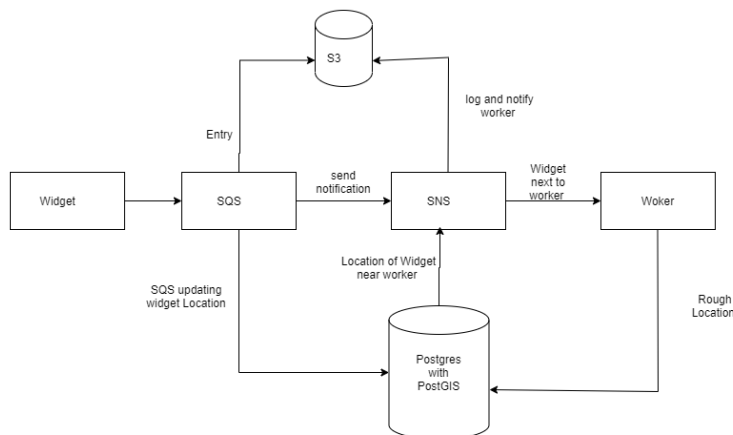


**Final Project
For
System Integration Fall 2020
By
Bishnu Poudyal**

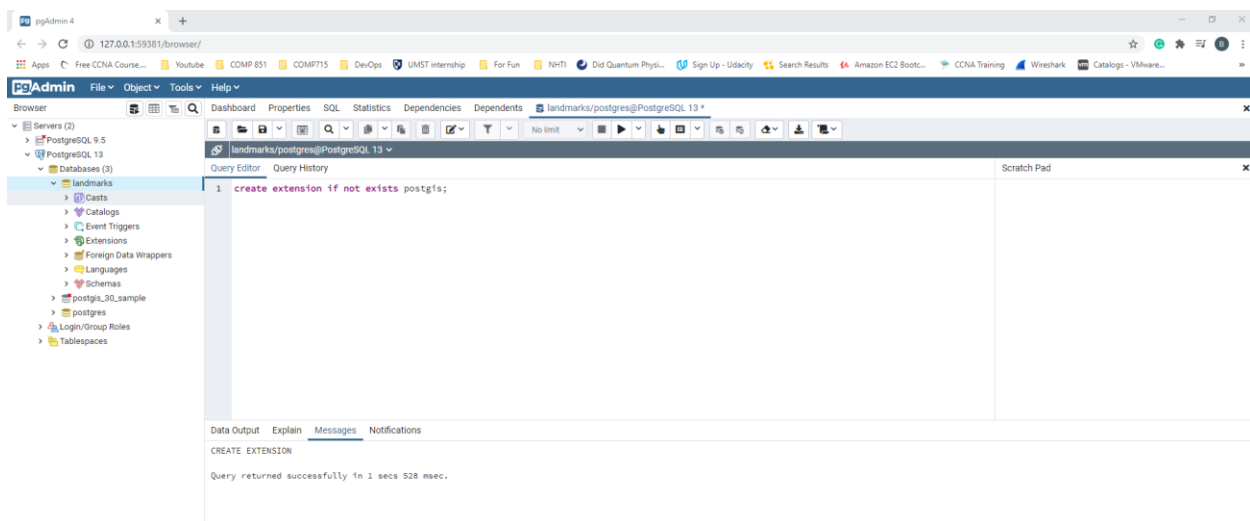
GitHub Link: https://github.com/bipoudyal16/SI_Final_Project.git

Project Topic:

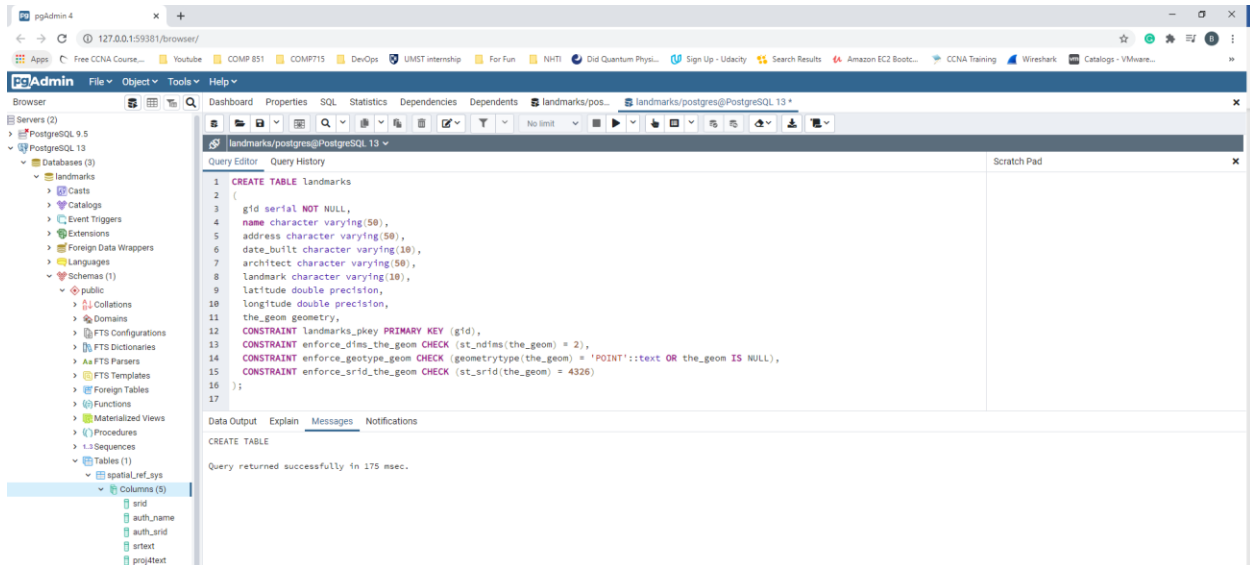
I have chosen option two for my project. In this project I have deployed PTWC Widgets as a communication GPS position for field operations. I am implementing database which can find out the widgets in the locations where worker is located. This will use PostGIS database and using latitude and longitude to provide site info.



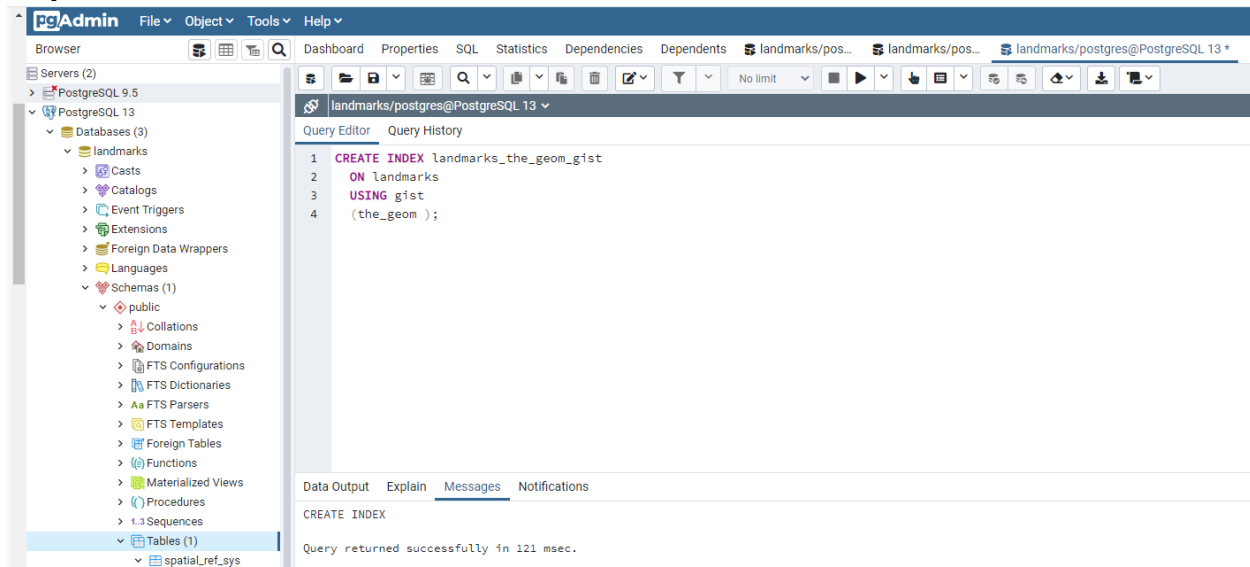
Step 1: Creating postgis



Step 2: Create the table in database



Step 3: Create Index



Step 4: Copy the CSV data into the Database

pgAdmin 4

127.0.0.1:59381/browser/

Apps Free CCNA Course... Youtube COMP 851 COMP715 DevOps UMST Internship For Fun NHIT Did Quantum Physi... Sign Up - Udacity Search Results Amazon EC2 Boot... CCNA Training Win

pgAdmin File Object Tools Help

Browser Servers (2) PostgreSQL 9.5 PostgreSQL 13 Databases (3) landmarks Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Schemas (1) public Collations Domains FTS Configurations FTS Dictionaries Aa FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Procedures Sequences Tables (2) landmarks Columns (9) gid name address date_built architect landmark latitude longitude the_geom

landmarks/postgres@PostgreSQL 13 *
Query Editor Query History

```
1 copy landmarks(name,gid,address,date_built,architect,landmark,latitude,longitude)
2 FROM 'C:\Users\Bishnu Poudyal\Desktop\Final_Project\Individual_Landmarks.csv' DELIMITERS ',' CSV HEADER;
```

Data Output Explain Messages Notifications

COPY 317

Query returned successfully in 96 msec.

pgAdmin 4

127.0.0.1:59381/browser/

Apps Free CCNA Course... Youtube COMP 851 COMP715 DevOps UMST Internship For Fun NHIT Did Quantum Physi... Sign Up - Udacity Search Results Amazon EC2 Boot... CCNA Training Wirehark Catalogs - VMware...

pgAdmin File Object Tools Help

Browser Servers (2) PostgreSQL 9.5 PostgreSQL 13 Databases (3) landmarks Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Schemas (1) public Collations Domains FTS Configurations FTS Dictionaries Aa FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Procedures Sequences Tables (2) landmarks Columns (9) gid name address date_built architect landmark latitude longitude the_geom

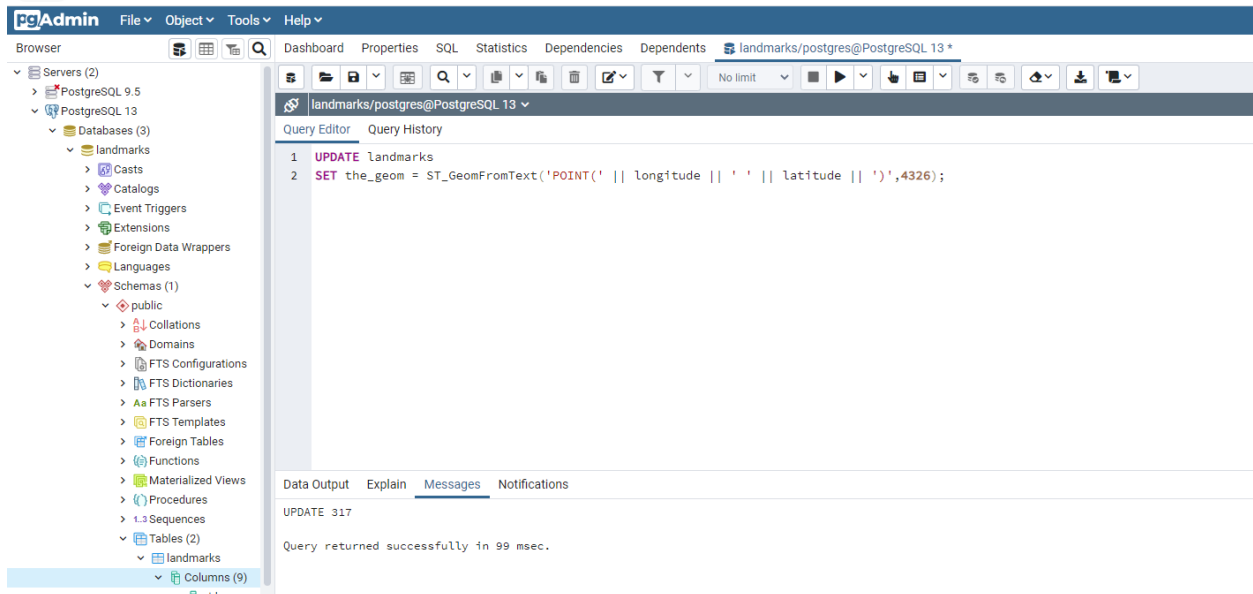
landmarks/postgres@PostgreSQL 13 *
Query Editor Query History

```
1 copy landmarks(name,gid,address,date_built,architect,landmark,latitude,longitude)
2 FROM 'C:\Users\Bishnu Poudyal\Desktop\Final_Project\Individual_Landmarks.csv' DELIMITERS ',' CSV HEADER;
3
4 SELECT * FROM landmarks
5
```

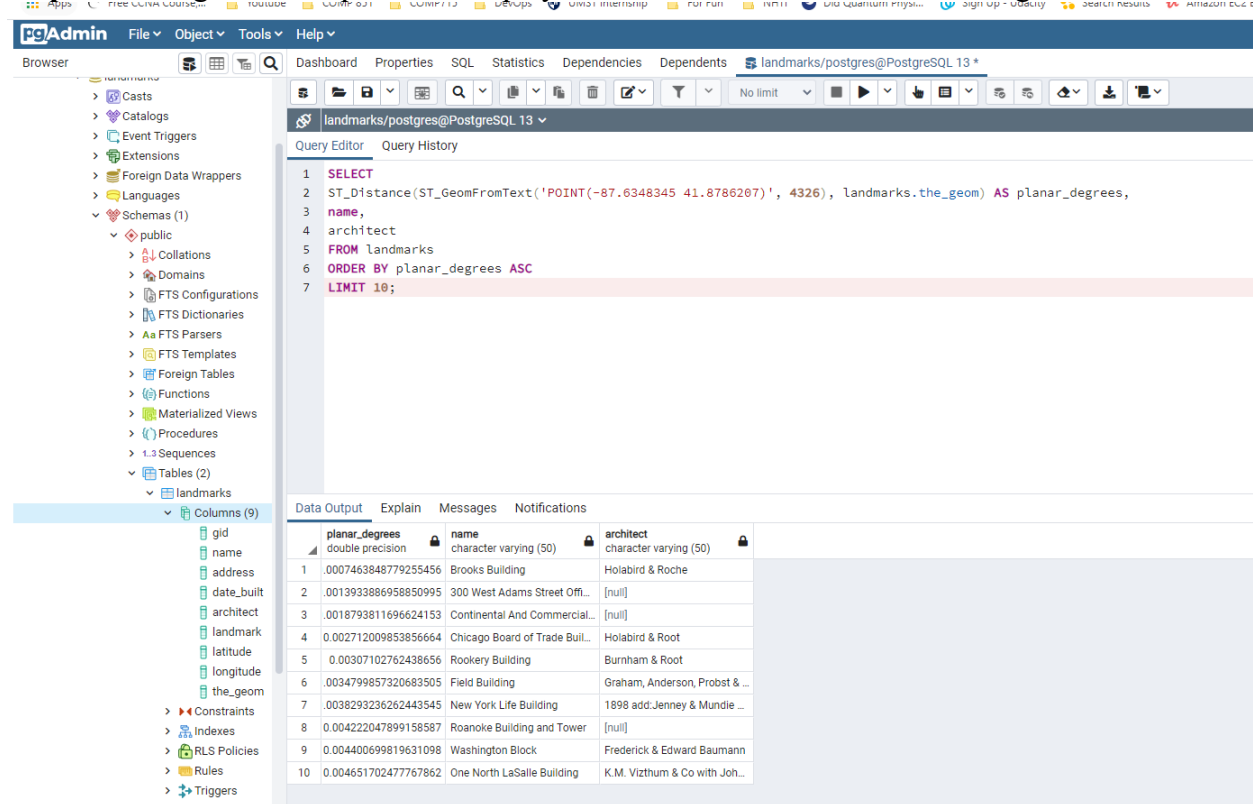
Data Output Explain Messages Notifications

	gid	name	address	date_built	architect	landmark	latitude	longitude	the_geom
1	L-265	Vassar Swiss Underwear Co...	2543-2545 W Diversey Av	[null]	[null]	07/30/2008	41.93162661	-87.6921001	
2	L-89	Mathilde Eliel House	4122 S Ellis Av	1886	Adler & Sullivan	10-02-91	41.81925575	-87.602788	
3	L-139	Manhattan Building	431 S Dearborn St	1891	William LeBaron Jenney	07-07-78	41.87606572	-87.62896445	
4	L-12	Machinery Hall at Illinois Ins...	100 W 33rd St	1901	Patton, Fisher & Miller	05/26/2004	41.83516141	-87.62922122	
5	L-88	Melissa Ann Elam House	4726 S Dr Martin Luther Kin...	1903	Henry L. Newhouse	03/21/1979	41.80852977	-87.61720439	
6	L-318	(Former) Pioneer Trust and ...	4000 W North Ave.	1924	Karl M. Vitzthum	06-06-12	41.91019211	-87.72661734	
7	L-85	DuPont-Whitehouse House	3558 S Artesian Av	1876	Oscar Cobb & Co.	04/16/1996	41.82858165	-87.68659368	
8	L-149	Montgomery Ward & Co. Cat...	618 W Chicago Av	1907-08	Richard E. Schmidt, Garden ...	05/17/2000	41.89743687	-87.64369542	
9	L-286	Vonwaerts Turner Hall	2431 W Roosevelt Rd	[null]	[null]	09-03-09	41.86615181	-87.68724694	
10	L-71	City Hall-County Building	121 N LaSalle St / 118 N Cla...	1905-08	Holabird and Roche	01/21/1982	41.88384254	-87.63165528	
11	L-119	Illinois and Michigan Canal	S Fork of Chicago River at W...	1845-48	[null]	05-09-96	41.84265823	-87.66461646	
12	L-242	Lake Shore & Michigan SouL...	Calumet River, N of 98th St &...	[null]	[null]	12-12-07	41.71968671	-87.54296094	
13	L-309	(Former) Schlitz Brewery Tie...	11400 S. Front Ave.	1906	Frommann and Jepsen	07-06-11	41.68712465	-87.61238874	

Step 5: Translate latitude and longitude into POINT geometry



Step 6: Writing PostGIS queries to display 10 location for this latitude and longitude



Using python for implementation:

```
C:\Users\Bishnu Poudyal\Desktop\Final_Project>python project.py
5 closest landmarks to -87.6348345 41.8786207
*****
Location-1
-----
Planar_Degrees - 0.0007463848779255456
Name - Brooks Building
Architect - Holabird & Roche
tuple index out of range
```

```
C:\Users\Bishnu Poudyal\Desktop\Final_Project>python project.py
relation "landmarks" already exists

PostgreSQL connection is closed
```

```
import psycopg2
import boto3
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT
import connection

try:
    #boto3 Got informtion from https://boto3.amazonaws.com/v1/documentation/api/latest/guide/sqs.html
    sqs = boto3.resource('sqs',aws_access_key_id = '',
                        aws_secret_access_key= '', region_name='us-west-2')
    queue = sqs.create_queue(QueueName='landmarks', Attributes={'DelaySeconds': '5'})

    #connecting to postgis https://www.postgresqltutorial.com/postgresql-python/connect/

    connection = psycopg2.connect(user="postgres",
                                   password="admin",
                                   host="127.0.0.1")

    connection.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT);
    cursor = connection.cursor()
    cursor.execute("drop database if exists landmarks;")
    createdatabase = """create database landmarks; """
    cursor.execute(createdatabase)
    connection.commit()

    #create extension postgis
```

```

create_extension_query_postgis = """create extension if not exists postgis;"""
cursor.execute(create_extension_query_postgis)
connection.commit()

#create tables and indexes in the databse called Landmark: using same SQL stat
ements used in PGAdmin
create_tables_landmarks = """ CREATE TABLE landmarks
(
gid character varying(5) NOT NULL,
name character varying(50),
address character varying(50),
date_built character varying(10),
architect character varying(50),
landmark character varying(10),
latitude double precision,
longitude double precision,
the_geom geometry,
CONSTRAINT landmarks_pkey PRIMARY KEY (gid),
CONSTRAINT enforce_dims_the_geom CHECK (st_ndims(the_geom) = 2),
CONSTRAINT enforce_geotype_geom CHECK (geometrytype(the_geom) = 'POINT'::text OR
the_geom IS NULL),
CONSTRAINT enforce_srid_the_geom CHECK (st_srid(the_geom) = 4326)
);
"""

cursor.execute(create_tables_landmarks)
connection.commit()
create_index = """ CREATE INDEX landmarks_the_geom_gist ON landmarks USING gist
(the_geom )"""
cursor.execute(create_index)
connection.commit()

#Copy the CSV data into the Database
insert_data = """copy landmarks(name,gid,address,date_built,architect,landmark
,latitude,longitude) FROM 'C:\\Users\\Bishnu Poudyal\\Desktop\\Final_Project\\Ind
ividual_Landmarks.csv' DELIMITERS ',' CSV HEADER """
cursor.execute(insert_data)
connection.commit()

#creating new message https://boto3.amazonaws.com/v1/documentation/api/latest/guide/sqs.html
response = queue.send_message(MessageBody='landmarks',MessageAttributes={
    'uploadmessage':{
        'StringValue':'Uploaded Successfully!!!',
        'DataType':'String'
    })
})

```

```

queue = sqs.get_queue_by_name(QueueName='landmarks')

#Translate latitude and longitude into POINT geometry
table_update = """UPDATE landmarks SET the_geom = ST_GeomFromText('POINT(' ||
longitude || ' ' || latitude || ')',4326) """
cursor.execute(table_update)
connection.commit()

#Writing PostGIS queries to display 10 location for this latitude and longitud
e https://www.postgresqltutorial.com/postgresql-python/query/

selece_queries = """SELECT distinct ST_Distance(ST_GeomFromText('POINT(-
87.6348345 41.8786207)', 4326), landmarks.the_geom) AS planar_degrees,
name,
architect
FROM landmarks
ORDER BY planar_degrees ASC
LIMIT 10 """
count = 1
cursor.execute(selece_queries)
connection.commit()
location_details=[]
records = cursor.fetchall()
print("5 closest landmarks to -87.6348345 41.8786207")
print(".....")
for row in records:
    print("Location-" + str(count))
    print(".....")
    print("Planar_Degrees - " + str(row[0]))
    print("Name - " + str(row[1]))
    print("Architect - " + str(row[2]))
    print("Latitude - " + str(row[3]))
    print("Longitude - " + str(row[4]))
    print(".....")
    count +=1
    location_details.append(str(row[0]))
    location_details.append(str(row[1]))
    location_details.append(str(row[2]))
    location_details.append(str(row[3]))
    location_details.append(str(row[4]))
    location_details.append(str(row[5]))
    location_details.append(str(row[6]))
    location_details.append(str(row[7]))
    location_details.append(str(row[8]))
    location_details.append(str(row[9]))

```



```
#sending location data to the queue
response = queue.send_message(MessageBody='landmarks',MessageAttributes={
    'Locations':{
        'StringValue':",".join(location_details),
        'DataType':'String'
    })
connection.commit()

except (Exception, psycopg2.Error) as error:
    if(connection):
        print(error)

finally:
    #closing database connection.
    if(connection):
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")
```