

## 1. Les types de données

MySQL dispose de 4 types de données permettant de conserver une date, un temps ou les deux.

<b>DATE</b>	Stocke une date au format 'AAAA-MM-JJ' allant de '1000-01-01' à '9999-12-31' Pas de valeur par défaut possible
<b>DATETIME</b>	Stocke une date et une heure au format 'AAAA-MM-JJ HH:MM:SS' allant de '1000-01-01 00:00:00' à '9999-12-31 23:59:59' Valeur par défaut possible : <b>Now()</b> ou CURRENT_TIMESTAMP Attention : Non fonctionnel sur les anciennes versions de MySQL
<b>TIMESTAMP</b>	Stocke une date sous forme numérique allant de '1970-01-01 00:00:00' à l'année 2037. Valeur par défaut possible : CURRENT_TIMESTAMP
<b>TIME</b>	Stocke l'heure au format 'HH:MM:SS', allant de '-838:59:59' à '838:59:59'

MySQL ne propose qu'un seul format de saisie : aaaa-mm-jj hh:mm:ss.

Sur les dernières versions de MySQL, une date invalide est refusée (sauf modification de la variable sql\_mode), sur les anciennes versions, la colonne reçoit la valeur '0000-00-00', c'est le cas aussi si la date ne peut être 'null'

Les opérateurs de comparaisons s'appliquent.

L'opérateur Like s'applique parfaitement sur le type Date



Sélection sur l'année : uneDate Like '2015%'

Sélection sur le mois : uneDate Like '%-01-%'

Sélection sur le jour : uneDate Like '%01'

De même pour le Type DateTime

Sélection sur l'année : uneDateHeure Like '2015%'

Sélection sur le mois : uneDateHeure Like '%-01-%'

Sélection sur le jour : uneDateHeure Like '%-%-01 %'

Sélection sur l'heure : uneDateHeure Like '%15:00:00'

Sélection sur les minutes : uneDateHeure Like '%:15:00'

Sélection sur les secondes : uneDateHeure Like '%01'

Il n'est cependant pas recommandé de l'utiliser.

MySQL ne sait pas additionner une date avec un entier pour obtenir une nouvelle date, il considère que la date devient un entier et additionne les deux pour donner un entier.

Exemple : Si dateFacture = '2017-11-15'

Select dateFacture + 25 From Facture affiche 20171140

C'est la même chose pour la différence entre deux dates et pour le type DateTime

Il faut donc obligatoirement passer par les fonctions AddDate() et SubDate().

Cependant MySQL offre cependant la possibilité d'utiliser l'addition et la soustraction en utilisant la clause **interval nb unit** qui permet d'ajouter ou de soustraire nb unité de temps (jour, mois, ...) à une date.



MySQL propose d'autres types de données

Documentation : <http://dev.mysql.com/doc/refman/5.7/en/data-types.html>

## 2. Les fonctions standards

Year(uneDate)	Retourne l'année
Month(uneDate)	Retourne le mois
Day(uneDate)	Retourne le jour
Hour(uneDateHeure)	Retourne l'heure
Minute(uneDateHeure)	Retourne les minutes
Second(uneDateHeure)	Retourne les secondes

## 3. Les principales fonctions spécifiques

Now() ou Current_Timestamp()	Retourne <b>la date et l'heure</b> courante
SysDate()	Retourne <b>la date et l'heure</b> courante
CurDate() ou Current_Date()	Retourne la date courante
CurTime() ou Current_Time()	Retourne l'heure courante
Extract(interval from uneDate)	Retourne une partie de la date Interval : day, week, month, quarter, year, year_month ...
Date_Add(date, interval n unit )	Ajoute un intervalle de temps à une date
Date_Sub(date, interval n unit )	Soustrait un intervalle de temps à une date
AddDate(date, nbJour)	Ajoute des jours à une date.
SubDate(date, nbJour)	Soustrait des jours à une date.
DateDiff(fin, debut)	Retourne la différence en jours entre deux dates
TimestampDiff(unit, debut, fin)	Retourne la différence en unité entre deux dates (en tenant compte des unités inférieures)
To_Days(date)	Retourne le nombre de jours depuis la date 0 Intéressant pour la comparaison date et datetime
DayOfWeek(date)	Numéro du jour de la semaine (1 dimanche, 7 samedi)
WeekDay(date)	Numéro du jour dans la semaine (0 lundi 6 dimanche)
WeekOfYear(date)	Numéro de la semaine dans la codification Française Equivalent : WEEK(uneDate, 3)
Date_Format(date, 'format')	Permet de renvoyer une date sous le format souhaité
LAST_DAY(date)	Retourne le dernier jour du mois d'une date

Exemple	Résultat
AddDate(CurDate(), 1)	Demain
Date_Add(CurDate(), interval 1 day )	
Date_Add(dateVente, INTERVAL 15 day) < now();	Date dépassée de plus de 15 jours
SubDate(CurDate(), 1)	Hier
Date_Sub(Now(), INTERVAL 1 day);	
DateDiff(now(), concat(year(curdate()), '-01-01'))	Nombre de jours écoulés depuis le premier janvier
TimestampDiff(month, concat(year(now()), '-01-01'), now())	Nombre de mois entiers écoulés depuis le premier janvier
DateDiff(now() , CONCAT(year(curdate()), '-01-01'))	Nombre de jours écoulés depuis le premier janvier
Extract(year_month From dateVente) = 202011	En novembre 2020
Date_Format(curdate(), '%d/%m/%Y')	Date du jour au format jj/mm/aaaa

Liste complète : <http://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html>

## 4. Format de saisie d'une colonne de type Date ou DateTime

La valeur d'une date doit être précisée sous la forme d'une chaîne de caractères

**Le seul format de saisie est 'aaaa-mm-jj' pour une date et 'aaaa-mm-jj hh:mm:ss.000' pour un champ de type datetime.**



Si la valeur n'est pas valide MySQL 5.7 retourne maintenant une erreur (Incorrect date value)  
La valeur '0000-00-00' n'est plus acceptée sauf modification des modes SQL strict et NO\_ZERO\_DATE  
[https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode\\_no\\_zero\\_date](https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode_no_zero_date).

Si la date est au format jj/mm/aaaa, il est possible de la convertir dans le format 'aaaa-mm-jj' à l'aide de la fonction :

```
STR_TO_DATE(date, format)
```

Exemple : Insert into message(date, contenu) values (str\_to\_date('10/03/2021', '%d/%m/%Y'), 'Bonjour')

## 5. Format d'affichage d'une date

Une date s'affiche sans transformation dans le format aaaa-mm-jj.

Si l'on souhaite afficher la date dans un autre format, il faut utiliser la fonction :

```
DATE_FORMAT(date, format)
```

Exemple : Select date\_format(date, '%d/%m/%Y') as dateMessage, contenu From message  
%H heure(0:23) %i minutes %s secondes : [https://sql.sh/fonctions/date\\_format](https://sql.sh/fonctions/date_format)

## 6. Comparaison et opération sur des valeurs de type DateTime et Date

Soit une table Operation contenant la colonne dateOperation de type dateTime.

Si la partie horaire est renseignée, et que l'on cherche à obtenir toutes les opérations d'une journée, il n'est pas possible de comparer dateOperation avec la date du jour (CurDate()).

Il faut donc trouver le moyen de ne prendre en compte que la partie date sans la partie horaire.

Pour cela, on peut utiliser la fonction Cast(valeur As Type) qui va permettre ici de convertir une valeur de type DateTime en Date. Cette fonction existe aussi sous SQL Server.

```
Select * From Operation  
Where Cast(dateOperation AS Date) = CurDate()
```

Il est aussi possible d'utiliser la fonction To\_Days qui convertit une date en un nombre entier

```
Select * From Operation  
Where To_Days(dateOperation) = To_Days(CurDate())
```

Pour une recherche dans un intervalle (dans les 15 jours) :

```
Select * From Operation  
Where dateOperation >= CurDate()  
And To_Day(dateOperation) <= To_Day(CurDate()) + 15;
```

## 7. Utilisation des opérateurs + et – associés au mot clé interval

MySQL offre une façon très originale pour réaliser des opérations sur les dates. Il est ainsi possible d'ajouter ou soustraire sur une date des intervalles de temps pour obtenir la date souhaitée :

Demain	CurDate() + interval 1 day
Hier	CurDate() - interval 1 day
Dans une semaine	CurDate() + interval 1 week
Dans un mois	CurDate() + interval 1 month
Dans un trimestre	CurDate() + interval 1 quarter
Dans un an	CurDate() + interval 1 year
Dans une seconde	Now() + interval 1 second
Dans 5 minutes	Now() + interval 5 minute
Dans une heure	Now() + interval 1 hour
Le lundi précédent	<b>current_date() - interval weekday(current_date) day</b> <i>ex : si on est un lundi weekday retourne 0 soit 0 jour plus tôt</i> <i>ex : si on est un mardi weekday retourne 1 soit 1 jour plus tôt</i>
Le dimanche suivant	curdate() - interval WeekDay(curdate()) day + interval 6 day <i>ex : si on est un lundi weekday retourne 0 (6 - 0) soit 6 jours plus tard</i> <i>ex : si on est un mardi weekday retourne 1 (6 - 1) soit 5 jours plus tard</i>
Le lundi précédent	current_date() - interval weekday(current_date) day - interval 7 day <i>ex : si on est un lundi weekday retourne 0 (7 - 0) soit 7 jours plus tard</i> <i>ex : si on est un mardi weekday retourne 1 (7 - 1) soit 6 jours plus tard</i>

## 8. Exemple

Soit la table Annonce (id, titre, dateAnnonce)

### Les annonces du mois de février

Select id, titre where month(dateAnnonce) = 2 and year(dateAnnonce) = 2018

Select id, titre where dateAnnonce beetwen '2021-02-01' and '2021-02-28'

### Les deux prochaines annonces

Select id, titre where dateAnnonce >= CurrentDate() order by dateAnnonce limit 2;

### Les deux dernières annonces

Select id, titre where dateAnnonce < CurrentDate() order by dateAnnonce desc limit 2;

### Les deux prochaines annonces dans les 15 jours à venir

Select id, titre where dateAnnonce >= CurrentDate()

And dateAnnonce < CurrentDate + interval 2 week

order by dateAnnonce limit 2;

### Les deux dernières annonces de moins de 15 jours

Select id, titre where dateAnnonce < CurrentDate()

And danteAnnonce >= CurrentDate – interval 2 week

order by dateAnnonce desc limit 2;