

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

```
MySQL [hw3]> select * from Job5 limit 5;
```

Job	Doc	Borough	House_No	Street_name	Job_type	Job_status	Job_Status_Descrp	Latest_Action_Date
140915936	1	MANHATTAN	175	EAST 93RD STREET	A3	P	PLAN EXAM - APPROVED	2020-06-24
340737929	1	BROOKLYN	6706	16TH AVE	A2	X	SIGNED OFF	2020-08-18
340737901	1	BROOKLYN	130	3RD STREET	A2	P	PLAN EXAM - APPROVED	2020-06-26
240275910	1	BRONX	2184	CEDAR AVENUE	A2	R	PERMIT ISSUED - ENTIRE JOB/WORK	2020-10-26
440601706	1	QUEENS	219-35	74TH AVENUE	A2	R	PERMIT ISSUED - ENTIRE JOB/WORK	2020-07-07

```
5 rows in set (0.003 sec)
```

```
MySQL [hw3]>
```

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:

```
MySQL [hw3]> select * from Job_api limit 5;
```

Job	Doc	Borough	House_No	Street_name	Job_type	Job_status	Job_Status_Descrp	Latest_Action_Date
140915936	1	MANHATTAN	175	EAST 93RD STREET	A3	P	PLAN EXAM - APPROVED	2020-06-24
340737929	1	BROOKLYN	6706	16TH AVE	A2	X	SIGNED OFF	2020-08-18
340737901	1	BROOKLYN	130	3RD STREET	A2	P	PLAN EXAM - APPROVED	2020-06-26
240275910	1	BRONX	2184	CEDAR AVENUE	A2	R	PERMIT ISSUED - ENTIRE JOB/WORK	2020-10-26
440601706	1	QUEENS	219-35	74TH AVENUE	A2	R	PERMIT ISSUED - ENTIRE JOB/WORK	2020-07-07

```
5 rows in set (0.003 sec)
```

```
MySQL [hw3]>
```

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_descr', '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()

```

```

MySQL [hw3]> select * from Job_cl limit 5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Job   | Doc  | Borough | House_No | 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 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 140915936 | 1 | MANHATTAN | 175 | EAST 93RD STREET | A3 | P | PLAN EXAM - APPROVED | N | N | 2020-06-24 | RES | RES | CORPORATION | APPLICATION FILE TO FACADE REPAIR AS STUCCO WORK AT WEST SIDE ELEVATION AS PER PLAN. NO CHANGE IN USE, EGRESS OR OCCUPANCY UNDER THIS APPLICATON. | 1.5 |
| 340737929 | 1 | BROOKLYN | 6706 | 16TH AVE | A2 | X | SIGNED OFF | N | N | 2020-08-18 | J-3 | J-3 | INDIVIDUAL | INSTALLATION OF SOLAR ENERGY SYSTEM ON ROOF. NO CHANGE OF OCCUPANCY, USE OR EGRESS. | 0.98 |
| 340737901 | 1 | BROOKLYN | 130 | 3RD STREET | A2 | P | PLAN EXAM - APPROVED | N | N | 2020-06-26 | B | B | CORPORATION | CONSTRUCT STEEL DUNNAGE FOR EMERGENCY GENERATOR ON YARD IN RELATION TO DOBNO#800283989. NO CHANGE IN USE, EGRESS AND OCCUPANCY | 0.75 |
| 240275910 | 1 | BRONX | 2184 | CEDAR AVENUE | A2 | R | PERMIT ISSUED - ENTIRE JOB/WORK | N | N | 2020-10-26 | J-3 | J-3 | INDIVIDUAL | FILLING TO OBTAIN PERMITS TO REMOVE STOP WORK ORDER AND TO REMOVE VIOLATION #39022450H ON CONSTRUCTION OF RETAINING WALL AT REAR YARD. | 1.5 |
| 440601706 | 1 | QUEENS | 219-35 | 74TH AVENUE | A2 | R | PERMIT ISSUED - ENTIRE JOB/WORK | N | N | 2020-07-07 | RES | RES | CORPORATION | INSTALLATION OF NEW STAINLESS STEEL CHIMNEY LINER FROM CELLAR THROUGH ROOF AND PAD IN CELLAR.NO CHANGE IN USE,EGRESS,OR OCCUPANCY. | 5 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

MySQL [hw3]>

```