

Correction des exercices Module 3 - Premiers pas en SQL

Exercice 1.2

Correction

1. Quelle est la différence entre **INT** et **INTEGER**, entre **INTEGER** et **TINYINT** ?

INT est un synonyme de **INTEGER** (version raccourcie). Il est stocké sur 4 bytes (4 octets), on peut donc stocker des nombres signés allant de -2147483648 à 2147483647. Si non signé (seulement positif), on peut aller jusqu'à 4294967295 !

TINYINT permet de stocker un entier sur un byte (1 octet). On peut donc stocker des nombres signés entre -128 et 127, et de 0 à 255 en non signé. C'est le plus petit entier qu'on peut affecter dans MySQL.

Pour en savoir plus, [visiter ce lien](#)

2. Quelle est la différence entre les types **CHAR** et **VARCHAR** ?

Le type **CHAR(N)** permet de stocker une chaîne de caractères de taille fixe **N** (**CHAR(3)** est un type qui permet de stocker des caractères de 3 octets). Si la valeur stockée est inférieure à 3 caractères, le reste de la mémoire est rempli d'espaces (*trailing spaces*). Il est performant (mémoire allouée fixe) mais contraignant (faire attention lors de la définition des données !).

Le type **VARCHAR(N)** permet de stocker une chaîne de caractères de taille maximale **N**. Si la valeur stockée est inférieure à **N** caractères, le reste de la mémoire est libéré. Il permet de stocker des chaînes de caractères (de petite taille) sans savoir à l'avance le nombre exact (seulement les bornes). Il est moins performant que **CHAR** (mémoire à allouer est dynamique, doit réserver un octet pour stocker la taille maximale stockable de chaîne) mais plus souple.

Pour en savoir plus, [visiter ce lien](#).

1. Quel type privilégier pour stocker un texte (ex: contenu d'un article) ?

Le type **TEXT** est à privilégier car il est fait pour cela, contrairement à **VARCHAR** qui est fait pour stocker des chaînes de caractères assez courtes mais dont on ignore la taille précise (on ne fixe que les bornes) à l'avance. Le type **TEXT** (comme **BLOB**) stocke potentiellement des grandes chaînes de caractères donc le moteur du SGBD fait des optimisations en interne sur leur traitement.

Voir les scripts `cre_parc.sql`, `desc_parc.sql`, `drop_parc.sql` et `ins_parc.sql` SQL pour le reste de la correction.