Loading data and Preprocessing

In [2]:
```python
import pandas as pd

df_customer = pd.read_excel("C:/Users/User/Desktop/DSA_-_CIC_Case_Study_Rubric_and_Data_Sets/Customer_data.xlsx")
df_sales = pd.read_excel("C:/Users/User/Desktop/DSA_-_CIC_Case_Study_Rubric_and_Data_Sets/Sales_data.xlsx")
df_store = pd.read_excel("C:/Users/User/Desktop/DSA_-_CIC_Case_Study_Rubric_and_Data_Sets/Store_data.xlsx")
```

In [3]:
```python
print(df_customer.shape)
print(df_sales.shape)
print(df_store.shape)
```
```
(54954, 8)
(403542, 6)
(67, 9)
```

In [4]:
```python
df_customer.head(10)
```
Out[4]:

	INSTNC_ID	CUSTOMER_ID	CITY	STATE	ZIP_CD	CUSTOMER_FIRST_PURCHASE	CUSTOMER_LAST_PURCHASE	CUSTOMER_PRIMARY_STORE_ID
0	1720109167	1290081864	AUSTIN	TX	78701	2021-01-04	2025-02-12	425.0
1	1760514191	1277673459	AUSTIN	TX	78701	2021-01-07	2025-02-22	428.0
2	1961170347	59863515	AUSTIN	TX	78701	2021-01-02	2025-02-20	768.0
3	1973432551	1480074402	AUSTIN	TX	78701	2021-01-28	2025-02-22	768.0
4	2097873243	1573404921	AUSTIN	TX	78701	2021-01-04	2025-02-23	404.0
5	2203539131	1318355205	AUSTIN	TX	78701	2021-01-28	2025-02-13	91.0
6	2228460375	1310887758	AUSTIN	TX	78701	2021-01-01	2025-02-23	428.0
7	2233668523	1148784342	AUSTIN	TX	78701	2021-01-05	2025-02-14	61.0
8	2249603179	1554595368	AUSTIN	TX	78701	2021-01-01	2025-02-23	774.0
9	2259084439	1694313318	AUSTIN	TX	78701	2021-06-29	2025-02-12	768.0

In [5]:
```python
df_sales.head(10)
```
Out[5]:

	CORP_ID	CUSTOMER_ID	ID_SHOPPING_TRANSACTION	DATE_ID	ITEM_SALES	DOLLAR_SALES
0	68	7787085	68011985780641	2025-01-19	4.191	22.26
1	428	1367615838	428011985772260	2025-01-19	22.000	113.35
2	29	1664880525	29011985977761	2025-01-19	11.875	53.26
3	404	1505734824	404011985114416	2025-01-19	3.000	20.98
4	768	1459480929	768112884238849	2024-11-28	9.000	64.02
5	768	1325541042	768122384364457	2024-12-23	15.835	87.45
6	754	1539490902	754122384164617	2024-12-23	11.000	29.74
7	591	1534805484	591112884426517	2024-11-28	13.000	69.05
8	591	1744414479	591112884424913	2024-11-28	6.000	23.40
9	768	2023802511	768122384365757	2024-12-23	40.310	89.00

In [6]:
```python
df_store.head(10)
```
Out[6]:

	LOB	REGION	ID_RGN	DISTRICT	CITY	STR_FORMAT	STR_SEGMENT	CORP_ID	STORE
0	SA F/D	CNTRL TEXAS	3	AUSTIN WEST	BEE CAVE	Non-Plus	Up	404	AUS21
1	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	BASTROP	Plus	Core	582	BASTROP
2	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	LULING	Non-Plus	Core	464	LULING
3	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	AUSTIN	Non-Plus	Up	639	AUS30
4	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	AUSTIN	Non-Plus	Up	465	AUS01
5	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	BUDA	Non-Plus	Core	477	BUDA
6	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	WIMBERLEY	Non-Plus	Up	708	WIMBERLEY
7	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	AUSTIN	Non-Plus	Core	229	AUS02
8	SA F/D	CNTRL TEXAS	3	AUSTIN SOUTH	LOCKHART	Non-Plus	Core	445	LOCKHART
9	SA F/D	CNTRL TEXAS	3	AUSTIN WEST	LAKEWAY	Non-Plus	Up	714	LAKEWAY

Removing (-) in Cusomter_ID

In [8]:
```python
df_negative = df_customer[df_customer['CUSTOMER_ID'] <= 0]
print(df_negative)
```
```
        INSTNC_ID  CUSTOMER_ID    CITY STATE  ZIP_CD CUSTOMER_FIRST_PURCHASE  \
6676   2811450599  -2108587950  AUSTIN    TX   78703              2025-01-30   
11693  2757773219  -2068329915  AUSTIN    TX   78703              2025-01-31   
17781  2745739775  -2059304829  AUSTIN    TX   78704              2025-01-28   
18401  2745739771  -2059304829  AUSTIN    TX   78704              2025-01-28   
19795  2683205039  -2012403780  AUSTIN    TX   78704              2024-08-12   
27967  2745737735  -2059303302  AUSTIN    TX   78704              2025-01-28   
32927  2597456079  -1948092060  AUSTIN    TX   78704              2024-02-07   
34163  2650242759  -1987211466  AUSTIN    TX   78704              2024-05-31   
36760  2695792655  -2021844492  AUSTIN    TX   78705              2024-09-24   
37035  2814426251  -2110819689  AUSTIN    TX   78705              2025-02-03   
42199  2684623963  -2013467973  AUSTIN    TX   78746              2024-08-17   
47219  2733858127  -2050393596  AUSTIN    TX   78746              2024-12-28   
47700  2648814187  -1986486324  AUSTIN    TX   78746              2024-05-31   
52998  2299833983  -1724875488  AUSTIN    TX   78746              2021-12-12   
53393  2733225131  -2049918849  AUSTIN    TX   78746              2024-12-26   

      CUSTOMER_LAST_PURCHASE  CUSTOMER_PRIMARY_STORE_ID  
6676              2025-02-17                      768.0  
11693             2025-02-21                       31.0  
17781             2025-02-23                      754.0  
18401             2025-02-23                      754.0  
19795             2024-12-19                      639.0  
27967             2025-01-30                      754.0  
32927             2025-02-10                        NaN  
34163             2025-02-22                      768.0  
36760             2025-02-17                      768.0  
37035             2025-02-22                      768.0  
42199             2024-12-13                       29.0  
47219             2025-02-23                      754.0  
47700             2025-02-22                      768.0  
52998             2025-02-15                       29.0  
53393             2024-12-26                      768.0  
```

In [9]:
```python
df_customer['CUSTOMER_ID'] = df_customer['CUSTOMER_ID'].abs()
```

In [10]:
```python
print(df_customer['CUSTOMER_ID'] <= 0)
```
```
0        False
1        False
2        False
3        False
4        False
         ...  
54949    False
54950    False
54951    False
54952    False
54953    False
Name: CUSTOMER_ID, Length: 54954, dtype: bool
```

Checking for duplicates

In [12]:
```python
print(df_customer.duplicated().sum())
print(df_sales.duplicated().sum())
print(df_store.duplicated().sum())
```
```
0
0
0
```

In [13]:
```python
duplicate_customer_ids = df_customer[df_customer.duplicated(subset=['CUSTOMER_ID','CITY','STATE','ZIP_CD','CUSTOMER_FIRST_PURCHASE','CUSTOMER_LAST_PURCHASE','CUSTOMER_PRIMARY_STORE_ID'])]
print("\nDuplicate Rows (Based on CUSTOMER_ID):")
print(duplicate_customer_ids)
```
```
Duplicate Rows (Based on CUSTOMER_ID):
        INSTNC_ID  CUSTOMER_ID    CITY STATE  ZIP_CD CUSTOMER_FIRST_PURCHASE  \
52     2706836215   1277673459  AUSTIN    TX   78701              2021-01-07   
133    2707389555   1912146195  AUSTIN    TX   78701              2023-08-31   
184    2563052859   1195573380  AUSTIN    TX   78701              2021-01-01   
280    2739719019   1411615887  AUSTIN    TX   78701              2021-01-09   
283    2742323879   1374996126  AUSTIN    TX   78701              2021-01-03   
...           ...          ...     ...   ...     ...                     ...   
54949  2812476203   1329835212  AUSTIN    TX   78746              2021-01-30   
54950  2812624663   1689941175  AUSTIN    TX   78746              2021-05-20   
54951  2816805271   1452243828  AUSTIN    TX   78746              2021-01-01   
54952  2819499419   1294661880  AUSTIN    TX   78746              2021-01-01   
54953  2820240855   1327464327  AUSTIN    TX   78746              2021-01-02   

      CUSTOMER_LAST_PURCHASE  CUSTOMER_PRIMARY_STORE_ID  
52                2025-02-22                      428.0  
133               2025-02-21                      768.0  
184               2025-02-19                      768.0  
280               2025-02-20                       34.0  
283               2025-02-23                      455.0  
...                      ...                        ...  
54949             2025-02-12                      425.0  
54950             2025-02-23                      768.0  
54951             2025-02-22                       29.0  
54952             2025-02-11                       29.0  
54953             2025-02-23                       29.0  

[24026 rows x 8 columns]
```

In [14]:
```python
df_customer = df_customer.drop_duplicates(subset=['CUSTOMER_ID', 'CITY', 'STATE', 'ZIP_CD', 'CUSTOMER_FIRST_PURCHASE', 'CUSTOMER_LAST_PURCHASE', 'CUSTOMER_PRIMARY_STORE_ID'])

print(df_customer.shape)
```
```
(30928, 8)
```

Checking for nulls

In [16]:
```python
print(df_customer.isnull().sum())
```
```
INSTNC_ID                      0
CUSTOMER_ID                    0
CITY                           0
STATE                          1
ZIP_CD                         0
CUSTOMER_FIRST_PURCHASE        6
CUSTOMER_LAST_PURCHASE         6
CUSTOMER_PRIMARY_STORE_ID    562
dtype: int64
```

In [17]:
```python
print(df_customer[df_customer["STATE"].isnull()])
```
```
        INSTNC_ID  CUSTOMER_ID    CITY STATE  ZIP_CD CUSTOMER_FIRST_PURCHASE  \
43285  2749085679     33516288  AUSTIN   NaN   78746              2021-01-01   

      CUSTOMER_LAST_PURCHASE  CUSTOMER_PRIMARY_STORE_ID  
43285             2025-02-23                      373.0  
```

In [18]:
```python
df_customer["STATE"].fillna("TX", inplace=True)
```
```
C:\Users\User\AppData\Local\Temp\ipykernel_21244\1009564771.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df_customer["STATE"].fillna("TX", inplace=True)
```

In [19]:
```python
print(df_customer[df_customer["CUSTOMER_FIRST_PURCHASE"].isnull()])
```
```
        INSTNC_ID  CUSTOMER_ID    CITY STATE  ZIP_CD CUSTOMER_FIRST_PURCHASE  \
21300  2736661183   2052495876  AUSTIN    TX   78704                     NaT   
22156  2690366599   1415727180  AUSTIN    TX   78704                     NaT   
26446  2729875955   2009424852  AUSTIN    TX   78704                     NaT   
31123  2509260567   1881945414  AUSTIN    TX   78704                     NaT   
43769  2715562847   1362259254  AUSTIN    TX   78746                     NaT   
49640  2557843603   1918382691  AUSTIN    TX   78746                     NaT   

      CUSTOMER_LAST_PURCHASE  CUSTOMER_PRIMARY_STORE_ID  
21300                    NaT                        NaN  
22156                    NaT                        NaN  
26446                    NaT                        NaN  
31123                    NaT                        NaN  
43769                    NaT                        NaN  
49640                    NaT                        NaN  
```

In [20]:
```python
print(df_customer[df_customer["CUSTOMER_PRIMARY_STORE_ID"].isnull()])
```
```
        INSTNC_ID  CUSTOMER_ID    CITY STATE  ZIP_CD CUSTOMER_FIRST_PURCHASE  \
44     2684481339   1440251898  AUSTIN    TX   78701              2021-01-07   
58     2743080079   1475184252  AUSTIN    TX   78701              2022-03-24   
81     2142960155   1607220105  AUSTIN    TX   78701              2021-01-04   
110    2574314323   1930525413  AUSTIN    TX   78701              2023-11-19   
292    2030787703   1523090766  AUSTIN    TX   78701              2021-01-24   
...           ...          ...     ...   ...     ...                     ...   
54287  2368366927   1776275184  AUSTIN    TX   78746              2022-09-13   
54323  2579692455   1410023796  AUSTIN    TX   78746              2021-04-21   
54368  2697962655   1256724165  AUSTIN    TX   78746              2021-03-21   
54510  2324732887   1336982940  AUSTIN    TX   78746              2022-03-23   
54694  1770149827   1251993705  AUSTIN    TX   78746              2021-01-09   

      CUSTOMER_LAST_PURCHASE  CUSTOMER_PRIMARY_STORE_ID  
44                2025-02-18                        NaN  
58                2024-02-13                        NaN  
81                2024-11-25                        NaN  
110               2024-12-22                        NaN  
292               2025-01-03                        NaN  
...                      ...                        ...  
54287             2025-01-19                        NaN  
54323             2024-12-06                        NaN  
54368             2024-11-28                        NaN  
54510             2024-07-13                        NaN  
54694             2024-12-09                        NaN  

[562 rows x 8 columns]
```

In [21]:
```python
df_customer.dropna(subset=["CUSTOMER_FIRST_PURCHASE"], inplace=True)
```

In [22]:
```python
print(df_customer.isnull().sum())
```
```
INSTNC_ID                      0
CUSTOMER_ID                    0
CITY                           0
STATE                          0
ZIP_CD                         0
CUSTOMER_FIRST_PURCHASE        0
CUSTOMER_LAST_PURCHASE         0
CUSTOMER_PRIMARY_STORE_ID    556
dtype: int64
```

In [23]:
```python
print(df_sales.isnull().sum())
```
```
CORP_ID                    0
CUSTOMER_ID                0
ID_SHOPPING_TRANSACTION    0
DATE_ID                    0
ITEM_SALES                 0
DOLLAR_SALES               0
dtype: int64
```

In [24]:
```python
print(df_store.isnull().sum())
```
```
LOB            0
REGION         0
ID_RGN         0
DISTRICT       0
CITY           0
STR_FORMAT     3
STR_SEGMENT    5
CORP_ID        0
STORE          0
dtype: int64
```

In [25]:
```python
print(df_store[df_store["STR_FORMAT"].isnull()])
```
```
       LOB       REGION  ID_RGN      DISTRICT        CITY STR_FORMAT  \
45  SA F/D  CNTRL TEXAS       3  CORE RESERVE     LEANDER        NaN   
52  SA F/D  CNTRL TEXAS       3  CORE RESERVE      AUSTIN        NaN   
56  SA F/D  CNTRL TEXAS       3  CORE RESERVE  ROUND ROCK        NaN   

   STR_SEGMENT  CORP_ID   STORE  
45         NaN      764   EFC#4  
52         NaN      140  EFC#11  
56         NaN      763   EFC#3  
```

In [26]:
```python
print(df_store[df_store["STR_SEGMENT"].isnull()])
```
```
       LOB       REGION  ID_RGN      DISTRICT        CITY STR_FORMAT  \
26  SA F/D  CNTRL TEXAS       3  AUSTIN NORTH       MANOR   Non-Plus   
45  SA F/D  CNTRL TEXAS       3  CORE RESERVE     LEANDER        NaN   
52  SA F/D  CNTRL TEXAS       3  CORE RESERVE      AUSTIN        NaN   
56  SA F/D  CNTRL TEXAS       3  CORE RESERVE  ROUND ROCK        NaN   
57  SA F/D  CNTRL TEXAS       3  AUSTIN SOUTH  San Marcos   Non-Plus   

   STR_SEGMENT  CORP_ID        STORE  
26         NaN      811        MANOR  
45         NaN      764        EFC#4  
52         NaN      140       EFC#11  
56         NaN      763        EFC#3  
57         NaN      822  SAN MARCOS2  
```

In [27]:
```python
df_store.dropna(subset=["STR_FORMAT"], inplace=True)
```

In [28]:
```python
print(df_store.isnull().sum())
```
```
LOB            0
REGION         0
ID_RGN         0
DISTRICT       0
CITY           0
STR_FORMAT     0
STR_SEGMENT    2
CORP_ID        0
STORE          0
dtype: int64
```

Checking dtype formatting

In [30]:
```python
df_customer.dtypes
```
Out[30]:
```
INSTNC_ID                             int64
CUSTOMER_ID                           int64
CITY                                 object
STATE                                object
ZIP_CD                                int64
CUSTOMER_FIRST_PURCHASE      datetime64[ns]
CUSTOMER_LAST_PURCHASE       datetime64[ns]
CUSTOMER_PRIMARY_STORE_ID           float64
dtype: object
```

In [31]:
```python
df_customer["CUSTOMER_PRIMARY_STORE_ID"] = pd.to_numeric(df_customer["CUSTOMER_PRIMARY_STORE_ID"], errors='coerce').astype("Int64")
df_customer.dtypes
```
Out[31]:
```
INSTNC_ID                             int64
CUSTOMER_ID                           int64
CITY                                 object
STATE                                object
ZIP_CD                                int64
CUSTOMER_FIRST_PURCHASE      datetime64[ns]
CUSTOMER_LAST_PURCHASE       datetime64[ns]
CUSTOMER_PRIMARY_STORE_ID             Int64
dtype: object
```

In [32]:
```python
df_sales.dtypes
```
Out[32]:
```
CORP_ID                             int64
CUSTOMER_ID                         int64
ID_SHOPPING_TRANSACTION             int64
DATE_ID                    datetime64[ns]
ITEM_SALES                        float64
DOLLAR_SALES                      float64
dtype: object
```

In [33]:
```python
df_sales = df_sales.round(2)
```

In [34]:
```python
df_store.dtypes
```
Out[34]:
```
LOB            object
REGION         object
ID_RGN          int64
DISTRICT       object
CITY           object
STR_FORMAT     object
STR_SEGMENT    object
CORP_ID         int64
STORE          object
dtype: object
```

Determine the day difference from First to last day purchase

In [36]:
```python
df_customer["time_difference_day"] = (df_customer["CUSTOMER_LAST_PURCHASE"] - df_customer["CUSTOMER_FIRST_PURCHASE"]).dt.days
df_customer.drop(columns=["INSTNC_ID"], inplace=True)
```

In [37]:
```python
df_customer.head(10)
```
Out[37]:

	CUSTOMER_ID	CITY	STATE	ZIP_CD	CUSTOMER_FIRST_PURCHASE	CUSTOMER_LAST_PURCHASE	CUSTOMER_PRIMARY_STORE_ID	time_difference_day
0	1290081864	AUSTIN	TX	78701	2021-01-04	2025-02-12	425	1500
1	1277673459	AUSTIN	TX	78701	2021-01-07	2025-02-22	428	1507
2	59863515	AUSTIN	TX	78701	2021-01-02	2025-02-20	768	1510
3	1480074402	AUSTIN	TX	78701	2021-01-28	2025-02-22	768	1486
4	1573404921	AUSTIN	TX	78701	2021-01-04	2025-02-23	404	1511
5	1318355205	AUSTIN	TX	78701	2021-01-28	2025-02-13	91	1477
6	1310887758	AUSTIN	TX	78701	2021-01-01	2025-02-23	428	1514
7	1148784342	AUSTIN	TX	78701	2021-01-05	2025-02-14	61	1501
8	1554595368	AUSTIN	TX	78701	2021-01-01	2025-02-23	774	1514
9	1694313318	AUSTIN	TX	78701	2021-06-29	2025-02-12	768	1324

Create new DF with the ZIP_CD [78701, 78703, 78704]

In [39]:
```python
zip_codes = [78701, 78703, 78704]
df_filtered = df_customer[df_customer["ZIP_CD"].isin(zip_codes)]
```

In [40]:
```python
df_filtered.head(10)
```
Out[40]:

	CUSTOMER_ID	CITY	STATE	ZIP_CD	CUSTOMER_FIRST_PURCHASE	CUSTOMER_LAST_PURCHASE	CUSTOMER_PRIMARY_STORE_ID	time_difference_day
0	1290081864	AUSTIN	TX	78701	2021-01-04	2025-02-12	425	1500
1	1277673459	AUSTIN	TX	78701	2021-01-07	2025-02-22	428	1507
2	59863515	AUSTIN	TX	78701	2021-01-02	2025-02-20	768	1510
3	1480074402	AUSTIN	TX	78701	2021-01-28	2025-02-22	768	1486
4	1573404921	AUSTIN	TX	78701	2021-01-04	2025-02-23	404	1511
5	1318355205	AUSTIN	TX	78701	2021-01-28	2025-02-13	91	1477
6	1310887758	AUSTIN	TX	78701	2021-01-01	2025-02-23	428	1514
7	1148784342	AUSTIN	TX	78701	2021-01-05	2025-02-14	61	1501
8	1554595368	AUSTIN	TX	78701	2021-01-01	2025-02-23	774	1514
9	1694313318	AUSTIN	TX	78701	2021-06-29	2025-02-12	768	1324

In [41]:
```python
df_filtered.shape
```
Out[41]:
```
(20275, 8)
```

Look at Sales from the Store 754 & 768

In [43]:
```python
stores = [754,768]
sale_stores = df_sales[df_sales["CORP_ID"].isin(stores)]
```

In [44]:
```python
sale_stores.shape
```
Out[44]:
```
(162842, 6)
```

In [45]:
```python
sales_customer_df = pd.merge(sale_stores, df_customer, on='CUSTOMER_ID', how='inner')
```

In [46]:
```python
sales_customer_df.head(50)
```
Out[46]:

	CORP_ID	CUSTOMER_ID	ID_SHOPPING_TRANSACTION	DATE_ID	ITEM_SALES	DOLLAR_SALES	CITY	STATE	ZIP_CD	CUSTOMER_FIRST_PURCHASE	CUSTOMER_LAST_PURCHASE	CUSTOMER_PRIMARY_STORE_ID	time_difference_day
0	768	1459480929	768112884238849	2024-11-28	9.00	64.02	AUSTIN	TX	78703	2021-01-01	2025-02-23	768	1514
1	768	1325541042	768122384364457	2024-12-23	15.84	87.45	AUSTIN	TX	78703	2021-01-03	2025-02-23	61	1512
2	754	1539490902	754122384164617	2024-12-23	11.00	29.74	AUSTIN	TX	78704	2021-03-21	2025-02-20	754	1432
3	768	2023802511	768122384365757	2024-12-23	40.31	89.00	AUSTIN	TX	78704	2024-10-03	2025-01-31	754	120
4	768	1683293418	768112884238863	2024-11-28	8.91	10.60	AUSTIN	TX	78703	2021-01-02	2025-02-20	768	1510
5	768	1532224455	768012385520264	2025-01-23	32.30	121.59	AUSTIN	TX	78703	2021-01-22	2025-02-18	768	1488
6	768	1683250386	768122384363019	2024-12-23	42.00	192.70	AUSTIN	TX	78746	2022-12-05	2025-01-29	768	786
7	768	1478317911	768112884240034	2024-11-28	15.47	30.90	AUSTIN	TX	78746	2021-01-04	2025-02-21	742	1509
8	768	1724084220	768122384363222	2024-12-23	36.55	87.23	AUSTIN	TX	78703	2021-12-08	2025-02-20	768	1170
9	768	1262090934	768122384364075	2024-12-23	16.00	75.24	AUSTIN	TX	78703	2021-01-02	2025-02-23	116	1513
10	768	1287122433	768112884240047	2024-11-28	8.62	15.27	AUSTIN	TX	78703	2021-01-03	2025-02-23	768	1512
11	768	1101419106	768122384366215	2024-12-23	35.22	105.56	AUSTIN	TX	78703	2022-05-20	2025-01-26	768	982
12	768	1101419106	768122384366215	2024-12-23	35.22	105.56	AUSTIN	TX	78746	2021-05-16	2025-02-21	768	1377
13	768	1227033132	768112884239333	2024-11-28	4.96	24.99	AUSTIN	TX	78746	2021-01-01	2025-02-23	768	1514
14	768	1366789404	768122384362861	2024-12-23	18.71	71.49	AUSTIN	TX	78704	2021-02-10	2025-01-31	420	1451
15	754	1768986720	754122384158722	2024-12-23	2.00	19.24	AUSTIN	TX	78703	2022-08-06	2025-02-20	694	929
16	768	1345650639	768122384366415	2024-12-23	2.00	13.96	AUSTIN	TX	78703	2021-01-06	2025-02-20	768	1506
17	768	1707014385	768112884239011	2024-11-28	46.63	291.74	AUSTIN	TX	78703	2021-09-05	2025-02-20	768	1264
18	768	1700368983	768012385522185	2025-01-23	26.96	90.73	AUSTIN	TX	78703	2021-04-22	2025-01-28	768	1377
19	754	1509024957	754012385382061	2025-01-23	19.39	73.98	AUSTIN	TX	78704	2021-01-03	2025-02-23	754	1512
20	768	1646745219	768112884240201	2024-11-28	6.00	9.41	AUSTIN	TX	78746	2021-01-02	2025-02-23	768	1513
21	768	1833636285	768122384366066	2024-12-23	33.02	247.84	AUSTIN	TX	78703	2021-03-20	2025-02-22	768	1435
22	768	1268796219	768012385520729	2025-01-23	10.39	77.54	AUSTIN	TX	78703	2021-01-31	2025-02-23	611	1484
23	768	1256095443	768012385518708	2025-01-23	14.00	80.14	AUSTIN	TX	78746	2021-01-05	2025-02-18	768	1505
24	754	1443253560	754122384162186	2024-12-23	27.50	64.38	AUSTIN	TX	78704	2021-01-14	2025-02-21	780	1499
25	754	1212601017	754122384161269	2024-12-23	8.00	37.79	AUSTIN	TX	78704	2021-01-02	2025-02-13	754	1503
26	768	299412978	768012385520965	2025-01-23	1.00	38.85	AUSTIN	TX	78746	2021-01-27	2024-05-12	<NA>	1201
27	754	1341160344	754012385383016	2025-01-23	4.00	10.93	AUSTIN	TX	78704	2021-01-01	2025-02-23	754	1514
28	768	1620702159	768112884239522	2024-11-28	10.00	77.03	AUSTIN	TX	78703	2021-01-01	2025-02-23	768	1514
29	768	1899598176	768122384367607	2024-12-23	68.34	340.38	AUSTIN	TX	78703	2022-05-02	2025-02-23	768	1028
30	754	1581353358	754122384162056	2024-12-23	8.00	40.92	AUSTIN	TX	78704	2021-01-01	2025-02-23	754	1514
31	754	1202683269	754122384164206	2024-12-23	16.90	92.70	AUSTIN	TX	78704	2021-01-08	2025-02-20	592	1504
32	754	1274559096	754122384160493	2024-12-23	30.80	87.32	AUSTIN	TX	78704	2021-01-03	2025-02-19	754	1508
33	754	1452070293	754122384162781	2024-12-23	90.64	442.91	AUSTIN	TX	78704	2021-01-05	2025-02-14	754	1501
34	768	1396045323	768112884238955	2024-11-28	14.33	48.29	AUSTIN	TX	78703	2021-01-28	2025-02-23	768	1487
35	754	1716428382	754012385381180	2025-01-23	12.69	55.10	AUSTIN	TX	78704	2021-10-28	2025-02-20	420	1211
36	768	1344080676	768122384367314	2024-12-23	2.00	13.45	AUSTIN	TX	78703	2021-01-07	2025-02-19	768	1504
37	754	1239381366	754122384163601	2024-12-23	1.00	3.19	AUSTIN	TX	78704	2021-01-05	2025-02-19	754	1506
38	768	1318088814	768122384367337	2024-12-23	18.14	51.33	AUSTIN	TX	78704	2021-01-02	2025-02-23	754	1513
39	768	1299410247	768122384364160	2024-12-23	13.30	79.30	AUSTIN	TX	78746	2021-01-03	2025-02-22	768	1511
40	754	1201066107	754012385385784	2025-01-23	5.31	26.63	AUSTIN	TX	78704	2021-01-11	2025-01-26	754	1476
41	768	82697886	768112884239800	2024-11-28	6.00	13.80	AUSTIN	TX	78703	2021-01-03	2025-02-22	61	1511
42	768	1292287923	768012385521781	2025-01-23	1.00	30.00	AUSTIN	TX	78703	2021-01-01	2025-02-22	768	1513
43	768	1591517010	768012385519436	2025-01-23	35.06	158.35	AUSTIN	TX	78704	2021-02-09	2025-01-23	768	1444
44	754	1598660313	754012385382205	2025-01-23	4.00	31.44	AUSTIN	TX	78704	2021-01-01	2025-02-12	420	1503
45	768	1316320743	768112884239193	2024-11-28	6.26	29.78	AUSTIN	TX	78746	2021-01-21	2025-02-01	61	1472
46	754	1731468666	754122384163418	2024-12-23	10.81	41.65	AUSTIN	TX	78704	2021-11-03	2025-02-23	754	1208
47	754	1197396783	754122384164309	2024-12-23	20.00	55.54	AUSTIN	TX	78704	2021-01-01	2025-02-23	754	1514
48	768	1477916724	768112884239122	2024-11-28	11.00	66.09	AUSTIN	TX	78703	2021-01-05	2025-02-23	768	1510
49	768	1747762833	768122384362576	2024-12-23	15.38	49.53	AUSTIN	TX	78701	2021-01-01	2025-02-21	768	1512

In [47]:
```python
print(sales_customer_df.duplicated().sum())
```
```
0
```

In [48]:
```python
print(df_customer.shape)
print(df_sales.shape)
print(df_store.shape)
```
```
(30922, 8)
(403542, 6)
(64, 9)
```

Exploratory Data Analysis

North Star metrics:

- Average Basket Size (Over time): Tracks the average number of items or transaction value per visit over time
- Visits per Customer (Over time): Tracks how often customers shop at the stores over time

Key Perfomance Indicators:

- Average Transaction Value (ATV): average spend per transaction
- Total Spend per Customer: cumulative spend by each customer over a period
- Number of transactions per customer:

In [51]:
```python
import sqlalchemy as sal
engine = sal.create_engine('mssql://DESKTOP-0AO6NA9/master?driver=ODBC+DRIVER+17+FOR+SQL+SERVER')
conn=engine.connect()
```

In [52]:
```python
df_customer.to_sql('df_customer', con=conn , index=False, if_exists = 'replace')
```
Out[52]:
```
6
```

In [53]:
```python
df_sales.to_sql('df_sales', con=conn , index=False, if_exists = 'replace')
```
Out[53]:
```
98
```

In [54]:
```python
df_store.to_sql('df_store', con=conn , index=False, if_exists = 'replace')
```
Out[54]:
```
64
```

In [55]:
```python
print(df_sales.shape)
print(df_customer.shape)
print(df_store.shape)
```
```
(403542, 6)
(30922, 8)
(64, 9)
```

In [ ]: