**Assignment 3**

**Question 1 (Load a flat file)– CODE**

!pip install pymysql

!pip install mysql

!pip install MySQL-python

#importing the csv file from local computer

import pandas as pd

data = pd.read\_csv("D:/my company/loss iland/DOB\_Job\_Application\_Filings.csv")

#selcting only rows which are required

data1 = data[['Job #', 'Doc #', 'Borough', 'House #', 'Street Name','Job Type', 'Job Status', 'Job Status Descrp','Latest Action Date']]

data1['Job #'].unique() #Checking the unique types in "Job #"

data1['test'] =data1['Job #'].str.isnumeric()

data1[data1['test'] == False] # here we can find the incorrct datatype in Column Job - we will delete this rows from our table

data1 = data1[data1['test'] != False] # removing the noise in the Job #

data2 = data1.drop(['test'],axis=1).head(100)

data2['Job #'] = data2['Job #'].astype(int) #converting Job # into integer type

data2['Latest Action Date'] = data2['Latest Action Date'].apply(pd.to\_datetime) #converting Date in date type

cols = "`,`".join([str(i) for i in data2.columns.tolist()])

data2 = data2.reset\_index(drop=True)

import pymysql

connection = pymysql.connect(host = '34.123.84.120',

user = 'root',

db = 'hw3',

password = 'NO',

cursorclass = pymysql.cursors.DictCursor)

print(connection)

cursor = connection.cursor()

sql1 = "Create table if not exists Job5 (Job int,Doc float,Borough varchar(255),House\_No varchar(255),Street\_name varchar(255),Job\_type varchar(255),Job\_status varchar(255), Job\_Status\_Descrp varchar(255),Latest\_Action\_Date date )"

cursor.execute(sql1)

# Insert DataFrame recrds one by one.

for i,row in data2.iterrows():

sql = "INSERT INTO `Job5` VALUES (" + "%s,"\*(len(row)-1) + "%s)"

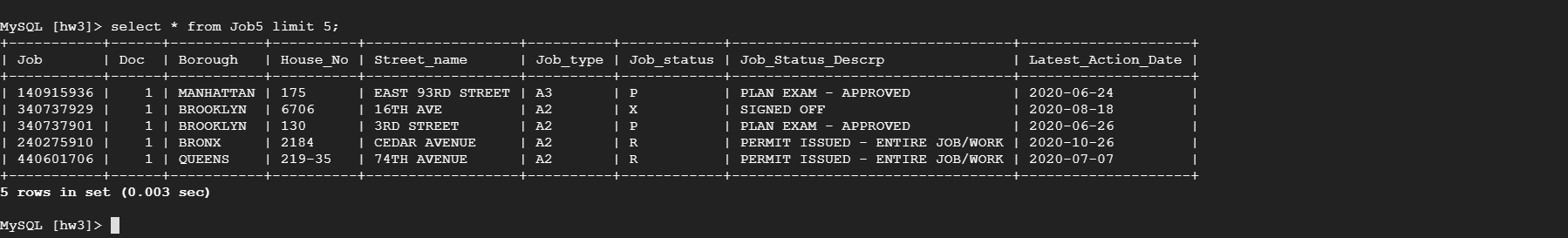
cursor.execute(sql, tuple(row))

# the connection is not autocommitted by default, so we must commit to save our changes

connection.commit()

connection.close()

**OUTPUT OF VM for Question -1**



**Question(load CSV data from an API) -2:**

import requests

import pandas as pd

url = "https://data.cityofnewyork.us/resource/ic3t-wcy2.csv"

req = requests. get(url)

url\_content = req. content

csv\_file = open('downloaded.csv','wb')

csv\_file. write(url\_content)

csv\_file. close()

data = pd.read\_csv('downloaded.csv')

#selcting only rows which are required

data1 = data[['job\_\_', 'doc\_\_', 'borough', 'house\_\_', 'street\_name','job\_type', 'job\_status', 'job\_status\_descrp','latest\_action\_date']]

data2 = data1.head(100)

data2 = data2.reset\_index(drop=True)

import pymysql

connection = pymysql.connect(host = '34.123.84.120',

user = 'root',

db = 'hw3',

password = 'NO',

cursorclass = pymysql.cursors.DictCursor)

print(connection)

cursor = connection.cursor()

sql1 = "Create table if not exists Job\_api1 (Job int,Doc float,Borough varchar(255),House\_No varchar(255),Street\_name varchar(255),Job\_type varchar(255),Job\_status varchar(255), Job\_Status\_Descrp varchar(255),Latest\_Action\_Date date )"

cursor.execute(sql1)

# Insert DataFrame recrds one by one.

for i,row in data2.iterrows():

sql = "INSERT INTO `Job\_api1` VALUES (" + "%s,"\*(len(row)-1) + "%s)"

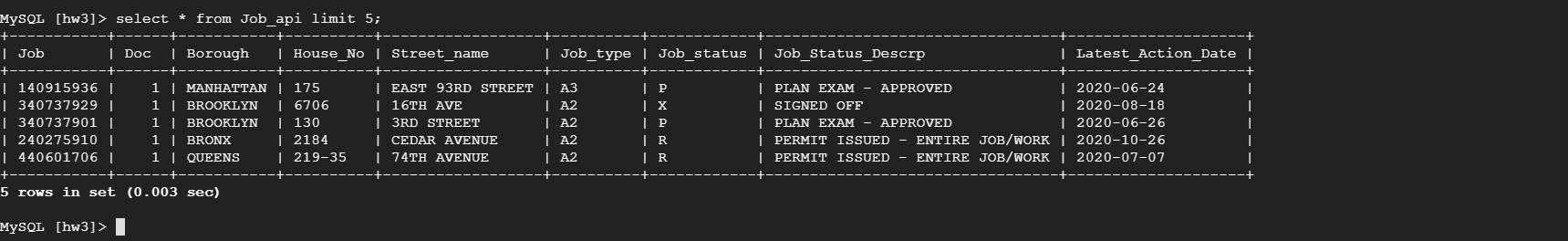
cursor.execute(sql, tuple(row))

# the connection is not autocommitted by default, so we must commit to save our changes

connection.commit()

connection.close()

**Question 2 output I on VM:**



**Question 3 (Clean data, Then load data) :**

import requests

url = "https://data.cityofnewyork.us/resource/ic3t-wcy2.csv"

req = requests. get(url)

url\_content = req. content

csv\_file = open('downloaded.csv','wb')

csv\_file. write(url\_content)

csv\_file. close()

data = pd.read\_csv('downloaded.csv')

#selcting only rows which are required

data1 = data[['job\_\_', 'doc\_\_', 'borough', 'house\_\_', 'street\_name','job\_type', 'job\_status', 'job\_status\_descrp','landmarked','adult\_estab','latest\_action\_date','existing\_occupancy', 'proposed\_occupancy','owner\_type','job\_description','initial\_cost' ]]

data1 = data1.dropna(subset = ['landmarked','adult\_estab','job\_description']).reset\_index()

data1['initial\_cost'] = data1['initial\_cost'].str.replace('$','').astype(float)/10000

data1['latest\_action\_date'] = data1['latest\_action\_date'].apply(pd.to\_datetime) #converting Date in date type

data1 = data1.drop('index',axis=1)

import pymysql

connection = pymysql.connect(host = '34.123.84.120',

user = 'root',

db = 'hw3',

password = 'NO',

cursorclass = pymysql.cursors.DictCursor)

print(connection)

cursor = connection.cursor()

sql1 = """Create table if not exists Job\_cl (Job int,Doc int,Borough varchar(255),House\_No varchar(255),

Street\_name varchar(255),Job\_type varchar(255),Job\_status varchar(255), Job\_Status\_Descrp varchar(255),

landmarked varchar(255), adult\_estab varchar(255) , Latest\_Action\_Date date ,existing\_occupancy varchar(255),

proposed\_occupancy varchar(255),owner\_type varchar(255) ,job\_description varchar(255),initial\_cost float )"""

cursor.execute(sql1)

# Insert DataFrame recrds one by one.

for i,row in data1.iterrows():

sql = "INSERT INTO `Job\_cl` VALUES (" + "%s,"\*(len(row)-1) + "%s)"

cursor.execute(sql, tuple(row))

# the connection is not autocommitted by default, so we must commit to save our changes

connection.commit()

connection.close()

