Table Of Content :-

- 1- Import libraries
- 2- Read the dataset files
- 3- Merge Data files
- 4- Data Exploration
- 5- Data cleaning

1- import libraries:-

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

2- Read the dataset file:-

In [2]:
```
## Read data files as dataframes with pandas

drugs_df          = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\drugs.csv")
doctor_df         = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\doctor.csv")
patient_df        = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\patient.csv")
supplier_df       = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\supplier.csv")
insurance_df      = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\insurance.csv")
prescriptions_df = pd.read_csv(r"C:\Users\dell\Desktop\powerbi projects\datasets\pharamacy_data\perscriptions.csv")
```

In [3]:
```
# Check columns
doctor_df.columns
```

Out[3]:
```
Index(['physID', 'name', 'address', 'phone'], dtype='object')
```

In [4]:
```
# change column name from physID to doctorID
doctor_df = doctor_df.rename(columns={'physID':'doctorID'})
prescriptions_df = prescriptions_df.rename(columns={'physID':'doctorID'})

print('doctor_df       :  ',doctor_df.columns)
print('prescriptions_df :  ',prescriptions_df.columns)
```

```
doctor_df       :  Index(['doctorID', 'name', 'address', 'phone'], dtype='object')
prescriptions_df :  Index(['patientID', 'doctorID', 'NDC', 'qty', 'days', 'refills', "status"], dtype="object")
```

3- Merge Data Files into one Dataframe:-

- Merge ALL The DataFrames into one Super DataFrame called Data using Left outer join

In [5]:
```
# 1. Merge patient and prescriptions on 'patientID'
data = pd.merge(patient_df, prescriptions_df, on='patientID', how='left')

# 2. Merge with doctor information on 'doctorID'
data = pd.merge(data, doctor_df, on='doctorID', how='left', suffixes=('_patient', '_doctor'))

# 3. Merge with drug information on 'NDC'
data = pd.merge(data, drugs_df, on='NDC', how='left', suffixes=('_merged', '_drug'))

# 4. Merge with supplier information on 'supID'
data = pd.merge(data, supplier_df, on='supID', how='left', suffixes=('_drug', '_supplier'))

# 5. Merge with insurance information on 'insurance' (from patient) and 'name' (from insurance)
data = pd.merge(data, insurance_df, left_on='insurance', right_on='name', how='left', suffixes=('_merged', '_insurance'))

# Display the shape and size of the merged DataFrame

print(f"\nShape of the final merged DataFrame: {data.shape}")
```

```
Shape of the final merged DataFrame: (23, 30)
```

- Show a brief of the dataframe:-

In [44]:
```
# print top 5 Rows
data.head()
```

Out[44]:

	firstName	lastName	birthdate	address_patient	phone_patient	gender	insurance	patientID	doctorID	NDC	...	expDate	supID	purchasePrice	sellPrice	name_supplier	address	phone_merged	name	phone_insurance
0	James	Smith	01/01/1987	652 Jill Dr.	(868)456-9876	M	Molina	1	9.0	23567-0	...	09/22	1.0	11.23	12.55	Cardinal Health	7000 Cardinal Place, Dublin, OH 43017	(614)553-4460	Molina	(800)890-0909
1	James	Smith	01/01/1987	652 Jill Dr.	(868)456-9876	M	Molina	1	9.0	67876-0	...	09/23	1.0	6.77	7.89	Cardinal Health	7000 Cardinal Place, Dublin, OH 43017	(614)553-4460	Molina	(800)890-0909
2	Huda	Saleh	09/22/1999	347 Moss Rd.	(313)459-9226	F	Alliance	2	2.0	78965-0	...	05/23	1.0	5.45	6.78	Cardinal Health	7000 Cardinal Place, Dublin, OH 43017	(614)553-4460	Alliance	(800)657-9032
3	Huda	Saleh	09/22/1999	347 Moss Rd.	(313)459-9226	F	Alliance	2	2.0	23567-0	...	09/22	1.0	11.23	12.55	Cardinal Health	7000 Cardinal Place, Dublin, OH 43017	(614)553-4460	Alliance	(800)657-9032
4	Huda	Saleh	09/22/1999	347 Moss Rd.	(313)459-9226	F	Alliance	2	2.0	43234-0	...	12/22	2.0	33.43	40.33	McKesson	6555 Sate Hwy, Irving, TX 75039	(734)427-2000	Alliance	(800)657-9032

5 rows × 30 columns

In [51]:
```
# Last 5 Rows
data.tail()
```

Out[51]:

	firstName	lastName	birthdate	address_patient	phone_patient	gender	insurance	patientID	doctorID	NDC	...	expDate	supID	purchasePrice	sellPrice	name_supplier	address	phone_merged	name	phone_i
18	Fatema	Almo	08/06/2004	768 Castle Cir.	(313)712-0908	F	Molina	13	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Molina	(800
19	Avery	Brandon	02/14/1955	569 Forrest Ln.	(134)786-6654	NaN	PriorityHealth	14	5.0	17863-0	...	12/25	1.0	12.34	15.99	Cardinal Health	7000 Cardinal Place, Dublin, OH 43017	(614)553-4460	PriorityHealth	(800
20	Avery	Brandon	02/14/1955	569 Forrest Ln.	(134)786-6654	NaN	PriorityHealth	14	5.0	45652-0	...	04/21	2.0	2.34	4.33	McKesson	6555 Sate Hwy, Irving, TX 75039	(734)427-2000	PriorityHealth	(800
21	Jose	Martinez	01/19/1988	555 Morris Rd.	(976)821-0090	M	NaN	15	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22	Rose	Johns	09/05/2000	897 Mallory Dr.	(456)897-0908	F	Molina	16	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Molina	(800

5 rows × 30 columns

4- Data Exploration:-

In [6]:
```
print('Size  of data       :',data.size)
print('Shape of data       :',data.shape)
print('Dimension of data :',data.ndim)
```

```
Size  of data       : 690
Shape of data       : (23, 30)
Dimension of data : 2
```

In [43]:
```
# Get some informations about the data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23 entries, 0 to 22
Data columns (total 30 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   firstName       23 non-null     object
 1   lastName        23 non-null     object
 2   birthdate       23 non-null     object
 3   address_patient  23 non-null     object
 4   phone_patient   23 non-null     object
 5   gender          19 non-null     object
 6   insurance       20 non-null     object
 7   patientID       23 non-null     int64
 8   doctorID        15 non-null     float64
 9   NDC             15 non-null     float64
 10  qty             15 non-null     float64
 11  days            15 non-null     float64
 12  refills         15 non-null     float64
 13  status          11 non-null     object
 14  name_drug       15 non-null     object
 15  address_doctor  15 non-null     object
 16  phone_doctor    15 non-null     object
 17  brandName       15 non-null     object
 18  genericName     15 non-null     object
 19  dosage          15 non-null     float64
 20  expDate         15 non-null     object
 21  supID           15 non-null     float64
 22  purchasePrice   15 non-null     float64
 23  sellPrice       15 non-null     float64
 24  name_supplier   15 non-null     object
 25  address         15 non-null     object
 26  phone_merged    15 non-null     object
 27  name            20 non-null     object
 28  phone_insurance  20 non-null     object
 29  coPay           20 non-null     object
dtypes: float64(9), int64(1), object(20)
memory usage: 5.4+ KB
```

In [76]:
```
# Get Statistical Informations
data.describe()
```

Out[76]:

	patientID	doctorID	NDC	qty	days	refills	dosage	supID	purchasePrice	sellPrice
count	23.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000	15.000000
mean	7.304348	3.733333	41429.133333	36.000000	30.000000	2.533333	48.666667	1.266667	13.272000	15.647333
std	4.704300	3.011091	22423.737050	15.83396	13.887301	2.325838	28.564880	0.457738	10.379694	11.578245
min	1.000000	1.000000	12365.000000	15.000000	15.000000	0.000000	10.000000	1.000000	2.340000	4.330000
25%	3.500000	1.000000	23517.000000	30.000000	22.500000	0.000000	22.500000	1.000000	6.110000	7.335000
50%	7.000000	2.000000	34543.000000	30.000000	30.000000	3.000000	50.000000	1.000000	11.230000	12.550000
75%	10.500000	6.000000	56787.000000	45.000000	30.000000	5.000000	70.000000	1.500000	14.050000	16.935000
max	16.000000	9.000000	78987.000000	60.000000	60.000000	5.000000	100.000000	2.000000	35.670000	40.330000

Metadata : -

1- NDC (National Drug Code)

- Standard 10/11-digit ID for U.S. medications
- Identifies manufacturer, product, and package details

2- coPay (Co-payment)

- Fixed amount a patient pays for medication
- Determined by insurance after deductible
- Yes --> insurance cover , No --> insurance dosenot cover

3- refills

- Number of times a prescription can be refilled "0" means no refills allowed

4- subID

- supplierid

5- dosage

- dose of druge

In [75]:
```
data['coPay'].value_counts()
```

Out[75]:
```
No     17
Yes     3
Name: coPay, dtype: int64
```

In [79]:
```
data['NDC'].value_counts()
```

Out[79]:
```
23567-0    2
67876-0    2
78965-0    1
43234-0    1
34543-0    1
12365-0    1
34321-0    1
23467-0    1
23456-0    1
45698-0    1
78987-0    1
17863-0    1
45652-0    1
Name: NDC, dtype: int64
```

In [79]:
```
data['refills'].value_counts()
```

Out[79]:
```
0.0    6
5.0    6
3.0    2
2.0    1
Name: refills, dtype: int64
```

In [49]:
```
# Check Duplicates
data.duplicated().sum()
```

Out[49]:
```
0
```

In [67]:
```
# Check Missing Values
data.isna().sum()
```

Out[67]:
```
firstName          0
lastName           0
birthdate          0
address_patient    0
phone_patient      0
gender             4
insurance          3
patientID          0
doctorID           8
NDC                8
qty                8
days               8
refills            8
status            12
name_drug          8
address_doctor     8
phone_doctor       8
brandName          8
genericName        8
dosage             8
expDate            8
supID              8
purchasePrice      8
sellPrice          8
name_supplier      8
address            8
phone_merged       8
name               3
phone_insurance    3
coPay              3
dtype: int64
```

In [23]:
```
daictionary = {
    # numerical data
    'sellprice':data['sellPrice'].median(),
    'purchasePrice':data['purchasePrice'].median(),

    ## categorical data
    'gender'   :data['gender'].fillna('unkown'),
    'status'   :data['status'].fillna('unkown'),
    'refills'  :data['refills'].fillna('unkown'),
    'insurance':data['insurance'].fillna('unkown'),
    'name_drug':data['name_drug'].fillna('unkown'),
    'brandName':data['brandName'].fillna('unkown'),
    'address_doctor':data['address_doctor'].fillna('unkown'),
    'phone_doctor':data['phone_doctor'].fillna('unkown'),
    'coPay':data['coPay'].fillna('unkown')
}
```

In [28]:
```
# Use The Data Dictionary to fill missing Values And Save The Results
data.fillna(value= daictionary, inplace=True)
```

In [29]:
```
# Delete missing values
data.dropna(inplace=True)
```

- DONE