Session7_SQL1 & SQL2_Assignment

March 28, 2019

1 Task 1:

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 [1]: import pandas as pd
```

```
In [2]: url= 'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
        adult = pd.read_csv(url, header=None )
```

```
In [3]: adult.head()
```

```
Out[3]:    0                  1       2          3   4                    5   \
        0  39          State-gov   77516  Bachelors  13        Never-married
        1  50   Self-emp-not-inc   83311  Bachelors  13   Married-civ-spouse
        2  38            Private  215646    HS-grad   9             Divorced
        3  53            Private  234721       11th   7   Married-civ-spouse
        4  28            Private  338409  Bachelors  13   Married-civ-spouse

                            6               7      8        9     10  11  12  \
        0        Adm-clerical   Not-in-family  White     Male   2174   0  40
        1     Exec-managerial         Husband  White     Male      0   0  13
        2   Handlers-cleaners   Not-in-family  White     Male      0   0  40
        3   Handlers-cleaners         Husband  Black     Male      0   0  40
        4      Prof-specialty            Wife  Black   Female      0   0  40

                       13      14
        0   United-States   <=50K
        1   United-States   <=50K
        2   United-States   <=50K
        3   United-States   <=50K
        4            Cuba   <=50K
```

```
In [4]: #Rename Columns
        adult.columns = ['age', 'workclass', 'fnlwgt', 'education', 'education_num', 'marital_st
```

```
In [5]: adult.head()
```

```
Out[5]:    age          workclass  fnlwgt  education  education_num  \
        0   39          State-gov   77516  Bachelors             13
        1   50   Self-emp-not-inc   83311  Bachelors             13
        2   38            Private  215646    HS-grad              9
        3   53            Private  234721       11th              7
        4   28            Private  338409  Bachelors             13

               marital_status         occupation   relationship   race     sex  \
        0       Never-married       Adm-clerical  Not-in-family  White    Male
        1  Married-civ-spouse    Exec-managerial        Husband  White    Male
        2            Divorced  Handlers-cleaners  Not-in-family  White    Male
        3  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
        4  Married-civ-spouse     Prof-specialty           Wife  Black  Female

           capital_gain  capital_loss  hours_per_week  native_country  salary
        0          2174             0              40   United-States   <=50K
        1             0             0              13   United-States   <=50K
        2             0             0              40   United-States   <=50K
        3             0             0              40   United-States   <=50K
        4             0             0              40            Cuba   <=50K
```

```
In [6]: import sqlite3 as db
```

```
In [7]: from pandasql import sqldf
        pysqldf = lambda q: sqldf(q, globals())
```

```
In [8]: connection = db.connect('sqladb')
        cursor = connection.cursor()
        cursor.execute("DROP TABLE IF EXISTS AdultTable;")
```

```
Out[8]: <sqlite3.Cursor at 0x86e1110>
```

```
In [9]: adult.to_sql('AdultTable',connection)
```

```
In [10]: #1. Select 10 records from the adult sqladb
         q = "SELECT * FROM AdultTable LIMIT 10;"
         pd.read_sql(q, connection)
```

```
Out[10]:    index  age         workclass  fnlwgt  education  education_num  \
         0      0   39         State-gov   77516  Bachelors             13
         1      1   50  Self-emp-not-inc   83311  Bachelors             13
         2      2   38           Private  215646    HS-grad              9
         3      3   53           Private  234721       11th              7
         4      4   28           Private  338409  Bachelors             13
         5      5   37           Private  284582    Masters             14
         6      6   49           Private  160187        9th              5
         7      7   52  Self-emp-not-inc  209642    HS-grad              9
         8      8   31           Private   45781    Masters             14
         9      9   42           Private  159449  Bachelors             13
```

```
           marital_status          occupation    relationship    race  \
0           Never-married        Adm-clerical   Not-in-family   White
1      Married-civ-spouse     Exec-managerial         Husband   White
2                Divorced   Handlers-cleaners   Not-in-family   White
3      Married-civ-spouse   Handlers-cleaners         Husband   Black
4      Married-civ-spouse      Prof-specialty            Wife   Black
5      Married-civ-spouse     Exec-managerial            Wife   White
6   Married-spouse-absent       Other-service   Not-in-family   Black
7      Married-civ-spouse     Exec-managerial         Husband   White
8           Never-married      Prof-specialty   Not-in-family   White
9      Married-civ-spouse     Exec-managerial         Husband   White

       sex  capital_gain  capital_loss  hours_per_week  native_country  salary
0     Male          2174             0              40   United-States   <=50K
1     Male             0             0              13   United-States   <=50K
2     Male             0             0              40   United-States   <=50K
3     Male             0             0              40   United-States   <=50K
4   Female             0             0              40            Cuba   <=50K
5   Female             0             0              40   United-States   <=50K
6   Female             0             0              16         Jamaica   <=50K
7     Male             0             0              45   United-States    >50K
8   Female         14084             0              50   United-States    >50K
9     Male          5178             0              40   United-States    >50K
```

In [11]:

```
#2. Show me the average hours per week of all men who are working in private sector
q = 'SELECT avg(hours_per_week) Average FROM AdultTable where sex = " Male" and  workcl
pd.read_sql(q, connection)
```

Out[11]:

```
     Average
0  42.221226
```

In [12]:

```
#3. Show me the frequency table for education, occupation and relationship, separately
q = "SELECT education , COUNT(education) frequency FROM AdultTable GROUP BY education "
pd.read_sql(q, connection)
```

Out[12]:

```
         education  frequency
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 [13]: q = "SELECT occupation, COUNT(occupation) frequency FROM AdultTable GROUP BY occupation
         pd.read_sql(q, connection)
```

```
Out[13]:            occupation  frequency
         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 [14]: q = "SELECT relationship, COUNT(relationship) frequency FROM AdultTable GROUP BY relati
         pd.read_sql(q, connection)
```

```
Out[14]:      relationship  frequency
         0         Husband      13193
         1   Not-in-family       8305
         2  Other-relative        981
         3       Own-child       5068
         4       Unmarried       3446
         5            Wife       1568
```

```
In [15]: #4. Are there any people who are married, working in private sector and having a master
         q="""select count(*) Result from AdultTable
             where (marital_status <>'Never-married' or marital_status <> 'Divorced') and workcl
             """
         pd.read_sql(q, connection)
```

```
Out[15]:    Result
         0     894
```

```
In [16]: #5. What is the average, minimum and maximum age group for people working in different
         q="select workclass, avg(age) Avg,min(age) Min, max(age) Max from AdultTable group by w
         pd.read_sql(q, connection)
```

```
Out[16]:           workclass        Avg  Min  Max
         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
```

```
In [17]: #6. Calculate age distribution by country
         q="select native_country,avg(age) Avg,min(age) Min, max(age) min from AdultTable group
         pd.read_sql(q, connection)
```

```
Out[17]:                 native_country        Avg  Min  min
         0                            ?  38.725557   17   90
         1                     Cambodia  37.789474   18   65
         2                       Canada  42.545455   17   80
         3                        China  42.533333   22   75
         4                     Columbia  39.711864   18   75
         5                         Cuba  45.768421   21   82
         6           Dominican-Republic  37.728571   18   78
         7                      Ecuador  36.642857   21   90
         8                  El-Salvador  34.132075   17   79
         9                      England  41.155556   17   90
         10                      France  38.965517   20   64
         11                     Germany  39.255474   18   74
         12                      Greece  46.206897   22   65
         13                   Guatemala  32.421875   19   66
         14                       Haiti  38.272727   17   63
         15          Holand-Netherlands  32.000000   32   32
         16                    Honduras  33.846154   18   58
         17                        Hong  33.650000   19   60
         18                     Hungary  49.384615   24   81
         19                       India  38.090000   17   61
         20                        Iran  39.418605   22   63
         21                     Ireland  36.458333   23   68
         22                       Italy  46.424658   19   77
         23                     Jamaica  35.592593   18   66
         24                       Japan  38.241935   19   61
         25                        Laos  34.722222   19   56
         26                      Mexico  33.290824   17   81
         27                   Nicaragua  33.617647   19   67
         28  Outlying-US(Guam-USVI-etc)  38.714286   21   63
         29                        Peru  35.258065   17   69
         30                 Philippines  39.444444   17   90
         31                      Poland  43.116667   17   85
```

```
32                  Portugal  40.297297   19   78
33               Puerto-Rico  40.508772   17   90
34                  Scotland  40.416667   18   62
35                     South  38.750000   19   90
36                    Taiwan  33.823529   20   61
37                  Thailand  34.944444   19   55
38           Trinadad&Tobago  41.315789   17   61
39             United-States  38.655674   17   90
40                   Vietnam  34.059701   19   73
41                Yugoslavia  38.812500   20   66
```

In [18]:
```
#7. Compute a new column as 'Net-Capital-Gain' from the two columns 'capital-gain' and
q="select capital_gain,capital_loss,(capital_gain - capital_loss) Net_capital_Gain from
pd.read_sql(q, connection)
```

Out[18]:
```
       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
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 x 3 columns]
```

2 Task 2:

Read the following data set: https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

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

```
In [19]: from sqlalchemy import create_engine
         engine = create_engine('sqlite:///sample', echo=True)
```

```
In [20]: conn = engine.connect()
```

```
2019-03-28 00:23:23,772 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS V
2019-03-28 00:23:23,790 INFO sqlalchemy.engine.base.Engine ()
2019-03-28 00:23:23,792 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns' AS
2019-03-28 00:23:23,794 INFO sqlalchemy.engine.base.Engine ()
```

```
In [21]: from sqlalchemy import Table, Column, Integer, String, MetaData, ForeignKey
         metadata = MetaData()
         AdultTable = Table('AdultTable', metadata,
                       Column('id', Integer, primary_key=True),
                       Column('age', Integer),
                       Column('workclass', String),
                       Column('fnlwgt', Integer),
                       Column('education', String),
                       Column('education_num', Integer),
                       Column('marital_status', String),
                       Column('occupation', String),
                       Column('relationship', String),
                       Column('race', String),
                       Column('sex', String),
                       Column('capital_gain', Integer),
                       Column('capital_loss', Integer),
                       Column('hours_per_week', Integer),
                       Column('native_country', String),
                       Column('salary', String)
         )
         metadata.create_all(engine)
```

```
2019-03-28 00:23:27,602 INFO sqlalchemy.engine.base.Engine PRAGMA table_info("AdultTable")
2019-03-28 00:23:27,607 INFO sqlalchemy.engine.base.Engine ()
```

```
In [32]: insert_1 = AdultTable.insert().values(age=39,workclass='State-gov', fnlwgt=77516
                                        ,education='Bachelors', education_num=13,marital_status=
                                     ,relationship='Not-in-family', race='white',sex='Male', cap
                                       ,capital_loss=0, hours_per_week=40,native_country='united
                                       )
         insert_2 = AdultTable.insert().values(age=50,workclass='Self-emp-not-inc', fnlwgt=83311
                                        ,education='Bachelors', education_num=13,marital_status=
                                       ,relationship='Husband', race='white',sex='Male', capital
                                       ,capital_loss=0, hours_per_week=13,native_country='united
                                       )
         insert_3 = AdultTable.insert().values(age=38,workclass='Private', fnlwgt=215646
                                        ,education='HS-grad', education_num=9,marital_status='Di
                                       ,relationship='Not-in-family', race='white',sex='Male', c
                                       ,capital_loss=0, hours_per_week=40,native_country='united
                                       )
         conn.execute(insert_1)
         conn.execute(insert_2)
         conn.execute(insert_3)
```

```
2019-03-28 01:01:26,446 INFO sqlalchemy.engine.base.Engine INSERT INTO "AdultTable" (age, workcl
2019-03-28 01:01:26,449 INFO sqlalchemy.engine.base.Engine (39, 'State-gov', 77516, 'Bachelors',
2019-03-28 01:01:26,453 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
2019-03-28 01:01:26,694 INFO sqlalchemy.engine.base.Engine INSERT INTO "AdultTable" (age, workcl
2019-03-28 01:01:26,698 INFO sqlalchemy.engine.base.Engine (50, 'Self-emp-not-inc', 83311, 'Bach
2019-03-28 01:01:26,711 INFO sqlalchemy.engine.base.Engine COMMIT
2019-03-28 01:01:26,869 INFO sqlalchemy.engine.base.Engine INSERT INTO "AdultTable" (age, workcl
2019-03-28 01:01:26,872 INFO sqlalchemy.engine.base.Engine (38, 'Private', 215646, 'HS-grad', 9,
2019-03-28 01:01:26,879 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
Out[32]: <sqlalchemy.engine.result.ResultProxy at 0x920b630>
```

```
In [33]: from sqlalchemy.sql import select
         result = conn.execute(select([AdultTable]))
         row = result.fetchall()
         print(row)
```

```
2019-03-28 01:01:49,584 INFO sqlalchemy.engine.base.Engine SELECT "AdultTable".id, "AdultTable".
FROM "AdultTable"
2019-03-28 01:01:49,585 INFO sqlalchemy.engine.base.Engine ()
[(1, 39, 'State-gov', 77516, 'Bachelors', 13, 'Never-married', 'Adm-clerical', 'Not-in-family',
```

```
In [34]: #2. Write two basic update queries
         from sqlalchemy import update

         stmt = update(AdultTable).where(AdultTable.c.id==1).values(age=60)
         stmt1 = update(AdultTable).where(AdultTable.c.id==1).values(marital_status='Divorced')
         conn.execute(stmt)
         conn.execute(stmt1)
```

```
2019-03-28 01:01:59,492 INFO sqlalchemy.engine.base.Engine UPDATE "AdultTable" SET age=? WHERE "
2019-03-28 01:01:59,493 INFO sqlalchemy.engine.base.Engine (60, 1)
2019-03-28 01:01:59,498 INFO sqlalchemy.engine.base.Engine COMMIT
2019-03-28 01:01:59,701 INFO sqlalchemy.engine.base.Engine UPDATE "AdultTable" SET marital_statu
2019-03-28 01:01:59,704 INFO sqlalchemy.engine.base.Engine ('Divorced', 1)
2019-03-28 01:01:59,711 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
Out[34]: <sqlalchemy.engine.result.ResultProxy at 0x9954a20>
```

```
In [35]: result = conn.execute(select([AdultTable]))
         row = result.fetchall()
         print(row)
```

```
2019-03-28 01:02:04,262 INFO sqlalchemy.engine.base.Engine SELECT "AdultTable".id, "AdultTable".
FROM "AdultTable"
2019-03-28 01:02:04,265 INFO sqlalchemy.engine.base.Engine ()
[(1, 60, 'State-gov', 77516, 'Bachelors', 13, 'Divorced', 'Adm-clerical', 'Not-in-family', 'whit
In [36]: #3. Write two delete queries
         stmt1=AdultTable.delete().where(AdultTable.c.id==1)
         stmt2=AdultTable.delete().where(AdultTable.c.id==2 and AdultTable.c.workclass=='Self-em
         conn.execute(stmt1)
         conn.execute(stmt2)
```

```
2019-03-28 01:02:11,865 INFO sqlalchemy.engine.base.Engine DELETE FROM "AdultTable" WHERE "Adult
2019-03-28 01:02:11,868 INFO sqlalchemy.engine.base.Engine (1,)
2019-03-28 01:02:11,870 INFO sqlalchemy.engine.base.Engine COMMIT
2019-03-28 01:02:12,092 INFO sqlalchemy.engine.base.Engine DELETE FROM "AdultTable" WHERE "Adult
2019-03-28 01:02:12,095 INFO sqlalchemy.engine.base.Engine (2,)
2019-03-28 01:02:12,105 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
Out[36]: <sqlalchemy.engine.result.ResultProxy at 0x878e978>
```

```
In [37]: result = conn.execute(select([AdultTable]))
         row = result.fetchall()
         print(row)
```

```
2019-03-28 01:02:17,092 INFO sqlalchemy.engine.base.Engine SELECT "AdultTable".id, "AdultTable".
FROM "AdultTable"
2019-03-28 01:02:17,094 INFO sqlalchemy.engine.base.Engine ()
[(3, 38, 'Private', 215646, 'HS-grad', 9, 'Divorced', 'Handlers-cleaners', 'Not-in-family', 'whi
```

```
In [38]: #4. Write two filter queries
         from sqlalchemy.orm import sessionmaker
         Session = sessionmaker(bind=engine)
         session = Session()
```

```
In [39]: stmt1= session.query(AdultTable).filter(AdultTable.c.sex=='Male').first()
         stmt2= session.query(AdultTable).filter(AdultTable.c.hours_per_week==40).first()
```

```
2019-03-28 01:02:35,469 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2019-03-28 01:02:35,482 INFO sqlalchemy.engine.base.Engine SELECT "AdultTable".id AS "AdultTable
FROM "AdultTable"
WHERE "AdultTable".sex = ?
 LIMIT ? OFFSET ?
2019-03-28 01:02:35,484 INFO sqlalchemy.engine.base.Engine ('Male', 1, 0)
2019-03-28 01:02:35,490 INFO sqlalchemy.engine.base.Engine SELECT "AdultTable".id AS "AdultTable
FROM "AdultTable"
WHERE "AdultTable".hours_per_week = ?
 LIMIT ? OFFSET ?
2019-03-28 01:02:35,491 INFO sqlalchemy.engine.base.Engine (40, 1, 0)
```

```
In [40]: #5. Write two function queries
         from sqlalchemy import func
```

```
stmt1 = session.query(func.count('*')).select_from(AdultTable).scalar()
stmt2 = session.query(func.sum(AdultTable.c.hours_per_week)).scalar()
print(stmt1)
print(stmt2)
```

```
2019-03-28 01:02:43,915 INFO sqlalchemy.engine.base.Engine SELECT count(?) AS count_1
FROM "AdultTable"
2019-03-28 01:02:43,921 INFO sqlalchemy.engine.base.Engine ('*',)
2019-03-28 01:02:43,926 INFO sqlalchemy.engine.base.Engine SELECT sum("AdultTable".hours_per_wee
FROM "AdultTable"
2019-03-28 01:02:43,930 INFO sqlalchemy.engine.base.Engine ()
4
133
```

```
In [ ]:
```