

# Types de Données

# Les différents types de données

## C'est quoi une clé primaire ?

La clé primaire est une donnée particulière qui permet d'identifier de manière unique une entrée dans une table. Elle prend souvent la forme d'un numéro (numéro d'adhérent par exemple).

Cette clé est gérée par le SGBDR.

Clé primaire	Nom	Prenom	Ville
12345678	Bond	James	Londres
12345679	Bond	James	Paris
12345680	Bond	Jean	Londres

Cette clé est très importante dans la manipulation de données. Elle permet notamment:

- D'identifier sans le moindre doute un enregistrement, rapidité
- D'éviter les doublons
- De créer des relations entre des tables
- Pour éviter les risques d'erreur ce champ est incrémenté automatiquement par le SGBD

# Les différents types de données

## Type de données

Passons en revue les types de données pour une table:

### Nombres entiers

Voici un tableau représentant des types entiers, ainsi que l'intervalle dans lequel la valeur peut être comprise pour chaque type.

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

Si vous essayez de stocker une valeur en dehors de l'intervalle permis par le type de votre champ, MySQL stockera la valeur la plus proche. Par exemple, si vous essayez de stocker 12457 dans un TINYINT la valeur stockée sera 127.

# Les différents types de données

## L'attribut UNSIGNED

Vous pouvez également préciser que vos colonnes sont UNSIGNED , c'est-à-dire qu'on ne précise pas s'il s'agit d'une valeur positive ou négative (on aura donc toujours une valeur positive).

Dans ce cas, la longueur de l'intervalle reste la même, mais les valeurs possibles sont décalées, le minimum valant 0.

Pour les TINYINT , on pourra par exemple aller de 0 à 255.

# Les différents types de données

## Type de données

### Donnée DECIMAL

En MySQL pour stocker un nombre décimal on utilise la fonction DECIMAL().

Cette fonction prend en compte deux paramètres - le premier est le nombre total de caractères dans le nombre et le second est le nombre total de nombres après la virgule.

Exemple:

Pour stocker le nombre 14.98, on peut appeler la fonction **DECIMAL(4,2)**, cela implique 4 chiffres au total et 2 chiffres après la virgule.

Pour calculer en octet un nombre décimal, un format binaire est utilisé, et il contient 9 chiffres dans 4 octets, cela modifie la taille de stockage des valeurs DECIMAL.

Exemple pour une colonne DECIMAL(18,9) nous avons 9 chiffres de chaque côté de la virgule, ce qui fait que la partie entière et la partie décimale demandent 4 octets chacune.

Une colonne DECIMAL(20,10) dispose de 10 chiffres de chaque côté de la virgule. Cela fait 4 octets pour chaque groupe de 9 chiffres, et 1 octet pour le reste.

# Les différents types de données

## Type de données

### Donnée FLOAT

Les types de données flottantes ont l'inconvénient d'être moins précises (sur les nombres avec beaucoup de chiffres après la virgule) que les type de données décimales mais l'avantage d'avoir des performances plus rapides.

On va utiliser les types de données décimales par exemple pour des données bancaires où l'on priorise la précision à la vitesse, par contre on va choisir des données flottantes pour enregistrer des coordonnées graphiques par exemple.

Le type de données flottant est une simple précision, qui utilise le type de données à virgule flottante 32 bits.

Dans le type de données decimal, nous pouvons utiliser un type de données à virgule flottante de 128 bits.

Exemple 1 :

```
create table exemple_float_decimal (exemple_id int, exemple_float float(4,2), exemple_Decimal decimal (4,2));  
  
Insert into exemple_float_decimal(exemple_id, exemple_float, exemple_Decimal) values(1, 2.1, 2.1), (2, 2.1, 2.1), (3, 2.1, 2.1) ;
```

Dans l'exemple ci-dessus on créer une table avec des champs de types float et decimal, on utilise les points de précision et en insère les mêmes données cela nous donne le résultat ci-dessous:

exemple_id	exemple_float	exemple_Decimal
1	2.10	2.10
2	2.10	2.10
3	2.10	2.10

# Les différents types de données

## Type de données

### Donnée FLOAT

Exemple 2 :

```
create table exemple_float_decimal (exemple_id int, exemple_float float, exemple_Decimal decimal);

Insert into exemple_float_decimal(exemple_id, exemple_float, exemple_Decimal) values(1, 2.1, 2.1), (2, 2.1, 2.1), (3, 2.1, 2.1);
```

Dans ce deuxième exemple on ne donne pas la précision nous obtenons le résultat ci-dessous :

exemple_id	exemple_float	exemple_Decimal
1	2.1	2
2	2.1	2
3	2.1	2

Dans mon cas comme j'utilise rarement des données sensibles je priorise donc le type FLOAT.

# Les différents types de données

## Type de données

### Donnée VARCHAR

Nom	Type
id 🔑	int
reference	varchar(20)
marque	varchar(20)
quantite	int

Le type de donnée VARCHAR est de type « chaînes de caractères » et l'on précise le nombre de caractères maxi que l'on peut insérer dans le champ.

Dans l'exemple de la table article les champs reference et marque peuvent supporter 20 caractères chacun.

**IL EST IMPORTANT DE BIEN ESTIMER LE NOMBRE DE CARACTERES RESERVE :**

Si vous ne réservez pas assez d'espace, la donnée sera tronquée et si trop d'espace est réservé, alors gaspillage de mémoire.

# Les différents types de données

## Types de champs - Horodatage

Il existe 3 types pour des dates:

- Date: composée de année, mois et jour
- Time: composée d'heure, minute, seconde
- Datetime et Timestamp: composition de date et time, vous avez la date à la seconde près

Dans la grande majorité des cas, il est primordial de déclarer un champ date ou time en tant que tel ! Sinon attendez-vous à des problèmes... Imaginez la confusion de votre ordinateur en lui demandant de calculer une durée entre deux ‘mots’ !

Il existe des fonctions bien utiles pour faire la conversion le cas échéant

‘AAAA-MM-JJ’ format d’une date sous MySQL

## L'art subtil de manier des dates

En tant que Développeurs vous pouvez travailler avec des clients à travers le monde, vous pourrez rencontrer deux types de problèmes avec les dates:

- Les décalages horaires
- L'ordre des éléments d'une date (08/10/2022 = 8 octobre ou 10 août ?)

En général les champs dates sont enregistrés selon un référentiel pour éviter les problèmes de décalage horaire (le fameux UTC)

Pour le deuxième cas c'est plus délicat. Plusieurs options:

- Vous accordez avec vos interlocuteurs sur le format ddmm/yyyy ou mmddyyyy
- Imposer un format “décroissant” souvent utilisé en informatique yyyyymmdd. Ce cas de figure a pour avantage si vous datez vos documents dans leur nom de les classer automatiquement dans l'ordre chronologique ! (date plus tardive = nombre plus élevé)

# Les différents types de données

## Types de champs - Booléens

C'est un type très pratique pour traduire un résultat binaire: 0/1, Vrai/Faux, On/Off

Réduire une question pour que la réponse devienne un booléen ouvre des possibilités de calculs et d'analyses automatisées

A noter qu'un booléen n'est pas forcément aussi binaire qu'on pourrait le croire... Imaginons que vous vouliez compter le nombre de "Vrai" de ce tableau.

V	V	F		V		F	F	V
---	---	---	--	---	--	---	---	---

Vous êtes probablement tenté de vouloir additionner les Vrai = 1. Mais comment traduire le tableau en binaire ?

1	1	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---

Et si vous voulez compter les Faux ? Ou les abstentions ?

1	1	0	?	1	?	0	0	1
---	---	---	---	---	---	---	---	---

C'est pour ça que très souvent en booléen, vous aurez un statut "null" (vide) différent de 0, soit 3 valeurs possibles en réalité !

1	1	0	Null	1	Null	0	0	1
---	---	---	------	---	------	---	---	---