Bases de Données (BDD1)

octobre 2021

TDM n°4

Ces exercices utilisent la même base que celle de la fiche précédente

Exercice 1 :
On attribue à chaque commune une catégorie selon sa superficie, par tranche de $5km^2$. Une catégorie est désignée par un nombre **entier** : 1 pour les communes de strictement moins de $5km^2$, 2 pour celles de 5 à moins de $10km^2$, etc...
Q 1 . Écrivez la requête SQL permettant d'obtenir toutes les informations de la relation `communes` plus un attribut nommé `cat_sup` (la catégorie)
Q 2 . Écrivez la requête SQL permettant d'obtenir le nombre de communes de chaque catégorie, et leur superficie moyenne (classées par catégorie croissante)
Q 3 . On attribue de la même manière, une catégorie calculée selon les tranches de 1000 habitants (catégorie 1 pour pour strictement moins de 1000 habitants, etc) en se fondant sur la population totale lors du recensement de 2016.
Écrivez la requête SQL permettant d'obtenir une relation qui étend la relation `communes` en lui ajoutant un attribut `cat_sup` (la catégorie de superficie), un attribut `cat_pop` (catégorie de population) et un attribut pop_2016 (population totale en 2016)
Q 4 . Écrivez les requêtes permettant d'obtenir

- Le nombre de communes et leurs superficies minimale, maximale et moyenne pour chaque catégorie de population (par ordre de catégorie de population). La moyenne sera arrondie à 2 décimales.
- Faites de même mais en n'obtenant que les catégories comportant plus de 5 communes et en classant par nombres de communes décroissants.

Exercice 2 :
Dans une base de données, il est possible des mémoriser des **VUES** (`VIEW`). Un vue peut être considérée comme une requête `SELECT` pré-enregistrée sur le serveur. Une vue ne contient pas de données en elle-même : à chaque consultation du contenu d'une vue, la requête `SELECT` enregistrée est ré-exécutée et c'est son résultat que l'on obtient.
Du point de vue SQL, on consulte le contenu d'une vue comme s'il s'agissait d'une table, c'est à dire par une requête `SELECT`.
Prenons l'exemple de la requête nécessaire pour obtenir la liste des communes, leur noms et populations totales en 2012 :

```
select insee, nom, pop_totale as pop_2012
from communes
natural join population
where recensement=2012
```

Cette requête peut êttre enregistrée dans la base sous forme d'une vue en la précédant de `create view` *nomDeVue* `as` :

```
create view comm_2012 as
select insee, nom, pop_totale as pop_2012
from communes
natural join population
where recensement=2012
```

Enregistrez-là et constatez sa présence dans l'onglet "vues" de phppgadmin. Puis vous afficherez son contenu en exécutant la requête SQL :

```
select * from comm_2012
```

Notez bien qu'exécuter `select * from comm_2012` c'est exactement comme exécuter la requête `select insee,... from communes natural join ...where ....`
Pour supprimer une vue :

```
drop view comm_2012
```

(supprimer une vue ne supprime aucune donnée)
Pour modifier une vue, on doit la redéfinir

```
create or replace view comm_2012 as
... nouvelle définition ...
```

Q 1 . Enregistrez comme vue la requête de l'exercice 1 Q3, puis réécrivez les requêtes de la question 4 en utilisant cette vue.

Exercice 3 :
Concevez les requêtes nécessaires pour obtenir la liste des communes (n° INSEE, nom) avec la population totale en 2012 et en 2016, ainsi que la progression de population entre ces 2 années et le pourcentage de progression (par rapport à la valeur 2012). On souhaite les obtenir par ordre croissant du pourcentage de progression.