Activité : les Bases de données relationnelles

Présentation

Quand nous avons une **grande quantité d'informations numériques** à gérer, il est nécessaire de les **organiser** et d'utiliser un **logiciel spécialisé,** le **S**ystèmes de **G**estion de **B**ases de **D**onnées pour les utiliser.

Actuellement la grande majorité des SGBD utilise le **modèle relationnel** qui consiste :

- à mettre ensemble des **données de même nature** en les rangeant dans des **tables** de données ;
- à définir des **relations** pour relier les données **entre elles**.
- à utiliser le **langage SQL** ((Structured Query Language) pour **interroger, modifier ou supprimer** des données.

> Une petite histoire des bases de données :
>
> - https://pixees.fr/lhistoire-des-base-de-donnees-ou-presque/

Dans cette activité je vais **interroger** les données sur les villes du site http://sql.sh en utilisant le langage de requêtes SQL :

- le **contenu** du fichier villes_point_virgule.csv a été mis dans une **table** appelée **icn*ville dans une base de données MySQL : * à partir de site Web http://icn.boonum.fr, je vais pouvoir exécuter des requêtes SQL pour faire des recherches dans les données de la table icn_ville.***

> Il existe plusieurs logiciels de gestion de base de données. **MySQL** (racheté par la société Oracle), continue à être une **base de données OpenSource** sous le nom de projet **MariaDB**.

===== Les requêtes SQL ===== ==== La projection ==== La projection consiste à choisir les informations, en utilisant la clause **select** suivie les champs à afficher : * Exemple : <code sql> select *from icnville ; </code>*

> Après la clause **select**, j'indique :
>
> - les **colonnes** à afficher **séparées par des virgules** ,
> - ou bien je mets le caractère * pour visualiser le contenu de **toutes les colonnes**.
>
> Puis j'indique avec la clause **from** la table dans laquelle se trouve les données.

> **Question 1 :** Ecrire la requête SQL pour avoir le **nom**, le **code postal** et le **numéro de département** (dans cet ordre) des villes.
>
> Résultat à obtenir :
>
> | nom | cp | dep |
> |---|---|---|
> | Ozan | 1190 | 1 |
> | Cormoranche-sur-Saône | 1290 | 1 |
>
> ...

* je peux renommer une colonne en définissant un alias avec le mot clé **as** et aussi faire des calculs ou utiliser des fonctions : <code sql> select concat(nom, ' ', cp) as adresse from icnville ; </code> **Résultat :** ^adresse^ |nom cp| |Ozan 1190| |Cormoranche-sur-Saône 1290|

> **Question 2 :** Ecrire la requête SQL qui indique pour chaque ville **l'augmentation** de la population entre 2010 et 2012. **Résultat à obtenir :**
>
> | ville | augmentation |
> |---|---|
> | Ozan | -118 |

ville	augmentation
Cormoranche-sur-Saône	-58
Plagne	-29
Tossiat	-6
Pouillat	12

...

==== Eviter des résultats en double ==== Des requêtes peuvent renvoyer des **résultats identiques** et il est parfois utile d'évier cela avec le mot clé **distinct**. * Exemple connaître la liste des départements sans utiliser **distinct** : <code sql> select dep as Département from icn*ville ; </code>* ***Résultat :*** *^Département^ |1| |1| |1| ... soit **36 701 lignes**. * Connaître la liste des départements en utilisant **distinct** qui doit être placé une **seule fois** juste après le mot clé **select** : <code sql> select distinct dep as Département from icn*ville ; </code> **Résultat :** ^Département^ |1| |2| |3| ... soit **103 lignes**.

Question 3 : Ecrire la requête SQL qui donne la liste des codes postaux. La requête SQL doit renvoyer uniquement **6 083 lignes**.

==== Trier les résultats obtenus ==== Les requêtes SQL renvoient en général les données dans **l'ordre** où elles sont disponibles dans la base de données. Pour obtenir un ordre de **tri différent** on utilise les mots clés **order by** suivi des colonnes à trier en ascendant, par défaut (**asc**) ou en descendant (**desc**). * Exemple connaître la liste villes par ordre alphabétique : <code sql> select nom as Ville from icn_ville order by nom asc ; </code> **Résultat :** ^Ville^ |Aast| |Abainville| |Abancourt| ... soit **36 701 lignes**.

Question 4 : Ecrire la requête SQL qui donne la liste des villes **selon le nom d'habitants** par ordre **décroissant** (indiquer la ville la plus peuplée premier)

==== La sélection ==== J'utilise la **sélection** si je ne souhaite avoir des données qui réponde à une **condition** en utilisant la clause **where** :
* Exemple : avoir toutes les information de la ville de Panazol <code sql> select * from icn*ville where nom = 'panazol'; </code>* ***Résultat :*** *1 ville(s) trouvée(s) ! ^dep ^nom ^cp ^nb*hab*2010 ^nb*hab*1999 ^nb*hab*2012 ^dens ^surf ^longitude ^latitude ^alt*min ^alt_max^ |87 |Panazol |87350 |10392 |9727 |10100 |518 |20 |1.3 |45.8333 |215 |351| Voici les opérateurs utilisables : === Les opérateurs de comparaison et logiques === ^Opérateur de comparaison^Description^Opérateurs logiques^Description^ ^=|égal à^and|les deux conditions doivent être vérifiées simultanément| ^<|inférieur à^or|au moins une des deux conditions doit être vérifiée| ^>|supérieur à| ^⇐|inférieur ou égal| ^>=|supérieur ou égal| ^<>|différent de|

Question 5 : Ecrire la requête SQL qui donne la liste des villes qui ont **plus de 5000 habitants** en **2012**. La requête SQL doit renvoyer uniquement **2 007 lignes**.

=== Les opérateur d'appartenance (ou non) d'un élément à un intervalle === ^BETWEEN^appartenance| ^NOT BETWEEN^non appartenance| Exemple : connaître les villes dont l'altitude maximale est comprise entre 200 et 300 m : <code sql> select nom as ville, alt*max*
*from icn*ville where alt*max between 200 and 300; </code>* ***Résultat :*** *^ville ^alt*max^ |Ozan |205| |Cormoranche-sur-Saône |211| |Replonges |207| ...

Question 6 : Ecrire la requête SQL qui donne la liste des villes dont la superficie est comprise entre 100 et 300 kilomètres carré. La requête SQL doit renvoyer uniquement **144 lignes**.

=== Les opérateur de comparaison de chaînes de caracatères === ^LIKE^comparaison de chaînes (identiques)| ^NOT LIKE^chaîne différente| % permet de remplacer n caractères _ permet de remplacer 1 caractère * Exemple : connaître les villes dont le nom commence par Limoges : <code sql> select nom as ville from icn_ville where nom like 'limoges%'; </code> **Résultat :** ^ville^ |Limoges-Fourches| |Limoges|

Question 7 : Ecrire la requête SQL qui donne la liste des villes dont le nom contient les caractères **paris**. La requête SQL doit renvoyer uniquement **10 lignes**.

==== Les activités ... ====

Je reviens à la liste des activités.