

info710 : compléments de bases de données

TP 3 : requêtes, suite et fin ; contraintes d'intégrité

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

Exercice 1 : requêtes imbriquées

Connectez-vous, par la méthode de votre choix, à la base `tp2-world`.

À la suite du mot clé `WHERE`, on peut utiliser un `IN`. Par exemple :

```
SELECT name FROM city WHERE id IN (SELECT capital FROM country) ;
```

donne la liste de toutes les capitales.

Question 1 : quelle est la liste des pays dont la capitale à plus de cinq millions d'habitants ?
(En utilisant un `IN`.)

Question 2 : répondez à la question précédente sans utiliser de `IN`.

Question 3 : il est possible de nier le sens de `IN` en rajoutant `NOT` devant. En utilisant cette possibilité, trouver la liste des villes qui ne sont pas des capitales et dont la population est supérieure à cinq millions. Est-ce que ça marche comme attendu ?

Question 4 : après un `WHERE`, il est possible d'utiliser d'autres mots clés tels que `ANY` ou `ALL`. Ceci permet d'utiliser des opérateur logique en références à "au moins" ou "toutes" les lignes d'une table.

Quels sont les pays dont toutes les ville ont moins de 500000 habitants ?

Question 5 : il existe une autre instruction qui est parfois plus pratique que `ANY` : il s'agit de l'instruction `EXISTS`. Cette commande, suivie d'une requête, prend la valeur vraie si la requête renvoie une table non vide, et prend la valeur fausse sinon.

Donner la liste des pays et le nombre de leur districts, à condition qu'ils aient au moins une ville de plus de cinq millions d'habitants. Le résultat devra être trié par ordre inverse de l'ordre alphabétique.

Exercice 2 : création de table, contraintes

Commencez par vous logger sur la base de données `tp3-login`. La commande SQL pour créer une nouvelle table dans une base données est

```
CREATE TABLE nom_table (att1 typ1, ... , attn typn)
```

où `att1`, ..., `attn` sont des noms d'attributs et `typ1`, ..., `typn` sont leurs types. Les types les plus fréquents sont

- `boolean`,
- `int`, et `bigint`,
- `varchar (n)` où `n` est la longueur maximal des chaînes,
- `text`,
- `date`,
- ...

Vous allez recréer une base de données du monde simplifiée. Elle ne contiendra que deux tables `pays` et `villes`. Les attributs seront

- `nom` pour le noms du pays

- `code` pour le code du pays
- `capitale` pour le code de la capitale
- `population` pour la population totale du pays
- `id` pour le code de la ville
- `pays` pour le code du pays auquel appartient la ville
- `nom` pour le nom de la ville
- `population` pour la population de la ville

Pendant tout cet exercice, n'oubliez pas de tester vos tables en ajoutant des lignes, en en supprimant et en faisant des requêtes. Vous pouvez utiliser des villes et pays imaginaires (en suivant une certaine cohérence logique quand-même)

Question 1 : en allant lire l'aide la commande `CREATE TABLE`, créez ces deux tables, en précisant les clés principales que vous aurez choisies. N'oubliez pas d'imposer certaines restrictions quand à l'utilisation de la valeur `NULL`.

Question 2 : allez voir les contraintes utilisées pour les tables correspondantes dans la base `tp3-world` et comparez les avec vos contraintes. Essayez de comprendre ce que font les autres contraintes du `tp3-world`.

Question 3 : Rajouter quelques contraintes (taille des villes supérieure à 10, ...)

Pour rajouter une contrainte à une table existante, il faut utiliser la commande

```
ALTER TABLE nom_table [DROP,ADD,RENAME,MODIFY] CONSTRAINT ... ;
```

Question 4 : rajouter les contraintes suivantes :

- chaque pays doit avoir sa capitale dans la liste des villes,
- chaque ville doit avoir un code de pays valide.

Qu'en pensez-vous ?

Question 5 : certaines contraintes peuvent être déclarées comme "différables". Il faut rajouter dans leur définition :

```
CONSTRAINT nom ... DEFERRABLE
```

Une telle contrainte peut être différée en utilisant

```
BEGIN ;
SET CONSTRAINT nom DEFERRED ;
...
SET CONSTRAINT nom IMMEDIATE ;
COMMIT ;
```

Essayez cette technologie sur les tables des villes et des pays. Que ce passe-t'il quand vous faites le `COMMIT` ? (La contrainte peut être ou non satisfaite par les tables...)

Question 6 : on peut supprimer une colonne dans une table avec

```
ALTER TABLE nom_table DROP COLUMN nom_col ;
```

Supprimez la colonne `population` de vos tables.

Question 7 : on peut supprimer une table avec

```
DROP TABLE nom_table ;
```

Si vous avez bien sauvegardé votre session SQL, supprimez les deux tables `ville` et `pays`.