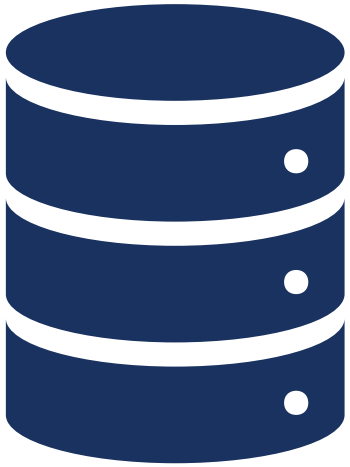


ESTABLISHMENT OF REGIS HEALTHCARE INFORMATICS DATABASE - CONTINUED

BY GILBERT ANTHONY BERNAL

OUTLINE

INTRODUCTION

Data Engineering project focus

Project Goal: Complete Regis Health informatics database.

Previous Project goal was to create database

- Was not able to get database on server
- Codes worked with a single format
- No data governance added to database

REQUIREMENTS & GOAL

Requirements

- Create a way to allow for multiple file formats to be uploaded.
- Create PostgreSQL database on sever
- Create upload form
- Automate python code processes

Goal

- Complete creation of Regis server health informatics PostgreSQL database on server and automate all processes.

python

```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ └─────── day of week (0 - 7)
# │ │ │ └──────────── month (1 - 12)
# │ │ └─────────────────── day of month (1 - 31)
# │ └──────────────────────── hour (0 - 23)
# └───────────────────────────── min (0 - 59)
```

cron

HTML

PostgreSQL

tableau

```python
import os, pandas,shutil
from xlrd import open_workbook


def convert(path, Archive):
    def files(path):
        for file in os.listdir(path):
            if os.path.isfile(os.path.join(path, file)):
                yield file
    for file in files(path):
        if 'json' in file:
            print (file +' is in JSON format must be coverted')
            full_path = (path + '\\' + file)
            name = file.replace("json", "csv")
            new_path=(path + '\\' + name)
            df = pandas.read_json(full_path, orient='records')
            print(df)
            df.to_csv(name, index=False)
            shutil.move(name, new_path)
            shutil.move(full_path,Archive)
        elif 'csv' in file:
            print (file + ' is in correct format format')
        else:
            print (file +' is in Excel format must be coverted')
            full_path =(path+'\\'+file)
            wb = open_workbook(full_path)
            print(full_path)
            print(wb)
            sheets = range(0, wb.nsheets)
            for i in sheets:
                sheet = wb.sheet_by_index(i)
                print sheet.name
                Sheet_name = sheet.name.replace(" ", "") + '.csv'
                print(Sheet_name)
                with open(Sheet_name, "w") as file:
                    writer = csv.writer(file, delimiter=",")
                    print sheet, sheet.name, sheet.ncols, sheet.nrows
                    header = [cell.value for cell in sheet.row(0)]
                    writer.writerow(header)
                    for row_idx in range(1, sheet.nrows):
                        row = [int(cell.value) if isinstance(cell.value, flo
                               for cell in sheet.row(row_idx)]
                        writer.writerow(row)
                    print(file)
                    print(Sheet_name)
                shutil.move(Sheet_name, path)
            shutil.move(full_path,Archive)
```

CONVERT.PY

JSON_CONV.py | excel_Conver.py | Import_Function.py | Database_Governance.py | test.py | Convert.py | info.py | Import_CSV.py

```python
import os, psycopg2, touch
from info import *

from datetime import datetime
root="C:\Users\eltac\Desktop\Regis_Homework\Test_Folder"


def files(path):
    for file in os.listdir(path):
        if os.path.isfile(os.path.join(path, file)):
            yield file


for file in files(root+'\\Archive' ):
    csvfile = file
    name = csvfile.replace(".csv", "")
    fullpath = (root+'\\Archive' + '\\' + csvfile)
    dt = datetime.fromtimestamp(os.stat(fullpath).st_ctime)
    today = datetime.now()
    date_diff = today - dt

    print(csvfile)
    print (name)
    print (fullpath)
    print(date_diff)

    if date_diff.days > 9:
        print("File greater than 9 Days")
        connection = psycopg2.connect(user=User, password=Password, host=Host, port=Port, database=Database)
        cursor = connection.cursor()
        cursor.execute('Drop table ' + name)
        connection.commit()
        print("Table dropped PostgreSQL ")
        cursor.close()
        connection.close()
        os.remove(fullpath)
    else:
        print("File less than 9 days")

touch.touch(root+'\\landing\\'+'Database_Governance.trig')
```

DATABASE_GOVERNANCE.PY

CREATE DATABASE

```
gbernal@cobalt:/home/gbernal
--------------------------------------------------------------------------------
Total                                                   182 kB/s | 7.4 MB  00:41
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-9.2.24-1.el7_5.x86_64                            1/4
  Installing : uuid-1.6.2-26.el7.x86_64                                    2/4
  Installing : postgresql-contrib-9.2.24-1.el7_5.x86_64                    3/4
  Installing : postgresql-server-9.2.24-1.el7_5.x86_64                     4/4
  Verifying  : postgresql-server-9.2.24-1.el7_5.x86_64                     1/4
  Verifying  : uuid-1.6.2-26.el7.x86_64                                    2/4
  Verifying  : postgresql-9.2.24-1.el7_5.x86_64                            3/4
  Verifying  : postgresql-contrib-9.2.24-1.el7_5.x86_64                    4/4

Installed:
  postgresql-contrib.x86_64 0:9.2.24-1.el7_5
  postgresql-server.x86_64 0:9.2.24-1.el7_5

Dependency Installed:
  postgresql.x86_64 0:9.2.24-1.el7_5         uuid.x86_64 0:1.6.2-26.el7

Complete!
[root@cobalt gbernal]#
```

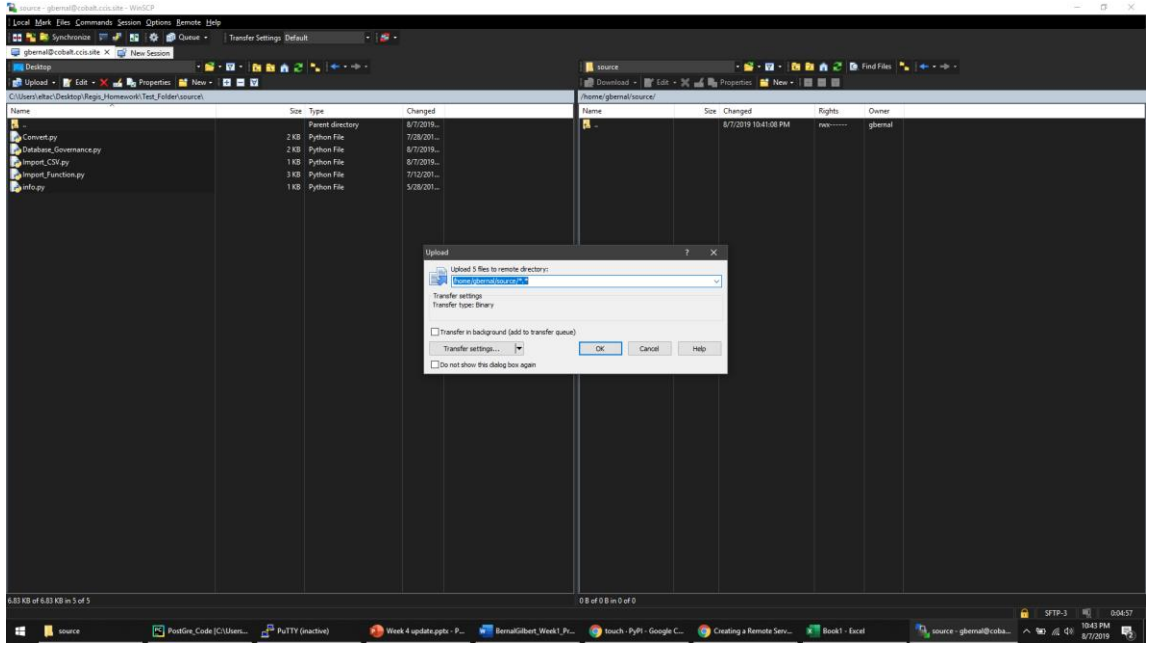
```
postgres=# create user gbernal with encrypted password 'SatMay18';
CREATE ROLE
postgres=# grant all privileges on database h_info to gbernal;
GRANT
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileg
es
-----------+----------+----------+-------------+-------------+-----------------
-----
 h_info    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =Tc/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres+
           |          |          |             |             | gbernal=CTc/postg
res
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
(4 rows)

postgres=# \du
                             List of roles
 Role name |                   Attributes                   | Member of
-----------+------------------------------------------------+-----------
 gbernal   |                                                | {}
 postgres  | Superuser, Create role, Create DB, Replication | {}

postgres=#
```

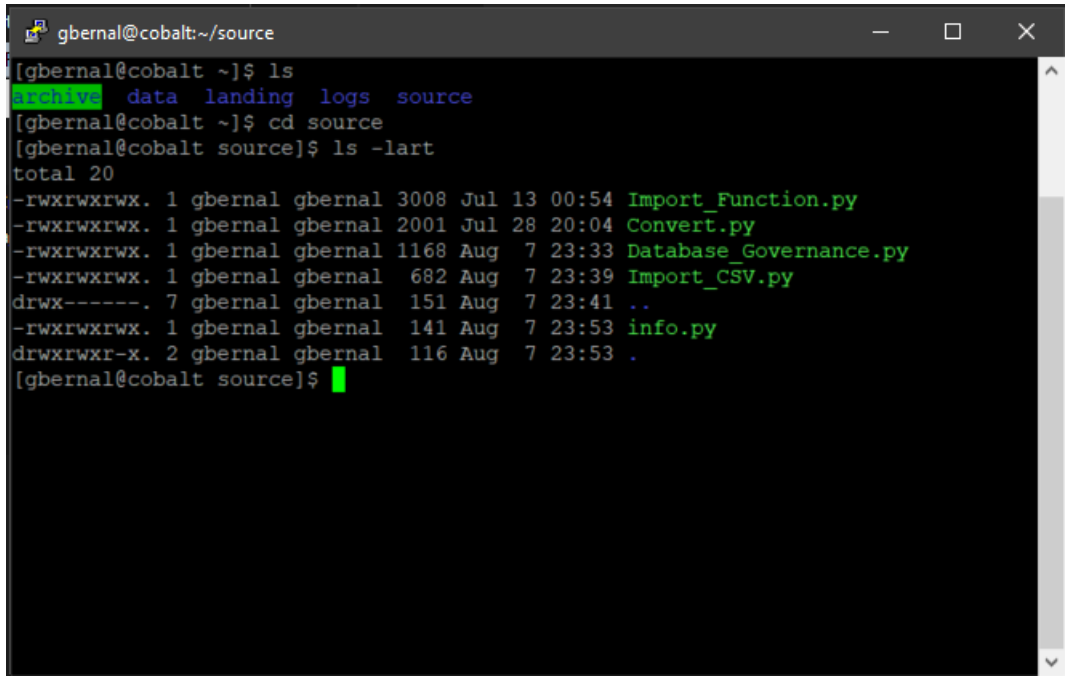
CREATE DATABASE - CONTINUED

```
postgres=# create database H_INFO;
CREATE DATABASE
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileg
es
-----------+----------+----------+-------------+-------------+------------------
-----
 h_info    | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
(4 rows)

postgres=#
```

```
gbernal@cobalt:~
# -----------------------------
# PostgreSQL configuration file
# -----------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes        Time units:  ms  = milliseconds
#                MB = megabytes                     s   = seconds
#                GB = gigabytes                     min = minutes
#                                                   h   = hours
#                                                   d   = days


#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'           # use data in another directory
                                        # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
                                        # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf' # ident configuration file
                                        # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                 # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = '*'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
"./data/postgresql.conf" 578L, 19843C
```

COPY AND TEST PYTHON CODES





COPY AND TEST PYTHON CODES

```
[gbernal@cobalt source]$ python2.7 Import_CSV.py
Gbernal_Chicago_Crime10.csv is in correct format format
Gbernal_Chicago_Crime10.csv
Gbernal_Chicago_Crime10
/home/gbernal/data/Gbernal_Chicago_Crime10.csv
create table h_info_t.Gbernal_Chicago_Crime10(
id int,
case_number varchar(9),
date varchar(16),
block varchar(35),
iucr varchar(4),
primary_type varchar(33),
description varchar(59),
location_description varchar(53),
arrest varchar(5),
domestic varchar(5),
beat smallint,
district varchar(2),
ward varchar(2),
community_area varchar(2),
fbi_code varchar(3),
x_coordinate varchar(7),
y_coordinate varchar(7),
year smallint,
updated_on varchar(16),
latitude varchar(11),
longitude varchar(12),
location varchar(29),
historical_wards_2003_2015 varchar(2),
zip_codes varchar(5),
community_areas varchar(2),
census_tracts varchar(3),
wards varchar(2),
boundaries_zip_codes varchar(2),
police_districts varchar(2),
police_beats varchar(3));
Table created successfully in PostgreSQL
```

```
gbernal@cobalt:~/data
h_info=# select * from h_info_t.gbernal_chicago_crime;
    id     | case_number |        date        |                 block                 | iucr |
     | arrest | domestic | beat | district | ward | community_area | fbi_code | x_coordinate
 community_areas | census_tracts | wards | boundaries_zip_codes | police_districts | police_b
----------+-------------+--------------------+---------------------------------------+------+----
-----+--------+----------+------+----------+------+----------------+----------+--------------
-----------------+---------------+-------+----------------------+------------------+---------
 11765259 | JC357779    | 7/20/2019 23:55    | 047XX W WRIGHTWOOD AVE                | 143A | WEA
     | TRUE   | FALSE    | 2521 | 25       | 31   | 19             | 15       | 1144253
 19              | 305           | 17    | 2                    | 6               | 176
 11765196 | JC357770    | 7/20/2019 23:51    | 052XX S LAKE PARK AVE                 | 560  | ASS
     | TRUE   | FALSE    |  234 | 2        | 4    | 41             | 08A      | 1187644
 8               | 500           | 10    | 10                   | 24              | 122
 11765172 | JC357853    | 7/20/2019 23:50    | 054XX S LUNA AVE                      | 1477 | WEA
     | FALSE  | FALSE    |  814 | 8        | 23   | 56             | 15       | 1140283
 53              | 607           | 6     | 7                    | 13              | 112
 11768457 | JC361199    | 7/20/2019 23:50    | 001XX E WACKER DR                     | 2825 | OTH
     | FALSE  | FALSE    |  114 | 1        | 42   | 32             | 26       | 1177683
 38              | 580           | 36    | 42                   | 22              | 87
 11765212 | JC357900    | 7/20/2019 23:47    | 030XX N CHRISTIANA AVE                | 502T | OTH
     | TRUE   | FALSE    | 1412 | 14       | 35   | 21             | 26       | 1153523
 22              | 216           | 12    | 39                   | 7               | 168
 11765239 | JC357742    | 7/20/2019 23:47    | 016XX W 46TH ST                       | 486  | BAT
     | FALSE  | FALSE    |  924 | 9        | 15   | 61             | 08B      | 1166020
 59              | 738           | 3     | 37                   | 23              | 108
 11765133 | JC357739    | 7/20/2019 23:44    | 058XX S RICHMOND ST                   | 460  | BAT
     | FALSE  | FALSE    |  824 | 8        | 16   | 63             | 08B      | 1157713
```

INSTALL CHRON

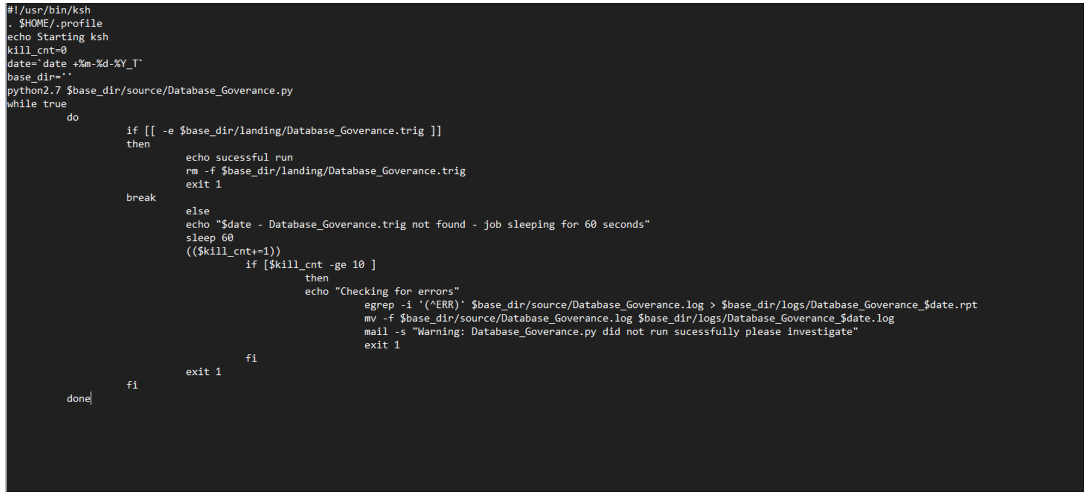
gbernal@cobalt:/home/gbernal/data

```
[gbernal@cobalt data]$ su
Password:
[root@cobalt data]# crontab -l
no crontab for root
[root@cobalt data]# yum install cronie
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Could not retrieve mirrorlist http://mirrorlist.centos.org/?release=7&arch=x86_6
4&repo=os&infra=stock error was
14: curl#6 - "Could not resolve host: mirrorlist.centos.org; Unknown error"
 * base: mirror.dc2.hackingand.coffee
 * centos-qemu-ev: mirror.netdepot.com
 * extras: linux.mirrors.es.net
 * updates: repos.dfw.quadranet.com
base                                                     | 3.6 kB     00:00
centos-ceph-luminous                                     | 2.9 kB     00:00
centos-openstack-rocky                                   | 2.9 kB     00:00
centos-qemu-ev                                           | 2.9 kB     00:00
extras                                                   | 3.4 kB     00:00
updates                                                  | 3.4 kB     00:00
(1/3): centos-qemu-ev/7/x86_64/primary_db                |  65 kB     00:00
(2/3): extras/7/x86_64/primary_db                        | 215 kB     00:00
(3/3): centos-openstack-rocky/7/x86_64/primary_db        | 1.0 MB     00:00
Package cronie-1.4.11-20.el7_6.x86_64 already installed and latest version
```

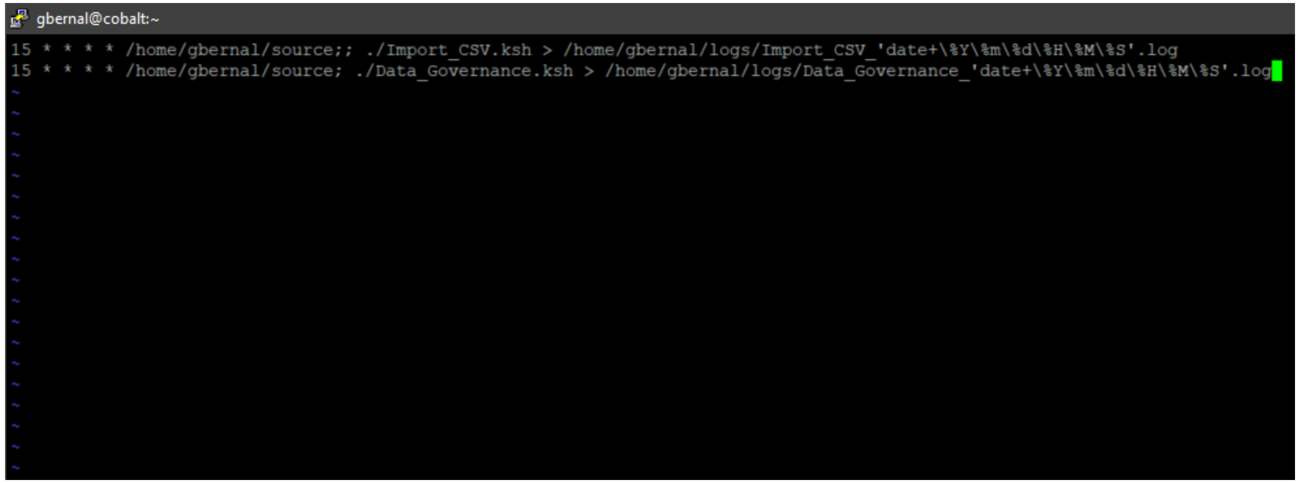
/home/gbernal/source/Import_CSV.ksh - gbernal@cobalt.ccis.site - Editor - WinSCP

```
#!/usr/bin/ksh
. $HOME/.profile
echo Starting ksh
kill_cnt=0
date=`date +%m-%d-%Y_T`
base_dir=''
python2.7 $base_dir/source/Import_CSV.py
while true
        do
                if [[ -e $base_dir/landing/Import_CSV.trig ]]
                then
                        echo sucessful run
                        rm -f $base_dir/landing/Import_CSV.trig
                        exit 1
                break
                        else
                        echo "$date - Import_CSV.trig not found - job sleeping for 60 seconds"
                        sleep 60
                        (($kill_cnt+=1))
                                if [$kill_cnt -ge 10 ]
                                        then
                                        echo "Checking for errors"
                                                egrep -i '(^ERR)' $base_dir/source/Import_CSV.log > $base_dir/logs/Import_CSV_$date.rpt
                                                mv -f $base_dir/source/Import_CSV.log $base_dir/logs/Import_CSV_$date.log
                                                mail -s "Warning: Import_CSV.py did not run sucessfully please investigate"
                                                exit 1
                                fi
                        exit 1
                fi
        done
```

INSTALL CHRON



```
#!/usr/bin/ksh
. $HOME/.profile
echo Starting ksh
kill_cnt=0
date=`date +%m-%d-%Y_T`
base_dir=''
python2.7 $base_dir/source/Database_Goverance.py
while true
        do
                if [[ -e $base_dir/landing/Database_Goverance.trig ]]
                then
                        echo sucessful run
                        rm -f $base_dir/landing/Database_Goverance.trig
                        exit 1
                break
                        else
                        echo "$date - Database_Goverance.trig not found - job sleeping for 60 seconds"
                        sleep 60
                        (($kill_cnt+=1))
                                if [$kill_cnt -ge 10 ]
                                        then
                                        echo "Checking for errors"
                                                egrep -i '(^ERR)' $base_dir/source/Database_Goverance.log > $base_dir/logs/Database_Goverance_$date.rpt
                                                mv -f $base_dir/source/Database_Goverance.log $base_dir/logs/Database_Goverance_$date.log
                                                mail -s "Warning: Database_Goverance.py did not run sucessfully please investigate"
                                                exit 1
                                fi
                        exit 1
                fi
        done
```



```
15 * * * * /home/gbernal/source;; ./Import_CSV.ksh > /home/gbernal/logs/Import_CSV_'date+\%Y\%m\%d\%H\%M\%S'.log
15 * * * * /home/gbernal/source; ./Data_Governance.ksh > /home/gbernal/logs/Data_Governance_'date+\%Y\%m\%d\%H\%M\%S'.log
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

HTML WEBFORM

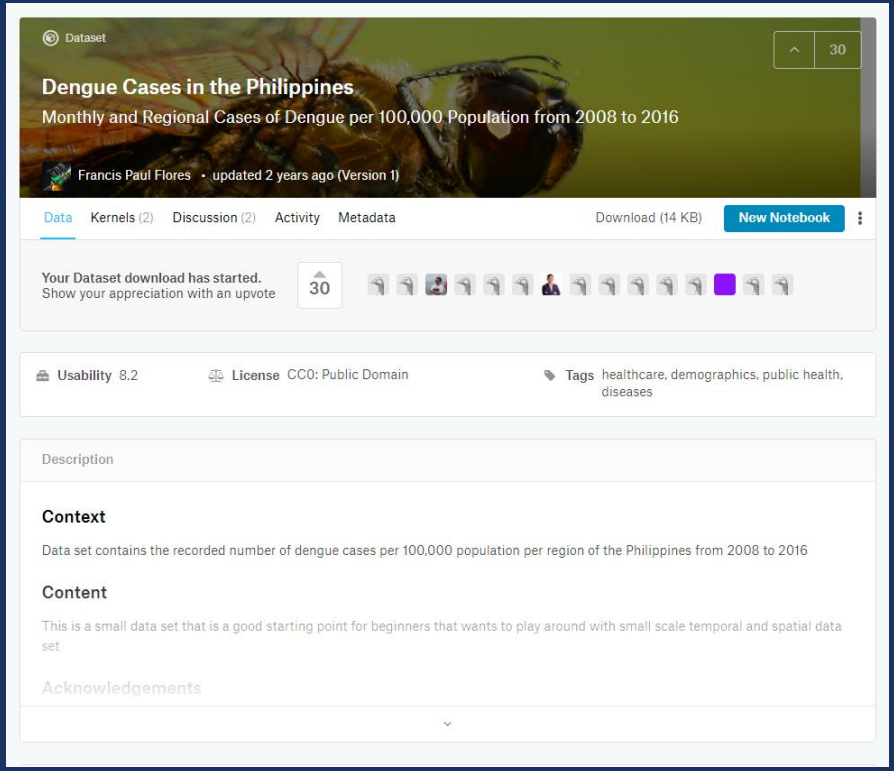
```html
<html>
<body onload="myFunction()">

<input type="file" id="myFile" multiple size="50" onchange="myFunction()">
<input type='submit' value='Submit' />
<p id="demo"></p>

<script>
function myFunction(){
  var x = document.getElementById("myFile");
  var txt = "";
  if ('files' in x) {
    if (x.files.length == 0) {
      txt = "Select one or more files.";
    } else {
      for (var i = 0; i < x.files.length; i++) {
        txt += "<br><strong>" + (i+1) + ". file</strong><br>";
        var file = x.files[i];
        if ('name' in file) {
          txt += "name: " + file.name + "<br>";
        }
        if ('size' in file) {
          txt += "size: " + file.size + " bytes <br>";
        }
      }
    }
  }
  else {
    if (x.value == "") {
      txt += "Select one or more files.";
    } else {
      txt += "The files property is not supported by your browser!";
      txt  += "<br>The path of the selected file: " + x.value; // If the browser does not support the files property,
    }
  }
  document.getElementById("demo").innerHTML = txt;
}
</script>

<p><strong>Tip:</strong> Use the Control or the Shift key to select multiple files.</p>

</body>
</html>
```

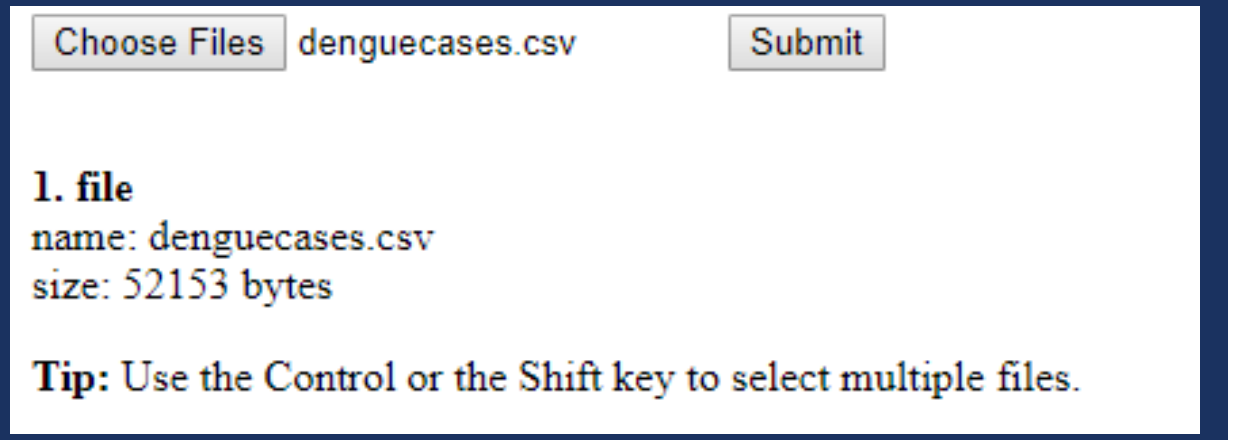
Choose Files GBERNAL_C...odes.csv Submit

1. file
name: GBERNAL_California_Zip_Codes.csv
size: 72270 bytes

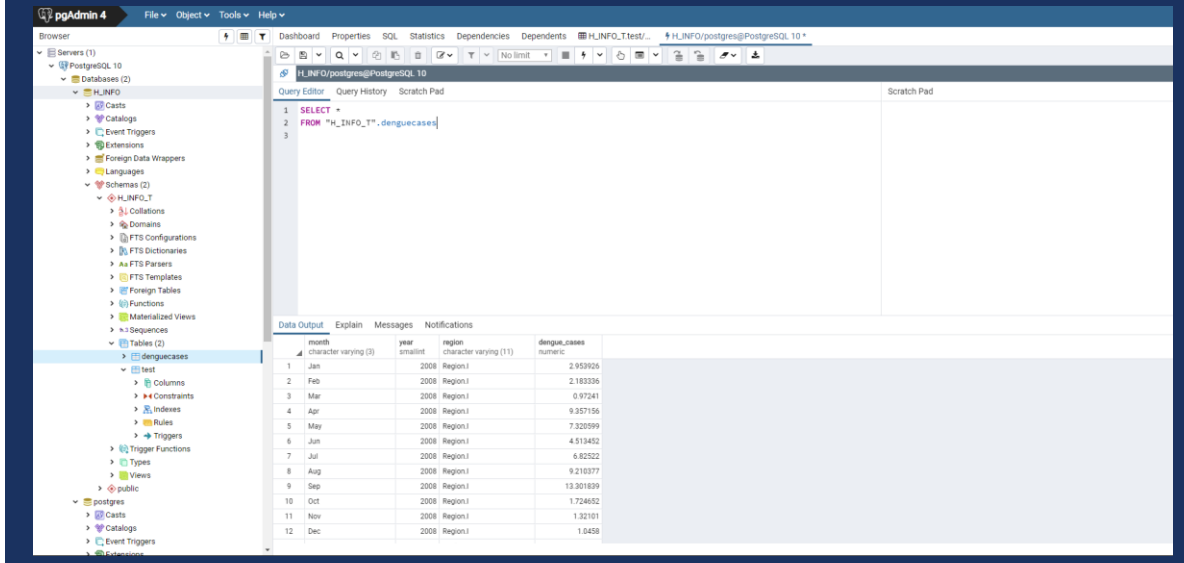
Tip: Use the Control or the Shift key to select multiple files.

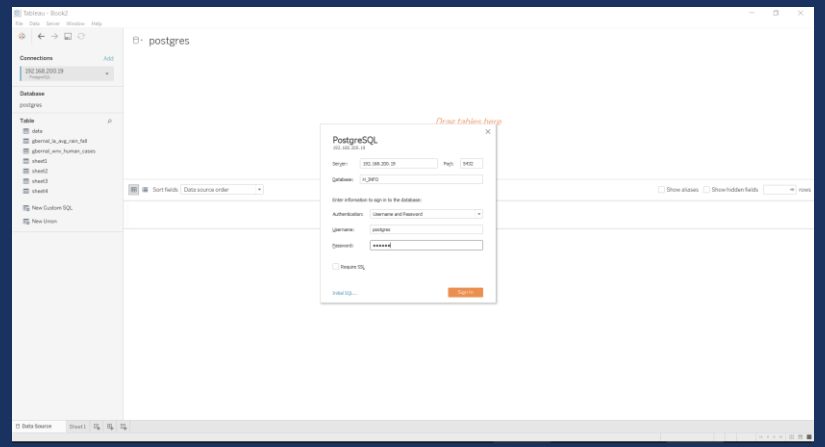
ANALYSIS: COLLECT DATA & UPLOAD TO SERVER



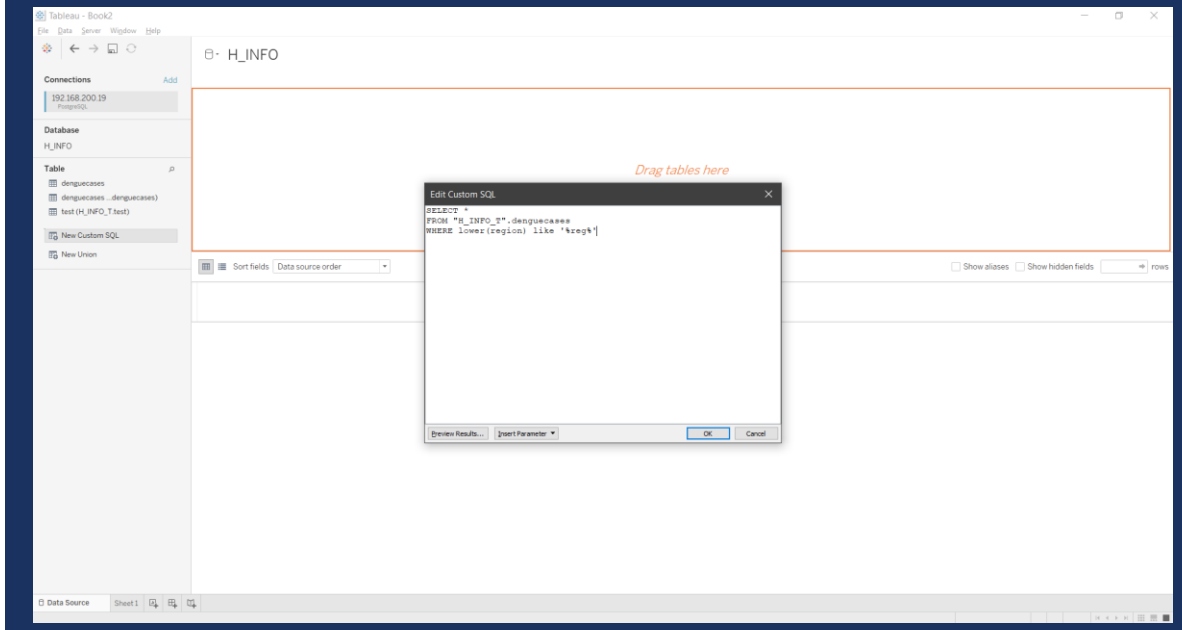


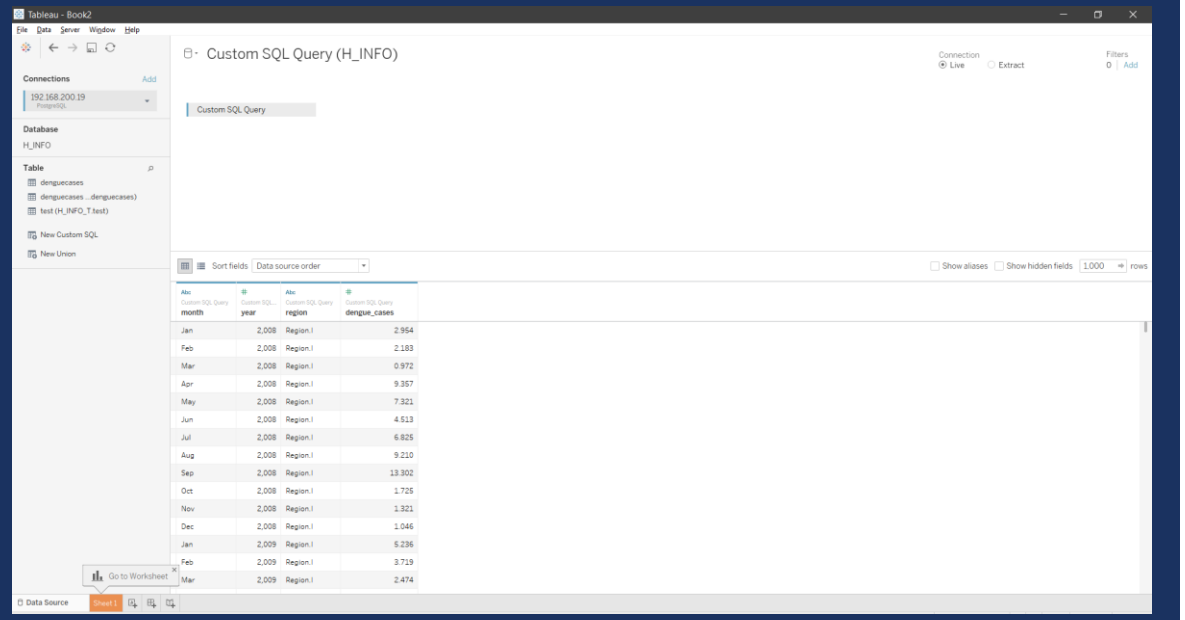
ANALYSIS: CHECK DATA & CONNECT TO TABLEAU



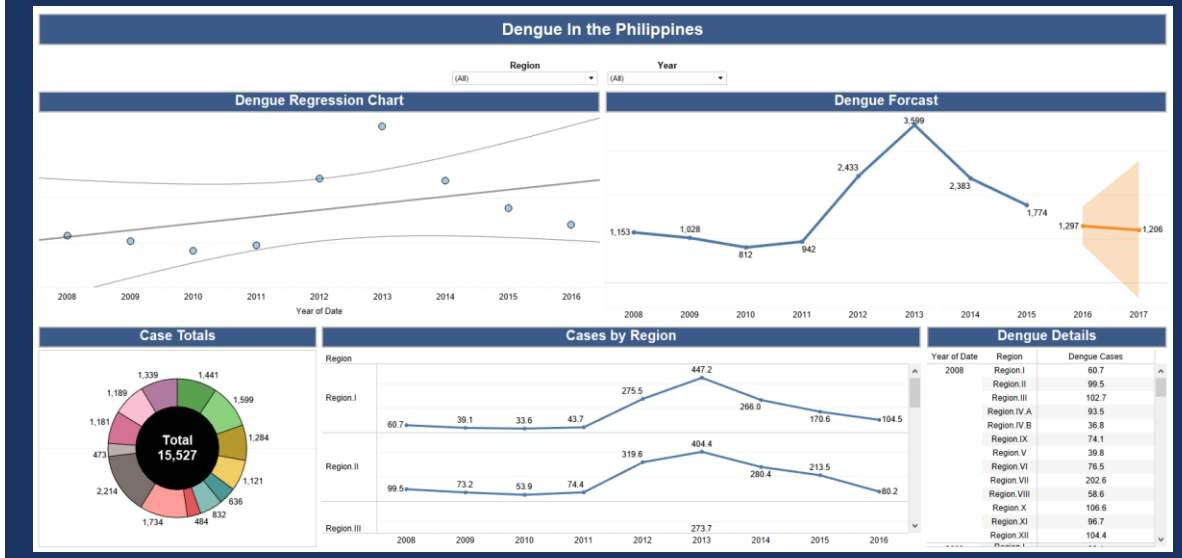


ANALYSIS: TABLEAU SQL AND DATA IMPORT

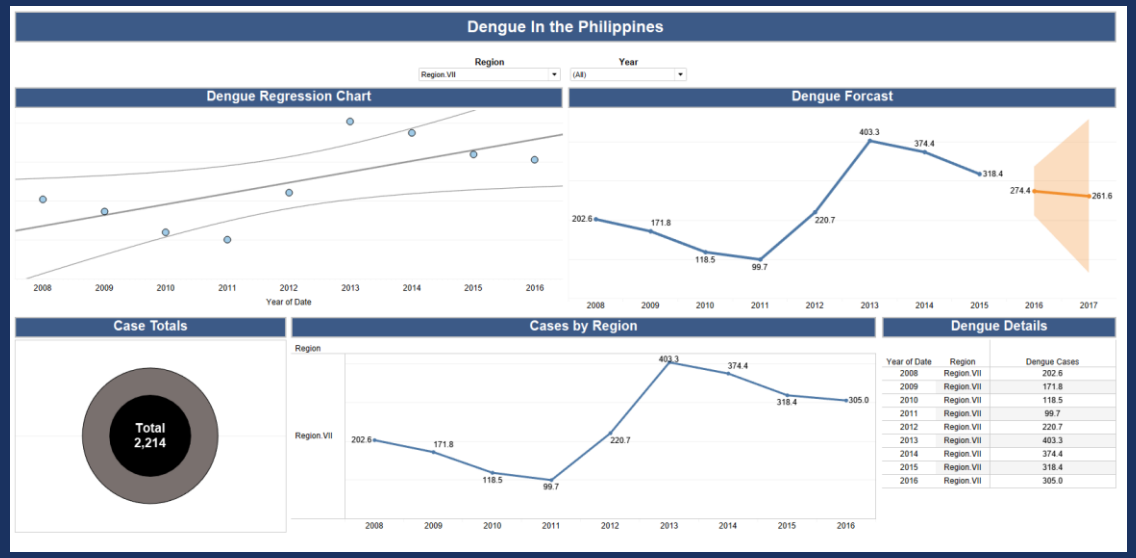


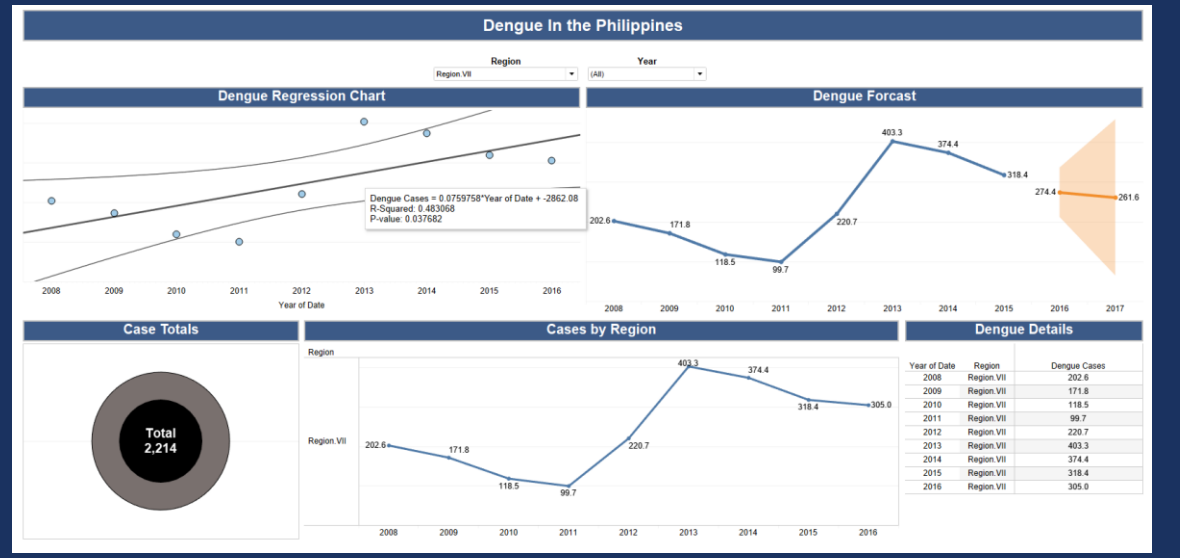


ANALYSIS: TABLEAU REPORT & RESULTS



ANALYSIS: TABLEAU REPORT & RESULTS





CONCLUSION

- Overall goals and requirement of project were met
  - Created a way to handle multiple file types for previous codes
  - Created PostgreSQL database on server
  - Added data governance to database
  - Automated database using python codes and Chron
  - Used automated process and Tableau to provide a proof of concept

REFERENCES

- Flores, F. P. (2017, October 30). Dengue Cases in the Philippines. Retrieved August 22, 2019, from https://www.kaggle.com/grosvenpaul/dengue-cases-in-the-philippines
- The World's Most Advanced Open Source Relational Database. (n.d.). Retrieved June 26, 2019, from https://www.postgresql.org/
- Welcome to Python.org. (n.d.). Retrieved June 26, 2019, from https://www.python.org/