

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>
 Task:

```
In [1]: import numpy as np
import pandas as pd
import sqlite3
```

```
In [2]: adult_data = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult_data.head()")
```

```
Out[2]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

```
In [3]: adult_col_name = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult_data.head()')
adult_col_name=adult_col_name.iloc[91:106].index.tolist()
adult_col_name=adult_col_name[1:]+adult_col_name[0::-1]
adult_data.columns = adult_col_name
adult_data.head()
```

```
Out[3]:
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	car
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	

In [5]:

```
sqladb = sqlite3.connect("sqladb.db")
sqladb.execute('''
    CREATE TABLE IF NOT EXISTS ADULTS (
        AGE            INTEGER,
        WORKCLASS       VARCHAR(100),
        FNLWGT          INTEGER,
        EDUCATION        VARCHAR(100),
        EDUCATION_NUM    INTEGER,
        MARITAL_STATUS  VARCHAR(100),
        OCCUPATION       VARCHAR(100),
        RELATIONSHIP     VARCHAR(100),
        RACE             VARCHAR(100),
        SEX             VARCHAR(20),
        CAPITAL_GAIN     INTEGER,
        CAPITAL_LOSS     INTEGER,
        HOURS_PER_WEEK   INTEGER,
        NATIVE_COUNTRY   VARCHAR(100),
        GT50_OR_LT50K    VARCHAR(20))
```

Out[5]: <sqlite3.Cursor at 0x29406da0650>

In [6]:

```
sql_insert = "INSERT INTO ADULTS (
    AGE,
    WORKCLASS,
    FNLWGT,
    EDUCATION,
    EDUCATION_NUM,
    MARITAL_STATUS,
    OCCUPATION,
    RELATIONSHIP,
    RACE,
    SEX,
    CAPITAL_GAIN,
    CAPITAL_LOSS,
    HOURS_PER_WEEK,
    NATIVE_COUNTRY,
    GT50_OR_LT50K) values
    (%d, '%s', %d, '%s', %d, '%s', '%s', '%s', '%s', '%s', %d, %d, %d, '%s', '%s'

for index, row in adult_data.iterrows():
    sqladb.execute(sql_insert % (row['age'],
        row['workclass'],
        row['fnlwgt'],
        row['education'],
        row['education-num'],
        row['marital-status'],
        row['occupation'],
        row['relationship'],
        row['race'], row['sex'],
        row['capital-gain'],
        row['capital-loss'],
        row['hours-per-week'],
        row['native-country'],
        row['>50K, <=50K.']))
```

##1. Select 10 records from the adult sqladb

```
In [7]: sql_select="SELECT * FROM ADULTS LIMIT 10;"
conn=sqladb

result_adult_data=pd.read_sql_query(sql_select, conn)
```

```
Out[7]:
```

	AGE	WORKCLASS	FNLWGT	EDUCATION	EDUCATION_NUM	MARITAL_STATUS	OCCUPATION	RELATIONSHIP
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
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-in-family
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband

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

```
In [8]: sql_select="SELECT SEX, WORKCLASS, AVG(HOURS_PER_WEEK) FROM ADULTS WHERE SEX=' Male' and WORKCLASS=' Private'"
result_avg_hr_per_week=pd.read_sql_query(sql_select, conn)
```

```
Out[8]:
```

	SEX	WORKCLASS	AVG(HOURS_PER_WEEK)
0	Male	Private	42.221226

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

In [9]: `## Education`

```
sql_select="SELECT EDUCATION, COUNT(EDUCATION) FROM ADULTS GROUP BY EDUCATION;"
frequency_education=pd.read_sql_query(sql_select, conn)
```

Out[9]:

	EDUCATION	COUNT(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 [15]: *##Occupation*

```
sql_select="SELECT OCCUPATION, COUNT(OCCUPATION) FROM ADULTS GROUP BY EDUCATION;"
frequency_occupation=pd.read_sql_query(sql_select, conn)
```

Out[15]:

	OCCUPATION	COUNT(OCCUPATION)
0	Handlers-cleaners	933
1	Sales	1175
2	Protective-serv	433
3	Machine-op-inspct	168
4	Machine-op-inspct	333
5	Craft-repair	646
6	Other-service	514
7	Tech-support	1067
8	Sales	1382
9	Prof-specialty	5355
10	?	413
11	Exec-managerial	10501
12	Exec-managerial	1723
13	Other-service	51
14	Prof-specialty	576
15	Protective-serv	7291

In [10]:

```
sql_select="SELECT RELATIONSHIP, COUNT(RELATIONSHIP) FROM ADULTS GROUP BY RELATIONSHIP;"
frequency_relationship=pd.read_sql_query(sql_select, conn)
```

Out[10]:

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

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

In [11]:

```
sql_select = "SELECT MARITAL_STATUS, WORKCLASS, EDUCATION, COUNT(*) FROM ADULTS "
sql_select = sql_select + " WHERE MARITAL_STATUS like ' Married%' AND WORKCLASS = ' Private' "
sql_select = sql_select + " GROUP BY MARITAL_STATUS, WORKCLASS, EDUCATION;"

result_married_private_sector_masters=pd.read_sql_query(sql_select, conn)
result_married_private_sector_masters
```

Out[11]:

	MARITAL_STATUS	WORKCLASS	EDUCATION	COUNT(*)
0	Married-civ-spouse	Private	Masters	531
1	Married-spouse-absent	Private	Masters	9

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

```
In [12]: sql_select = "SELECT WORKCLASS, AVG(AGE) , MIN(AGE), MAX(AGE) FROM ADULTS GROUP BY WORKCLASS"
         people_diff_sector=pd.read_sql_query(sql_select, conn)
```

Out[12]:

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

##6. Calculate age distribution by country

```
In [13]: sql_select = "SELECT NATIVE_COUNTRY, AGE, COUNT(AGE) FROM ADULTS GROUP BY NATIVE_COUNTRY  
age_distribution_by_country=pd.read_sql_query(sql_select, conn)
```

```
Out[13]:
```

	NATIVE_COUNTRY	AGE	COUNT(AGE)
0	?	17	2
1	?	18	8
2	?	19	5
3	?	20	10
4	?	21	11
5	?	22	12
6	?	23	6
7	?	24	14
8	?	25	11
9	?	26	18
10	?	27	15
11	?	28	19
12	?	29	12
13	?	30	19
14	?	31	18
15	?	32	17
16	?	33	13
17	?	34	24
18	?	35	18
19	?	36	23
20	?	37	22
21	?	38	20
22	?	39	19
23	?	40	12
24	?	41	22
25	?	42	24
26	?	43	14
27	?	44	10
28	?	45	17
29	?	46	15
...
1251	Vietnam	37	2
1252	Vietnam	38	1
1253	Vietnam	40	1
1254	Vietnam	41	1
1255	Vietnam	43	2
1256	Vietnam	44	3
1257	Vietnam	45	3
1258	Vietnam	46	1

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

```
In [14]: sql_select = "SELECT ADULTS.*, (capital_gain + capital_loss) as 'Net-Capital-Gain' FROM  
Capital_Gain=
```

Out[14]:

	AGE	WORKCLASS	FNLWGT	EDUCATION	EDUCATION_NUM	MARITAL_STATUS	OCCUPATION	RELATIONSH
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-fami
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husbar
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fami
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husbar
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wi