In [1]:
```
# read the following data set
#https://archive.ics.uci.edu/ml/machine-learning-databases/adult/
# rename the columns as per the description from this file
#https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names
```

In [2]:
```
import numpy as np
import pandas as pd
from pandas import DataFrame, Series
import sqlite3 as db
```

In [5]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
col_list = ['age','workclass','fnlwgt','education','education-num','marital-status
            'relationship','race','sex','capital-gain','capital-loss','hours-per-we
adult = pd.read_csv(url,sep=",",delimiter=",",names=col_list,skipinitialspace=True
sqladb = adult.copy()
```

In [6]:
```
print(sqladb.columns)

import re
sqladb.columns = [re.sub("[-]", "_", col) for col in adult.columns]

print(sqladb.columns)
```

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'Label'],
      dtype='object')
Index(['age', 'workclass', 'fnlwgt', 'education', 'education_num',
       'marital_status', 'occupation', 'relationship', 'race', 'sex',
       'capital_gain', 'capital_loss', 'hours_per_week', 'native_country',
       'Label'],
      dtype='object')
```

In [7]:
```
print(sqladb.education.unique())
print(sqladb.workclass.unique())
print(sqladb.relationship.unique())
print(sqladb.sex.unique())
print(sqladb.marital_status.unique())
print(sqladb.race.unique())
```

```
['Bachelors' 'HS-grad' '11th' 'Masters' '9th' 'Some-college' 'Assoc-acdm'
 'Assoc-voc' '7th-8th' 'Doctorate' 'Prof-school' '5th-6th' '10th'
 '1st-4th' 'Preschool' '12th']
['State-gov' 'Self-emp-not-inc' 'Private' 'Federal-gov' 'Local-gov' '?'
 'Self-emp-inc' 'Without-pay' 'Never-worked']
['Not-in-family' 'Husband' 'Wife' 'Own-child' 'Unmarried' 'Other-relative']
['Male' 'Female']
['Never-married' 'Married-civ-spouse' 'Divorced' 'Married-spouse-absent'
 'Separated' 'Married-AF-spouse' 'Widowed']
['White' 'Black' 'Asian-Pac-Islander' 'Amer-Indian-Eskimo' 'Other']
```

In [8]:
```
sqladb.head()
```

Out[8]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	ra
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	Wh
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Wh
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	Wh
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Bla
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Bla

In [9]:
```
sqladb.shape
```

Out[9]:
```
(32561, 15)
```

In [10]:
```
sqladb.describe()
```

Out[10]:

	age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

In [11]:
```
from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
```

In [12]:
```
#  Select 10 records from the adult sqladb

pysqldf("SELECT * FROM sqladb LIMIT 10;")
```

Out[12]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	ra
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	Wh
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Wh
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	Wh
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Bla
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Bla
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	Wh
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Bla
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	Wh
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	Wh
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	Wh

In [14]:
```
# Show me the average hours per week of all men who are working in private sector

q = """ select sex,workclass,avg(hours_per_week) from sqladb where sex ='Male' and

pysqldf(q)
```

Out[14]:

	sex	workclass	avg(hours_per_week)
0	Male	Private	42.221226

In [15]:
```
# Show me the frequency table for education, occupation and relationship, separate

q = """ select education,count(education) as frequency from sqladb group by educat

pysqldf(q)
```

Out[15]:

	education	frequency
0	10th	933
1	11th	1175
2	12th	433
3	1st-4th	168
4	5th-6th	333
5	7th-8th	646
6	9th	514
7	Assoc-acdm	1067
8	Assoc-voc	1382
9	Bachelors	5355
10	Doctorate	413
11	HS-grad	10501
12	Masters	1723
13	Preschool	51
14	Prof-school	576
15	Some-college	7291

In [16]:
```
q = """ select occupation,count(occupation) as Frequency from sqladb group by occu

pysqldf(q)
```

Out[16]:

	occupation	Frequency
0	?	1843
1	Adm-clerical	3770
2	Armed-Forces	9
3	Craft-repair	4099
4	Exec-managerial	4066
5	Farming-fishing	994
6	Handlers-cleaners	1370
7	Machine-op-inspct	2002
8	Other-service	3295
9	Priv-house-serv	149
10	Prof-specialty	4140
11	Protective-serv	649
12	Sales	3650
13	Tech-support	928
14	Transport-moving	1597

In [17]:
```
q = """ select relationship,count(relationship) as Frequency from sqladb group by

pysqldf(q)
```

Out[17]:

	relationship	Frequency
0	Husband	13193
1	Not-in-family	8305
2	Other-relative	981
3	Own-child	5068
4	Unmarried	3446
5	Wife	1568

In [20]:
```
# Are there any people who are married, working in private sector and having a mas

q = """
 select count(*) as count_of_people from sqladb WHERE marital_status != 'Never-mar
"""
pysqldf(q)
```

Out[20]:

	count_of_people
0	660

In [21]:
```
# What is the average, minimum and maximum age group for people working in differe

q = """
    select avg(age),max(age),min(age),workclass from sqladb group by workclass ;
"""

pysqldf(q)
```

Out[21]:

	avg(age)	max(age)	min(age)	workclass
0	40.960240	90	17	?
1	42.590625	90	17	Federal-gov
2	41.751075	90	17	Local-gov
3	20.571429	30	17	Never-worked
4	36.797585	90	17	Private
5	46.017025	84	17	Self-emp-inc
6	44.969697	90	17	Self-emp-not-inc
7	39.436055	81	17	State-gov
8	47.785714	72	19	Without-pay

In [22]:
```
# Calculate age distribution by country

q = """
    select native_country, max(age),min(age),avg(age) from sqladb group by native_
"""

pysqldf(q)
```

Out[22]:

	native_country	max(age)	min(age)	avg(age)
0	?	90	17	38.725557
1	Cambodia	65	18	37.789474
2	Canada	80	17	42.545455
3	China	75	22	42.533333
4	Columbia	75	18	39.711864
5	Cuba	82	21	45.768421
6	Dominican-Republic	78	18	37.728571
7	Ecuador	90	21	36.642857
8	El-Salvador	79	17	34.132075
9	England	90	17	41.155556
10	France	64	20	38.965517
11	Germany	74	18	39.255474
12	Greece	65	22	46.206897
13	Guatemala	66	19	32.421875
14	Haiti	63	17	38.272727
15	Holand-Netherlands	32	32	32.000000
16	Honduras	58	18	33.846154
17	Hong	60	19	33.650000
18	Hungary	81	24	49.384615
19	India	61	17	38.090000
20	Iran	63	22	39.418605
21	Ireland	68	23	36.458333
22	Italy	77	19	46.424658
23	Jamaica	66	18	35.592593
24	Japan	61	19	38.241935
25	Laos	56	19	34.722222
26	Mexico	81	17	33.290824
27	Nicaragua	67	19	33.617647
28	Outlying-US(Guam-USVI-etc)	63	21	38.714286
29	Peru	69	17	35.258065

	native_country	max(age)	min(age)	avg(age)
30	Philippines	90	17	39.444444
31	Poland	85	17	43.116667
32	Portugal	78	19	40.297297
33	Puerto-Rico	90	17	40.508772
34	Scotland	62	18	40.416667
35	South	90	19	38.750000
36	Taiwan	61	20	33.823529
37	Thailand	55	19	34.944444
38	Trinadad&Tobago	61	17	41.315789
39	United-States	90	17	38.655674
40	Vietnam	73	19	34.059701
41	Yugoslavia	66	20	38.812500

In [23]:
```
# Compute a new column as 'Net-Capital-Gain' from the two columns 'capital-gain' a

q = """
    select (capital_gain - capital_loss) as Net_Capital_Gain from sqladb;
"""
pysqldf(q)
```

Out[23]:

	Net_Capital_Gain
0	2174
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	14084
9	5178
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	-2042
24	0
25	0
26	0
27	0
28	0
29	0
...	...

	Net_Capital_Gain
32531	0
32532	0
32533	0
32534	0
32535	0
32536	0
32537	0
32538	15020
32539	0
32540	0
32541	0
32542	0
32543	0
32544	0
32545	0
32546	0
32547	0
32548	1086
32549	0
32550	0
32551	0
32552	0
32553	0
32554	0
32555	0
32556	0
32557	0
32558	0
32559	0
32560	15024

32561 rows × 1 columns

In [ ]: