TINS PANTRY DESIGN
14948    United Kingdom         4.95    SET OF 3 CAKE TINS PANTRY DESIGN
14949    United Kingdom         4.95    SET OF 3 CAKE TINS PANTRY DESIGN
14950    United Kingdom         9.90    SET OF 3 CAKE TINS PANTRY DESIGN
14951    United Kingdom         9.90    SET OF 3 CAKE TINS PANTRY DESIGN

[223 rows x 9 columns]
```

In [19]:
```
# Remove the duplicated data.

table4 = table3[~table3.duplicated()]
table4
```

```
# Preview the first five rows of the data.


table4.head()
```

Out[19]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
1	536373	85123A	6	2010-12-01 09:02:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
2	536375	85123A	6	2010-12-01 09:32:00	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
3	536390	85123A	64	2010-12-01 10:19:00	2.55	17511	United Kingdom	163.2	CREAM HANGING HEART T-LIGHT HOLDER
4	536394	85123A	32	2010-12-01 10:39:00	2.55	13408	United Kingdom	81.6	CREAM HANGING HEART T-LIGHT HOLDER

Correct date formats

In [20]:
```
# Get the data types for every column in the DataFrame.

table4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13256 entries, 0 to 17951
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Invoice      13256 non-null  object
 1   StockCode    13256 non-null  object
 2   Quantity     13256 non-null  int64
 3   InvoiceDate  13256 non-null  object
 4   Price        13256 non-null  float64
 5   CustomerID   12455 non-null  object
 6   Country      13256 non-null  object
 7   TotalAmount  13256 non-null  float64
 8   Description  13256 non-null  object
dtypes: float64(2), int64(1), object(6)
memory usage: 1.0+ MB
```

In [21]:
```
# Convert "InvoiceDate" to a "%Y-%m-%d" datetime format.

table4["InvoiceDate"] = pd.to_datetime(table4["InvoiceDate"]).dt.strftime("%Y-%m-%d")
```

```
C:\Users\Dennis\AppData\Local\Temp\ipykernel_7396\2658595164.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  table4["InvoiceDate"] = pd.to_datetime(table4["InvoiceDate"]).dt.strftime("%Y-%m-%d")
```

In [22]:
```
table4["InvoiceDate"]
```

Out[22]:
```
0        2010-12-01
1        2010-12-01
2        2010-12-01
3        2010-12-01
4        2010-12-01
            ...
17027    2011-12-08
17028    2011-12-08
17029    2011-12-08
17030    2011-12-08
17031    2011-12-09
Name: InvoiceDate, Length: 13256, dtype: object
```

In [23]:
```
# Convert "InvoiceDate" to a "%Y-%m-%d" datetime format.

table4["InvoiceDate"] = pd.to_datetime(table4["InvoiceDate"])
```

```
C:\Users\Dennis\AppData\Local\Temp\ipykernel_7396\2658790773.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  table4["InvoiceDate"] = pd.to_datetime(table4["InvoiceDate"])
```

In [24]:
```
# Get the data types for every column in the converted DataFrame.

table4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13256 entries, 0 to 17951
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Invoice      13256 non-null  object
 1   StockCode    13256 non-null  object
 2   Quantity     13256 non-null  int64
 3   InvoiceDate  13256 non-null  datetime64[ns]
 4   Price        13256 non-null  float64
 5   CustomerID   12455 non-null  object
 6   Country      13256 non-null  object
 7   TotalAmount  13256 non-null  float64
 8   Description  13256 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 1.0+ MB
```

Examine the table before finishing

In [25]:
```
# Preview the first five rows of the data.

table4.head()
```

Out[25]:

	Invoice	StockCode	Quantity	InvoiceDate	Price	CustomerID	Country	TotalAmount	Description
0	536365	85123A	6	2010-12-01	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
1	536373	85123A	6	2010-12-01	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
2	536375	85123A	6	2010-12-01	2.55	17850	United Kingdom	15.3	CREAM HANGING HEART T-LIGHT HOLDER
3	536390	85123A	64	2010-12-01	2.55	17511	United Kingdom	163.2	CREAM HANGING HEART T-LIGHT HOLDER
4	536394	85123A	32	2010-12-01	2.55	13408	United Kingdom	81.6	CREAM HANGING HEART T-LIGHT HOLDER

Load the dataset into a pickle file

In [26]:
```
# Save the dataset as a pickle file named online_history_cleaned.pickle.

table4.to_pickle("online_history_cleaned.pickle")
```

In [27]:
```
# Close any connections to the database.

db.close()
```

In [ ]:
```

```