**Project 2 Report**

The following report summarizes the results provided by the developed python programs, sql dadabae and mapreduce excursive requested for Project 2. The code and the text output files are included in the following report and no files were placed under any project folder in the cloud under but the whole project can be accessed by pulling the git hub repository CloudCompFall2014Project2 under the user gonzalo2kx.

Part 1: Relational Theory.

For this exercise a database named solarenergyplant.db was created to contain a full raw table containing all data from the textfile and multiple two column tables containing information constructed in a relational mode. This operation was done on file Part1a.py. Afterwards, 48 tables were created containing the following information to comply with the 5 normalization rules. This operation can be found on Part1b.py and it can be observed that each comment before starting the creation for his table states the data which will be contained in that particular table.

The created tables can be found on the following page:

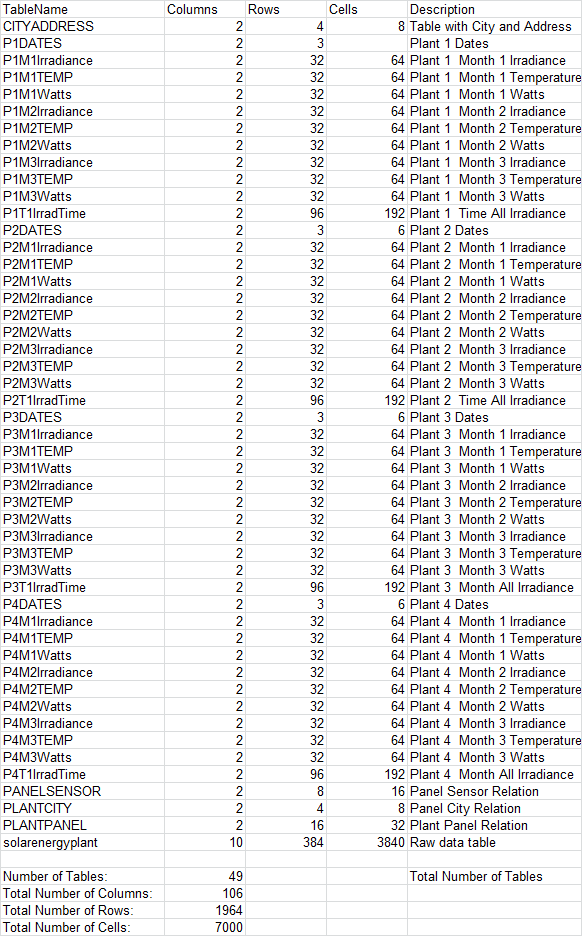
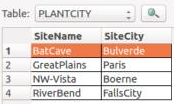
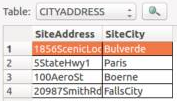
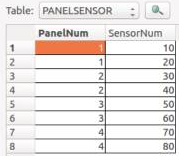
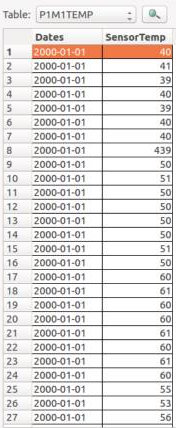
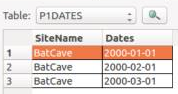


Table Samples:

Part 2: Relational Implementation

For this exercise the data from the created tables (except for the Raw data table) was extracted to provide answers to the questions made. Several methods were used such as Inner Joining tables to filter out results. The code to answer each question can be found in the code files Part2a.py, Part2b.py and Part2c.py each corresponding to questions 1,2,3 from this section.

Question 1 Answer.

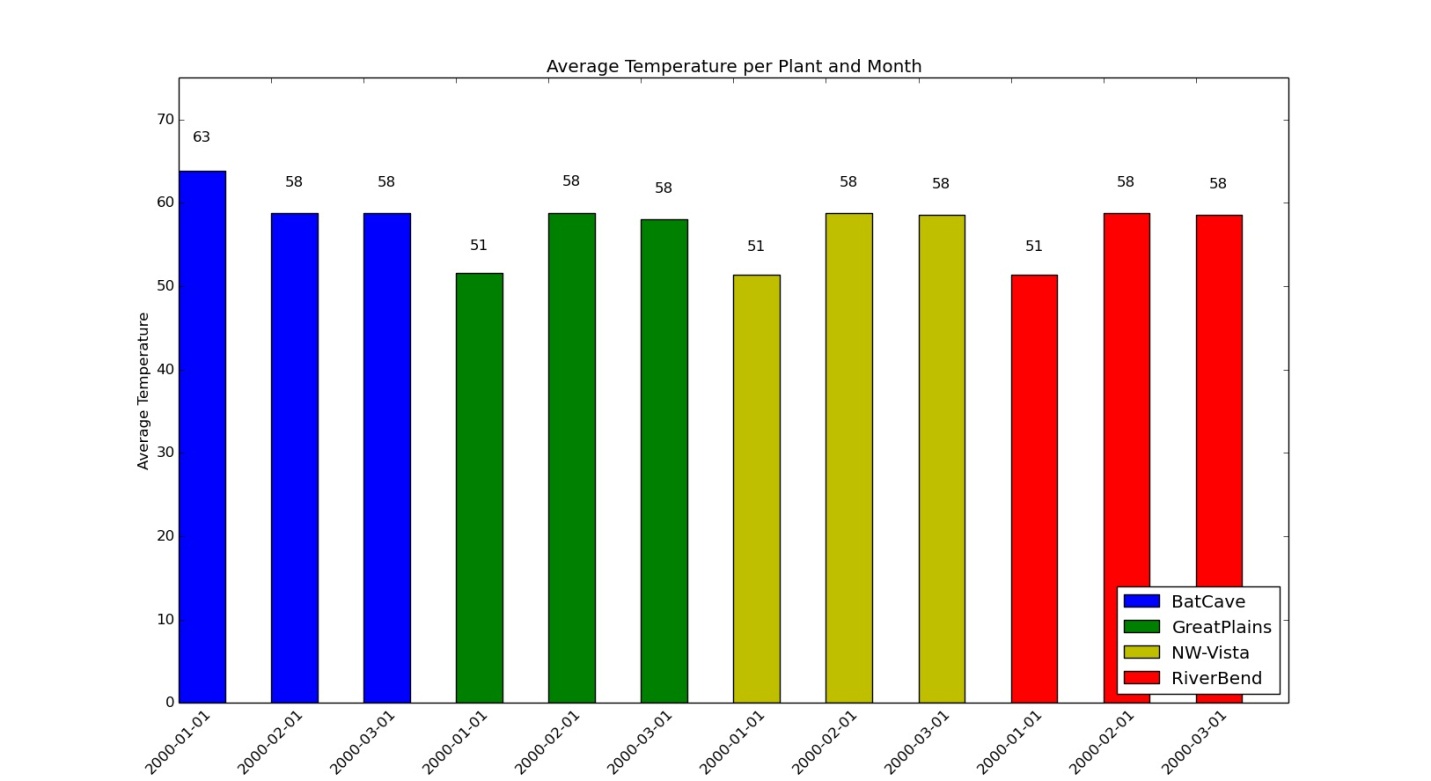


Fig 1. The following figure presents the average temperature per month per plant.



Raw 1. Raw data to compute Fig 1

Question 2 Answer.

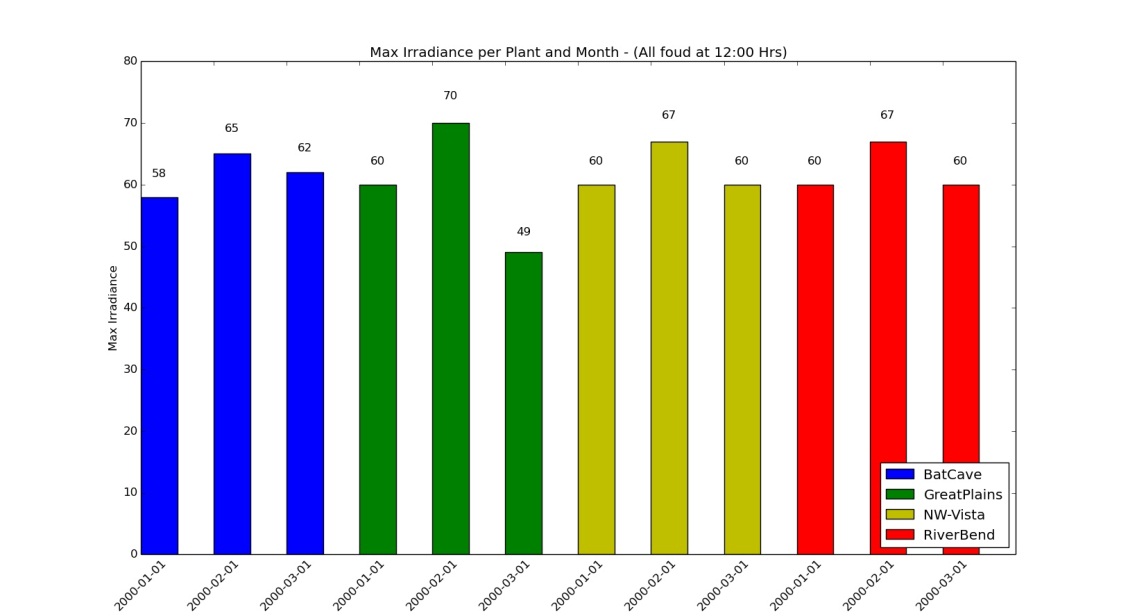
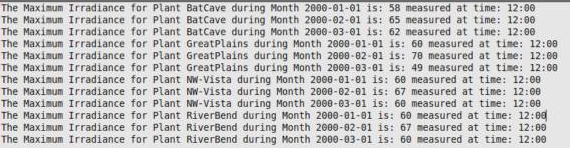


Fig 2. The following figure presents the maximum irradiance per month per plant where all maximum values were found at 12:00 Hours.



Raw 2. Raw data to compute Fig 2

Question 3 Answer.

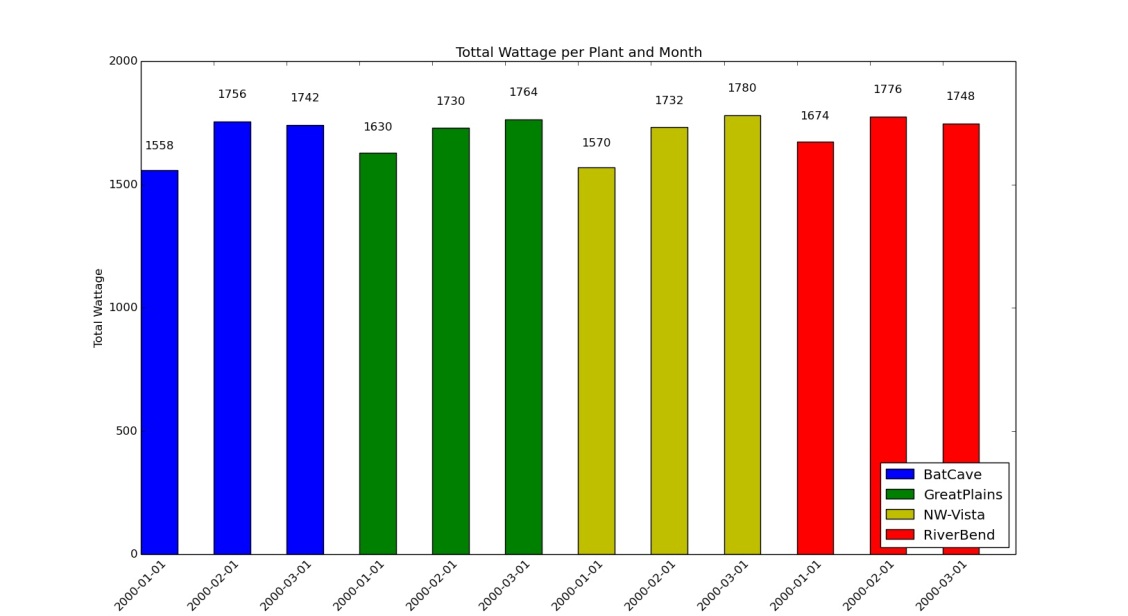


Fig 3. The following figure presents the total wattage produced per month per plant, and since there is only 1 day per month were data is registered; the figure also presents the total wattage per day per plant.



Raw 3. Raw data to compute Fig 3

Part 3: Map/Reduce

For this exercise the data from the solardata.csv file was extracted. Afterwards a mapper was developed to select only two columns (Date and ETR(W/m^2))). On the second section of the program, a reducer was developed to compute the daily ETR average. The code where these computations were made can be found on Part3a.py and they computed results can be seen on the following figure:

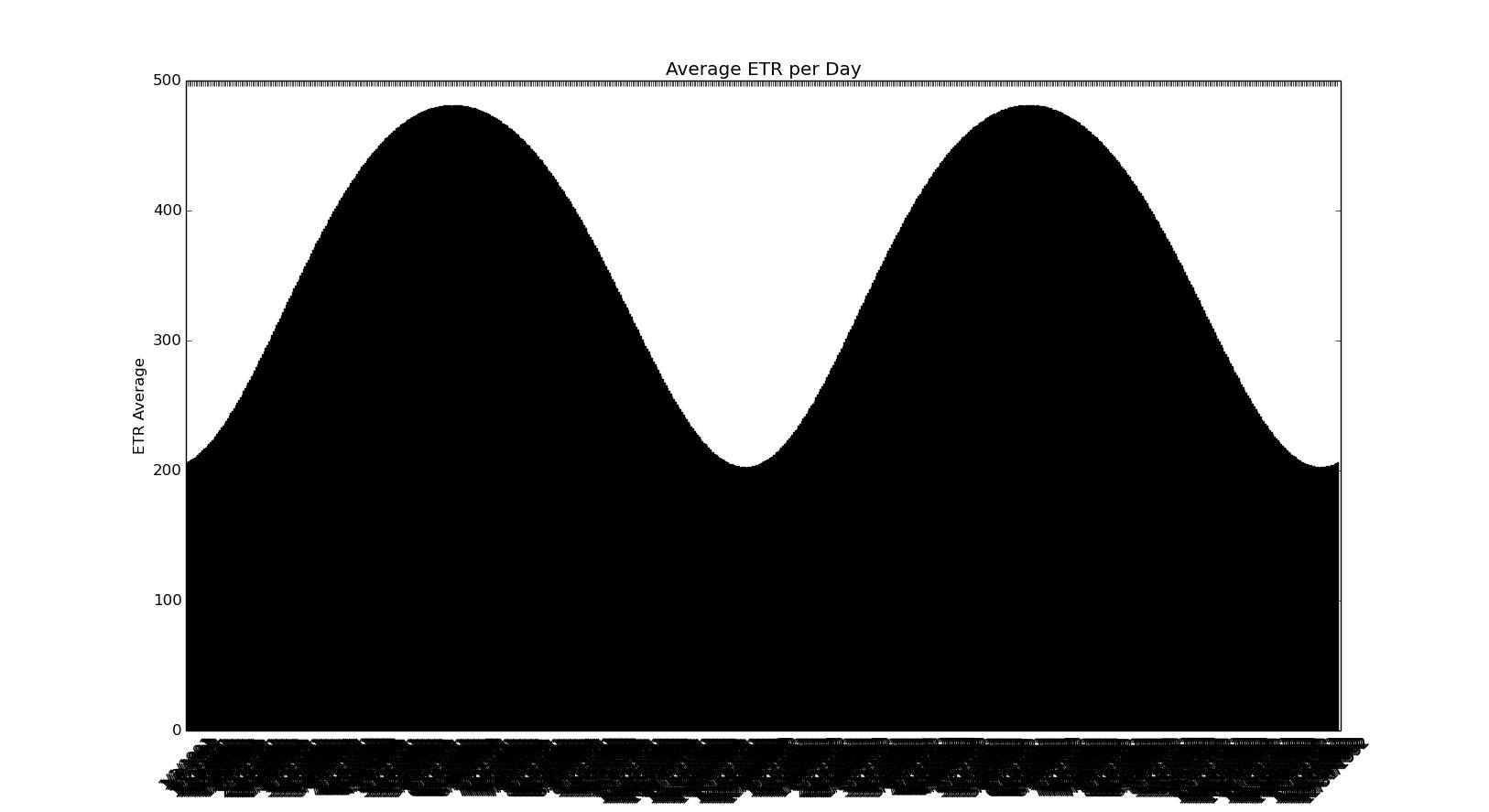


Figure 4. Average ETR per Day (The X label represents the date for each computed average. Unfortunately because of the massive amount of plotted data it can’t be seen clearly)

During the development of this project Mr. Sheriff Lawal and I were sharing ideas on how to solve some of the issues we were encountering. No code was copied from one another. If you have any questions regarding our codes please don’t hesitate to call us at any time.

Sheriff Lawal: (202)8306485

Gonzalo De La Torre: (210)837-0335

**Question 1 – Part1a.py**

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

import csv

#The following section represents the Mapper

text\_file=open("solardata.csv",'r') #Document with data of SolarData

file = open('solardata.csv')

reader = csv.reader(file)

next(reader, None)

dailysamps=24 #Number of Samples per day

#Required variabes

day=[] #Daily date list

etr=[] #Daily ETR list

aetr=[] #Average ETR

detr=[] #Temporal Average ETR

daylist=[] #Final Day List

#Read lines and store day and ETR value in corresponding lists

for line in reader:

day.append(str(line[0]))

etr.append(int(line[4]))

#The following section represents the Reducer which will provide the average

for hours in range(0, len(etr),dailysamps):

detr=np.average(etr[hours:hours+dailysamps])

aetr.append(detr)

daylist.append(day[hours])

print aetr

print daylist

####################################################

#PLOT

####################################################

N = len(aetr) #Total Bars

ind = np.arange(N) #Index per Bar

width = 0.5 #Bar width

fig, ax = plt.subplots()

rects1 = ax.bar(ind, aetr, width, color='r')

#Add X and Y Axis labels and Title

ax.set\_ylabel('ETR Average')

ax.set\_title('Average ETR per Day')

ax.set\_xticks(ind)

ax.set\_xticklabels(daylist, rotation=45)

plt.axis([0, N+1, 0, 500])

#Define Bar Labels

def autolabel(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.05\*height, '%d'%int(height),

ha='center', va='bottom')

#autolabel(rects1)

plt.show()

####################################################

#END - PLOT

####################################################

**Question 1 – Part1b.py**

#The following code creates the tables required for faster computations and

#A better logical structure that will provide all requested data

#Above each section of code that creates each table you will find a comment stating

#In two fields corresponding to Column1/Column2 or

#In Three fields corresponding to Table/Column1/Column2

#Data Tables that has Watts,Temperature and Irradiance data were created from the

#Data table solarenergyplant which was created on Part1a.py

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

conn = sqlite3.connect('solarenergyplant.db')

c=conn.cursor()

#for row in c.execute('SELECT \* FROM solardata'):

# print row

####################################################

#DROP TABLES

####################################################

c.execute("DROP TABLE IF EXISTS PLANTMAIN")

c.execute("DROP TABLE IF EXISTS PLANTCITY")

c.execute("DROP TABLE IF EXISTS PLANTPANEL")

c.execute("DROP TABLE IF EXISTS PANELSENSOR")

c.execute("DROP TABLE IF EXISTS CITYADDRESS")

c.execute("DROP TABLE IF EXISTS P1DATES")

c.execute("DROP TABLE IF EXISTS P2DATES")

c.execute("DROP TABLE IF EXISTS P3DATES")

c.execute("DROP TABLE IF EXISTS P4DATES")

c.execute("DROP TABLE IF EXISTS P1M1TEMP")

c.execute("DROP TABLE IF EXISTS P1M2TEMP")

c.execute("DROP TABLE IF EXISTS P1M3TEMP")

c.execute("DROP TABLE IF EXISTS P2M1TEMP")

c.execute("DROP TABLE IF EXISTS P2M2TEMP")

c.execute("DROP TABLE IF EXISTS P2M3TEMP")

c.execute("DROP TABLE IF EXISTS P3M1TEMP")

c.execute("DROP TABLE IF EXISTS P3M2TEMP")

c.execute("DROP TABLE IF EXISTS P3M3TEMP")

c.execute("DROP TABLE IF EXISTS P4M1TEMP")

c.execute("DROP TABLE IF EXISTS P4M2TEMP")

c.execute("DROP TABLE IF EXISTS P4M3TEMP")

c.execute("DROP TABLE IF EXISTS P1M1Irradiance")

c.execute("DROP TABLE IF EXISTS P1M2Irradiance")

c.execute("DROP TABLE IF EXISTS P1M3Irradiance")

c.execute("DROP TABLE IF EXISTS P2M1Irradiance")

c.execute("DROP TABLE IF EXISTS P2M2Irradiance")

c.execute("DROP TABLE IF EXISTS P2M3Irradiance")

c.execute("DROP TABLE IF EXISTS P3M1Irradiance")

c.execute("DROP TABLE IF EXISTS P3M2Irradiance")

c.execute("DROP TABLE IF EXISTS P3M3Irradiance")

c.execute("DROP TABLE IF EXISTS P4M1Irradiance")

c.execute("DROP TABLE IF EXISTS P4M2Irradiance")

c.execute("DROP TABLE IF EXISTS P4M3Irradiance")

c.execute("DROP TABLE IF EXISTS P1M1Watts")

c.execute("DROP TABLE IF EXISTS P1M2Watts")

c.execute("DROP TABLE IF EXISTS P1M3Watts")

c.execute("DROP TABLE IF EXISTS P2M1Watts")

c.execute("DROP TABLE IF EXISTS P2M2Watts")

c.execute("DROP TABLE IF EXISTS P2M3Watts")

c.execute("DROP TABLE IF EXISTS P3M1Watts")

c.execute("DROP TABLE IF EXISTS P3M2Watts")

c.execute("DROP TABLE IF EXISTS P3M3Watts")

c.execute("DROP TABLE IF EXISTS P4M1Watts")

c.execute("DROP TABLE IF EXISTS P4M2Watts")

c.execute("DROP TABLE IF EXISTS P4M3Watts")

c.execute("DROP TABLE IF EXISTS P1T1IrradTime")

c.execute("DROP TABLE IF EXISTS P2T1IrradTime")

c.execute("DROP TABLE IF EXISTS P3T1IrradTime")

c.execute("DROP TABLE IF EXISTS P4T1IrradTime")

####################################################

#END - DROP TABLES

####################################################

####################################################

#SITE/CITY TABLE

####################################################

c.execute('''CREATE TABLE PLANTCITY (SiteName TEXT primary key,SiteCity TEXT);''')

conn.execute("INSERT INTO PLANTCITY (SiteName,SiteCity) \

VALUES ('BatCave','Bulverde')");

conn.execute("INSERT INTO PLANTCITY (SiteName,SiteCity) \

VALUES ('GreatPlains','Paris')");

conn.execute("INSERT INTO PLANTCITY (SiteName,SiteCity) \

VALUES ('NW-Vista','Boerne')");

conn.execute("INSERT INTO PLANTCITY (SiteName,SiteCity) \

VALUES ('RiverBend','FallsCity')");

conn.commit()

####################################################

#END - SITE/CITY TABLE

####################################################

####################################################

#CITY/ADDRESS TABLE

####################################################

c.execute('''CREATE TABLE CITYADDRESS (SiteAddress TEXT primary key,SiteCity TEXT);''')

conn.execute("INSERT INTO CITYADDRESS (SiteAddress,SiteCity) \

VALUES ('1856ScenicLoop','Bulverde')");

conn.execute("INSERT INTO CITYADDRESS (SiteAddress,SiteCity) \

VALUES ('5StateHwy1','Paris')");

conn.execute("INSERT INTO CITYADDRESS (SiteAddress,SiteCity) \

VALUES ('100AeroSt','Boerne')");

conn.execute("INSERT INTO CITYADDRESS (SiteAddress,SiteCity) \

VALUES ('20987SmithRd.','FallsCity')");

conn.commit()

####################################################

#END - CITY/ADDRESS TABLE

####################################################

####################################################

#SITE/PANEL TABLE

####################################################

c.execute('''CREATE TABLE PLANTPANEL (SiteName TEXT,PanelNum Integer);''')

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('BatCave',1)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('BatCave',2)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('BatCave',3)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('BatCave',4)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('GreatPlains',1)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('GreatPlains',2)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('GreatPlains',3)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('GreatPlains',4)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('NW-Vista',1)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('NW-Vista',2)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('NW-Vista',3)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('NW-Vista',4)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('RiverBend',1)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('RiverBend',2)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('RiverBend',3)");

conn.execute("INSERT INTO PLANTPANEL (SiteName,PanelNum) \

VALUES ('RiverBend',4)");

conn.commit()

####################################################

#END - SITE/PANEL TABLE

####################################################

####################################################

#PANEL/SENSORS TABLE

####################################################

c.execute('''CREATE TABLE PANELSENSOR (PanelNum Integer,SensorNum Integer PRIMARY KEY);''')

conn.execute("INSERT INTO PANELSENSOR(PanelNum,SensorNum) \

VALUES (1,10)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (1,20)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (2,30)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (2,40)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (3,50)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (3,60)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (4,70)");

conn.execute("INSERT INTO PANELSENSOR (PanelNum,SensorNum) \

VALUES (4,80)");

conn.commit()

####################################################

#END - PANEL/SENSORS TABLE

####################################################

####################################################

#PLANTS/DATES TABLES

####################################################

#TABLE FOR PLANT 1 DATES

c.execute('''CREATE TABLE P1DATES(SiteName TEXT,Dates TEXT primary key);''')

conn.execute("INSERT INTO P1DATES (SiteName,Dates) \

VALUES ('BatCave','2000-01-01')");

conn.execute("INSERT INTO P1DATES (SiteName,Dates) \

VALUES ('BatCave','2000-02-01')");

conn.execute("INSERT INTO P1DATES (SiteName,Dates) \

VALUES ('BatCave','2000-03-01')");

conn.commit()

#TABLE FOR PLANT 2 DATES

c.execute('''CREATE TABLE P2DATES(SiteName TEXT,Dates TEXT primary key);''')

conn.execute("INSERT INTO P2DATES (SiteName,Dates) \

VALUES ('GreatPlains','2000-01-01')");

conn.execute("INSERT INTO P2DATES (SiteName,Dates) \

VALUES ('GreatPlains','2000-02-01')");

conn.execute("INSERT INTO P2DATES (SiteName,Dates) \

VALUES ('GreatPlains','2000-03-01')");

conn.commit()

#TABLE FOR PLANT 3 DATES

c.execute('''CREATE TABLE P3DATES(SiteName TEXT,Dates TEXT primary key);''')

conn.execute("INSERT INTO P3DATES (SiteName,Dates) \

VALUES ('NW-Vista','2000-01-01')");

conn.execute("INSERT INTO P3DATES (SiteName,Dates) \

VALUES ('NW-Vista','2000-02-01')");

conn.execute("INSERT INTO P3DATES (SiteName,Dates) \

VALUES ('NW-Vista','2000-03-01')");

conn.commit()

#TABLE FOR PLANT 4 DATES

c.execute('''CREATE TABLE P4DATES(SiteName TEXT,Dates TEXT primary key);''')

conn.execute("INSERT INTO P4DATES (SiteName,Dates) \

VALUES ('RiverBend','2000-01-01')");

conn.execute("INSERT INTO P4DATES (SiteName,Dates) \

VALUES ('RiverBend','2000-02-01')");

conn.execute("INSERT INTO P4DATES (SiteName,Dates) \

VALUES ('RiverBend','2000-03-01')");

conn.commit()

####################################################

#END - PLANTS/DATES TABLES

####################################################

####################################################

#PLANTS/DATES/TEMPERAURE TABLES

####################################################

#TABLE FOR PLANT 1 DATE 1 Temperature

tplant=('BatCave',)

c.execute('''CREATE TABLE P1M1TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P1M1TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#c.execute('SELECT \* FROM P1M1TEMP ORDER BY SensorTemp')

#print c.fetchall()

#TABLE FOR PLANT 1 DATE 2 Temperature

c.execute('''CREATE TABLE P1M2TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P1M2TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 1 DATE 3 Temperature

c.execute('''CREATE TABLE P1M3TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P1M3TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 2 DATE 1 Temperature

tplant=('GreatPlains',)

c.execute('''CREATE TABLE P2M1TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P2M1TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 2 DATE 2 Temperature

c.execute('''CREATE TABLE P2M2TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P2M2TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 2 DATE 3 Temperature

c.execute('''CREATE TABLE P2M3TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P2M3TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 3 DATE 1 Temperature

tplant=('NW-Vista',)

c.execute('''CREATE TABLE P3M1TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P3M1TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 3 DATE 2 Temperature

c.execute('''CREATE TABLE P3M2TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P3M2TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 3 DATE 3 Temperature

c.execute('''CREATE TABLE P3M3TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P3M3TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 4 DATE 1 Temperature

tplant=('RiverBend',)

c.execute('''CREATE TABLE P4M1TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P4M1TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 4 DATE 2 Temperature

c.execute('''CREATE TABLE P4M2TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P4M2TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

#TABLE FOR PLANT 4 DATE 3 Temperature

c.execute('''CREATE TABLE P4M3TEMP(Dates TEXT,SensorTemp INTEGER);''')

c.execute('SELECT Dates,SensorTemp FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, temptemperature = row

conn.execute("INSERT INTO P4M3TEMP (Dates,SensorTemp) \

VALUES (?,?);",(tempdates, temptemperature));

conn.commit()

####################################################

#END - PLANTS/DATES/TEMPERAURE TABLES

####################################################

####################################################

#PLANTS/DATES/IRRADIANCE TABLES

####################################################

#TABLE FOR PLANT 1 DATE 1 irradiance

tplant=('BatCave',)

c.execute('''CREATE TABLE P1M1Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P1M1Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#c.execute('SELECT \* FROM P1M1Irradiance ORDER BY SensorIrradiance')

#print c.fetchall()

#TABLE FOR PLANT 1 DATE 2 irradiance

c.execute('''CREATE TABLE P1M2Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P1M2Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 1 DATE 3 irradiance

c.execute('''CREATE TABLE P1M3Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P1M3Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 2 DATE 1 irradiance

tplant=('GreatPlains',)

c.execute('''CREATE TABLE P2M1Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P2M1Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 2 DATE 2 irradiance

c.execute('''CREATE TABLE P2M2Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P2M2Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 2 DATE 3 irradiance

c.execute('''CREATE TABLE P2M3Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P2M3Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 3 DATE 1 irradiance

tplant=('NW-Vista',)

c.execute('''CREATE TABLE P3M1Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P3M1Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 3 DATE 2 irradiance

c.execute('''CREATE TABLE P3M2Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P3M2Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 3 DATE 3 irradiance

c.execute('''CREATE TABLE P3M3Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P3M3Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 4 DATE 1 irradiance

tplant=('RiverBend',)

c.execute('''CREATE TABLE P4M1Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P4M1Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 4 DATE 2 irradiance

c.execute('''CREATE TABLE P4M2Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P4M2Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

#TABLE FOR PLANT 4 DATE 3 irradiance

c.execute('''CREATE TABLE P4M3Irradiance(Dates TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Dates,SensorIrradiance FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradiance = row

conn.execute("INSERT INTO P4M3Irradiance (Dates,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradiance));

conn.commit()

####################################################

#END - PLANTS/DATES/IRRADIANCE TABLES

####################################################

####################################################

#PLANTS/DATES/IRRADIANCE-TIME TABLES

####################################################

#TABLE FOR PLANT 1 TIME irradiance

tplant=('BatCave',)

c.execute('''CREATE TABLE P1T1IrradTime(Time TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Time,SensorIrradiance FROM solarenergyplant WHERE SiteName=?',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradtime = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P1T1IrradTime (Time,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradtime));

conn.commit()

#c.execute('SELECT \* FROM P1T1IrradTime')

#print c.fetchall()

#TABLE FOR PLANT 2 time irradiance

tplant=('GreatPlains',)

c.execute('''CREATE TABLE P2T1IrradTime(Time TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Time,SensorIrradiance FROM solarenergyplant WHERE SiteName=?',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradtime = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P2T1IrradTime (Time,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradtime));

conn.commit()

#c.execute('SELECT \* FROM P2T1IrradTime')

#print c.fetchall()

#TABLE FOR PLANT 3 time irradiance

tplant=('NW-Vista',)

c.execute('''CREATE TABLE P3T1IrradTime(Time TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Time,SensorIrradiance FROM solarenergyplant WHERE SiteName=?',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradtime = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P3T1IrradTime (Time,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradtime));

conn.commit()

#c.execute('SELECT \* FROM P3T1IrradTime')

#print c.fetchall()

#TABLE FOR PLANT 4 time irradiance

tplant=('RiverBend',)

c.execute('''CREATE TABLE P4T1IrradTime(Time TEXT,SensorIrradiance INTEGER);''')

c.execute('SELECT Time,SensorIrradiance FROM solarenergyplant WHERE SiteName=?',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempirradtime = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P4T1IrradTime (Time,SensorIrradiance) \

VALUES (?,?);",(tempdates, tempirradtime));

conn.commit()

#c.execute('SELECT \* FROM P4T1IrradTime')

#print c.fetchall()

####################################################

#END - PLANTS/DATES/IRRADIANCE TABLES

####################################################

####################################################

#PLANTS/DATES/WATTS TABLES

####################################################

#TABLE FOR PLANT 1 DATE 1 Watts

tplant=('BatCave',)

c.execute('''CREATE TABLE P1M1Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

#print('{0} , {1}'.format(row[0], row[1]))

conn.execute("INSERT INTO P1M1Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#c.execute('SELECT \* FROM P1M1Watts ORDER BY PanelWatts')

#print c.fetchall()

#TABLE FOR PLANT 1 DATE 2 Watts

c.execute('''CREATE TABLE P1M2Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P1M2Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 1 DATE 3 Watts

c.execute('''CREATE TABLE P1M3Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P1M3Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 2 DATE 1 Watts

tplant=('GreatPlains',)

c.execute('''CREATE TABLE P2M1Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P2M1Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 2 DATE 2 Watts

c.execute('''CREATE TABLE P2M2Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P2M2Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 2 DATE 3 Watts

c.execute('''CREATE TABLE P2M3Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P2M3Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 3 DATE 1 Watts

tplant=('NW-Vista',)

c.execute('''CREATE TABLE P3M1Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P3M1Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 3 DATE 2 Watts

c.execute('''CREATE TABLE P3M2Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P3M2Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 3 DATE 3 Watts

c.execute('''CREATE TABLE P3M3Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P3M3Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 4 DATE 1 Watts

tplant=('RiverBend',)

c.execute('''CREATE TABLE P4M1Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-01-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P4M1Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 4 DATE 2 Watts

c.execute('''CREATE TABLE P4M2Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-02-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P4M2Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

#TABLE FOR PLANT 4 DATE 3 Watts

c.execute('''CREATE TABLE P4M3Watts(Dates TEXT,PanelWatts INTEGER);''')

c.execute('SELECT Dates,PanelWatts FROM solarenergyplant WHERE SiteName=? AND Dates="2000-03-01"',tplant)

all\_rows = c.fetchall()

for row in all\_rows:

tempdates, tempWatts = row

conn.execute("INSERT INTO P4M3Watts (Dates,PanelWatts) \

VALUES (?,?);",(tempdates, tempWatts));

conn.commit()

####################################################

#END - PLANTS/DATES/WATTS TABLES

####################################################

c.execute("SELECT name FROM sqlite\_master WHERE type='table';")

print(c.fetchall())

conn.close()

**Question 2 – Part2a.py**

#The following code provides the answers for the first question in part 2 by

#Retrieving the information from the previously created data tables on Part1b.py

#Average operation is done outside of the SQL statement for demonstration purposes

#On data extraction and manipulation using python

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

conn = sqlite3.connect('solarenergyplant.db')

c=conn.cursor()

####################################################

#Average Temperature for Plant 'BatCave'

####################################################

P1avgTemp=[] #Average Temperature Collector

P1Months=[] #Month Collector

#Temperature Average Computation for Plant BatCave during Month 2000-01-01

c.execute('SELECT SensorTemp FROM P1M1TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

tplant='BatCave' #Plant Name

Dates="2000-01-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P1avgTemp.append(temperatureavg) #Average Temperature Collector

P1Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant BatCave during Month 2000-02-01

c.execute('SELECT SensorTemp FROM P1M2TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-02-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P1avgTemp.append(temperatureavg) #Average Temperature Collector

P1Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant BatCave during Month 2000-03-01

c.execute('SELECT SensorTemp FROM P1M3TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-03-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P1avgTemp.append(temperatureavg) #Average Temperature Collector

P1Months.append(Dates) #Month Collector

####################################################

#End - Average Temperature for Plant 'BatCave'

####################################################

#print P1avgTemp #Average Temperature Collector

#print P1Months #Month Collector

####################################################

#Average Temperature for Plant 'GreatPlains'

####################################################

P2avgTemp=[] #Average Temperature Collector

P2Months=[] #Month Collector

#Temperature Average Computation for Plant GreatPlains during Month 2000-01-01

c.execute('SELECT SensorTemp FROM P2M1TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

tplant='GreatPlains' #Plant Name

Dates="2000-01-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P2avgTemp.append(temperatureavg) #Average Temperature Collector

P2Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant GreatPlains during Month 2000-02-01

c.execute('SELECT SensorTemp FROM P2M2TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-02-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P2avgTemp.append(temperatureavg) #Average Temperature Collector

P2Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant GreatPlains during Month 2000-03-01

c.execute('SELECT SensorTemp FROM P2M3TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-03-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P2avgTemp.append(temperatureavg) #Average Temperature Collector

P2Months.append(Dates) #Month Collector

####################################################

#End - Average Temperature for Plant 'GreatPlains'

####################################################

#print P2avgTemp #Average Temperature Collector

#print P2Months #Month Collector

####################################################

#Average Temperature for Plant 'NW-Vista'

####################################################

P3avgTemp=[] #Average Temperature Collector

P3Months=[] #Month Collector

#Temperature Average Computation for Plant NW-Vista during Month 2000-01-01

c.execute('SELECT SensorTemp FROM P3M1TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

tplant='NW-Vista' #Plant Name

Dates="2000-01-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P3avgTemp.append(temperatureavg) #Average Temperature Collector

P3Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant NW-Vista during Month 2000-02-01

c.execute('SELECT SensorTemp FROM P3M2TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-02-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P3avgTemp.append(temperatureavg) #Average Temperature Collector

P3Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant NW-Vista during Month 2000-03-01

c.execute('SELECT SensorTemp FROM P3M3TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-03-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P3avgTemp.append(temperatureavg) #Average Temperature Collector

P3Months.append(Dates) #Month Collector

####################################################

#End - Average Temperature for Plant 'NW-Vista'

####################################################

#print P3avgTemp #Average Temperature Collector

#print P3Months #Month Collector

####################################################

#Average Temperature for Plant 'RiverBend'

####################################################

P4avgTemp=[] #Average Temperature Collector

P4Months=[] #Month Collector

#Temperature Average Computation for Plant RiverBend during Month 2000-01-01

c.execute('SELECT SensorTemp FROM P4M1TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

tplant='RiverBend' #Plant Name

Dates="2000-01-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P4avgTemp.append(temperatureavg) #Average Temperature Collector

P4Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant RiverBend during Month 2000-02-01

c.execute('SELECT SensorTemp FROM P4M2TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-02-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P4avgTemp.append(temperatureavg) #Average Temperature Collector

P4Months.append(Dates) #Month Collector

#Temperature Average Computation for Plant RiverBend during Month 2000-03-01

c.execute('SELECT SensorTemp FROM P4M3TEMP') #Retrieve Temperature from Table

all\_rows = c.fetchall() #Get Values from Table

numelements=0 #Initialize Values

temperaturesum=0 #Initialize Values

for row in all\_rows:

temptemperature = int(''.join(map(str,row)))

temperaturesum=temperaturesum+temptemperature #Add Temperature Values

numelements=numelements+1 #Add Increment

Dates="2000-03-01" #Current Date

temperatureavg=float(temperaturesum)/float(numelements) #Average Calculation

print "The Average temperature for Plant %s during Month %s is: %f" %(tplant,Dates,temperatureavg)

P4avgTemp.append(temperatureavg) #Average Temperature Collector

P4Months.append(Dates) #Month Collector

####################################################

#End - Average Temperature for Plant 'RiverBend'

####################################################

#print P4avgTemp #Average Temperature Collector

#print P4Months #Month Collector

conn.close()

####################################################

#PLOT

####################################################

Tmonths=P1Months+P2Months+P3Months+P4Months

N = 3 #Bars per Plant

N2= 12 #Total X Labels

ind = np.arange(N) #Bars per Plant

ind2= np.arange(N2) #Total X Labels

width = 0.5 #Bar width

fig, ax = plt.subplots()

rects1 = ax.bar(ind, P1avgTemp, width, color='b')

rects2 = ax.bar(ind+3, P2avgTemp, width, color='g')

rects3 = ax.bar(ind+6, P3avgTemp, width, color='y')

rects4 = ax.bar(ind+9, P4avgTemp, width, color='r')

#Add X and Y Axis labels and Title

ax.set\_ylabel('Average Temperature')

ax.set\_title('Average Temperature per Plant and Month')

ax.set\_xticks(ind2)

ax.set\_xticklabels(Tmonths, rotation=45)

#Define Legend

ax.legend( (rects1[0],rects2[0],rects3[0],rects4[0]), ('BatCave','GreatPlains','NW-Vista','RiverBend') ,loc=4)

plt.axis([0, 12, 0, 75])

#Define Bar Labels

def autolabel(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.05\*height, '%d'%int(height),

ha='center', va='bottom')

autolabel(rects1)

autolabel(rects2)

autolabel(rects3)

autolabel(rects4)

plt.show()

####################################################

#END - PLOT

####################################################

**Question 2 – Part2b.py**

#The following code provides the answers for the second question in part 2 by

#Inner Joining and retrieving the information from the previously created data tables on Part1b.py

#Max operation is done inside of the SQL statement for demonstration purposes

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

conn = sqlite3.connect('solarenergyplant.db')

c=conn.cursor()

####################################################################

#Maximum Irradiance per Plant per Month for Plant 'BatCave'

####################################################################

P1maxirrad=[] #Max Irradiance Collector

P1Months=[] #Month Collector

P1time=[] #Time Collector

#Max Irradiance for Plant BatCave with Corresponding Date(2000-01-01) and Time

tplant='BatCave' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P1M1Irradiance x INNER JOIN P1T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P1maxirrad.append(tempirrad1) #Max Irradiance Collector

P1Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P1time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant BatCave with Corresponding Date(2000-02-01) and Time

tplant='BatCave' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P1M2Irradiance x INNER JOIN P1T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P1maxirrad.append(tempirrad1) #Max Irradiance Collector

P1Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P1time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant BatCave with Corresponding Date(2000-03-01) and Time

tplant='BatCave' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P1M3Irradiance x INNER JOIN P1T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P1maxirrad.append(tempirrad1) #Max Irradiance Collector

P1Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P1time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

####################################################################

#END - Maximum Irradiance per Plant per Month for Plant 'BatCave'

####################################################################

#print P1maxirrad #Max Irradiance Collector

#print P1Months #Month Collector

#print P1time #Time Collector

####################################################################

#Maximum Irradiance per Plant per Month for Plant 'GreatPlains'

####################################################################

P2maxirrad=[] #Max Irradiance Collector

P2Months=[] #Month Collector

P2time=[] #Time Collector

#Max Irradiance for Plant GreatPlains with Corresponding Date(2000-01-01) and Time

tplant='GreatPlains' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P2M1Irradiance x INNER JOIN P2T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P2maxirrad.append(tempirrad1) #Max Irradiance Collector

P2Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P2time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant GreatPlains with Corresponding Date(2000-02-01) and Time

tplant='GreatPlains' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P2M2Irradiance x INNER JOIN P2T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P2maxirrad.append(tempirrad1) #Max Irradiance Collector

P2Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P2time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant GreatPlains with Corresponding Date(2000-03-01) and Time

tplant='GreatPlains' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P2M3Irradiance x INNER JOIN P2T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P2maxirrad.append(tempirrad1) #Max Irradiance Collector

P2Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P2time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

####################################################################

#END - Maximum Irradiance per Plant per Month for Plant 'GreatPlains'

####################################################################

#print P2maxirrad #Max Irradiance Collector

#print P2Months #Month Collector

#print P2time #Time Collector

####################################################################

#Maximum Irradiance per Plant per Month for Plant 'NW-Vista'

####################################################################

P3maxirrad=[] #Max Irradiance Collector

P3Months=[] #Month Collector

P3time=[] #Time Collector

#Max Irradiance for Plant NW-Vista with Corresponding Date(2000-01-01) and Time

tplant='NW-Vista' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P3M1Irradiance x INNER JOIN P3T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P3maxirrad.append(tempirrad1) #Max Irradiance Collector

P3Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P3time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant NW-Vista with Corresponding Date(2000-02-01) and Time

tplant='NW-Vista' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P3M2Irradiance x INNER JOIN P3T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P3maxirrad.append(tempirrad1) #Max Irradiance Collector

P3Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P3time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant NW-Vista with Corresponding Date(2000-03-01) and Time

tplant='NW-Vista' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P3M3Irradiance x INNER JOIN P3T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P3maxirrad.append(tempirrad1) #Max Irradiance Collector

P3Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P3time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

####################################################################

#END - Maximum Irradiance per Plant per Month for Plant 'NW-Vista'

####################################################################

#print P3maxirrad #Max Irradiance Collector

#print P3Months #Month Collector

#print P3time #Time Collector

####################################################################

#Maximum Irradiance per Plant per Month for Plant 'RiverBend'

####################################################################

P4maxirrad=[] #Max Irradiance Collector

P4Months=[] #Month Collector

P4time=[] #Time Collector

#Max Irradiance for Plant RiverBend with Corresponding Date(2000-01-01) and Time

tplant='RiverBend' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P4M1Irradiance x INNER JOIN P4T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P4maxirrad.append(tempirrad1) #Max Irradiance Collector

P4Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P4time.append(str(''.join(map(str,temptime))).rstrip())#Time Collector

#Max Irradiance for Plant RiverBend with Corresponding Date(2000-02-01) and Time

tplant='RiverBend' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P4M2Irradiance x INNER JOIN P4T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P4maxirrad.append(tempirrad1) #Max Irradiance Collector

P4Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P4time.append(str(''.join(map(str,temptime))).rstrip()) #Time Collector

#Max Irradiance for Plant RiverBend with Corresponding Date(2000-03-01) and Time

tplant='RiverBend' #Plant Name

#Retrieve Date,Irradiance and Time from Inner Joined Tables

c.execute('SELECT x.Dates,MAX(x.SensorIrradiance), y.Time, y.SensorIrradiance FROM P4M3Irradiance x INNER JOIN P4T1IrradTime y WHERE x.SensorIrradiance=y.SensorIrradiance')

all\_rows = c.fetchall() #Get Values from Table

for row in all\_rows:

tempdates,tempirrad1,temptime,tempirrad2 = row

print "The Maximum Irradiance for Plant %s during Month %s is: %i measured at time: %s" %(tplant,tempdates,tempirrad1,temptime)

P4maxirrad.append(tempirrad1) #Max Irradiance Collector

P4Months.append(str(''.join(map(str,tempdates)))) #Month Collector

P4time.append(str(''.join(map(str,temptime))).rstrip()) #Time Collector

####################################################################

#END - Maximum Irradiance per Plant per Month for Plant 'RiverBend'

####################################################################

#print P4maxirrad #Max Irradiance Collector

#print P4Months #Month Collector

#print P4time #Time Collector

conn.close()

####################################################

#PLOT

####################################################

Tmonths=P1Months+P2Months+P3Months+P4Months

Ttime=P1time+P2time+P3time+P4time

N = 3 #Bars per Plant

N2= 12 #Total X Labels

ind = np.arange(N) #Bars per Plant

ind2= np.arange(N2) #Total X Labels

width = 0.5 #Bar width

fig, ax = plt.subplots()

rects1 = ax.bar(ind, P1maxirrad, width, color='b')

rects2 = ax.bar(ind+3, P2maxirrad, width, color='g')

rects3 = ax.bar(ind+6, P3maxirrad, width, color='y')

rects4 = ax.bar(ind+9, P4maxirrad, width, color='r')

#Add X and Y Axis labels and Title

ax.set\_ylabel('Max Irradiance')

ax.set\_title('Max Irradiance per Plant and Month - (All foud at 12:00 Hrs)')

ax.set\_xticks(ind2)

ax.set\_xticklabels(Tmonths, rotation=45)

#Define Legend

ax.legend( (rects1[0],rects2[0],rects3[0],rects4[0]), ('BatCave','GreatPlains','NW-Vista','RiverBend') ,loc=4)

plt.axis([0, 12, 0, 80])

#Define Bar Labels

def autolabel(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.05\*height, '%d'%int(height),

ha='center', va='bottom')

autolabel(rects1)

autolabel(rects2)

autolabel(rects3)

autolabel(rects4)

plt.show()

####################################################

#END - PLOT

####################################################

**Question 2 – Part2c.py**

#The following code provides the answers for the third question in part 2 by

#Retrieving the information from the previously created data tables on Part1b.py

#Sum operation is done inside of the SQL statement for demonstration purposes

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

conn = sqlite3.connect('solarenergyplant.db')

c=conn.cursor()

####################################################

#Total Wattage per Month for Plant 'BatCave'

####################################################

P1twattage=[] #Total Wattage Collector

P1Months=[] #Month Collector

#Total Watts for Plant BatCave during Month 2000-01-01

c.execute('SELECT sum(PanelWatts) FROM P1M1Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='BatCave' #Plant Name

Dates="2000-01-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P1twattage.append(tempwatts) #Total Wattage Collector

P1Months.append(Dates) #Month Collector

#Total Watts for Plant BatCave during Month 2000-02-01

c.execute('SELECT sum(PanelWatts) FROM P1M2Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='BatCave' #Plant Name

Dates="2000-02-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P1twattage.append(tempwatts) #Total Wattage Collector

P1Months.append(Dates) #Month Collector

#Total Watts for Plant BatCave during Month 2000-03-01

c.execute('SELECT sum(PanelWatts) FROM P1M3Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='BatCave' #Plant Name

Dates="2000-03-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P1twattage.append(tempwatts) #Total Wattage Collector

P1Months.append(Dates) #Month Collector

####################################################

#END - Total Wattage per Month for Plant 'BatCave'

####################################################

#print P1twattage #Total Wattage Collector

#print P1Months #Month Collector

####################################################

#Total Wattage per Month for Plant 'GreatPlains'

####################################################

P2twattage=[] #Total Wattage Collector

P2Months=[] #Month Collector

#Total Watts for Plant GreatPlains during Month 2000-01-01

c.execute('SELECT sum(PanelWatts) FROM P2M1Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='GreatPlains' #Plant Name

Dates="2000-01-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P2twattage.append(tempwatts) #Total Wattage Collector

P2Months.append(Dates) #Month Collector

#Total Watts for Plant GreatPlains during Month 2000-02-01

c.execute('SELECT sum(PanelWatts) FROM P2M2Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='GreatPlains' #Plant Name

Dates="2000-02-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P2twattage.append(tempwatts) #Total Wattage Collector

P2Months.append(Dates) #Month Collector

#Total Watts for Plant GreatPlains during Month 2000-03-01

c.execute('SELECT sum(PanelWatts) FROM P2M3Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='GreatPlains' #Plant Name

Dates="2000-03-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P2twattage.append(tempwatts) #Total Wattage Collector

P2Months.append(Dates) #Month Collector

####################################################

#END - Total Wattage per Month for Plant 'GreatPlains'

####################################################

#print P2twattage #Total Wattage Collector

#print P2Months #Month Collector

####################################################

#Total Wattage per Month for Plant 'NW-Vista'

####################################################

P3twattage=[] #Total Wattage Collector

P3Months=[] #Month Collector

#Total Watts for Plant NW-Vista during Month 2000-01-01

c.execute('SELECT sum(PanelWatts) FROM P3M1Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='NW-Vista' #Plant Name

Dates="2000-01-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P3twattage.append(tempwatts) #Total Wattage Collector

P3Months.append(Dates) #Month Collector

#Total Watts for Plant NW-Vista during Month 2000-02-01

c.execute('SELECT sum(PanelWatts) FROM P3M2Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='NW-Vista' #Plant Name

Dates="2000-02-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P3twattage.append(tempwatts) #Total Wattage Collector

P3Months.append(Dates) #Month Collector

#Total Watts for Plant NW-Vista during Month 2000-03-01

c.execute('SELECT sum(PanelWatts) FROM P3M3Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='NW-Vista' #Plant Name

Dates="2000-03-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P3twattage.append(tempwatts) #Total Wattage Collector

P3Months.append(Dates) #Month Collector

####################################################

#END - Total Wattage per Month for Plant 'NW-Vista'

####################################################

#print P3twattage #Total Wattage Collector

#print P3Months #Month Collector

####################################################

#Total Wattage per Month for Plant 'RiverBend'

####################################################

P4twattage=[] #Total Wattage Collector

P4Months=[] #Month Collector

#Total Watts for Plant RiverBend during Month 2000-01-01

c.execute('SELECT sum(PanelWatts) FROM P4M1Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='RiverBend' #Plant Name

Dates="2000-01-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P4twattage.append(tempwatts) #Total Wattage Collector

P4Months.append(Dates) #Month Collector

#Total Watts for Plant RiverBend during Month 2000-02-01

c.execute('SELECT sum(PanelWatts) FROM P4M2Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='RiverBend' #Plant Name

Dates="2000-02-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P4twattage.append(tempwatts) #Total Wattage Collector

P4Months.append(Dates) #Month Collector

#Total Watts for Plant RiverBend during Month 2000-03-01

c.execute('SELECT sum(PanelWatts) FROM P4M3Watts') #Retrieve the Sum of Watts from Table

all\_rows = c.fetchall() #Get Values from Table

tempwatts = int(''.join(map(str,all\_rows[0])))

tplant='RiverBend' #Plant Name

Dates="2000-03-01" #Current Date

print "The total Wattage for Plant %s during Month %s is: %i" %(tplant,Dates,tempwatts)

P4twattage.append(tempwatts) #Total Wattage Collector

P4Months.append(Dates) #Month Collector

####################################################

#END - Total Wattage per Month for Plant 'RiverBend'

####################################################

#print P4twattage #Total Wattage Collector

#print P4Months #Month Collector

conn.close()

####################################################

#PLOT

####################################################

Tmonths=P1Months+P2Months+P3Months+P4Months

N = 3 #Bars per Plant

N2= 12 #Total X Labels

ind = np.arange(N) #Bars per Plant

ind2= np.arange(N2) #Total X Labels

width = 0.5 #Bar width

fig, ax = plt.subplots()

rects1 = ax.bar(ind, P1twattage, width, color='b')

rects2 = ax.bar(ind+3, P2twattage, width, color='g')

rects3 = ax.bar(ind+6, P3twattage, width, color='y')

rects4 = ax.bar(ind+9, P4twattage, width, color='r')

#Add X and Y Axis labels and Title

ax.set\_ylabel('Total Wattage')

ax.set\_title('Tottal Wattage per Plant and Month')

ax.set\_xticks(ind2)

ax.set\_xticklabels(Tmonths, rotation=45)

#Define Legend

ax.legend( (rects1[0],rects2[0],rects3[0],rects4[0]), ('BatCave','GreatPlains','NW-Vista','RiverBend') ,loc=4)

plt.axis([0, 12, 0, 2000])

#Define Bar Labels

def autolabel(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.05\*height, '%d'%int(height),

ha='center', va='bottom')

autolabel(rects1)

autolabel(rects2)

autolabel(rects3)

autolabel(rects4)

plt.show()

####################################################

#END - PLOT

####################################################

**Question 3 – Part3a.py**

import numpy as np

import matplotlib.pyplot as plt

import scipy as sp

import sqlite3

import csv

#The following section represents the Mapper

text\_file=open("solardata.csv",'r') #Document with data of SolarData

file = open('solardata.csv')

reader = csv.reader(file)

next(reader, None)

dailysamps=24 #Number of Samples per day

#Required variabes

day=[] #Daily date list

etr=[] #Daily ETR list

aetr=[] #Average ETR

detr=[] #Temporal Average ETR

daylist=[] #Final Day List

#Read lines and store day and ETR value in corresponding lists

for line in reader:

day.append(str(line[0]))

etr.append(int(line[4]))

#The following section represents the Reducer which will provide the average

for hours in range(0, len(etr),dailysamps):

detr=np.average(etr[hours:hours+dailysamps])

aetr.append(detr)

daylist.append(day[hours])

print aetr

print daylist

####################################################

#PLOT

####################################################

N = len(aetr) #Total Bars

ind = np.arange(N) #Index per Bar

width = 0.5 #Bar width

fig, ax = plt.subplots()

rects1 = ax.bar(ind, aetr, width, color='r')

#Add X and Y Axis labels and Title

ax.set\_ylabel('ETR Average')

ax.set\_title('Average ETR per Day')

ax.set\_xticks(ind)

ax.set\_xticklabels(daylist, rotation=45)

plt.axis([0, N+1, 0, 500])

#Define Bar Labels

def autolabel(rects):

for rect in rects:

height = rect.get\_height()

ax.text(rect.get\_x()+rect.get\_width()/2., 1.05\*height, '%d'%int(height),

ha='center', va='bottom')

#autolabel(rects1)

plt.show()

####################################################

#END - PLOT

####################################################