

In [3]: `import pandas as pd`

In [5]: `# reading dataset
SELECT * FROM DATASET_1;
df=pd.read_csv(r'C:\Users\pandu\OneDrive\Documents\Desktop\NIT\datasets\dataset_
df.head()`

Out[5]:

	destination	passanger	weather	temperature	time	coupon	expiration	ge
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Fe
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Fe
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Fe
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Fe
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Fe

5 rows × 27 columns



In [7]: `# read 2 attributes from dataset
select weather, temperature from dataset_1;
df[['weather', 'temperature']]`

Out[7]:

	weather	temperature
0	Sunny	55
1	Sunny	80
2	Sunny	80
3	Sunny	80
4	Sunny	80
...
12679	Rainy	55
12680	Rainy	55
12681	Snowy	30
12682	Snowy	30
12683	Sunny	80

12684 rows × 2 columns

In [9]: `# extract top 10 records
select * from dataset_1 limit 10;`

```
df.head(10)
```

Out[9]:

	destination	passanger	weather	temperature	time	coupon	expiration	ge
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Fe
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Fe
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Fe
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Fe
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Fe
5	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Fe
6	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Fe
7	No Urgent Place	Kid(s)	Sunny	80	10AM	Restaurant(<20)	2h	Fe
8	No Urgent Place	Kid(s)	Sunny	80	10AM	Carry out & Take away	2h	Fe
9	No Urgent Place	Kid(s)	Sunny	80	10AM	Bar	1d	Fe

10 rows × 27 columns



```
In [13]: # extract unique values from passanger column
# select distinct(passanger) from dataset_1;
df['passanger'].unique()
```

```
Out[13]: array(['Alone', 'Friend(s)', 'Kid(s)', 'Partner'], dtype=object)
```

```
In [15]: # extract the records whose destination is home
# select * from dataset_1 where destination='Home';
df[df['destination']=='Home']
```

Out[15]:

	destination	passanger	weather	temperature	time	coupon	expiration
13	Home	Alone	Sunny	55	6PM	Bar	1c
14	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1c
15	Home	Alone	Sunny	80	6PM	Coffee House	2h
35	Home	Alone	Sunny	55	6PM	Bar	1c
36	Home	Alone	Sunny	55	6PM	Restaurant(20-50)	1c
...
12675	Home	Alone	Snowy	30	10PM	Coffee House	2h
12676	Home	Alone	Sunny	80	6PM	Restaurant(20-50)	1c
12677	Home	Partner	Sunny	30	6PM	Restaurant(<20)	1c
12678	Home	Partner	Sunny	30	10PM	Restaurant(<20)	2h
12679	Home	Partner	Rainy	55	6PM	Carry out & Take away	1c

3237 rows × 27 columns



In [20]:

```
# sorting the data ascending or descending order
# select * from dataset_1 order by coupon; --ascending
df.sort_values('coupon')
```

Out[20]:

	destination	passanger	weather	temperature	time	coupon	expiration
11702	Home	Partner	Sunny	30	10PM	Bar	2h
9930	No Urgent Place	Alone	Snowy	30	2PM	Bar	1c
10632	Home	Alone	Rainy	55	6PM	Bar	1c
7997	No Urgent Place	Friend(s)	Rainy	55	10PM	Bar	2h
11166	Work	Alone	Snowy	30	7AM	Bar	1c
...
10476	Home	Alone	Sunny	80	6PM	Restaurant(<20)	1c
5447	Home	Alone	Sunny	80	10PM	Restaurant(<20)	2h
10478	Home	Alone	Snowy	30	10PM	Restaurant(<20)	2h
5440	No Urgent Place	Alone	Sunny	80	2PM	Restaurant(<20)	2h
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1c

12684 rows × 27 columns



In [22]: `#select * from dataset_1 order by coupon desc; --descending`
`df.sort_values(('coupon'),ascending=False)`

Out[22]:

	destination	passanger	weather	temperature	time	coupon	expiration
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1c
2503	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h
7480	Work	Alone	Sunny	80	7AM	Restaurant(<20)	1c
7478	Work	Alone	Snowy	30	7AM	Restaurant(<20)	2h
7477	Work	Alone	Rainy	55	7AM	Restaurant(<20)	2h
...
6666	Work	Alone	Rainy	55	7AM	Bar	1c
804	Work	Alone	Sunny	55	7AM	Bar	1c
9964	Work	Alone	Rainy	55	7AM	Bar	1c
6676	Home	Alone	Sunny	55	6PM	Bar	2h
10993	No Urgent Place	Friend(s)	Sunny	80	10PM	Bar	1c

12684 rows × 27 columns



```
In [34]: # rename destination to Destination
# select destination as Destination from dataset_1;
df.rename(columns={'destination': 'Destination'}, inplace=True)
print(df['Destination'])
```

```
0      No Urgent Place
1      No Urgent Place
2      No Urgent Place
3      No Urgent Place
4      No Urgent Place
...
12679      Home
12680      Work
12681      Work
12682      Work
12683      Work
Name: Destination, Length: 12684, dtype: object
```

```
In [38]: df.head()
```

Out[38]:

	Destination	passanger	weather	temperature	time	coupon	expiration	ge
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Fe
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Fe
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Fe
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Fe
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Fe

5 rows × 27 columns



In [52]:

```
# calculate total count by accupation
# select occupation,count(occupation) as count from dataset_1 group by occupation
df.groupby('occupation')['occupation'].count().to_frame('count').reset_index()
```

Out[52]:

	occupation	count
0	Architecture & Engineering	175
1	Arts Design Entertainment Sports & Media	629
2	Building & Grounds Cleaning & Maintenance	44
3	Business & Financial	544
4	Community & Social Services	241
5	Computer & Mathematical	1408
6	Construction & Extraction	154
7	Education&Training&Library	943
8	Farming Fishing & Forestry	43
9	Food Preparation & Serving Related	298
10	Healthcare Practitioners & Technical	244
11	Healthcare Support	242
12	Installation Maintenance & Repair	133
13	Legal	219
14	Life Physical Social Science	170
15	Management	838
16	Office & Administrative Support	639
17	Personal Care & Service	175
18	Production Occupations	110
19	Protective Service	175
20	Retired	495
21	Sales & Related	1093
22	Student	1584
23	Transportation & Material Moving	218
24	Unemployed	1870

In [54]: *# calculate average temperature by weather*
select weather, avg(temperature) as avg_temp from dataset_1 group by weather;
`df.groupby('weather')['temperature'].mean().to_frame('avg_temp').reset_index()`

Out[54]:

	weather	avg_temp
0	Rainy	55.000000
1	Snowy	30.000000
2	Sunny	68.946271

```
In [56]: # calculate total count of temperature by weather
# select weather,count(temperature) as count_temp from dataset_1 group by weather
df.groupby('weather')['temperature'].count().to_frame('count_temp').reset_index()
```

Out[56]:

	weather	count_temp
0	Rainy	1210
1	Snowy	1405
2	Sunny	10069

```
In [58]: # calculate total unique count of temperature by weather
# select weather ,count(distinct temperature) AS count_distinct_temp from dataset_1
df.groupby('weather')['temperature'].nunique().to_frame('count_unique_temp').reset_index()
```

Out[58]:

	weather	count_unique_temp
0	Rainy	1
1	Snowy	1
2	Sunny	3

```
In [60]: # calculate sum of temperature by weather
# select weather,sum(temperature) as sum_temp from dataset_1 group by weather;
df.groupby('weather')['temperature'].sum().to_frame('sum_temp').reset_index()
```

Out[60]:

	weather	sum_temp
0	Rainy	66550
1	Snowy	42150
2	Sunny	694220

```
In [62]: # calculate minimum temperature by weather
# select weather,min(temperature) as min_temp from dataset_1 group by weather;
df.groupby('weather')['temperature'].min().to_frame('min_temp').reset_index()
```

Out[62]:

	weather	min_temp
0	Rainy	55
1	Snowy	30
2	Sunny	30

```
In [78]: # calculate maximum temperature by weather
# select weather,max(temperature) as max_temp from dataset_1 group by weather;
df.groupby('weather')['temperature'].max().to_frame('max_temp').reset_index()
```

Out[78]:

	weather	max_temp
0	Rainy	55
1	Snowy	30
2	Sunny	80

```
In [88]: # count the occupation of student by its occupation
# select occupation,count(occupation) as count from dataset_1 group by occupation
result = df.groupby('occupation')['occupation'].count().to_frame('count').reset_
result[result['occupation'] == 'Student']
```

Out[88]:

	occupation	count
22	Student	1584

```
In [114... # read table_to_union dataset
df1=pd.read_csv(r'C:\Users\pandu\OneDrive\Documents\Desktop\NIT\datasets\table_t
df1
```

Out[114...

	destination	passanger	weather	temperature	time	coupon	expiration	ger
0	UNION	UNION	UNION	55	2PM	Restaurant(<20)	1d	Fer

1 rows × 27 columns



```
In [116... df1.rename(columns={'destination':'Destination'},inplace=True)
```

```
In [118... # extract distinct destination from 2 datasets
# select distinct destination from (select * from dataset_1 union select * from
pd.concat([df,df1])['Destination'].drop_duplicates()
```

Out[118...

0	No Urgent Place
13	Home
16	Work
0	UNION

Name: Destination, dtype: object

```
In [120... # read table_to_join dataset
df2=pd.read_csv(r'C:\Users\pandu\OneDrive\Documents\Desktop\NIT\datasets\table_t
df2
```

Out[120...

	time	part_of_day
0	2PM	Afternoon
1	10AM	Morning
2	6PM	Evening
3	7AM	Morning
4	10PM	Night

In [124... `# extract destination,time,part_of_day from 2 datasets`
`# select a.destination,a.time,b.part_of_day from dataset_1 as a inner join table`
`pd.merge(df,df2[['time','part_of_day']],on='time',how='inner')[['Destination','t`

Out[124...

	Destination	time	part_of_day
0	No Urgent Place	2PM	Afternoon
1	No Urgent Place	10AM	Morning
2	No Urgent Place	10AM	Morning
3	No Urgent Place	2PM	Afternoon
4	No Urgent Place	2PM	Afternoon
...
12679	Home	6PM	Evening
12680	Work	7AM	Morning
12681	Work	7AM	Morning
12682	Work	7AM	Morning
12683	Work	7AM	Morning

12684 rows × 3 columns

In [128... `# extract alone passangers with destination`
`# select destination,passanger from (select * from dataset_1 where passanger='Al`
`df[df['passanger']=='Alone'][['Destination','passanger']]`

Out[128...

	Destination	passanger
0	No Urgent Place	Alone
13	Home	Alone
14	Home	Alone
15	Home	Alone
16	Work	Alone
...
12676	Home	Alone
12680	Work	Alone
12681	Work	Alone
12682	Work	Alone
12683	Work	Alone

7305 rows × 2 columns

In [134... `# extract weather which starts with sun`
`# select * from dataset_1 where weather like 'sun%';`

```
df[df['weather'].str.startswith('Sun')]
```

Out[134...

	Destination	passanger	weather	temperature	time	coupon	expiration
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1c
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2l
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2l
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2l
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1c
...
12673	Home	Alone	Sunny	30	6PM	Carry out & Take away	1c
12676	Home	Alone	Sunny	80	6PM	Restaurant(20-50)	1c
12677	Home	Partner	Sunny	30	6PM	Restaurant(<20)	1c
12678	Home	Partner	Sunny	30	10PM	Restaurant(<20)	2l
12683	Work	Alone	Sunny	80	7AM	Restaurant(20-50)	2l

10069 rows × 27 columns



In [142...

```
# extract distinct temperature between 29 and 75
# select distinct temperature from dataset_1 where temperature between 29 and 75
df[(df['temperature']>=29) & (df['temperature']<=75)]['temperature'].unique()
```

Out[142...

```
array([55, 30], dtype=int64)
```

In [164...

```
# extract sales&related and management occupation from the data
# select occupation from dataset_1 where occupation in('Sales & Related','Manage
df[df['occupation'].isin(['Sales & Related','Management'])]['occupation'].head
```

Out[164...

occupation**193** Sales & Related**194** Sales & Related**195** Sales & Related**196** Sales & Related**197** Sales & Related**198** Sales & Related**199** Sales & Related**200** Sales & Related**201** Sales & Related**202** Sales & Related**203** Sales & Related**204** Sales & Related**205** Sales & Related**206** Sales & Related**207** Sales & Related**208** Sales & Related**209** Sales & Related**210** Sales & Related**211** Sales & Related**212** Sales & Related**213** Sales & Related**214** Sales & Related**215** Management**216** Management**217** Management**218** Management**219** Management**220** Management**221** Management**222** Management

In []: