In [2]:

```python
import pandas as pd

df = pd.read_csv("Downloads/customer_shopping_behavior.csv")
df.head()
```

Out[2]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Cc
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	G
1	2	19	Male	Sweater	Clothing	64	Maine	L	Marc
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Marc
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Marc
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquc

In [29]:

```python
#Check Null values

df.isnull().sum()
```

Out[29]:

```
Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating             37
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```

In [43]:

```python
df["Review Rating"] = df["Review Rating"].fillna(df["Review Rating"].median())
```

In [44]:

```python
df.isnull().sum()
```

Out[44]:

```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

In [45]:

```
#Alter Columns names

df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(" ","_")
df = df.rename(columns={"purchase_amount_(usd)":"purchase_amount"})
df.columns
```

Out[45]:

```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

In [46]:

```
# create a column age_group
labels = ["Young-Adult", "Adult", "Middle-aged", "Senior"]
df["age_group"] = pd.qcut(df["age"], q=4, labels = labels)
df[["age","age_group"]].head(10)
```

Out[46]:

	age	age_group
0	55	Middle-aged
1	19	Young-Adult
2	50	Middle-aged
3	21	Young-Adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young-Adult
8	26	Young-Adult
9	57	Middle-aged

In [47]:

```
# create column purchase_frequency_days

frequency_mapping = {
    "Fortnightly":14,
    "Weekly" : 7,
    "Monthly":30,
    "Quarterly": 90,
    "Bi-Weekly": 14,
    "Annually":365,
    "Every 3 months":90
}
df["purchase_frequency_days"] = df["frequency_of_purchases"].map(frequency_mappi
df[["purchase_frequency_days","frequency_of_purchases"]].head(10)
```

Out[47]:

	purchase_frequency_days	**frequency_of_purchases**
0	14.0	Fortnightly
1	14.0	Fortnightly
2	7.0	Weekly
3	7.0	Weekly
4	365.0	Annually
5	7.0	Weekly
6	90.0	Quarterly
7	7.0	Weekly
8	365.0	Annually
9	90.0	Quarterly

In [48]:

```
#Check whether discount and promo code columns contains same value

df[["discount_applied","promo_code_used"]].head(10)
```

Out[48]:

	discount_applied	**promo_code_used**
0	Yes	Yes
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes
9	Yes	Yes

In [49]:

```
(df["discount_applied"] == df["promo_code_used"]).all()
```

Out[49]:

```
np.True_
```

In [50]:

```
# create promo code column as it is same as discount applied coulmn

df = df.drop("promo_code_used",axis = 1)
```

In [51]:

```
df.columns
```

Out[51]:

```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
      dtype='object')
```

In [1]:

```
pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\shreyas\appdata\local
\programs\python\python313\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\shreyas\appdata\local\progr
ams\python\python313\lib\site-packages (2.0.44)
Requirement already satisfied: greenlet>=1 in c:\users\shreyas\appdata\local\prog
rams\python\python313\lib\site-packages (from sqlalchemy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\shreyas\appda
ta\local\programs\python\python313\lib\site-packages (from sqlalchemy) (4.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:

```
# --- Step 1: Import Libraries ---
import pandas as pd
from sqlalchemy import create_engine
import urllib.parse

# --- Step 2: Database connection details ---
username = "postgres"
password = "Mamta@5566"
host = "localhost"
port = "5432"
database = "Customer_behavior"

# --- Step 3: Encode password ---
encoded_password = urllib.parse.quote_plus(password)

# --- Step 4: Create engine ---
engine = create_engine(f"postgresql+psycopg2://{username}:{encoded_password}@{hc

# --- Step 5: Load CSV into DataFrame ---
df = pd.read_csv("Downloads/customer_shopping_behavior.csv")

# --- Step 6: Make edits / cleaning in Jupyter ---
# Fill missing review ratings with the median
df["Review Rating"] = df["Review Rating"].fillna(df["Review Rating"].median())

# Check missing values (optional)
df.isnull().sum()

# Standardize column names
df.columns = df.columns.str.lower().str.replace(" ", "_")
```

```
df = df.rename(columns={"purchase_amount_(usd)": "purchase_amount"})

# Map frequency values to numeric days
frequency_mapping = {
    "Fortnightly": 14,
    "Weekly": 7,
    "Monthly": 30,
    "Quarterly": 90,
    "Bi-Weekly": 14,
    "Annually": 365,
    "Every 3 months": 90
}
df["purchase_frequency_days"] = df["frequency_of_purchases"].map(frequency_mappi

# create a column age_group
labels = ["Young-Adult", "Adult", "Middle-aged", "Senior"]
df["age_group"] = pd.qcut(df["age"], q=4, labels = labels)
df[["age","age_group"]].head(10)

# Drop duplicate / unnecessary columns
if "promo_code_used" in df.columns:
    df = df.drop("promo_code_used", axis=1)

# --- Step 7: Export final DataFrame to PostgreSQL ---
table_name = "customer"
try:
    df.to_sql(table_name, engine, if_exists="replace", index=False)
    print(f"✅ Data successfully loaded into table '{table_name}' in database '
except Exception as e:
    print("❌ Data load failed:", e)
```

```
✅ Data successfully loaded into table 'customer' in database 'Customer_behavio
r'.
```