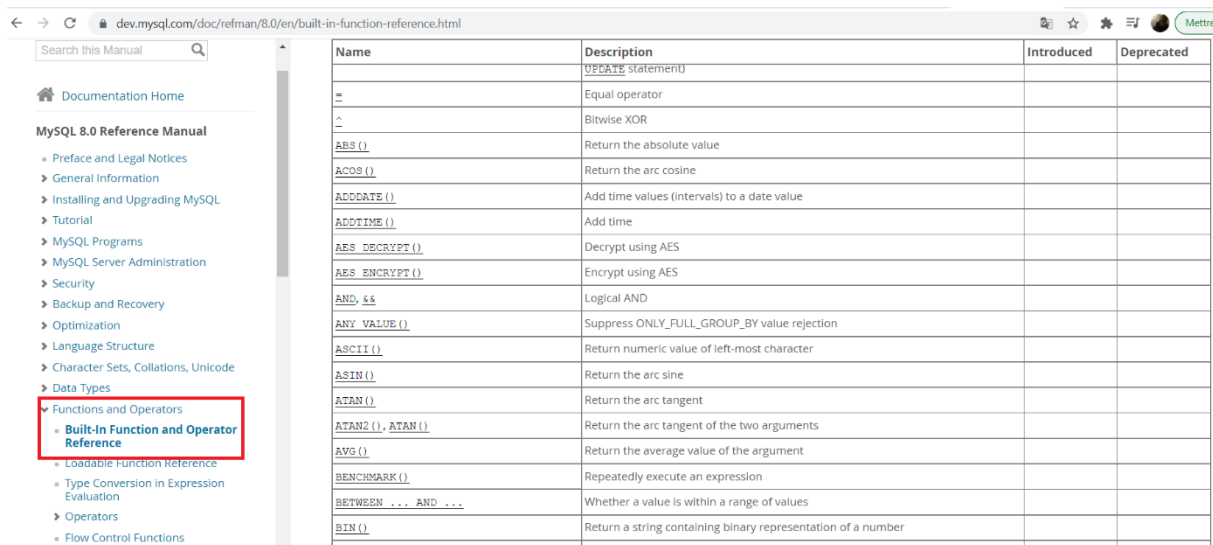


15) Les Fonctions numériques

Nous avons vu comment réaliser les différentes requêtes qui nous permettent de modifier et de récupérer les informations. En plus de ces différentes requêtes là, MySQL a toutes une série de fonctions qui vous permettent de manipuler les données et de les récupérer de différentes manières. L'ensemble de ces fonctions est décrit dans la documentation (dans le manuel de MySQL). Si vous tapez sur votre navigateur : [fonctions and operators MySQL](https://dev.mysql.com/doc/refman/8.0/en/built-in-function-reference.html). Si vous cliquez sur Built-in Function... Vous tombez sur cette page avec la liste de tous les opérateurs et fonctions.



Name	Description	Introduced	Deprecated
UPDATE	UPDATE statement()		
=	Equal operator		
^	Bitwise XOR		
ABS()	Return the absolute value		
ACOS()	Return the arc cosine		
ADDDATE()	Add time values (intervals) to a date value		
ADDTIME()	Add time		
AES_DECRYPT()	Decrypt using AES		
AES_ENCRYPT()	Encrypt using AES		
AND, &&	Logical AND		
ANY_VALUE()	Suppress ONLY_FULL_GROUP_BY value rejection		
ASCII()	Return numeric value of left-most character		
ASIN()	Return the arc sine		
ATAN()	Return the arc tangent		
ATAN2(), ATAN()	Return the arc tangent of the two arguments		
AVG()	Return the average value of the argument		
BENCHMARK()	Repeatedly execute an expression		
BETWEEN ... AND ...	Whether a value is within a range of values		
BIN()	Return a string containing binary representation of a number		

Comme vous le voyez il y en a beaucoup, on va en voir quelques-unes. Les fonctions essentielles que vous pourrez avoir à utiliser dans vos bases de données. Il y a plusieurs fonctions classiques en mathématique, lié au chiffre et nombre. On en a utilisé précédemment comme le **AVG()**, **SUM()**, **MAX()** et **MIN()**. Il y a aussi la valeur absolu **ABS()** ou encore la racine carré d'un nombre **SQRT()** ou arrondir un chiffre **ROUND()** ou **MOD()** pour modulo (division entière) ou encore la fonction **TRUNCATE()** qui permet de choisir le nombre de chiffre après la virgule.

Vous pouvez, par exemple, tester vos fonctions directement comme ci-dessous :

SELECT *, TRUNCATE(1.26542254 ,3) FROM `users` WHERE 1

☐ Profilage [Éditer en ligne] [Éditer] [Explique]

☐ 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	TRUNCATE(1.26542254 ,3)
<input type="checkbox"/> Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	1.265
<input type="checkbox"/> Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	1.265
<input type="checkbox"/> Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	1.265
<input type="checkbox"/> Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	1.265
<input type="checkbox"/> Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	1.265
<input type="checkbox"/> Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	1.265
<input type="checkbox"/> Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	1.265
<input type="checkbox"/> Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-08-23	Bruxelles	76	1.265

(Exos)

- 1) Affichez la racine carrée des poids, le nom et prénom de tous les utilisateurs.
- 2) Affichez-moi la racine carré avec 2 chiffres après la virgule de l'exercice 1.
- 3) Arrondissez-moi le résultat de la racine carré de l'exercice 1.

16) Les fonctions pour les chaines de caractères

Il y en a aussi plusieurs, vous pouvez le voir dans la documentation. Ceux que vous pourrez utiliser par exemple c'est la concaténation de plusieurs chaines de caractères en utilisant **CONCAT()**. Exemple on va essayer de concaténer le prénom et le nom de nos utilisateurs :

SELECT *, CONCAT(`firstname`, ' ', `lastname`) as "Le prénom et nom" FROM `users`

☐ Profilage [Éditer en ligne] [Éditer] [E]

☐ 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	Le prénom et nom
<input type="checkbox"/> Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	James Bond
<input type="checkbox"/> Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	Jack Bauer
<input type="checkbox"/> Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	Lara Croft
<input type="checkbox"/> Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	Cool Blues
<input type="checkbox"/> Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	Beyonce Knowles
<input type="checkbox"/> Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	Catelyne Jenner
<input type="checkbox"/> Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	Alicia Keys
<input type="checkbox"/> Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-08-23	Bruxelles	76	Hakima Darmouch
<input type="checkbox"/> Éditer Copier Supprimer	13	Kevin	Debruyne	M	1990-07-10	Anvers	79	Kevin Debruyne

On voit qu'il a concaténé le prénom et le nom, en laissant un espace entre grâce au « ' ' ».

On a aussi la fonction **LENGTH()**, qui va récupérer la taille d'une chaîne de caractères.

On a aussi les fonctions un peu plus spéciales comme **LOWER()** et **UPPER()** qui permettent de mettre en minuscule ou en majuscule la chaîne de caractère. La fonction **SUBSTR()** (**substring**) est une fonction que vous retrouverez dans beaucoup de langages de programmation, elle permet de couper la chaîne de caractères.

```
SELECT SUBSTR(`lastname`,2) FROM `users` WHERE 1
```

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtre

+ Options

```
SUBSTR(`lastname`,2)
ond
auer
```

Ici vous remarquerez que j'ai mis 2, pour dire qu'on veut commencer à partir du deuxième caractère donc par exemple pour « **Bond** », ça affichera « **ond** ».

On peut aussi donner une limite.

```
SELECT SUBSTR(`lastname`,2,3), `lastname` FROM `users` WHERE `firstname`="Beyonce"
```

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

+ Options

	SUBSTR(`lastname`,2,3)	lastname
<input type="checkbox"/> Éditer	now	Knowles

On voit bien qu'on commence au deuxième caractère et qu'on prend ensuite 3 caractères.

Attention, ici quand on donne la position on ne commence pas à 0 mais bien à 1.

On a la fonction **REPLACE()** qui permet de remplacer une chaîne de caractère par une autre.

```
SELECT REPLACE(`firstname`, "ame", "ok") FROM `users` W
```

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

+ Options

```
REPLACE(`firstname`, "ame", "ok")
Joks
Jack
Lara
Cool
```

Il a remplacé dans **JAMES** « **ame** » par « **ok** », ce qui donne **Joks**. **Attention** cette fonction fait attention au majuscule et minuscule (**Case Sensitive**). Elle est très utile cette fonction si vous voulez faire des modifications en masses avec le **Update**.

(Exos)

- 1) Affichez les utilisateurs qui sont né à Bruxelles ou à Anvers ou à London ou à New-York. Je veux une colonne en plus avec comme nom « Voici le genre, la ville et le poids » ainsi que leurs données séparées par des virgules et un espace. Pour le poids, je veux en kilos.

Voici le sexe, la ville et le poids.

M, London, 70kg

M, New-York, 124kg

- 2) Affichez les 5 premiers utilisateurs ainsi que le nombre de lettre dans leurs prénoms. En ajoutant comme nom de colonne « Le nombre de lettres dans le prénom ». Trier par ordre décroissant sur le nombre de lettre du prénom.

	weight_kg	Le nombre de lettres dans le prénom
les	86	8
isa	70	7

- 3) Affichez-moi les noms en majuscule et les prénoms en minuscule de tous les utilisateurs ayant un « a » dans leur nom ou prénom.
- 4) Affichez-moi les villes ainsi que les villes sans leurs 3 premières lettres.
- 5) Affichez-moi les villes ainsi que les villes que de la 4^{ème} lettre à la 7^{ème}.
- 6) Remplacez-moi dans les villes, « xelle » par « ssel ». En gros traduire Bruxelles en Anglais.
- 7) Traduisez de manière inverse London en Londres en remplaçant seulement les lettre nécessaire... ATTENTION à ne pas modifier d'autres villes, il faut une condition.

17) Les fonctions pour les dates

On va finir avec les **fonctions pour les dates**. Ces fonctions sont très utiles et vous pourrez les utiliser souvent. Avec MySQL vous pourrez récupérer les données en fonction d'un évènement temporelle. Par exemple voir les derniers articles posté pendant un mois etc.

Ici imaginons que je veux rajouter à un utilisateur un mois à son abonnement mensuel ou annuel etc. Donc je veux rajouter un mois à la date de fin de son abonnement. Dans notre cas ici je vais vous montrer via les dates de naissances, mais imaginons que ces dates de naissances soient des dates de fin de contrat ou d'abonnement.

On utilisera la fonction **ADDDATE()**, elle prend en premier paramètre une date, ensuite un nombre de jours.

```
SELECT *, ADDDATE(`date_of_birth`,10) FROM `users`
```

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

☐ 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	ADDDATE('date_of_birth',10)
<input type="checkbox"/> Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	2007-07-17
<input type="checkbox"/> Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	2005-01-03
<input type="checkbox"/> Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	1997-08-11
<input type="checkbox"/> Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	2000-03-05
<input type="checkbox"/> Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	1980-11-30
<input type="checkbox"/> Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	1970-12-20
<input type="checkbox"/> Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	1987-11-28
<input type="checkbox"/> Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-01-01	Bruxelles	76	1982-01-11
<input type="checkbox"/> Éditer Copier Supprimer	13	Kevin	Debruyne	M	1990-07-10	Anvers	79	1990-07-20

On voit bien qu'il a rajouté à nos dates d'anniversaire 10 jours.

Vous pouvez aussi préciser si vous voulez rajouter une année par exemple en tapant :

- **SELECT *, ADDDATE(date_of_birth, INTERVAL 1 YEAR) FROM users**

Avec le INTERVAL, vous pouvez rajouter différentes informations, par exemple en mois (MONTH), en semaine (WEEK), en jour (DAY), en heure (HOUR), en minute, etc.

Vous avez la fonction qui fait le contraire, donc qui va soustraire du temps à nos dates.

- **SELECT *, SUBDATE(date_of_birth, INTERVAL 1 YEAR) FROM users**

Sachez que l'équivalent de ces deux commandes existent : **DATE_SUB()** et **DATE_ADD()**. La seule différence c'est qu'ils ne prennent pas un entier en paramètre seul. Ils acceptent qu'avec **INTERVAL**.

On peut aussi via **SUBDATE()**, ressortir toutes les personnes qui ont moins de 30 ans par exemple :

```
SELECT * FROM `users` WHERE date_of_birth > SUBDATE("2021/11/18", INTERVAL 30 YEAR)
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le c

☐ 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 Copier Supprimer	1	James	Bond	M	2007-07-07	London	70
<input type="checkbox"/> Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124
<input type="checkbox"/> Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63
<input type="checkbox"/> Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100
<input type="checkbox"/> Éditer Copier Supprimer	14	Dark	Vador	M	2021-11-15	Tetouane	96

En gros ici on prend la date du jour en lui enlevant 30 années, ensuite cette date on la compare aux dates de naissance pour voir qui est né avant cette date. On peut aussi remplacer la date du jour écrite manuellement pour rendre plus dynamique la commande, et remplacer par **CURRENT_DATE**. Ça nous donnera la date du jour automatiquement :

- **SELECT * FROM users WHERE date_of_birth > SUBDATE(CURRENT_DATE, INTERVAL 30 YEAR)**

D'autres fonctions peuvent aussi donner la date du jour. (**CURDATE()**, **NOW()**, **CURRENT_TIME** pour l'heure du jour etc.)

```
SELECT CURRENT_DATE, NOW(), CURRENT_TIME FROM `users`
```

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer l

+ Options

CURRENT_DATE	NOW()	CURRENT_TIME
2021-11-18	2021-11-18 09:16:26	09:16:26
2021-11-18	2021-11-18 09:16:26	09:16:26
2021-11-18	2021-11-18 09:16:26	09:16:26

Il y a aussi la fonction **ADDTIME()** qui comme son nom l'indique peut rajouter des heures, minutes, secondes.

```
SELECT date_of_birth, ADDTIME(date_of_birth, "1:0:59") FROM users
```

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

Options

			date_of_birth	ADDTIME(date_of_birth, "1:0:59")
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2007-07-07 2007-07-07 01:00:59
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2004-12-24 2004-12-24 01:00:59

Il existe la fonction inverse qui s'appelle **SUBTIME()**.

(Exos)

- 1) Ajoutez 365 jours à toutes les femmes et hommes venant d'une ville contenant « e ».
- 2) Enlevez 15 mois à tous les autres(X).
- 3) Affichez-moi toutes les femmes et autres(X) qui ont moins de 36 ans, ordonné du plus vieux au plus jeune.
- 4) Affichez-moi les hommes qui ont plus de 18 ans, ordonné du plus jeune au plus vieux.
- 5) Affichez-moi les utilisateurs qui ont entre 10 et 25 ans, et ordonnez-les du plus vieux au plus jeune.