

Indications

La base de données à télécharger se situe sur <http://cahier-de-prepa.fr/mpsi-kju/> dans Informatique->Documents à télécharger->Pour les MP et porte le nom `dots.sql`. On peut le renommer en `dots.sqlite` pour le rendre plus facilement visible par l'extension SQLite Manager de Firefox.

Il est recommandé d'ouvrir avec SQLite Manager (plugin de Firefox) cette base de données, pour la parcourir et essayer ici ou là quelques syntaxes de requêtes SQL.

Quelques exemples de requêtes SQL à partir de la base de données suivent (ne pas perdre de temps à les essayer toutes, mais vous pouvez vous en inspirer pour les questions du TP) :

1 Requêtes sur une seule table

- La Gironde a pour identifiant 34 (et pour "code" 33..., mais la clé primaire est l'attribut "id" de la table `deps`, et celui-ci vaut 34, la faute aux deux départements corses 2A et 2B...).

Deux polygones en décrivent le contour. Les points du premier peuvent être obtenus ainsi :

```
SELECT x,y FROM pointsdep WHERE iddept=34 AND poly=0;
```

(remplacer `poly=0` par `poly=1` pour le second bien sûr, lequel correspond semble-t-il aux îles de Malprat, près du Teich)

- Ordonner les identifiants des départements par nombre décroissant de polygones :

```
SELECT iddept FROM pointsdep GROUP BY iddept ORDER BY MAX(poly);
```

(Petit rappel : `MAX(poly)` est une fonction d'agrégation qui est ici évaluée sur tous les enregistrement d'un paquet déterminé par l'attribut `iddept`. Si on voulait sélectionner des enregistrements selon une condition donnée, après `GROUP BY` on utiliserait `HAVING` plutôt que `WHERE`.)

2 Requête sur plusieurs tables

- Donner le nombre de polygones du contour du Finistère :

```
SELECT MAX(poly) FROM pointsdep, deps WHERE deps.id = iddept AND nom="FINISTERE";
```

et la même chose avec une jointure (pas naturelle, cela ne marcherait pas avec les attributs de même nom `id` des deux tables) :

```
SELECT MAX(poly) FROM pointsdep JOIN deps ON deps.id=iddept WHERE nom="FINISTERE";
```

- Donner le nom du département dont le contour comporte le plus de points :

```
SELECT deps.nom, COUNT(*) as p FROM pointsdep, deps WHERE deps.id=iddept  
GROUP BY iddept ORDER BY p DESC LIMIT 1;
```

La même chose avec une jointure plutôt qu'un produit cartésien :

```
SELECT deps.nom, COUNT(*) as p FROM pointsdep JOIN deps ON deps.id=iddept  
GROUP BY iddept ORDER BY p DESC LIMIT 1;
```