



13) Modification d'un enregistrement

Pour modifier des données la requête est relativement simple c'est avec les mots clé UPDATE... SET :

- UPDATE [nom table] SET champ=valeur, champ=valeur WHERE [conditions]

On va modifier le nom et prénom d'un enregistrement.

(Exo) Ouvrez l'onglet SQL et essayer de modifier le prénom et nom de votre 3eme utilisateur en utilisant la commande ci-dessus.

On peut s'amuser, par exemple si on veut modifier la ville de tout les hommes on pourra faire :

- UPDATE users SET city= "Anvers" WHERE sexe = "M"

(Exo) Crée CET enregistrement(Beyonce...) avec les commandes de création.

8	Cool	Blues	X	2000-02-24	Anvers	0
9	Beyonce	Knowles	F	1980-11-20	Kinshasa	0

Ensuite modifier sa date de naissance (4 Septembre 1981) et sa ville (Houston). Montrez !

Le UPDATE, comme toutes les autres commandes, est irréversible donc c'est bien de mettre des LIMIT si vous voulez être sûr de votre coup.

14) Sélectionner des données

Maintenant que l'on sait comment manipuler des données dans les tables, on va passer à la sélection de données. C'est une opération que vous allez rencontrer le plus souvent lorsque vous allez ensuite développer votre site web. En effet lorsque vous allez vouloir créer un site, il va vous falloir créer les requêtes qui vont vous permettre de récupérer les articles par exemple ou encore créer une requête pour récupérer une catégorie etc. Vous allez en gros devoir utiliser beaucoup de requêtes de récupération de données.

Pour récupérer les données la requête est relativement simple, on fait SELECT suivi des champs à récupérer, dans notre cas, dans la table utilisateurs, on voudra récupérer que les prénoms et les noms puis on précise qu'ils viennent de la table users :

- SELECT firstname, lastname FROM users

Dans ma table cela donne ce résultat :



The screenshot shows a SQL query editor interface. At the top, a text box contains the query: `SELECT firstname, lastname FROM `users``. Below the text box are several links: [Profilage](#), [\[Éditer en ligne \]](#), [\[Éditer \]](#), [\[Expliquer SQL \]](#), and [\[Créer le code source \]](#). Below these links are two checkboxes: ☐ Tout afficher and a dropdown menu for 'Nombre de lignes' set to 25. Below the dropdown is a 'Filtrer les lignes:' label. Below the filter is a '+ Options' link. Below the options is a table with two columns: 'firstname' and 'lastname'. The table contains five rows of data, each with a checkbox, an 'Éditer' icon, a 'Copier' icon, and a 'Supprimer' icon. The data rows are: James Bond, Jack Bauer, Lara Croft, Cool Blues, and Beyonce Knowles.

	firstname	lastname
<input type="checkbox"/> Éditer Copier Supprimer	James	Bond
<input type="checkbox"/> Éditer Copier Supprimer	Jack	Bauer
<input type="checkbox"/> Éditer Copier Supprimer	Lara	Croft
<input type="checkbox"/> Éditer Copier Supprimer	Cool	Blues
<input type="checkbox"/> Éditer Copier Supprimer	Beyonce	Knowles

La syntaxe c'est donc :

- SELECT champ, champ FROM [nom table] WHERE [conditions]

A noter que pour sélectionner tous les éléments d'une table il suffit de mettre à la suite du SELECT une * :

- SELECT * FROM users

(Exo) Récupérer le nom et prénom de l'utilisateur qui a son ID à 1.

Récupérer la ville de toutes les femmes.

Récupérer la date de naissance de Beyonce.

Pagination

Pour ce qui est de l'ajout d'une limite, on peut choisir l'endroit où la limite commence.

```
SELECT * FROM `users` LIMIT 1, 2
```

☐ Pro

Options

		id_user	firstname	lastname	sexe	date_of_birth	city
<input type="checkbox"/>	Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York
<input type="checkbox"/>	Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington

Vous voyez j'ai fait un SELECT ALL (*) mais au lieu de m'afficher tout les enregistrement, il m'a juste affiché ce que je demandais dans LIMIT donc de 1 à 2 (donc le 2^{ème} et 3^{ème} enregistrement). Ça se lit, je veux 2 enregistrement à partir du 1. Si on avait mis LIMIT 0, 2 ça aurait pris le premier et deuxième enregistrement. Un peu comme les tableaux et listes dans quasi tous les langages de programmation, ça commence à 0.

C'est grâce à ça qu'on peut faire des paginations, par exemple sur une table de 100 enregistrements et qu'on veut les montrer 10 par 10, on pourra faire à chaque fois LIMIT 0, 10, puis pour la page 2 : LIMIT 10, 10 puis LIMIT 20, 10 puis LIMIT 30, 10 etc.

(Exo) Affichez 2 enregistrements à partir du 3^{ème}. Montrez-moi une fois finit.

Ordre d'organisation des données

Pour pouvoir organiser vos enregistrement vous pouvez utiliser le mot clé ORDER BY. Si vous voulez afficher tout les utilisateurs en ordonnant par le nom de manière donc alphabétique (croissant). Il suffit de faire :

```
SELECT * FROM `users` ORDER BY lastname
```

☐ F

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : A

+ Options

		id_user	firstname	lastname	1	sexe	date_of_birth	city
<input type="checkbox"/>	Éditer Copier Supprimer	2	Jack	Bauer		M	2004-12-24	New-York
<input type="checkbox"/>	Éditer Copier Supprimer	8	Cool	Blues		X	2000-02-24	Anvers
<input type="checkbox"/>	Éditer Copier Supprimer	1	James	Bond		M	2007-07-07	London
<input type="checkbox"/>	Éditer Copier Supprimer	3	Lara	Croft		F	1997-08-01	Washington
<input type="checkbox"/>	Éditer Copier Supprimer	9	Beyonce	Knowles		F	1981-09-04	Houston

Ici on voit bien qu'il a bien récupéré les enregistrements par ordre croissant (alphabétique).

Pour avoir par ordre décroissant (du plus grand ou plus petit), il faut juste rajouter DESC.

The screenshot shows a SQL query editor with the following query: `SELECT * FROM `users` ORDER BY lastname DESC`. The word `DESC` is highlighted with a red box. Below the query, there are buttons for 'Profilage', 'Éditer en ligne', 'Éditer', 'Expliquer SQL', and 'Créer le code source PHP'. Further down, there are options for 'Tout afficher', 'Nombre de lignes' (set to 25), and 'Filtrer les lignes'. At the bottom, there is a section for 'Options' with a dropdown menu. Below this, a table of results is displayed with columns: `id_user`, `firstname`, `lastname`, and `sex`. The table contains five rows of data, sorted by lastname in descending order. The 'lastname' column is highlighted with a red box.

id_user	firstname	lastname	sex
9	Beyonce	Knowles	F
3	Lara	Croft	F
1	James	Bond	M
8	Cool	Blues	X
2	Jack	Bauer	M

A savoir que par défaut quand on met juste ORDER BY nom_champs, c'est équivalent à mettre ORDER BY nom_champs ASC. Testez le.

On sait aussi mettre plusieurs ordre, si on obtient par exemple un champ repris dans plusieurs enregistrement différents, on pourra lui donner de trier sur un second ordre.

(Exos)

- 1) Affichez les noms et prénoms de tout les utilisateurs par ordre alphabétique de leurs prénoms. Montrez-moi une fois finis.
- 2) Affichez les noms et dates de naissance des 3 premiers utilisateurs ordonné par leurs date de naissance, du plus jeune au plus vieux. Montrez-moi une fois finis.
- 3) Affichez les noms, prénoms et villes du 2^{ème} au 5^{ème} utilisateurs ordonné par leur ville, par ordre décroissant.

Bonus Affichez les noms et sexe des utilisateurs par ordre croissant par rapport au sexe et par ordre décroissant par rapport au nom. Vous constaterez un petit problème.

Le comptage COUNT()

On peut aussi savoir combien il y a d'utilisateur, combien d'utilisateur venant par exemple d'une même ville, ou il y a combien d'homme ou de femme etc. C'est avec le mot clé COUNT():

The screenshot shows a SQL query editor with the following query: `SELECT COUNT(*) FROM `users``. The query is highlighted with a red box. Below the query, there are buttons for 'Profilage', 'Éditer en ligne', 'Éditer', and 'Expliquer SQL'. Further down, there is a section for 'Options' with a dropdown menu. Below this, a table of results is displayed with columns: `COUNT(*)` and `5`. The `COUNT(*)` column is highlighted with a red box.

COUNT(*)
5

On voit bien qu'il y a 5 utilisateurs dans ma table. Pour savoir par exemple le nombre de personne s'appelant Jack, il faut mettre une condition :

```
SELECT COUNT(`firstname`) FROM `users` Where firstname="Jack"
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code so

+ Options

COUNT('firstname')
1

Les Alias (as)

On voit ici que le nom est archaïque, count('firstname'), si vous voulez vous pouvez renommer le nom de la table pour y voir plus clair en utilisant un Alias. C'est suivi d'un « as » qu'on fait ça.

```
SELECT COUNT(`firstname`) as "Le nombre de prenom Jack" FROM `users` Where  
firstname="Jack"
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actuali

+ Options

Le nombre de prenom Jack
1

(Exos)

- 1) Affichez le nombre de femmes en donnant comme titre de colonne le_nbr_femme.
- 2) Affichez le nombre d'hommes en donnant comme titre de colonne le_nbr_homme.

Le GROUP BY

Le GROUP BY va permettre de grouper les éléments. Par exemple si on veut en même temps compter le nombre d'homme et de femme.

```
SELECT COUNT(*) as le_nombre, sexe FROM `users` GROUP BY sexe
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: [

+ Options

le_nombre	sexe
2	M
2	F
1	X

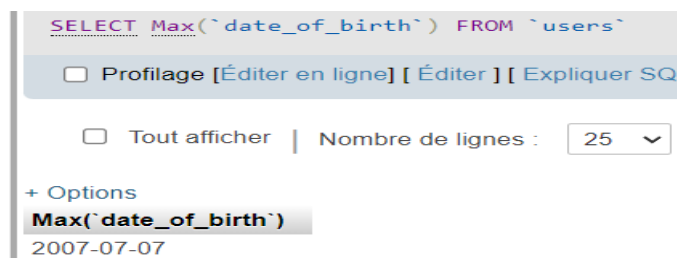
On peut faire un GROUP BY qui va regrouper tous les M qu'il va retrouver, donc dans la colonne le_nombre on aura 2 qui représentera les 2 hommes de ma table. Il fera pareil pour toutes les femmes et tous les autres.

Attention dans ma table tous mes utilisateurs ont un prénom, par contre si on avait mis prénom pouvait avoir un champ NULL et qu'on en avait un. En faisant COUNT(firstname) j'aurai eu 4 comme résultat alors que j'en aurai 5. Il est donc préférable de mettre COUNT(*).

En gros count ne compte pas les lignes qui ont pour valeur NULL ce champ-là.

Maximum, Minimum et Average

On a la possibilité aussi de chercher les valeurs maximum ou minimum d'un champ. Par exemple pour la date de naissance **MAX**(champ) :



```
SELECT Max(`date_of_birth`) FROM `users`
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL]

☐ Tout afficher | Nombre de lignes : 25

+ Options

Max(`date_of_birth`)
2007-07-07

On peut essayer de rajouter « group by sexe » par exemple, et on verra donc la date maximum par sexe.

Bien entendu c'est bien mieux avec des champs comme l'âge ou le prix etc.

On peut aussi bien entendu faire pareil pour le minimum, **MIN**(champ).

On peut aussi faire une moyenne d'un champ, par exemple si on avait un champ âge ou poids ou prix pour un article etc. Il suffirait d'utiliser **AVG**(champ) :

- SELECT AVG(nom_champ) FROM nomtable

On peut aussi demander la somme en faisant **SUM**(champ).

Petit rappels nous somme en Case Insensitive donc vous pouvez mettre en minuscule ou majuscule c'est pareil même pour les champs.

(Exos) Ajoutez en SQL une colonne « weight_kg » de type entier de taille 5 et NOT NULL.

Ajoutez des poids différents à tous les utilisateurs sauf le 1^{er} et le dernier qui auront le même poids. Affichez le poids Minimum des hommes. Affichez le poids maximum des femmes. Affichez la somme des poids des utilisateurs.

Affichez la moyenne des poids des utilisateurs en donnant comme nom de colonne La moyenne des poids des utilisateurs. Affichez la moyenne des poids des femmes, des hommes et des autres en mettant comme nom de colonne la_moyenne.