

# TERRASTACK INTERNSHIP ASSIGNMENT

Shreyas Katdare, 22B0636

31/03/2024

GitHub: [https://github.com/ShreyasKatdare/sawangi\\_terrastack.git](https://github.com/ShreyasKatdare/sawangi_terrastack.git)

Google Drive: [here](#)

## Problem 1

In this, we have to load the sawangi.shp shapefile into QGIS. So first I made new connection between sawangi.test database in pgAdmin application and QGIS and then added the sawangi.shp shapefile into QGIS.

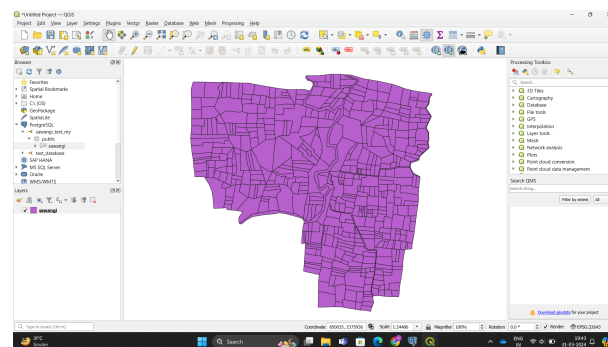


Figure 1: Sawangi Village Plot

I am utilizing pgAdmin to interact with SQL queries and facilitate connectivity between QGIS and the database. This involves employing SQL commands within pgAdmin to manage and query spatial data, enabling seamless integration with QGIS for geospatial analysis and visualization.

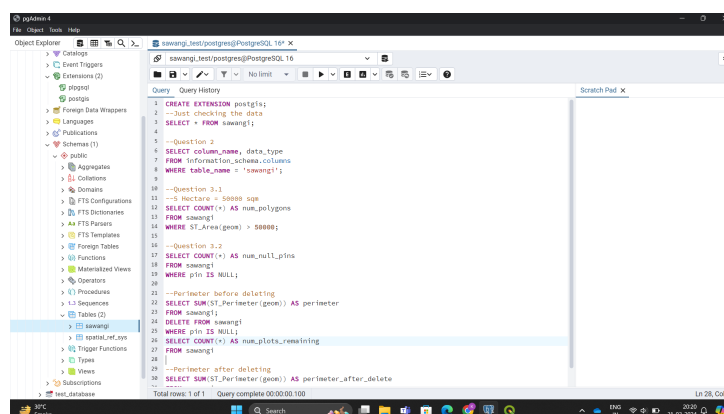


Figure 2: pgAdmin interface

## Problem 2

To get datatypes of all columns, we will use following query code

```
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'sawangi';
```

This gives following output :

Table 1: Datatype of columns

column_name	data_type
id	integer
geom	USER-DEFINED
gid	bigint
ccode	character varying
pin	character varying
dtncode	character varying
thncode	character varying
vincode	character varying
vil_name	character varying
dtname	character varying
thname	character varying
cncode11	character varying
cncode01	character varying
lgd_code	character varying
ef_code	character varying

In this id column has datatype of 'INTEGER', gid column has datatype 'BIGINT', geom column is user-defined datatype to store shapes like points, lines, or polygons, whereas other columns have datatype 'VARCHAR' to store string of variable length.

## Problem 3

### 3.1

To find number of polygons with an area greater than 5 Ha, we will use following query code :

```
--5 Hectare = 50000 sqm
SELECT COUNT(*) AS num_polygons
FROM sawangi
WHERE ST_Area(geom) > 50000;
```

Output :

	num_polygons bigint
1	30

Figure 3: Output for Problem 3.1

### 3.2

First let's check how many rows have pin as 'null'.

```
SELECT COUNT(*) AS num_null_pins
FROM sawangi
WHERE pin IS NULL;
```

Output :

	num_null_pins bigint
1	17

Figure 4: Output for Problem 3.2 : Number rows having null pin

Now let's delete these rows i.e. polygons corresponding to these rows :

```
DELETE FROM sawangi
WHERE pin IS NULL;
```

Output :

```
DELETE 17  
  
Query returned successfully in 83 msec.
```

Figure 5: Output for Problem 3.2 : Deleted rows

Now if we check remaining rows :

```
SELECT COUNT(*) AS num_plots_remaining  
FROM sawangi
```

Output :

	num_plots_remaining bigint
1	484

Figure 6: Output for Problem 3.2 : Number of remaining rows

Yeah ! At start there were 501 rows and we deleted 17 rows so indeed remaining rows are 484.

After deletion, the plot looks like this :

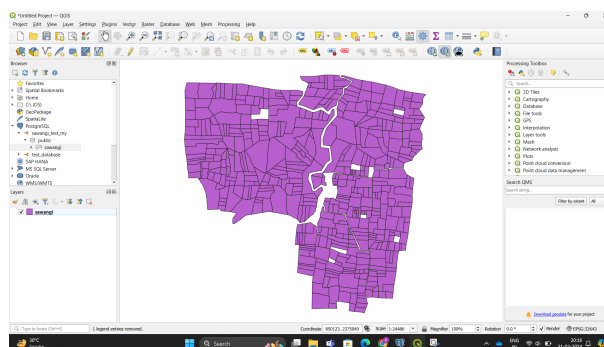


Figure 7: Output for Problem 3.2 : Sawangi plot after deletion

## Problem 3

Now let's calculate the outer total perimeter of the sawangi village. For this we will first take union of all polygons to make the polygon which encloses the village and then use ST\_perimeter to find its perimeter. Also we should do this before deleting the rows i.e. on original plot!

```
--sawangi_og is original table
WITH union_polygon AS (
    SELECT ST_Union(geom) AS geom
    FROM sawangi_og
)
-- Calculate perimeter of the union polygon
SELECT ST_Perimeter(geom) AS perimeter
FROM union_polygon;
```

Output:


	perimeter double precision 
1	15551.237334192108

Figure 8: Output for Problem 3.3 : Perimeter of Village

So the total perimeter of village is  $15551.2373 \text{ m}^2$