In [1]:

```
import pandas as pd
import numpy as np
df = pd.read_csv('C:/Users/rubyb/Desktop/CustomerShoppingBehavior/customer_shopping_behavior.csv')
```

In [2]:

```
df
```

Out[2]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3895	3896	40	Female	Hoodie	Clothing	28	Virginia	L	Turquoise	Summer	4.2	No	2-Day Shipping
3896	3897	52	Female	Backpack	Accessories	49	Iowa	L	White	Spring	4.5	No	Store Pickup
3897	3898	46	Female	Belt	Accessories	33	New Jersey	L	Green	Spring	2.9	No	Standard
3898	3899	44	Female	Shoes	Footwear	77	Minnesota	S	Brown	Summer	3.8	No	Express
3899	3900	52	Female	Handbag	Accessories	81	California	M	Beige	Spring	3.1	No	Store Pickup

3900 rows × 18 columns

In [3]:

```
df.head()
```

Out[3]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Disc Ap
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             3900 non-null   int64  
 1   Age                     3900 non-null   int64  
 2   Gender                  3900 non-null   object 
 3   Item Purchased          3900 non-null   object 
 4   Category                3900 non-null   object 
 5   Purchase Amount (USD)   3900 non-null   int64  
 6   Location                3900 non-null   object 
 7   Size                    3900 non-null   object 
 8   Color                   3900 non-null   object 
 9   Season                  3900 non-null   object 
 10  Review Rating           3863 non-null   float64
 11  Subscription Status     3900 non-null   object 
 12  Shipping Type           3900 non-null   object 
 13  Discount Applied        3900 non-null   object 
 14  Promo Code Used         3900 non-null   object 
 15  Previous Purchases      3900 non-null   int64  
 16  Payment Method          3900 non-null   object 
 17  Frequency of Purchases  3900 non-null   object 
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

```
In [5]: df.describe()
```

Out[5]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

In [6]:

```
df.describe(include='all')
```

Out[6]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN

In [7]:

```
# Checking if missing data or null values are present in the dataset

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

Out[7]:

```
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 [8]:

```
# Imputing missing values in Review Rating column with the median rating of the product category.
# Median is used because it is immune to outliers unlike mean

df['Review Rating'] = df.groupby('Category')['Review Rating'].transform(lambda x: x.fillna(x.median()))
```

In [9]:

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

Out[9]:

```
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 [10]:

```
# Renaming columns according to snake casing for better readability and documentation

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

In [11]:

```
df.columns
```

Out[11]:

```
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 [12]:

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

In [13]:

```
df[['age','age_group']].head(10)
```

Out[13]:

	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 [14]:

```
# create new 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_mapping)
```

In [15]:

```
df[['purchase_frequency_days','frequency_of_purchases']].head(10)
```

Out[15]:

	purchase_frequency_days	frequency_of_purchases
0	14	Fortnightly
1	14	Fortnightly
2	7	Weekly
3	7	Weekly
4	365	Annually
5	7	Weekly
6	90	Quarterly
7	7	Weekly
8	365	Annually
9	90	Quarterly

In [16]:

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

Out[16]:

	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 [17]:

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

Out[17]:

```
np.True_
```

In [18]:

```
# Dropping promo code used column

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

In [19]:

```
df.columns
```

Out[19]:

```
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 [20]:

```
#POSTGRESSQL Connection
```

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

```
Requirement already satisfied: psycopg2-binary in c:\users\rubyb\anaconda3\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\rubyb\anaconda3\lib\site-packages (1.4.54)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\rubyb\anaconda3\lib\site-packages (from sqlalchemy) (3.1.1)
```

In [21]:

```
#MS SQL Connection

!pip install sqlalchemy pyodbc
```

```
Requirement already satisfied: sqlalchemy in c:\users\rubyb\anaconda3\lib\site-packages (1.4.54)
Requirement already satisfied: pyodbc in c:\users\rubyb\anaconda3\lib\site-packages (5.2.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\rubyb\anaconda3\lib\site-packages (from sqlalchemy) (3.1.1)
```

In [23]:

```
!pip install python-dotenv
```

```
Requirement already satisfied: python-dotenv in c:\users\rubyb\anaconda3\lib\site-packages (1.1.0)
```

In [35]:

```
import pandas as pd
from sqlalchemy import create_engine
from urllib.parse import quote_plus
import pyodbc
import getpass


host = "localhost"
port = "1433"
database = "customer_behavior"

driver = quote_plus("ODBC Driver 18 for SQL Server")

connection_string = f"mssql+pyodbc://{host},{port}/{database}?driver={driver}&trusted_connection=yes&TrustServerCertificate=ye
engine = create_engine(connection_string)

df.to_sql("customer", engine, if_exists="replace", index=False)

# Read back sample (SQL Server uses TOP instead of LIMIT)
pd.read_sql("SELECT TOP 5 * FROM customer;", engine)
```

Out[35]:

	customer_id	age	gender	item_purchased	category	purchase_amount	location	size	color	season	review_rating	subscrip
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	

In [ ]:

```
print("✅ Connection string syntax is now correct. The 'engine' object has been created.")
```

In [ ]: