

# LES TYPES DE DONNEES DANS MySQL

## NOMBRES ENTIERS :

Type	Nombre d'octets	Minimum	Maximum
<b>TINYINT</b>	1	-128	127
<b>SMALLINT</b>	2	-32768	32767
<b>MEDIUMINT</b>	3	-8388608	8388607
<b>INT</b>	4	-2147483648	2147483647
<b>BIGINT</b>	8	-9223372036854775808	9223372036854775807

Si vous n'utilisez que des valeurs positives, vous pouvez préciser l'attribut UNSIGNED. Ainsi pour un TINYINT, la plage de valeurs va de 0 à 255

### **Zéro de Conduite (Zerofill) :**

- Si vous utilisez l'attribut ZEROFILL sur un INT, MySQL ajoutera automatiquement des zéros devant le nombre jusqu'à la taille définie pour la colonne. Par exemple, si vous définissez un champ comme INT(5) ZEROFILL et que vous insérez la valeur 23, MySQL stockera 00023. Les INT ZEROFILL sont toujours UNSIGNED.

# LES TYPES DE DONNEES DANS MySQL



## DECIMAL et NUMERIC :

- DECIMAL et NUMERIC sont des types de données qui stockent des valeurs numériques exactes. Ces types sont souvent utilisés pour stocker des valeurs monétaires ou d'autres quantités où la précision des calculs est importante.
- Ils peuvent être déclarés avec une précision et une échelle (par exemple, DECIMAL(5,2) permettrait de stocker jusqu'à 999.99).
- DECIMAL et NUMERIC sont interchangeables ; il n'y a aucune différence entre eux dans MySQL.

## FLOAT :

FLOAT peut s'utiliser sans paramètres, auquel cas quatre octets sont utilisés pour stocker les valeurs de la colonne. Il est cependant possible de spécifier une précision et une échelle, de la même manière que pour DECIMAL et NUMERIC.

## DOUBLE et REAL :

REAL et DOUBLE ne supportent pas de paramètres. REAL est un synonyme de DOUBLE dans MySQL

# LES TYPES DE DONNEES DANS MySQL



## Valeurs exactes vs valeurs approchées :

Les nombres stockés en tant que NUMERIC ou DECIMAL sont stockés sous forme de chaînes de caractères. Par conséquent, c'est la valeur exacte qui est stockée. Par contre, les types FLOAT, DOUBLE et REAL sont stockés sous forme de nombres, et c'est une valeur approchée qui est stockée.

En effet, FLOAT, DOUBLE et REAL utilisent le format de virgule flottante pour le stockage des nombres.

Les nombres à virgule flottante sont représentés en binaire, ce qui signifie que le nombre est divisé en une mantisse et un exposant. Cette représentation n'est pas capable d'exprimer exactement toutes les valeurs décimales, en particulier celles qui sont périodiques dans leur forme binaire.

Cela signifie que si vous stockez par exemple 56,6789 dans une colonne de type FLOAT, en réalité, MySQL stockera une valeur qui se rapproche de 56,6789 (par exemple, 56,678900000000000001). Cela peut poser problème pour des comparaisons notamment (56,678900000000000001 n'étant pas égal à 56,6789). S'il est nécessaire de conserver la précision exacte de vos données (l'exemple type est celui des données bancaires), il est donc conseillé d'utiliser un type numérique à valeur exacte (NUMERIC ou DECIMAL donc).

## CHAR :

- CHAR est un type de données à longueur fixe. Lorsque vous déclarez un champ CHAR, vous devez spécifier sa longueur, par exemple CHAR(5). Peu importe la longueur réelle de la chaîne que vous stockez, MySQL va utiliser l'espace pour le nombre de caractères spécifié.
- Si la chaîne est plus courte que la longueur déclarée, MySQL la complètera avec des espaces à droite lors de l'enregistrement et les supprimera lors de la récupération des données.
- CHAR est plus rapide lorsque toutes les valeurs sont de longueurs presque identiques parce que MySQL n'a pas besoin de chercher la fin de la valeur.

## VARCHAR :

- VARCHAR est un type de données à longueur variable. Vous définissez également une longueur maximale pour VARCHAR, par exemple VARCHAR(100), mais MySQL n'utilisera que l'espace nécessaire pour stocker les données réelles plus un ou deux octets supplémentaires pour enregistrer la longueur de la chaîne.
- Si la chaîne est plus courte que la longueur maximale, MySQL stockera uniquement la chaîne elle-même et sa longueur. Cela signifie que VARCHAR est plus économique en espace de stockage pour les données qui varient beaucoup en longueur.
- VARCHAR est généralement plus lent que CHAR pour la recherche et l'accès car MySQL doit d'abord déterminer la longueur réelle de la chaîne.
- VARCHAR est préférable pour stocker des chaînes de caractères de longueur variable, comme les noms, les adresses ou les descriptions.

# LES TYPES DE DONNEES DANS MySQL



## TEXT :

Pour stocker des textes de plus de 255 caractères, il suffit alors d'utiliser le type TEXT, ou un de ses dérivés TINYTEXT, MEDIUMTEXT ou LONGTEXT. La différence entre ceux-ci est la place qu'ils permettent d'occuper en mémoire.

Type	Longueur maximale	Mémoire occupée
TINYTEXT	$2^8$ octets	Longueur de la chaîne + 1 octet
TEXT	$2^{16}$ octets	Longueur de la chaîne + 2 octets
MEDIUMTEXT	$2^{24}$ octets	Longueur de la chaîne + 3 octets
LONGTEXT	$2^{32}$ octets	Longueur de la chaîne + 4 octets