Lab 7: SQL and PostGIS

In this lab we explored SQL and PostGIS on the ROGER super computer. We learned how to access a SQL database on ROGER. Many SQL operations were demonstrated including, creating and querying tables; adding and deleting entries; and finding statistical values such as ‘max’, ‘min’, ‘mean’ and ‘count’. Then we accessed PostGIS, created tables and performed queries and manipulations on the data set.

SQL:

```
g479=> select * from karoth4;
 key | attr  | value
-----+-------+-------
   1 | attr0 |   100
   2 | attr1 |   101
   3 | attr1 |   102
   4 | attr1 |   103
(4 rows)
```

Fig 1: Query of all entries in the karoth4 table

```
g479=> select * from karoth4 where attr='attr1';
 key | attr  | value
-----+-------+-------
   2 | attr1 |   101
   3 | attr1 |   102
   4 | attr1 |   103
(3 rows)
```

Fig 2: Conditional query of all entries with attr =’attr1’

```
g479=> select * from karoth4 where key=2;
 key | attr  | value
-----+-------+-------
   2 | attr1 |   101
(1 row)
```

Fig 3: Conditional query of all entries with key=2

```
g479=> select key, value from karoth4 limit 1;
 key | value
-----+-------
   1 |   100
(1 row)
```

Fig 4: Select the first key value pair

```
g479=> select * from karoth4 Order by key asc;
 key | attr  | value
-----+-------+-------
   1 | attr0 |     1
   2 | attr1 |   101
   3 | attr1 |   105
   4 | attr1 |   103
(4 rows)

g479=> select * from karoth4 Order by key desc;
 key | attr  | value
-----+-------+-------
   4 | attr1 |   103
   3 | attr1 |   105
   2 | attr1 |   101
   1 | attr0 |     1
(4 rows)
```

Fig 5: Display the table in ascending and descending order respectively

```
g479=> select count(*) from karoth4 where attr like '%1';
 count
-------
     3
(1 row)

g479=> select count(*) from karoth4
;
 count
-------
     4
(1 row)
```

Fig 6: Count how many entries are in the table given certain conditions.

```
g479=> select max(value) from karoth4; select avg(value) from karoth4
where attr like '%1%';
 max
-----
 105
(1 row)

 avg
-----
 103
(1 row)
```

Fig 7: Find the maximum and average of “value”

```
g479=> select karoth4.*, karoth4_NEW.attr1, karoth4_NEW.value from kar
oth4 inner join karoth4_NEW on karoth4.key=karoth4_NEW.key;
 key | attr  | value |  attr1  | value
-----+-------+-------+---------+-------
   2 | attr1 |   101 |  attr20 |   202
   1 | attr0 |     1 |  attr10 |   101
(2 rows)

g479=> select karoth4.*, karoth4_NEW.attr1, karoth4_NEW.value from kar
oth4 left join karoth4_NEW on karoth4.key=karoth4_NEW.key;
 key | attr  | value |  attr1  | value
-----+-------+-------+---------+-------
   2 | attr1 |   101 |  attr20 |   202
   1 | attr0 |     1 |  attr10 |   101
   3 | attr1 |   105 |         |
(3 rows)

g479=> select karoth4.*, karoth4_NEW.attr1, karoth4_NEW.value from kar
oth4 right join karoth4_NEW on karoth4.key=karoth4_NEW.key;
 key | attr  | value |  attr1   | value
-----+-------+-------+----------+-------
   1 | attr0 |     1 |  attr10  |   101
   2 | attr1 |   101 |  attr20  |   202
     |       |       |  attr50  |   505
     |       |       |  attr100 |  1010
(4 rows)

g479=> select karoth4.*, karoth4_NEW.attr1, karoth4_NEW.value from kar
oth4 full join karoth4_NEW on karoth4.key=karoth4_NEW.key;
 key | attr  | value |  attr1   | value
-----+-------+-------+----------+-------
   2 | attr1 |   101 |  attr20  |   202
   1 | attr0 |     1 |  attr10  |   101
   3 | attr1 |   105 |          |
     |       |       |  attr100 |  1010
     |       |       |  attr50  |   505
(5 rows)
```

Fig 8: Examples of 4 different types of joins. The inner join only includes entries where all values are present. A left join includes entries where all of the first database values are present. Similarly, a right join includes all of the second tables values are present. And a full join joins all entries regardless of whether all the data values are present.

```
g479=> select * from karoth4 natural join karoth4_NEW;
 key | value | attr | attr1
-----+-------+------+-------
(0 rows)

g479=> select * from karoth4_NEW
g479-> ;
 key |  attr1   | value
-----+----------+-------
   1 |  attr10  |   101
   2 |  attr20  |   202
   5 |  attr50  |   505
  10 |  attr100 |  1010
(4 rows)

g479=> select * from karoth4;
 key | attr  | value
-----+-------+-------
   2 | attr1 |   101
   1 | attr0 |     1
   3 | attr1 |   105
(3 rows)
```

Fig 9: Demonstrates a natural join and the contents of the two tables that were joined in Fig 8 above.

PostGIS:

```
g479=> CREATE TABLE postgis_karoth4 (ID int4);
CREATE TABLE
g479=> SELECT AddGeometryColumn('', 'postgis_karoth4', 'geometry', 4326, 'POLYGON', 2 );
                        addgeometrycolumn
-----------------------------------------------------------------
 public.postgis_karoth4.geometry SRID:4326 TYPE:POLYGON DIMS:2
(1 row)

g479=> SELECT AddGeometryColumn('postgis_karoth4','centroid',4326,'POINT',2);
                       addgeometrycolumn
-----------------------------------------------------------------
 public.postgis_karoth4.centroid SRID:4326 TYPE:POINT DIMS:2
(1 row)

g479=> select column_name, data_type from INFORMATION_SCHEMA.COLUMNS where table_name = 'postgis_karoth4';
 column_name |  data_type
-------------+-------------
 id          | integer
 geometry    | USER-DEFINED
 centroid    | USER-DEFINED
(3 rows)
```

Fig 10: Creation of the postGIS table, and a query of its contents

```
 f_table_catalog | f_table_schema |  f_table_name   | f_geometry_column | coord_dimension | srid |  type
-----------------+----------------+-----------------+-------------------+-----------------+------+---------
 g479            | public         | postgis_karoth4 | geometry          |               2 | 4326 | POLYGON
 g479            | public         | postgis_karoth4 | centroid          |               2 | 4326 | POINT
(2 rows)
```

```
 id |                              st_astext                                |       st_astext
----+-----------------------------------------------------------------------+-----------------------
  1 | POLYGON((-128 50,-128 49.948,-127.874 49.948,-127.874 50,-128 50)) | POINT(-127.37 49.974)
  2 | POLYGON((-130 50,-130 20,-70 20,-70 50,-130 50))                   | POINT(-100 25)
(2 rows)
```

Fig 11: Different queries of the postgis_karoth4 tabl

```
        town
--------------------
 ABINGTON
 ACTON
 ACUSHNET
 ADAMS
 AGAWAM
 ALFORD
 AMESBURY
 AMHERST
 ANDOVER
 AQUINNAH
 ARLINGTON
 ASHBURNHAM
```

Fig 12: A list of all the towns in the towns table (concatenated in this figure)

The remainder of the functions were not able to be performed because when the query “town = ‘BOSTON’” was performed, there were zero entries. This made it impossible to do any other data manipulations. Various queries were attempted for other town names with the same results.