

```
In [1]: #import numpy as np
import pandas as pd
from pandas import DataFrame, Series
import sqlite3 as db
```

```
In [2]: # Problem Statement: Read the following data set: https://archive.ics.uci.edu/ml/m
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
col_list = ['age','workclass','fnlwgt','education','education-num','marital-status',
'relationship','race','sex','capital-gain','capital-loss','hours-per-we
```

```
In [3]: # Import the data from the url above into Pandas DataFrame

adult = pd.read_csv(url,sep=",",delimiter=",",names=col_list,skipinitialspace=True)
```

```
In [4]: print(adult.columns)
import re
adult.columns = [re.sub("[ -]", "_", col) for col in adult.columns]

print(adult.columns)
```

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'Label'],
      dtype='object')
Index(['age', 'workclass', 'fnlwgt', 'education', 'education_num',
       'marital_status', 'occupation', 'relationship', 'race', 'sex',
       'capital_gain', 'capital_loss', 'hours_per_week', 'native_country',
       'Label'],
      dtype='object')
```

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

Out[5]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	ra
0	39	State-gov	77516	Bachelors		13	Never-married	Adm-clerical	Not-in-family Wh
1	50	Self-emp-not-inc	83311	Bachelors		13	Married-civ-spouse	Exec-managerial	Husband Wh
2	38	Private	215646	HS-grad		9	Divorced	Handlers-cleaners	Not-in-family Wh
3	53	Private	234721	11th		7	Married-civ-spouse	Handlers-cleaners	Husband Bla
4	28	Private	338409	Bachelors		13	Married-civ-spouse	Prof-specialty	Wife Bla

```
In [6]: # Task
```

```
In [7]: # Create an sqlalchemy engine using a sample from the data set  
# By importing corresponding sqlalchemy library  
  
from sqlalchemy import *  
from sqlalchemy import create_engine, ForeignKey  
from sqlalchemy import Column, Date, Integer, String  
from sqlalchemy.ext.declarative import declarative_base  
from sqlalchemy.orm import relationship, backref
```

```
In [8]: # Create database engine  
  
engine = create_engine('sqlite://', echo=True)
```

In [9]: # connect to database and Load the dataframe adultdb into sqlite

```
adult.to_sql('adultdb', con=engine)
```

```
2018-09-16 22:43:50,228 INFO sqlalchemy.engine.base.Engine SELECT CAST('test p
lain returns' AS VARCHAR(60)) AS anon_1
2018-09-16 22:43:50,231 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:43:50,233 INFO sqlalchemy.engine.base.Engine SELECT CAST('test u
nicode returns' AS VARCHAR(60)) AS anon_1
2018-09-16 22:43:50,235 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:43:50,237 INFO sqlalchemy.engine.base.Engine PRAGMA table_info
("adultdb")
2018-09-16 22:43:50,238 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:43:50,242 INFO sqlalchemy.engine.base.Engine
CREATE TABLE adultdb (
    "index" BIGINT,
    age BIGINT,
    workclass TEXT,
    fnlwgt BIGINT,
    education TEXT,
    education_num BIGINT,
    marital_status TEXT,
    occupation TEXT,
    relationship TEXT,
    race TEXT,
    sex TEXT,
    capital_gain BIGINT,
    capital_loss BIGINT,
    hours_per_week BIGINT,
    native_country TEXT,
    "Label" TEXT
)

```

```
2018-09-16 22:43:50,243 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:43:50,245 INFO sqlalchemy.engine.base.Engine COMMIT
2018-09-16 22:43:50,248 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_adu
ltdb_index ON adultdb ("index")
2018-09-16 22:43:50,249 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:43:50,252 INFO sqlalchemy.engine.base.Engine COMMIT
2018-09-16 22:43:50,290 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 22:43:51,007 INFO sqlalchemy.engine.base.Engine INSERT INTO adultdb
("index", age, workclass, fnlwgt, education, education_num, marital_status, oc
cupation, relationship, race, sex, capital_gain, capital_loss, hours_per_week,
native_country, "Label") VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?)
2018-09-16 22:43:51,009 INFO sqlalchemy.engine.base.Engine ((0, 39, 'State-go
v', 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', 'Husba
nd', 'White', 'Male', 0, 0, 13, 'United-States', '<=50K'), (2, 38, 'Private',
215646, 'HS-grad', 9, 'Divorced', 'Handlers-cleaners', 'Not-in-family', 'Whit
e', 'Male', 0, 0, 40, 'United-States', '<=50K'), (3, 53, 'Private', 234721, '1
1th', 7, 'Married-civ-spouse', 'Handlers-cleaners', 'Husband', 'Black', 'Mal
e', 0, 0, 40, 'United-States', '<=50K'), (4, 28, 'Private', 338409, 'Bachelor
s', 13, 'Married-civ-spouse', 'Prof-specialty', 'Wife', 'Black', 'Female', 0,
```

```
0, 40, 'Cuba', '<=50K'), (5, 37, 'Private', 284582, 'Masters', 14, 'Married-civ-spouse', 'Exec-managerial', 'Wife', 'White', 'Female', 0, 0, 40, 'United-States', '<=50K'), (6, 49, 'Private', 160187, '9th', 5, 'Married-spouse-absent', 'Other-service', 'Not-in-family', 'Black', 'Female', 0, 0, 16, 'Jamaica', '<=50K'), (7, 52, 'Self-emp-not-inc', 209642, 'HS-grad', 9, 'Married-civ-spouse', 'Exec-managerial', 'Husband', 'White', 'Male', 0, 0, 45, 'United-States', '>50K') ... displaying 10 of 32561 total bound parameter sets ... (32559, 22, 'Private', 201490, 'HS-grad', 9, 'Never-married', 'Adm-clerical', 'Own-child', 'White', 'Male', 0, 0, 20, 'United-States', '<=50K'), (32560, 52, 'Self-emp-in-c', 287927, 'HS-grad', 9, 'Married-civ-spouse', 'Exec-managerial', 'Wife', 'White', 'Female', 15024, 0, 40, 'United-States', '>50K'))  
2018-09-16 22:43:51,209 INFO sqlalchemy.engine.base.Engine COMMIT
```

```
In [12]: from sqlalchemy.orm import Session
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

# create an Object to hold SQLAlchemy data types to map properties of Python class

Base = declarative_base(engine)
class Adultdb(Base):
    """
    eg. fields: id, title
    """
    __tablename__ = 'adultdb'
    __table_args__ = {'autoload': True}

    index = Column(Integer(), primary_key=True)
    age = Column(Integer())
    workclass = Column(String())

#Create SQLAlchemy Sessions

def loadSession():
    """
    metadata = Base.metadata
    Session = sessionmaker(bind=engine)
    session = Session()
    return session
```

```
2018-09-16 22:48:55,590 INFO sqlalchemy.engine.base.Engine PRAGMA table_info
("adultdb")
2018-09-16 22:48:55,592 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,598 INFO sqlalchemy.engine.base.Engine SELECT sql FROM (S
ELECT * FROM sqlite_master UNION ALL SELECT * FROM sqlite_temp_master) WHERE
name = 'adultdb' AND type = 'table'
2018-09-16 22:48:55,601 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,605 INFO sqlalchemy.engine.base.Engine PRAGMA foreign_key_
list("adultdb")
2018-09-16 22:48:55,607 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,610 INFO sqlalchemy.engine.base.Engine SELECT sql FROM (S
ELECT * FROM sqlite_master UNION ALL SELECT * FROM sqlite_temp_master) WHERE
name = 'adultdb' AND type = 'table'
2018-09-16 22:48:55,612 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,614 INFO sqlalchemy.engine.base.Engine PRAGMA index_list
("adultdb")
2018-09-16 22:48:55,616 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,619 INFO sqlalchemy.engine.base.Engine PRAGMA index_info
("ix_adultdb_index")
2018-09-16 22:48:55,621 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,624 INFO sqlalchemy.engine.base.Engine PRAGMA index_list
("adultdb")
2018-09-16 22:48:55,626 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,628 INFO sqlalchemy.engine.base.Engine PRAGMA index_info
("ix_adultdb_index")
2018-09-16 22:48:55,631 INFO sqlalchemy.engine.base.Engine ()
2018-09-16 22:48:55,634 INFO sqlalchemy.engine.base.Engine SELECT sql FROM (S
ELECT * FROM sqlite_master UNION ALL SELECT * FROM sqlite_temp_master) WHERE
```

```
name = 'adultdb' AND type = 'table'
2018-09-16 22:48:55,637 INFO sqlalchemy.engine.base.Engine ()
```

In [13]: #Querying Data with SQLAlchemy ORM

```
if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).first()
    print("AGE", "SEX", "WORKCLAS", "COUNTRY", "OCCUPATION")
    print("*"*80)
    print(rows.age, rows.sex, rows.workclass, rows.native_country, rows.occupation)
```

```
2018-09-16 22:49:42,733 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 22:49:42,736 INFO sqlalchemy.engine.base.Engine SELECT adultdb."ind
ex" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_
workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_edu
cation, adultdb.education_num AS adultdb_education_num, adultdb.marital_status
AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.r
elationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex
AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_
loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_wee
k, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adult
db_Label"
FROM adultdb
LIMIT ? OFFSET ?
2018-09-16 22:49:42,738 INFO sqlalchemy.engine.base.Engine (1, 0)
AGE SEX WORKCLAS COUNTRY OCCUPATION
*****
**
39 Male State-gov United-States Adm-clerical
```

In [14]: # Write two basic update queries

In [15]: # Update query-1

```
if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).filter_by(fnlwgt=77516).first()
    print(rows)
    rows.occupation = 'Student'
    rows.workclass = 'Study'
    session.commit()
```

```
2018-09-16 22:59:45,702 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 22:59:45,708 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adultdb_Label"
FROM adultdb
WHERE adultdb.fnlwgt = ?
    LIMIT ? OFFSET ?
2018-09-16 22:59:45,711 INFO sqlalchemy.engine.base.Engine (77516, 1, 0)
<__main__.Adultdb object at 0x000002578863BE48>
2018-09-16 22:59:45,720 INFO sqlalchemy.engine.base.Engine UPDATE adultdb SET workclass=? , occupation=? WHERE adultdb."index" = ?
2018-09-16 22:59:45,722 INFO sqlalchemy.engine.base.Engine ('Study', 'Student', 0)
2018-09-16 22:59:45,727 INFO sqlalchemy.engine.base.Engine COMMIT
```

In [16]: # Verify update results

```
if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).filter_by(fnlwgt=77516).first()
    print("occupation : ",rows.occupation,"workclass : ",rows.workclass,"fnlwgt : "
```

```
2018-09-16 23:01:04,683 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:01:04,689 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adultdb_Label"
FROM adultdb
WHERE adultdb.fnlwgt = ?
    LIMIT ? OFFSET ?
2018-09-16 23:01:04,691 INFO sqlalchemy.engine.base.Engine (77516, 1, 0)
occupation : Student workclass : Study fnlwgt : 77516
```

In [17]: # update query-2

```
if __name__ == "__main__":
    session = loadSession()
    row = session.query(Adultdb).filter_by(age=60, education='Masters').all()
    for i in row:
        i.occupation = 'Retired'
    session.commit()
```

```
2018-09-16 23:01:54,814 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:01:54,820 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "aduldb_Label"
FROM adultdb
WHERE adultdb.age = ? AND adultdb.education = ?
2018-09-16 23:01:54,822 INFO sqlalchemy.engine.base.Engine (60, 'Masters')
2018-09-16 23:01:54,842 INFO sqlalchemy.engine.base.Engine UPDATE adultdb SET occupation=? WHERE adultdb."index" = ?
2018-09-16 23:01:54,845 INFO sqlalchemy.engine.base.Engine (('Retired', 1504), ('Retired', 5129), ('Retired', 5501), ('Retired', 7397), ('Retired', 10177), ('Retired', 12354), ('Retired', 12744), ('Retired', 14004) ... displaying 10 of 20 total bound parameter sets ... ('Retired', 30700), ('Retired', 31302))
2018-09-16 23:01:54,853 INFO sqlalchemy.engine.base.Engine COMMIT
```

In [18]: # Verify query-2

```
if __name__ == "__main__":
    session = loadSession()
    row = session.query(Adultdb).filter_by(age=60, education='Masters').all()
    for i in row:
        print(i.age, i.education, i.sex, i.occupation, i.workclass)
```

```
2018-09-16 23:02:33,003 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:02:33,008 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adultdb_Label"
FROM adultdb
WHERE adultdb.age = ? AND adultdb.education = ?
2018-09-16 23:02:33,012 INFO sqlalchemy.engine.base.Engine (60, 'Masters')
60 Masters Male Retired ?
60 Masters Male Retired Self-emp-not-inc
60 Masters Male Retired Private
60 Masters Male Retired Private
60 Masters Male Retired Private
60 Masters Male Retired Local-gov
60 Masters Female Retired Private
60 Masters Female Retired State-gov
60 Masters Male Retired Local-gov
60 Masters Female Retired Local-gov
60 Masters Male Retired Local-gov
60 Masters Female Retired Self-emp-not-inc
60 Masters Male Retired Self-emp-not-inc
60 Masters Male Retired Private
60 Masters Male Retired Local-gov
60 Masters Male Retired State-gov
60 Masters Male Retired Private
60 Masters Male Retired ?
60 Masters Male Retired Local-gov
60 Masters Male Retired Private
```

In [19]: # Write two delete queries

In [20]: #Check for all rows which have occupation as ? - Querying Data with SQLAlchemy ORM

```
if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).filter_by(occupation="?").all()
    print("Count of rows before delete operation : ",len(rows))
```

```
2018-09-16 23:05:00,352 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:05:00,356 INFO sqlalchemy.engine.base.Engine SELECT adultdb."ind
ex" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb
_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_edu
cation, adultdb.education_num AS adultdb_education_num, adultdb.marital_status
AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.r
elationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex
AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_
loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_wee
k, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adult
db_Label"
FROM adultdb
WHERE adultdb.occupation = ?
2018-09-16 23:05:00,358 INFO sqlalchemy.engine.base.Engine ('?',)
Count of rows before delete operation : 1841
```

In [21]: # Delete-1

```
#Delete rows which have occupation as "?" - Delete Data with SQLAlchemy ORM
if __name__ == "__main__":
    session = loadSession()
    session.query(Adultdb).filter_by(occupation="?").delete(synchronize_session='f')
    session.commit()
    rows = session.query(Adultdb).filter_by(occupation=?).all()
    print("Count of rows after delete operation : ",len(rows))
```

```
2018-09-16 23:05:25,257 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:05:25,260 INFO sqlalchemy.engine.base.Engine SELECT adultdb."ind
ex" AS adultdb_index
FROM adultdb
WHERE adultdb.occupation = ?
2018-09-16 23:05:25,263 INFO sqlalchemy.engine.base.Engine ('?',)
2018-09-16 23:05:25,296 INFO sqlalchemy.engine.base.Engine DELETE FROM adultdb
WHERE adultdb.occupation = ?
2018-09-16 23:05:25,298 INFO sqlalchemy.engine.base.Engine ('?',)
2018-09-16 23:05:25,324 INFO sqlalchemy.engine.base.Engine COMMIT
2018-09-16 23:05:25,330 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:05:25,335 INFO sqlalchemy.engine.base.Engine SELECT adultdb."ind
ex" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_
workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_edu
cation, adultdb.education_num AS adultdb_education_num, adultdb.marital_status
AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.r
elationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex
AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_
loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_wee
k, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adult
db_Label"
FROM adultdb
WHERE adultdb.occupation = ?
2018-09-16 23:05:25,338 INFO sqlalchemy.engine.base.Engine ('?',)
Count of rows after delete operation :  0
```

In [22]: # Delete-2

```
#Check for all rows which have education as Some-college- Querying Data with SQLA
if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).filter_by(education="Some-college").all()
    print("count of rows :",len(rows))
```

```
2018-09-16 23:06:28,648 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:06:28,652 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "aduldb_Label"
FROM adultdb
WHERE adultdb.education = ?
2018-09-16 23:06:28,654 INFO sqlalchemy.engine.base.Engine ('Some-college',)
count of rows : 6775
```

```
In [23]: # #delete rows from table adultdb which have education as "Some-college" - Delete

if __name__ == "__main__":
    session = loadSession()
    session.query(Adultdb).filter_by(education="Some-college").delete(synchronize_session=False)
    session.commit()
    rows = session.query(Adultdb).filter_by(education="Some-college").all()
    print("Count of rows after Delete : ", len(rows))
```

```
2018-09-16 23:07:27,120 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:07:27,124 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index
FROM adultdb
WHERE adultdb.education = ?
2018-09-16 23:07:27,128 INFO sqlalchemy.engine.base.Engine ('Some-college',)
2018-09-16 23:07:27,168 INFO sqlalchemy.engine.base.Engine DELETE FROM adultdb
WHERE adultdb.education = ?
2018-09-16 23:07:27,170 INFO sqlalchemy.engine.base.Engine ('Some-college',)
2018-09-16 23:07:27,219 INFO sqlalchemy.engine.base.Engine COMMIT
2018-09-16 23:07:27,223 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-16 23:07:27,225 INFO sqlalchemy.engine.base.Engine SELECT adultdb."index" AS adultdb_index, adultdb.age AS adultdb_age, adultdb.workclass AS adultdb_workclass, adultdb.fnlwgt AS adultdb_fnlwgt, adultdb.education AS adultdb_education, adultdb.education_num AS adultdb_education_num, adultdb.marital_status AS adultdb_marital_status, adultdb.occupation AS adultdb_occupation, adultdb.relationship AS adultdb_relationship, adultdb.race AS adultdb_race, adultdb.sex AS adultdb_sex, adultdb.capital_gain AS adultdb_capital_gain, adultdb.capital_loss AS adultdb_capital_loss, adultdb.hours_per_week AS adultdb_hours_per_week, adultdb.native_country AS adultdb_native_country, adultdb."Label" AS "adult_db_Label"
FROM adultdb
WHERE adultdb.education = ?
2018-09-16 23:07:27,226 INFO sqlalchemy.engine.base.Engine ('Some-college')
Count of rows after Delete : 0
```

```
In [24]: # Write two filter queries
```

```
In [25]: # Filter Query-1
# Querying Data with SQLAlchemy ORM based on filter on workclass column having a value 'Private'

if __name__ == "__main__":
    session = loadSession()
    rows = session.query(Adultdb).filter_by(workclass='Private').all()
    for i in rows:
        print(i.age,i.sex,i.workclass,i.native_country)
```

```
50 Female Private Mexico
52 Female Private United-States
53 Male Private United-States
30 Female Private United-States
23 Male Private Mexico
23 Male Private ?
40 Male Private United-States
55 Male Private United-States
56 Male Private United-States
23 Male Private United-States
25 Male Private United-States
27 Female Private Mexico
21 Female Private United-States
27 Female Private United-States
33 Male Private United-States
28 Female Private United-States
28 Male Private United-States
29 Male Private United-States
46 Female Private United-States
```

In [27]: # Querying the Data with SQLALCHEMY ORM based on filter on marital_status column here

```
count = 0
for rows in engine.execute("SELECT * FROM adultdb where marital_status == 'Never-married'"):
    print(rows.age,",",rows.marital_status,",",rows.occupation)
    count+=1

print("Total Number of rows fetched : ",count)
```

27 , Never-married , Exec-managerial
30 , Never-married , Prof-specialty
33 , Never-married , Transport-moving
37 , Never-married , Other-service
41 , Never-married , Craft-repair
18 , Never-married , Handlers-cleaners
24 , Never-married , Tech-support
28 , Never-married , Other-service
20 , Never-married , Transport-moving
22 , Never-married , Tech-support
26 , Never-married , Prof-specialty
20 , Never-married , Farming-fishing
24 , Never-married , Sales
18 , Never-married , Other-service
17 , Never-married , Handlers-cleaners
34 , Never-married , Sales
39 , Never-married , Other-service
19 , Never-married , Other-service
29 , Never-married , Prof-specialty
17 Never-married Other-service

In [28]: # Querying Data with SQLAlchemy ORM based on filter on native_country column having

```
count = 0
for rows in engine.execute("SELECT age,sex,occupation FROM adultdb where native_country = 'Columbia'"):
    print(rows.age, " ", rows.sex, " ", rows.occupation)
    count+=1
print("Total Records Retrieved : ",count)
```

```
2018-09-16 23:30:12,448 INFO sqlalchemy.engine.base.Engine SELECT age,sex,occupation FROM adultdb where native_country = 'Columbia'
2018-09-16 23:30:12,451 INFO sqlalchemy.engine.base.Engine ()
75   Female  Adm-clerical
26   Male    Prof-specialty
49   Female  Exec-managerial
38   Male    Sales
50   Male    Adm-clerical
53   Female  Machine-op-inspct
26   Female  Machine-op-inspct
43   Male    Tech-support
29   Male    Exec-managerial
64   Male    Prof-specialty
26   Male    Transport-moving
25   Female  Machine-op-inspct
23   Male    Craft-repair
49   Male    Other-service
33   Female  Machine-op-inspct
28   Male    Craft-repair
31   Male    Prof-specialty
36   Female  Adm-clerical
31   Male    Machine-op-inspct
21   Male    Adm-clerical
32   Male    Handlers-cleaners
34   Female  Machine-op-inspct
36   Male    Adm-clerical
18   Male    Handlers-cleaners
51   Male    Craft-repair
41   Female  Priv-house-serv
23   Male    Other-service
39   Female  Other-service
45   Male    Adm-clerical
45   Male    Machine-op-inspct
29   Female  Other-service
29   Female  Other-service
25   Female  Other-service
39   Female  Prof-specialty
62   Female  Tech-support
32   Male    Handlers-cleaners
64   Male    Craft-repair
25   Female  Sales
32   Female  Machine-op-inspct
52   Male    Machine-op-inspct
49   Male    Craft-repair
56   Male    Exec-managerial
57   Female  Craft-repair
48   Female  Other-service
46   Female  Machine-op-inspct
```

```
49    Male    Transport-moving
49    Male    Prof-specialty
Total Records Retrieved : 47
```

In [29]: # Write two function queries

In [30]: # Querying Data with SQLAlchemy ORM - Use aggregate Function -count the number of

```
# Function Query-1
from sqlalchemy.sql import func
if __name__ == "__main__":
    session = loadSession()
    result = session.query(func.count(Adultdb.age).label('count_age'), Adultdb.sex)
    print("Count of people based on sex ")
    print('*'*80)

    for rows in result:
        print(rows)
```

```
2018-09-17 09:40:35,473 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-17 09:40:35,489 INFO sqlalchemy.engine.base.Engine SELECT count(adultdb.age) AS count_age, adultdb.sex AS adultdb_sex
FROM adultdb GROUP BY adultdb.sex
2018-09-17 09:40:35,490 INFO sqlalchemy.engine.base.Engine ()
Count of people based on sex
*****
(7391, 'Female')
(16554, 'Male')
```

In [31]: # Querying Data with SQLAlchemy ORM - Use aggregate Function -average age and mini

```
# Function Query-2
if __name__ == "__main__":
    session = loadSession()
    result = session.query(func.avg(Adultdb.age).label('avg_age'),func.min(Adultdb.age))
    print("Average age,minimum age of people based on sex ")
    print('*'*80)
    for rows in result:
        print(rows)
```

```
2018-09-17 09:42:12,173 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2018-09-17 09:42:12,175 INFO sqlalchemy.engine.base.Engine SELECT avg(adultdb.age) AS avg_age, min(adultdb.age) AS min_age, adultdb.sex AS adultdb_sex
FROM adultdb GROUP BY adultdb.sex
2018-09-17 09:42:12,175 INFO sqlalchemy.engine.base.Engine ()
Average age,minimum age of people based on sex
*****
(37.77864970910567, 17, 'Female')
(39.699287181345895, 17, 'Male')
```

In []:

