Project 1: Sales Analysis

We will analyse sales by demographic Analysis of customers e.g. city, age, gender etc. The goal of this process is to give more information about our data so that the marketing team prepares to intensify the efficiency based on the data and information we will provide!

Goals:

(1) **Helping marketing team decide the most valuable age group to target for increasing sales.**
(2) **To identify the type of customers with most purchasing powers.**
(3) **To identify the cities with most sales made.**
(4) **Finding the type of most reluctant residents in each city and help marketing team target these groups to increase sales.**

We will be analysing the following six parameters:

- Gender
- Age
- City
- Stability
- Occupation
- Products

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
import seaborn as sns
from collections import Counter
%matplotlib inline
import os
print(os.listdir("../workspace"))
```

```
['.ipynb_checkpoints', '.metadata', '.recommenders', 'BlackFriday.csv', 'Untitled.ipynb', 'Untitled1.ipynb']
```

In [2]:
```
df = pd.read_csv('../workspace/BlackFriday.csv')
df.shape
```

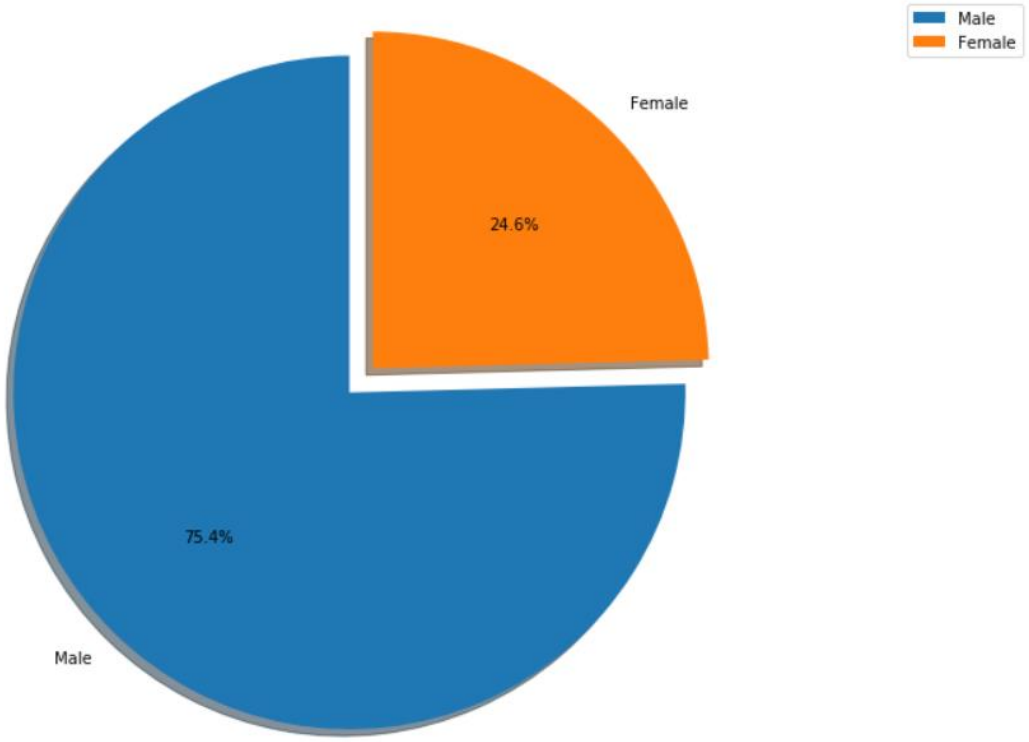
Out[2]: (537577, 12)

1- Gender:

Results categorising purchases based on gender:

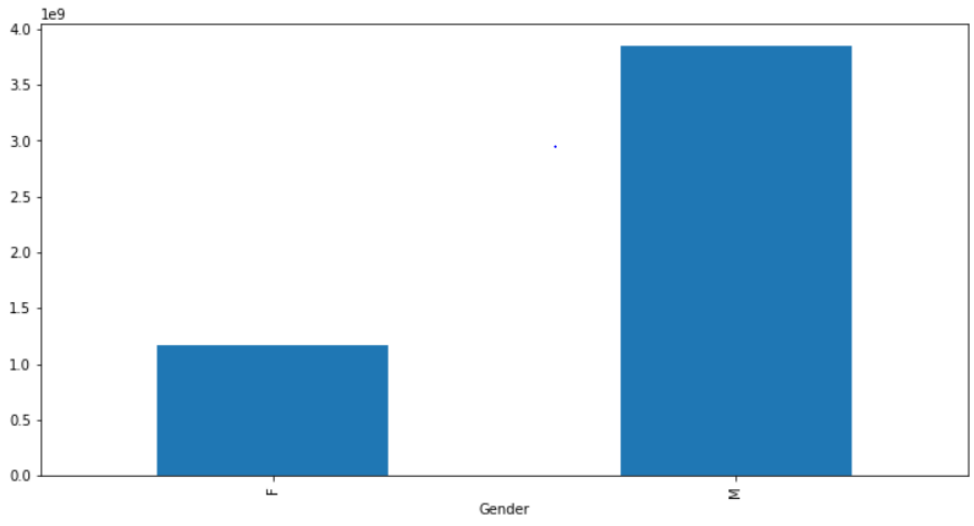
In [5]:
```
# Gender

explode = (0.1,0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df['Gender'].value_counts(), explode=explode,labels=['Male','Female'], autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```



In [7]:
```
# Gender wise purchase
def plot(group,column,plot):
    ax=plt.figure(figsize=(12,6))
    df.groupby(group)[column].sum().sort_values().plot(kind=plot)

plot('Gender','Purchase','bar')
```



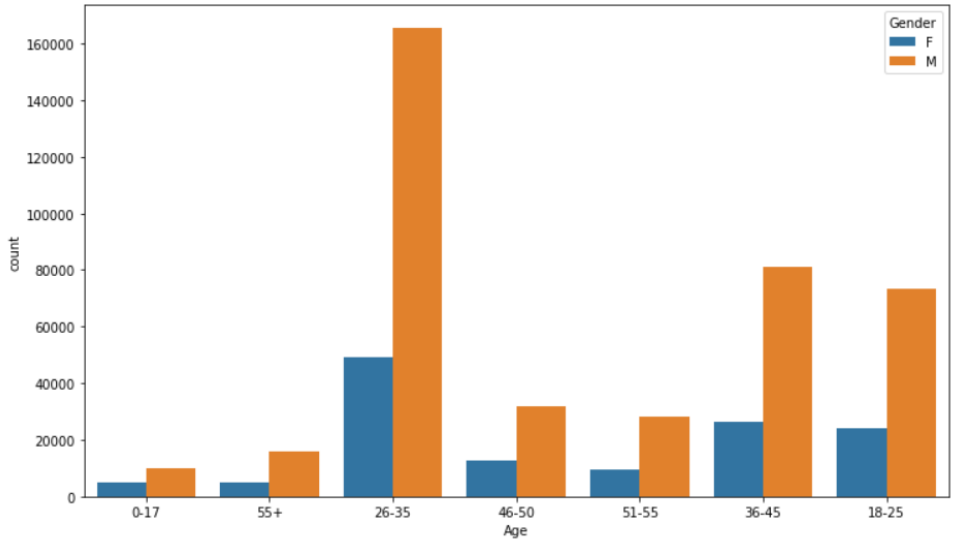
Analysis: Men's purchasing power is greater than women's purchasing power, even in normal circumstances. This is likely to affect the owner of the money, but there has been a high turnout of men in the store. About 75% of the customers have made sales of men of all ages.

2-Age:

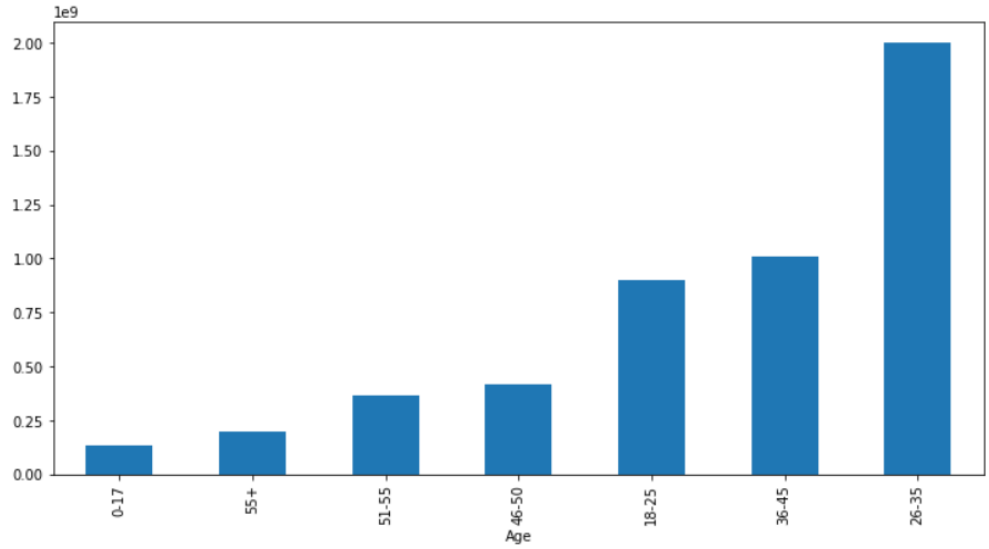
Results categorising different age groups and their purchasing power:

```
In [8]: # Age wise data
        fig1, ax1 = plt.subplots(figsize=(12,7))
        sns.countplot(df['Age'],hue=df['Gender'])
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x2164e2a0cc8>
```



In [9]:
```
# Age wise purchase
plot('Age','Purchase','bar')
```

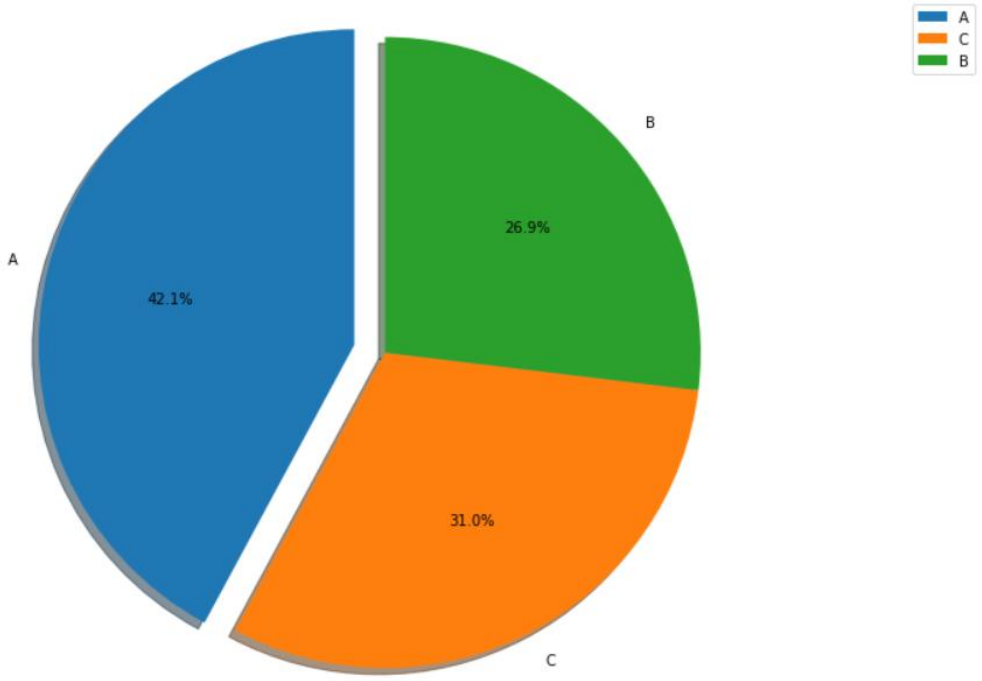


<u>Analysis:</u> We can consider that the target age group of our stores is the age group of 26-35 years, we have achieved sales of more than 3 billion in the age group of 26-45 years.

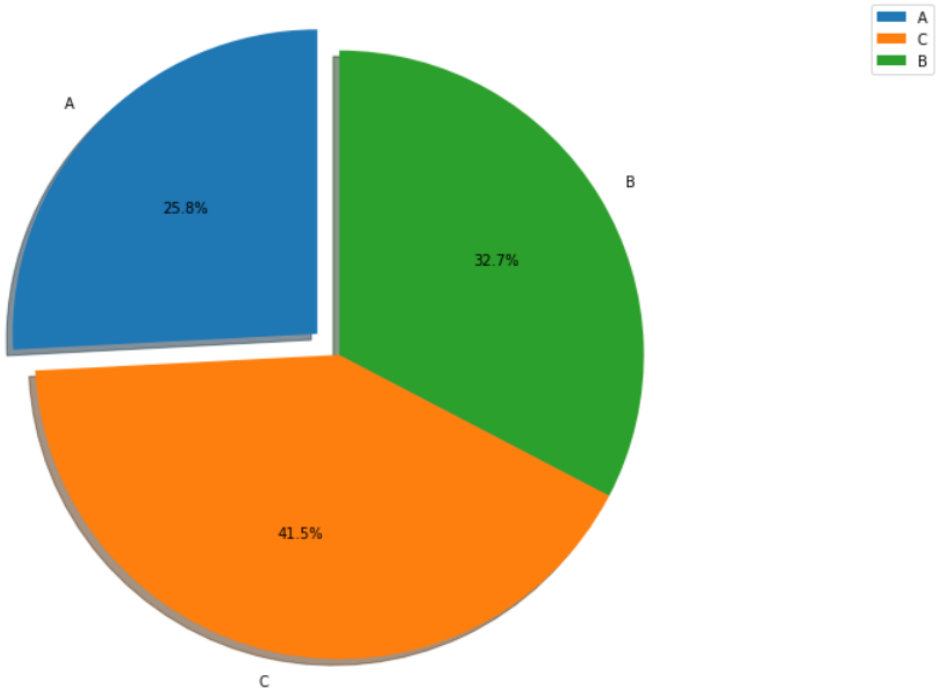
3-City:

Analysing data in terms of City in which the customers reside:

In [10]:
```
# City
explode = (0.1, 0, 0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df['City_Category'].value_counts(),explode=explode, labels=df['City_Category'].unique(), autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```

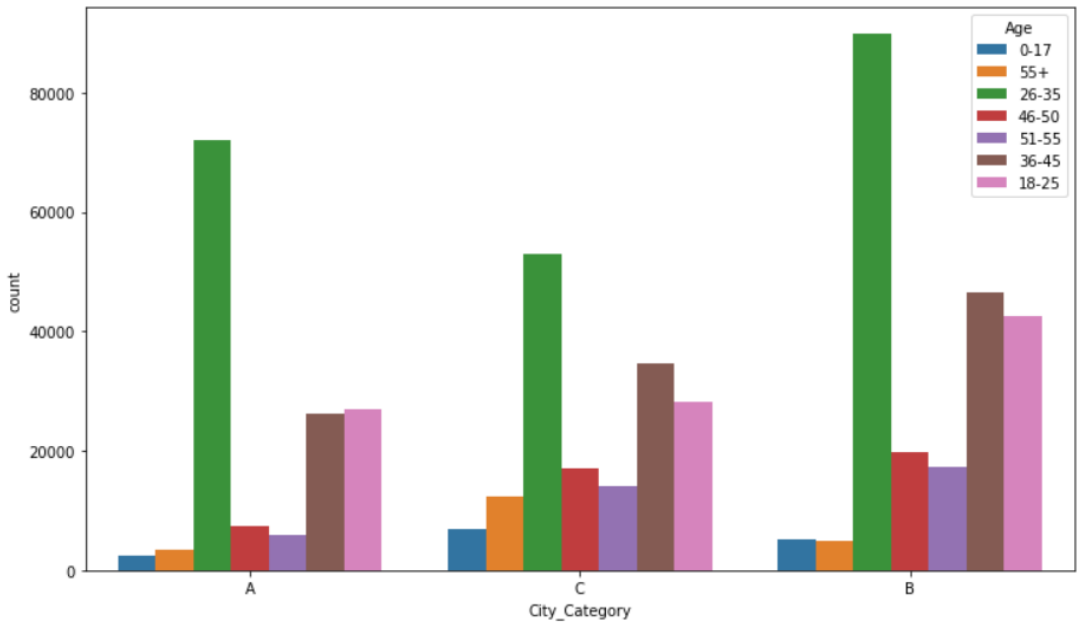


In [11]:
```
# City
explode = (0.1, 0, 0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df.groupby('City_Category')['Purchase'].sum(),explode=explode, labels=df['City_Category'].unique(), autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```

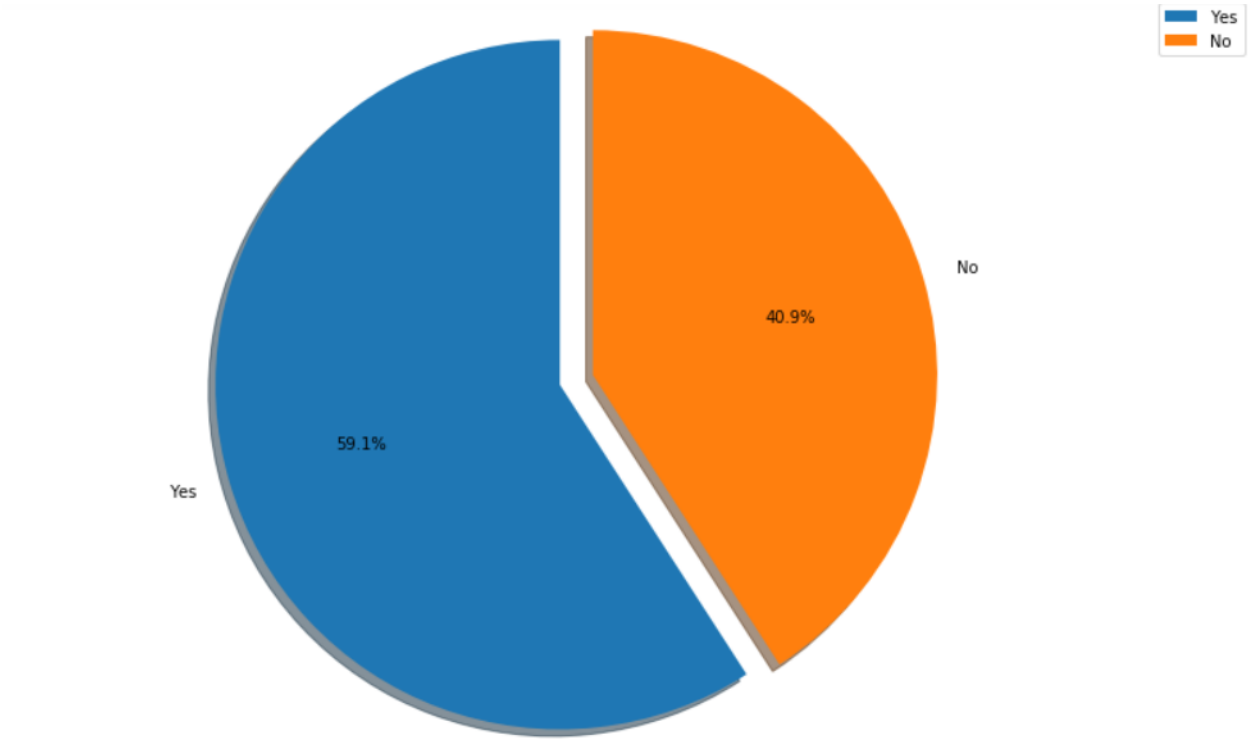


In [12]:
```
fig1, ax1 = plt.subplots(figsize=(12,7))
sns.countplot(df['City_Category'],hue=df['Age'])
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x2164fc19548>



In [13]:
```
#label=['Underage 0-17','Retired +55','Middleage 26-35','46-50 y/o','Oldman 51-55','Middleage+ 36-45','Youth']
explode = (0.1, 0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df['Marital_Status'].value_counts(),explode=explode, labels=['Yes','No'], autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```

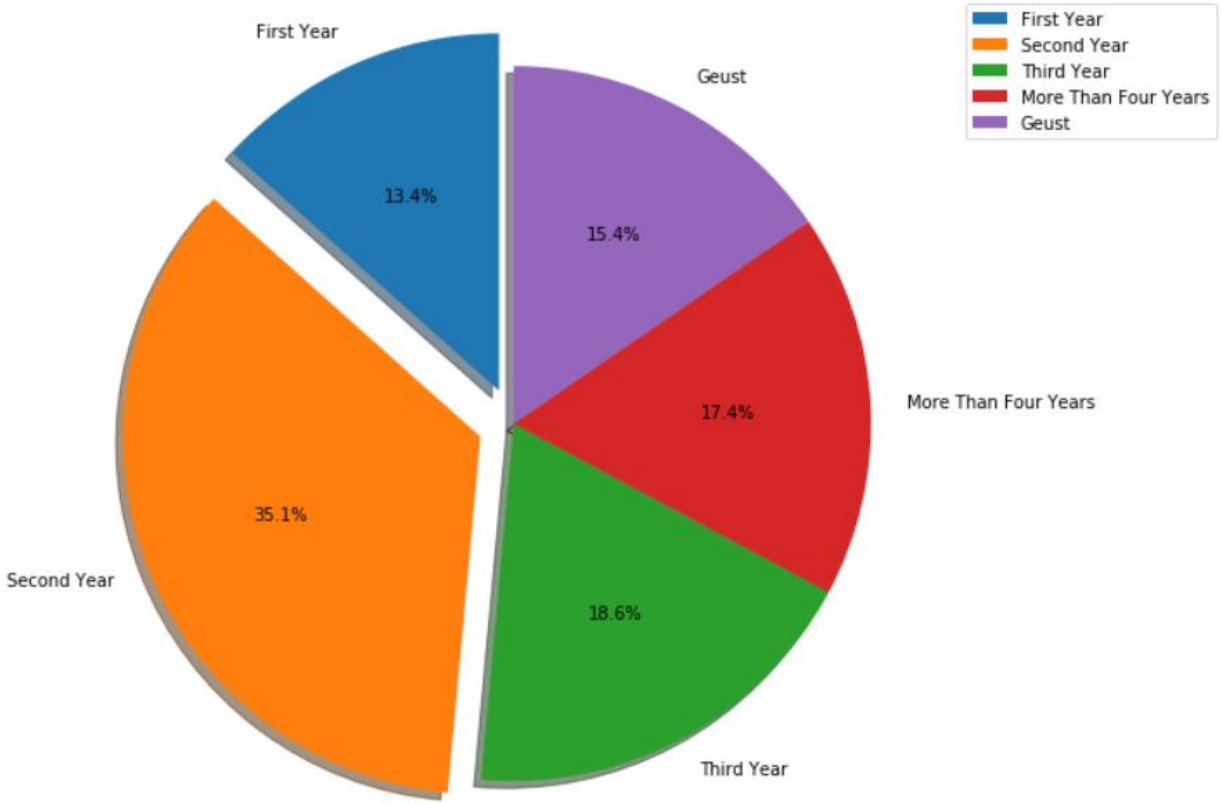


Analysis: City A is the most cities followed by B and then C, the distribution of ages on the procurement map is very close, we have to focus on the category of work averages of 36-45. Most of our customers are more than 60% married, I see that the strategy of targeting families to ensure more clients succeed.

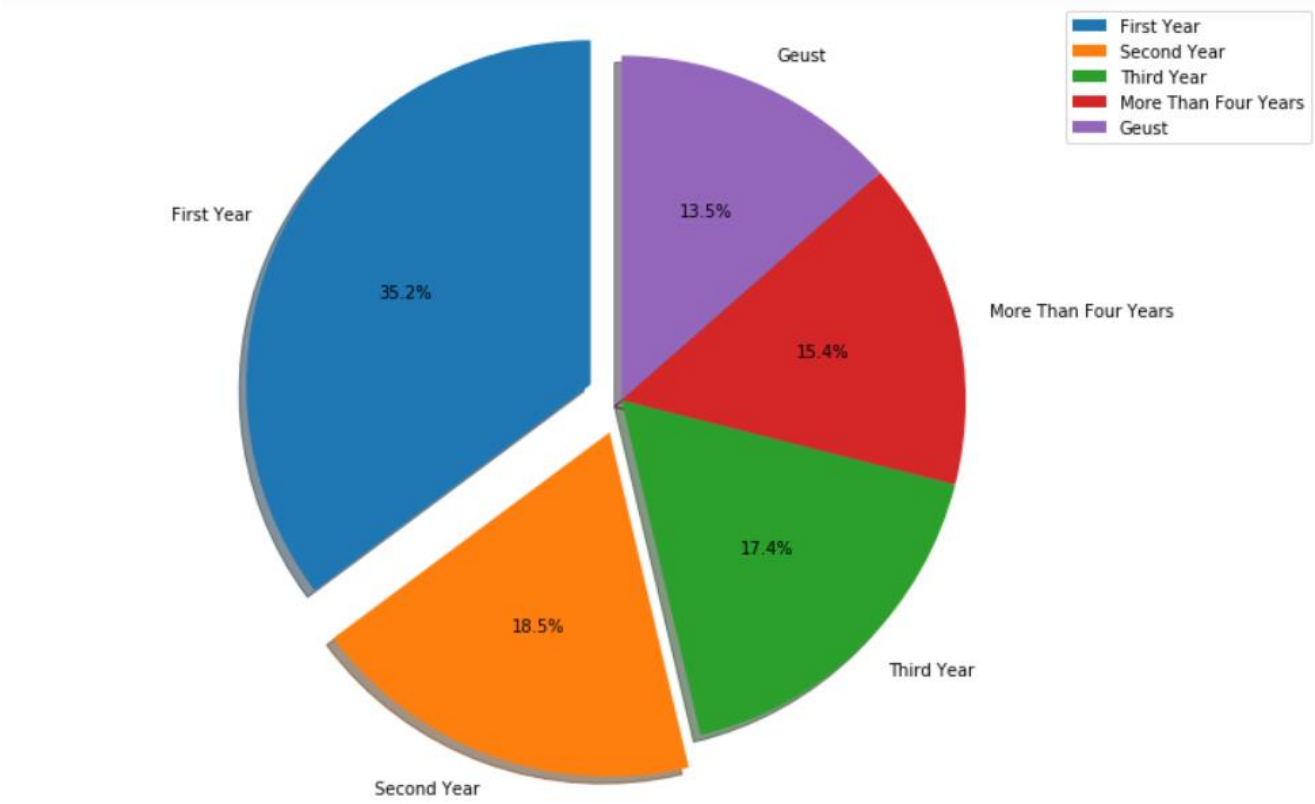
4-Stability:

Analysis on the stability of frequenting customers:

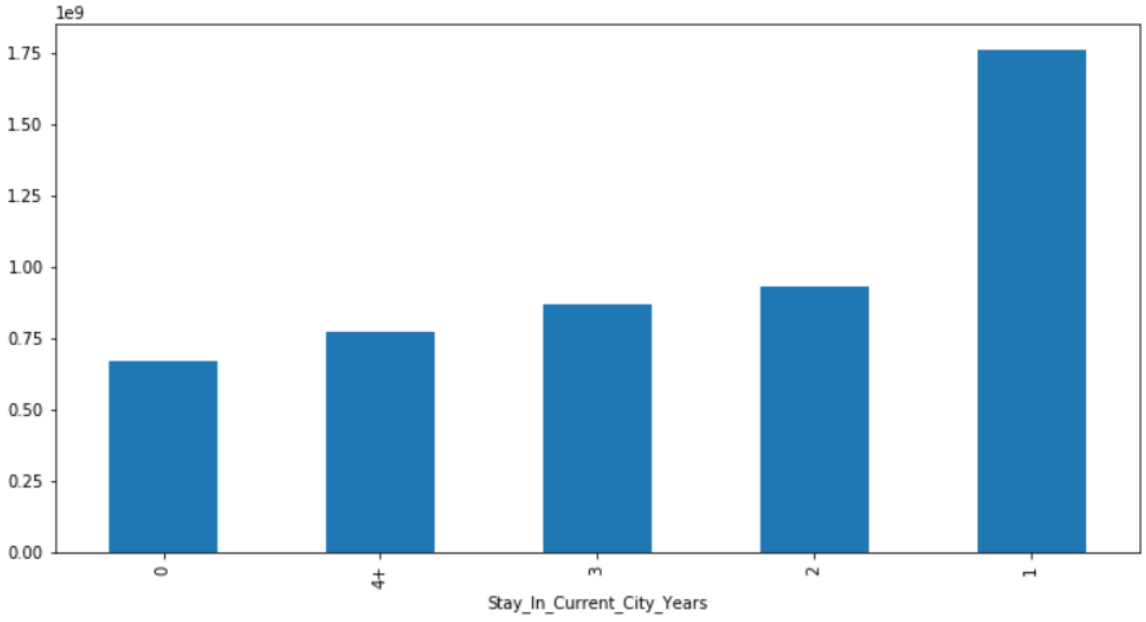
In [14]:
```
# Stability
labels=['First Year','Second Year','Third Year','More Than Four Years','Geust']
explode = (0.1, 0.1,0,0,0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df.groupby('Stay_In_Current_City_Years')['Purchase'].sum(),explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```



In [15]:
```
# Stability
labels=['First Year','Second Year','Third Year','More Than Four Years','Geust']
#label=['Underage 0-17','Retired +55','Middleage 26-35','46-50 y/o','Oldman 51-55','Middleage+ 36-45','Youth']
explode = (0.1, 0.1,0,0,0)
fig1, ax1 = plt.subplots(figsize=(12,7))
ax1.pie(df['Stay_In_Current_City_Years'].value_counts(),explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.legend()
plt.show()
```



In [16]:
```
plot('Stay_In_Current_City_Years','Purchase','bar')
```



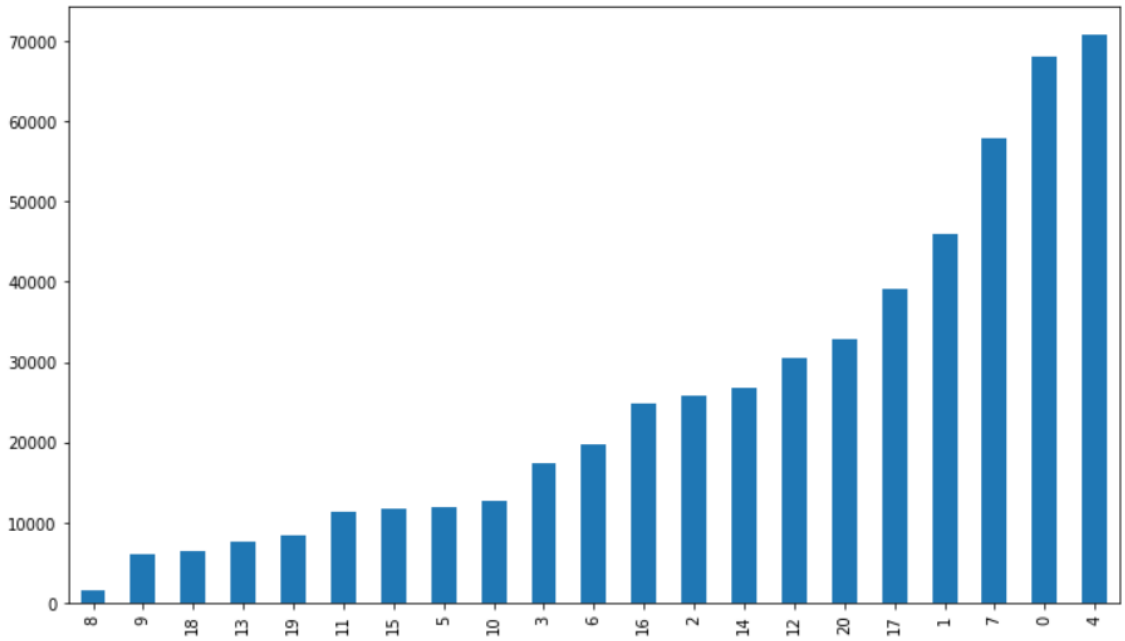
<u>Analysis:</u> We have worked hard in the past two years and have achieved a large percentage of sales from the new population of cities, but these figures indicate that the older city dwellers have less passion for our products.

5-Occupation:

Analysis on the occupation of the customers:

```
In [18]: # Occupation
         fig1, ax1 = plt.subplots(figsize=(12,7))
         df['Occupation'].value_counts().sort_values().plot(kind='bar')
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x2164d9d6788>
```

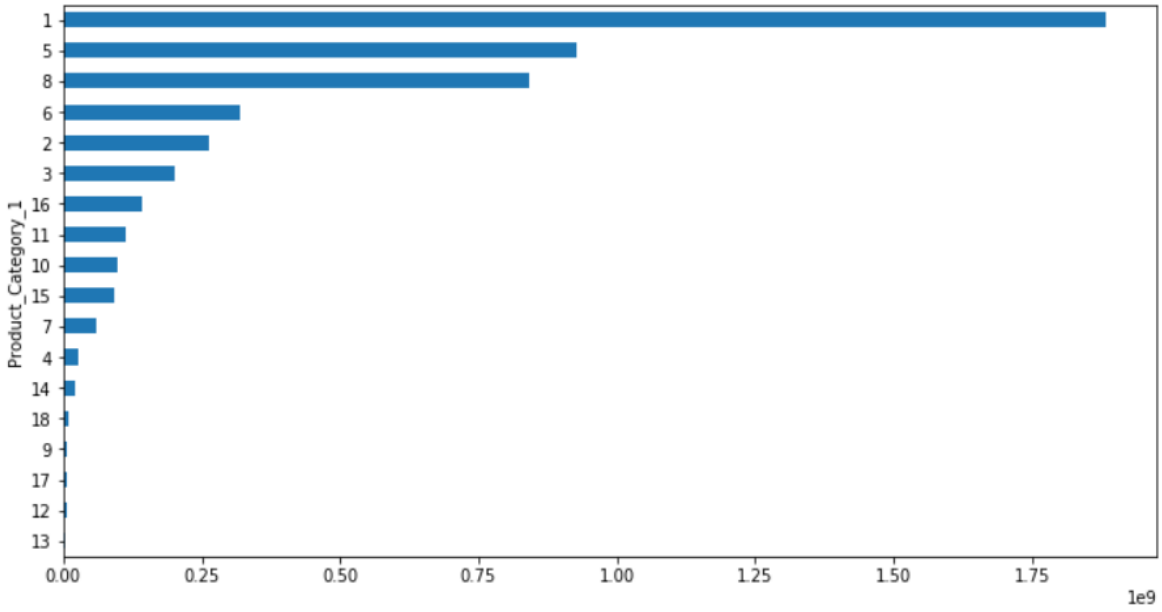


Analysis: **We also note here that purchasing power is closely related to the Occupation in some cases as seen in the table but there are some differences we will notice when checking the number of purchases and the value of those purchases.**

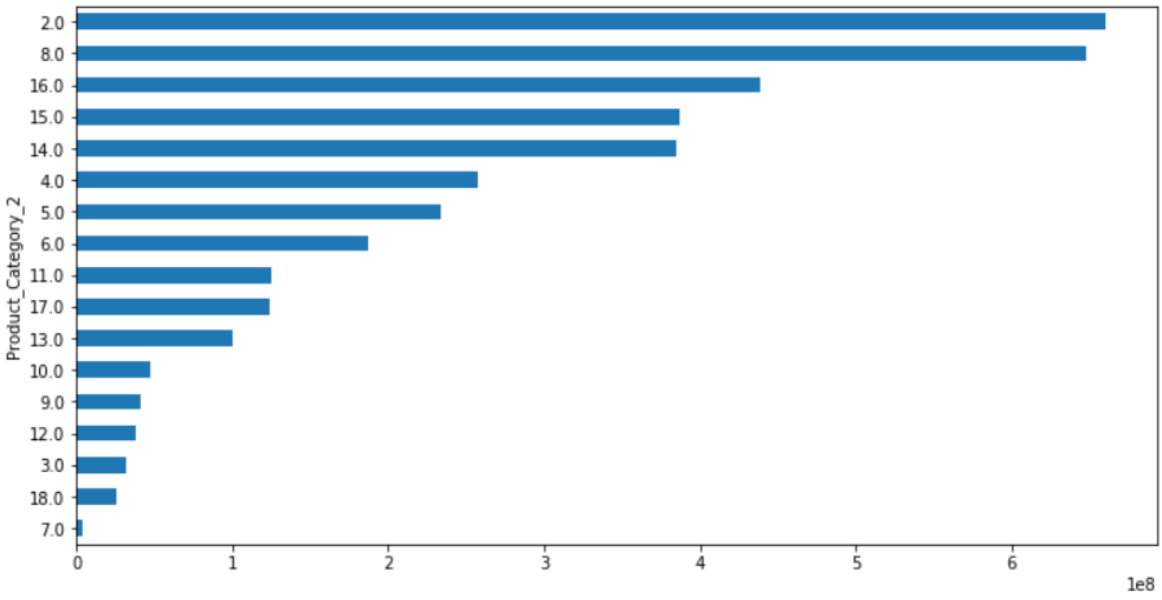
6-Products and Categories:

Data representing different types of products:

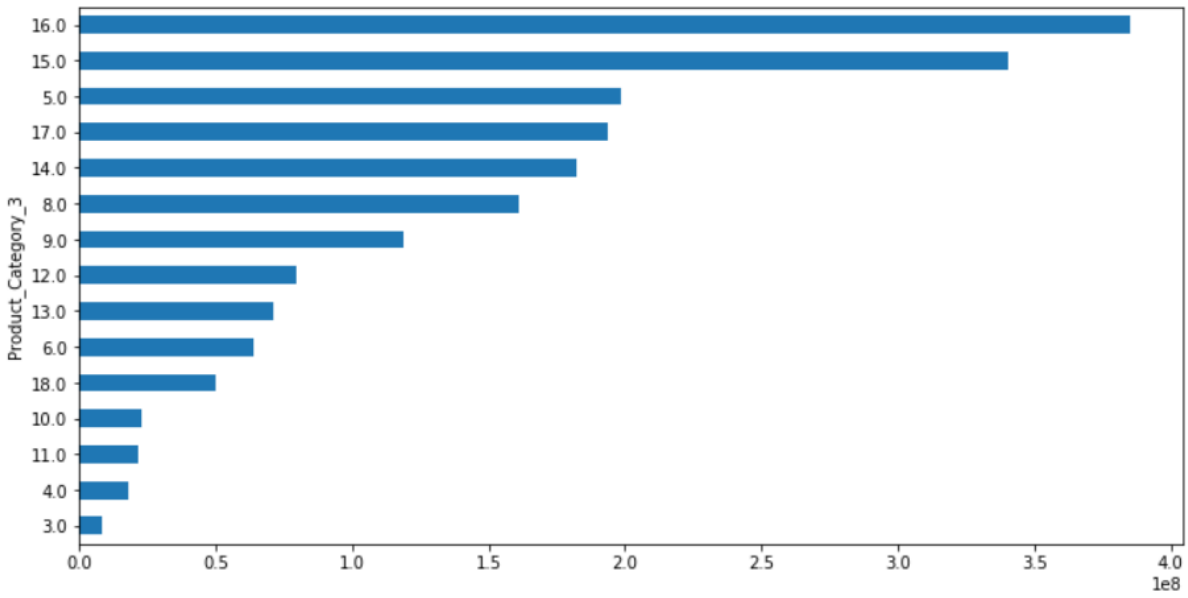
In [19]:
```
# Products and Categories
plot('Product_Category_1','Purchase','barh')
```



In [20]:
```
plot('Product_Category_2','Purchase','barh')
```

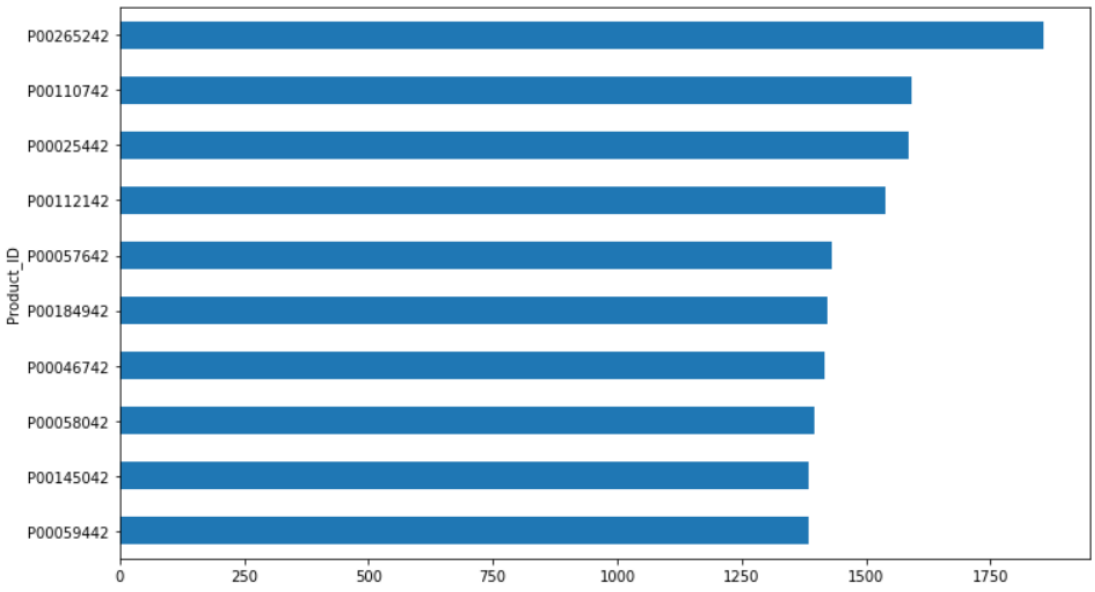


In [21]:
```
plot('Product_Category_3','Purchase','barh')
```



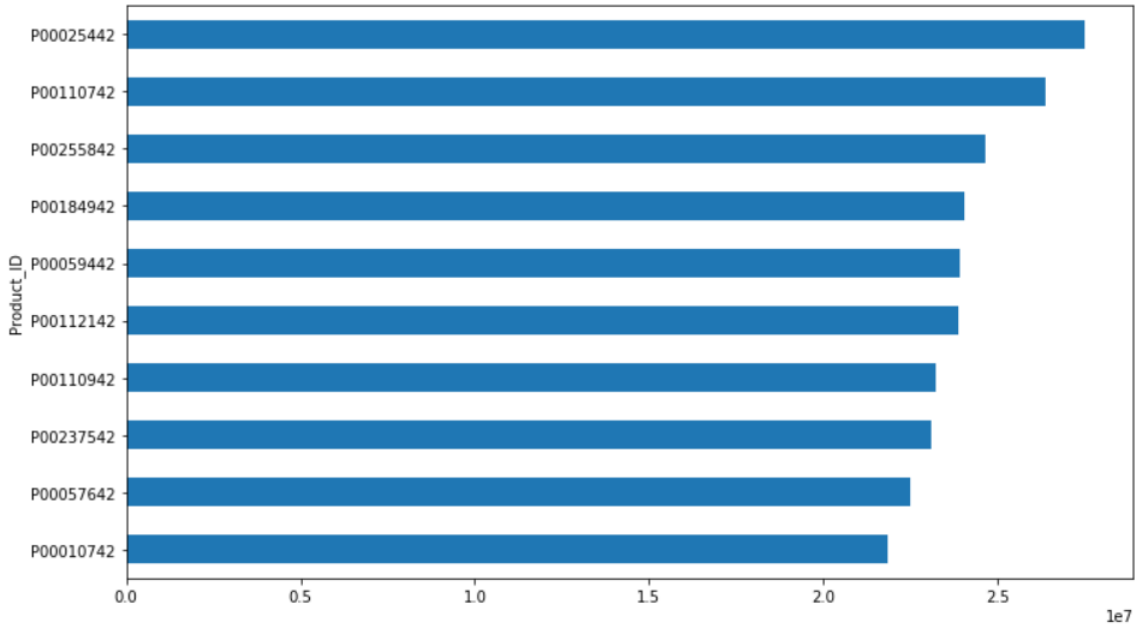
In [23]:
```
fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Purchase'].count().nlargest(10).sort_values().plot(kind='barh')
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21650732348>
```



In [24]:
```
fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Purchase'].sum().nlargest(10).sort_values().plot(kind='barh')
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x21650a9acc8>
```



Analysis: we have the top 10 products for the top 10 profits, and first 10 category for each product.

Conclusion:

(1) The overall conclusion is that the most frequent buyers as well as the buyers with most purchasing power are married. So, the marketing team should target clients with families for increasing the sales.

(2) The target age group should be 26-45 years, as this age group buys the most products.

(3) We need to increase our sales in city C, as this city has least buyers.

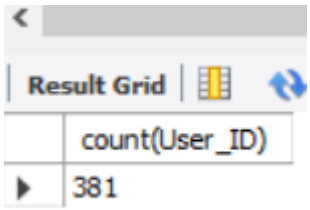
(4) Older population of cities are not that interested in our products. We need to increase our sales in this section of the population by bringing in the products they like.

SQL QUERIES:

1. **To find total number of residents residing in a city:**

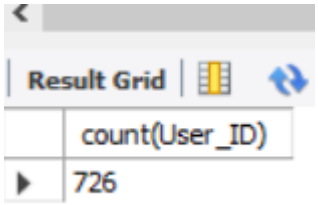
   City A:

```
select distinct count(User_ID) from sales_analysis
where City_Category='A';
```



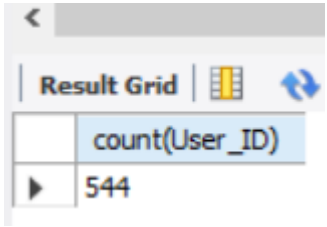
   City B:

```
select distinct count(User_ID) from sales_analysis
where City_Category='B';
```



   City C:

```
select distinct count(User_ID) from sales_analysis
where City_Category='C';
```



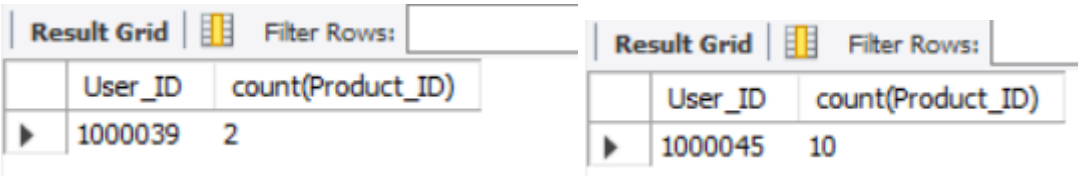
2. **To find total number of orders placed by each buyer:**

   Here we used stored procedure:

```
delimiter $$
create procedure total_orders_placed()
begin
        declare x int(7);
   set x=1000001;
   loop_label: LOOP
                if x>1000050 then
                        leave loop_label;
                end if;
      select User_ID, count(Product_ID) from sales_analysis
      where User_ID=x;
      set x=x+1;
        end loop;
end $$


call total_orders_placed();
```

Sample output:



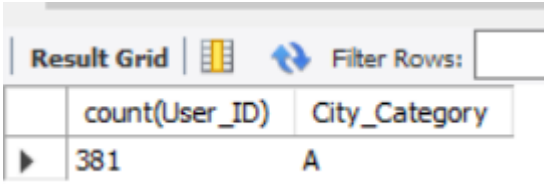
User_ID	count(Product_ID)
1000039	2

User_ID	count(Product_ID)
1000045	10

3. **To find the number of residents in each city and their duration of stay:**

City A:

```
select distinct count(User_ID), City_Category from sales_analysis
where City_Category='A';
```

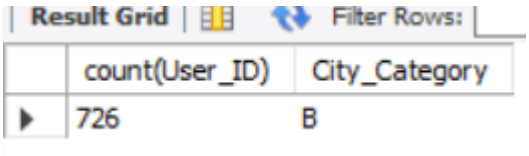


Result Grid | Filter Rows:

count(User_ID)	City_Category
381	A

City B:

```
select distinct count(User_ID), City_Category from sales_analysis
where City_Category='B';
```

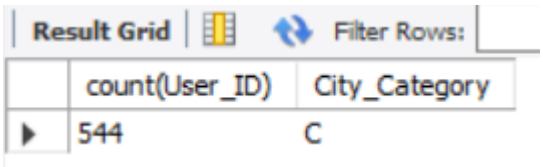


Result Grid | Filter Rows:

count(User_ID)	City_Category
726	B

City C:

```
select distinct count(User_ID), City_Category from sales_analysis
where City_Category='C';
```

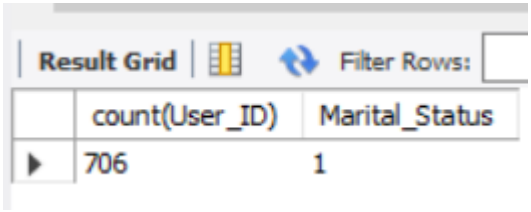


Result Grid | Filter Rows:

count(User_ID)	City_Category
544	C

4. **To find the total number of married buyers:**

```
select distinct count(User_ID), Marital_Status from sales_analysis
where Marital_Status=1;
```



Result Grid | Filter Rows:

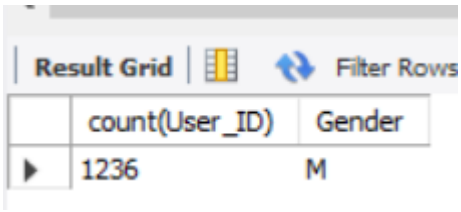
count(User_ID)	Marital_Status
706	1

5. **To find total number of male and female buyers:**

Male:

```
select distinct count(User_ID), Gender from sales_analysis

where Gender='M';
```



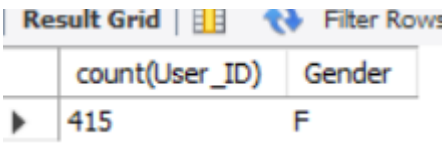
Result Grid | Filter Rows

	count(User_ID)	Gender
▶	1236	M

Female:

```
select distinct count(User_ID), Gender from sales_analysis

where Gender='F';
```



Result Grid | Filter Rows

	count(User_ID)	Gender
▶	415	F