In [93]:
```
import pandas as pd
import numpy as np
```

In [94]:
```
df0=pd.read_excel(r'Case Study Dataset.xls')
df1=pd.read_excel(r'Case Study Dataset.xls', sheet_name=1 )
df2=pd.read_excel(r'Case Study Dataset.xls', sheet_name=2 )
df3=pd.read_excel(r'Case Study Dataset.xls', sheet_name=3 )
```

In [95]:
```
print("PRINTING THE DATA \n")
print(df0)
print(df1)
print(df2)
print(df3)
```

```
PRINTING THE DATA

      Transaction ID Customer ID       Customer Name Order Date  \
0             103982    AA-10375        Allen Armold 2021-11-14
1             147039    AA-10375        Allen Armold 2022-09-07
2             131065    AA-10375        Allen Armold 2022-09-07
3             131065    AA-10375        Allen Armold 2022-12-11
4             169488    AA-10480        Andrew Allen 2021-07-17
...              ...         ...                 ...        ...
3427          167682    ZC-21910    Zuschuss Carroll 2021-03-08
3428          147991    ZC-21910    Zuschuss Carroll 2021-04-08
3429          152471    ZC-21910    Zuschuss Carroll 2022-11-06
3430          152471    ZD-21925  Zuschuss Donatelli 2021-04-03
3431          141481    ZD-21925  Zuschuss Donatelli 2021-05-05

      Sales Person Name Product ID Category   Product Name  Dollar Sales  \
0            John Blake     J-2001    Jeans    Black Denim            25
1         Alex Ferguson     J-2003    Jeans         Chinos           200
2         Alex Ferguson     J-2002    Jeans     Blue Denim            30
3         Alex Ferguson     J-2003    Jeans         Chinos           120
4     Samuel Washington     J-2003    Jeans         Chinos           120
...                 ...        ...      ...            ...           ...
3427         John Blake    SO-3003    Shoes   Formal Shoes           270
3428        Jane Austin    SO-3003    Shoes   Formal Shoes           360
3429      Mike Davidson    SO-3001    Shoes       Sneakers            50
3430        Jane Austin    SO-3002    Shoes  Running Shoes           280
3431         John Blake    SO-3001    Shoes       Sneakers           250

      Returns  Quantity
0           0         1
1           0         5
2           0         2
3           0         3
4           0         3
...       ...       ...
3427       12         3
3428       19         4
3429        0         1
3430        2         4
3431       41         5

[3432 rows x 11 columns]
   Sales Person Name  Sales Person ID
0         John Blake             1451
1      Mike Davidson             1706
2  Samuel Washington             1451
3        Jane Austin             2091
4      Alex Ferguson             1391
5       Saul Goodman             3045
        Product Name Category Product ID  Per Unit Price ($)
0           Sneakers   Shoes     SO-3001                  50
1      Running Shoes   Shoes     SO-3002                  70
2       Formal Shoes   Shoes     SO-3003                  90
3        Black Denim    Jeans     J-2001                  25
4         Blue Denim    Jeans     J-2002                  15
5             Chinos    Jeans     J-2003                  40
6  Full Sleeve Shirt   Shirts    SH-1001                  23
7  Half Sleeve Shirt   Shirts    SH-1002                  18
8            T Shirt   Shirts    SH-1003                  15
          Customer Name Customer ID
0          Allen Armold    AA-10375
1          Andrew Allen    AA-10480
2         Anna Andreadi    AA-10645
3         Aaron Bergman    AB-10015
```

```
4       Adam Bellavance    AB-10060
..                  ...         ...
734       Xylona Preis    XP-21865
735     Yoseph Carroll    YC-21895
736      Yana Sorensen    YS-21880
737   Zuschuss Carroll    ZC-21910
738  Zuschuss Donatelli    ZD-21925

[739 rows x 2 columns]
```

In [96]:
```
#Filtering out the required data on the basis of order date

df0["Order Date"] = pd.to_datetime(df0['Order Date'])
newdf0 = (df0['Order Date'] < '2022-01-01')
newdf0 = df0.loc[newdf0]
newdf0
```

Out[96]:

	Transaction ID	Customer ID	Customer Name	Order Date	Sales Person Name	Product ID	Category	Product Name	Dollar Sales
0	103982	AA-10375	Allen Armold	2021-11-14	John Blake	J-2001	Jeans	Black Denim	25
4	169488	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2003	Jeans	Chinos	120
5	169488	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2003	Jeans	Chinos	40
6	100230	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2002	Jeans	Blue Denim	120
7	100230	AA-10480	Andrew Allen	2021-08-26	Samuel Washington	J-2002	Jeans	Blue Denim	30
...	...	...	...	...	...	...	...	...	...
3424	102288	XP-21865	Xylona Preis	2021-08-26	John Blake	SO-3001	Shoes	Sneakers	150
3427	167682	ZC-21910	Zuschuss Carroll	2021-03-08	John Blake	SO-3003	Shoes	Formal Shoes	270
3428	147991	ZC-21910	Zuschuss Carroll	2021-04-08	Jane Austin	SO-3003	Shoes	Formal Shoes	360
3430	152471	ZD-21925	Zuschuss Donatelli	2021-04-03	Jane Austin	SO-3002	Shoes	Running Shoes	280
3431	141481	ZD-21925	Zuschuss Donatelli	2021-05-05	John Blake	SO-3001	Shoes	Sneakers	250

1499 rows × 11 columns

In [97]:
```
newdf0['freq'] = newdf0.groupby('Transaction ID')['Transaction ID'].transform('cou
newdf0
```

```
C:\Users\dheer\AppData\Local\Temp\ipykernel_12332\1460291830.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  newdf0['freq'] = newdf0.groupby('Transaction ID')['Transaction ID'].transform('c
ount')
```

Out[97]:

	Transaction ID	Customer ID	Customer Name	Order Date	Sales Person Name	Product ID	Category	Product Name	Dollar Sales
0	103982	AA-10375	Allen Armold	2021-11-14	John Blake	J-2001	Jeans	Black Denim	25
4	169488	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2003	Jeans	Chinos	120
5	169488	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2003	Jeans	Chinos	40
6	100230	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2002	Jeans	Blue Denim	120
7	100230	AA-10480	Andrew Allen	2021-08-26	Samuel Washington	J-2002	Jeans	Blue Denim	30
...	...	...	...	...	...	...	...	...	...
3424	102288	XP-21865	Xylona Preis	2021-08-26	John Blake	SO-3001	Shoes	Sneakers	150
3427	167682	ZC-21910	Zuschuss Carroll	2021-03-08	John Blake	SO-3003	Shoes	Formal Shoes	270
3428	147991	ZC-21910	Zuschuss Carroll	2021-04-08	Jane Austin	SO-3003	Shoes	Formal Shoes	360
3430	152471	ZD-21925	Zuschuss Donatelli	2021-04-03	Jane Austin	SO-3002	Shoes	Running Shoes	280
3431	141481	ZD-21925	Zuschuss Donatelli	2021-05-05	John Blake	SO-3001	Shoes	Sneakers	250

1499 rows × 12 columns

In [98]:

```
#creating a new df for frequency of transactions
freq = pd.DataFrame([newdf0['Customer ID'],newdf0['freq']])
freq = freq.transpose()

freq
```

Out[98]:

	Customer ID	freq
0	AA-10375	1
4	AA-10480	2
5	AA-10480	2
6	AA-10480	2
7	AA-10480	2
...	...	...
3424	XP-21865	2
3427	ZC-21910	1
3428	ZC-21910	1
3430	ZD-21925	1
3431	ZD-21925	1

1499 rows × 2 columns

In [99]:
```
#REMOVING THE DUPLICATE DATA FROM DATAFRAME
freq2 = freq.drop_duplicates(subset=['Customer ID'], keep='first')
freq2
```

Out[99]:

	Customer ID	freq
0	AA-10375	1
4	AA-10480	2
10	AB-10060	1
17	AB-10150	1
25	AC-10660	1
...	...	...
3351	TB-21520	3
3363	TD-20995	1
3380	TP-21415	1
3387	TS-21160	2
3396	TS-21430	1

550 rows × 2 columns

In [100…
```
conditions = [(freq2['freq']< 5),(freq2['freq']>=5)&(freq2['freq']<=8),(freq2['freq
values =['10% disc','20% disc','30%disc']

freq2['Disc Category'] = np.select(conditions, values)

freq2
```

```
C:\Users\dheer\AppData\Local\Temp\ipykernel_12332\1709665276.py:4: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  freq2['Disc Category'] = np.select(conditions, values)
```

Out[100]:

	Customer ID	freq	Disc Category
0	AA-10375	1	10% disc
4	AA-10480	2	10% disc
10	AB-10060	1	10% disc
17	AB-10150	1	10% disc
25	AC-10660	1	10% disc
...	...	...	...
3351	TB-21520	3	10% disc
3363	TD-20995	1	10% disc
3380	TP-21415	1	10% disc
3387	TS-21160	2	10% disc
3396	TS-21430	1	10% disc

550 rows × 3 columns

In [101…

```
df0
```

Out[101]:

	Transaction ID	Customer ID	Customer Name	Order Date	Sales Person Name	Product ID	Category	Product Name	Dollar Sales
0	103982	AA-10375	Allen Armold	2021-11-14	John Blake	J-2001	Jeans	Black Denim	25
1	147039	AA-10375	Allen Armold	2022-09-07	Alex Ferguson	J-2003	Jeans	Chinos	200
2	131065	AA-10375	Allen Armold	2022-09-07	Alex Ferguson	J-2002	Jeans	Blue Denim	30
3	131065	AA-10375	Allen Armold	2022-12-11	Alex Ferguson	J-2003	Jeans	Chinos	120
4	169488	AA-10480	Andrew Allen	2021-07-17	Samuel Washington	J-2003	Jeans	Chinos	120
...	...	...	...	...	...	...	...	...	...
3427	167682	ZC-21910	Zuschuss Carroll	2021-03-08	John Blake	SO-3003	Shoes	Formal Shoes	270
3428	147991	ZC-21910	Zuschuss Carroll	2021-04-08	Jane Austin	SO-3003	Shoes	Formal Shoes	360
3429	152471	ZC-21910	Zuschuss Carroll	2022-11-06	Mike Davidson	SO-3001	Shoes	Sneakers	50
3430	152471	ZD-21925	Zuschuss Donatelli	2021-04-03	Jane Austin	SO-3002	Shoes	Running Shoes	280
3431	141481	ZD-21925	Zuschuss Donatelli	2021-05-05	John Blake	SO-3001	Shoes	Sneakers	250

3432 rows × 11 columns

In [102…

```
freq3 = df0.groupby('Customer ID')['Dollar Sales'].sum()
freq3
```

Out[102]:

```
Customer ID
AA-10375     451
AA-10480     744
AA-10645    1060
AB-10015     136
AB-10060    1417
            ...
XP-21865     854
YC-21895     265
YS-21880     690
ZC-21910    1633
ZD-21925     822
Name: Dollar Sales, Length: 739, dtype: int64
```

In [103…

```
freq4 = pd.merge(freq2,freq3, how='right',left_on=['Customer ID'],right_on=['Custor
freq4
```

Out[103]:

	Customer ID	freq	Disc Category	Dollar Sales
0	AA-10375	1	10% disc	451
1	AA-10480	2	10% disc	744
2	AA-10645	3	10% disc	1060
3	AB-10015	1	10% disc	136
4	AB-10060	1	10% disc	1417
...	...	...	...	...
733	WB-21850	1	10% disc	1774
734	XP-21865	1	10% disc	854
735	YC-21895	1	10% disc	265
737	ZC-21910	1	10% disc	1633
738	ZD-21925	1	10% disc	822

550 rows × 4 columns

In [104…

```
#QUE1 CLASSIFICATION OF CUSTOMERS ON THE BASIS OF DISCOUNT PERCENT
conditions = [(freq4['freq']>8) | (freq4['Dollar Sales']>5000),((freq4['freq']>=5)&
values =['30% disc','20% disc','10% disc']

freq4['Disc Category'] = np.select(conditions, values)

freq4
```

Out[104]:

	Customer ID	freq	Disc Category	Dollar Sales
0	AA-10375	1	10% disc	451
1	AA-10480	2	10% disc	744
2	AA-10645	3	10% disc	1060
3	AB-10015	1	10% disc	136
4	AB-10060	1	10% disc	1417
...	...	...	...	...
733	WB-21850	1	10% disc	1774
734	XP-21865	1	10% disc	854
735	YC-21895	1	10% disc	265
737	ZC-21910	1	10% disc	1633
738	ZD-21925	1	10% disc	822

550 rows × 4 columns

In [105…

```
#QUE1 COUNT OF TOTAL CUSTOMERS FALLING IN DIFF. CATEGORY OF DISCOUNTS
count1=len(freq4[freq4['Disc Category']=='10% disc'])
print('TOTAL CUSTOMERS WITH 10% DISCOUNT ARE - ',count1)
count2=len(freq4[freq4['Disc Category']=='20% disc'])
print('TOTAL CUSTOMERS WITH 20% DISCOUNT ARE - ',count2)
count3=len(freq4[freq4['Disc Category']=='30% disc'])
print('TOTAL CUSTOMERS WITH 30% DISCOUNT ARE - ',count3)
```

```
TOTAL CUSTOMERS WITH 10% DISCOUNT ARE -  515
TOTAL CUSTOMERS WITH 20% DISCOUNT ARE -  35
TOTAL CUSTOMERS WITH 30% DISCOUNT ARE -  0
```

In [106…
```
freq4[freq4['Dollar Sales']==freq4['Dollar Sales'].max()]
```

Out[106]:

	Customer ID	freq	Disc Category	Dollar Sales
669	SV-20365	1	20% disc	3172

In [107…
```
#QUE2 printing the table containing Customer ID, number of transactions made, categ
freq4.rename(columns = {'freq':'number of transactions made'}, inplace = True)
freq4
```

Out[107]:

	Customer ID	number of transactions made	Disc Category	Dollar Sales
0	AA-10375	1	10% disc	451
1	AA-10480	2	10% disc	744
2	AA-10645	3	10% disc	1060
3	AB-10015	1	10% disc	136
4	AB-10060	1	10% disc	1417
...	...	...	...	...
733	WB-21850	1	10% disc	1774
734	XP-21865	1	10% disc	854
735	YC-21895	1	10% disc	265
737	ZC-21910	1	10% disc	1633
738	ZD-21925	1	10% disc	822

550 rows × 4 columns

In [108…
```
#importing the data for last 6 months
df0["Order Date"] = pd.to_datetime(df0['Order Date'])
newdf1 = (df0['Order Date'] > '2022-07-01')
newdf1 = df0.loc[newdf1]
newdf1
```

Out[108]:

	Transaction ID	Customer ID	Customer Name	Order Date	Sales Person Name	Product ID	Category	Product Name	Dollar Sales
1	147039	AA-10375	Allen Armold	2022-09-07	Alex Ferguson	J-2003	Jeans	Chinos	200
2	131065	AA-10375	Allen Armold	2022-09-07	Alex Ferguson	J-2002	Jeans	Blue Denim	30
3	131065	AA-10375	Allen Armold	2022-12-11	Alex Ferguson	J-2003	Jeans	Chinos	120
9	121671	AA-10645	Anna Andreadi	2022-11-05	Mike Davidson	J-2001	Jeans	Black Denim	50
13	114601	AB-10060	Adam Bellavance	2022-09-16	Alex Ferguson	J-2002	Jeans	Blue Denim	60
...	...	...	...	...	...	...	...	...	...
3415	169103	VG-21790	Vivek Gonzalez	2022-07-29	Mike Davidson	SO-3001	Shoes	Sneakers	100
3417	156986	VP-21760	Victoria Pisteka	2022-10-16	John Blake	SO-3003	Shoes	Formal Shoes	630
3423	102288	WB-21850	William Brown	2022-12-10	Alex Ferguson	SO-3002	Shoes	Running Shoes	350
3426	167682	YS-21880	Yana Sorensen	2022-08-18	Mike Davidson	SO-3001	Shoes	Sneakers	150
3429	152471	ZC-21910	Zuschuss Carroll	2022-11-06	Mike Davidson	SO-3001	Shoes	Sneakers	50

1256 rows × 11 columns

In [109…

```
#finding the sum of total dollar sales in part 6 months
sum1 = newdf1.groupby('Customer ID')['Dollar Sales'].sum()
sum1
```

Out[109]:

```
Customer ID
AA-10375     380
AA-10645      50
AB-10060    1119
AB-10105     547
AB-10150     320
            ...
VW-21775     130
WB-21850     350
XP-21865     315
YS-21880     150
ZC-21910      50
Name: Dollar Sales, Length: 502, dtype: int64
```

In [110…

```
#QUE3 printing the top 10 customers in part 6 months based on their Purchase Amount
sorted1=sum1.sort_values(ascending=False)
sorted1.head(10)
```

Out[110]:
```
Customer ID
SV-20365    2562
RB-19795    1981
KH-16510    1846
PO-18850    1620
AH-10075    1470
LC-16870    1458
BS-11380    1381
SF-20200    1325
MS-17980    1265
SN-20710    1260
Name: Dollar Sales, dtype: int64
```

In [113…
```
sum2 = df0.groupby('Sales Person Name')['Dollar Sales'].sum()
sum2
```

Out[113]:
```
Sales Person Name
Alex Ferguson         86978
Jane Austin           91596
John Blake            80111
Mike Davidson        165356
Samuel Washington     33257
Saul Goodman          52270
Name: Dollar Sales, dtype: int64
```

In [114…
```
#QUE4 PRINTING THE TOP 2 SALESMAN
sorted2=sum2.sort_values(ascending=False)
sorted2.head(2)
```

Out[114]:
```
Sales Person Name
Mike Davidson    165356
Jane Austin       91596
Name: Dollar Sales, dtype: int64
```

In [115…
```
sum3 = df0.groupby('Product Name')['Quantity'].sum()
sum3
```

Out[115]:
```
Product Name
Black Denim          1405
Blue Denim           1378
Chinos               1338
Formal Shoes         1554
Full Sleeve Shirt    1368
Half Sleeve Shirt    1368
Running Shoes        1539
Sneakers             1537
T Shirt              1315
Name: Quantity, dtype: int64
```

In [117…
```
#QUE5 TOP SELLING PRODUCTS
sorted3=sum3.sort_values(ascending=False)
sorted3
```

Out[117]:
```
Product Name
Formal Shoes         1554
Running Shoes        1539
Sneakers             1537
Black Denim          1405
Blue Denim           1378
Full Sleeve Shirt    1368
Half Sleeve Shirt    1368
Chinos               1338
T Shirt              1315
Name: Quantity, dtype: int64
```