Experiment No.5

Title: Execution of Spatial database queries

Batch: B-4 **Roll No.: 16010422234** **Name: Chandana Ramesh Galgali**

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 or distance) between geometries, compute measurements, overlays and geometry construction 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
8. Create account on GEE using Somaiya email id
9. Upload the shapefile
10. Visualize it on GEE
    Access the video resources **GE_shapefile-SMP.mp4** and **shptocsv,mp4** from https://drive.google.com/drive/folders/1jB7t4zVtyANA70XfHiwF2qU_JSMlU1_n?usp=drive_link

var shapefile : Table `projects/ee-suchitrapatil23/assets/indian_waterbodies_area//` this you need to do for your shapefile in assets

```
// Load the shapefile as a FeatureCollection.
var shapefile = shapefileUrl;

// Display the shapefile on the map.
Map.centerObject(shapefile, 10); // Center the map on the shapefile
Map.addLayer(shapefile, {}, 'Shapefile');

// Print the FeatureCollection to the console to inspect its properties.
print(shapefile);

// Convert the shapefile to a CSV table.
var csvTable = shapefile;
// Print the resulting CSV table.
print(csvTable);

// Export the CSV table to Google Drive.
Export.table.toDrive({
  collection: csvTable,
  description: 'indian_waterbodies_shapefilecsv',
  fileFormat: 'CSV'
});
```

```
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)

```
create extension postgis;
select * from USA_water_areas_dcw;
select * from USA_roads;
select * from USA_adm1;
select * from USA_roads, USA_water_areas_dcw where
st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
select * from USA_adm1, USA_roads where st_within(USA_adm1.geom,
USA_roads.geom);
select * from USA_adm1, USA_water_areas_dcw where
st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and
USA_water_areas_dcw.gid<10;
select st_length(geom) from USA_roads;
select st_area(geom) from USA_water_areas_dcw;
select * from USA_adm1, USA_roads where st_Dwithin(USA_adm1.geom,
USA_roads.geom) and USA_roads.gid<10;
```

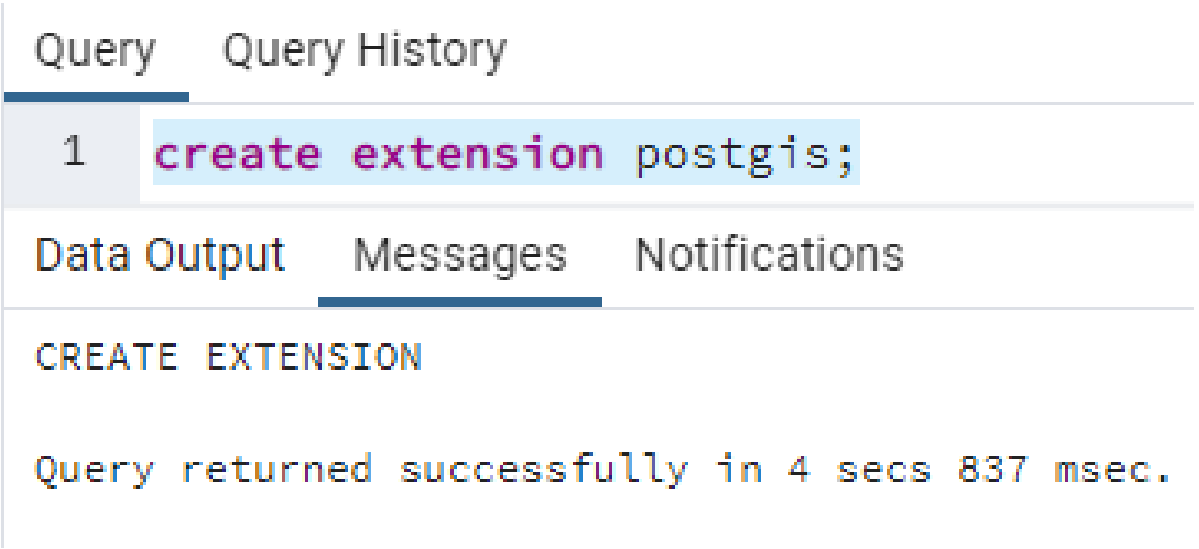
Query Query History

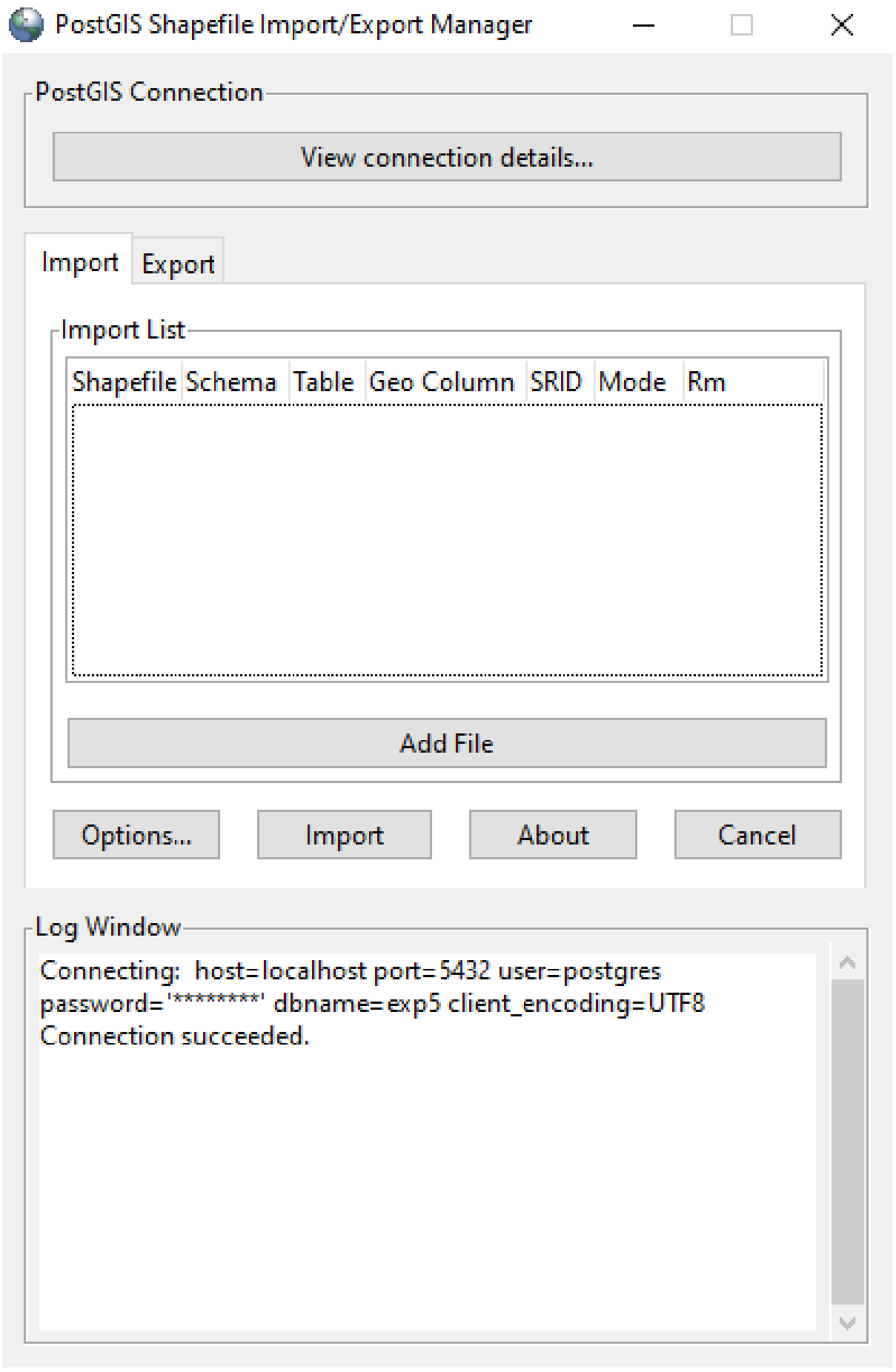
```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
6  select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
7  select * from USA_adm1, USA_water_areas_dcw where st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and USA_water_areas_dcw.gid<10;
8  select st_length(geom) from USA_roads;
9  select st_area(geom) from USA_water_areas_dcw;
10 select * from USA_adm1, USA_roads where st_Dwithin(USA_adm1.geom, USA_roads.geom, 1) and USA_roads.gid<10;
```

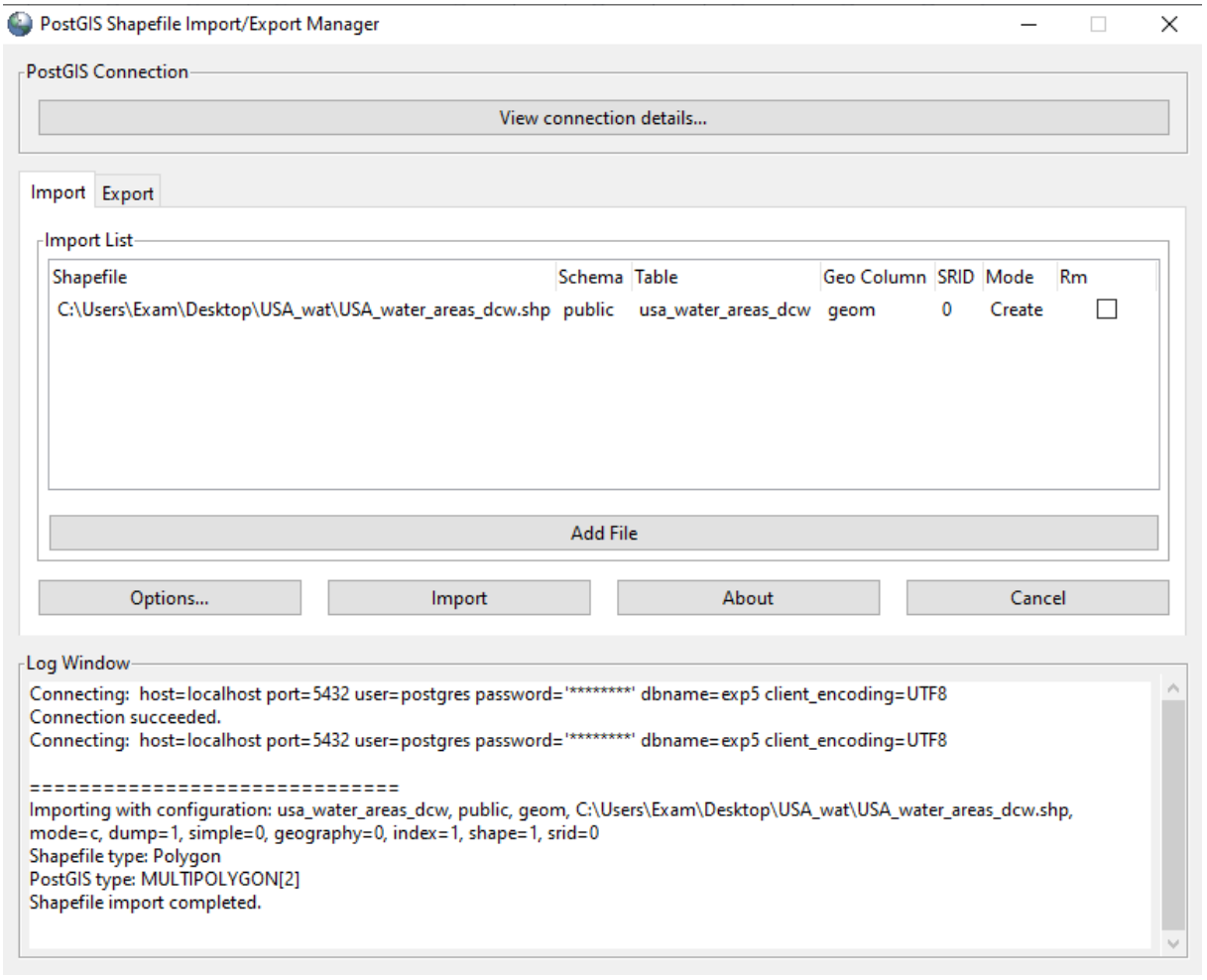
Data Output Messages Notifications

	gid integer	id_0 double precision	iso character varying (3)	name_0 character varying (75)	id_1 double precision	name_1 character varying (75)	type_1 character varying (50)	engtype_1 character varying (50)	nl_name_1 character varyin
1	2	244	USA	United States	2	Alaska	State	State	[null]
2	3	244	USA	United States	2	Alaska	State	State	[null]
3	2	244	USA	United States	2	Alaska	State	State	[null]
4	3	244	USA	United States	2	Alaska	State	State	[null]
5	2	244	USA	United States	2	Alaska	State	State	[null]
6	3	244	USA	United States	2	Alaska	State	State	[null]
7	2	244	USA	United States	2	Alaska	State	State	[null]
8	3	244	USA	United States	2	Alaska	State	State	[null]
9	2	244	USA	United States	2	Alaska	State	State	[null]
10	3	244	USA	United States	2	Alaska	State	State	[null]
11	2	244	USA	United States	2	Alaska	State	State	[null]
12	3	244	USA	United States	2	Alaska	State	State	[null]

Total rows: 18 of 18 Query complete 00:00:05.772 Ln 10, Col 107




PostGIS Shapefile Import/Export Manager
PostGIS Connection
View connection details...
Import
Export
Import List
Shapefile
Schema
Table
Geo Column
SRID
Mode
Rm
Add File
Options...
Import
About
Cancel
Log Window
Connecting: host=localhost port=5432 user=postgres password='********' dbname=exp5 client_encoding=UTF8
Connection succeeded.



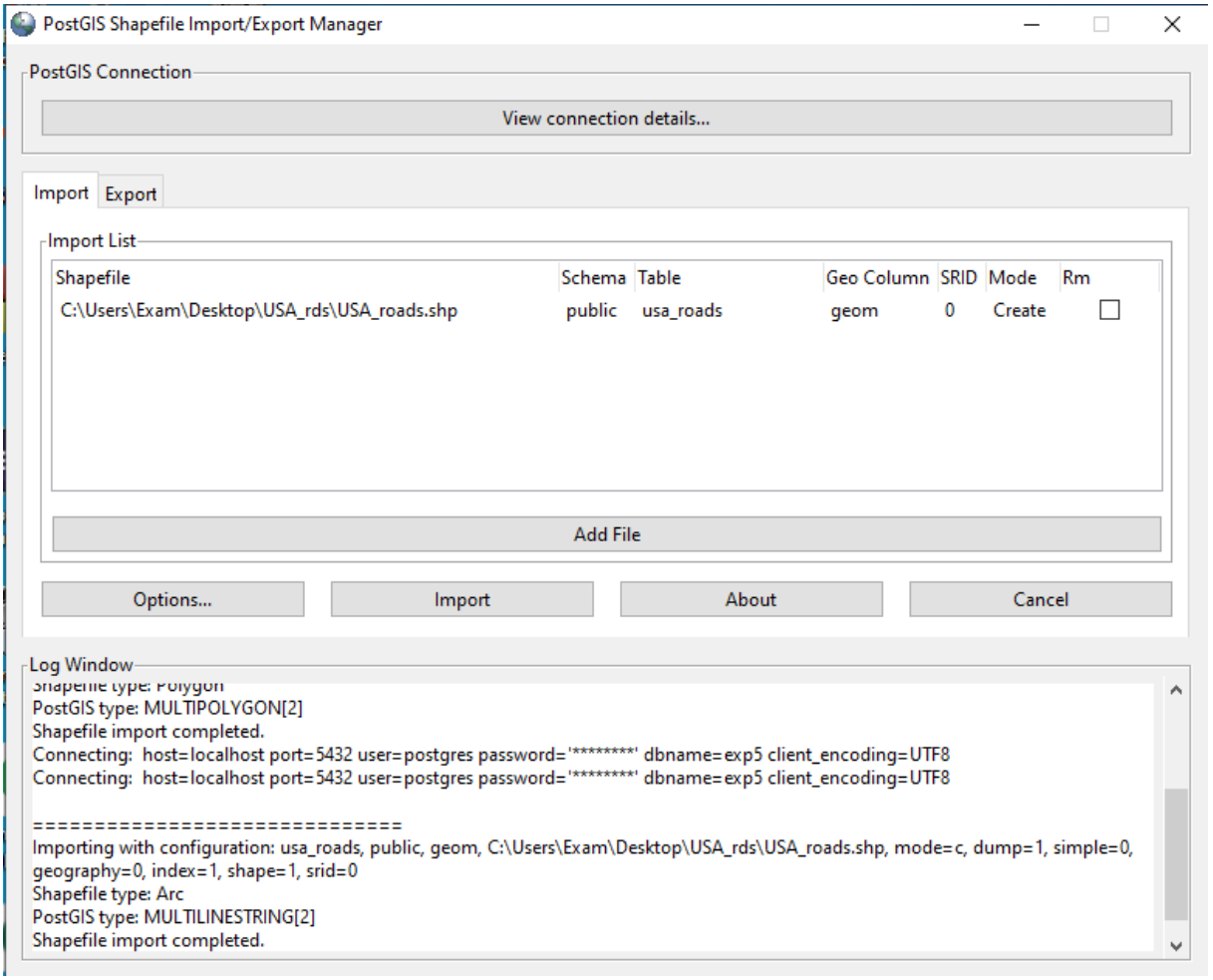
Query | Query History

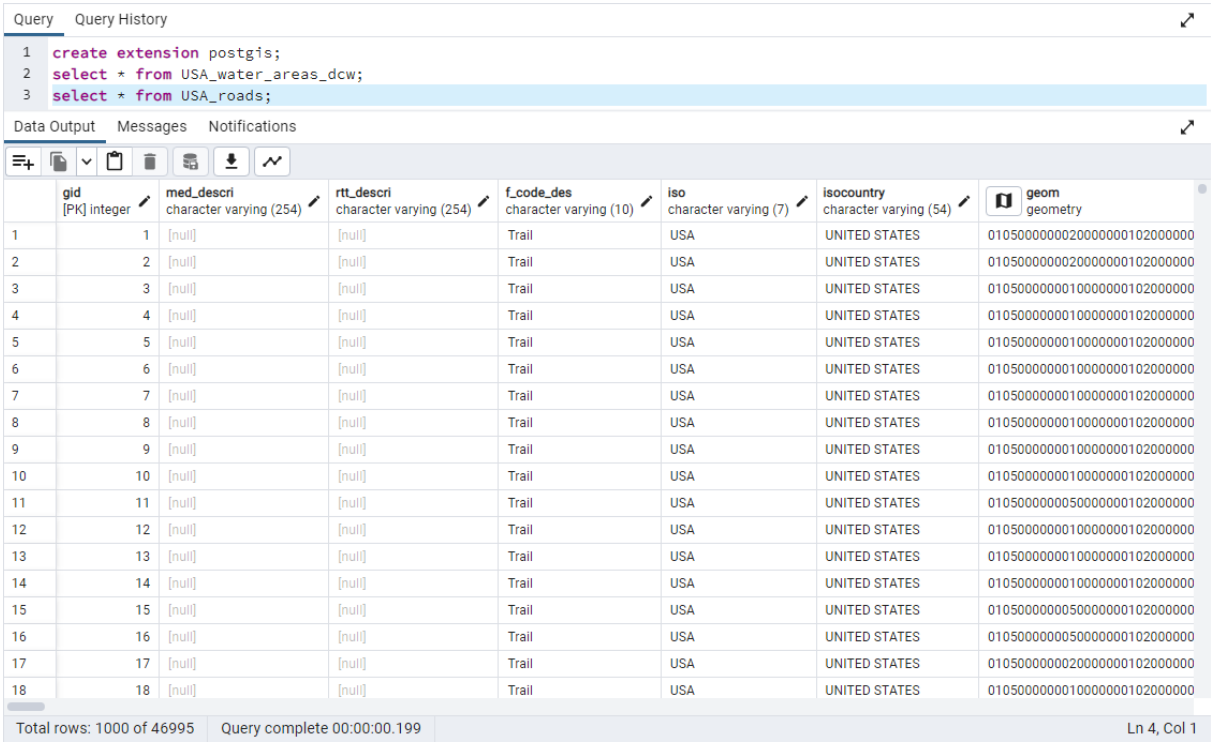
```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
```

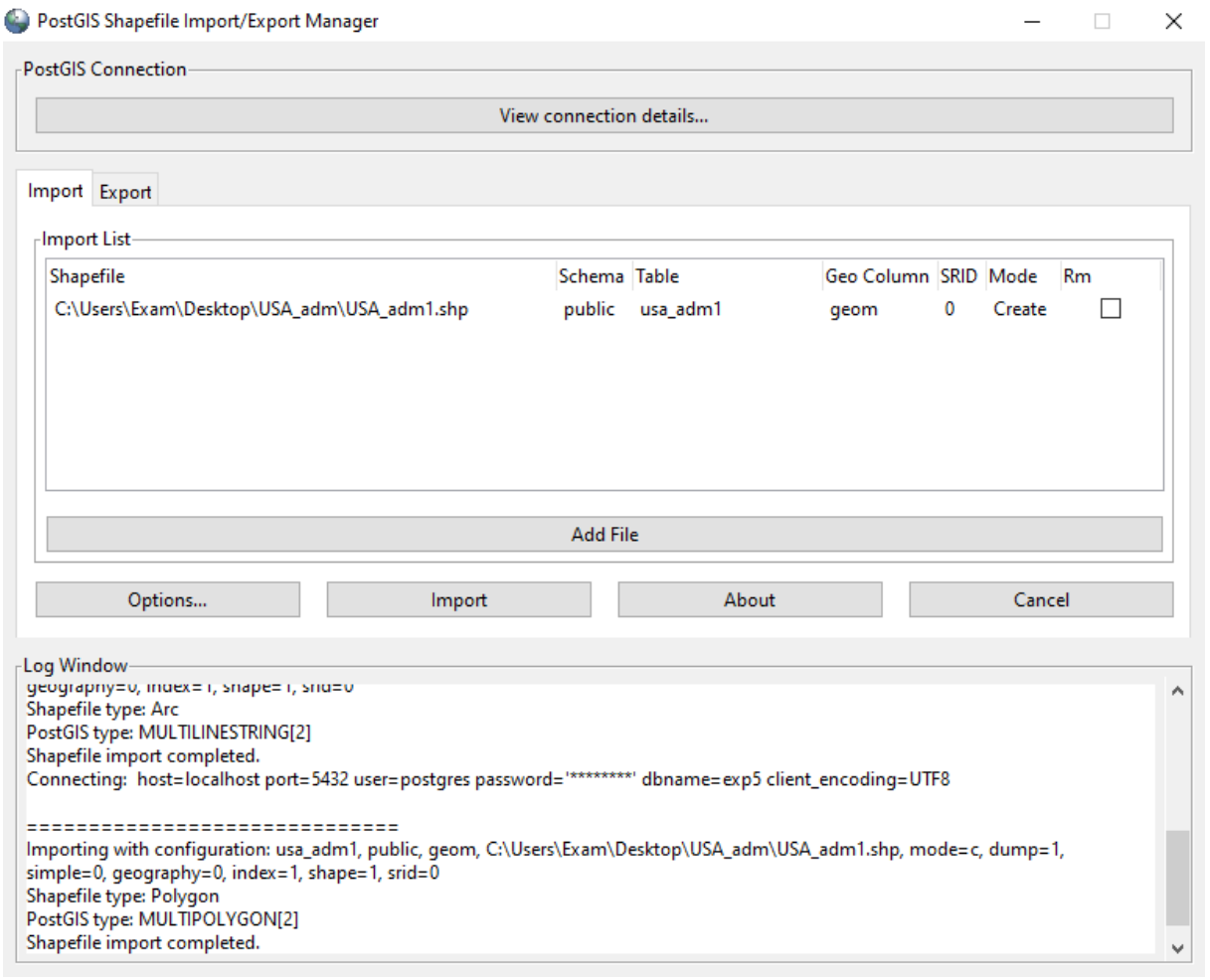
Data Output | Messages | Notifications

	gid [PK] integer	iso character varying (7)	country character varying (54)	f_code_des character varying (254)	hyc_descri character varying (254)	name character varying (254)	geom geometry
1	1	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
2	2	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
3	3	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
4	4	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
5	5	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
6	6	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
7	7	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
8	8	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
9	9	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
10	10	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
11	11	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
12	12	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
13	13	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
14	14	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
15	15	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
16	16	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
17	17	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
18	18	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000
19	19	USA	United States	Inland Water	Perennial/Permanent	UNK	0106000000010000000103000000

Total rows: 1000 of 23835 | Query complete 00:00:00.675 | Ln 2, Col 35


PostGIS Shapefile Import/Export Manager
PostGIS Connection
View connection details...
Import
Export
Import List
Shapefile
Schema
Table
Geo Column
SRID
Mode
Rm
C:\Users\Exam\Desktop\USA_rds\USA_roads.shp
public
usa_roads
geom
0
Create
Add File
Options...
Import
About
Cancel
Log Window
PostGIS type: MULTIPOLYGON[2]
Shapefile import completed.
Connecting: host=localhost port=5432 user=postgres password='********' dbname=exp5 client_encoding=UTF8
Connecting: host=localhost port=5432 user=postgres password='********' dbname=exp5 client_encoding=UTF8
==========
Importing with configuration: usa_roads, public, geom, C:\Users\Exam\Desktop\USA_rds\USA_roads.shp, mode=c, dump=1, simple=0, geography=0, index=1, shape=1, srid=0
Shapefile type: Arc
PostGIS type: MULTILINESTRING[2]
Shapefile import completed.


Query
Query History
1 create extension postgis;
2 select * from USA_water_areas_dcw;
3 select * from USA_roads;
Data Output
Messages
Notifications
gid [PK] 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)
geom geometry
1 1 [null] [null] Trail USA UNITED STATES 0105000000020000000102000000
2 2 [null] [null] Trail USA UNITED STATES 0105000000020000000102000000
3 3 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
4 4 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
5 5 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
6 6 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
7 7 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
8 8 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
9 9 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
10 10 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
11 11 [null] [null] Trail USA UNITED STATES 0105000000050000000102000000
12 12 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
13 13 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
14 14 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
15 15 [null] [null] Trail USA UNITED STATES 0105000000050000000102000000
16 16 [null] [null] Trail USA UNITED STATES 0105000000050000000102000000
17 17 [null] [null] Trail USA UNITED STATES 0105000000020000000102000000
18 18 [null] [null] Trail USA UNITED STATES 0105000000010000000102000000
Total rows: 1000 of 46995
Query complete 00:00:00.199
Ln 4, Col 1



Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
```

Data Output | Messages | Notifications

	gid [PK] integer	id_0 double precision	iso character varying (3)	name_0 character varying (75)	id_1 double precision	name_1 character varying (75)	type_1 character varying (50)	engtype_1 character varying (50)
1	1	244	USA	United States	1	Alabama	State	State
2	2	244	USA	United States	2	Alaska	State	State
3	3	244	USA	United States	2	Alaska	State	State
4	4	244	USA	United States	3	Arizona	State	State
5	5	244	USA	United States	4	Arkansas	State	State
6	6	244	USA	United States	5	California	State	State
7	7	244	USA	United States	6	Colorado	State	State
8	8	244	USA	United States	7	Connecticut	State	State
9	9	244	USA	United States	8	Delaware	State	State
10	10	244	USA	United States	9	District of Columbia	Federal District	Federal District
11	11	244	USA	United States	10	Florida	State	State
12	12	244	USA	United States	11	Georgia	State	State
13	13	244	USA	United States	12	Hawaii	State	State
14	14	244	USA	United States	13	Idaho	State	State
15	15	244	USA	United States	14	Illinois	State	State
16	16	244	USA	United States	15	Indiana	State	State
17	17	244	USA	United States	16	Iowa	State	State

Total rows: 52 of 52 | Query complete 00:00:01.338 | Ln 4, Col 24

Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
```

Data Output | Messages | Notifications

	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)	geom geometry
1	2	[null]	[null]	Trail	USA	UNITED STATES	0105000000020000000102000000020
2	8	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000060
3	9	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000050
4	10	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000120
5	13	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000070
6	14	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000080
7	16	[null]	[null]	Trail	USA	UNITED STATES	0105000000050000000102000000080
8	23	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000140
9	23	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000140
10	23	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000140
11	29	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000080
12	32	[null]	[null]	Trail	USA	UNITED STATES	010500000001000000010200000004C
13	32	[null]	[null]	Trail	USA	UNITED STATES	010500000001000000010200000004C
14	35	[null]	[null]	Trail	USA	UNITED STATES	0105000000010000000102000000040
15	38	[null]	[null]	Trail	USA	UNITED STATES	010500000001000000010200000001CC
16	38	[null]	[null]	Trail	USA	UNITED STATES	010500000001000000010200000001CC

Total rows: 1000 of 3382 | Query complete 00:00:06.017 | Ln 5, Col 105

Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
6  select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
```

Data Output | Messages | Notifications

	gid integer	id_0 double precision	iso character varying (3)	name_0 character varying (75)	id_1 double precision	name_1 character varying (75)	type_1 character varying (50)	engtype_1 character varying (50)

Total rows: 0 of 0 | Query complete 00:00:00.314 | Ln 6, Col 82

Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
6  select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
7  select * from USA_adm1, USA_water_areas_dcw where st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and USA_water_areas_dcw.gid<10;
```

Data Output | Messages | Notifications

	gid integer	id_0 double precision	iso character varying (3)	name_0 character varying (75)	id_1 double precision	name_1 character varying (75)	type_1 character varying (50)	engtype_1 character varying (50)	nl_name_1 character varying (50)
1	3	244	USA	United States	2	Alaska	State	State	[null]
2	3	244	USA	United States	2	Alaska	State	State	[null]
3	3	244	USA	United States	2	Alaska	State	State	[null]
4	3	244	USA	United States	2	Alaska	State	State	[null]
5	3	244	USA	United States	2	Alaska	State	State	[null]
6	3	244	USA	United States	2	Alaska	State	State	[null]
7	3	244	USA	United States	2	Alaska	State	State	[null]
8	3	244	USA	United States	2	Alaska	State	State	[null]
9	3	244	USA	United States	2	Alaska	State	State	[null]

Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
6  select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
7  select * from USA_adm1, USA_water_areas_dcw where st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and USA_water_areas_dcw.gid<
8  select st_length(geom) from USA_roads;
```

Data Output | Messages | Notifications

	st_length double precision
1	0.05649860967592968
2	0.13401387565234193
3	0.012420308331309185
4	0.09407750084658022
5	0.07687244685166987
6	0.01083482304992872
7	0.3484743051259539
8	0.1081814747383678
9	0.09213425486326829
10	0.4188107556979498
11	0.1178686133561139
12	0.11364582625508828
13	0.12606482349852288
14	0.23042671761619862
15	0.127327749635544

Total rows: 1000 of 46995 | Query complete 00:00:00.147 | Ln 8, Col 39

Query | Query History

```
1  create extension postgis;
2  select * from USA_water_areas_dcw;
3  select * from USA_roads;
4  select * from USA_adm1;
5  select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
6  select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
7  select * from USA_adm1, USA_water_areas_dcw where st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and USA_water_areas_dcw.gid<
8  select st_length(geom) from USA_roads;
9  select st_area(geom) from USA_water_areas_dcw;
```

Data Output | Messages | Notifications

	st_area double precision
1	0.0008543629180921356
2	0.00020025930037554957
3	0.0003309198961184148
4	0.0002333338694344614
5	0.0009253541942257727
6	0.0006402343376923322
7	0.000251968465969159
8	0.00037607566370233806
9	0.00037514979269729514
10	0.0001498335377811393
11	0.00024048788144905746
12	0.00029827463055501734
13	0.00046809498637398895
14	0.00036058996317318474

Total rows: 1000 of 23835 | Query complete 00:00:00.171 | Ln 9, Col 47

Query | Query History

```
create extension postgis;
select * from USA_water_areas_dcw;
select * from USA_roads;
select * from USA_adm1;
select * from USA_roads, USA_water_areas_dcw where st_crosses(USA_roads.geom, USA_water_areas_dcw.geom);
select * from USA_adm1, USA_roads where st_within(USA_adm1.geom, USA_roads.geom);
select * from USA_adm1, USA_water_areas_dcw where st_intersects(USA_adm1.geom, USA_water_areas_dcw.geom) and USA_water_areas_dcw.gid<10;
select st_length(geom) from USA_roads;
select st_area(geom) from USA_water_areas_dcw;
select * from USA_adm1, USA_roads where st_Dwithin(USA_adm1.geom, USA_roads.geom, 1) and USA_roads.gid<10;
```

Data Output | Messages | Notifications

	gid integer	id_0 double precision	iso character varying (3)	name_0 character varying (75)	id_1 double precision	name_1 character varying (75)	type_1 character varying (50)	engtype_1 character varying (50)	nl_name_1 character varyir
1	2	244	USA	United States	2	Alaska	State	State	[null]
2	3	244	USA	United States	2	Alaska	State	State	[null]
3	2	244	USA	United States	2	Alaska	State	State	[null]
4	3	244	USA	United States	2	Alaska	State	State	[null]
5	2	244	USA	United States	2	Alaska	State	State	[null]
6	3	244	USA	United States	2	Alaska	State	State	[null]
7	2	244	USA	United States	2	Alaska	State	State	[null]
8	3	244	USA	United States	2	Alaska	State	State	[null]
9	2	244	USA	United States	2	Alaska	State	State	[null]
10	3	244	USA	United States	2	Alaska	State	State	[null]
11	2	244	USA	United States	2	Alaska	State	State	[null]
12	3	244	USA	United States	2	Alaska	State	State	[null]

Total rows: 18 of 18 | Query complete 00:00:05.772 | Ln 10, Col 107

Google Earth Engine

Search places and datasets...

ee-chandanag

Scripts | Docs | Assets

NEW | ADD A PROJECT

CLOUD ASSETS
- ee-chandanag
  - usa_water_areas

LEGACY ASSETS
- users/chandanag

usa_waterbodies * | Get Link | Save | Run | Reset | Apps

```
Imports (1 entry)
  var shapefileUrl: Table projects/ee-chandanag/assets/usa_water_areas
// Load the shapefile as a FeatureCollection.
var shapefile = shapefileUrl;

// Display the shapefile on the map.
Map.centerObject(shapefile, 10); // Center the map on the shapefile
Map.addLayer(shapefile, {}, 'Shapefile');

// Print the FeatureCollection to the console to inspect its properties.
print(shapefile);

// Convert the shapefile to a CSV table.
var csvTable = shapefile;
// Print the resulting CSV table.
print(csvTable);

// Export the CSV table to Google Drive.
Export.table.toDrive({
  collection: csvTable,
  description: 'usa_waterbodies_shapefilecsv',
  fileFormat: 'CSV'
});
```

Inspector | Console | Tasks

Search or cancel multiple tasks in the Task Manager

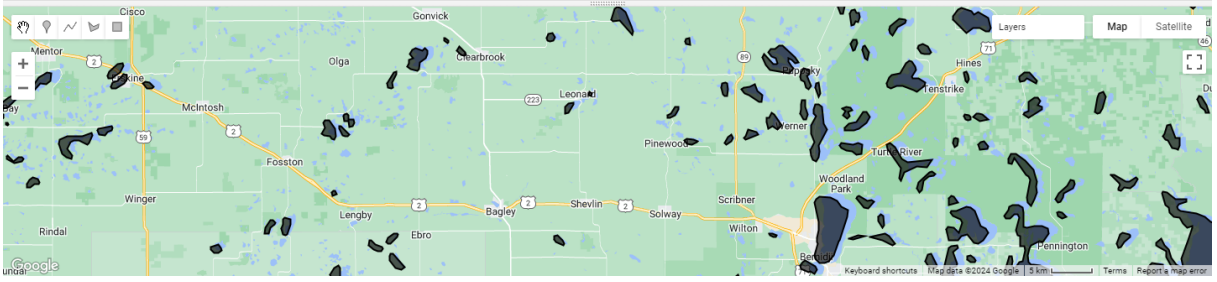
SUBMITTED TASKS

usa_waterbodies_shapefilecsv ✓ <1m

Ingest table: "projects/ee-chandanag/... ✓ 3m

Asset name: **projects/ee-chandanag/assets/usa_water_areas**
ID: **CU2YJJL4KBWMMWL6YXYETSZC**
Phase: **Completed**
Runtime: **3m** (started 2024-02-06 12:15:19 +0530)
Attempted **1** time

View asset



usa_waterbodies_shapefilecsv

File | Edit | View | Insert | Format | Data | Tools | Extensions | Help

Share

A1 | system:index

	A system:index	B COUNTRY	C F_CODE_DES	D HYC_DESCRI	E ISO	F NAME	G .geo	H	I
1	system:index	COUNTRY	F_CODE_DES	HYC_DESCRI	ISO	NAME	.geo		
2	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-100.65160486891571,46.681941012729		
3	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-100.53869599919334,46.545046575861		
4	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-100.74365407506458,46.655538629820		
5	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-100.55353588012593,46.515594205291		
6	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-118.09803836230506,44.687723960827		
7	1524	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-118.18936977038712,44.733416399647		
8	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-106.2742681669643,35.60370574715875		
9	1657	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.82238009131108,35.108143553501		
10	1658	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.80070444922322,35.096438434143		
11	1667	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-89.44823382970753,34.535724305729		
12	1670	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-89.90773974962		
13	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.36347177141725,32.511301296133		
14	1691	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.8012751593		
15	1694	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.77170687027395,35.087604921763		
16	1696	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.81170056706364,35.083970785171		
17	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.31316413463345,34.938898375466		
18	1.60E+05	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.30035316752577,34.943179127600		
19	1.60E+07	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.40980622039203,34.920379744397		
20	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.50622543583751,34.901125326922		
21	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-93.4299926326301,34.924473192547		
22	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-90.13346838151362,34.791703466283		
23	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-90.0981210717166,34.824517928094		
24	00000000000000	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-90.08732563075296,34.813017900437		
25	1731	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-90.05400717051957,34.810342461960		
26	1732	United States	Land Subject to	Non-Perennial/Ir	USA		{"type":"Polygon","coordinates":[[[-90.11119068382573,34.801812238206		

usa_waterbodies_shapefilecsv

Questions:

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

Ans: Spatial databases handle data that has a spatial component, such as geographic information system (GIS) data, maps, or any information related to locations on Earth's surface. Spatial functions are used to perform operations on spatial data. Here are some common spatial functions:

Distance Function:
Example: ST_Distance(geometry1, geometry2)
Explanation: Calculates the distance between two geometries, which could be points, lines, or polygons. It's often used to find the nearest neighbors or to analyze proximity.

Intersection Function:
Example: ST_Intersects(geometry1, geometry2)
Explanation: Checks whether two geometries intersect with each other. It's frequently used to find overlapping or intersecting spatial features.

Buffer Function:
Example: ST_Buffer(geometry, distance)
Explanation: Creates a buffer zone around a geometry. This is useful for proximity analysis and to study the influence of one spatial feature on another within a certain distance.

Area Function:
Example: ST_Area(polygon)
Explanation: Calculates the area of a polygon. This is useful for tasks such as land-use planning or environmental analysis.

Within Function:
Example: ST_Within(geometry1, geometry2)
Explanation: Checks whether one geometry is completely within another. This can be used to identify containment relationships between spatial features.

Centroid Function:
Example: ST_Centroid(geometry)
Explanation: Computes the center point of a geometry. It's often used in mapping applications to label or symbolize features.

2. Explain any two applications of spatial databases.

Ans: GIS (Geographic Information Systems):
Spatial databases are extensively used in GIS for mapping and analyzing spatial data. They store and manage geographic information such as maps, satellite imagery, and terrain data. GIS applications are widely employed in urban planning, environmental monitoring, natural resource management, and disaster response. For instance, a city planner might use a spatial database to analyze the impact of a new development on the existing infrastructure.

Location-Based Services (LBS):
Spatial databases play a crucial role in Location-Based Services, where the user's location is used to provide relevant information or services. Examples include mapping applications (like Google Maps), location-based advertising, and location-based social networking. Spatial databases help in efficiently storing and querying large datasets of geographical points of interest and providing real-time information based on the user's location.

Emergency Response and Public Safety:

Spatial databases are vital in emergency response systems. They can store information about the locations of hospitals, emergency shelters, fire stations, and other critical infrastructure. During a disaster, such as a natural calamity or a pandemic, spatial databases can be used to quickly analyze affected areas, plan evacuation routes, and allocate resources effectively.

Environmental Monitoring:
Spatial databases are used to store and manage environmental data, including information about ecosystems, habitats, and climate. Scientists and researchers use spatial databases to analyze spatial patterns, track changes over time, and make informed decisions about conservation and environmental management.

Transportation and Logistics:
In transportation and logistics, spatial databases are employed to manage and optimize the movement of goods and people. They are used to store information about road networks, traffic patterns, and distribution centers. Route planning, fleet management, and supply chain optimization are examples of applications where spatial databases play a crucial role.

These applications highlight the versatility and importance of spatial databases in various domains where location-based information is critical for decision-making and analysis.

Outcomes: Design advanced database systems using Object relational, Spatial and NOSQL databases and its implementation.

Conclusion: (Conclusion to be based on outcomes achieved)
The outcomes of the experiment underscore PostGIS as a powerful tool for executing spatial queries, making it a valuable asset in spatial data management and analysis. The successful execution of diverse spatial functions demonstrates its applicability across different domains, positioning PostGIS as a reliable solution for handling and extracting insights from spatial datasets.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

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