Read the following data set:

https://archive.ics.uci.edu/ml/machine-learning-databases/adult/ (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 (https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names)

In [1]:

```
import pandas as pd
import sqlite3 as db
```

In [2]:

```
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.dat
```

In [6]:

```
conn = db.connect("sqladb.db")
```

In [14]:

```
df.to_sql("adult", conn, if_exists="replace",index=False)
```

1. Select 10 records from the adult sqladb

In [21]:

```
sql1="""
SELECT *
FROM adult
limit 10
"""

pd.read_sql_query(sql1, conn)
```

Out[21]:

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

2. Show me the average hours per week of all men who are working in private sector

In [31]:

```
sql2="""
SELECT  avg(hours_per_week)
FROM adult
where trim(sex)="Male"
and trim(workclass)="Private"

"""

pd.read_sql_query(sql2, conn)
```

Out[31]:

	avg(hours_per_week)
0	42.221226

3. Show me the frequency table for education, occupation and relationship, separately

In [34]:

```
sql3="""
SELECT education,count(*)
FROM adult
group by education

"""

pd.read_sql_query(sql3, conn)
```

Out[34]:

	education	count(*)
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 [35]:

```
sql4="""
SELECT occupation,count(*)
FROM adult
group by occupation

"""

pd.read_sql_query(sql4, conn)
```

Out[35]:

	occupation	count(*)
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 [36]:

```
sql5="""
SELECT relationship,count(*)
FROM adult
group by relationship

"""

pd.read_sql_query(sql5, conn)
```

Out[36]:

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

4. Are there any people who are married, working in private sector and having a masters degree

In [44]:

```
sql6="""
SELECT count(*)
FROM adult
where trim(relationship) like '%Married%'
and trim(workclass) = 'Private'
and trim(education) = 'Masters'
"""

pd.read_sql_query(sql6, conn)
```

Out[44]:

	count(*)
0	53

5. What is the average, minimum and maximum age group for people working in different sectors

In [51]:

```
sql7="""
SELECT workclass,max(age) as MAX_Age,min(age) as MIN_Age,avg(age) as AVG_Age
FROM adult
group by(workclass)
"""

pd.read_sql_query(sql7, conn)
```

Out[51]:

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

In [53]:

```
sql8="""
SELECT occupation,max(age) as MAX_Age,min(age) as MIN_Age,avg(age) as AVG_Age
FROM adult
group by(occupation)
"""

pd.read_sql_query(sql8, conn)
```

Out[53]:

	occupation	MAX_Age	MIN_Age	AVG_Age
0	?	90	17	40.882800
1	Adm-clerical	90	17	36.964456
2	Armed-Forces	46	23	30.222222
3	Craft-repair	90	17	39.031471
4	Exec-managerial	90	17	42.169208
5	Farming-fishing	90	17	41.211268
6	Handlers-cleaners	90	17	32.165693
7	Machine-op-inspct	90	17	37.715285
8	Other-service	90	17	34.949621
9	Priv-house-serv	81	17	41.724832
10	Prof-specialty	90	17	40.517633
11	Protective-serv	90	17	38.953775
12	Sales	90	17	37.353973
13	Tech-support	73	17	37.022629
14	Transport-moving	90	17	40.197871

6. Calculate age distribution by country

In [57]:

```
sql9="""
SELECT native_country as Country,count(*) as Distribution
FROM adult
group by(native_country)
"""

pd.read_sql_query(sql9, conn)
```

...

7. Compute a new column as 'Net-Capital-Gain' from the two columns 'capital-gain' and 'capital-loss'

In [62]:

```
sql10="""
SELECT capital_gain,capital_loss,(capital_gain-capital_loss) as Net_Capital_Gain
FROM adult
"""

pd.read_sql_query(sql10, conn)
```

Out[62]:

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

	capital_gain	capital_loss	Net_Capital_Gain
28	0	0	0
29	0	0	0
...	...	...	...
32531	0	0	0
32532	0	0	0
32533	0	0	0
32534	0	0	0
32535	0	0	0
32536	0	0	0
32537	0	0	0
32538	15020	0	15020
32539	0	0	0
32540	0	0	0
32541	0	0	0
32542	0	0	0
32543	0	0	0
32544	0	0	0
32545	0	0	0
32546	0	0	0
32547	0	0	0
32548	1086	0	1086
32549	0	0	0
32550	0	0	0
32551	0	0	0
32552	0	0	0
32553	0	0	0
32554	0	0	0
32555	0	0	0
32556	0	0	0
32557	0	0	0
32558	0	0	0
32559	0	0	0
32560	15024	0	15024

32561 rows × 3 columns

In [ ]:

```

```