SGD LAB EXP – 3B

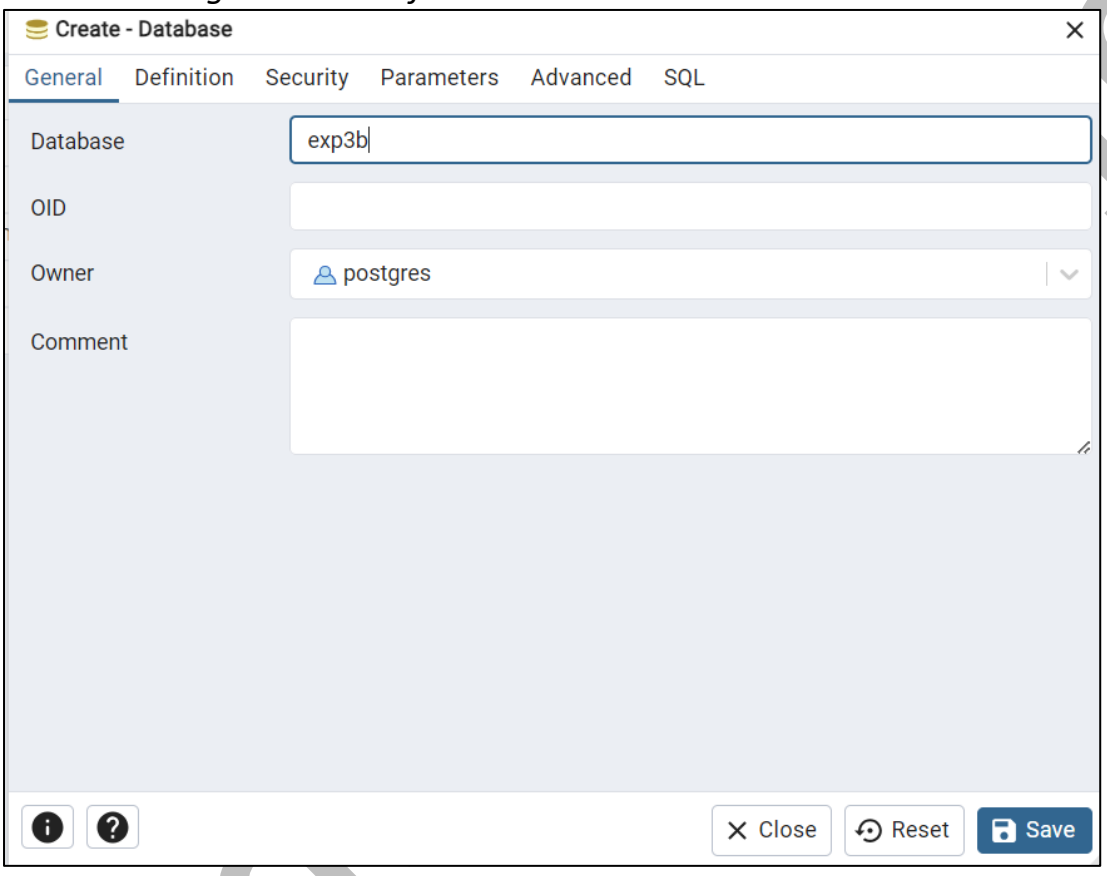
Name : Aditi Chhajed; **Reg. No.** : 221081009
Branch : IT ; **Course Instructor** : Prof. Vedashree Awati

Aim:

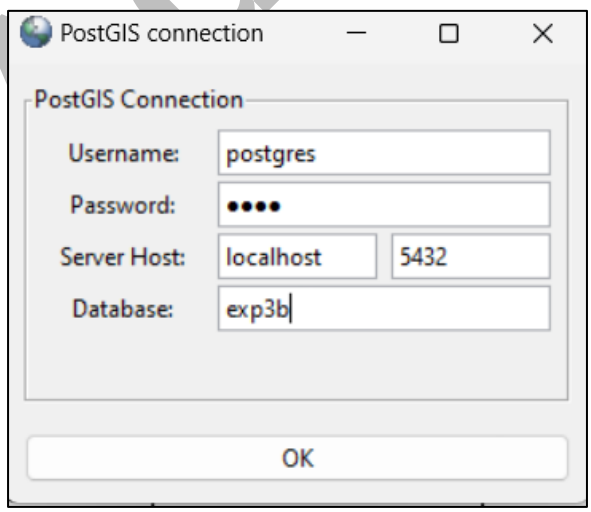
To write spatial queries using spatial functions like ST_X, ST_Y, ST_Touches,ST_Area etc.

Implementation:

1. *Creating a database first.*



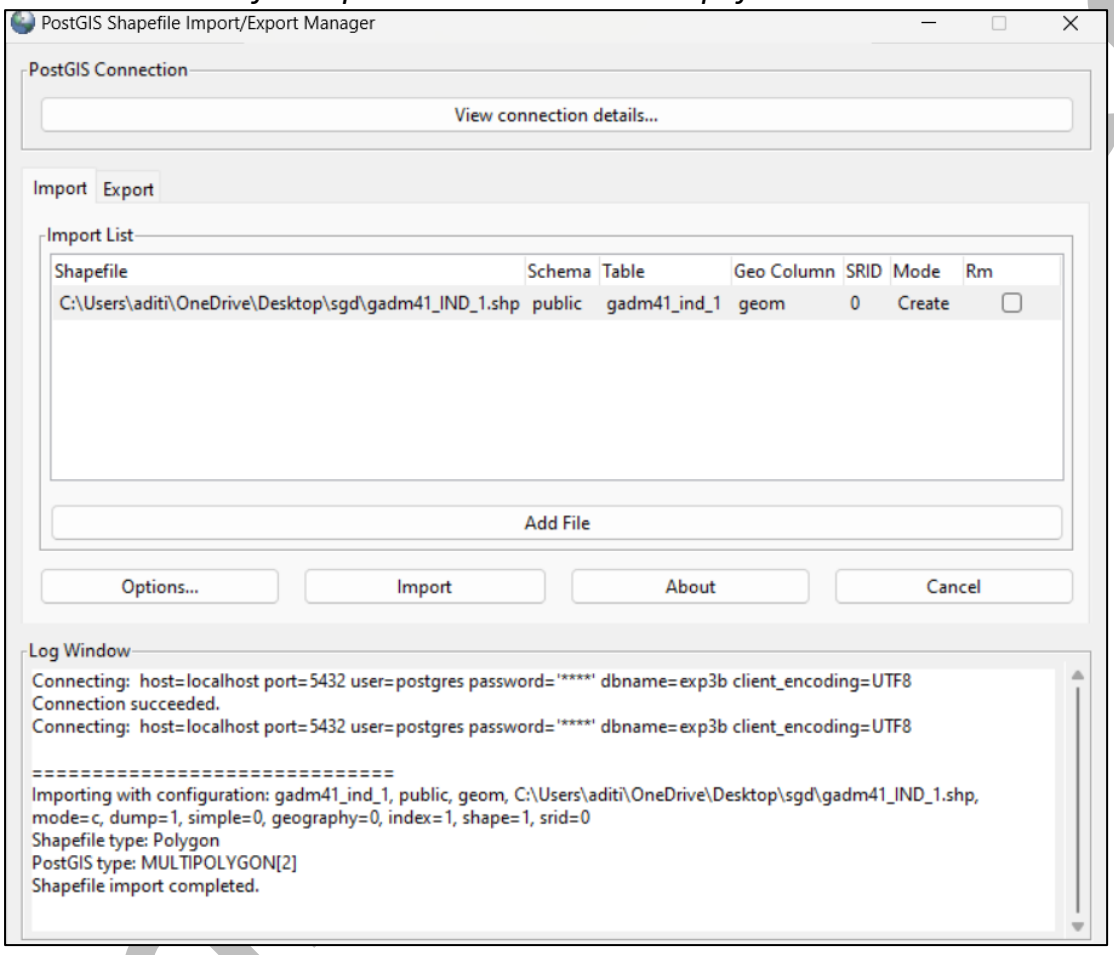
2. *Establish connection.*



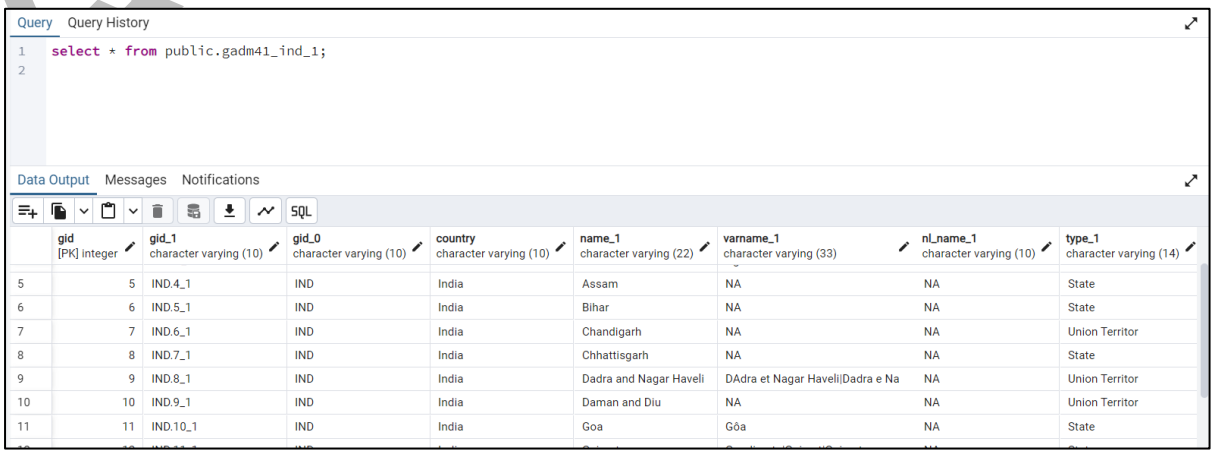
3. *Import Shapefiles*

Navigate to c:\Program Files\PostgreSQL\15\bin\postgis gui and run executable file shp2pgsql-gui.exe

- *Click on view connection details*
- *Enter username as postgres, password, server host as localhost and port as 5432 and database name is the same as that was created above in step A.1. After clicking on OK, it should give a message connection succeeded.*
- *Select the add files option and select the shapefile.*



Checking the loaded sql file.



	gid [PK] integer	gid_1 character varying (10)	gid_0 character varying (10)	country character varying (10)	name_1 character varying (22)	varname_1 character varying (33)	nl_name_1 character varying (10)	type_1 character varying (14)
5	5	IND.4_1	IND	India	Assam	NA	NA	State
6	6	IND.5_1	IND	India	Bihar	NA	NA	State
7	7	IND.6_1	IND	India	Chandigarh	NA	NA	Union Territor
8	8	IND.7_1	IND	India	Chhattisgarh	NA	NA	State
9	9	IND.8_1	IND	India	Dadra and Nagar Haveli	DAdra et Nagar Haveli|Dadra e Na	NA	Union Territor
10	10	IND.9_1	IND	India	Daman and Diu	NA	NA	Union Territor
11	11	IND.10_1	IND	India	Goa	Gôa	NA	State

Execution :

1. *Find names of all the states which are neighbors of Maharashtra.*

```
SELECT name_1 AS neighbor_state
FROM public.gadm41_ind_1
WHERE ST_Touches(
    (SELECT geom FROM public.gadm41_ind_1 WHERE name_1 = 'Maharashtra'),
    geom
);
```

Data Output | Messages | Notifications

	neighbor_state character varying (22)
1	Chhattisgarh
2	Telangana
3	Dadra and Nagar Haveli
4	Madhya Pradesh
5	Gujarat
6	Goa
7	Karnataka

2. *Find states that are ~~315~~ km 100km away from ~~Delhi~~ Maharashtra.*
   (changes made after discussing with the faculty)

Query | Query History

```
--we found that the srid is 0 and hence it throws some issues while calculating geometry.
SELECT Find_SRID('public', 'gadm41_ind_1', 'geom');

--hence we update the srid to 4326 and then have defined geom in metrs.
UPDATE gadm41_ind_1
SET geom = ST_SetSRID(geom, 4326);

--we have updated the place from "Delhi" to "Maharashtra" here as the dataset doesn't support Delhi's info
SELECT name_1 AS state_name
FROM gadm41_ind_1
WHERE ST_DWithin(
        (SELECT ST_Transform(geom, 3857) FROM gadm41_ind_1 WHERE name_1 = 'Maharashtra'),
        ST_Transform(geom, 3857),
        100000
    )
AND name_1 != 'Maharashtra';
```

Data Output | Messages | Notifications

	state_name character varying (22)
1	Chhattisgarh
2	Telangana
3	Dadra and Nagar Haveli
4	Daman and Diu
5	Goa
6	Gujarat
7	Karnataka
8	Madhya Pradesh

3. *Find the distance of all cities from ~~Kolkata~~ Daman and Diu in descending order.*
(changes made after discussing with the faculty)

```
18  SELECT name_1 AS state_name,
19         ST_Distance(
20             ST_Centroid(geom),
21             (SELECT ST_Centroid(geom) FROM gadm41_ind_1 WHERE name_1 = 'Daman and Diu')
22         ) AS distance_from_dnd
23  FROM gadm41_ind_1
24  ORDER BY distance_from_dnd DESC;
```

Data Output | Messages | Notifications

	state_name character varying (22)	distance_from_dnd double precision
1	Arunachal Pradesh	24.86107147388707
2	Arunachal Pradesh	23.363546536210677
3	Nagaland	22.87741259638031
4	Andaman and Nicobar	22.674737347827726
5	Manipur	22.01424906820657
6	Assam	21.366386728471802
7	Mizoram	20.746995424897467
8	Tripura	19.73025718352128
9	Meghalaya	19.65374159704829
10	Sikkim	17.668168994124457
11	West Bengal	16.04689531352605
12	Bihar	14.294924257869026
13	Jammu and Kashmir	13.944680267644795
14	Jharkhand	13.646738681882843
15	Himachal Pradesh	13.2122872577972

Total rows: 41 of 41 Query complete 00:00:00.170 Ln 17, Col 1

	state_name character varying (22)	distance_from_dnd double precision
16	Uttarakhand	12.896764249741233
17	Uttarakhand	12.734833339839856
18	Himachal Pradesh	12.656982478150793
19	Himachal Pradesh	12.445619198514498
20	Odisha	12.138872942038947
21	Uttarakhand	11.853731078450913
22	Tamil Nadu	11.30033345419305
23	Chandigarh	11.170078592176251
24	Kerala	10.86847509992806
25	Puducherry	10.850175565411762
26	Punjab	10.799369212023924
27	Uttar Pradesh	10.478714435269959
28	Lakshadweep	10.075898745024796
29	Chhattisgarh	9.785674521439518
30	Haryana	9.593061637824917
31	NCT of Delhi	9.465882364217851
32	Andhra Pradesh	9.025690695135804
33	Telangana	7.3020414219551535
34	Karnataka	6.982444889584799
35	Madhya Pradesh	6.733705307737485
36	Rajasthan	6.277110558654269
37	Goa	5.444407840311289

Total rows: 41 of 41 Query complete 00:00:00.170 Ln 17, Col 1

4. *Find the area of all the states.*

```
SELECT name_1 AS state_name,
       ST_Area(geom) AS area_sq_meters
FROM public.gadm41_ind_1;
```

Data Output | Messages | Notifications

	state_name character varying (22)	area_sq_meters double precision
1	Andaman and Nicobar	0.6304843464429322
2	Andhra Pradesh	13.498647313738594
3	Arunachal Pradesh	1.3589390771855903
4	Arunachal Pradesh	6.166371436309463
5	Assam	7.098459262903407
6	Bihar	8.47379541752592
7	Chandigarh	0.011125417065885554
8	Chhattisgarh	11.7900867251004
9	Dadra and Nagar Haveli	0.04247816031851931
10	Daman and Diu	0.007234084477863497
11	Goa	0.31107504350952814
12	Gujarat	16.323922794085572
13	Haryana	4.0845184419212845
14	Himachal Pradesh	5.263291861384931
15	Himachal Pradesh	0.029393484661076382
16	Himachal Pradesh	0.009527383580412786
17	Jammu and Kashmir	10.146449984888852
18	Jharkhand	7.084292207345808
19	Karnataka	16.066965210019028
20	Kerala	3.1136770874001374
21	Lakshadweep	0.002878582441590596
22	Madhya Pradesh	27.246918727389577
23	Maharashtra	26.447830329388093
24	Manipur	1.991971109085024
25	Meghalaya	2.022243362234796
26	Mizoram	1.8760775831937446
27	Nagaland	1.497437360590739
28	NCT of Delhi	0.13863507718670018
29	Odisha	13.479603734214425
30	Puducherry	0.045424446456520925
31	Punjab	4.736016969377465
32	Rajasthan	30.986406294624505
33	Sikkim	0.6473022273565309
34	Tamil Nadu	10.779729114117305
35	Telangana	9.776575621421395
36	Tripura	0.9331205383161846
37	Uttar Pradesh	21.882486533554182
38	Uttarakhand	4.922055228476776

5. *List all states along with the number of neighbors they have in descending order with respect to the number of neighboring states.*

```
26  SELECT s1.name_1 AS state_name,
27         COUNT(s2.name_1) AS neighbor_count
28  FROM public.gadm41_ind_1 s1
29  LEFT JOIN public.gadm41_ind_1 s2
30  ON ST_Touches(s1.geom, s2.geom) AND s1.name_1 <> s2.name_1
31  GROUP BY s1.name_1
32  ORDER BY neighbor_count DESC;
```

Data Output | Messages | Notifications

	state_name character varying (22)	neighbor_count bigint
1	Uttar Pradesh	9
2	Assam	8
3	Himachal Pradesh	7
4	Maharashtra	7
5	Haryana	6
6	Karnataka	6
7	Chhattisgarh	6
8	West Bengal	5
9	Andhra Pradesh	5
10	Gujarat	5
11	Jharkhand	5
12	Madhya Pradesh	5
13	Odisha	5
14	Punjab	5
15	Rajasthan	5
16	Telangana	5
17	Tamil Nadu	4
18	Uttarakhand	4
19	Manipur	3
20	Kerala	3
21	Bihar	3
22	Mizoram	3
23	Nagaland	3
24	Puducherry	3
25	Arunachal Pradesh	3
26	Tripura	2
27	Chandigarh	2
28	Goa	2
29	Dadra and Nagar Haveli	2
30	Jammu and Kashmir	2
31	NCT of Delhi	2
32	Daman and Diu	1
33	Meghalaya	1
34	Sikkim	1
35	Lakshadweep	0
36	Andaman and Nicobar	0

Conclusion:

This experiment highlights the importance of accurate spatial reference systems (SRIDs) and geometry handling when performing spatial analyses in PostgreSQL/PostGIS.

*1**. Spatial Reference System (SRID):** Initially, the geometries had an SRID of `0`, which meant they lacked a defined spatial reference. This led to inaccurate distance calculations, as the units were ambiguous. Assigning the appropriate SRID (`4326`) and transforming it to a meter-based SRID (`3857`) provided reliable units for distance measurements.*

*2. **Centroids vs. Full Geometry**: Using centroids for distance calculations initially provided an approximation, which may be sufficient for certain use cases. However, calculating distances based on the entire geometry (boundary-to-boundary) gave me more accurate results, especially when interested in distances between regions rather than single points within them.*

3. ***Choosing Appropriate Units**: By transforming the geometries to SRID `3857`, which uses meters, we could confidently set precise distance thresholds (like 100,000 meters for 100 km). This highlights the importance of choosing projections that match the measurement units required for spatial queries.*

Mistakes I made :

- Always define an SRID for geometries to avoid ambiguous calculations.

- Choose centroid-based or boundary-based distance calculations based on the specific analysis needs. I chose boundary-based here.

- Use an appropriate projection (e.g., SRID `3857` for meters) when performing measurements in specific units, ensuring accurate spatial analysis.