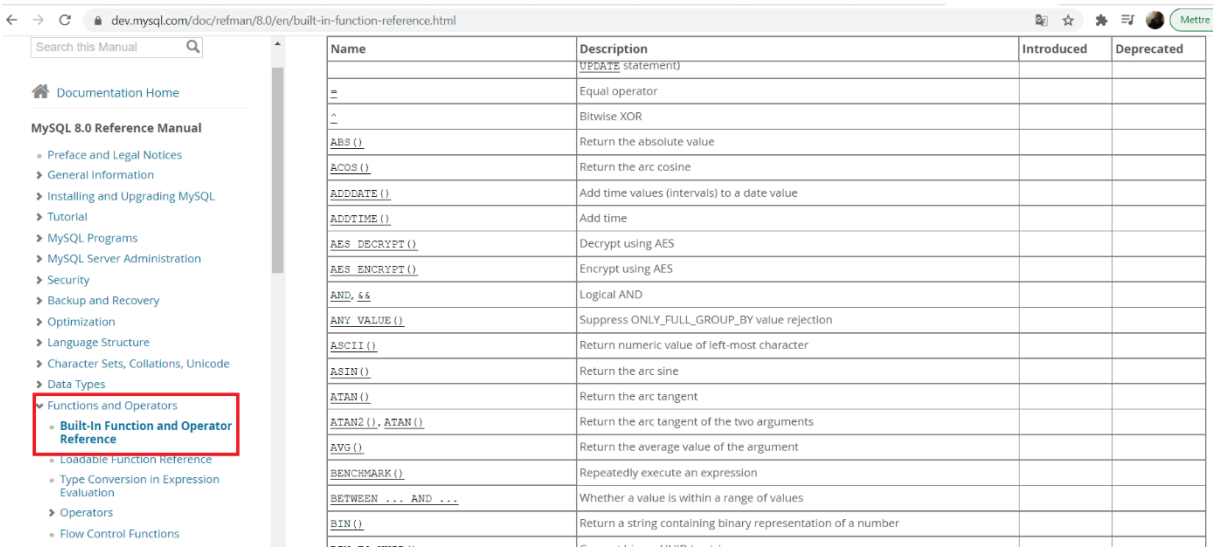
15) Les Fonctions numériques

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 : functions and operators MySQL. Si vous cliquez sur Built-in Function…Vous tombez sur cette page avec la liste de tous les opérateurs et fonctions.



Comme vous le voyez il y en 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)
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	1.265
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	1.265
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	1.265
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	1.265
Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	1.265
Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	1.265
Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	1.265
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
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	James Bond
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	Jack Bauer
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	Lara Croft
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	Cool Blues
Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	Beyonce Knowles
Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	Catelyne Jenner
Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	Alicia Keys
Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-08-23	Bruxelles	76	Hakima Darmouch
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 chaine 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 chaine de caractère. La fonction **SUBSTR**() (**substring**) est une fonction que vous retrouverez dans beaucoup de langages de programmation, elle permet de couper la chaine 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 tabl

+ Options

	SUBSTR(`lastname`,2,3)	lastname
Éditer Copier Supprimer	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 chaine 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 l'**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
sa	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 4ème lettre à la 7ème.**
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 lettres 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)
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	2007-07-17
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	2005-01-03
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	1997-08-11
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	2000-03-05
Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	1980-11-30
Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	1970-12-20
Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	1987-11-28
Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-01-01	Bruxelles	76	1982-01-11
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(e

Options

	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100
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: Chercher dans cette

Options

			date_of_birth	ADDTIME(date_of_birth, "1:0:59")
Éditer	Copier	Supprimer	2007-07-07	2007-07-07 01:00:59
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.**

La fonction **DATE_FORMAT()** va nous permettre d'afficher comme on le souhaite nos dates. C'est fini le format ISO avec années mois jours. On pourra mettre par exemple Jours Mois Années. Elle s'utilise via la syntaxe suivante :

- **SELECT DATE_FORMAT(date, format)**

Le premier paramètre correspond à une donnée de type DATE (ou DATETIME), tandis que le deuxième paramètre est une chaîne de caractère contenant le choix de formatage :

- %a : nom du jour de la semaine abrégé (Sun, Mon, … Sat)
- %b : nom du mois abrégé (Jan, Fev, … Dec)
- %c : numéro du mois (0, 1, 2, … 12) (numérique)
- %D : numéro du jour, avec le suffixe anglais (0th, 1st, 2nd, 3rd, …)
- %d : numéro du jour du mois, sous 2 décimales (00..31) (numérique)
- %e : numéro du jour du mois (0..31) (numérique)
- %f : microsecondes (000000..999999)
- %H : heure (00..23)
- %h : heure (01..12)
- %I : heure (01..12)
- %i : minutes (00..59) (numérique)
- %j : jour de l'année (001..366)
- %k : heure (0..23)
- %l : heure (1..12)
- %M : nom du mois (January..December)
- %m : mois (00..12) (numérique)
- %p : AM ou PM
- %r : heure au format 12-heures (hh:mm:ss suivi par AM ou PM)
- %S : secondes (00..59)
- %s : secondes (00..59)
- %T : heure au format 24 heures (hh:mm:ss)
- %U : Semaine (00..53), pour lequel le dimanche est le premier jour de la semaine; WEEK() mode 0
- %u : Semaine (00..53), pour lequel le lundi est le premier jour de la semaine; WEEK() mode 1
- %V : Semaine (01..53), pour lequel le dimanche est le premier jour de la semaine; WEEK() mode 2; utilisé avec %X
- %v : Semaine (01..53), pour lequel le lundi est le premier jour de la semaine; WEEK() mode 3; utilisé avec %x
- %W : nom du jour de la semaine (Sunday, Monday, … Saturday)
- %w : numéro du jour de la semaine (0=dimanche, 1=lundi, … 6=samedi)
- %X : numéro de la semaine de l'année, pour lequel le dimanche est le premier jour de la semaine, numérique, 4 digits ; utilisé avec %V
- %x : numéro de la semaine de l'année, pour lequel le lundi est le premier jour de la semaine, numérique, 4 digits ; utilisé avec %v
- %Y : année, numérique (avec 4 digits)
- %y : année, numérique (avec 2 digits)
- %% : un caractère %

Voici un exemple pour afficher nos dates de manières européennes :

```
SELECT *, DATE_FORMAT(date_of_birth, "%d/%m/%Y") as date_normale FROM `users` WHERE 1
```

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

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

+ Options

←T→	id_user	firstname	lastname	sexe	date_of_birth	city	weight_kg	date_normale
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	07/07/2007
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	24/12/2004
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	01/08/1997
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	24/02/2000
Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	20/11/1980
Éditer Copier Supprimer	10	Catelyne	Jenner	X	1970-12-10	Bruxelles	86	10/12/1970
Éditer Copier Supprimer	11	Alicia	Keys	F	1987-11-18	Anvers	72	18/11/1987
Éditer Copier Supprimer	12	Hakima	Darmouch	F	1982-01-01	Bruxelles	76	01/01/1982
Éditer Copier Supprimer	13	Kevin	Debruyne	M	1990-07-10	Anvers	79	10/07/1990
Éditer Copier Supprimer	14	Dark	Vador	M	2021-11-15	Tetouane	96	15/11/2021

On voit ici que j'ai choisi le format classique pour nous.

(Exos)

1) **Affichez le nom prénom dans une seule colonne et date de naiss. sous le format :**

a)
7th Jul 2007
24th Dec 2004

b)
07-07-2007
24-12-2004

c)
07 July 2007
24 December 2004

d)
07/07/2007

e)
Saturday 7 July 2007
Friday 24 December 2004

On a aussi des fonctions plus simple qui affiche juste l'année (**YEAR()**), le mois (**MONTH()**), les jours (**DAY()**)…

```
SELECT *, YEAR(date_of_birth) as année_de_naissance FROM `users`
```

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

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	année_de_naissance
Éditer Copier Supprimer	1	James	Bond	M	2007-07-07	London	70	2007
Éditer Copier Supprimer	2	Jack	Bauer	M	2004-12-24	New-York	124	2004
Éditer Copier Supprimer	3	Lara	Croft	F	1997-08-01	Washington	63	1997
Éditer Copier Supprimer	8	Cool	Blues	X	2000-02-24	Anvers	100	2000
Éditer Copier Supprimer	9	Beyonce	Knowles	F	1980-11-20	Kinshasa	70	1980

On peut aussi mettre cette partie dans une condition pour tester.

- **SELECT * FROM users WHERE MONTH(date_of_birth) = 1**

Ici ça nous donnera tous les utilisateurs nés en janvier.

On peut aussi l'utiliser dans un **GROUP BY**.

(Exos)

1) **Affichez-moi toutes les personnes nées entre 1990 et 2000 inclus.**
2) **Affichez les personnes qui sont né entre février et Septembre.**
3) **Affichez les femmes ou autres (X) qui sont né entre le 12ème et le 26ème jour.**
4) **Affichez les hommes ou femmes qui sont né en Janvier ou Mars ou Mai ou Juillet ou Octobre ou Décembre.**
5) **Affichez tous les utilisateurs qui sont né un jour pair. (Par exemple le 2 novembre, le 8 janvier, le 28 décembre etc.).**
6) **(Difficile) Essayez de m'afficher le nombre de fois qu'un mois apparait dans les dates de naissances, regroupé par les mois.**

1	2
2	7

Exemple : Un qui est né en février et 2 qui sont né en Juillet etc.

7) **(Difficile) Affichez moi le nombre de fois qu'une année apparait, et regroupé les par année.**

1	1970
2	1980
1	1982