NAME: Vishnu Vardhan Reddy Bijjam

1. Retrieve Locations of specific features.

1.1 --retrieving types of buildings and their count in New York.

```
SELECT type AS Building_type_in_NY, COUNT(*) AS count
FROM gis_osm_buildings_a_free_1
GROUP BY type;
```

1.2 --retrieve percentage of land usage in New York

```
SELECT fclass AS type_of_land,
   (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM gis_osm_landuse_a_free_1)) AS
usage_percentage
FROM gis_osm_landuse_a_free_1
GROUP BY fclass;
```

1.3 --retrieving types of roads and their speed limit in ny;

```
SELECT fclass AS Road_types_in_NY, name, maxspeed
FROM gis_osm_roads_free_1
Where name IS NOT NULL;
```

2. Calculate Distance between points

2.1 -- distance between two locations

--calculated the distance between Burger King and New City Library.

```
SELECT
   'Burger King' AS loc1,
   'New City Library' AS loc2,
   ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('0101000000848DA1525B7F52C00498E8E225
924440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('010100000011CE5D013B7F52C08B47F31142
944440', 'hex')), 4326)
   ) / 1609.34 AS distance_in_miles;
```

2.2 --distance between two points

```
SELECT
   'South Ferry' as loc1,
   'Centre Avenue' as loc2,
   ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('0101000000C6BBBF304F1452C03843BB568
B524440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('01010000007959130B7C6A52C07C6A620EF7
524440', 'hex')), 4326)
   ) / 1609.34 AS distance_in_miles;
```

2.3 --distance between two points

```
SELECT
          'Millbrook'as loc1,
          'New York' as loc2,
   ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('010100000079CE71C94C6C52C001971128AC
E44440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('0101000000D9D1938D628052C05938A4AC3
A5B4440', 'hex')), 4326)
   ) / 1609.34 AS distance_in_miles;
```

3. Calculate Areas of Interest

3.1 --Calculating landtype areas and its geometry type

```
SELECT
   fclass AS land_type,
   GeometryType(geom) AS geometry_type,
   SUM(ST_Area(geom::geography)) AS total_area
FROM
   gis_osm_landuse_a_free_1
WHERE
   name IS NOT NULL
GROUP BY
   fclass, GeometryType(geom);
```

3.2--Calculating Building areas and its geometry type

```
SELECT
   name AS building_name,
   GeometryType(geom) AS geometry_type,
   ST_Area(geom::geography) AS areas
FROM
   gis_osm_buildings_a_free_1
WHERE
   name IS NOT NULL;
```

4. Analyze the queries

4.1 --analyze the query 3.2

```
EXPLAIN ANALYZE
SELECT
   name AS building_name,
   GeometryType(geom) AS geometry_type,
   ST_Area(geom::geography) AS areas
FROM
   gis_osm_buildings_a_free_1
WHERE
   name IS NOT NULL;
```

4.2 – analyze the query 3.1

```
 EXPLAIN ANALYZE
SELECT
   fclass AS land_type,
   GeometryType(geom) AS geometry_type,
   SUM(ST_Area(geom::geography)) AS total_area
FROM
   gis_osm_landuse_a_free_1
WHERE
   name IS NOT NULL
GROUP BY
   fclass, GeometryType(geom);
```

4.3—analyze the query 1.2

```
EXPLAIN ANALYZE
SELECT fclass AS type_of_land,
   (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM gis_osm_landuse_a_free_1)) AS
usage_percentage
FROM gis_osm_landuse_a_free_1
GROUP BY fclass;
```

5. Sorting and Limit Executions

5.1 – sorting and limit executions on query 4.1

```
SELECT
   name AS building_name,
   GeometryType(geom) AS geometry_type,
   ST_Area(geom::geography) AS areas
FROM
   gis_osm_buildings_a_free_1
WHERE
   name IS NOT NULL
ORDER BY
   areas DESC
LIMIT 5;
```

5.2– sorting and limit executions on query 4.2

```
SELECT
   fclass AS land_type,
   GeometryType(geom) AS geometry_type,
   SUM(ST_Area(geom::geography)) AS total_area
FROM
   gis_osm_landuse_a_free_1
WHERE
   name IS NOT NULL
GROUP BY
   fclass, GeometryType(geom)
ORDER BY
   total_area DESC;
```

5.3– sorting and limit executions on query 4.3

```
SELECT fclass AS type_of_land,
   (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM gis_osm_landuse_a_free_1)) AS
usage_percentage
FROM gis_osm_landuse_a_free_1
GROUP BY fclass
ORDER BY usage_percentage DESC
LIMIT 10;
```

6. Optimize the queries to speed up execution time

6.1—optimize the query 5.1

```
CREATE INDEX idx_name ON gis_osm_buildings_a_free_1 (name);
EXPLAIN ANALYZE
SELECT
```

```
   name AS building_name,
   GeometryType(geom) AS geometry_type,
   ST_Area(geom::geography) AS areas
FROM
   gis_osm_buildings_a_free_1
WHERE
   name IS NOT NULL
ORDER BY
   areas DESC
LIMIT 5;
```

6.2—optimize the query 5.2

```
CREATE INDEX IF NOT EXISTS idx_name3 ON gis_osm_landuse_a_free_1 (name);
CREATE INDEX IF NOT EXISTS idx_geom2 ON gis_osm_landuse_a_free_1 USING GIST
(geom);

EXPLAIN ANALYZE
SELECT
   fclass AS land_type,
   GeometryType(geom) AS geometry_type,
   SUM(ST_Area(geom::geography)) AS total_area
FROM
   gis_osm_landuse_a_free_1
WHERE
   name IS NOT NULL
GROUP BY
   fclass, GeometryType(geom)
ORDER BY
   total_area DESC
LIMIT
   10;
```

6.3—optimize the query 5.3

```
CREATE INDEX IF NOT EXISTS idx_fclass ON gis_osm_landuse_a_free_1 (fclass);

EXPLAIN ANALYZE
SELECT fclass AS type_of_land,
   (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM gis_osm_landuse_a_free_1)) AS
usage_percentage
FROM gis_osm_landuse_a_free_1
GROUP BY fclass
ORDER BY usage_percentage DESC
LIMIT 10;
```

7. N-Optimization of queries

7.1

```
-- Create an index on the fclass column
CREATE INDEX idx_fclass2 ON gis_osm_places_free_1 (fclass);

-- Create an index on the population column
CREATE INDEX idx_population ON gis_osm_places_free_1 (population);

SELECT
    fclass AS city_type,
    SUM(population) AS total_population
FROM
    gis_osm_places_free_1
WHERE
    fclass IS NOT NULL AND population IS NOT NULL
GROUP BY
    city_type
ORDER BY
    total_population DESC;
```

7.2

```
CREATE INDEX idx_geo_point ON gis_osm_transport_free_1 USING GIST (geom);
SELECT
    'Centre Avenue' AS loc1,
    'Woodside' AS loc2,
    'Long Island City' AS loc3,
    ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('01010000007959130B7C6A52C07C26A02EF7
524440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('0101000000489D256DCF7952C0FFC8192F82
5F4440', 'hex')), 4326)
    ) / 1609.34 AS distance_A_to_B_in_miles,
    ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('0101000000489D256DCF7952C0FFC8192F82
5F4440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('010100000086657D143F7D52C02F1686C8E9
5E4440', 'hex')), 4326)
    ) / 1609.34 AS distance_B_to_C_in_miles,
    ST_DistanceSphere(

ST_SetSRID(ST_GeomFromEWKB(decode('010100000086657D143F7D52C02F1686C8E9
5E4440', 'hex')), 4326),

ST_SetSRID(ST_GeomFromEWKB(decode('01010000007959130B7C6A52C07C26A02EF7
524440', 'hex')), 4326)
    ) / 1609.34 AS distance_C_to_A_in_miles;
```