In [1]:
```
import pandas as pd
```

In [2]:
```
!pip install mysql-connector-python
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: mysql-connector-python in c:\users\itsja\appdata\roaming\python\python312\site-packages
(9.3.0)
```

In [3]:
```
#connecting with mysql Database
import mysql.connector
from sqlalchemy import create_engine
```

In [4]:
```
engine = create_engine('mysql+mysqlconnector://root:Sharma03@localhost/ecommerce_data')
```

In [5]:
```
conn = mysql.connector.connect(host="localhost",
                               user="root",
                               password="Sharma03",
                               database="ecommerce_data")
```

In [6]:
```
query = 'select * from data_new'
```

In [7]:
```
df = pd.read_sql(query,engine)
df
```

Out[7]:

	Order_ID	Product_ID	User_ID	Order_Date	Return_Date	Product_Category	Product_Price	Order_Quantity	Re
0	ORD00002000	PROD00002000	USER00002000	2024-08-31	2024-08-20	Books	332.72	1	Cl
1	ORD00002002	PROD00002002	USER00002002	2023-08-29	2024-12-20	Clothing	158.02	5	
2	ORD00002003	PROD00002003	USER00002003	2023-06-20	2023-03-07	Home	122.15	2	
3	ORD00002004	PROD00002004	USER00002004	2023-08-03	2024-05-20	Electronics	482.22	3	
4	ORD00002007	PROD00002007	USER00002007	2024-01-11	2023-03-15	Toys	83.23	3	
...	...	...	...	...	...	...	...	...	
1471	ORD00004990	PROD00004990	USER00004990	2023-06-14	2024-05-05	Home	342.64	5	Cl
1472	ORD00004992	PROD00004992	USER00004992	2024-04-06	2023-06-22	Books	258.17	3	
1473	ORD00004996	PROD00004996	USER00004996	2023-11-29	2023-03-04	Toys	374.12	4	Cl
1474	ORD00004997	PROD00004997	USER00004997	2023-07-13	2023-03-27	Toys	208.99	4	Cl
1475	ORD00004999	PROD00004999	USER00004999	2023-10-20	2024-11-27	Toys	194.27	4	

1476 rows × 17 columns

In [8]:
```
# Overview of columns and data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1476 entries, 0 to 1475
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Order_ID          1476 non-null   object
 1   Product_ID        1476 non-null   object
 2   User_ID           1476 non-null   object
 3   Order_Date        1476 non-null   object
 4   Return_Date       1476 non-null   object
 5   Product_Category  1476 non-null   object
 6   Product_Price     1476 non-null   float64
 7   Order_Quantity    1476 non-null   int64
 8   Return_Reason     1476 non-null   object
 9   Return_Status     1476 non-null   object
 10  Days_to_Return    1476 non-null   int64
 11  User_Age          1476 non-null   int64
 12  User_Gender       1476 non-null   object
 13  User_Location     1476 non-null   object
 14  Payment_Method    1476 non-null   object
 15  Shipping_Method   1476 non-null   object
 16  Discount_Applied  1476 non-null   float64
dtypes: float64(2), int64(3), object(12)
memory usage: 196.2+ KB
```

In [9]:
```
# Check for missing values
df.isnull().sum()
```

Out[9]:
```
Order_ID            0
Product_ID          0
User_ID             0
Order_Date          0
Return_Date         0
Product_Category    0
Product_Price       0
Order_Quantity      0
Return_Reason       0
Return_Status       0
Days_to_Return      0
User_Age            0
User_Gender         0
User_Location       0
Payment_Method      0
Shipping_Method     0
Discount_Applied    0
dtype: int64
```

In [10]:
```
df['Product_Category'].unique()
```

Out[10]:
```
array(['Books', 'Clothing', 'Home', 'Electronics', 'Toys'], dtype=object)
```

In [11]:
```
df['User_Location'].unique()
```

Out[11]:
```
array(['City33', 'City60', 'City2', 'City53', 'City14', 'City84',
       'City79', 'City23', 'City37', 'City36', 'City22', 'City8',
       'City25', 'City5', 'City27', 'City97', 'City76', 'City56',
       'City38', 'City19', 'City59', 'City41', 'City21', 'City20',
       'City100', 'City1', 'City75', 'City86', 'City67', 'City24',
       'City13', 'City81', 'City45', 'City10', 'City74', 'City50',
       'City69', 'City26', 'City78', 'City63', 'City87', 'City40',
       'City51', 'City7', 'City91', 'City32', 'City83', 'City89',
       'City62', 'City72', 'City94', 'City65', 'City61', 'City96',
       'City12', 'City42', 'City47', 'City88', 'City16', 'City80',
       'City95', 'City66', 'City43', 'City34', 'City6', 'City17',
       'City11', 'City44', 'City39', 'City93', 'City28', 'City68',
       'City85', 'City48', 'City46', 'City3', 'City4', 'City35', 'City90',
       'City31', 'City73', 'City55', 'City9', 'City57', 'City52',
       'City92', 'City18', 'City82', 'City49', 'City98', 'City58',
       'City29', 'City70', 'City30', 'City15', 'City71', 'City77',
       'City99', 'City54', 'City64'], dtype=object)
```

In [12]:
```
df['Shipping_Method'].unique()
```

Out[12]:
```
array(['Standard', 'Next-Day', 'Express'], dtype=object)
```

In [13]:
```
df['Payment_Method'].unique()
```

Out[13]:
```
array(['Credit Card', 'PayPal', 'Gift Card', 'Debit Card'], dtype=object)
```

In [14]:
```
df.shape[0]
```

Out[14]:
```
1476
```

In [15]:
```
df.columns
```

Out[15]:
```
Index(['Order_ID', 'Product_ID', 'User_ID', 'Order_Date', 'Return_Date',
       'Product_Category', 'Product_Price', 'Order_Quantity', 'Return_Reason',
       'Return_Status', 'Days_to_Return', 'User_Age', 'User_Gender',
       'User_Location', 'Payment_Method', 'Shipping_Method',
       'Discount_Applied'],
      dtype='object')
```

In [16]:
```
type(df)
```

Out[16]:
```
pandas.core.frame.DataFrame
```

In [17]:
```
len(df.columns)
```

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

In [18]:
```
df.drop_duplicates()
```

Out[18]:

	Order_ID	Product_ID	User_ID	Order_Date	Return_Date	Product_Category	Product_Price	Order_Quantity	Re
0	ORD00002000	PROD00002000	USER00002000	2024-08-31	2024-08-20	Books	332.72	1	Cl
1	ORD00002002	PROD00002002	USER00002002	2023-08-29	2024-12-20	Clothing	158.02	5	
2	ORD00002003	PROD00002003	USER00002003	2023-06-20	2023-03-07	Home	122.15	2	
3	ORD00002004	PROD00002004	USER00002004	2023-08-03	2024-05-20	Electronics	482.22	3	
4	ORD00002007	PROD00002007	USER00002007	2024-01-11	2023-03-15	Toys	83.23	3	
...	...	...	...	...	...	...	...	...	
1471	ORD00004990	PROD00004990	USER00004990	2023-06-14	2024-05-05	Home	342.64	5	Cl
1472	ORD00004992	PROD00004992	USER00004992	2024-04-06	2023-06-22	Books	258.17	3	
1473	ORD00004996	PROD00004996	USER00004996	2023-11-29	2023-03-04	Toys	374.12	4	Cl
1474	ORD00004997	PROD00004997	USER00004997	2023-07-13	2023-03-27	Toys	208.99	4	Cl
1475	ORD00004999	PROD00004999	USER00004999	2023-10-20	2024-11-27	Toys	194.27	4	

1476 rows × 17 columns

In [19]:
```
df.describe()
```

Out[19]:

	Product_Price	Order_Quantity	Days_to_Return	User_Age	Discount_Applied
count	1476.000000	1476.000000	1476.000000	1476.000000	1476.000000
mean	247.709411	3.043360	-4.964092	44.394986	25.004485
std	141.480787	1.426914	296.153711	15.613569	14.406271
min	5.190000	1.000000	-673.000000	18.000000	0.010000
25%	122.617500	2.000000	-218.000000	31.000000	12.635000
50%	244.135000	3.000000	-5.500000	45.000000	24.670000
75%	367.520000	4.000000	208.250000	58.000000	37.312500
max	499.710000	5.000000	726.000000	70.000000	49.930000

In [20]:
```
df.describe(include='object')
```

Out[20]:

	Order_ID	Product_ID	User_ID	Order_Date	Return_Date	Product_Category	Return_Reason	Return_Status	U
count	1476	1476	1476	1476	1476	1476	1476	1476	
unique	1476	1476	1476	638	622	5	4	1	
top	ORD00002000	PROD00002000	USER00002000	2023-09-03	2024-08-20	Books	Defective	Returned	
freq	1	1	1	9	8	308	404	1476	

In [21]:
```
df.isnull().sum()
```

Out[21]:
```
Order_ID            0
Product_ID          0
User_ID             0
Order_Date          0
Return_Date         0
Product_Category    0
Product_Price       0
Order_Quantity      0
Return_Reason       0
Return_Status       0
Days_to_Return      0
User_Age            0
User_Gender         0
User_Location       0
Payment_Method      0
Shipping_Method     0
Discount_Applied    0
dtype: int64
```

In [22]:
```
df.duplicated().sum()
```

Out[22]:
```
0
```

In [23]:
```
import seaborn as sns
import matplotlib as mat
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
mat.rcParams['font.size'] = 14
mat.rcParams['figure.figsize'] = (12,6)
mat.rcParams['figure.facecolor'] = '#00000000'
```

In [24]:
```
df['Product_Category'] = df['Product_Category'].str.lower().str.strip()
print(df['Product_Category'])
```

```
0             books
1          clothing
2              home
3       electronics
4              toys
           ...
1471           home
1472          books
1473           toys
1474           toys
1475           toys
Name: Product_Category, Length: 1476, dtype: object
```

In [25]:
```
df['Return_Reason'] = df['Return_Reason'].str.lower().str.strip()
```

In [26]:
```
df['Return_Status'] = df['Return_Status'].str.lower().str.strip()
```

In [27]:
```
df['User_Gender'] = df['User_Gender'].str.lower().str.strip()
```

In [28]:
```
df['User_Location'] = df['User_Location'].str.lower().str.strip()
```

In [29]:
```
df['Payment_Method'] = df['Payment_Method'].str.lower().str.strip()
```

In [30]:
```
df['Shipping_Method'] = df['Shipping_Method'].str.lower().str.strip()
```

In [31]:
```
df['is_returned'] = df['Return_Status'].apply(lambda x: 1 if str(x).lower().strip() == 'returned' else 0)
```

In [32]:
```
category_return_rate = df.groupby('Product_Category')['is_returned'].mean()
```

In [33]:
```
region_return_rate = df.groupby('User_Location')['is_returned'].mean()
```

In [34]:
```
df_encoded = pd.get_dummies(df, columns=['Product_Category', 'User_Location'], drop_first=True)
```

In [35]:
```
df['return_items'] = df['Order_Quantity'] - 1
df['return_items']
```

Out[35]:
```
0       0
1       4
2       1
3       2
4       2
       ..
1471    4
1472    2
1473    3
1474    3
1475    3
Name: return_items, Length: 1476, dtype: int64
```

In [36]:
```
dff = df.head(10)
dff
```

Out[36]:

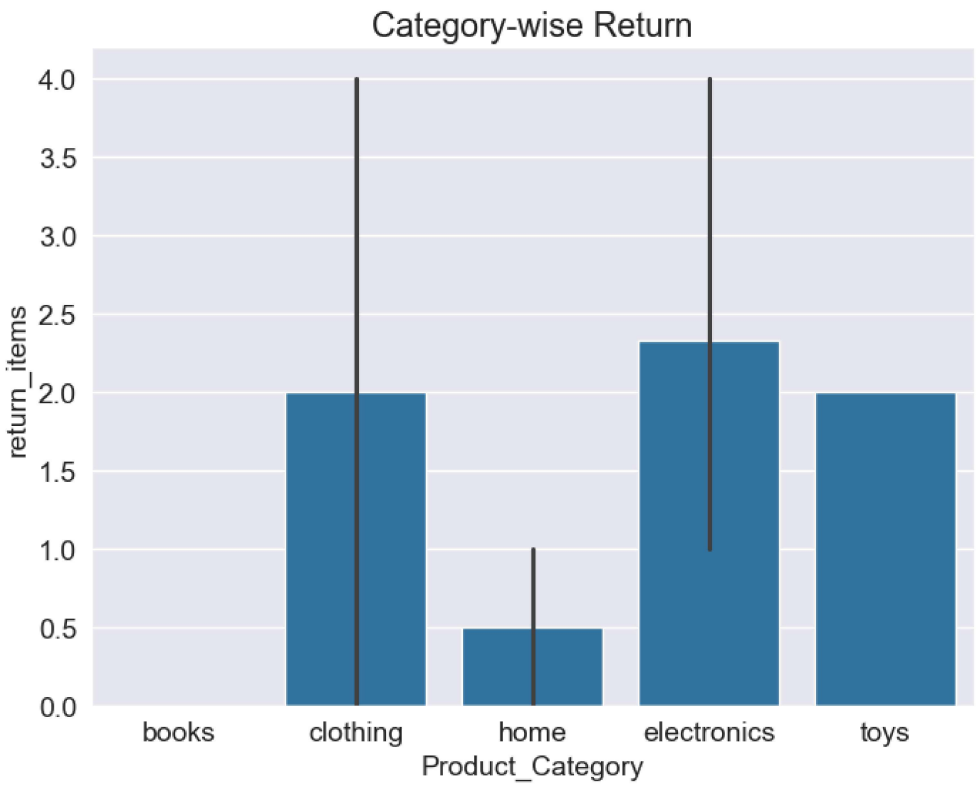
	Order_ID	Product_ID	User_ID	Order_Date	Return_Date	Product_Category	Product_Price	Order_Quantity	Return
0	ORD00002000	PROD00002000	USER00002000	2024-08-31	2024-08-20	books	332.72	1	chang
1	ORD00002002	PROD00002002	USER00002002	2023-08-29	2024-12-20	clothing	158.02	5	
2	ORD00002003	PROD00002003	USER00002003	2023-06-20	2023-03-07	home	122.15	2	
3	ORD00002004	PROD00002004	USER00002004	2023-08-03	2024-05-20	electronics	482.22	3	wr
4	ORD00002007	PROD00002007	USER00002007	2024-01-11	2023-03-15	toys	83.23	3	wr
5	ORD00002008	PROD00002008	USER00002008	2023-07-14	2023-09-14	home	167.63	1	
6	ORD00002010	PROD00002010	USER00002010	2024-04-17	2023-12-13	electronics	106.54	5	c
7	ORD00002011	PROD00002011	USER00002011	2024-06-02	2023-07-01	clothing	391.61	3	c
8	ORD00002015	PROD00002015	USER00002015	2024-07-22	2023-10-15	clothing	443.96	1	c
9	ORD00002016	PROD00002016	USER00002016	2023-04-28	2024-05-29	electronics	395.66	2	wr

In [37]:

```
import seaborn as sns
import matplotlib.pyplot as plt

mat.rcParams['figure.figsize'] = (8, 6)
plt.title('Category-wise Return')

# Plot 'is_returned' on x-axis and 'Product_Category' on y-axis
sns.barplot(x='Product_Category', y='return_items', data=dff)
plt.show()
```



In [ ]: