In [1]:

```
# press TAB button while typing for suggestions
# import required libraries
import pandas as pd
import numpy as np
```

In [2]:

```
# check for available dataset in our system
import os
print(os.listdir())
```

```
['.ipynb_checkpoints', 'alexaAI ChatBot.ipynb', 'AttentionLayer.py', 'banklo
an2.csv', 'bitwiseNumber.ipynb', 'chatbot.py', 'datacleaning.csv', 'DataClea
ningYTube.csv', 'DataCleaningYTube.ipynb', 'dataset1.csv', 'DataSetAggregati
onP3.ipynb', 'datetime.ipynb', 'dic.pkl', 'employees.csv', 'example_chat.JP
G', 'frequancy count.ipynb', 'houseprice.csv', 'inv.pkl', 'LICENSE', 'Linear
RegressionModel.ipynb', 'math.ipynb', 'model_pickle', 'P2DataCleaning.ipyn
b', 'practical 4 of AI.pdf', 'random.ipynb', 'README.md', 'Salaries.csv', 'S
D.txt', 'seq2seq-chatbot-keras-with-attention.ipynb', 'Untitled.ipynb', 'Unt
itled1.ipynb']
```

In [3]:

```
# store dataset in our dataframe
df = pd.read_csv('employees.csv')
```

In [4]:

```
# print and verify dataset
df.head()
```

Out[4]:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	.
0	198	Donald	OConnell	DOCONNEL	650.507.9833	21-JUN-07	SH_
1	199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-08	SH_
2	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD
3	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	M
4	202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	M

In [5]:

```
# check for null or missing values
df.isnull().sum()
```

Out[5]:

```
EMPLOYEE_ID       0
FIRST_NAME        0
LAST_NAME         0
EMAIL             0
PHONE_NUMBER      0
HIRE_DATE         0
JOB_ID            0
SALARY            0
COMMISSION_PCT    0
MANAGER_ID        0
DEPARTMENT_ID     0
dtype: int64
```

In [6]:

```
# drop unnecessary columns
df = df.drop(columns=['COMMISSION_PCT'],axis=1)
```

In [7]:

```
# confirm drop columns are not present
df.head()
```

Out[7]:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	.
0	198	Donald	OConnell	DOCONNEL	650.507.9833	21-JUN-07	SH_
1	199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-08	SH_
2	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD
3	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	M
4	202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	M

In [8]:

```
# Returns sum of total values
df.EMPLOYEE_ID.value_counts()
```

Out[8]:

```
198    1
128    1
118    1
119    1
120    1
121    1
122    1
123    1
124    1
125    1
126    1
127    1
129    1
199    1
130    1
131    1
132    1
133    1
134    1
135    1
136    1
137    1
138    1
139    1
117    1
116    1
115    1
114    1
200    1
201    1
202    1
203    1
204    1
205    1
206    1
100    1
101    1
102    1
103    1
104    1
105    1
106    1
107    1
108    1
109    1
110    1
111    1
112    1
113    1
140    1
Name: EMPLOYEE_ID, dtype: int64
```

In [9]:

```
# Total number of jobs counts
df.JOB_ID.value_counts()
```

Out[9]:

```
ST_CLERK      16
ST_MAN         5
PU_CLERK       5
FI_ACCOUNT     5
IT_PROG        5
AD_VP          2
SH_CLERK       2
AC_ACCOUNT     1
AD_ASST        1
AC_MGR         1
PR_REP         1
FI_MGR         1
HR_REP         1
PU_MAN         1
MK_REP         1
MK_MAN         1
AD_PRES        1
Name: JOB_ID, dtype: int64
```

In [10]:

```
# Displays columns which satisfies the given condition
df [df['SALARY'] > 23000 ].value_counts()
```

Out[10]:

```
EMPLOYEE_ID  FIRST_NAME  LAST_NAME  EMAIL  PHONE_NUMBER  HIRE_DATE  JOB_ID
SALARY  MANAGER_ID  DEPARTMENT_ID
100          Steven      King       SKING  515.123.4567  17-JUN-03  AD_PRES
24000    -          90               1
dtype: int64
```

In [11]:

```
# Feature engineering
# Let's create full name column from given first and last names
df['FULL NAME '] = df['FIRST_NAME'] +' ' + df['LAST_NAME']
```

In [12]:

```
#df.drop(columns=['FIRST_NAME','LAST_NAME'])
```

In [13]:

```
# Aggregation
df.aggregate(['sum','min','max','mean','median'])
```

```
<ipython-input-13-cfbb5596f758>:2: FutureWarning: ['FIRST_NAME', 'LAST_NAM
E', 'EMAIL', 'PHONE_NUMBER', 'HIRE_DATE', 'JOB_ID', 'MANAGER_ID', 'FULL NAME
'] did not aggregate successfully. If any error is raised this will raise in
a future version of pandas. Drop these columns/ops to avoid this warning.
  df.aggregate(['sum','min','max','mean','median'])
```

Out[13]:

	EMPLOYEE_ID	FIRST_NAME	
sum	6738.00	DonaldDouglasJenniferMichaelPatSusanHermannShe...	OConnellGrantWhalenH
min	100.00	Adam	
max	206.00	William	
mean	134.76	NaN	
median	124.50	NaN	

In [14]:

```
df.agg(['sum'])
```

Out[14]:

	EMPLOYEE_ID	FIRST_NAME	
sum	6738	DonaldDouglasJenniferMichaelPatSusanHermannShe...	OConnellGrantWhalenHarts

In [15]:

```
df.agg({'SALARY':['sum','min','max','median']})
```

Out[15]:

	SALARY
sum	309116.0
min	2100.0
max	24000.0
median	4600.0