

info710 : Compléments de bases de données
TP 1 : prise en main de PostgreSQL

Pierre Hyvernât
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernât@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernât/>
wiki : <http://www.lama.univ-savoie.fr/wiki>

Remarque : la documentation de PostgreSQL se trouve en ligne sur

<http://www.postgresqlfr.org/>

section “Documentation”, version “8.3.?” . Je vous conseille de garder un navigateur web ouvert sur cette page pendant les TP...

Toutes les données utilisées pour ce TP ont été récupérées sur le site de l'INSEE :

<http://www.insee.fr/fr/methodes/nomenclatures/cog/telechargement.asp>

puis simplifiées pour les besoins du TP...

Le serveur de bases de données se trouve physiquement sur une machine dédiée. (Elle ne sert qu'à cela.) Cette machine se trouve dans le bâtiment Mont Blanc et a comme url eco.univ-savoie.fr. Vous pouvez vous connecter sur cette machine avec la commande

```
> ssh phyve@eco.univ-savoie.fr
```

en remplaçant “phyve” par votre propre identifiant et en utilisant votre mot de passe habituel (celui du portail).

Exercice 1 : premiers pas avec PostgreSQL

Question 1. Pour se connecter à la base `communes-francaises`, il faut utiliser la commande

```
> psql communes-francaises
```

Le mot de passe de votre utilisateur pour les bases de données est égal à votre login.

La première chose à faire est de modifier ce mot de passe en utilisant la commande suivante : (dans l'environnement PostgreSQL)

```
> ALTER ROLE phyve WITH PASSWORD 'mot-de-passe' ;
```

(Bien entendu, il faut remplacer “phyve” par votre propre identifiant et “'mot-de-passe'” par votre nouveau mot de passe).

Remarque : comme `psql` garde un historique des commandes, il est conseillé de quitter l'environnement `psql` et de faire un “> `rm .psql_history`” afin de supprimer le fichier où apparaît votre nouveau mot de passe.

Question 2. Reloggez-vous sur la base `communes-francaises` du serveur ; vous pouvez obtenir des informations sur les tables présentes dans cette base avec la commande “> `\dt`”

Le résultat est

```
List of relations
 Schema |      Name      | Type | Owner
-----+-----+-----+-----
 public | communes      | table | phyve
 public | departements  | table | phyve
 public | regions       | table | phyve
(3 rows)
```

ce qui indique la présence de trois tables nommées `communes`, `departements` et `regions`. Pour obtenir des informations sur une table particulière, il faut utiliser `> \d nom_de_table`.

Quelles sont les attributs (et leurs types) de chacune de ces tables ? Quelles sont les clés principales et étrangères ?

Question 3. D'autres commandes sont disponibles :

- `\l` pour obtenir la liste des bases de données sur le serveur ;
- `\du` la liste des utilisateurs sur le serveur ;
- `\?` pour obtenir la liste des commandes possibles ;
- `\h COMMANDE` pour obtenir de l'aide sur une commande SQL.

Testez certaines de ces commandes et essayer d'interpréter les résultats.

Question 4. les conventions sont d'utiliser des majuscules pour les mot clés SQL. Ceci n'est pas obligatoire car PostgreSQL ne fait pas de différence entre majuscules et minuscules. (Sauf dans les chaînes de caractères...)

Testez les commandes suivantes

- `> SELECT * FROM regions ;`
- `> select * from REGIONS ;`
- `> SeLeCt * fRoM RegIonS ;`

Essayer maintenant les commandes suivantes

- `> SELECT * FROM "regions" ;`
- `> SELECT * fRoM "RegionS" ;`
- `> SELECT * "FROM" Regions ;`

Qu'en pensez-vous ? Quelle est la différence entre `FROM` et `regions` ?

Question 5. La commande `\dS` permet d'obtenir la liste des tables "système". Par exemple, la table `pg_user` contient la liste des utilisateurs...

Regardez certaines de ces tables et essayez de deviner à quoi correspondent leurs attributs.

Exercice 2 : premières requêtes SQL

Question 1. Regardez attentivement les départements d'outremer et leurs communes. Que constatez-vous ? (Ces départements sont : Guadeloupe, Martinique, Guyane et La Réunion)

Question 2. Reprenez le TD2, et testez les requêtes "pour de vrai".

Question 3. Essayer d'écrire d'autres requêtes.

Question 4. Afficher le nombres de communes dans les régions comportant strictement plus de 3 départements.

(Apprenez à utiliser l'aide en ligne et les messages d'erreur de `psql`...)

Exercice 3 : interface graphique

Comme la machine `eco.univ-savoie.fr` est connectée au réseau, on peut utiliser d'autres outils pour accéder à la base de données. Ouvrez un navigateur internet et allez sur la page <https://eco.univ-savoie.fr/phpgadmin>

Connectez-vous en utilisant votre identifiant habituel et le mot de passe "bases de données".

Question 1. Explorez l'interface graphique phpgadmin pendant quelques minutes.

La colonne de gauche contient la liste de toutes les bases de données qui sont sur le serveur. Vous pouvez par exemple vous connecter sur la base `communes-francaises` et parcourir les tables `Departements`, `Communes` et `Regions`.

Question 2. Refaites quelques requêtes à partir de l'interface graphique.

(Pour taper une commande SQL, il faut cliquer sur le lien "SQL" dans le bandeau en haut à droite...)

Exercice 4 : insertion et suppression d'éléments

Connectez-vous sur la base de données `communes-francaises`. Pour faire des tests, une nouvelle région a été créée : Brobdingnag.

Question 1. Quels sont les départements de cette région ? Quelles en sont les villes ?

Question 2. Avec l'interface "texte", rajoutez une ville ou un département dans cette région. Ne rentrez aucune valeur pour l'attribut "id".

Vérifiez que les modifications sont visibles depuis l'interface "graphique".

Question 3. Idem, à l'envers.

Vérifiez que la liste des villes contient également celles que vos voisins ont rajoutées.

Question 4. Que constatez-vous à propos de la valeur de l'attribut "id" des départements et des communes ajoutés ?

Comment interprétez-vous la valeur par défaut de cet attribut proposé par l'interface graphique ?

Exercice 5 : les "vues"

Il est parfois pratique de stocker le résultat de certaines requêtes dans une table "fictive". Une telle table est appelé une "vue". Pour créer une vue temporaire, on utilise

```
CREATE TEMP VIEW nom_vue AS ...
```

où "..." peut en particulier être une clause SELECT.

Question 1. En utilisant les vues, refaites la dernière question du TP1.

Question 2. Si vous utilisez "Créer une vue" dans l'interface graphique, la vue n'est pas temporaire. Dans ce cas, pour éviter que vos vues n'interfèrent (tout le monde travaille sur la même base), donnez un nom unique à vos vues. (Par exemple, `vue_TP1_login...`)

Quelle est la différence entre une vue temporaire et une vue non-temporaire ?

Question 3. Créez une vue qui contienne toutes les communes de la région Brobdingnag, avec leur département.

Rajoutez une ville dans cette région, et faites une requête SELECT sur la table. Que constatez-vous ?

Essayez de rajouter une ville dans la vue. Que constatez-vous ?

Question 4. Quelques requêtes supplémentaires :

- quels sont les départements ayant une commune appelée "Saint-Sulpice" ? (Donnez plusieurs solutions...)
- quels sont les dix départements ayant le plus de communes ?
- quels sont les départements qui ont un nombre de communes supérieur à la moyenne ?