Importing pandas library

In [63]:
```
import pandas as pd
```

1. Export data into csv file and read into Dataframe.

In [64]:
```
df=pd.read_csv('C:\sqlite\student_travel.csv')
```

2. Display information of dataframe. Name the column who has missing value.

In [65]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   SID             20 non-null     int64 
 1   COURSE          18 non-null     object
 2   SNAME           20 non-null     object
 3   CITY            19 non-null     object
 4   TRAVEL_BY       20 non-null     object
 5   TRAVEL_EXPENSE  20 non-null     int64 
dtypes: int64(2), object(4)
memory usage: 1.1+ KB
```

3. Replace missing value by mean of that column.

In [66]:
```
df.fillna(df.mean())
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_12896\634187881.py:1: FutureWarning: Dropping of
nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a f
uture version this will raise TypeError.  Select only valid columns before calling the re
duction.
  df.fillna(df.mean())
```

Out[66]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
0	1	BCA	LAXMAN	SURAT	BIKE	50
1	2	BCA	XMAN	BARDOLI	BUS	25
2	3	BCA	AXMAN	VESU	CAR	100
3	4	BCA	LAMAN	SURAT	CAR	100
4	5	BCA	AMAN	BARDOLI	BUS	25
5	6	BCA	CHMAN	VESU	CAR	100
6	7	BCA	MAN	BARDOLI	BUS	20
7	8	BCA	ABC	VESU	CAR	100
8	9	BCA	XYZ	SURAT	BUS	20
9	10	BCA	MNO	BARDOLI	BIKE	50
10	11	BCA	CDN	VESU	BIKE	50
11	12	BBA	SRM	BARDOLI	BUS	20
12	13	NaN	OM	BARDOLI	BIKE	50
13	14	BBA	SAI	SURAT	BUS	20
14	15	BCA	RAMA	VESU	CAR	100
15	16	BCA	AYUSH	BARDOLI	BUS	40
16	17	BCA	MEET	NaN	BIKE	10
17	18	NaN	DEV	SURAT	BUS	500
18	19	BBA	SMIT	BARDOLI	CAR	70
19	20	BCA	ANZAR	BARDOLI	BUS	30

4. Print first two records only.

In [67]:
```
df.head(2)
```

Out[67]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
0	1	BCA	LAXMAN	SURAT	BIKE	50
1	2	BCA	XMAN	BARDOLI	BUS	25

5. Print all record except last 5 records.

In [68]:
```
df.head(-5)
```

Out[68]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
0	1	BCA	LAXMAN	SURAT	BIKE	50
1	2	BCA	XMAN	BARDOLI	BUS	25
2	3	BCA	AXMAN	VESU	CAR	100
3	4	BCA	LAMAN	SURAT	CAR	100
4	5	BCA	AMAN	BARDOLI	BUS	25
5	6	BCA	CHMAN	VESU	CAR	100
6	7	BCA	MAN	BARDOLI	BUS	20
7	8	BCA	ABC	VESU	CAR	100
8	9	BCA	XYZ	SURAT	BUS	20
9	10	BCA	MNO	BARDOLI	BIKE	50
10	11	BCA	CDN	VESU	BIKE	50
11	12	BBA	SRM	BARDOLI	BUS	20
12	13	NaN	OM	BARDOLI	BIKE	50
13	14	BBA	SAI	SURAT	BUS	20
14	15	BCA	RAMA	VESU	CAR	100

6. Print all record except first 10 records.

In [69]:
```
df.tail(-10)
```

Out[69]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
10	11	BCA	CDN	VESU	BIKE	50
11	12	BBA	SRM	BARDOLI	BUS	20
12	13	NaN	OM	BARDOLI	BIKE	50
13	14	BBA	SAI	SURAT	BUS	20
14	15	BCA	RAMA	VESU	CAR	100
15	16	BCA	AYUSH	BARDOLI	BUS	40
16	17	BCA	MEET	NaN	BIKE	10
17	18	NaN	DEV	SURAT	BUS	500
18	19	BBA	SMIT	BARDOLI	CAR	70
19	20	BCA	ANZAR	BARDOLI	BUS	30

7. Group student based on city name.

In [78]:
```
df.groupby(['CITY','SNAME']).sum()
```

Out[78]:

CITY	SNAME	SID	TRAVEL_EXPENSE
BARDOLI	AMAN	5	25
	ANZAR	20	30
	AYUSH	16	40
	MAN	7	20
	MNO	10	50
	OM	13	50
	SMIT	19	70
	SRM	12	20
	XMAN	2	25
SURAT	DEV	18	500
	LAMAN	4	100
	LAXMAN	1	50
	SAI	14	20
	XYZ	9	20
VESU	ABC	8	100
	AXMAN	3	100
	CDN	11	50
	CHMAN	6	100
	RAMA	15	100

8. Sort records based on Travel by column.

In [79]:
```
df.sort_values(by='TRAVEL_BY')
```

Out[79]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
0	1	BCA	LAXMAN	SURAT	BIKE	50
16	17	BCA	MEET	NaN	BIKE	10
12	13	NaN	OM	BARDOLI	BIKE	50
10	11	BCA	CDN	VESU	BIKE	50
9	10	BCA	MNO	BARDOLI	BIKE	50
6	7	BCA	MAN	BARDOLI	BUS	20
8	9	BCA	XYZ	SURAT	BUS	20
11	12	BBA	SRM	BARDOLI	BUS	20
13	14	BBA	SAI	SURAT	BUS	20
15	16	BCA	AYUSH	BARDOLI	BUS	40
1	2	BCA	XMAN	BARDOLI	BUS	25
17	18	NaN	DEV	SURAT	BUS	500
4	5	BCA	AMAN	BARDOLI	BUS	25
19	20	BCA	ANZAR	BARDOLI	BUS	30
5	6	BCA	CHMAN	VESU	CAR	100
7	8	BCA	ABC	VESU	CAR	100
18	19	BBA	SMIT	BARDOLI	CAR	70
3	4	BCA	LAMAN	SURAT	CAR	100
2	3	BCA	AXMAN	VESU	CAR	100
14	15	BCA	RAMA	VESU	CAR	100

9. Display those records who travel from “Bardoli” in “Bike”

In [80]:
```
df[(df['CITY']=='BARDOLI') & (df['TRAVEL_BY']=='BIKE')]
```

Out[80]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
9	10	BCA	MNO	BARDOLI	BIKE	50
12	13	NaN	OM	BARDOLI	BIKE	50

10. How many students travel by bus, bike and car?

In [89]:
```
df.groupby(['TRAVEL_BY']).size().reset_index(name='Number of Student')
```

Out[89]:

	TRAVEL_BY	Number of Student
0	BIKE	5
1	BUS	9
2	CAR	6

11. Display only city name. [don’t repeat city name].

In [96]:
```
df['CITY'].unique()
```

Out[96]:
```
array(['SURAT', 'BARDOLI', 'VESU', nan], dtype=object)
```

12. Display SNAME and TRAVEL BY columns data only.

In [98]:
```
df[['SNAME','TRAVEL_BY']]
```

Out[98]:

	SNAME	TRAVEL_BY
0	LAXMAN	BIKE
1	XMAN	BUS
2	AXMAN	CAR
3	LAMAN	CAR
4	AMAN	BUS
5	CHMAN	CAR
6	MAN	BUS
7	ABC	CAR
8	XYZ	BUS
9	MNO	BIKE
10	CDN	BIKE
11	SRM	BUS
12	OM	BIKE
13	SAI	BUS
14	RAMA	CAR
15	AYUSH	BUS
16	MEET	BIKE
17	DEV	BUS
18	SMIT	CAR
19	ANZAR	BUS

13. Display those record where students are from “BBA” coming in “Car” and “Expense < 500”.

In [99]:
```
df[(df['COURSE']=='BBA') & (df['TRAVEL_BY']=='CAR') & (df['TRAVEL_EXPENSE']<500)]
```

Out[99]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
18	19	BBA	SMIT	BARDOLI	CAR	70

14. Display records from 5 to 9.

In [104]:
```
df[4:9]
```

Out[104]:

	SID	COURSE	SNAME	CITY	TRAVEL_BY	TRAVEL_EXPENSE
4	5	BCA	AMAN	BARDOLI	BUS	25
5	6	BCA	CHMAN	VESU	CAR	100
6	7	BCA	MAN	BARDOLI	BUS	20
7	8	BCA	ABC	VESU	CAR	100
8	9	BCA	XYZ	SURAT	BUS	20

15. Display student name and city of “BCA” department from records 11 to 15.

In [122]:
```
df1=df.iloc[10:15,0:]
df1[['SNAME','CITY']] [df1['COURSE'] == 'BCA']
```

Out[122]:

	SNAME	CITY
10	CDN	VESU
14	RAMA	VESU