

# SQL :

## langage de définition des données

Les bases de données sont très utilisées dans les applications Web. La création, l'interrogation et la manipulation des données de la base sont réalisées en SQL. Dans cette série d'articles vous apprendrez le langage SQL ainsi que les bases nécessaires pour communiquer avec une base de données à partir d'un script PHP.

---

### Cet article explique :

- Définition d'une base de données.
- Présentation du langage SQL.
- Création de bases et de tables.
- Gestion des contraintes.

### Ce qu'il faut savoir :

- Aucun prérequis.
- 

De nombreuses applications Web (forums, galeries d'images, sites marchands, ...) reposent sur des bases de données. L'interaction avec la base de données, la création de la base et des tables, la manipulation des données, ainsi que le contrôle des droits d'accès aux données sont réalisés avec le langage SQL.

Dans cette série d'articles sur SQL, vous allez apprendre les notions qui vous permettront de créer une base de données et d'ajouter, consulter, modifier ou supprimer des informations dans cette base. Vous verrez comment communiquer avec elle via une console SQL, l'interface graphique PhpMyAdmin ou depuis un script PHP.

Pour appliquer ce que vous allez apprendre dans cet article, il est nécessaire d'installer un serveur de base de données MySQL, de préférence la version 5, un serveur web avec PHP version 5 et le client graphique phpMyAdmin. Les distributions XAMPP (Windows, Linux, Mac OS), WAMP (Windows), EasyPHP (Windows) ou MAMP (Mac OS) vous fourniront l'environnement de travail nécessaire pour communiquer avec MySQL et créer des applications Web.

L'article sera illustré par la création d'une base de données très simple concernant la gestion d'une bibliothèque privée.

### Base de données

Une base de données est un ensemble structuré d'informations centralisées ou réparties. Un ou plusieurs utilisateurs, ou des programmes, peuvent manipuler les

données de la base en fonction des droits qui leur ont été accordés.

### SGBD

L'interaction avec la base de données ainsi que la création de la base sont réalisées par un système de gestion de base de données (SGBD). Ce dernier est composé de logiciels permettant :

- la création de la base, le choix du type de données, la gestion des contraintes,
- l'ajout, la suppression, la modification ou la consultation des données,
- la gestion des mots de passe et des privilèges des utilisateurs sur les données,
- le partage des ressources en prenant en compte les accès concurrents,
- l'annulation de transactions,
- la restauration de la base en cas de destruction et la reprise automatique en cas d'incident.

Un des modèles les plus répandus de SGBD est le relationnel, introduit à la fin des années 1970 (Ingres, Oracle, ...). Dans ce modèle, les données sont représentées sous la forme de tables à deux dimensions. Par exemple, un auteur pourrait être stocké dans une ligne d'une table nommée auteur, dont les colonnes représenteraient les champs nom, prénom, date de naissance, etc... Cette table comporterait autant de lignes qu'il y a d'auteurs distincts.



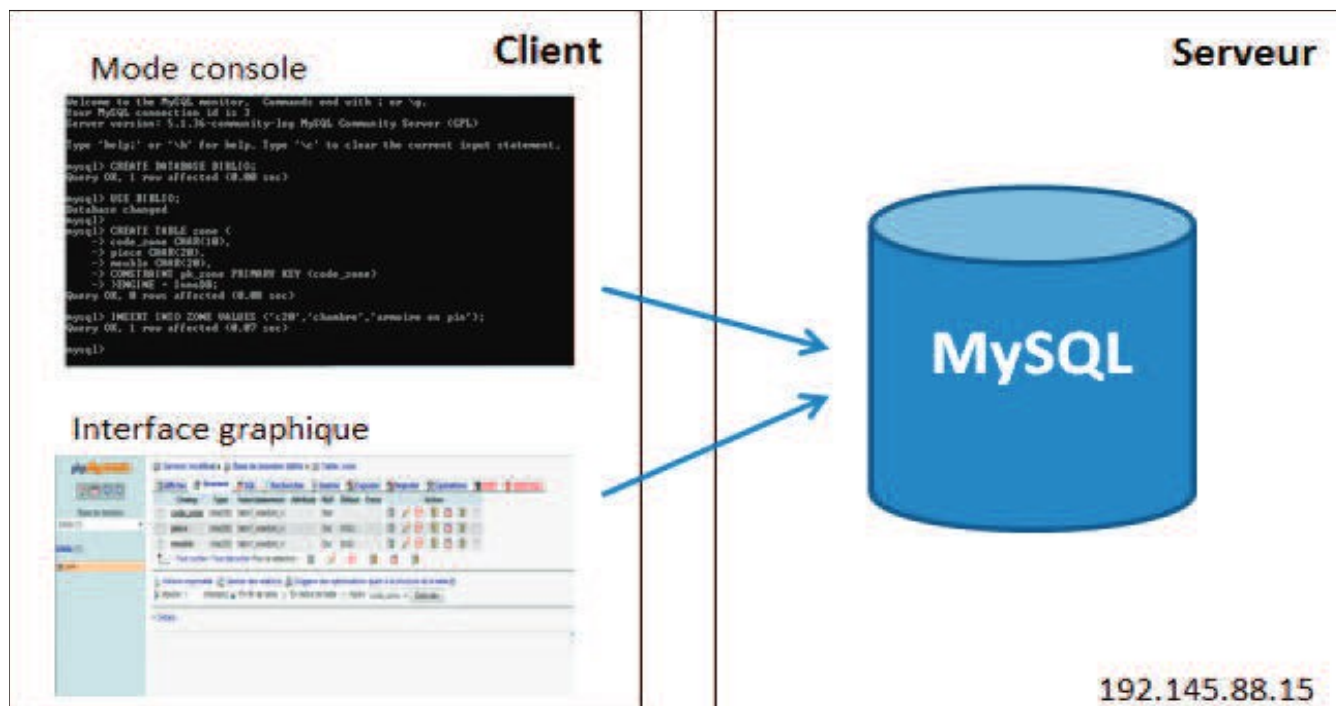


Figure 1. Client/Serveur

## SQL

SQL (*Structured Query Language*) est le langage utilisé dans les bases de données relationnelles pour la manipulation des données, leur définition et le contrôle de l'accès aux données. C'est un langage non procédural issu du langage SEQUEL (*Structured English Query Language*) développé par IBM. SQL a été standardisé en 1986 (ANSI SQL/86). Les versions SQL2 et SQL3 sont des révisions du standard SQL, datant respectivement de 1992 et de 1999. SQL2008 est la dernière version de la norme.

Les principaux SGBD relationnels du marché respectent le niveau *Entry Level* de la norme SQL2, c'est-à-dire qu'ils implémentent l'ensemble minimum d'instructions définies dans la norme. Chaque SGBD propose des fonctionnalités supplémentaires (fonctions, types de données, ...).

Le langage SQL comporte plusieurs parties :

- Le langage de définition des données (LDD) permet de définir la structure de la base : création, modification, suppression de tables ou de bases, ajout de contraintes,
- Le langage de contrôle des données (LCD) permet de définir les privilèges des utilisateurs sur les données,
- Le langage de manipulation des données (LMD) permet d'insérer des données dans la base, de les extraire, de les modifier ou de les supprimer.

Dans cet article, vous allez apprendre à utiliser le LDD et le LCD. Le LMD sera quant à lui présenté dans les prochains articles de cette série dédiée à SQL.

## MySQL

De nombreuses plateformes de développement d'applications Web reposent sur MySQL. Ce SGBD client/serveur open source est distribué sous licence GPL et sous licence commerciale. MySQL est simple d'utilisation et rapide. Il est disponible pour de nombreux systèmes d'exploitation. C'est pourquoi les exemples qui vont illustrer cet article seront tous testés sous MySQL.

La communication avec un serveur de bases de données MySQL peut s'effectuer depuis un client MySQL en mode console, en utilisant le langage de requêtes SQL, ou en utilisant l'interface graphique web phpMyAdmin (Figure 1). L'utilisateur peut ainsi, depuis son navigateur, envoyer des requêtes à MySQL sans avoir recours au langage SQL. Les requêtes sont automatiquement créées par phpMyAdmin, à partir des données saisies et des actions effectuées dans l'interface. Il sera cependant toujours possible d'envoyer des requêtes à la base en utilisant la console SQL intégrée de phpMyAdmin.

Vous apprendrez à utiliser ces clients dans la dernière partie de l'article.

## Création de la base et des utilisateurs

La commande `CREATE SCHEMA` permet de créer une base de données. La marque de fin d'instruction est le caractère point-virgule.

```
CREATE SCHEMA nom_base;
```

Dans l'exemple de cet article, la base contient les données d'une bibliothèque. La base dont le nom est `biblio` est créée par l'instruction suivante :

```
CREATE SCHEMA biblio;
```



## Pour les débutants

**Tableau 1.** Différents types de données proposés par MySQL

Classe	Type	Description
Numérique (entiers)	TINYINT	Permet de stocker 2 <sup>8</sup> nombres
	SMALLINT	Permet de stocker 2 <sup>16</sup> nombres
	MEDIUMINT	Permet de stocker 2 <sup>24</sup> nombres
	INTEGER	Permet de stocker 2 <sup>32</sup> nombres
Numérique (réels)	FLOAT	Nombre à virgule flottante stocké sur 4 octets
	DOUBLE, REAL	Nombre à virgule flottante stocké sur 8 octets
	DECIMAL(M,D)	DECIMAL(5,2) stocke un nombre à virgule fixe comportant 5 chiffres dont 2 décimaux
Chaîne de caractères	CHAR(n)	CHAR(8) stocke une chaîne de longueur fixe sur 8 caractères
	VARCHAR(n)	VARCHAR(25) stocke une chaîne de longueur variable et de maximum 25 caractères
	TINYTEXT	Permet de stocker 2 <sup>8</sup> caractères
	TEXT	Permet de stocker 2 <sup>16</sup> caractères
	MEDIUMTEXT	Permet de stocker 2 <sup>24</sup> caractères
	LONGTEXT	Permet de stocker 2 <sup>32</sup> caractères
Binaire	TINYBLOB	Permet de stocker 2 <sup>8</sup> caractères en objet binaire
	BLOB	Permet de stocker 2 <sup>16</sup> caractères en objet binaire
	MEDIUMBLOB	Permet de stocker 2 <sup>24</sup> caractères en objet binaire
	LONGBLOB	Permet de stocker 2 <sup>32</sup> caractères en objet binaire
Liste de chaînes	ENUM('val1', 'val2', ...)	Le champ prend 0 ou une valeur parmi la liste proposée
	SET('val1', 'val2', ...)	Le champ peut prendre 0, une ou plusieurs valeurs de la liste proposée
Date et temps	DATE	'2010-10-23'
	DATETIME	'2010-10-23 09:14:05'
	YEAR	2010
	TIME	'09:14:05'
	TIMESTAMP	1287818045

**Tableau 2.** Structure de la table ZONE

Colonne	Type de données	Description
code_zone	chaîne	Clé primaire, identifiant de la zone composé de la première lettre de la pièce suivi du numéro de l'étagère
piece	chaîne	Nom de la pièce dans laquelle se situe le livre
meuble	chaîne	Description du meuble dans lequel se situe le livre

Par convention les mots réservés du langage (`CREATE`, `SCHEMA`, ...) s'écrivent en majuscules et les mots définis par l'utilisateur en minuscules (`biblio`).

Une fois la base de données conçue, il faut créer un utilisateur et lui donner des droits afin qu'il puisse accéder à la base. Il est important de donner des droits différents aux utilisateurs de la base pour la protéger, assurer la confidentialité des données et leur intégrité.

Pour accorder des droits, il faut suivre la syntaxe :

```
GRANT privilege1 [, privilege2, ...] ON objet
TO nom_utilisateur1 [, nom_utilisateur2,
1;
```

La liste des privilèges peut comprendre `SELECT`, `DELETE`, `INSERT`, `UPDATE` ou tous à la fois (`ALL PRIVILEGES`). Les crochets sont utilisés dans la syntaxe pour indiquer les parties optionnelles de la commande. Les droits `SELECT`, `INSERT`, `UPDATE` et `DELETE` permettent respectivement de consulter, d'ajouter, de mettre à jour ou de supprimer des données. Ces privilèges peuvent être accordés en une seule commande à un ou plusieurs utilisateurs sur un objet (une ou plusieurs tables).

Dans MySQL, il est possible de créer un utilisateur et de lui donner des privilèges dans la même commande. Les instructions suivantes donnent à l'utilisateur lecteur le droit de lecture sur la table `livre` de la base



`biblio` et tous les droits au bibliothécaire sur toutes les tables (caractère `*`) de la base `biblio`.

```
GRANT SELECT ON biblio.livre TO lecteur IDENTIFIED BY 'motdepasse1';
GRANT ALL PRIVILEGES ON biblio.* TO bibliothecaire IDENTIFIED BY 'motdepasse2';
```

Il est possible de supprimer ultérieurement des privilèges avec la commande `REVOKE` :

```
REVOKE privilege1 [, privilege2, ...] ON objet
FROM nom_utilisateur;
```

## Type de données

Les valeurs des colonnes d'une table sont définies sur un domaine. La norme SQL définit quatre grandes classes de types de données : les numériques, les chaînes de caractères, les binaires et les dates. Les différents SGBD du marché proposent un sous-ensemble standard de types de données ainsi que des types spécifiques. La liste des types disponibles est définie dans la documentation de chaque SGBD.

### Numérique

Les types SQL2 `SMALLINT` et `INTEGER` stockent des nombres entiers, respectivement sur 2 ou 4 octets. Les types entiers disponibles dans le SGBD MySQL sont listés dans le Tableau 1. Le type peut être suivi par l'option `UNSIGNED`, celle-ci permet de préciser que les valeurs attendues doivent être positives (non signées). Le domaine des entiers positifs sera donc écrit dans MySQL :

```
INTEGER UNSIGNED
```

Suivant le type de données défini pour une colonne, les informations peuvent être modifiées lors d'une insertion ou d'une mise à jour. Par exemple, l'insertion d'un nombre négatif dans une colonne typée `UNSIGNED` peut entraîner le changement de la donnée en 0. Pour pallier cela, il faut paramétrer le mode SQL de MySQL en strict. Pour ce faire, il est nécessaire d'ajouter dans le fichier de configuration de MySQL (fichier `my.ini` sous Windows, `my.cnf` pour Linux et Mac OS X) la ligne suivante :

```
sql_mode = strict_all_tables
```

Ainsi, une mauvaise insertion sera rejetée et la donnée ne sera pas altérée.

Les types SQL2 `DECIMAL` et `FLOAT` stockent respectivement des nombres à virgule fixe (prix, notes, ...) et flottante (pourcentages, fractions, ...). La liste des types numériques disponibles dans le SGBD MySQL est donnée dans le Tableau 1. La définition des nombres à virgule fixe suit la syntaxe :

```
DECIMAL(M, D) [UNSIGNED]
```

`M` représente le nombre total de chiffres stockés dans la base, dont `D` décimales. C'est la somme du nombre `D` de chiffres derrière la virgule et du nombre de chiffres avant la virgule. Ainsi `DECIMAL(5,2)` permettra de représenter des nombres de -999.99 à 999.99 (cinq chiffres dont deux sont placés derrière la virgule).

### Chaînes de caractères

Les types `CHAR` et `VARCHAR` stockent respectivement des chaînes de caractères de taille fixe ou variable. La taille maximale est indiquée entre parenthèses et doit être comprise entre 1 et 255 caractères :

```
CHAR(25)
```

Pour des chaînes de plus grande taille, il faut utiliser les types textes MySQL présentés dans le Tableau 1.

Le type `CHAR` est bien adapté pour stocker des informations dont la taille est identique ou varie peu (numéro d'immatriculation, numéro ISBN de livres, ...). Lorsque les valeurs entrées dans la colonne n'atteignent pas la taille définie, la chaîne est complétée par des espaces. Toutes les valeurs de la colonne sont stockées sur le même nombre d'octets.

Le type `VARCHAR` est utilisé lorsque les données ont une longueur variable (titre de livre, nom d'auteur, ...). Le nombre d'octets utilisé pour stocker chaque champ varie en fonction du nombre de caractères de la chaîne, plus un octet pour mémoriser cette taille.

Le choix du type de données a donc un impact sur l'espace disque utilisé par la base de données.

Si l'insertion est supérieure à la taille du champ, le comportement par défaut de MySQL sera de tronquer du texte. Si MySQL est en mode strict, l'insertion sera refusée.

### Temps

Les types temporels de la norme SQL sont les suivants :

- `DATE` : une date comprenant le jour, le mois et l'année sur quatre chiffres,
- `TIME` : un horaire avec heure(s), minute(s) et seconde(s),
- `TIMESTAMP` : nombre de secondes depuis le 1er janvier 1970 à ce jour.

Le SGBD MySQL propose des types temporels supplémentaires comme indiqués dans le Tableau 1.

## Création de tables

La commande `CREATE TABLE` ajoute une table dans la base. Elle permet de nommer la table, de définir le nom et le type de ses colonnes et de fixer les contraintes d'intégrité. La syntaxe de la commande est la suivante :





## Listing 1. création base de données

```
-- creer la base
CREATE SCHEMA biblio;

-- creer les utilisateurs
GRANT SELECT ON biblio.livre TO lecteur IDENTIFIED BY
'motdepasse1';
GRANT ALL PRIVILEGES ON biblio.* TO bibliothecaire
IDENTIFIED BY 'motdepasse2';

-- selectionner la base de travail
USE biblio;

-- creer les tables
CREATE TABLE zone (
code_zone CHAR(10),
piece VARCHAR(20),
meuble VARCHAR(20),
CONSTRAINT pk_zone PRIMARY KEY (code_zone)
)ENGINE = InnoDB;

CREATE TABLE livre (
isbn CHAR(20),
titre VARCHAR(30) NOT NULL,
genre ENUM('roman','policier','theatre','historique','fant
astique'),
date_parution YEAR,
nb_pages INTEGER UNSIGNED,
code_zone CHAR(10),
CONSTRAINT pk_livre PRIMARY KEY (isbn),
CONSTRAINT fk_zone FOREIGN KEY (code_zone) REFERENCES
zone(code_zone) ON DELETE SET NULL ON UPDATE CASCADE
)ENGINE = InnoDB;

-- ajouter une colonne
ALTER TABLE livre ADD (langue ENUM('francais','anglais','
allemand','espagnol','chinois') DEFAULT 'francais');

-- afficher la liste des tables de la base
SHOW TABLES;

-- afficher la structure des tables
DESCRIBE zone;
DESCRIBE livre;
```

```
CREATE TABLE nom_table (
nom_colonne1 type [option],
nom_colonne2 type [option],
...
nom_colonneN type [option],
CONSTRAINT nom_constraint1 type _
constraint1,
CONSTRAINT nom_constraint2 type _
constraint2,
...
CONSTRAINT nom_constraintN type _
constraintN
);
```

Ces instructions vont être décrites dans les prochaines sections.

Pour créer une colonne, il faut suivre la syntaxe :

```
nom_colonne type [option]
```

Le type de la colonne est une contrainte d'intégrité de domaine. Il indique le domaine des valeurs de la colonne (cf. section *Types de données*). Par exemple, les colonnes `isbn` et `nb_pages` de la table `livre` (Tableau 3), sont créés par l'instruction :

```
isbn CHAR(20),
nb_pages INTEGER UNSIGNED
```

Le type peut être suivi d'une ou plusieurs options :

- NOT NULL : le champ doit être obligatoirement rempli, il ne peut pas contenir la valeur NULL. Cette valeur représente l'absence d'information, elle ne doit pas être confondue avec la valeur numérique 0 ou une chaîne de caractères vide,
- UNIQUE : les valeurs de la colonne doivent être distinctes,
- DEFAULT valeur : valeur attribuée par défaut au champ concerné lorsque celui-ci n'est pas renseigné,
- AUTO\_INCREMENT : lorsqu'une ligne est ajoutée dans la table, une colonne de type AUTO\_INCREMENT prendra un entier unique auto incrémenté dans ce champ. Cette option ne fait pas partie de la norme SQL mais est disponible dans MySQL.

La colonne `titre` de la table `livre` doit forcément être renseignée et la colonne `langue` peut seulement prendre une des valeurs prédéfinies par le biais d'une liste (ENUM), par défaut le français (Listing 1). Ainsi pour les créer, l'instruction est de la forme :

```
titre VARCHAR(30) NOT NULL,
langue ENUM('francais','anglais','allemand','es
pagnol','chinois') DEFAULT 'francais'
```

Les caractères autorisés dans les noms des objets peuvent varier selon le SGBD. Pour être compatible, il est préférable de commencer par une lettre de l'alphabet et de n'utiliser ensuite que les lettres de l'alphabet en minuscule et majuscule, les chiffres et le caractère souligné. Bien que MySQL supporte l'utilisation de mots réservés du langage (create, insert, ...), il est toutefois déconseillé de les utiliser pour nommer vos tables ou vos colonnes.

Suivant le système d'exploitation, le nom des tables et des colonnes peut être sensible à la casse.

## Définition des contraintes

Les données de la base doivent rester conformes à ce qui est attendu. Afin de vérifier que les données restent cohérentes lors de la manipulation de la base, il est nécessaire de définir des contraintes d'intégrité. Celles-ci permettent de rejeter des actions qui ne respecteraient pas la cohérence de la base et peuvent être définies par la syntaxe :

```
CONSTRAINT nom_constraint type_constraint
```

Il est nécessaire de leur donner un nom (norme SQL2) et de spécifier de quel type de contrainte il s'agit (cle primaire, cle unique, cle étrangère, vérification). La plupart des SGBD acceptent que les contraintes ne soient pas nommées explicitement lors de la création, dans ce cas c'est le SGBD



**Tableau 3.** Structure de la table LIVRE

Colonne	Type de données	Description
isbn	chaîne	Clé primaire, identifiant du livre composé de son numéro ISBN
titre	chaîne	Titre du livre
genre	énumération	Genre
date_parution	année	Année de parution
nb_pages	entier positif	Nombre de pages
code_zone	chaîne	Clé étrangère référençant la colonne code_zone de la table zone

**Tableau 4.** Contenu de la table ZONE

code_zone	piece	meuble
c10	Chambre	Armoire en pin
c20	Chambre	Armoire en pin
s8	Salon	Bibliotheque noire

**Tableau 5.** Contenu de la table LIVRE

isbn	titre	genre	date_parution	langue	nb_pages	code_zone
128-5-56985-5	Fall of giants	historique	2010	anglais	255	c20
598-5-55596-2	Notre Dame de Paris	roman	1831	français	123	s8
102-2-35419-5	Les fourmis	roman	1991	français	300	c10
523-5-65472-9	David Copperfield	roman	1850	anglais	458	s8
320-2-02365-5	Le bourgeois gentilhomme	theatre	1670	français	152	NULL
152-5-55695-2	Le seigneur des anneaux	fantastique	1954	français	832	c10

qui attribue un nom par défaut. Nommer les contraintes permet d'une part de les modifier plus facilement et, d'autre part, d'obtenir des messages d'erreur plus clairs que ceux qui font référence à une contrainte dont le nom a été attribué automatiquement par le SGBD. Le nom donné à la contrainte doit être unique pour toute la base. Il devrait être composé du nom de la table préfixé par une abréviation représentant le type de contrainte. Par exemple, la clé primaire dans la table `auteur` sera nommée `pk_auteur` (`pk` = *primary key*) et une clé étrangère dans la table `livre` pourra être nommée `fk_zone` (`fk` = *foreign key*).

### Clé primaire

Une clé primaire est un sous-ensemble minimal de colonnes d'une table dont les valeurs identifient de manière unique une ligne de la table. Toute table d'une base de données requiert une clé primaire qui doit être renseignée pour chaque enregistrement. Par exemple dans la table `livre`, la clé primaire est le numéro ISBN car il représente un seul et unique livre (Tableaux 3 et 5).

Une contrainte de clé primaire peut porter sur une ou plusieurs colonnes et s'écrit de la manière suivante :

```
PRIMARY KEY (col1 [, col2, ...])
```

L'instruction qui suit indique que la clé primaire de la table `livre` est la colonne `isbn` (Tableau 3) :

```
CONSTRAINT pk_livre PRIMARY KEY (isbn)
```

### Clé unique

Une clé unique est un sous-ensemble de colonnes d'une table pour lequel aucun doublon n'est autorisé.

Une contrainte de type clé unique s'écrit :

```
UNIQUE KEY (col1 [, col2, ...])
```

Une clé primaire est unique et non nulle, c'est-à-dire que la ou les colonnes de la clé ne peuvent pas contenir la valeur `NULL` et qu'une même valeur ne peut pas apparaître plus d'une fois.

### Clé étrangère

Une clé étrangère est un sous-ensemble de colonnes d'une table qui fait référence à une clé primaire ou unique d'une autre table. Les colonnes qui constituent la clé étrangère doivent avoir le même type que les colonnes référencées. Par exemple, dans la table `livre`, la colonne `code_zone` est clé étrangère et référence la clé primaire `code_zone` de la table `zone` (Tableau 3). Un livre ne peut pas référencer une zone qui ne serait pas enregistrée dans la base. Il est cependant possible d'enregistrer un livre qui ne serait pas encore rangé à sa place en insérant la valeur `NULL` dans le champ `code_zone` (Tableau 5).

L'utilisation d'une clé étrangère empêche le classement de certains livres dans des zones qui n'existeraient pas, ce qui garantit la cohérence de la base.

Une contrainte de type clé étrangère doit être définie ainsi :



```
FOREIGN KEY (col1 [, col2, ...]) REFERENCES nom_
table (col1 [, col2, ...]) [clause action]
```

A la suite de `FOREIGN KEY`, il faut lister la ou les colonnes de la table qui sont clés étrangères. Ces colonnes référencent les colonnes d'une autre table dont le nom est indiqué à la suite de `REFERENCES`. Généralement, les colonnes clés étrangères portent le même nom que les colonnes qu'elles référencent.

La création d'une table comportant des clés étrangères doit être faite après la création des tables référencées sinon une erreur sera générée.

Dans la table `livre`, la colonne `code_zone` est une clé étrangère qui référence la clé primaire `code_zone` de la table `zone`. L'instruction suivante permet de le spécifier :

```
CONSTRAINT fk_zone FOREIGN KEY (code_zone) RE-
FERENCES zone (code_zone)
```

Par défaut, lorsqu'un champ référence de la table `zone` est modifié ou supprimé, une erreur est générée et empêche la modification ou la suppression. Ce comportement peut-être redéfini. Il est possible de spécifier une clause de la forme `ON UPDATE` ou `ON DELETE`. Derrière ces clauses, il faut indiquer l'action à entreprendre en cas de mise à jour ou de suppression :

- **CASCADE** : toute modification ou suppression d'un champ référence entraîne la modification de la valeur ou la suppression de la ligne dans la table comportant la clé étrangère
- **SET NULL** : toute modification ou suppression d'un champ référence entraîne l'attribution de la valeur `NULL` au champ correspondant dans la table comportant la clé étrangère

Par exemple, en cas de suppression de la zone `c10` de la table `zone` (tableau 4), il ne faudrait pas supprimer les livres qui référencent cette zone. Une telle suppression doit entraîner l'attribution de la valeur `NULL` aux champs `code_zone` dont la valeur est `c10` dans la table `livre`. Ceci est réalisé grâce à la clause `ON DELETE SET NULL`. En cas de mise à jour, l'instruction ne devra pas être bloquée : les modifications seront reportées dans les champs qui référencent les valeurs modifiées grâce à la clause `ON DELETE CASCADE`. Par exemple, si `s8` est modifié en `s1` dans `zone`, la modification sera repercutee automatiquement dans `livre`.

```
CONSTRAINT fk_zone FOREIGN KEY (code_zone)
REFERENCES zone (code_zone) ON DELETE SET
NULL ON UPDATE CASCADE
```

Afin que MySQL permette la déclaration de clés étrangères et gère les transactions, il faut indiquer que les

tables sont de type InnoDB. Autant la table référencée que la table déclarant la contrainte de type clé étrangère. Pour indiquer qu'une table est de type InnoDB, dans la syntaxe de création de la table, il faut ajouter l'instruction `ENGINE = InnoDB` avant le ';' (Listing 1).

### Contrainte sur le domaine

La contrainte d'intégrité `CHECK` permet de vérifier qu'une valeur insérée est conforme à une condition, par exemple qu'un nombre est compris entre deux bornes.

```
CONSTRAINT nom_contrainte CHECK (condition)
```

MySQL ne gère pas les contraintes de type `CHECK`. Il existe trois manières de contourner l'absence de cette fonctionnalité :

- l'option `UNSIGNED` interdit l'insertion de nombres négatifs (colonne `nb_pages` : Listing 1),
- le type `ENUM` permet de restreindre les valeurs autorisées à une liste (colonnes `langue` et `genre`),
- un *trigger* peut être déclenché pour vérifier une condition lors d'une insertion ou d'une mise à jour.

### Modification de la structure

Une fois les tables créées, il est possible d'y ajouter des colonnes, de modifier le type de ces dernières ou d'ajouter une contrainte d'intégrité. Il est également possible de supprimer des éléments de la base.

Les commandes `ALTER TABLE` et `DROP` permettent de modifier la structure de la base.

### Ajout de colonne

Pour ajouter une ou plusieurs colonnes, il faut utiliser la commande `ALTER TABLE` avec le mot-clé `ADD` suivi de la définition de chaque colonne.

```
ALTER TABLE nom_table ADD (
nom_col1 type1, nom_col2 type2, ...);
```

L'instruction suivante ajoute une colonne `langue` à la table `livre` :

```
ALTER TABLE livre ADD (langue ENUM('français',
'anglais', 'allemand', 'espagnol', 'chinois'));
```

### Modification du type d'une colonne

Pour modifier une ou plusieurs colonnes dans le SGBD MySQL, il faut utiliser la clause `MODIFY` suivie de la nouvelle définition de chaque colonne.

```
ALTER TABLE nom_table MODIFY (
nom_col1 nouveau_type1,
nom_col2 nouveau_type2,
...);
```



### Ajout de contrainte

La clause `ADD CONSTRAINT` permet d'ajouter une contrainte d'intégrité. La définition de la contrainte est la même que celle décrite dans la section intitulée *Définition des contraintes*.

```
ALTER TABLE nom_table
ADD CONSTRAINT nom_contrainte type_contrainte;
```

### Suppression de contrainte

De même qu'il est possible de modifier la structure de la base, des commandes permettent d'en supprimer des éléments.

La clause `DROP CONSTRAINT` de la commande `ALTER TABLE` permet de supprimer la contrainte d'intégrité dont le nom est indiqué dans l'instruction :

```
ALTER TABLE nom_table DROP CONSTRAINT nom_contrainte;
```

Dans MySQL, les commandes de suppression des clés primaires et étrangères sont de la forme :

```
ALTER TABLE nom_table DROP PRIMARY KEY;
ALTER TABLE nom_table DROP FOREIGN KEY nom_contrainte;
```

L'instruction ci-après supprime de la table `livre` la contrainte de type clé étrangère, dont le nom est `fk_zone`, qui référence la clé primaire de la table `zone` :

```
ALTER TABLE livre DROP FOREIGN KEY fk_zone;
```

### Suppression de table

La commande `DROP TABLE` permet de supprimer une table dans la base. Par exemple, l'instruction suivante permet la suppression de la table `livre` :

```
DROP TABLE livre;
```

Si on tente de supprimer une table dont la clé primaire est référencée en tant que clé étrangère dans une autre table, MySQL génère une erreur et la table n'est pas supprimée.

```
DROP TABLE zone;
>>> ERROR 1217 : Cannot delete or update a
parent row : a foreign key constraint fails
```

Si vous voulez tout de même supprimer la table `zone`, il faut commencer par supprimer la table `livre` ou la contrainte de type clé étrangère qui lie les deux tables.

### Suppression de base

La commande `DROP SCHEMA` supprime définitivement une base, toutes ses tables et leur contenu.

```
DROP SCHEMA nom_base;
```

## Clients MySQL

Pour communiquer avec la base de données, il est possible d'entrer les commandes en mode console ou grâce à une interface graphique. Les deux solutions sont équivalentes et vont être présentées ci-après. Il est nécessaire pour leur utilisation d'avoir démarré le serveur de base de données de la distribution WAMP, XAMPP, MAMP ou EasyPHP.

### Console

Pour vous connecter au serveur depuis la console, vous devez entrer l'instruction suivante :

```
mysql [-h nom_hote] [-u login] [-p]
```

Lorsque le nom de l'hôte n'est pas spécifié, la connexion est tentée en local sur l'ordinateur, c'est-à-dire que le serveur est considéré comme étant sur le même ordinateur que le client. L'option `-u` indique qu'il va être spécifié un nom d'utilisateur. Lors de votre première connexion en local, l'utilisateur attendu est `root` (administrateur de la base). L'option `-p` précise qu'il est nécessaire de renseigner un mot de passe. Par défaut, l'administrateur n'a pas besoin d'entrer un mot de passe lors de sa première connexion. Si une erreur d'authentification est générée, vérifiez qu'un mot de passe n'a pas été défini par défaut par la plateforme que vous utilisez.

Une fois authentifié, vous pouvez entrer les commandes SQL du Listing 1 dans la console.

L'exécution de toutes les commandes aura permis la création de la base `biblio`, des utilisateurs `lecteur` et `bibliothecaire` et des deux tables `livre` et `zone`.

La commande `QUIT` permet de quitter le client.

### PhpMyAdmin

L'interface Web phpMyAdmin permet de créer des bases, des tables et de manipuler les données sans avoir de connaissances SQL. Chaque action effectuée via cet outil envoie une ou plusieurs requêtes SQL au serveur MySQL. Là où les requêtes envoyées sont affichées en haut de la page dans l'interface graphique. L'interface propose également une console SQL qui permet de rentrer des commandes SQL directement.

Si vous utilisez XAMPP, WAMP, MAMP ou EasyPHP, vous avez sur la page d'accueil du serveur web un lien vers phpMyAdmin. Sinon entrez l'URL de votre application phpMyAdmin dans votre navigateur.

L'interface graphique de phpMyAdmin comporte deux cadres. Le cadre gauche permet de sélectionner une base de données parmi la liste des bases hébergées sur le serveur MySQL. Une fois qu'une base a été sélectionnée, la liste de ses tables est affichée sous le nom de la base.

Le cadre de droite permet d'agir sur les bases de données : création de bases et de tables, modification, suppression, gestion des privilèges, consultation et manipulation des données.





## Pour les débutants

phpMyAdmin

Base de données: biblio

Table: livre

Champ	Type	Taille/Valeurs <sup>1</sup>	Défaut <sup>2</sup>	Interclassement	Attributs	Null
isbn	CHAR	20	Aucun			<input type="checkbox"/>
titre	VARCHAR	30	Aucun			<input type="checkbox"/>
genre	ENUM	fantastique	Aucun			<input checked="" type="checkbox"/>
date_parution	YEAR		Aucun			<input checked="" type="checkbox"/>
nb_pages	INT		Aucun		UNSIGNED	<input checked="" type="checkbox"/>
code_zone	CHAR	10	Aucun			<input checked="" type="checkbox"/>

Commentaires sur la table:

Moteur de stockage: InnoDB

Interclassement:

Figure 2. PhpMyAdmin : déclaration des types

Severeur: localhost Base de données: biblio Table: livre

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Vider Supprimer

✓ La table 'biblio'.'livre' a été créée.

```

CREATE TABLE 'biblio'.'livre' (
  'isbn' CHAR(20) NOT NULL,
  'titre' VARCHAR(30) NOT NULL,
  'genre' ENUM('roman','policier','theatre','historique','fantastique') NULL,
  'date_parution' YEAR NULL,
  'nb_pages' INT UNSIGNED NULL,
  'code_zone' CHAR(10) NULL,
  ENGINE = INNODB
)

```

Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input checked="" type="checkbox"/> isbn	char(20)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> titre	varchar(30)	latin1_swedish_ci		Non	Aucun		
<input type="checkbox"/> genre	enum('roman','policier','theatre','historique','fantastique')	latin1_swedish_ci		Oui	NULL		
<input type="checkbox"/> date_parution	year(4)			Oui	NULL		
<input type="checkbox"/> nb_pages	int(10)		UNSIGNED	Oui	NULL		
<input type="checkbox"/> code_zone	char(10)			Oui	NULL		

Tout cocher / Tout décocher Pour la sélection

Version imprimable Gestion des relations

Ajouter 1 champ(s) En fin de table

Aucun index n'est défini

Announcement de la page http://localhost :  
 ? Vous voulez vraiment effectuer :  
 ALTER TABLE 'livre' ADD PRIMARY KEY('isbn')  
 OK Annuler

Figure 3. PhpMyAdmin : structure de la table

Severeur: localhost Base de données: biblio Table: livre

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

Relié à

FOREIGN KEY (INNODB)	
isbn	Aucun index n'est défini!
titre	Aucun index n'est défini!
genre	Aucun index n'est défini!
date_parution	Aucun index n'est défini!
nb_pages	Aucun index n'est défini!
code_zone	biblio.zone.code_zone ON DELETE SET NULL ON UPDATE CASCADE

Figure 4. PhpMyAdmin : déclaration d'une clé étrangère



## Sur Internet

- <http://www.mysql.fr> – MySQL,
- <http://www.wampserver.com> – Outil Wamp,
- <http://sourceforge.net/projects/xampp> – Outil Xamp,
- <http://www.mamp.info> – Outil Mamp,
- <http://www.easyphp.org> – Outil EasyPHP,
- <http://www.phpmyadmin.net> – phpMyAdmin.

La page d'accueil comporte un champ texte permettant d'indiquer le nom de la base à créer. L'envoi du formulaire gère et soumet automatiquement à MySQL la commande :

```
CREATE SCHEMA nom_base;
```

Il est maintenant possible de créer une table dans la base grâce au champ intitulé *Créer une nouvelle table*. Il faut indiquer le nombre de colonnes que celle-ci contiendra. Pour finir de créer la table, il faut indiquer le nom des champs et leur type ainsi que le moteur de la table (InnoDB : Figure 2). PhpMyAdmin gère la commande de création de table, l'envoie à MySQL et affiche sa structure (Figure 3). Les clés primaires et étrangères sont définies à ce niveau. Pour indiquer la clé primaire, il suffit de cliquer sur l'icône en forme de clef (Figure 3). Un clic sur le lien *Gestion des relations* permet d'enregistrer les clés étrangères dont les colonnes concernées doivent avoir été indexées au préalable par un clic sur l'icône en forme d'éclair (Figure 4).

L'onglet *Privileges* permet de gérer les utilisateurs et leurs droits. Une fois la base sélectionnée, vous pouvez cocher les privilèges accordés par rapport au LMD, au LDD et au LCD. Il est possible de définir des privilèges sur toute la base ou sur des tables.

## Conclusion

Vous avez appris dans cet article les bases des langages de définition et de contrôle de données : création de bases et de tables, définition de contraintes, gestion des utilisateurs. Dans le prochain numéro, vous apprendrez à insérer et manipuler des données.

## CILIA MAURO, MAGALI CONTENSIN

Cilia Mauro est gestionnaire de bases de données et développeur d'applications web au CNRS. Elle enseigne les bases de données et PHP à l'université.

Contact : [cilia.mro@gmail.com](mailto:cilia.mro@gmail.com)

Magali Contensin est chef de projet en développement d'applications au CNRS. Elle enseigne depuis plus de dix ans le développement d'applications web à l'université et est l'auteur de nombreux articles sur le développement web en PHP.

Contact : <http://magali.contensin.online.fr>

# Rejoignez le Club .PRO

Pour plus de renseignement : [editor@phpsolmag.org](mailto:editor@phpsolmag.org)



## Stonfield Inworld

Stonfield Inworld propose aux entreprises des solutions globale d'intégration d'Internet et des Univers Virtuels dans leur stratégie de développement. Au-delà de ses services, la société consacre 30% de ses ressources à des travaux de R&D sur le e-Commerce et le e-Learning dans les Mondes Virtuels



## COGNIX Systems

Conseil, conception et développement d'applications évoluées pour les systèmes d'informations Internet/intranet/extranet. Alliant les compétences d'une SSII et d'une Web Agency, Cognix Systems conçoit des applicatifs et portails web à l'ergonomie travaillée et des sites Internet à forte valeur ajoutée.  
<http://www.cognix-systems.com>



## WEB82

Création et hébergements de sites web pour particuliers, associations, entreprises, e-commerce. Développement entièrement aux normes W3C ([www.w3.org](http://www.w3.org)) de sites web de qualité, au graphisme soigné et employant les dernières technologies du web (PHP5, MySQL5, Ajax, XHTML, CSS2).  
<http://www.web82.net>



## Core-Techs

Expert des solutions de gestion et de communication d'entreprise en Open Source, Core-Techs conçoit, intègre, déploie et maintient des systèmes de Gestion de Contenu Web, de Gestion Documentaire, de Gestion de la Relation Client (CRM), d'e-commerce et de travail collaboratif.  
<http://www.core-techs.fr>



## POP FACTORY

PoP Factory, SSII spécialisée Web. Développement de solutions applicatives spécifiques ; offre de solutions packagées : catalogue numérique, e-commerce, livre/magazine numérique, envoi SMS. Nous accompagnons nos clients tout au long de leur projet : audit, conseil, développement, suivi et gestion.  
<http://www.popfactory.com> / [info@popfactory.fr](mailto:info@popfactory.fr)



## Blue Note Systems

Spécialistes en CRM Open Source, nous proposons une offre complète de prestations sur la solution SugarCRM. Notre valeur ajoutée réside dans une expertise réactive et une expérience des problématiques de la GRC. Nous vous aidons à tirer le meilleur parti de votre solution CRM.  
<http://www.bluenote-systems.com>



## Intelligence Power

Conseil, Expertises, Formations et Projets E-business centrés au tour du cœur de métier : la Business Intelligence. Intelligence Power vous propose des solutions innovantes pour aligner la technologie sur la stratégie de votre entreprise.  
<http://www.intelligencepower.com>



## Web Alliance

Vous souhaitez être en première page des moteurs de recherche ? Rejoignez-nous, 100% des clients Web Alliance sont en 1ère page de Google. Web Alliance, société de conseil spécialisée dans le référencement internet, vous propose son expertise (référencement, liens sponsorisés, web-marketing).  
[www.web-alliance.fr](http://www.web-alliance.fr)

