In [1]:
```
import numpy as np
import pandas as pd
```

In [27]:
```
df_tr = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',sheet_name=
```

In [28]:
```
df_tr['transaction_date'] = pd.to_datetime(df_tr['transaction_date'])
```

In [29]:
```
df_tr['product_first_sold_date'] = pd.to_datetime(df_tr['product_first_sol
```

In [30]:
```
df_tr.head()
```

Out[30]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	bra
0	1	2	2950	2017-02-25	0.0	Approved	So
1	2	3	3120	2017-05-21	1.0	Approved	T Bicy
2	3	37	402	2017-10-16	0.0	Approved	O Cy
3	4	88	3135	2017-08-31	0.0	Approved	No Bicy
4	5	78	787	2017-10-01	1.0	Approved	G Bicy

In [9]:
```
df_newcust = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',sheet_
```

```
/tmp/ipykernel_3439/450638472.py:1: FutureWarning: Inferring datetime64[n
s] from data containing strings is deprecated and will be removed in a fu
ture version. To retain the old behavior explicitly pass Series(data, dty
pe=datetime64[ns])
  df_newcust = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',she
et_name='NewCustomerList',skiprows=1)
```

In [10]:
```
df_newcust.head()
```

Out[10]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job
0	Chickie	Brister	Male	86	1957-07-12	Ge Mar
1	Morly	Genery	Male	69	1970-03-22	Stru Eng
2	Ardelis	Forrester	Female	10	1974-08-28	Senior Accou
3	Lucine	Stutt	Female	64	1979-01-28	Acc Represen
4	Melinda	Hadlee	Female	34	1965-09-21	Fina Ar

5 rows × 23 columns

In [11]:
```
df_custdemo = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',sheet
/tmp/ipykernel_3439/2089153917.py:1: FutureWarning: Inferring datetime64
[ns] from data containing strings is deprecated and will be removed in a
future version. To retain the old behavior explicitly pass Series(data, d
type=datetime64[ns])
  df_custdemo = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',sh
eet_name='CustomerDemographic',skiprows=1)
```

In [12]:
```
df_custdemo.head()
```

Out[12]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	D
0	1	Laraine	Medendorp	F	93	1953-1(
1	2	Eli	Bockman	Male	81	1980-12
2	3	Arlin	Dearle	Male	61	1954-01
3	4	Talbot	NaN	Male	33	1961-1(
4	5	Sheila-kathryn	Calton	Female	56	1977-05

In [13]:
```
df_custaddr = pd.read_excel('KPMG_VI_New_raw_data_update_final.xlsx',sheet
```

In [14]:
```
df_custaddr.head()
```

Out[14]:

	customer_id	address	postcode	state	country	property_valuation
0	1	060 Morning Avenue	2016	New South Wales	Australia	10
1	2	6 Meadow Vale Court	2153	New South Wales	Australia	10
2	4	0 Holy Cross Court	4211	QLD	Australia	9
3	5	17979 Del Mar Point	2448	New South Wales	Australia	4
4	6	9 Oakridge Court	3216	VIC	Australia	9

Transactions DB

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   transaction_id           20000 non-null  int64
 1   product_id               20000 non-null  int64
 2   customer_id              20000 non-null  int64
 3   transaction_date         20000 non-null  datetime64[ns]
 4   online_order             19640 non-null  float64
 5   order_status             20000 non-null  object
 6   brand                    19803 non-null  object
 7   product_line             19803 non-null  object
 8   product_class            19803 non-null  object
 9   product_size             19803 non-null  object
 10  list_price               20000 non-null  float64
 11  standard_cost            19803 non-null  float64
 12  product_first_sold_date  19803 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(3), object(5)
memory usage: 2.0+ MB
```

In [16]:
```
df_tr.isnull().sum()
```

Out[16]:
```
transaction_id               0
product_id                   0
customer_id                  0
transaction_date             0
online_order               360
order_status                 0
brand                      197
product_line               197
product_class              197
product_size               197
list_price                   0
standard_cost              197
product_first_sold_date    197
dtype: int64
```

- Columns `online_order`, `brand`, `product_line`, `product_class`, `product_size`, `standard_cost` and `product_first_sold_date` have missing values. They can either be treated or dropped depending on the analysis

In [18]:
```
df_tr.nunique()
```

Out[18]:
```
transaction_id             20000
product_id                   101
customer_id                 3494
transaction_date             364
online_order                   2
order_status                   2
brand                          6
product_line                   4
product_class                  3
product_size                   3
list_price                   296
standard_cost                103
product_first_sold_date      100
dtype: int64
```

```
In [19]: df_tr.shape
```

```
Out[19]: (20000, 13)
```

- There are no duplicate transactions as the transaction_id column has all unique values

```
In [21]: df_tr.duplicated().sum()
```

```
Out[21]: 0
```

```
In [36]: df_tr['product_first_sold_date'].dt.date.unique()
```

```
Out[36]: array([datetime.date(1970, 1, 1), NaT], dtype=object)
```

- All values in the `product_first_sold_date` are from the same day, this can be an anomaly

New Customers DB

```
In [37]: df_newcust.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 23 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   first_name                           1000 non-null   object
 1   last_name                            971 non-null    object
 2   gender                               1000 non-null   object
 3   past_3_years_bike_related_purchases  1000 non-null   int64
 4   DOB                                  983 non-null    datetime64[ns]
 5   job_title                            894 non-null    object
 6   job_industry_category                835 non-null    object
 7   wealth_segment                       1000 non-null   object
 8   deceased_indicator                   1000 non-null   object
 9   owns_car                             1000 non-null   object
 10  tenure                               1000 non-null   int64
 11  address                              1000 non-null   object
 12  postcode                             1000 non-null   int64
 13  state                                1000 non-null   object
 14  country                              1000 non-null   object
 15  property_valuation                   1000 non-null   int64
 16  Unnamed: 16                          1000 non-null   float64
 17  Unnamed: 17                          1000 non-null   float64
 18  Unnamed: 18                          1000 non-null   float64
 19  Unnamed: 19                          1000 non-null   float64
 20  Unnamed: 20                          1000 non-null   int64
 21  Rank                                 1000 non-null   int64
 22  Value                                1000 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(6), object(11)
memory usage: 179.8+ KB
```

- There are 5 columns unnamed, we'll have to drop them unless we can get the information from the company

```
In [52]: df_newcust.drop([i for i in df_newcust.columns if i.lower()[:7] == 'unname
```

```
In [53]: df_newcust.columns
```

```
Out[53]: Index(['first_name', 'last_name', 'gender',
                'past_3_years_bike_related_purchases', 'DOB', 'job_title',
                'job_industry_category', 'wealth_segment', 'deceased_indicator',
                'owns_car', 'tenure', 'address', 'postcode', 'state', 'country',
                'property_valuation', 'Rank', 'Value'],
               dtype='object')
```

```
In [54]: df_newcust.isnull().sum()
```

```
Out[54]: first_name                               0
         last_name                               29
         gender                                   0
         past_3_years_bike_related_purchases      0
         DOB                                     17
         job_title                              106
         job_industry_category                  165
         wealth_segment                           0
         deceased_indicator                       0
         owns_car                                 0
         tenure                                   0
         address                                  0
         postcode                                 0
         state                                    0
         country                                  0
         property_valuation                       0
         Rank                                     0
         Value                                    0
         dtype: int64
```

- Columns `last_name` , `DOB` , `job_title` and `job_industry_category` have missing values. They can either be treated or dropped depending on the analysis

```
In [55]: df_newcust.duplicated().sum()
```

```
Out[55]: 0
```

- There are no duplicate values

Customer Demographic DB

```
In [56]: df_custdemo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 13 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   customer_id                          4000 non-null   int64
 1   first_name                           4000 non-null   object
 2   last_name                            3875 non-null   object
 3   gender                               4000 non-null   object
 4   past_3_years_bike_related_purchases  4000 non-null   int64
 5   DOB                                  3913 non-null   datetime64[ns]
 6   job_title                            3494 non-null   object
 7   job_industry_category                3344 non-null   object
 8   wealth_segment                       4000 non-null   object
 9   deceased_indicator                   4000 non-null   object
 10  default                              3698 non-null   object
 11  owns_car                             4000 non-null   object
 12  tenure                               3913 non-null   float64
dtypes: datetime64[ns](1), float64(1), int64(2), object(9)
memory usage: 406.4+ KB
```

In [57]:
```
df_custdemo.isnull().sum()
```

Out[57]:
```
customer_id                              0
first_name                               0
last_name                              125
gender                                   0
past_3_years_bike_related_purchases      0
DOB                                     87
job_title                              506
job_industry_category                  656
wealth_segment                           0
deceased_indicator                       0
default                                302
owns_car                                 0
tenure                                  87
dtype: int64
```

- Columns `last_name` , `DOB` , `job_title` , `job_industry_category` , `default` and `tenure` have missing values. They can either be treated or dropped based on the analysis

In [58]:
```
df_custdemo.customer_id.nunique()
```

Out[58]:
```
4000
```

In [59]:
```
df_custdemo.shape
```

Out[59]:
```
(4000, 13)
```

In [60]:
```
df_custdemo.duplicated().sum()
```

Out[60]:
```
0
```

- There are no duplicate values

```
In [61]: df_custdemo.gender.value_counts()
```

```
Out[61]: Female    2037
         Male      1872
         U           88
         F            1
         Femal        1
         M            1
         Name: gender, dtype: int64
```

- Values for gender haven't been captured properly or not cleaned

```
In [64]: df_custdemo.replace({'gender':{'M':'Male','Femal':'Female','F':'Female'}},
```

```
In [65]: df_custdemo.gender.value_counts()
```

```
Out[65]: Female    2039
         Male      1873
         U           88
         Name: gender, dtype: int64
```

- There are still 88 Unknown values

```
In [66]: df_custdemo.default.value_counts()
```

```
Out[66]: 100                                         113
         1                                           112
         -1                                          111
         -100                                         99
         Ù¡Ù¢Ù£                                        53
                                                    ...
         testâ testâ«                                 31
         /dev/null; touch /tmp/blns.fail ; echo       30
         âªâªtestâª                                   29
         ì¸ëëºí ë¥´                                     27
         ,ãã»:*:ã»ãâ( â» Ï â» )ãã»:*:ã»ãâ          25
         Name: default, Length: 90, dtype: int64
```

- Default data has to be re-collected or dropped from the DB

Customer address DB

```
In [67]: df_custaddr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   customer_id         3999 non-null   int64 
 1   address             3999 non-null   object
 2   postcode            3999 non-null   int64 
 3   state               3999 non-null   object
 4   country             3999 non-null   object
 5   property_valuation  3999 non-null   int64 
dtypes: int64(3), object(3)
memory usage: 187.6+ KB
```

In [70]:
```
df_custaddr.isnull().sum()
```

Out[70]:
```
customer_id           0
address               0
postcode              0
state                 0
country               0
property_valuation    0
dtype: int64
```

- No missing values

In [71]:
```
df_custaddr.duplicated().sum()
```

Out[71]:
```
0
```

In [72]:
```
df_custaddr.customer_id.nunique()
```

Out[72]:
```
3999
```

In [73]:
```
df_custaddr.shape
```

Out[73]:
```
(3999, 6)
```