Data Science Masters :Assignment 12

```
Create a sql db from adult dataset and name it sqladb
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 [29]:
```
# Solution
import pandas as pd
from pandas import DataFrame, Series
import sqlite3 as db

df = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",header=None, names = ['age'
df = df.drop(['prob'],axis=1)
df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)
df
```

11	30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0
12	23	Private	122272	Bachelors	13	Never-married	Adm-clerical	Own-child	White	Female	0	0
13	32	Private	205019	Assoc-acdm	12	Never-married	Sales	Not-in-family	Black	Male	0	0
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0
15	34	Private	245487	7th-8th	4	Married-civ-spouse	Transport-moving	Husband	Amer-Indian-Eskimo	Male	0	0
16	25	Self-emp-	176756	HS-grad	9	Never-	Farming-	Own-child	White	Male	0	0

In [2]:
```
con = db.connect(':memory:')
cursor = con.cursor()
# Create a sql db from adult dataset and name it sqladb
df.to_sql("sqladb", con, if_exists='replace', index=False)
```

In [30]:
```
# 1. Select 10 records from the adult sqladb
cursor.execute("SELECT * FROM sqladb LIMIT 10;")
print(cursor.fetchall())
```

```
[(39, 'State-gov', 77516, 'Bachelors', 13, 'Never-married', 'Adm-clerical', 'Not-in-family', 'White', 'Male', 2174, 0,
40, 'United-States'), (50, 'Self-emp-not-inc', 83311, 'Bachelors', 13, 'Married-civ-spouse', 'Exec-managerial', 'Husban
d', 'White', 'Male', 0, 0, 13, 'United-States'), (38, 'Private', 215646, 'HS-grad', 9, 'Divorced', 'Handlers-cleaners',
'Not-in-family', 'White', 'Male', 0, 0, 40, 'United-States'), (53, 'Private', 234721, '11th', 7, 'Married-civ-spouse',
'Handlers-cleaners', 'Husband', 'Black', 'Male', 0, 0, 40, 'United-States'), (28, 'Private', 338409, 'Bachelors', 13,
'Married-civ-spouse', 'Prof-specialty', 'Wife', 'Black', 'Female', 0, 0, 40, 'Cuba'), (37, 'Private', 284582, 'Master
s', 14, 'Married-civ-spouse', 'Exec-managerial', 'Wife', 'White', 'Female', 0, 0, 40, 'United-States'), (49, 'Private',
160187, '9th', 5, 'Married-spouse-absent', 'Other-service', 'Not-in-family', 'Black', 'Female', 0, 0, 16, 'Jamaica'),
(52, 'Self-emp-not-inc', 209642, 'HS-grad', 9, 'Married-civ-spouse', 'Exec-managerial', 'Husband', 'White', 'Male', 0,
0, 45, 'United-States'), (31, 'Private', 45781, 'Masters', 14, 'Never-married', 'Prof-specialty', 'Not-in-family', 'Whi
te', 'Female', 14084, 0, 50, 'United-States'), (42, 'Private', 159449, 'Bachelors', 13, 'Married-civ-spouse', 'Exec-man
agerial', 'Husband', 'White', 'Male', 5178, 0, 40, 'United-States')]
```

In [31]:
```
# 2. Show me the average hours per week of all men who are working in private sector
cursor.execute("SELECT AVG (hoursperweek) FROM sqladb WHERE sex='Male' AND workclass='Private';")
print("Average hours per week of all men who are working in private sector: %.2f " % cursor.fetchone())
```

```
Average hours per week of all men who are working in private sector: 42.22
```

In [32]:
```
# 3. Show me the frequency table for education, occupation and relationship, separately
# Showing table for education...
cursor.execute("""SELECT education,COUNT(education) as freq_education FROM sqladb
GROUP BY education """)
results = cursor.fetchall()
res_df = pd.DataFrame(results,columns=['edcuation','freq_education'])
res_df
```

Out[32]:

	edcuation	freq_education
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 [33]:
```
# Showing table for occupation...
cursor.execute("""SELECT occupation,COUNT(occupation) as freq_occupation FROM sqladb
GROUP BY occupation """)
results = cursor.fetchall()
res_df = pd.DataFrame(results,columns=['occupation','freq_occupation'])
res_df
```

Out[33]:

	occupation	freq_occupation
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 [34]:
```
# Showing table for relationship...
cursor.execute("""SELECT relationship,COUNT(relationship) as freq_relationship FROM sqladb
GROUP BY relationship """)
results = cursor.fetchall()
res_df = pd.DataFrame(results,columns=['relationship','freq_relationship'])
res_df
```

Out[34]:

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

In [35]:
```
# 4. Are there any people who are married, working in private sector and having a masters degree
# Solution

cursor.execute("""SELECT COUNT(*) FROM sqladb WHERE workclass='Private' AND education='Masters' AND (maritalstatus ='Marr
print("Are there any people who are married, working in private sector and having a masters degree? - ")
count = cursor.fetchone()[0]
if(count == 0):
    print("No")
else:
    print("Yes,",count)
```

```
Are there any people who are married, working in private sector and having a masters degree? -
Yes, 540
```

In [36]:
```
# 5. What is the average, minimum and maximum age group for people working in different sectors
# Solution

cursor.execute("""SELECT round(AVG(age),2),MIN(age),MAX(age) FROM sqladb;""")
ageValues = cursor.fetchone()
print("Average Age Value:",ageValues[0])
print("Maximum Age Value:",ageValues[1])
print("Minimum Age Value:",ageValues[2])
```

```
Average Age Value: 38.58
Maximum Age Value: 17
Minimum Age Value: 90
```

In [37]:
```
#6. Calculate age distribution by country
# Solution
cursor.execute("""SELECT nativecountry,SUM(CASE WHEN age < 18 THEN 1 ELSE 0 END) AS [Under 18],
        SUM(CASE WHEN age BETWEEN 18 AND 24 THEN 1 ELSE 0 END) AS [18-35],
        SUM(CASE WHEN age BETWEEN 25 AND 34 THEN 1 ELSE 0 END) AS [36-50],
        SUM(CASE WHEN age > 50 THEN 1 ELSE 0 END) AS [Above 50]
 FROM sqladb
 WHERE nativecountry != "?"
 GROUP BY nativecountry
""")
results = cursor.fetchall()
res_df = pd.DataFrame(results,columns=['Country','Age under 18','Age b/w 18-35','Age b/w 36-50','Age above 50'])
res_df
```

Out[37]:

	Country	Age under 18	Age b/w 18-35	Age b/w 36-50	Age above 50
0	Cambodia	0	1	6	2
1	Canada	2	11	29	38
2	China	0	5	20	20
3	Columbia	0	6	21	13
4	Cuba	0	5	16	36
5	Dominican-Republic	0	14	18	13
6	Ecuador	0	4	12	2
7	El-Salvador	2	26	34	12
8	England	1	10	25	21
9	France	0	2	13	6
10	Germany	0	16	47	28

In [43]:
```
ne two columns 'capital-gain' and 'capital-loss'

n,educationnum,maritalstatus,occupation,capitalgain,capitalloss,(capitalgain-capitalloss) as NetCaptailGain,relationship,r

s','fnlwgt','education','educationnum','maritalstatus','occupation','capitalgain','capitalloss','netcaptialgain','relation
```

3]:

	age	workclass	fnlwgt	education	educationnum	maritalstatus	occupation	capitalgain	capitalloss	netcaptialgain	relationship	race	sex	ho
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	2174	0	2174	Not-in-family	White	Male	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	0	0	0	Husband	White	Male	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	0	0	0	Not-in-family	White	Male	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	0	0	0	Husband	Black	Male	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	0	0	0	Wife	Black	Female	
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	0	0	0	Wife	White	Female	
6	49	Private	160187	9th	5	Married-spouse-	Other-service	0	0	0	Not-in-family	Black	Female	