2.​ Problem Statement

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>

import numpy as np

import pandas as pd

from pandas import DataFrame, Series

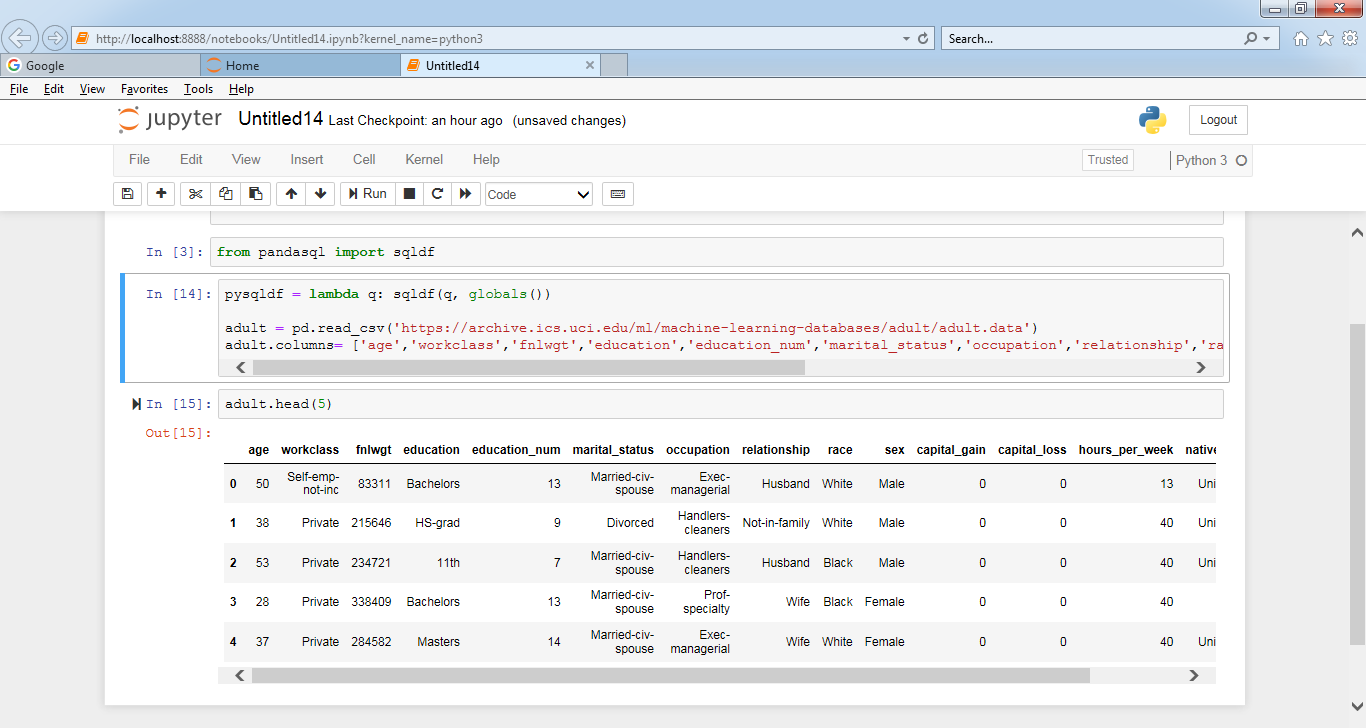
import sqlite3 as db

from pandasql import sqldf

pysqldf = lambda q: sqldf(q, globals())

adult = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data')

adult.columns= ['age','workclass','fnlwgt','education','education\_num','marital\_status','occupation','relationship','race','sex','capital\_gain','capital\_loss','hours\_per\_week','native\_country','class']



adultname = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test')

Task:

conn = db.connect('sqladb.db') #creating database connection\n",

cur = conn.cursor() # getting cursor from connection\n",

cur.execute("""create table adult(

id INTEGER Primary key AUTOINCREMENT,

age int,

workclass VARCHAR(50),

fnlwgt int,

education VARCHAR(50),

education\_num int,

marital\_status VARCHAR(50),

occupation VARCHAR(50),

relationship VARCHAR(50),

race VARCHAR(20),

sex VARCHAR(20),

capital\_gain int,

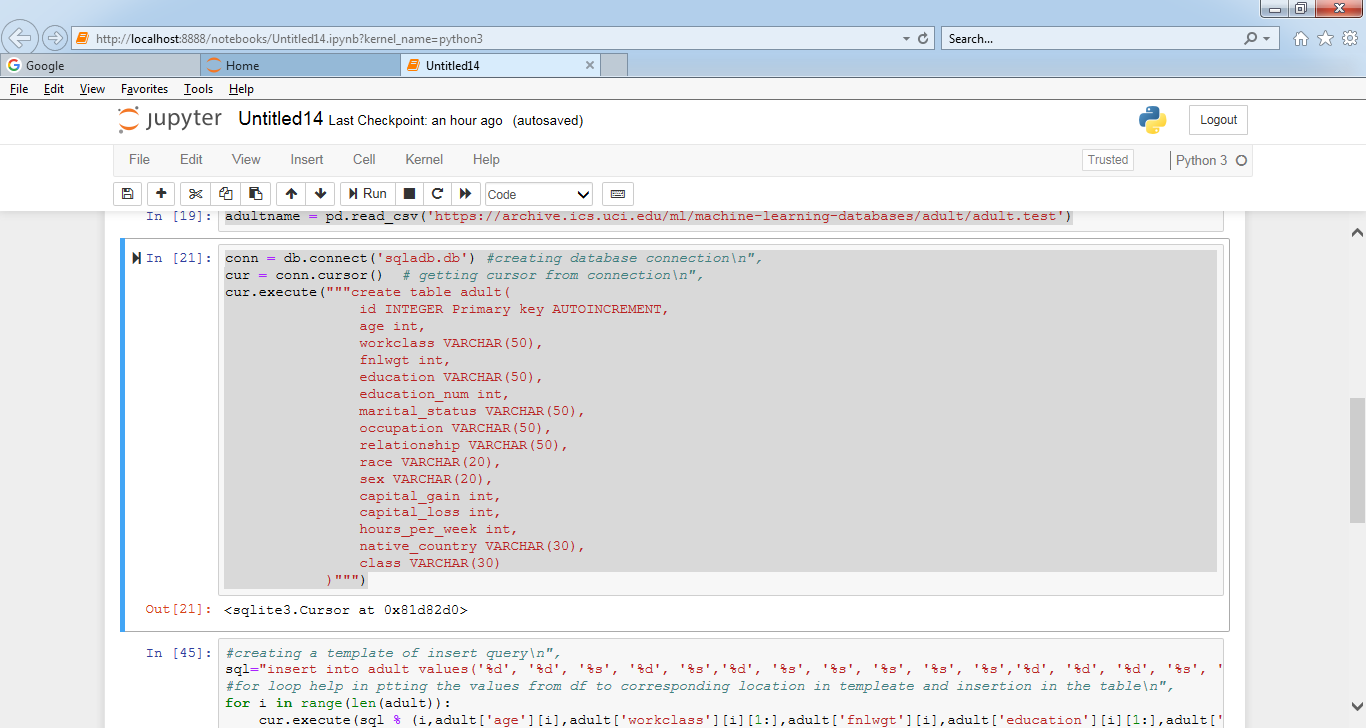
capital\_loss int,

hours\_per\_week int,

native\_country VARCHAR(30),

class VARCHAR(30)

)""")



Create a sql db from adult dataset and name it sqladb

#creating a template of insert query\n",

sql="insert into adult values('%d', '%d', '%s', '%d', '%s','%d', '%s', '%s', '%s', '%s', '%s','%d', '%d', '%d', '%s', '%s')"

#for loop help in ptting the values from df to corresponding location in templeate and insertion in the table\n",

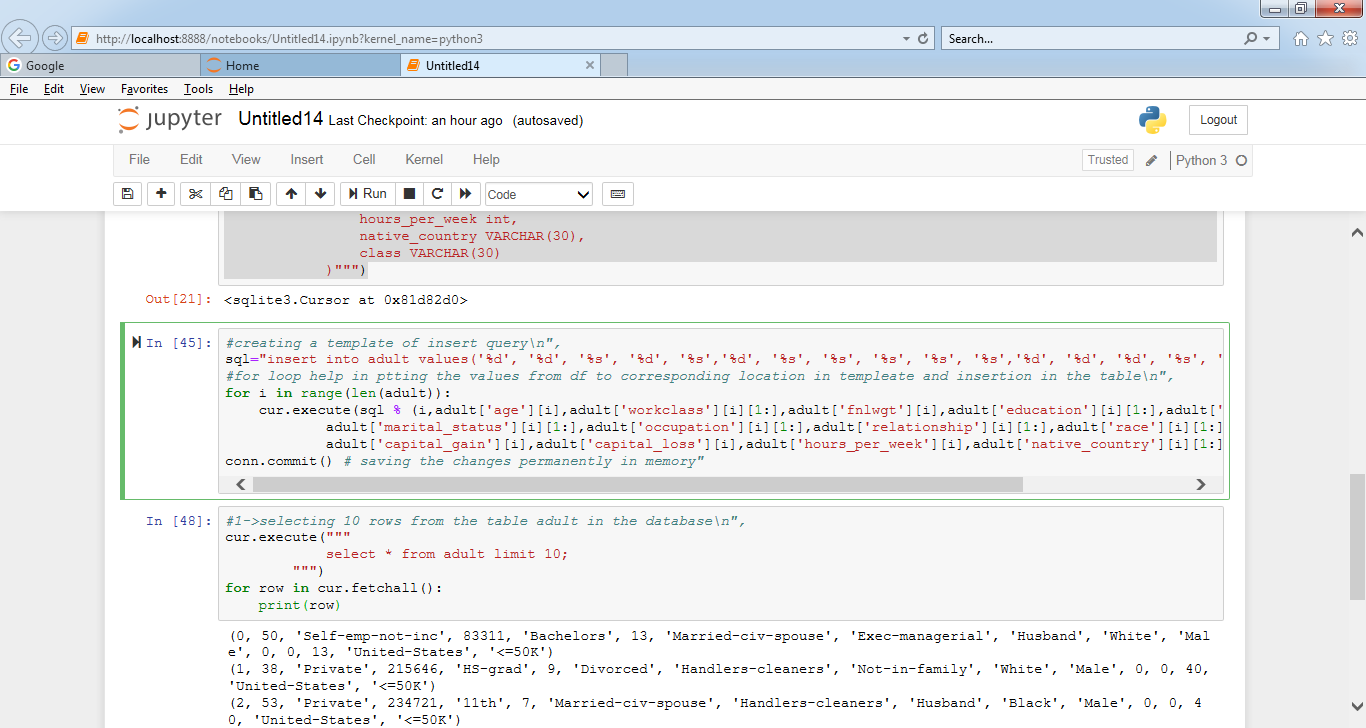
for i in range(len(adult)):

cur.execute(sql % (i,adult['age'][i],adult['workclass'][i][1:],adult['fnlwgt'][i],adult['education'][i][1:],adult['education\_num'][i],

adult['marital\_status'][i][1:],adult['occupation'][i][1:],adult['relationship'][i][1:],adult['race'][i][1:],adult['sex'][i][1:],

adult['capital\_gain'][i],adult['capital\_loss'][i],adult['hours\_per\_week'][i],adult['native\_country'][i][1:],adult['class'][i][1:]))

conn.commit() # saving the changes permanently in memory"



**1. Select 10 records from the adult sqladb**

#1->selecting 10 rows from the table adult in the database\n",

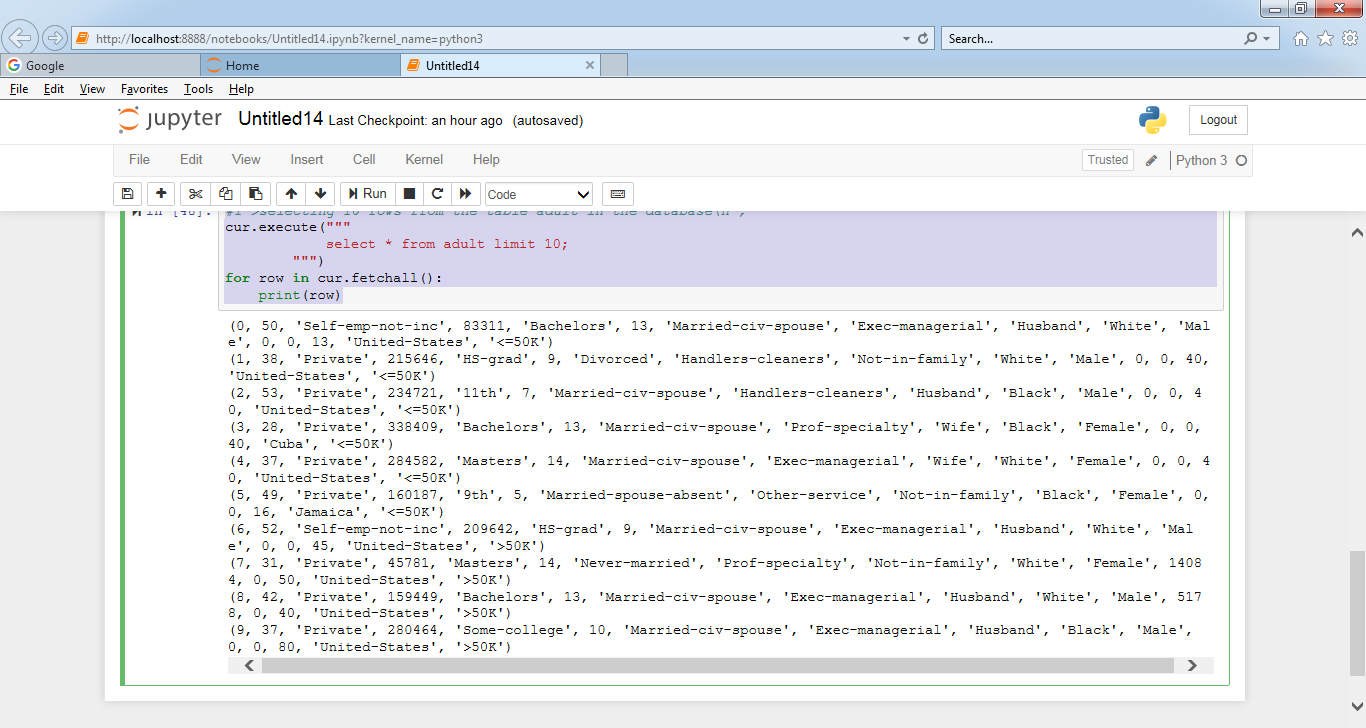
cur.execute("""

select \* from adult limit 10;

""")

for row in cur.fetchall():

print(row)



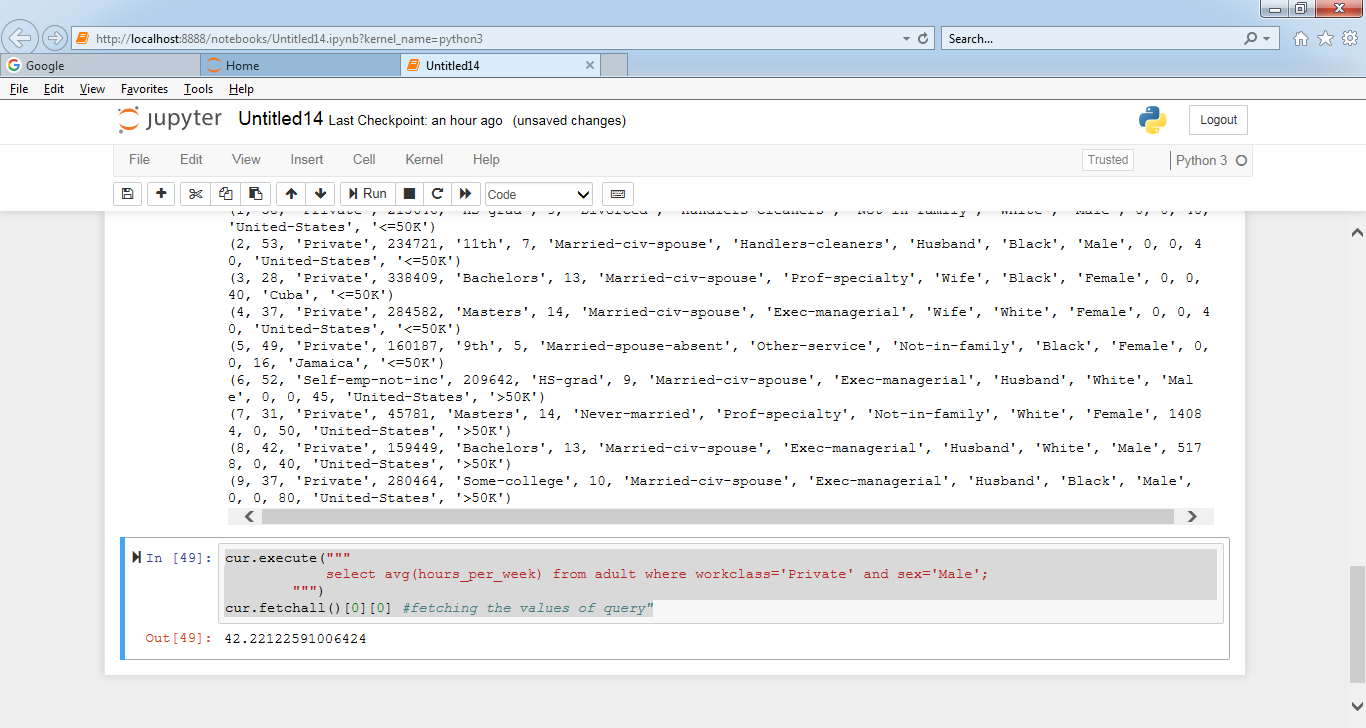
2. Show me the average hours per week of all men who are working in private sector

cur.execute("""

select avg(hours\_per\_week) from adult where workclass='Private' and sex='Male';

""")

cur.fetchall()[0][0] #fetching the values of query"



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

cur.execute("""

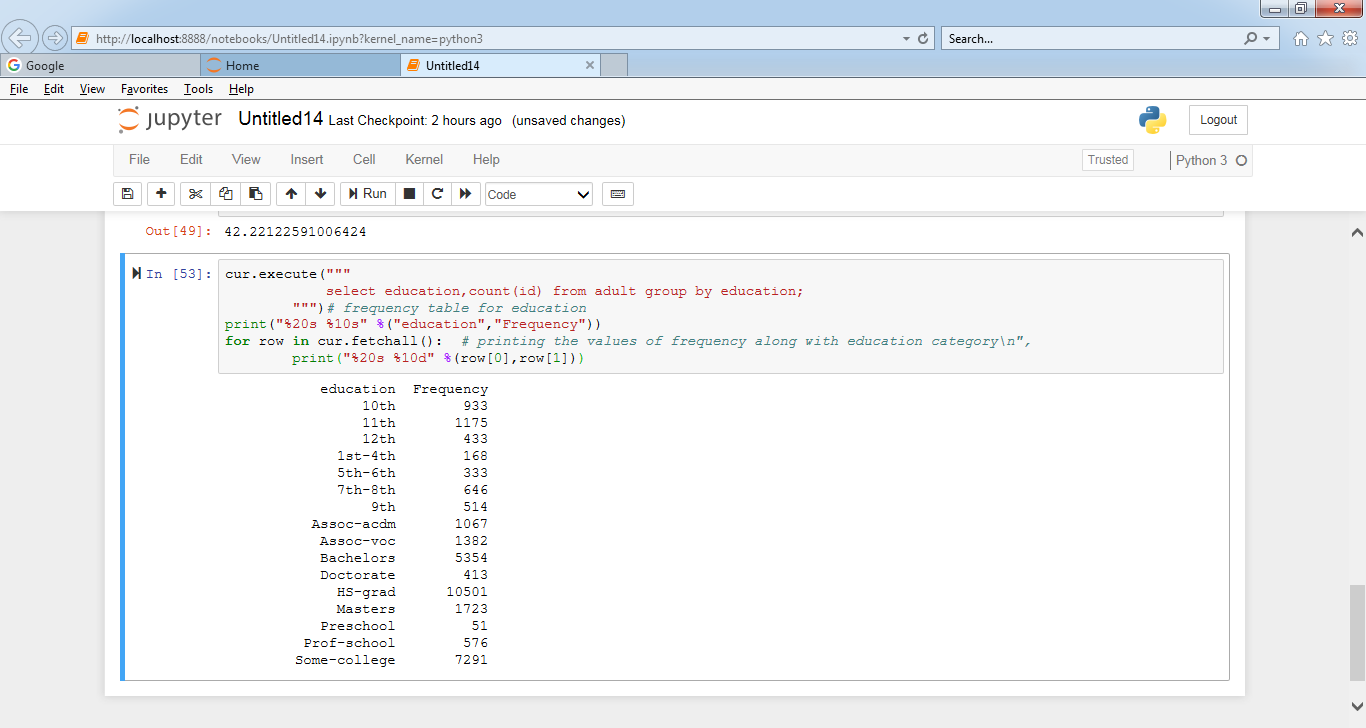
select education,count(id) from adult group by education;

""")# frequency table for education

print("%20s %10s" %("education","Frequency"))

for row in cur.fetchall(): # printing the values of frequency along with education category\n",

print("%20s %10d" %(row[0],row[1]))



cur.execute("""

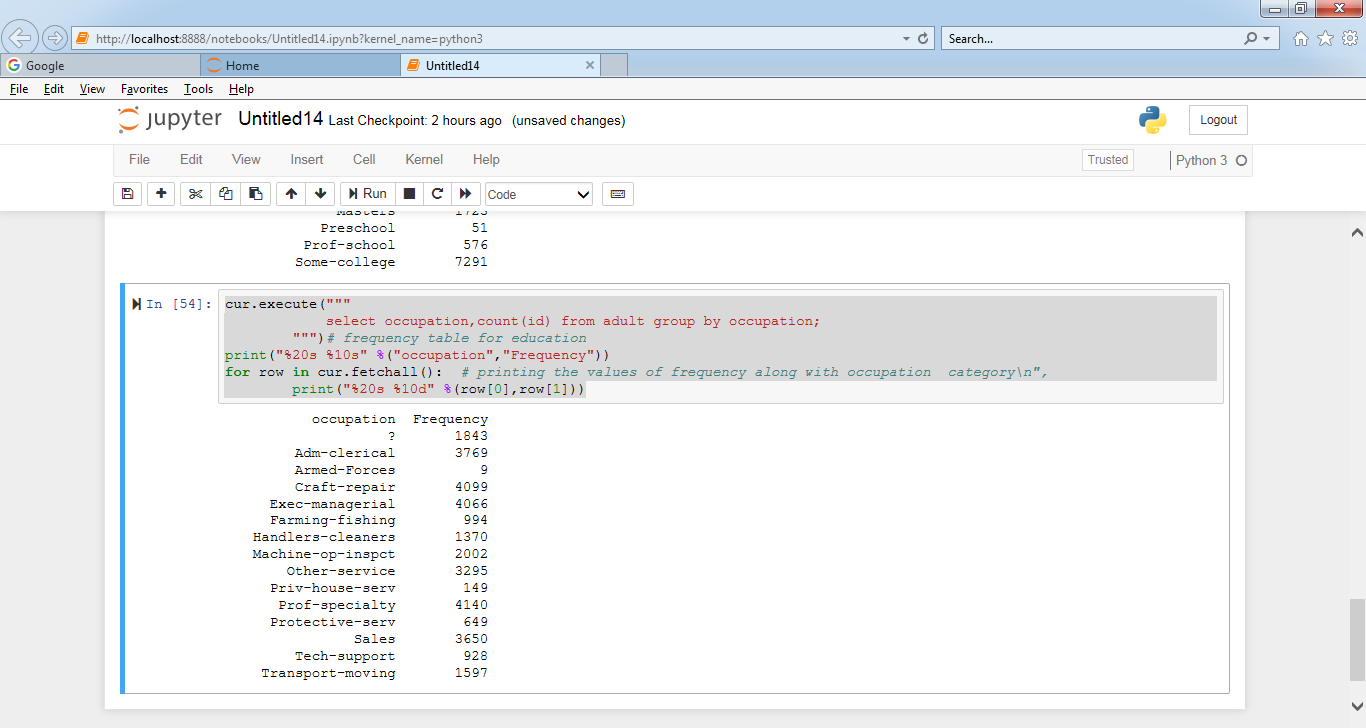
select occupation,count(id) from adult group by occupation;

""")# frequency table for education

print("%20s %10s" %("occupation","Frequency"))

for row in cur.fetchall(): # printing the values of frequency along with occupation category\n",

print("%20s %10d" %(row[0],row[1]))



cur.execute("""

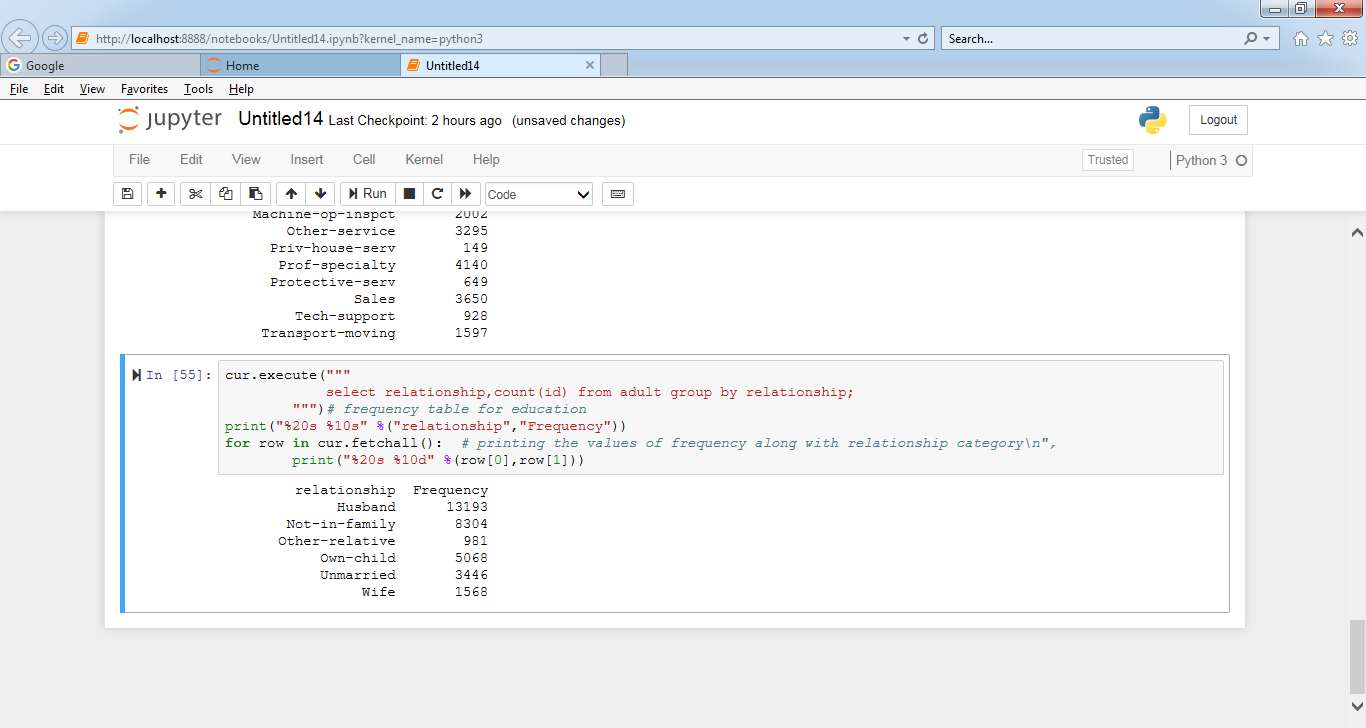
select relationship,count(id) from adult group by relationship;

""")# frequency table for education

print("%20s %10s" %("relationship","Frequency"))

for row in cur.fetchall(): # printing the values of frequency along with relationship category\n",

print("%20s %10d" %(row[0],row[1]))



4. Are there any people who are married, working in private sector and having a masters

Degree

#Are there any people who are married, working in private sector and having a masters degree

cur.execute("""

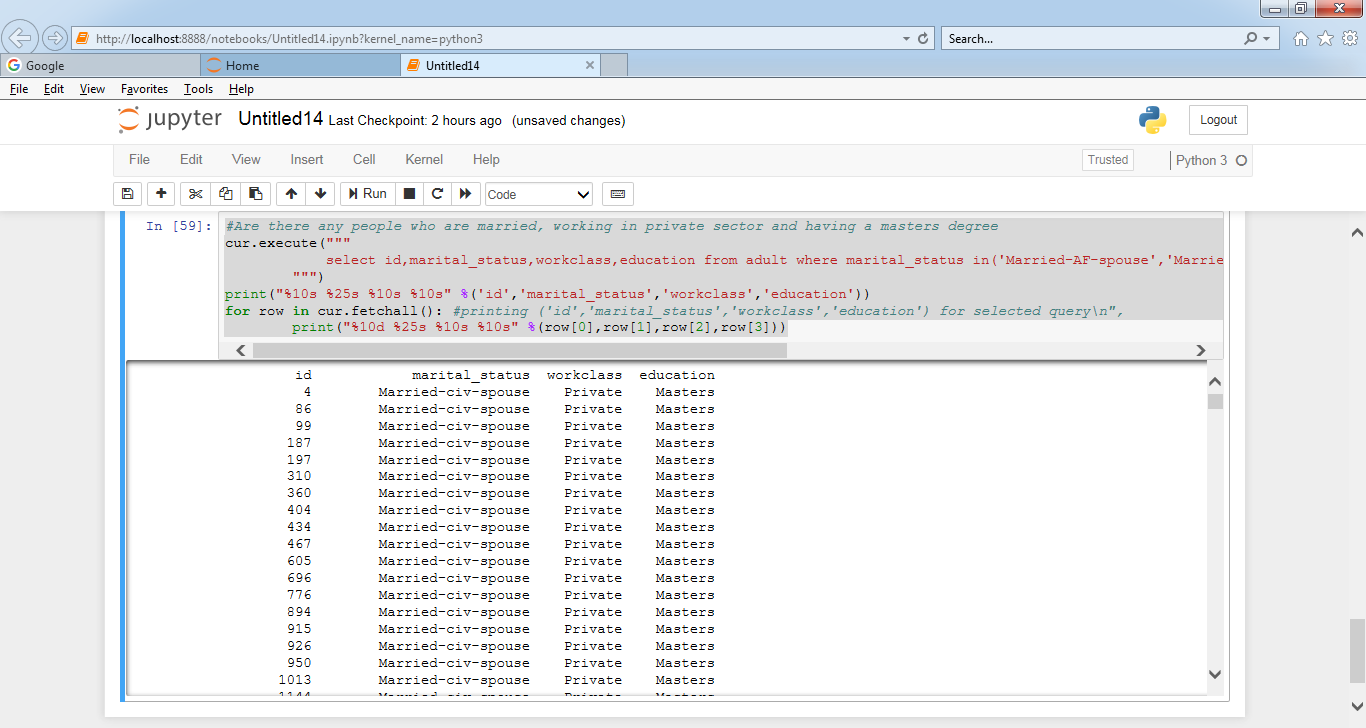
select id,marital\_status,workclass,education from adult where marital\_status in('Married-AF-spouse','Married-civ-spouse','Married-spouse-absent') and workclass=='Private' and education=='Masters';

""")

print("%10s %25s %10s %10s" %('id','marital\_status','workclass','education'))

for row in cur.fetchall(): #printing ('id','marital\_status','workclass','education') for selected query\n",

print("%10d %25s %10s %10s" %(row[0],row[1],row[2],row[3]))



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

different sectors

cur.execute("""

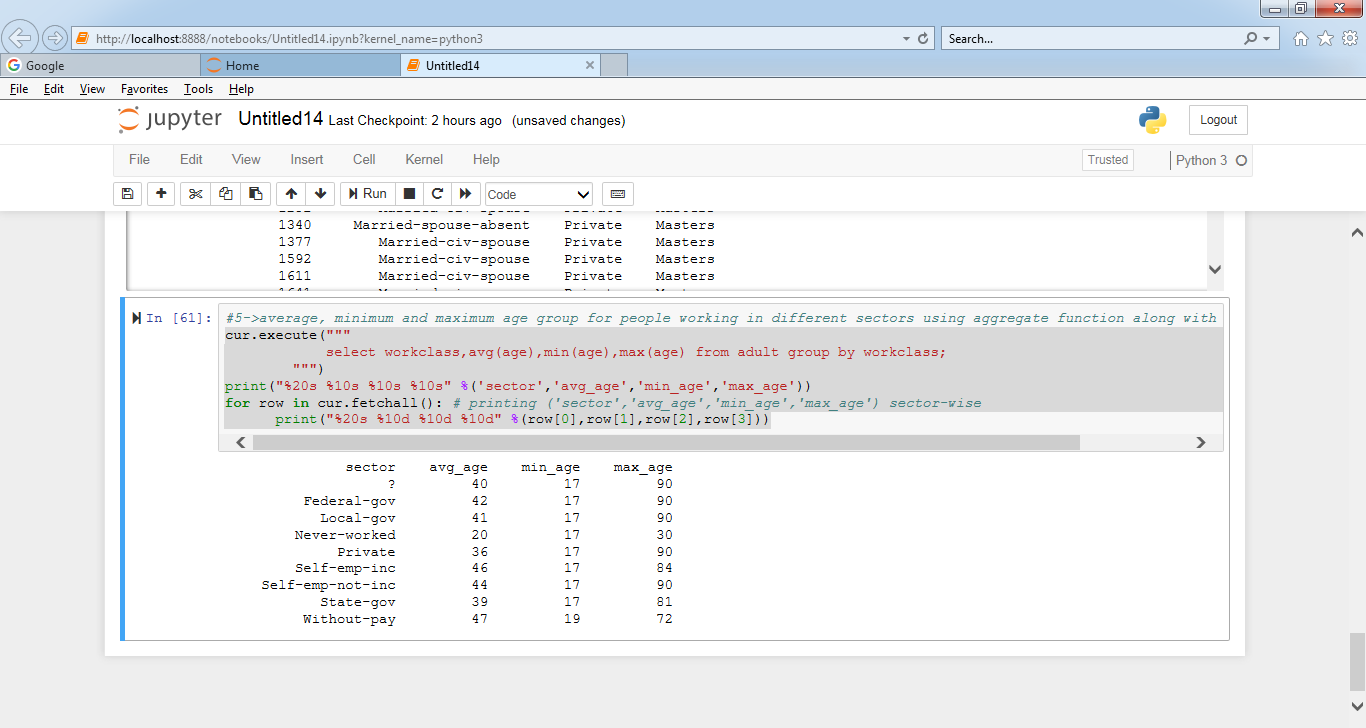
select workclass,avg(age),min(age),max(age) from adult group by workclass;

""")

print("%20s %10s %10s %10s" %('sector','avg\_age','min\_age','max\_age'))

for row in cur.fetchall(): # printing ('sector','avg\_age','min\_age','max\_age') sector-wise

print("%20s %10d %10d %10d" %(row[0],row[1],row[2],row[3]))



6. Calculate age distribution by country

cur.execute("""

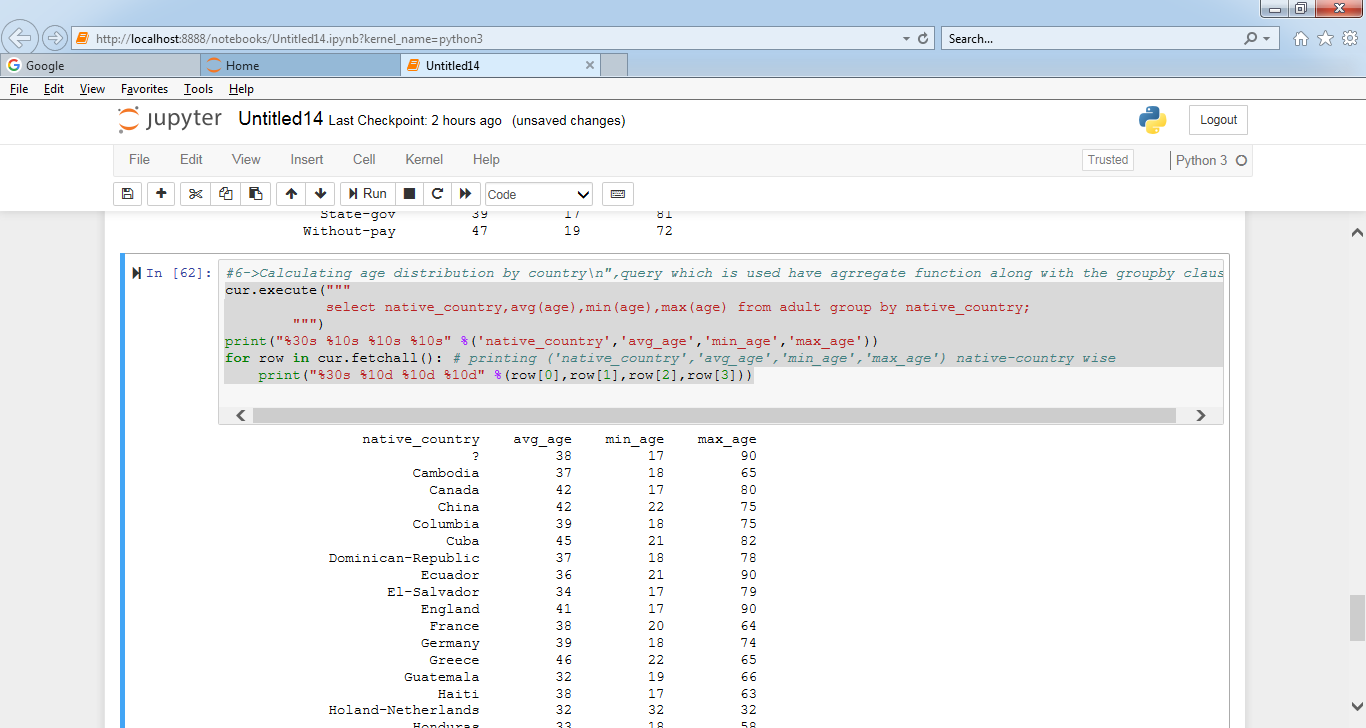
select native\_country,avg(age),min(age),max(age) from adult group by native\_country;

""")

print("%30s %10s %10s %10s" %('native\_country','avg\_age','min\_age','max\_age'))

for row in cur.fetchall(): # printing ('native\_country','avg\_age','min\_age','max\_age') native-country wise

print("%30s %10d %10d %10d" %(row[0],row[1],row[2],row[3]))



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

'capital-loss'

#7->altering the table and adding a new column Net\_Capital\_Gain

cur.execute("""

ALTER TABLE adult ADD COLUMN Net\_Capital\_Gain int AFTER capital\_loss;

""")

conn.commit() #commiting the connection permantly in memory"

#updating adult table set value of Net\_Capital\_Gain = capital\_gain - capital\_loss\n",

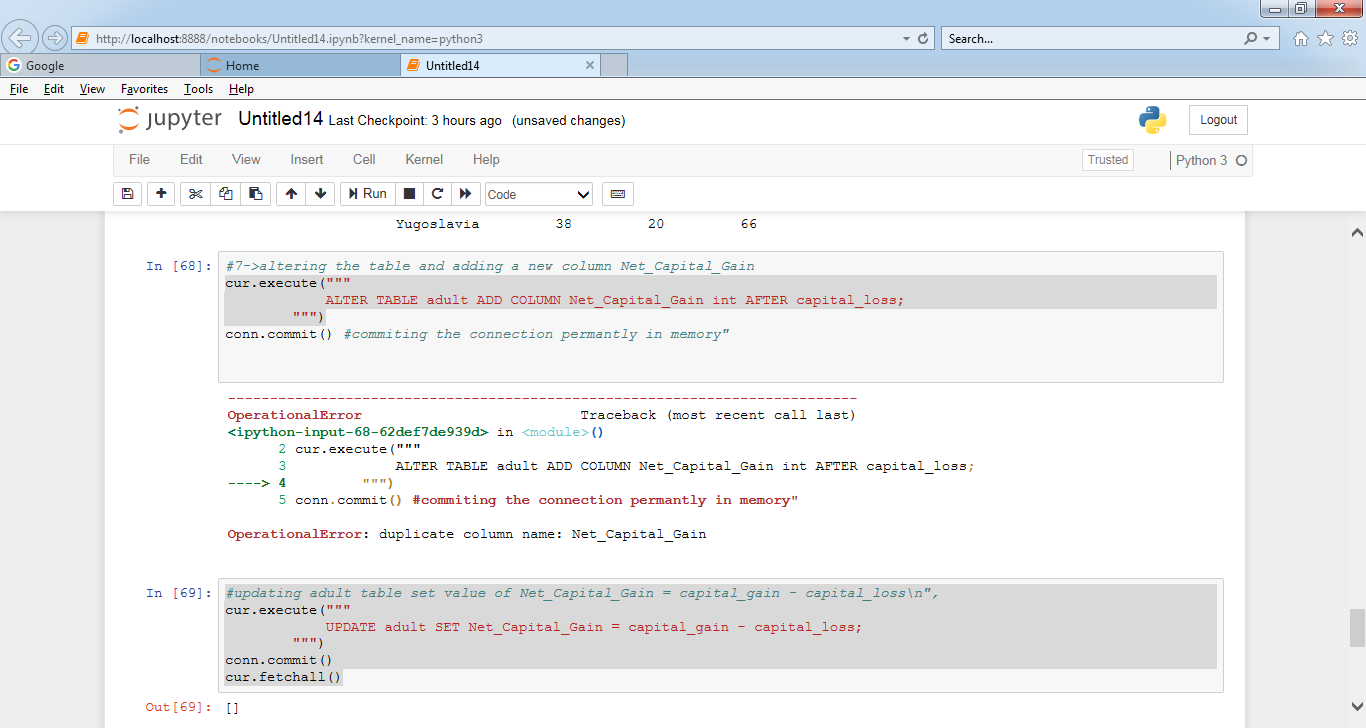
cur.execute("""

UPDATE adult SET Net\_Capital\_Gain = capital\_gain - capital\_loss;

""")

conn.commit()

cur.fetchall()



OperationalError has to ignore here as i hit statement 2 times

cur.execute("""

Select \* from adult;

""")

#print("%30s %10s %10s %10s" %('native\_country','avg\_age','min\_age','max\_age'))

for row in cur.fetchall(): # printing ('native\_country','avg\_age','min\_age','max\_age') native-country wise

print(row)

