In [3]:
```
import pandas as pd
import numpy as np
import sqlalchemy
```

Connect python to sql server

In [4]:
```
engine = sqlalchemy.create_engine(
    "mssql+pyodbc://@DESKTOP-H6EN8UM/superstore?driver=ODBC+Driver+17+for+SQL+Serve
)
```

Extract the data from the excel file

In [9]:
```
df = pd.read_excel("cleaned superstore.xlsx", sheet_name="cleaned superstore")
```

In [16]:
```
df.columns
```

Out[16]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

Transform

Dimension and Fact tables

In [48]:
```
Dim_Orders = df[["Order ID", "Order Date", "Ship Date", "Ship Mode",
    "Country", "City", "State", "Postal Code", "Region"]].drop_duplicates()
```

In [50]:
```
Dim_Customer = df[["Customer ID", "Customer Name", "Segment"]].drop_duplicates()
```

In [58]:
```
Dim_Product = df[["Product ID", "Product Name", "Sub-Category", "Category"]].drop_d
```

In [60]:
```
Dim_Product = Dim_Product.drop_duplicates(subset=["Product ID"])
```

In [62]:
```
Fact_Sales = df[["Row ID", "Order ID", "Customer ID", "Product ID", "Sales",
                 "Quantity", "Discount", "Profit"]]
```

Rename the Columns to match column names in star schema in sql server

In [74]:
```
Dim_Orders = Dim_Orders.rename(columns={"Order ID":"Order_ID", "Order Date": "Order
                                        "Ship Date":"Ship_Date", "Ship Mode": "Ship_
```

In [86]:
```
Dim_Customer = Dim_Customer.rename(columns={"Customer ID":"Customer_ID", "Customer
```

In [91]:
```
Dim_Product = Dim_Product.rename(columns={"Product ID":"Product_ID","Product Name":
                                          "Sub-Category":"Sub_Category"})
```

In [96]:
```
Fact_Sales = Fact_Sales.rename(columns={"Row ID":"Fact_ID", "Order ID":"Order_ID",
                            "Product ID":"Product_ID"})
```

In [56]:
```
df["Product Name"].nunique()
```

Out[56]:
```
1850
```

In [29]:
```
df["Product ID"].value_counts()
```

Out[29]:
```
Product ID
OFF-PA-10001970    19
TEC-AC-10003832    18
FUR-FU-10004270    16
FUR-CH-10002647    15
FUR-CH-10001146    15
                   ..
TEC-MA-10004552     1
TEC-MA-10003589     1
OFF-AP-10003099     1
TEC-PH-10002645     1
OFF-ST-10001627     1
Name: count, Length: 1862, dtype: int64
```

In [63]:
```
inspector = sqlalchemy.inspect(engine)
```

In [64]:
```
tables = inspector.get_table_names()
```

In [65]:
```
tables
```

Out[65]:
```
['customer', 'Orders', 'Products', 'Sales']
```

In [84]:
```
for table in tables:
    print(f"\nTable: {table}")
    columns = inspector.get_columns(table)
    for col in columns:
        print(f" - {col['name']} ({col['type']})")
```

```
Table: customer
 - Customer_ID (INTEGER)
 - Customer_Name (VARCHAR(50) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Segment (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")

Table: Orders
 - Order_id (INTEGER)
 - Order_Date (DATE)
 - Ship_Date (DATE)
 - Postal_code (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - City (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Stat_e (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Region (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Country (VARCHAR(20) COLLATE "SQL_Latin1_General_CP1_CI_AS")

Table: Products
 - Product_id (INTEGER)
 - Product_Name (VARCHAR(100) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Sub_category (VARCHAR(50) COLLATE "SQL_Latin1_General_CP1_CI_AS")
 - Category (VARCHAR(50) COLLATE "SQL_Latin1_General_CP1_CI_AS")

Table: Sales
 - Fact_ID (INTEGER)
 - Order_ID (INTEGER)
 - Customer_ID (INTEGER)
 - Product_ID (INTEGER)
 - Sales (DECIMAL(10, 2))
 - Quantity (INTEGER)
 - Discount (DECIMAL(5, 2))
 - Profit (DECIMAL(10, 2))
```

Load

In [78]:
```
Dim_Orders.to_sql("Orders", engine, if_exists="append", index=False)
```

Out[78]: 116

In [ ]:
```
Dim_Customer.to_sql("Customer", engine, if_exists="append", index=False)
```

In [93]:
```
Dim_Product.to_sql("Products", engine, if_exists="append", index=False)
```

Out[93]: 290

In [97]:
```
Fact_Sales.to_sql("Sales", engine, if_exists="append", index=False)
```

Out[97]: 38

In [ ]:
```

```