












Les opérateurs

Nous avons aussi des opérateurs classique de comparaisons : =, !=, >, >=, <, <=, <> et NOT.

```
SELECT * FROM `users` where date_of_birth > "2000/01/01"
```

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

Options

			id_user	firstname	lastname	sexe	date_of_birth	city				
<input type="checkbox"/>		Éditer		Copier		Supprimer	1	James	Bond	M	2007-07-07	London
<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

On voit bien qu'il aura pris tous les utilisateurs ayant une date supérieur au 1^{er} janvier 2000 .

L'opérateur « != » c'est pour dire une négation donc quand on met par exemple sexe != M, ca voudra dire on ne veut aucun homme. « != » c'est équivalent à « <> ».

```
SELECT * FROM `users` WHERE sexe != "M"
```

☐ Profilage [

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Trier par clé :

Options

						id_user	firstname	lastname	sexe	date_of_birth	city	weight
<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	3	Lara	Croft	F	1997-08-01	Washington	
<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	8	Cool	Blues	X	2000-02-24	Anvers	
<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	<div><div><div></div></div><div><div></div></div></div>	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	

(Exos)

- 1) Affichez les utilisateurs qui sont né avant 2000.
- 2) Affichez les utilisateurs qui ont plus de 70kg.
- 3) Affichez les femmes qui ont plus de 65 Kg.
- 4) Affichez les hommes qui ont moins de 90 kg.

Le Like

On peut faire une petite condition spéciale, c'est une recherche partiel. Par exemple je veux récupérer les utilisateurs dont la première lettre est « J ». Voici la syntaxe :

```
SELECT * FROM `users` WHERE firstname LIKE "J%"
```

☐ Profil

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

Options

	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	James	Bond	M	2007-07-07	London	70
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124

Ici ce que signifie le « J% », c'est donne moi tout les prénoms qui commence par « J » et qui ont n'importe quoi d'autres derrière. Donc si on avait fait par exemple « %E », ça voudrait dire, donne moi tout ce qui se termine par E. **NOT LIKE** c'est la négation.

Attention si vous faites par exemple %A, si vous avez JACK et LARA dans votre table il montrera que LARA, parce que JACK remplit bien le fait qu'on ait n'importe quoi devant le A mais pas derrière.

Vous l'aurez compris pour pouvoir afficher aussi JACK il aurait fallu rajouter un %. Donc faire %A%, de ce fait on prend n'importe quoi avant le A ainsi qu'après.

(Exos)

- 1) Affichez tous les prénoms et noms, dont le nom commence par B par ordre décroissant sur le nom.
- 2) Affichez tous les noms et prénoms, des femmes ayant A dans leurs prénoms, par ordre croissant sur le prénoms.
- 3) Affichez tous les noms et prénoms des hommes n'ayant pas A dans leurs prénoms.
- 4) Affichez tous les utilisateurs dont la dernière lettre du prénom se termine par un E.
- 5) Affichez tous les noms, prénoms et poids des utilisateurs ayant E dans leurs noms ordonner par leur poids, du plus gros au moins gros.

Le OR et AND

Si on veut combiner des conditions, on peut utiliser ces opérateurs. Le OR (le ou) comme son nom l'indique va permettre d'afficher en fonction de chaque condition qu'on rajoute.

```
SELECT * FROM `users` WHERE `id_user`=1 OR `id_user`= 2
```

☐ Profilage [Éditer]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options

	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	James	Bond	M	2007-07-07	London	70
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124

On voit bien ici qu'on a affiché les deux utilisateurs. Affiche moi le user 1 ou le user 2. Le OR va vraiment chercher séparément les conditions. Il trouve le 1 il affiche ensuite deuxième condition il trouve le 2 et l'affiche aussi.

Le AND lui va plutôt vérifier que les conditions ensemble soient remplies. Donc si on fait la même commande que pour le OR. On demande d'afficher le user 1 ET le user 2. Voyons le résultat.

```
SELECT * FROM `users` WHERE `id_user`=1 AND `id_user`=2
```

id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
---------	-----------	----------	------	---------------	------	-----------

Opérations sur les résultats de la requête

☐ Créer une vue

On voit bien ici qu'on a aucun résultat. Par ce qu'il n'existe aucun utilisateur/enregistrement qui a comme id_user 1 et en même temps id_user 2. Par contre une autre condition aurait fonctionner par exemple.

```
SELECT * FROM `users` WHERE `sexe`="M" AND `weight_kg` > 75
```

☐ Profilage

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124

On voit bien ici que Jack Bauer remplit les 2 conditions, c'est bien un homme et il fait bien plus de 75 Kg.

(Exos) (Quand je dis affichez les femmes/hommes ça veut dire prénoms et noms)

- 1) Affichez les femmes, leurs sexes et leurs poids faisant plus de 50 Kilos et ordonnez les par prénoms décroissant.
- 2) Affichez les hommes, leurs sexes et leurs villes, dont la ville contient un A, par ordre sur le nom.
- 3) Affichez les utilisateurs qui sont nés avant le 15 Janvier 1999 et faisant plus de 65 kilos. Par ordre
- 4) Affichez les femmes ou hommes (+ leurs poids et leurs dates de naissances) qui ont plus de 90 kilos ou leurs prénoms commençant par un B ou qui sont né après 1990. Ordonnez du plus jeune au plus vieux.
- 5) Affichez les 3 premiers utilisateurs ayant dans leurs prénoms un C ou contenant dans leurs noms un E. Ordonnez par leur noms.

Le IN

Nous avons vu qu'on pouvait utiliser par exemple plusieurs OR de suite pour enchaîner les conditions. Donc si on veut mettre comme conditions, d'afficher tous les utilisateurs avec comme id_users ou 1 ou 2 ou 3 ou 5. On aurait du écrire à chaque fois Where id_user=1 OR id_user = 2 OR id_user = 3 OR id_user = 5. Vous comprenez vite que si on veut encore beaucoup plus d'utilisateurs, ca peut rallonger la requête SQL.

```
SELECT * FROM `users` WHERE `id_user`=1 OR `id_user`=2 OR `id_user`=3 OR `id_user` = 5
```

Pour simplifier cela, en SQL on a le IN qui s'utilise de cette manière.

```
SELECT * FROM `users` WHERE `id_user` IN (1,2,3,5)
```

Vous pouvez voir sa syntaxe. Vous pouvez bien entendu combinez avec d'autres conditions à la suite. Par exemple

```
SELECT * FROM `users` WHERE `id_user` IN (1,2,3,5) OR `firstname` IN ("Jack", "James", "Beyonce")
```

☐ Profilage [t]


☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

+ Options

	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	James	Bond	M	2007-07-07	London	70
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70

(Exos récap)

- 0) Ajoutez 4 utilisateurs différents dont 1 femme , 1 homme et les deux autres sont des Autres (X). 2 venants de Bruxelles et 2 de Anvers, dont 2 sont né avant 2000 et 2 après 2000. Je veux aussi des poids et dates différentes. MONTREZ MOI avant d'exécuter la commande. (PDF part 2)
- 1) Affichez les utilisateurs qui ont comme nom soit Bond, soit Bauer, soit Knowles ou Croft et qui pèse moins de 85 Kilos. Ordonnez par leur prénoms.
- 2) Affichez les 6 premiers utilisateurs qui ont comme ID soit 1, soit 3, soit 5 ou 7, ou qui soient des hommes ou des autres.
- 3) Affichez le nombre de personnes qui sont né avant 2000, regroupé par leur sexe. En donnant comme titre à ce nombre : « Le nombre des personnes né avant 2000 ».
- 4) Affichez les 2 enregistrements à partir du quatrième enregistrement.
- 5) Affichez les 2 enregistrements à partir du cinquième qui ont sont né avant juin 2010. Organiser par noms décroissant.
- 6) Affichez le poids de la plus grosses personnes qui est né après 2000. En donnant comme titre : « Le poids de la plus grosse personnes né après 2000 »
- 7) Affichez la moyenne des poids des Bruxellois.
- 8) Affichez le poids de la personnes la plus légère de chaque catégorie des sexes. Donnez comme nom : « le poids de la personne la moins grosses ».
- 9) Affichez les prénoms et date de naissance des personnes né entre février 1990 et mars 2020. Ordonnez du plus vieux au plus jeune.
- 10) Affichez les 3 utilisateurs à partir du 5^{ème} qui ont un E ou un A dans leurs prénoms. Ordonnez par leurs sexe.
- 11) Affichez la moyenne des poids des Anversois et des Bruxellois. Avec ce titre : « La moyenne des poids »



The screenshot shows a table with two rows. The first row is for 'Anvers' with an average weight of 83.6667. The second row is for 'Bruxelles' with an average weight of 81.0000. Each row has a checkbox, an edit icon, and buttons for 'Copier' and 'Supprimer'.

	La moyenne des poids	city
<input type="checkbox"/> Éditer Copier Supprimer	83.6667	Anvers
<input type="checkbox"/> Éditer Copier Supprimer	81.0000	Bruxelles

- 12) Affichez tous les utilisateurs ne venant ni de Bruxelles ni de Anvers, par ordre sur le nom de leurs villes.
- 13) Affichez la somme des poids de tous les utilisateurs ne venant pas de Bruxelles. En donnant comme nom : « La somme des poids de tous les utilisateurs en dehors de Bruxelles ».
- 14) Affichez les 4 derniers utilisateurs du plus jeune au plus vieux.
- 15) Affichez les utilisateurs dont leur ville est soit Bruxelles, soit Anvers, soit New-York ou Houston et qui sont né avant décembre 1990. Du plus gros au moins gros.