**CS 623**

**PROJECT**

**Guidelines**

* This is a group project that you will have to do in a group of 3 students (maximum).
* Post your team group as well as the data source for your group’s data set in the spreadsheet.
* You will use PostgreSQL (rather than MySQL).
* Your code should also be on your individual GitHub. This is where I will check it. The code is developed as a team but available on the GitHub of participating students.
* You have two parts, the Practical and the Theory part. There is an extra 1 mark available for attempting the project.

**Deliverables**

* Code in GitHub(individually) and link to the github. I will check the code there.
* Submit a Video of < 3 minutes to show and explain your work • Screenshots of the code plus output.
* PDF/Word doc of solutions to theory questions

**Description**

Involves working with spatial data and utilizing the access methods and query executions and optimizations we would discuss in class. The project would involve writing SQL queries to retrieve information such as the locations of specific features, distances between points, and areas of interest. Using indexing, aggregate and join executors, sort+ limit executors, sorting, and top-N optimization.

**Practical Part (75%)**

**Goal**

Creating a Geographic Information System (GIS) Analysis: A project that involves analyzing geographic data such as maps and spatial data. You will need a database that supports spatial data types, like PostgreSQL (PostGIS).

1. **Retrieve Locations of specific features (10 marks)**

**Query:** SELECT

feature\_name,

ST\_AsText(feature\_location) AS location

FROM features

WHERE feature\_name IN ('Pubs', 'Hospital', 'School',’Hotels’,’Colleges’,’Stadium’);

A screenshot of a computer

Description automatically generated

1. **Calculate Distance between points (10 marks)**

**Query:** SELECT

f1.name AS place1,

f2.name AS place2,

ST\_Distance(

f1.location,

f2.location

) AS distance

FROM places f1, places f2

WHERE f1.name='School'

AND f2.name='Hospital';

A screenshot of a computer

Description automatically generated

1. **Calculate Areas of Interest (specific to each group) (10 marks)**

**Query:** WITH region AS (

SELECT ST\_ConvexHull(ST\_Collect(location)) AS geom

FROM places

WHERE name IN ('Hospital', 'School')

)

SELECT ST\_Area(geom) AS area\_of\_interest FROM region;

A screenshot of a computer

Description automatically generated

1. **Analyze the queries (10 marks)**

Ans) a. The firsts query enables one to retrieve features location according to the specific requirement

b. The second query enable the user to obtain the distance between the specified points

c. The third query calculates the area of a specified area in the query

1. **Sorting and Limit Executions (10 marks)**

**Query1**: SELECT

f1.name,

f2.name,

ST\_Distance(f1.location, f2.location) AS distance

FROM places f1 JOIN places f2

ON f1.name='School'

AND f2.name='Hospital'

ORDER BY distance

LIMIT 1;

A screenshot of a computer

Description automatically generated

**Query 2:** SELECT \*

FROM places

WHERE ST\_Within(location,

ST\_MakeEnvelope(-71.05, -71.0, 48.43, 48.45, 4326))

LIMIT 10;

A screenshot of a computer

Description automatically generated

1. **Optimize the queries to speed up execution time (10 marks)**

**Query**: SELECT f1.name, f2.name, ST\_Distance(f1.geom, f2.geom) AS distance

FROM (

SELECT \*

FROM places

WHERE geom && ST\_MakeEnvelope(-71.05, -71.0, 48.43, 48.45)

LIMIT 100

) f1

JOIN (

SELECT \*

FROM places

WHERE geom && ST\_MakeEnvelope(-71.05, -71.0, 48.43, 48.45)

LIMIT 100

) f2 ON f1.id != f2.id

WHERE f1.name IN ('School', 'Hospital')

AND f2.name IN ('School', 'Hospital')

ORDER BY distance

LIMIT 10;

A screenshot of a computer

Description automatically generated

1. **N-Optimization of queries (5 marks)**

**Query**: SELECT places.\*

FROM places

JOIN geo\_names\_table

ON ST\_DWithin(places.geom, geo\_names\_table.geom, 50000);

A screenshot of a computer

Description automatically generated