

TP : Keywords

1. Creer une base de données appelée keyword dans phpMyAdmin
2. Importer la base existante "users.sql".
=> Cliquer sur la base de données > Importer > choisir le fichier > importer
3. Testez les différentes requêtes ci-dessous

Mysql dispose de nombreux mots clefs principalement utilisés pour sélectionner les données, nous allons voir en détail tous les mots clefs existant, vous ne les utiliserez pas tous mais vous pourrez vous reporter à cet exercice plus tard pour utiliser les bons mots clefs.

• SELECT DISTINCT

Select distinct permet de sélectionner les données uniques, testez sur phpmyadmin, lancez la requête suivante :

```
SELECT prenom,nom,pays from `users` where 1
```

Comme vous pouvez le voir, il y a des utilisateurs ayant plusieurs fois le même pays, pour obtenir la liste des pays sans les doubles, lancez la commande suivante :

```
SELECT DISTINCT pays from `users` where 1
```

Les pays en double n'apparaissent plus.

● ORDER BY

Order by permet de trier les résultats par rapport aux valeurs d'une colonne , lancez la requête suivante :

```
SELECT * from `users` where 1 ORDER BY nom ASC
```

Les utilisateurs sont retournés par ordre alphabétique (de a à z) en utilisant la colonne "nom".

```
SELECT * from `users` where 1 ORDER BY nom DESC
```

Les utilisateurs sont retournés par ordre alphabétique (de z à a) en utilisant la colonne "nom".

A retenir :

ASC => Du plus petit au plus grand

DESC => Du plus grand au plus petit

Moyen mnémotechnique : ASCendant , DESCendant

- **MIN et MAX**

Min permet de sélectionner la valeur minimale d'une colonne parmi les enregistrements d'une table.

Max permet de sélectionner la valeur maximale d'une colonne parmis les enregistrements. Lancez la requête suivante :

```
SELECT MIN(argent) from `users` where 1
```

La requête va retourner la valeur la plus petite pour la colonne "argent" soit 150.

```
SELECT MAX(argent) from `users` where 1
```

La requête va retourner la valeur la plus élevée pour la colonne "argent" soit 99999.

Note :

il est courant d'utiliser un alias pour faciliter la récupération de la valeur, lancez :

```
SELECT MIN(argent) as argentMin from `users` where 1
```

Regardez bien le nom de la colonne dans le résultat retourné, il est écrit argentMin au lieu de MIN(argent)

- COUNT, AVG et SUM

Count permet de compter le nombre d'enregistrements présents dans une table qui remplisse la condition spécifiée par la clause where , lancez :

```
SELECT count(*) from `users` where argent<50000
```

Ici, je vais demander à mysql de compter le nombre d'utilisateurs qui ont moins de 50000 comme valeur de colonne "argent"

Avg permet de renvoyer la moyenne des valeurs d'une colonne qui remplit la condition spécifiée dans la clause where, lancez :

```
SELECT avg(argent) from `users` where 1
```

Mysql me retourne la moyenne de la colonne "argent" pour tous les utilisateurs.
Sum permet de renvoyer la somme des valeurs d'une colonne qui remplit la condition de la clause where, lancez :

```
SELECT sum(argent) from `users` where 1
```

Mysql me retourne la somme de la colonne argent de tous les enregistrements de la table "users"

- **LIKE**

L'opérateur LIKE agit un peu comme =, mais il permet d'utiliser des expressions.

Lancez :

```
SELECT * FROM `users` WHERE prenom LIKE 'j%'
```

La requête va vous retourner tous les utilisateurs dont la colonne prenom commence par la lettre j.

Lancez :

```
SELECT * FROM `users` WHERE prenom like '%s'
```

La requête va vous retourner tous les utilisateurs dont la colonne prénom se termine par la lettre s.

Lancez :

```
SELECT * FROM `users` WHERE prenom like '%a%'
```

La requête va vous retourner tous les utilisateurs dont la colonne prenom contient la lettre a (quelque soit la position).