1 Importing library and data

In [4]:
```
import pandas as pd
tr = pd.read_csv('trans.csv')
```

2 Check the data type

In [15]:
```
tr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1119 entries, 0 to 1118
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   OID                     1119 non-null   object 
 1   bonusPointsEarned       544 non-null    float64
 2   bonusPointsEarnedReason 544 non-null    object 
 3   CREATEDATE              1119 non-null   object 
 4   DATESCANNED             1119 non-null   object 
 5   FINISHEDDATE            568 non-null    object 
 6   MODIFYDATE              1119 non-null   object 
 7   AWARDDATE               537 non-null    object 
 8   pointsEarned            609 non-null    float64
 9   PURCHASEDATE            671 non-null    object 
 10  purchasedItemCount      635 non-null    float64
 11  rewardsReceiptStatus    1119 non-null   object 
 12  totalSpent              684 non-null    float64
 13  userId                  1119 non-null   object 
dtypes: float64(4), object(10)
memory usage: 122.5+ KB
```

In [16]:
```
tr.head()
```

Out[16]:

	OID	bonusPointsEarned	bonusPointsEarnedReason	CREATEDATE	DAT
0	5ff1e1eb0a720f0523000575	500.0	Receipt number 2 completed, bonus point schedu...	2021-01-03 10:25:31	
1	5ff1e1bb0a720f052300056b	150.0	Receipt number 5 completed, bonus point schedu...	2021-01-03 10:24:43	
2	5ff1e1f10a720f052300057a	5.0	All-receipts receipt bonus	2021-01-03 10:25:37	
3	5ff1e1ee0a7214ada100056f	5.0	All-receipts receipt bonus	2021-01-03 10:25:34	
4	5ff1e1d20a7214ada1000561	5.0	All-receipts receipt bonus	2021-01-03 10:25:06	

In [5]:

```
# Check the type of each feature in the trans table
print(tr.dtypes)
```

```
OID                           object
bonusPointsEarned            float64
bonusPointsEarnedReason       object
CREATEDATE                    object
DATESCANNED                   object
FINISHEDDATE                  object
MODIFYDATE                    object
AWARDDATE                     object
pointsEarned                 float64
PURCHASEDATE                  object
purchasedItemCount           float64
rewardsReceiptStatus          object
totalSpent                   float64
userId                        object
dtype: object
```

3 The number of null value of each column

In [13]:

```
# Count the number of null value of each column
num_missing = tr.isnull().sum()
print(num_missing)
```

```
OID                          0
bonusPointsEarned          575
bonusPointsEarnedReason    575
CREATEDATE                   0
DATESCANNED                  0
FINISHEDDATE               551
MODIFYDATE                   0
AWARDDATE                  582
pointsEarned               510
PURCHASEDATE               448
purchasedItemCount         484
rewardsReceiptStatus         0
totalSpent                 435
userId                       0
dtype: int64
```

In [19]:

```
# Calculate the percentage of null values in each column
null_percentages = (tr.isnull().sum() / len(tr)) * 100

print(null_percentages)
```

```
OID                         0.000000
bonusPointsEarned          51.385165
bonusPointsEarnedReason    51.385165
CREATEDATE                  0.000000
DATESCANNED                 0.000000
FINISHEDDATE               49.240393
MODIFYDATE                  0.000000
AWARDDATE                  52.010724
pointsEarned               45.576408
PURCHASEDATE               40.035746
purchasedItemCount         43.252904
rewardsReceiptStatus        0.000000
totalSpent                 38.873995
userId                      0.000000
dtype: float64
```

According to the result above, we can find that there are a lot of missing value in the dataset. Therefore, it might lead to inaccurate analysis results.

4 Percentage of duplicate value in 'DATESCANNED' column

In [30]:

```
# Check the number of duplicate value in 'userId' column
duplicate_var = tr['DATESCANNED'].duplicated().sum()
print(duplicate_var)
```

```
13
```

In [33]:

```
# Calculate the percentage of duplicate value in 'userId' column
duplicates_num = (tr['DATESCANNED'].count() - tr['DATESCANNED'].nunique())
percentage_dup = (duplicates_num / tr['DATESCANNED'].count()) * 100
print("The percentage of dupplicate value in 'DATESCANNED' column is: ",
                                                            percentage_dup)
```

```
The percentage of dupplicate value in 'DATESCANNED' column is:  1.16175156389
6336
```

5 Check for the latest date and earliest date in the 'date' column

In [32]:

```
# convert date column to datetime object
tr['DATESCANNED'] = pd.to_datetime(tr['DATESCANNED'])

# find the earliest date
earliest_date = tr['DATESCANNED'].min()

# find the latest date
latest_date = tr['DATESCANNED'].max()

print('Earliest date:', earliest_date)
print('Latest date:', latest_date)
```

```
Earliest date: 2020-10-30 16:17:59
Latest date: 2021-03-01 18:17:34
```

```
According to the result above, we can find out what date we should set in
our query. (The most recent month and perivous month)
```