

T. D. n° 4

Sélection de données en MySQL

Résumé

Ce document est le TD n° 4 du module **Base de données**. Il reprend rapidement des éléments du cours et propose une mise en pratique interactive des règles de sélection de données.

1 Sélection de données

Please select a payment method:

Me:



source : joke4fun.com

1.1 Retour à MySQL

Reprenez votre base de donnée. Cette base contient table *Animal* doit être composée *a minima* de six colonnes : *id*, *espece*, *sexe*, *date_naissance*, *nom* et *commentaires*.

1.2 Sélection

Pour sélectionner toutes les colonnes d'une table, on réalise la commande suivante :

```
SELECT *  
FROM Table;
```

Donner explicitement le nom des colonnes dont vous avez besoin vous demande d'écrire davantage, mais présente deux avantages :

- d'une part, vous êtes certains de ce que vous récupérez

- vous récupérez uniquement ce dont vous avez vraiment besoin, ce qui permet d'économiser des ressources.

Pour sélectionner plusieurs colonnes, on utilise la syntaxe suivante :

```
SELECT col1, ..., coln,  
FROM Table;
```

La sélection d'une ou plusieurs ligne se fait donc via une ou plusieurs conditions à respecter. Ces conditions sont associés aux valeurs d'une ou plusieurs colonnes. Pour préciser ces conditions, on utilisera la clause **WHERE**.

WHERE permet de restreindre les résultats selon des critères de recherche. Sa syntaxe est la suivante

```
SELECT col1, col2,...  
  
FROM Table  
  
WHERE colx operateur condition;
```

À vous !

- Sélectionnez l'espèce, le nom et le sexe des animaux présents dans la table *Animal*.
- Sélectionner l'espèce *chien*
- Rappelez l'opérateur "DIFFÉRENT DE" en SQL
- Sélectionner tous les animaux née avant le 2008-01-01
- Sélectionner tous les animaux sauf les chats

2 Sélection avec conditions multiples

Si vous souhaitez mettre plusieurs conditions, la syntaxe est :

```
SELECT *  
  
FROM table  
  
WHERE colx operateur condition  
  
AND coly operateur condition ;
```

Une autre syntaxe possible est d'utiliser l'opérateur "ET"

```
SELECT *  
  
FROM table  
  
WHERE colx operateur condition  
  
&& coly operateur condition ;
```

À vous !

- Sélectionner les animaux qui sont à la fois des chats ET des mâles.
- Rappelez l'opérateur "OU BIEN" en SQL (deux opérateurs sont possibles).
- Sélectionnez des tortues et des perroquets. C'est à dire les animaux qui sont des tortues OU BIEN des perroquets.
- Sélectionnez tous les animaux femelles sauf les chiennes
- Cherchez la définition de l'opérateur SQL XOR
- Sélection des animaux qui sont soit des mâles, soit des perroquets (mais pas les deux).

3 Sélection avec conditions hiérarchique

3.1 Arbre algébrique

Il est parfois nécessaire de décomposer la requêtes en sous niveaux lorsque qu'on souhaite avoir des conditions sur différentes colonne. Pour cela, je vous conseil d'écrire votre requête sur papier via un arbre algébrique.

Concernant la syntaxe, comme pour les opérations algébriques classiques, les parenthèse servent à définir la priorité dans le traitement.

```
SELECT *
FROM Table
WHERE colx opérateur condition
opérateur
( colx opérateur condition
opérateur
( colx opérateur condition
opérateur
( colx opérateur condition
opérateur
...
)
)
) ;
```

À vous !

- a) On souhaite obtenir les animaux qui sont soit née après 2009, soit des chats mâles, soit des chats femelles. Dans le cas des chats femelles, ces dernières doivent être nées avant juin 2017.
 - Formulez cette demande via un arbre algébrique.
 - Formulez cette demande via une requêtes MySQL.
- b) Sélectionnez les animaux sans commentaires en utilisant l'opérateur "=".
- c) Sélectionnez les animaux avec commentaires en utilisant l'opérateur "<>".
- d) Sélectionnez les animaux avec commentaires en utilisant l'opérateur "<=>".
- e) Sélectionnez les animaux sans commentaires en utilisant l'opérateur "IS".
- f) Sélectionnez les animaux avec commentaires en utilisant l'opérateur "IS NOT".
- g) Que pouvez vous conclure sur ces opérateurs lorsqu'ils sont utilisés avec NULL.

3.2 Contrainte d'affichage

Il est possible de contraindre l'affichage de votre requête grâce à l'opérateur `LIMIT` qui limitera le nombre de ligne affichées, et l'opérateur `ORDER BY` qui fera un affichage trié selon la colonne spécifié. Dans le cas de l'opérateur `LIMIT`, il est possible de spécifier si vous voulez que l'affichage soit décalé (c'est à dire n'afficher qu'à partir de la n-ème ligne) via l'opérateur `OFFSET`.

Voici la syntaxe pour l'opérateur `LIMIT` :

```
SELECT *  
    FROM Table  
LIMIT nombre_de_lignes [OFFSET decalage];
```

Voici la syntaxe pour l'opérateur `ORDER BY` :

```
SELECT *  
    FROM Table  
ORDER BY nom_colonne;
```

Si vous souhaitez afficher les valeurs possibles que contient une colonne, vous pouvez utiliser l'opérateur `DISTINCT` via la syntaxe suivante :

```
SELECT DISTINCT nom_colonne  
    FROM Table ;
```

À vous !

- a) Affichez 6 animaux de la table `animal` sachant qu'ils sont affichés par ordre croissant d'`id` et que les deux premier sont masqués.
- b) Affichez les 10 premier chiens sachant qu'ils sont affichés par date de naissance croissante.
- c) Affichez les 10 premier animaux sachant qu'ils sont affichés par espèce (ordre alphabétique) puis par date de naissance croissante.

- d) Listez toutes les espèces distinctes
- e) Listez tous les commentaires distincts

3.3 Recherche pour une chaîne de caractères

Si vous souhaitez faire une recherche sur des chaînes de caractère, vous pouvez utiliser l'opérateur `LIKE` et rajouter l'opérateur spécifiant votre condition :

- Commence par ... (un lettre ou bien le début d'une phrase) : Lettre/texte + Opérateur `%`
- Contient ... : Opérateur `%` + Lettre/texte + Opérateur `%`
- Fini par ... : Opérateur `%` + Lettre/texte

Il est bien sur possible de combiner ces requêtes.

Voici trois exemples de syntaxe :

```
SELECT *
  FROM Table
WHERE nom LIKE 'r%'; -- nom commençant par r
```

```
SELECT *
  FROM Table
WHERE nom LIKE '%ol%'; -- nom contenant ol
```

```
SELECT *
  FROM Table
WHERE nom LIKE '%o'; -- nom finissant par o
```

Si vous souhaitez chercher le caractère "%" il vous faudra ajouter le caractère d'échappement "%\".

À vous !

- a) Affichez les animaux dont le nom commence par "b".
- b) Affichez les animaux dont le nom fini par "o".
- c) Affichez les animaux dont le nom contient "er".
- d) Affichez les animaux dont le nom commence par "F" et contient "er".
- e) Affichez les animaux dont le nom contient "er" et fini par "o".
- f) Affichez les animaux dont le commentaire contient "%".
- g) Affichez les animaux dont le nom contient pas "o".

3.4 Recherche dans un intervalle ou dans un set de critère

Il est également possible de chercher des valeurs dans un intervalle de valeur continue ou un ensemble de valeurs discrètes. Cette recherche ne diffère pas vraiment d'une requête avec `WHERE` et plusieurs `AND`, elle est simplement plus lisible. Pour les intervalles de valeurs continue, on utilisera les opérateurs `BETWEEN` et `AND` avec la syntaxe suivante :

```
SELECT *  
WHERE nom_colonne BETWEEN valeur_min AND valeur_max;
```

Pour l'ensemble de valeurs on utilisera l'opérateur **IN** suivi des valeurs possibles entre parenthèses (séparé par une virgule).

```
SELECT *  
WHERE nom_colonne IN (val1,val2,...,valn) ;
```

À vous !

- a) En utilisant l'opérateur **BETWEEN**, affichez les animaux nées entre le '2008-01-05' et le '2009-03-23'
- b) Peut on formuler une requête similaire mais sans utiliser l'opérateur **BETWEEN** ?
- c) En utilisant l'opérateur **IN** cherchez les animaux dont le nom appartient à la liste suivante : Moka, Bilba, Tortilla, Balou, Dana, Redbul, Gingko.
- d) Formulez une requête similaire mais sans utiliser l'opérateur **IN**.