Bases de données avancées

TP: GeoSpatial

1. Create a table `monuments` and a table `shops` containing shops near monuments with the following attributes: `name`, `street_address`, `city`, `postal_code`, `location`.

   Tip: The location attribute should be SDO_GEOMETRY type including longitude and latitude as SDO_POINT_TYPE as shown below:

   ```
   SDO_GEOMETRY(2001, 8307,
           SDO_POINT_TYPE (<lng>, <lat>, NULL), NULL, NULL));
   ```

2. Using a web mapping service (e.g., OpenStreetMap) find and insert in the respective table:
   a. 2 monuments (e.g., *'Eiffel Tower'* and *'Arc de Triomphe'*).
   b. At least 5 nearby shops (e.g., *'Starbucks'*) to each of the monuments you added previously.

3. Add `metadata` to the spatial view USER_SDO_GEOM_METADATA for both tables as shown below (<dim> refers to each one of the two dimensions lng and lat, <lb> and <up> are the limits of each dimension, and tolerance is the precision loss (in meters) that we can tolerate – could be set to 0.5 for our case):

   ```
   INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO,
   SRID)
    VALUES (<table>, <column>,
    SDO_DIM_ARRAY(
     SDO_DIM_ELEMENT (<dim>, <lb>, <up>, <tolerance>),
     ...),
    8307);
   ```

4. Once data has been loaded into the spatial tables, a spatial index MDSYS.SPATIAL_INDEX must be created for each table on the geometry column as shown below:

   ```
   CREATE INDEX <table_idx> ON <table> (<column>)
    INDEXTYPE IS MDSYS.SPATIAL_INDEX;
   ```

5. Write the spatial SQL queries (see class slides) to find:
   a. The X closest shops to the first monument with name, e.g., *'Eiffel Tower'*.
   b. All the shops within Y km distance from the other monument with name, e.g., *'Arc de Triomphe'*.