Problem Statement:

The transactions made by a UK-based, registered, non-store online retailer between December 1, 2010, and December 9,2011, are all included in the transnat ional data set known as online retail. The company primarily offer one-of-a-kin d gifts for every occasion. The company has a large number of wholesalers as cl ients.Company Objective Using the global online retail dataset, we will design a clustering model and select the ideal group of clients for the business to ta rget.

In [1]:

```
#import libraries
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\OnlineRetailData (1).csv")
df
```

Out[2]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	I
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:1	2.55	17850.0	I
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:1	3.39	17850.0	I
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:1	2.75	17850.0	I
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:1	3.39	17850.0	I
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:1	3.39	17850.0	I
...	...	...	...	...	...	...	...	
65530	541696	21205	MULTICOLOUR 3D BALLS GARLAND	1	20-01-2011 18:08	2.46	NaN	I
65531	541696	21208	PASTEL COLOUR HONEYCOMB FAN	2	20-01-2011 18:08	1.63	NaN	I
65532	541696	21209	MULTICOLOUR HONEYCOMB FAN	1	20-01-2011 18:08	1.63	NaN	I
65533	541696	21212	PACK OF 72 RETROSPOT CAKE CASES	1	20-01-2011 18:08	1.25	NaN	I
65534	541696	21217	RED RETROSPOT ROUND CAKE TINS	1	20-01-2011 18:08	20.79	NaN	I

65535 rows × 8 columns

In [3]:

```
df.head()
```

Out[3]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:1	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:1	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:1	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:1	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:1	3.39	17850.0	United Kingdom

In [4]:

```
df.tail()
```

Out[4]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	
65530	541696	21205	MULTICOLOUR 3D BALLS GARLAND	1	20-01-2011 18:08	2.46	NaN	I
65531	541696	21208	PASTEL COLOUR HONEYCOMB FAN	2	20-01-2011 18:08	1.63	NaN	I
65532	541696	21209	MULTICOLOUR HONEYCOMB FAN	1	20-01-2011 18:08	1.63	NaN	I
65533	541696	21212	PACK OF 72 RETROSPOT CAKE CASES	1	20-01-2011 18:08	1.25	NaN	I
65534	541696	21217	RED RETROSPOT ROUND CAKE TINS	1	20-01-2011 18:08	20.79	NaN	I

In [5]:

```
df.describe()
```

Out[5]:

	Quantity	UnitPrice	CustomerID
count	65535.000000	65535.000000	40218.000000
mean	8.363119	5.856143	15384.033517
std	413.694482	145.755953	1766.863499
min	-74215.000000	0.000000	12346.000000
25%	1.000000	1.250000	14001.000000
50%	2.000000	2.510000	15358.000000
75%	8.000000	4.240000	17019.000000
max	74215.000000	16888.020000	18283.000000

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65535 entries, 0 to 65534
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   InvoiceNo    65535 non-null  object 
 1   StockCode    65535 non-null  object 
 2   Description  65369 non-null  object 
 3   Quantity     65535 non-null  int64  
 4   InvoiceDate  65535 non-null  object 
 5   UnitPrice    65535 non-null  float64
 6   CustomerID   40218 non-null  float64
 7   Country      65535 non-null  object 
dtypes: float64(2), int64(1), object(5)
memory usage: 4.0+ MB
```

In [7]:

```
df['CustomerID'].value_counts()
```

Out[7]:

```
CustomerID
12748.0    695
17841.0    481
14606.0    421
15311.0    418
14911.0    377
          ...
13883.0      1
18233.0      1
13829.0      1
17616.0      1
12875.0      1
Name: count, Length: 1204, dtype: int64
```

In [8]:

```
df['Quantity'].value_counts()
```

Out[8]:

```
Quantity
 1       21712
 2       10237
 12       5620
 3        4870
 6        4572
         ...
-177         1
-723         1
 320         1
-223         1
-1400        1
Name: count, Length: 247, dtype: int64
```

In [9]:

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

Out[9]:

```
InvoiceNo          0
StockCode          0
Description      166
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID     25317
Country            0
dtype: int64
```

In [10]:

```
df.fillna(method='ffill',inplace=True)
```

In [11]:

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

Out[11]:

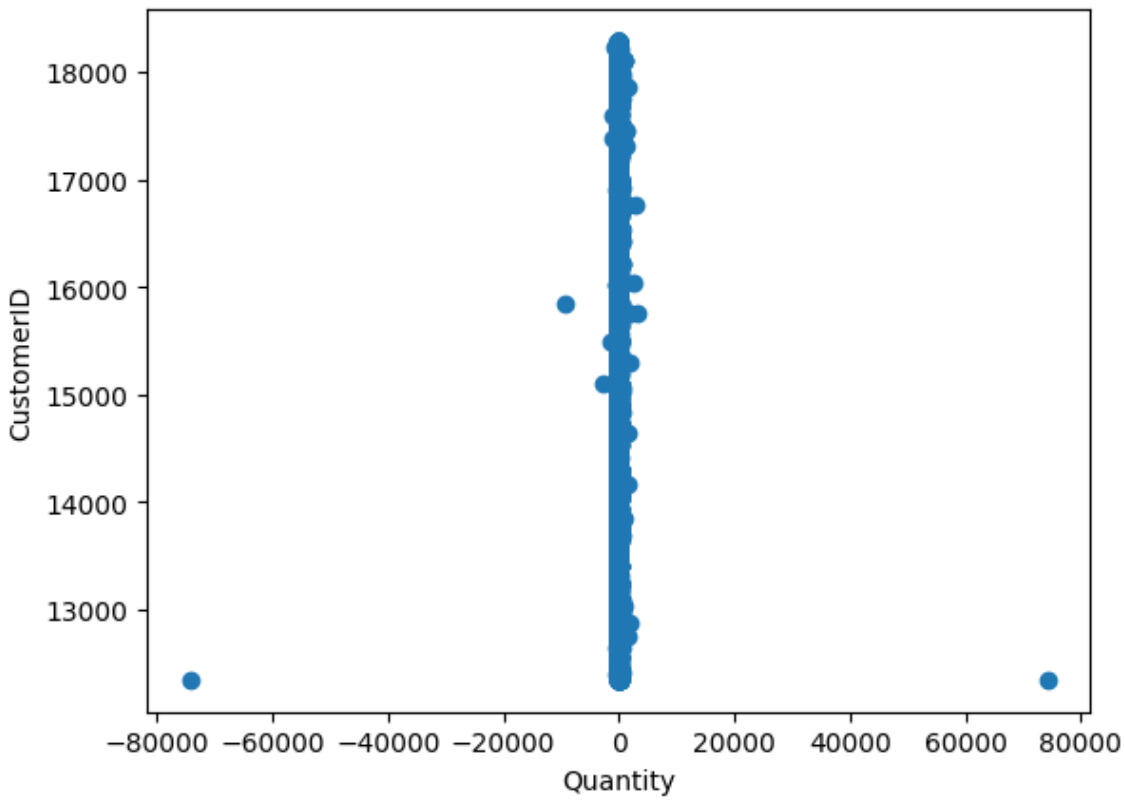
```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

In [12]:

```
plt.scatter(df["Quantity"],df["CustomerID"])
plt.xlabel("Quantity")
plt.ylabel("CustomerID")
```

Out[12]:

```
Text(0, 0.5, 'CustomerID')
```



K-Means Clustering:

In [13]:

```
from sklearn.cluster import KMeans
```

In [14]:

```
km=KMeans()
km
```

Out[14]:

```
KMeans()
```

In [15]:

```
y_predicted=km.fit_predict(df[["Quantity","CustomerID"]])
y_predicted
```

```
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
```

Out[15]:

```
array([5, 5, 5, ..., 5, 5, 5])
```

In [16]:

```
df["Cluster"]=y_predicted
df.head()
```

Out[16]:

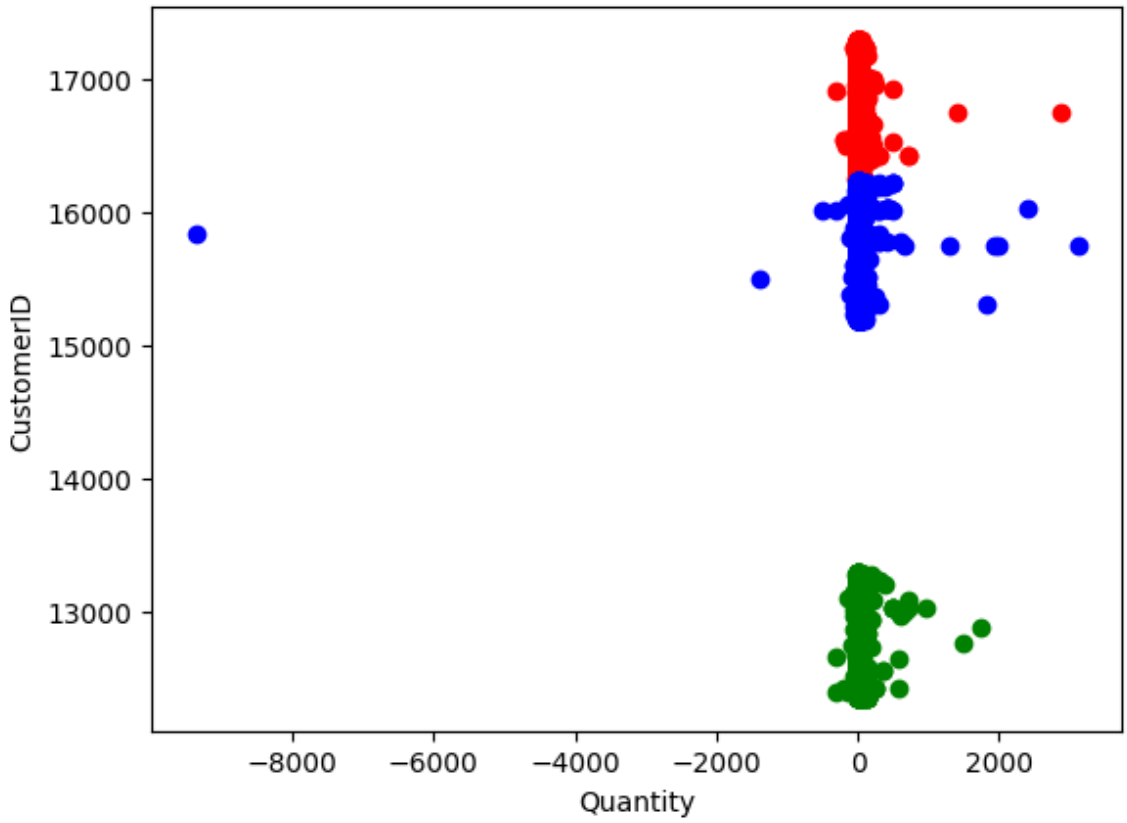
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:1	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:1	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:1	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:1	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:1	3.39	17850.0	United Kingdom

In [18]:

```
df1=df[df.Cluster==0]
df2=df[df.Cluster==2]
df3=df[df.Cluster==3]
plt.scatter(df1["Quantity"],df1["CustomerID"],color="red")
plt.scatter(df2["Quantity"],df2["CustomerID"],color="blue")
plt.scatter(df3["Quantity"],df3["CustomerID"],color="green")
plt.xlabel("Quantity")
plt.ylabel("CustomerID")
```

Out[18]:

```
Text(0, 0.5, 'CustomerID')
```



In [19]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [20]:

```
scaler=MinMaxScaler()
```

In [21]:

```
scaler.fit(df[["CustomerID"]])
df["CustomerID"]=scaler.transform(df[["CustomerID"]])
df.head()
```

Out[21]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01-12-2010 08:1	2.55	0.927068	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01-12-2010 08:1	3.39	0.927068	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01-12-2010 08:1	2.75	0.927068	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01-12-2010 08:1	3.39	0.927068	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01-12-2010 08:1	3.39	0.927068	United Kingdom

In [22]:

```
scaler.fit(df[["Quantity"]])
df["Quantity"]=scaler.transform(df[["Quantity"]])
df.head()
```

Out[22]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.500040	01-12-2010 08:1	2.55	0.927068	United Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500054	01-12-2010 08:1	2.75	0.927068	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom

In [23]:

```
km=KMeans()
```

In [24]:

```
y_predicted=km.fit_predict(df[["Quantity","CustomerID"]])
y_predicted
```

```
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
```

Out[24]:

```
array([3, 3, 3, ..., 1, 1, 1])
```

In [25]:

```
df["New cluster"]=y_predicted
df.head()
```

Out[25]:

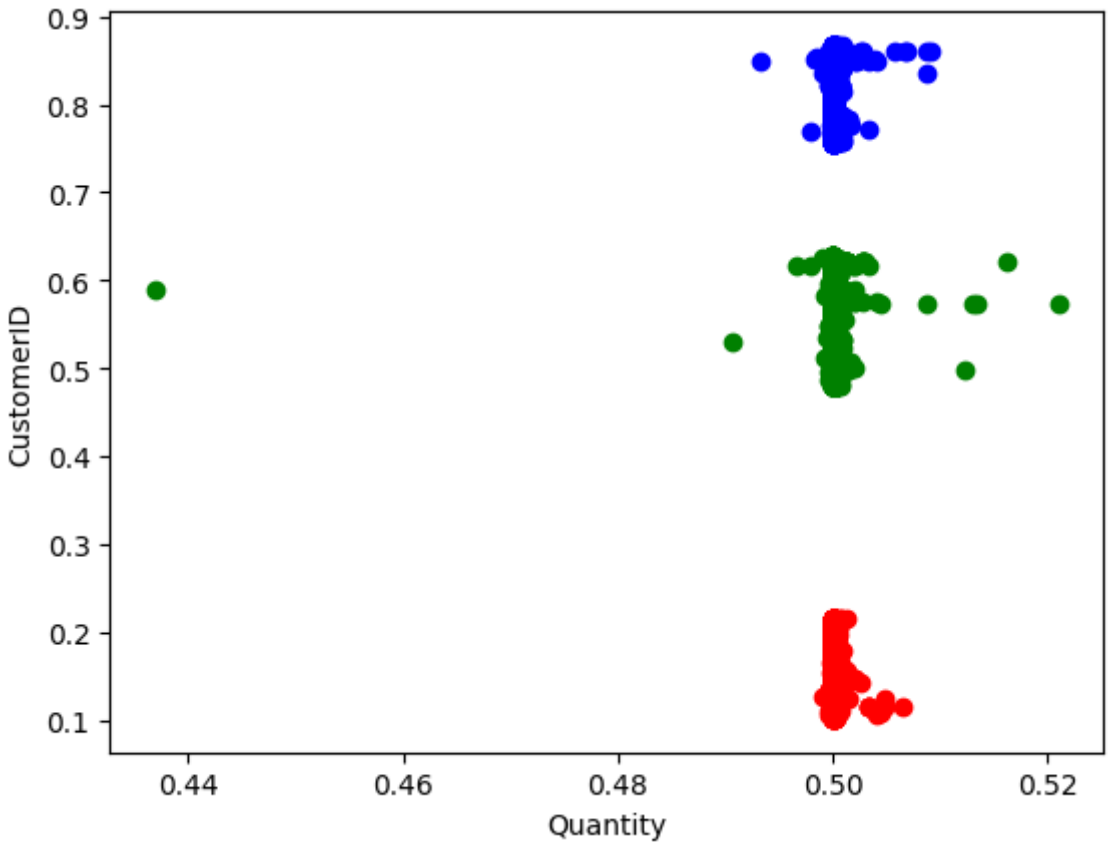
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.500040	01-12-2010 08:1	2.55	0.927068	United Kingdom
1	536365	71053	WHITE METAL LANTERN	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.500054	01-12-2010 08:1	2.75	0.927068	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	0.500040	01-12-2010 08:1	3.39	0.927068	United Kingdom

In [27]:

```
df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["Quantity"],df1["CustomerID"],color="red")
plt.scatter(df2["Quantity"],df2["CustomerID"],color="blue")
plt.scatter(df3["Quantity"],df3["CustomerID"],color="green")
plt.xlabel("Quantity")
plt.ylabel("CustomerID")
```

Out[27]:

```
Text(0, 0.5, 'CustomerID')
```



In [28]:

```
km.cluster_centers_
```

Out[28]:

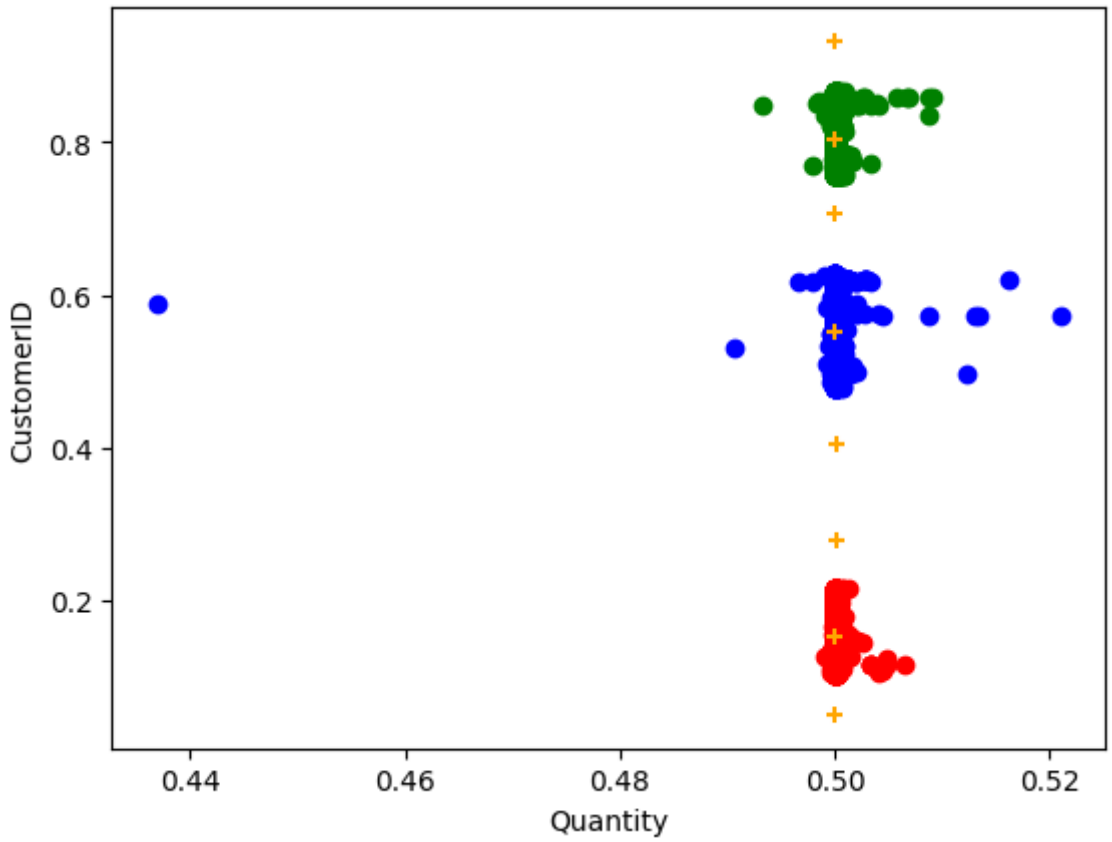
```
array([[0.50006063, 0.15330971],
       [0.50004424, 0.80431768],
       [0.50004769, 0.55293493],
       [0.50004718, 0.93277083],
       [0.50009414, 0.27934016],
       [0.50004826, 0.70557626],
       [0.50007229, 0.40418268],
       [0.50005717, 0.05024609]])
```

In [29]:

```
df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["Quantity"],df1["CustomerID"],color="red")
plt.scatter(df2["Quantity"],df2["CustomerID"],color="green")
plt.scatter(df3["Quantity"],df3["CustomerID"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker="+")
plt.xlabel("Quantity")
plt.ylabel("CustomerID")
```

Out[29]:

```
Text(0, 0.5, 'CustomerID')
```



In [30]:

```
k_rng=range(1,10)
sse=[]
```

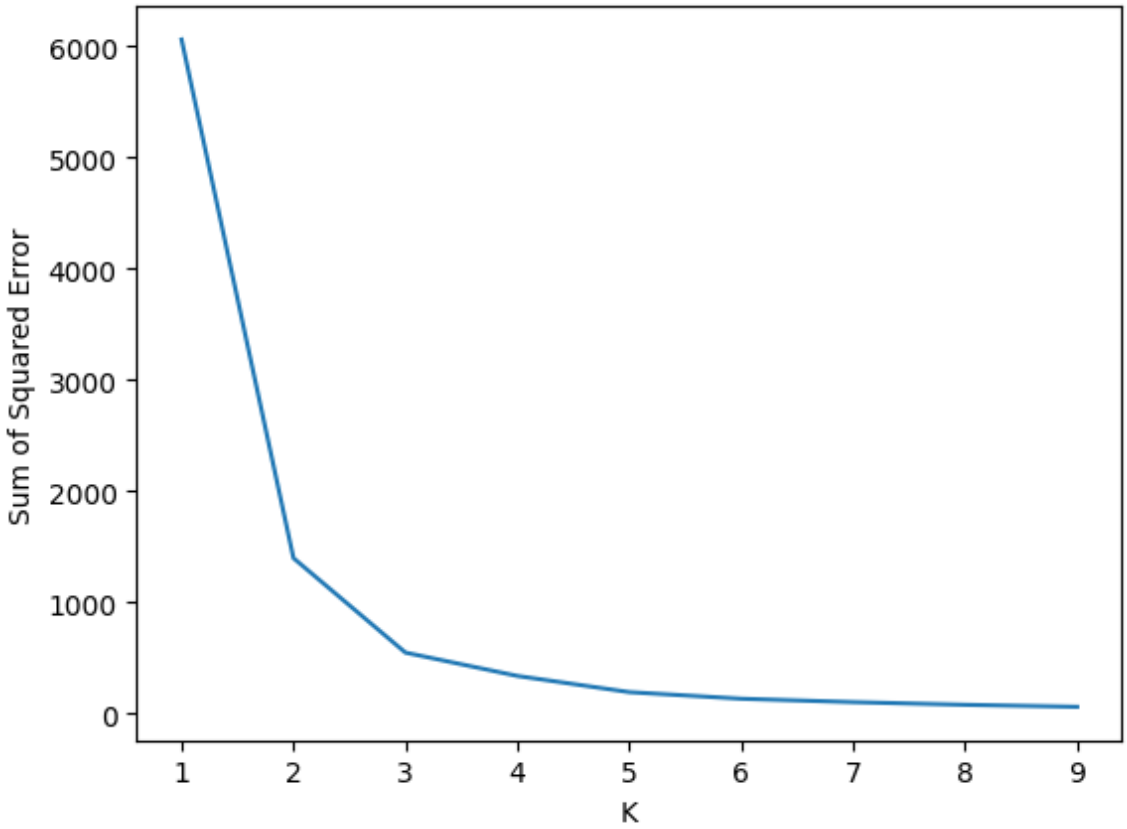
In [31]:

```
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["Quantity","CustomerID"]])
 sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square errorprint(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-packages\sk
learn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  warnings.warn(
```

Out[31]:

```
Text(0, 0.5, 'Sum of Squared Error')
```



Conclusion: ¶

In This dataset we are performing clustering on Quantity and CustomerID.By usi ng kMeans Algorithm.so we conclude that KMeans alogorithm is best fot this Data set.

In [ ]: