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/)

In [7]:

```
import pandas as pd
from sqlalchemy import create_engine
```

In [3]:

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

Create an sqlalchemy engine using a sample from the data set

In [8]:

```
engine = create_engine('sqlite://', echo=False)
```

In [9]:

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

Write two basic update queries

In [33]:

```
sql1="""
update adult
set workclass='State_gov'
where trim(workclass)='State-gov'
"""

sql2="""
select * from
adult where trim(workclass)='State_gov'
limit 1
"""
engine.execute(sql1)
engine.execute(sql2).fetchall()
```

Out[33]:

```
[(39, 'State_gov', 77516, ' Bachelors', 13, ' Never-married', ' Adm-clerica
l', ' Not-in-family', ' White', ' Male', 2174, 0, 40, ' United-States', ' <=
50K')]
```

In [34]:

```
sql3="""
update adult
set relationship='Spouse'
where trim(relationship)='Husband'
"""

sql4="""
select * from
adult where trim(relationship)='Spouse'
limit 1
"""

engine.execute(sql3)
engine.execute(sql4).fetchall()
```

Out[34]:

```
[(50, ' Self-emp-not-inc', 83311, ' Bachelors', 13, ' Married-civ-spouse', '
Exec-managerial', 'Spouse', ' White', ' Male', 0, 0, 13, ' United-States', '
<=50K')]
```

Write two delete queries

In [35]:

```
sql5="""
delete from adult
where trim(relationship)='Spouse'
"""

sql6="""
select * from
adult where trim(relationship)='Spouse'
"""

engine.execute(sql5)
engine.execute(sql6).fetchall()
```

Out[35]:

```
[]
```

In [37]:

```
sql7="""
delete from adult
where age=50
"""

sql8="""
select * from
adult
where age=50
"""

engine.execute(sql7)
engine.execute(sql8).fetchall()
```

Out[37]:

```
[]
```

Write two filter queries

In [42]:

```
sql9="""
SELECT  *
FROM adult
where trim(sex)="Male"
and trim(workclass)="Private"
limit 10
"""

pd.read_sql_query(sql9, conn)
```

Out[42]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relati
0	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-i
1	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husb
2	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husb
3	37	Private	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husb
4	32	Private	205019	Assoc-acdm	12	Never-married	Sales	Not-i
5	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husb
6	34	Private	245487	7th-8th	4	Married-civ-spouse	Transport-moving	Husb
7	32	Private	186824	HS-grad	9	Never-married	Machine-op-inspct	Unma
8	38	Private	28887	11th	7	Married-civ-spouse	Sales	Husb
9	40	Private	193524	Doctorate	16	Married-civ-spouse	Prof-specialty	Husb

In [44]:

```
sql10="""
SELECT *
FROM adult
where trim(relationship) like '%Married%'
and trim(workclass) = 'Private'
and trim(education) = 'Masters'
limit 10
"""

pd.read_sql_query(sql10, conn)
```

Out[44]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relati
0	44	Private	128354	Masters	14	Divorced	Exec-managerial	Unma
1	47	Private	87490	Masters	14	Divorced	Exec-managerial	Unma
2	37	Private	175232	Masters	14	Divorced	Exec-managerial	Unma
3	48	Private	125421	Masters	14	Divorced	Exec-managerial	Unma
4	43	Private	37937	Masters	14	Divorced	Exec-managerial	Unma
5	37	Private	262409	Masters	14	Divorced	Exec-managerial	Unma
6	47	Private	169092	Masters	14	Divorced	Prof-specialty	Unma
7	49	Private	178749	Masters	14	Married-spouse-absent	Adm-clerical	Unma
8	45	Private	125489	Masters	14	Divorced	Prof-specialty	Unma
9	46	Private	182128	Masters	14	Divorced	Prof-specialty	Unma

Write two function queries

In [47]:

```
sql11="""
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(sql11, conn)
```

Out[47]:

	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	Without-pay	72	19	47.785714
8	State_gov	81	17	39.436055

In [48]:

```
sql12="""
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(sql12, conn)
```

Out[48]:

	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

In [ ]:

```

```