

## info710 : compléments de bases de données

### TP 2 : interrogation d'une base de données

Pierre Hyvernat  
Laboratoire de mathématiques de l'université de Savoie  
bâtiment Chablais, bureau 22  
téléphone : 04 79 75 94 22  
email : [Pierre.Hyvernat@univ-savoie.fr](mailto:Pierre.Hyvernat@univ-savoie.fr)  
www : <http://www.lama.univ-savoie.fr/~hyvernat/>

*Vous devrez m'envoyer un petit compte-rendu de ce TP par email avant le mercredi 6 décembre.  
Le sujet de l'email devra comporter "TP2-info710" et le compte-rendu consistera en un seul fichier  
"TP2-nom1-nom2.sql" dont le squelette se trouve sur*

*<http://www.lama.univ-savoie.fr/~hyvernat/>*

*(dans la section "Enseignement"...)*

*Ce fichier devra être un fichier SQL valide : toutes les explications demandées devront donc être en  
commentaire. N'oubliez pas de tester votre script avant de me l'envoyer.*

*Pour l'exercice 1, seule la question 3 est à traiter dans le compte-rendu...*

**Remarque :** la documentation de PostgreSQL se trouve en ligne sur

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

#### **Exercice 1 : interface graphique / ligne de commande**

Au TP précédent, vous avez utilisé le programme `psql` pour vous connecter sur une base de donnée. Ce programme s'exécutait sur la machine `eco.univ-savoie.fr`, qui contient également les données.

Comme cette machine est connecté au réseau (c'est grâce à ça que vous avez pu vous connecter par ssh), on peut utiliser d'autres outils pour accéder à la base de données. Pour commencer, redémarrez votre ordinateur sous Linux, et connectez-vous à `eco.univ-savoie.fr` par ssh. Ouvrez ensuite un navigateur internet (Firefox ou Mozilla) et allez sur la page

<https://eco.univ-savoie.fr/phppgadmin>

Connectez-vous en utilisant votre login habituel et le mot de passe "bases de données". (C'est votre login si vous ne l'avez pas changé au TP1.)

La colonne de droite contient la liste de toutes les bases de données qui sont sur le serveur. Vous pouvez par exemple vous connecter sur la base `tp1` et parcourir les tables `cours`, `cours_etu` et `etu`.

*Question 1 :* explorez l'interface graphique `phppgadmin` pendant quelques minutes.

*Question 2 :* vérifiez que les modifications apportées à l'aide d'une interface sont répercutées dans l'autre interface. Pour ceci, rajoutez une ligne dans une des tables (cf TP1) et vérifiez qu'elle apparaît dans l'autre.

Pour parcourir une table, vous pouvez passer par (colonne de gauche)

`nom_base_de_donnée --> public --> Tables --> nom_table`

et utiliser "parcourir" dans l'écran central. Vous pouvez également utiliser une requête "SELECT" en cliquant sur "SQL" dans le bandeau du haut.

À partir de l'interface graphique, rajouter des lignes de deux manières différentes : en utilisant SQL directement et en utilisant l'interface.

Pouvez-vous supprimer ou modifier les lignes des tables ? Testez plusieurs tables du TP1...

*Question 3 :* certaines facilités de l'interface `psql` sont moins accessible à partir de `phppgadmin`. Par exemple, pour obtenir la liste des utilisateurs, \du suffisait avec `psql`. Pour faire la même chose avec `phppgadmin`, il faudra utiliser la requête SQL suivante :

```
SELECT * FROM pg_roles ;
```

Essayez... Comment interprétez-vous la requête précédente ?

## Exercice 2 : utilisation de SELECT, fonctions d'agrégats

*Question 1* : connectez vous sur la base de données `tp2-world`. En vous aidant de l'interface graphique ou de l'interface ligne de commande, explorez les tables pour comprendre ce qu'elles contiennent.

*Question 2* : quelle requête SQL utilisez-vous pour obtenir la liste de toutes les langues parlées dans le monde ? Que constatez-vous ?

En allant lire la documentation de `SELECT`, essayez de corriger le problème.

*Question 3* : dans une clause `SELECT`, la liste des attributs peut utiliser des fonctions SQL. Par exemple, pour obtenir le nombre d'habitants, en millions, du pays qui a la population la plus élevée, on peut faire :

```
SELECT MAX(population)/1000000 FROM country ;
```

On dit que `MAX` est une fonction d'agrégat. D'autres fonctions d'agrégat sont `MIN`, `AVG`, `COUNT` qui permettent respectivement de trouver le minimum, la moyenne ou le nombre des attributs concernés.

Combien de pays sont présents dans la base de données ?

*Question 4* : combien de pays ont une espérance de vie inférieure à la moyenne ? Écrivez une requête SQL pour trouver la réponse...

*Indice* : on peut utiliser le résultat d'une requête `SELECT` à l'intérieur d'une autre clause `SELECT`.

*Question 5* : calculez de la même manière le nombre de pays dont l'espérance de vie est supérieur ou égal à la moyenne. Est-ce que l'on obtient ainsi le nombre total de pays ? Essayer de trouver une explication...

## Exercice 3 : jointures et sélections

Pour faire la jointure de plusieurs tables, on utilise un clause `SELECT ... WHERE`

*Question 1* : quelle la liste des pays qui utilisent le suédois comme langage ? (Comme la base est en anglais, il faut utiliser "Swedish" comme langue.)

*Question 2* : parmi ceux-ci, quels sont ceux pour lesquels le suédois est langue officielle ?

*Question 3* : 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`.

En utilisant les vues, donner pour chaque pays qui utilise le suédois, la liste de ses langues. 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_suedois_login...`)

Pour trier le résultat selon les langues, il faut utiliser l'option `ORDER BY`. Essayer.

*Question 4* : trouvez la liste des villes dont la population représente plus de la moitié de la population nationale.

*Question 5* : essayer de comprendre comment fonctionne les options `GROUP BY` et `HAVING` de `SELECT`. Utilisez ces options pour produire la liste des pays qui parlent plus de 10 langues. La réponse devra avoir la forme

name	nblang
-----+-----	
Pays	n
...	...

où la deuxième colonne contient le nombre de langues parlées dans le pays en question.

*Question 6* : choisissez une question du même type que les précédentes, posez-la en langue naturelle, et cherchez la réponse avec une requête SQL.

*Question 7* : recommencez le TP en partant de la question 6.