Problem Statement

Read the following data set:

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

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

from pandasql import sqldf


url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

### Create a sql db from adult dataset and name it sqladb
df = pd.read_csv(url , sep=',',header=None, index_col=None)
df
```

Out[1]:

	0	1	2	3	4	5	6	7	8	9	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	
5	37	Private	284582	Masters	14	Married-civ-	Exec-	Wife	White	Female	

1. Create an sqlalchemy engine using a sample from the data set

In [2]:
```
df=df.head(10) # Taking the sample DB of size 10
df
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40

Adding columns to the dataframe

In [3]:
```
# Adding columns to the dataframe
df.columns = ['age', 'workclass', 'fnlwgt', 'education', 'education-num',
    'marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain'
    'capital-loss', 'hours-per-week', 'native-country', 'income']
df
```

Out[3]:

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

Data Preprocessing

In [4]:
```
# Triming all the strings to remove whitespaces from R.H.S. and L.H.S.
def trimAllColumns(df):
    trimStrings = lambda x : x.strip() if type(x) is str else x
    return df.applymap(trimStrings)
df = trimAllColumns(df)

# Replace all the columns having hyphen ("-")  to underscore ("_") as "-" columns
df.columns = df.columns.str.replace("-","_")
df
```

Out[4]:

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

1. Create an sqlalchemy engine using a sample from the data set

In [5]:
```
import sqlalchemy
import sqlite3
from sqlalchemy import create_engine
engine = create_engine('sqlite:///sqladb', echo=False) # Setting Echo to false to
df.to_sql('sqladb', engine, if_exists ='replace') #copying data from dataframe to
connection = sqlite3.connect("sqladb")  #creating connection to sqldb
cursor = connection.cursor() # creating curser
```

In [6]:
```
connection
```

Out[6]:
```
<sqlite3.Connection at 0x28c74c232d0>
```

In [7]:
```
cursor
```

Out[7]:
```
<sqlite3.Cursor at 0x28c74bdc1f0>
```

2. Write two basic update queries

In [8]:
```
sql = "update sqladb set education = 'Bachelors123' where education = 'Bachelors'
cursor.execute(sql)
connection.commit()
pd.read_sql_query("SELECT * FROM sqladb", connection)
```

Out[8]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Never-married	Adm-clerical	Not-in-
1	1	50	Self-emp-not-inc	83311	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu
2	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
3	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
4	4	28	Private	338409	Bachelors123	13	Married-civ-spouse	Prof-specialty	
5	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
6	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
7	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu
8	8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-
9	9	42	Private	159449	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu

In [9]:
```
sql = "update sqladb set marital_status = 'Unmarried' where marital_status = 'Neve
cursor.execute(sql)
connection.commit()
pd.read_sql_query("SELECT * FROM sqladb", connection)
```

Out[9]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	1	50	Self-emp-not-inc	83311	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu
2	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
3	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
4	4	28	Private	338409	Bachelors123	13	Married-civ-spouse	Prof-specialty	
5	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
6	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
7	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu
8	8	31	Private	45781	Masters	14	Unmarried	Prof-specialty	Not-in-
9	9	42	Private	159449	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu

3. Write two delete queries

In [10]:
```
sql = "delete from sqladb where marital_status = 'Unmarried' and education ='Maste
cursor.execute(sql)
connection.commit()
pd.read_sql_query("SELECT * FROM sqladb", connection)
```

Out[10]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	1	50	Self-emp-not-inc	83311	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu
2	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
3	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
4	4	28	Private	338409	Bachelors123	13	Married-civ-spouse	Prof-specialty	
5	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
6	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
7	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu
8	9	42	Private	159449	Bachelors123	13	Married-civ-spouse	Exec-managerial	Hu

In [11]:
```
sql = "delete from sqladb where marital_status = 'Married-civ-spouse' and educati
cursor.execute(sql)
connection.commit()
pd.read_sql_query("SELECT * FROM sqladb", connection)
```

Out[11]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
2	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
3	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
4	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
5	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu

4. Write two filter queries

In [12]:
```
pd.read_sql_query("SELECT * FROM sqladb where relationship ='Not-in-family' and o

#selecting 2 rows
```

Out[12]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-

In [13]:
```
pd.read_sql_query("SELECT * FROM sqladb where workclass ='Private' and race in ('
#selecting 4 rows
```

Out[13]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationsl
0	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-fan
1	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husba
2	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	W
3	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-fan

5. Write two function queries

In [14]:
```
pd.read_sql_query("SELECT * FROM sqladb   ", connection)
```

Out[14]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-
2	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
3	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
4	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
5	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu

In [15]:
```
def new_update(new_data):
    query_string = "update sqladb set marital_status=? where marital_status='Divo
    cursor.execute(query_string, new_data)
    #cursor.close()
    connection.commit()

new_update(['Saperated'])

pd.read_sql_query("SELECT * FROM sqladb", connection)
```

Out[15]:

	index	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relatio
0	0	39	State-gov	77516	Bachelors123	13	Unmarried	Adm-clerical	Not-in-
1	2	38	Private	215646	HS-grad	9	Saperated	Handlers-cleaners	Not-in-
2	3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Hu
3	5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	
4	6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-
5	7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Hu

In [16]:
```
def max_age():
    query_string = "select max(age) from sqladb "
    cursor.execute(query_string)
    #cursor.close()
    connection.commit()
    return cursor.fetchall()
print(max_age())

#pd.read_sql_query("SELECT * FROM sqladb", connection)
```

```
[(53,)]
```