BUT 3
Info-FI

TP - Application de base de données spatiale

K. Zeitouni

Pré-requis

Le projet est à faire en binôme et dure 2h30. Le rapport sera constitué des captures d'écran commentées. Une démo sera demandée. En cas de difficulté technique, formulez simplement les requêtes sans le test. Vous utiliserez le SGBD Oracle Spatial. Aidez-vous des tutos vus en cours.

Travail à réaliser

I) Création et requêtes d'une table jouet de villes

Soit la liste suivante des villes données par un identifiant numérique, le nom, la latitude et la longitude

(1, 'Munich', 11.5174, 48.17571)
(2, 'Vienne', 16.3688, 48.20254)
(3, 'San Francisco', -122.4194200, 37.7749300);
(4, 'Londres', -0.15307, 51.54931);

1) Créer la table correspondante.
2) On souhaite transformer les coordonnées en type géométrique. Décrivez la méthode et appliquez la à la table précédente.
3) Donnez la distance de Munich aux autres villes en Km. SDO_DISTANCE permet de spécifier l'unité ('unit=km').
4) Donnez la distance entre les villes deux à deux
5) (optionnel) donnez en SQL la matrice de distances entre ces villes

II) Cas d'étude

Avec le mouvement *Open Data*, de plus en plus d'administrations dont des villes en France ouvrent l'accès à leurs données au public. C'est le cas des données géographiques liées à la gestion du territoire. On s'intéresse ici à la ville de Paris et son site http://opendata.paris.fr

On se limite dans ce TP aux données suivantes : arrondissements, liste des collèges et les Stations VELIB.

1) Chargez les données à partir des données fournies au format shapefile. Etape la plus délicate. Suivre la documentation : https://docs.oracle.com/en/database/oracle/oracle-database/19/spatl/esri-shapefiles.html#GUID-2E1C765C-E29E-4EDC-9918-14FF4A73E049
2) Donnez le nombre de stations Velib par arrondissement.
3) Donnez pour chaque collèges le nombre de stations Velib ayant le même code postal (Indication : Utilisez LIKE)
4) Vérifiez que ce code postal associé à un collège dans la table est géométriquement correcte. *Utilisez l'intersection de géométries.*
5) Donnez pour chaque collèges la station Velib la plus proche et affichez la distance qui les sépare (à vol d'oiseau)
6) Trouvez collèges desservies les mieux desservies par des Velib (dont la distance moyenne à ses 3 plus proches stations Velib est la plus courte).
7) Fusionnez les arrondissements de Paris pour générer le contour de la ville
8) Visualisez les arrondissements ayant le plus de stations Velib.

Requeêes utiles:

Afficher les systems de reference, souvent sources d'erreur.

SELECT f_table_name, f_geometry_column, srid FROM geometry_columns;

Pour affecter les systems de reference, (sans transformer les coordonnées) :

SELECT UpdateGeometrySRID('roads','geom',4326);

Ou : ALTER TABLE roads ALTER COLUMN geom TYPE geometry(MULTILINESTRING, 4326) USING ST_SetSRID(geom,4326);

SELECT UpdateGeometrySRID('rivers','the_geom',4326);

Pour transformer d'un système à un autre :

ALTER TABLE roads

ALTER COLUMN geom TYPE geometry(MULTILINESTRING, 3857) **USING ST_Transform(ST_SetSRID(geom,4326),3857) ;**

Solutions des exercices sur les requêtes

Q3.

select buildings.id, count(*)

from buildings, personnes

where st_within(personnes.the_geom, buildings.the_geom)

group by buildings.id

Q4.

select count(*) from personnes, parcs

where st_within(personnes.the_geom,parcs.the_geom)

ou :

select count(*) from personnes, parcs

where st_contains(parcs.the_geom, personnes.the_geom)

ou encore :

select count(*) from personnes p1

where exists

(select * from parcs p2 where st_contains(p2.the_geom, p1.the_geom))

Q5.

SELECT r.*, st_distance(ri.the_geom,r.geom) from roads r, rivers ri

where st_distance(ri.the_geom,r.geom)<300000;

Q6.

Affichage du résultat de Q5 soit via l'interface pgadmin (on peut aussi créer une vue « Rivieres_proches_routes » et la visualiser) ou executer la requête dans QGIS et la rajouter comme une nouvelle couche.

Q7.

select arr_libelle, count(*) as nb_ecoles

from etablissements_scolaires

group by arr_insee, arr_libelle

order by arr_insee;

Q8.

1ère sous-requête : arrondissement des stations velib :

select s.*, c_arinsee, l_aroff

from velib_stations s, arrondissements a

where s.name like '%' || l_aroff ||'%';

Il suffit de faire la jointure avec etablissements_scolaires sur le code postal :

select s.*, c_arinsee, l_aroff

from velib_stations s, arrondissements a, etablissements_scolaires e

where s.name like '%' || l_aroff ||'%' AND a. c_arinsee = e. arr_insee;

Q9.

```
select count(e.*) as nb_affectations_incorrectes
from etablissements_scolaires e, arrondissements a
where e.arr_insee = a.c_arinsee AND ST_DISJOINT (e.geom, a.geom);
```

Q10.

```
select e.*, st_distance(e.geom, rc.geom)
from etablissements_scolaires e, "reseau-cyclable" rc
where  ST_DISTANCE (e.geom, rc.geom) <200;
```

ou bien :

```
select e.*, st_distance(e.geom, rc.geom)
from etablissements_scolaires e, "reseau-cyclable" rc
where  ST_INTERSECTS (ST_BUFFER(e.geom,200), rc.geom);
```

-- Remarque : le temps d'exécution est très long, à moins de créer des index sur les géométries. Vous pouvez interrompre par le bouton « cancel query).

Q11.

```
select t.libelle, min (d) distance_Ecole_Velib
from (Select e.*, st_distance(e.geom, s.geom) as d
        from etablissements_scolaires e, velib_stations s
                where  ST_DISTANCE (e.geom, s.geom) <200) as t
group by libelle
order by libelle;
```

-- prend environ 9 secondes.

Q12.

```
select  ST_union (a.geom) as contour_Paris
from arrondissements a ;
```

-- génère la geométrie du contour de Paris par fusion des geométries de ses arrondissements.

Q13.

Utilisation de pgadmin geometry Viewer ou QGIS si requêtes retourne une géometrie.