# Experiment No.5

## Title: Execution of Spatial database queries

**Batch: A2** **Roll No.: 16010421059** **Experiment No.:5**

**Aim: To execute spatial queries using PostGIS.**

---

**Resources needed:** PostgreSQL 9.6, PostGIS 2.0

---

**Theory**

**PostGIS** is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC). PostGIS turns the PostgreSQL Database Management System into a spatial database by adding support for the three features: spatial types, indexes, and functions. Because it is built on PostgreSQL, PostGIS automatically inherits important "enterprise" features as well as open standards for implementation. PostgreSQL is a powerful, object-relational database management system (ORDBMS). It is also open source software.

**Features of** PostGIS

- Geometry types for points, line strings, polygons, multi-points, multi-line-strings, multi-polygons and geometry collections.
- Spatial predicates for determining the interactions of geometries using the 3x3 Egenhofer matrix (provided by the GEOS software library).
- Spatial operators for determining geospatial measurements like area, distance, length and perimeter.
- Spatial operators for determining geospatial set operations, like union, difference, symmetric difference and buffers (provided by GEOS).
- R-tree-over-GiST (Generalised Search Tree) spatial indexes for high speed spatial querying.
- Index selectivity support, to provide high performance query plans for mixed spatial/non-spatial queries.
- For raster data

Geometry is and abstract type and concrete subtypes can be **atomic** or **collection** types

- Atomic
  - Point : It represents a single location in coordinate space
    e.g. POINT(3, 4), POINT (3,5,4,8)

  - LineString : It is a 1-dimensional line formed by a contiguous sequence of line segments. Each line segment is defined by two points, with the end point of one segment forming the start point of the next segment
    e.g. LINESTRING (1 2, 3 4, 5 6)

  - LineRing : It is a LineString which is both closed and simple. The first and last points must be equal, and the line must not self-intersect
    e.g. LINEARRING (0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0)

  - Polygon : It is a 2-dimensional planar region, delimited by an exterior boundary (the shell) and zero or more interior boundaries (holes). Each boundary is a LinearRing.
    e.g. POLYGON ((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))

- **Collection**

- MultiPoint : It is a collection of points
  e.g. MULTIPOINT ( (0 0), (1 2) )

- MultiLineString : It is a collection of LineStrings. A MultiLineString is closed if each of its elements is closed
  e.g. MULTILINESTRING ( (0 0,1 1,1 2), (2 3,3 2,5 4) )

- MultiPolygon : It is a collection of non-overlapping, non-adjacent polygons. Polygons in the collection may touch only at a finite number of points.
  e.g. MULTIPOLYGON (((1 5, 5 5, 5 1, 1 1, 1 5)), ((6 5, 9 1, 6 1, 6 5)))

- GeometryCollection : It is a is a heterogeneous (mixed) collection of geometries
  e.g. GEOMETRYCOLLECTION ( POINT(2 3), LINESTRING(2 3, 3 4))

- Also there are PolyHedralSurface, Triangle and TIN

PostGIS provides different functions for determining relationships(topological ordistance) between geometries, compute measurements, overlays and geometry construction also besides other provisions.

Few of the functions are

**Measurement functions**

ST_Area : `float ST_Area(geometry g1);`

Returns the area of a polygonal geometry

ST_Length : `float ST_Length(geometry a_2dlinestring);` R

Returns the 2D Cartesian length of the geometry if it is a LineString, MultiLineString, ST_Curve, ST_MultiCurve

ST_Perimeter : `float ST_Perimeter(geometry g1);`

Returns the 2D perimeter of the geometry/geography if it is a ST_Surface, ST_MultiSurface (Polygon, MultiPolygon)

**Named Spatial Relationships**

For determining common spatial relationships, OGC SFS defines a set of named spatial relationship predicates. PostGIS provides these as the functions

ST_Contains : `boolean ST_Contains(geometry geomA, geometry geomB);`
ST_Crosses : `boolean ST_Crosses(geometry g1, geometry g2);`
ST_Disjoint : `boolean ST_Disjoint( geometry A , geometry B );`
ST_Equals : `boolean ST_Equals(geometry A, geometry B);`
ST_Intersects : `boolean ST_Intersects( geometry geomA , geometry geomB );`
ST_Overlaps : `boolean ST_Overlaps(geometry A, geometry B);`
ST_Touches : `boolean ST_Touches(geometry A, geometry B);`
ST_Within. : `boolean ST_Within(geometry A, geometry B);`

It also defines the non-standard relationship predicates
ST_Covers : `boolean ST_Covers(geometry geomA, geometry geomB);`
ST_CoveredBy : `boolean ST_CoveredBy(geometry geomA, geometry geomB);`
ST_ContainsProperly : `boolean ST_ContainsProperly(geometry geomA, geometry geomB);`

Spatial predicates are usually used as conditions in SQL WHERE or JOIN clauses.

```
SELECT city.name, state.name, city.geom
```

```
FROM city JOIN state ON ST_Intersects(city.geom, state.geom);
```

---

**Procedure:**

1. Installation of relational database PostgreSQL 9.6 (download from http://www.enterprisedb.com/products-services-training/pgdownload )
2. Installation of PostGIS using Application stack builder.
3. Download spatial data from **`https://www.diva-gis.org/gdata`** (OR similar website with FREE usable data) Get it for any country with minimum 3 subjects.
4. Import the data in your PostgreSQL
5. Identify spatial relationship between any two geometric entities (any 3 named relationships)
6. Perform any two measurement functions for geometric data.
7. Execute any one range query

```
SELECT ST_Distance(geom, 'SRID=3005;POINT(1011102 450541)'::geometry) as d,edabbr,
vaabbr

FROM va2005

ORDER BY d limit 10;


        d         | edabbr | vaabbr
------------------+--------+--------
                0 | ALQ    | 128
 5541.57712511724 | ALQ    | 129A
 5579.67450712005 | ALQ    | 001
  6083.4207708641 | ALQ    | 131
  7691.2205404848 | ALQ    | 003
 7900.75451037313 | ALQ    | 122
 8694.20710669982 | ALQ    | 129B
 9564.24289057111 | ALQ    | 130
  12089.665931705 | ALQ    | 127
 18472.5531479404 | ALQ    | 002
(10 rows)
```

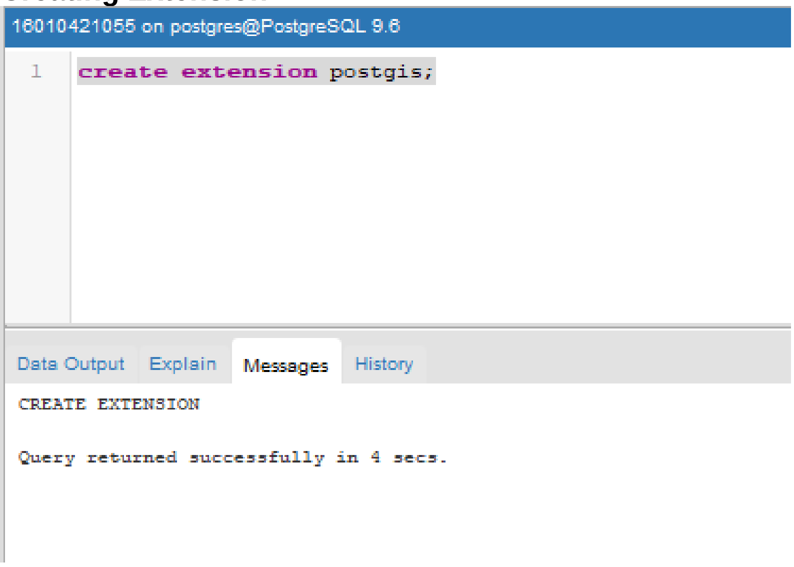
**Range query in Postgis**

**SELECT ST_Reclass(rast, 1,**
'[0-90]:0,(90-100):1,[100-1000):2',
'4BUI', 0) AS rast FROM sometable
WHERE filename = '123.tif';

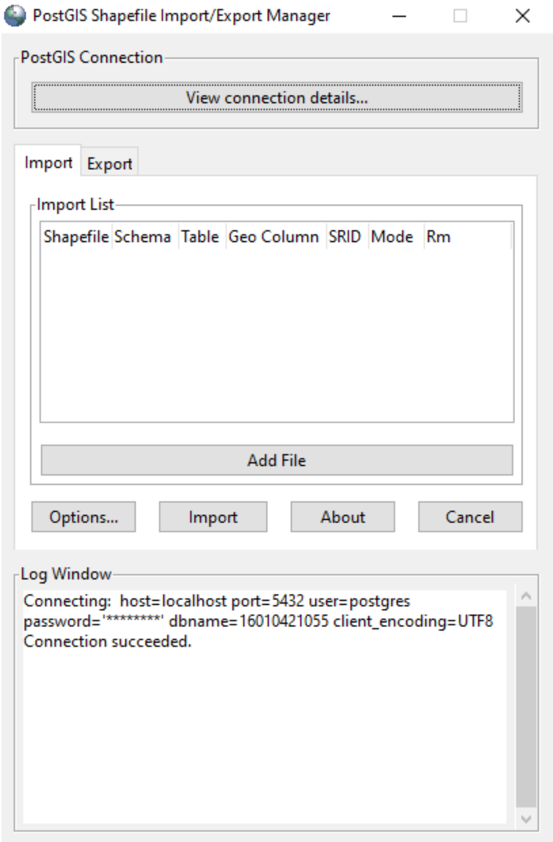
---

**Results: (Program printout with output)**

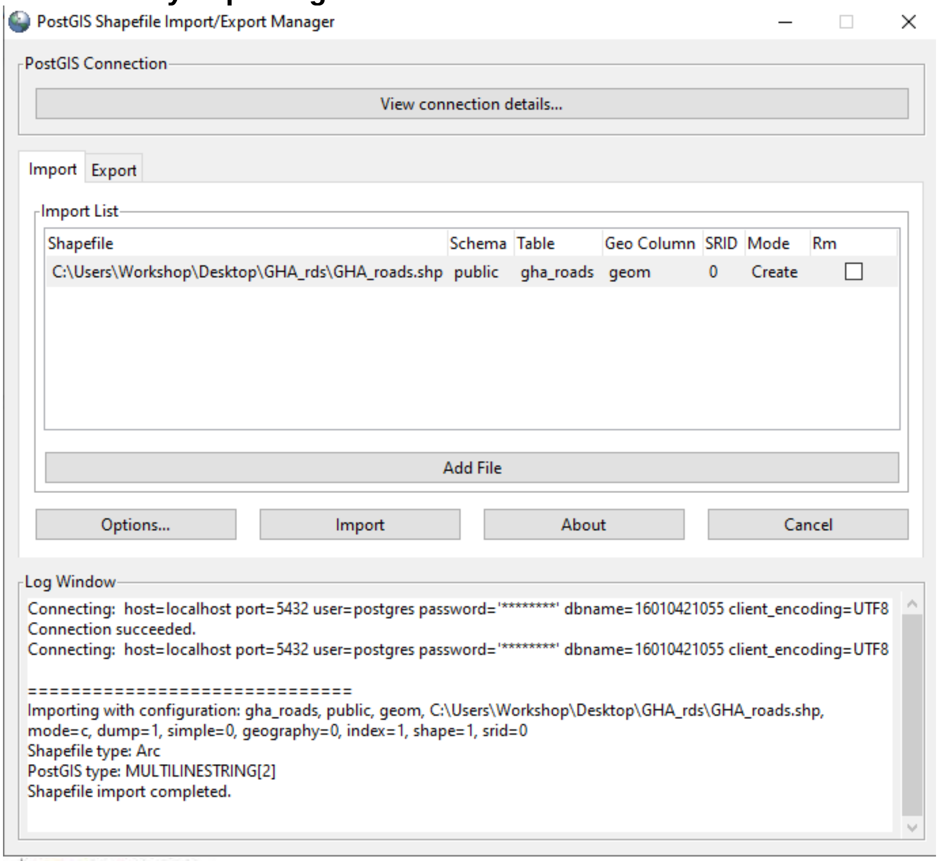
**Creating Extension-**

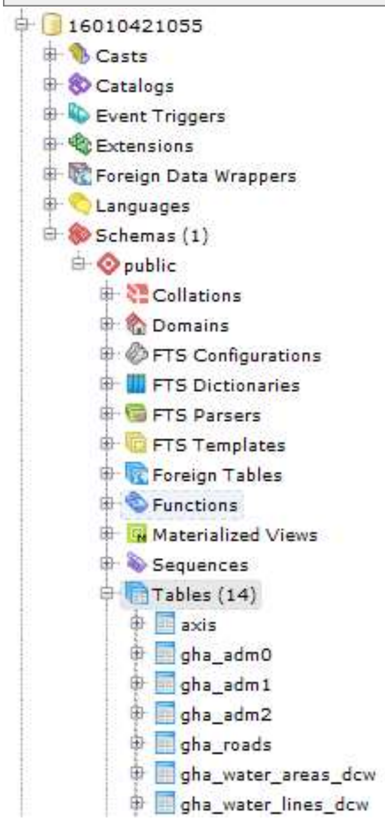



**Connecting to the database-**



## Successfully importing the files



## SELECTING TABLE-

```
select * from gha_roads
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying (5 |
|---|---|---|---|---|---|
| 1 | Without Median | Secondary Route | Road | GHA | GHANA |
| 2 | Without Median | Secondary Route | Road | GHA | GHANA |
| 3 | Unknown | Unknown | Road | GHA | GHANA |
| 4 | Without Median | Secondary Route | Road | GHA | GHANA |
| 5 | Without Median | Secondary Route | Road | GHA | GHANA |
| 6 | Without Median | Secondary Route | Road | GHA | GHANA |
| 7 | Without Median | Secondary Route | Road | GHA | GHANA |
| 8 | Without Median | Secondary Route | Road | GHA | GHANA |
| 9 | Without Median | Secondary Route | Road | GHA | GHANA |
| 10 | Without Median | Secondary Route | Road | GHA | GHANA |
| 11 | With Median | Primary Route | Road | GHA | GHANA |
| 12 | Without Median | Secondary Route | Road | GHA | GHANA |
| 13 | Without Median | Secondary Route | Road | GHA | GHANA |
| 14 | Without Median | Secondary Route | Road | GHA | GHANA |
| 15 | Without Median | Secondary Route | Road | GHA | GHANA |
| 16 | Without Median | Secondary Route | Road | GHA | GHANA |
| 17 | Without Median | Secondary Route | Road | GHA | GHANA |
| 18 | Without Median | Secondary Route | Road | GHA | GHANA |
| 19 | Without Median | Secondary Route | Road | GHA | GHANA |
| 20 | Without Median | Secondary Route | Road | GHA | GHANA |
| 21 | Without Median | Secondary Route | Road | GHA | GHANA |
| 22 | Without Median | Secondary Route | Road | GHA | GHANA |
| 23 | Without Median | Secondary Route | Road | GHA | GHANA |
| 24 | Without Median | Secondary Route | Road | GHA | GHANA |
| 25 | With Median | Primary Route | Road | GHA | GHANA |

16010421055 on postgres@PostgreSQL 9.6

```
select * from gha_water_areas_dcw;

select * from gha_roads,gha_water_areas_dcw where ST_Contains( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | iso character varying (7) | country character varying (54) | f_code_des character varying (254) | hyc_descri character varying (254) | name character varying (2 |
|---|---|---|---|---|---|
| 1 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 2 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 3 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 4 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 5 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 6 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 7 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 8 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 9 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 10 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 11 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 12 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 13 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 14 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |
| 15 | GHA | Ghana | Land Subject to Inundation | Non-Perennial/Intermittent/Fluctu... | [null] |

16010421055 on postgres@PostgreSQL 9.6

```
select * from gha_adm0;

select * from gha_roads,gha_water_areas_dcw where ST_Contains( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | id_0 bigint | iso character varying (3) | name_0 character varying (75) | objectid_1 bigint | iso3 character varying (3) | name_engli character varying (50) |
|---|---|---|---|---|---|---|
| 1 | 87 | GHA | Ghana | 88 | GHA | Ghana |

## ST_Area-

```
select ST_Area(gha_roads.geom) from gha_roads;
```

Data Output | Explain | Messages | History

| st_area double p... |
|---|
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |

## ST_Length-

```
select ST_Length(gha_roads.geom) from gha_roads;
```

Data Output | Explain | Messages | History

| st_length double p... |
|---|
| 06678817 |
| 08797219 |
| 22496945 |
| 75830811 |
| 19988311 |
| 48594733 |
| 57070169 |
| 77791647 |
| 74252307 |
| 70999496 |
| 96747563 |

## ST_perimeter-

```sql
select ST_Perimeter(gha_roads.geom) from gha_roads;
```

Data Output | Explain | Messages | History

| st_perim... double p... |
|---:|
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |

## ST_intersection-

```sql
select * from gha_roads,gha_water_areas_dcw where ST_Intersects( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying (5 |
|---:|---|---|---|---|---|
| 160 | Without Median | Secondary Route | Road | GHA | GHANA |
| 160 | Without Median | Secondary Route | Road | GHA | GHANA |
| 208 | Without Median | Secondary Route | Road | GHA | GHANA |
| 161 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 243 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 241 | Without Median | Secondary Route | Road | GHA | GHANA |
| 241 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 237 | Without Median | Secondary Route | Road | GHA | GHANA |

## ST_equals-

10010421055 on postgres@PostgreSQL 9.6

```
select * from gha_roads;

select * from gha_roads,gha_water_areas_dcw where ST_Equals( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying (5 |
|---|---|---|---|---|---|

## ST_crosses-

```
select * from gha_roads,gha_water_areas_dcw where ST_Crosses( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying |
|---|---|---|---|---|---|
| 160 | Without Median | Secondary Route | Road | GHA | GHANA |
| 160 | Without Median | Secondary Route | Road | GHA | GHANA |
| 208 | Without Median | Secondary Route | Road | GHA | GHANA |
| 161 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 209 | Without Median | Secondary Route | Road | GHA | GHANA |
| 243 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 221 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 241 | Without Median | Secondary Route | Road | GHA | GHANA |
| 241 | Without Median | Secondary Route | Road | GHA | GHANA |
| 227 | Without Median | Secondary Route | Road | GHA | GHANA |
| 237 | Without Median | Secondary Route | Road | GHA | GHANA |
| 241 | Without Median | Secondary Route | Road | GHA | GHANA |
| 236 | Without Median | Secondary Route | Road | GHA | GHANA |
| 237 | Without Median | Secondary Route | Road | GHA | GHANA |
| 243 | Without Median | Secondary Route | Road | GHA | GHANA |
| 249 | Without Median | Secondary Route | Road | GHA | [illegible] |
| 293 | Without Median | Secondary Route | Road | GHA | [illegible] |
| 281 | Without Median | Secondary Route | Road | GHA | [illegible] |

Total query runtime: 815 msec.
147 rows retrieved.

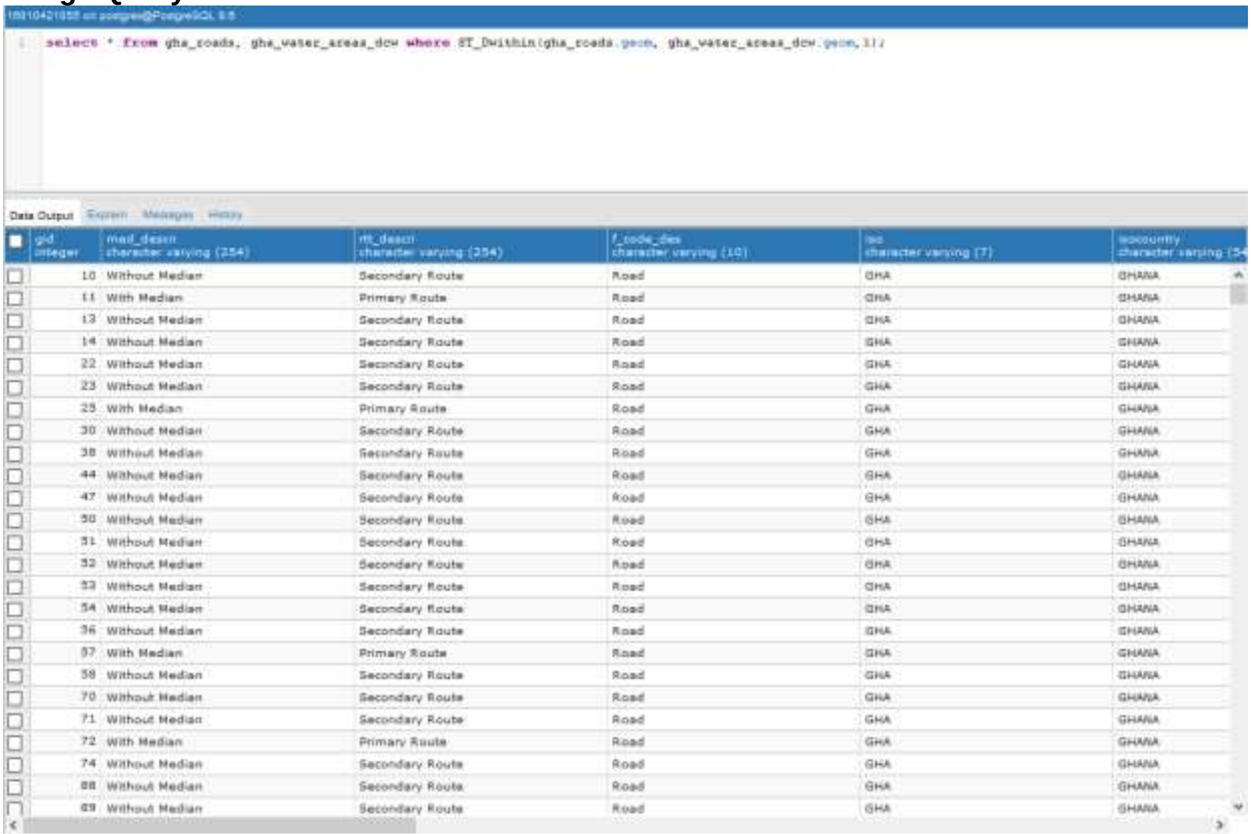
## ST_contains-

```
select * from gha_roads,gha_water_areas_dcw where ST_Contains( gha_roads.geom,gha_water_areas_dcw.geom);
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying (5 |
|---|---|---|---|---|---|

**Range Query-**



18010421055 on postgres@PostgreSQL 9.6

```
1 select * from gha_roads, gha_water_areas_dcw where ST_Dwithin(gha_roads.geom, gha_water_areas_dcw.geom,1);
```

Data Output | Explain | Messages | History

| gid integer | med_descri character varying (254) | rtt_descri character varying (254) | f_code_des character varying (10) | iso character varying (7) | isocountry character varying (54 |
|---|---|---|---|---|---|
| 10 | Without Median | Secondary Route | Road | GHA | GHANA |
| 11 | With Median | Primary Route | Road | GHA | GHANA |
| 13 | Without Median | Secondary Route | Road | GHA | GHANA |
| 14 | Without Median | Secondary Route | Road | GHA | GHANA |
| 22 | Without Median | Secondary Route | Road | GHA | GHANA |
| 23 | Without Median | Secondary Route | Road | GHA | GHANA |
| 25 | With Median | Primary Route | Road | GHA | GHANA |
| 30 | Without Median | Secondary Route | Road | GHA | GHANA |
| 38 | Without Median | Secondary Route | Road | GHA | GHANA |
| 44 | Without Median | Secondary Route | Road | GHA | GHANA |
| 47 | Without Median | Secondary Route | Road | GHA | GHANA |
| 50 | Without Median | Secondary Route | Road | GHA | GHANA |
| 51 | Without Median | Secondary Route | Road | GHA | GHANA |
| 52 | Without Median | Secondary Route | Road | GHA | GHANA |
| 53 | Without Median | Secondary Route | Road | GHA | GHANA |
| 54 | Without Median | Secondary Route | Road | GHA | GHANA |
| 56 | Without Median | Secondary Route | Road | GHA | GHANA |
| 57 | With Median | Primary Route | Road | GHA | GHANA |
| 58 | Without Median | Secondary Route | Road | GHA | GHANA |
| 70 | Without Median | Secondary Route | Road | GHA | GHANA |
| 71 | Without Median | Secondary Route | Road | GHA | GHANA |
| 72 | With Median | Primary Route | Road | GHA | GHANA |
| 74 | Without Median | Secondary Route | Road | GHA | GHANA |
| 88 | Without Median | Secondary Route | Road | GHA | GHANA |
| 89 | Without Median | Secondary Route | Road | GHA | GHANA |

**Q: Show Gid, Roads Description and distance from Roads and water areas for all the roads that are within a distance of 100m from water bodies-**

**QUERY-select gha_roads.gid as roads_gid,**
**gha_roads.rtt_descri as roads_description,**
**ST_distance(gha_roads.geom, gha_water_areas_dcw.geom) as distance**
**from gha_roads, gha_water_areas_dcw**
**where ST_Dwithin(gha_roads.geom, gha_water_areas_dcw.geom,100);**

**OUTPUT-**
**(pic inserted below)**

```sql
select gha_roads.gid as roads_gid,
gha_roads.rtt_descri as roads_description,
ST_distance(gha_roads.geom, gha_water_areas_dcw.geom) as distance
from gha_roads, gha_water_areas_dcw
where ST_Dwithin(gha_roads.geom, gha_water_areas_dcw.geom,100);
```

Data Output | Explain | Messages | History

| roads_gid integer | roads_description character varying (254) | distance double p... |
|---|---|---|
| 7 | Secondary Route | .80175318 |
| 10 | Secondary Route | 57181436 |
| 11 | Primary Route | 29623645 |
| 13 | Secondary Route | 74125838 |
| 14 | Secondary Route | 74125838 |
| 15 | Secondary Route | 31335791 |
| 17 | Secondary Route | 50771851 |
| 18 | Secondary Route | 24488046 |
| 22 | Secondary Route | 58958502 |
| 23 | Secondary Route | 51906279 |
| 24 | Secondary Route | 55770903 |
| 25 | Primary Route | 28282816 |
| 27 | Secondary Route | 67117604 |
| 28 | Unknown | 55770903 |
| 30 | Secondary Route | 75593641 |
| 31 | Unknown | 18287623 |
| 32 | Unknown | 97577602 |
| 33 | Secondary Route | 44050546 |
| 34 | Unknown | 05345383 |
| 35 | Secondary Route | 97577602 |
| 36 | Secondary Route | 91465738 |
| 38 | Secondary Route | 17403521 |

## Questions:

### 1. Explain the spatial functions used for these queries in detail.

**Ans:** ST_Contains: This function is used to determine whether one geometry (such as a polygon or point) is completely contained within another geometry. For example, you could use this function to find all the cities that are completely contained within a particular state.

ST_Intersects: This function is used to determine whether two geometries intersect (overlap). For example, you could use this function to find all the roads that intersect a particular river.

ST_Distance: This function is used to calculate the distance between two geometries. For example, you could use this function to find the distance between two points on a map.

ST_Area: This function is used to calculate the area of a geometry. For example, you could use this function to find the area of a polygon representing a park or a city.

ST_Length: This function is used to calculate the length of a geometry. For example, you could use this function to find the length of a line representing a road or a river.

ST_Crosses: This function is used to determine whether two geometries cross each other. For example, you could use this function to find all the roads that cross a particular river.

ST_Contains: This function is used to determine whether one geometry is completely contained within another geometry. For example, you could use this function to find all the cities that are completely contained within a particular state.

ST_Equals: This function is used to determine whether two geometries are equal. For example, you could use this function to find all the properties that have the exact same shape as a particular building.

2. **Explain any two applications of spatial database.**

**Ans: GIS (Geographic Information Systems):** GIS is a system designed to capture, store, manipulate, analyze, and display spatial or geographic data. GIS is used in a variety of fields, including urban planning, environmental science, and public health. Spatial databases are a critical component of GIS because they allow for the storage and efficient retrieval of large amounts of spatial data. For example, a city government might use a GIS to store information about zoning regulations, transportation infrastructure, and public services, which can then be used to make informed decisions about land use planning and urban development.

**Logistics and Supply Chain Management:** Spatial databases can be used to optimize logistics and supply chain management by analyzing the spatial relationships between various components of the system. For example, a company might use a spatial database to store information about the locations of its warehouses, suppliers, and customers, as well as data about transportation routes, traffic patterns, and delivery times. This information can then be used to optimize delivery routes, reduce transportation costs, and improve overall efficiency. Spatial databases can also be used to analyze the impact of factors such as weather patterns or natural disasters on the supply chain, allowing companies to respond quickly and effectively to disruptions.

## Outcomes:

**CO 2.** Design advanced database systems using Object relational, Spatial and NOSQL databases and its implementation.

## Conclusion: (Conclusion to be based on outcomes achieved)

Through this experiment, we were able to understand how to execute spatial database queries using the PostGIS software program.

1. Elmasri and Navathe, “Fundamentals of Database Systems”, Pearson Education
2. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems” 3rd Edition, McGraw Hill,2002
3. Korth, Silberchatz, Sudarshan, “Database System Concepts” McGraw Hill
4. http://www.bostongis.com/PrinterFriendly.aspx?content_name=postgis_tut01