



# Bases de données : créer une base de données

Karën Fort

karen.fort@sorbonne-universite.fr



# Sources d'inspiration

- ▶ Home and Learn - Free PHP :  
<https://www.homeandlearn.co.uk/php/php12p3.html#>
- ▶ <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1959710-decouvrez-mysql>
- ▶ Cours de B. Habert (ENS Lyon)
- ▶ Cours de N. Chaignaud (Rouen)
- ▶ Cours de G. Lejeune (SU)

## Système de Gestion de Bases de Données (SGBD)

Rappel sur les niveaux d'abstraction

MySQL

Principaux types de données disponibles

Créer une BD

phpMyAdmin

Pour finir

# Système de Gestion de Bases de Données (SGBD)

Rappel sur les niveaux d'abstraction

MySQL

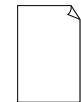
Principaux types de données disponibles

Créer une BD

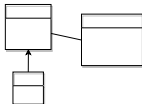
phpMyAdmin

Pour finir

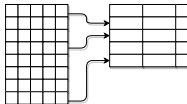
# Rappel sur les niveaux d'abstraction



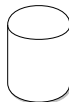
Clarification



Modélisation  
conceptuelle



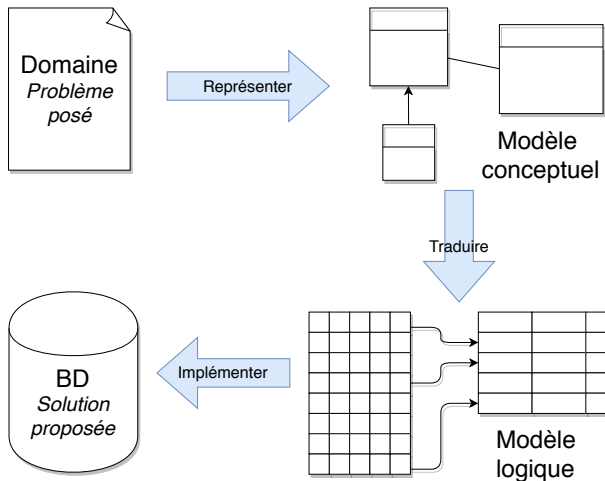
Modélisation  
logique



Implémentation



## Rappel sur les niveaux d'abstraction



# Système de Gestion de Bases de Données (SGBD)

Rappel sur les niveaux d'abstraction

MySQL

Principaux types de données disponibles

Créer une BD

phpMyAdmin

Pour finir

# Système de Gestion de Base de Données (SGBD)

## Définition

Un SGBD est un logiciel (ou un ensemble de logiciels) permettant de *manipuler* les données d'une base de données.

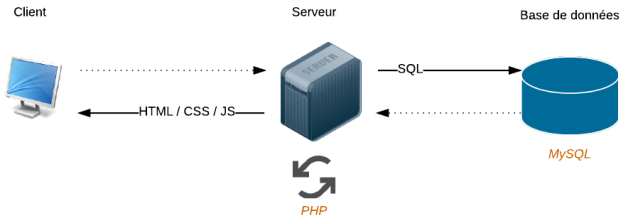
## Manipuler

Manipuler, c'est sélectionner et afficher des informations tirées de la base, modifier des données, en ajouter ou en supprimer (ce groupe de quatre opérations étant souvent appelé "CRUD", pour *Create, Read, Update, Delete*).

→ MySQL est un système de gestion de bases de données **relationnel** (SGBDR).



# Un modèle client-serveur



<https://openclassrooms.com/fr/courses/1946386-comprendre-le-web/2212835-les-bases-de-donnees>

- ▶ la BD est *hébergée* sur un serveur
- ▶ pour interagir avec la BD, il faut utiliser un logiciel "client" :
  - ▶ interroge le serveur
  - ▶ transmet ses réponses
- ▶ langage pour discuter avec le client : SQL



- ▶ un des SGBDR les plus utilisés
- ▶ logiciel *open source* (le code source est disponible)
- ▶ 1995 : première version officielle
- ▶ 2008 : société rachetée par Sun Microsystems, qui est elle-même rachetée par Oracle Corporation en 2010.

→ Documentation officielle :

<https://dev.mysql.com/doc/refman/8.0/en/>

# (Certains) autres SGBD

Oracle, Postgresql, SQLite

ORACLE®



## Système de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

- Types numériques

- Chaînes de caractères

- Autres types disponibles

Créer une BD

phpMyAdmin

Pour finir

# Choisir le type de données

pour chaque champ (anciennement, propriété)

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

Pour chaque champ (par ex, *ID\_Client*) :

- ▶ est-ce un champ numérique ? texte ? date ? autre ?
- ▶ de quelle taille de nombre, texte, ai-je besoin ?

# Bien choisir le type de données

Pour éviter :

- ▶ les gaspillages de mémoire
- ▶ les problèmes de performance : par ex. il est plus rapide de faire une recherche sur un nombre que sur une chaîne de caractères
- ▶ un comportement contraire à celui attendu : par ex. un tri sur des chaînes de caractères ne donne pas le même résultat que sur des entiers
- ▶ l'impossibilité d'utiliser des fonctionnalités propres à un type de données : par ex. pour les dates

## Système de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

- Types numériques

- Chaînes de caractères

- Autres types disponibles

Créer une BD

phpMyAdmin

Pour finir

# Les entiers (*integers* ou *int*)

## *signed / unsigned*

correspondent à des valeurs comprenant les négatifs (*signed*) ou non (*unsigned*).

- ▶ TINYINT : -128 à 127 (*signed*) ou 0 à 255 (*unsigned*)
- ▶ SMALLINT : -32768 à 32767 ou 0 à 65535
- ▶ MEDIUMINT : -8388608 à 8388607 ou 0 à 16777215
- ▶ INT : -2147483648 à 2147483647 ou 0 à 4294967295
- ▶ BIGINT : - $2^{63}$  à  $2^{63}$  ou 0 à 18446744073709551615



## Exemple

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

- ▶ ID\_Client : de quelle taille ai-je besoin ? Autrement dit : combien vais-je avoir de clients dans la base (en gros) ?
- ▶ Nom, Prénom, Adresse : probablement pas des types numériques
- ▶ Num\_Tel : est-ce un type numérique ?

## Exemple

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

- ▶ ID\_Client : de quelle taille ai-je besoin ? Autrement dit : combien vais-je avoir de clients dans la base (en gros) ? → **SMALLINT (probablement)**
- ▶ Nom, Prénom, Adresse : probablement pas des types numériques → **types textuels**
- ▶ Num\_Tel : est-ce un type numérique ? → **Non : on ne fait pas de calculs avec, donc c'est plutôt une chaîne de caractères**

# Attention !

Si vous essayez de stocker 2345 dans un TINYINT, la valeur stockée sera 127 (ou 255) !

# Les décimaux

- ▶ `DECIMAL` et `NUMERIC`, avec deux paramètres :
  - ▶ précision : nombre de chiffres significatifs stockés (dans 033, 0 n'est pas significatif)
  - ▶ échelle : nombre de chiffres après la virgule

## Exemple

`DECIMAL(5,3)` permet de stocker des nombres de 5 chiffres significatifs au maximum, dont 3 chiffres sont après la virgule. Par exemple : 12,354, -54,258, 89,2 ou -56.

- ▶ `FLOAT` : quatre octets sont utilisés pour stocker les valeurs de la colonne (on peut ajouter les paramètres précision et échelle)
- ▶ `REAL` et `DOUBLE` : pas de paramètres, stockage dans 8 octets (caractères) dans les deux cas pour MySQL

## Système de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

Types numériques

Chaînes de caractères

Autres types disponibles

Créer une BD

phpMyAdmin

Pour finir

## Les types textuels (*text*)

- ▶ TINYTEXT : 256 octets (caractères)
- ▶ TEXT : 64 kilo octets
- ▶ MEDIUMTEXT : 16 mega octets
- ▶ LONGTEXT : 4 giga octets

## CHAR vs VARCHAR

CHAR :

- ▶ maximum 255 caractères
- ▶ CHAR(10) : réserve 10 caractères et remplit avec des blancs : "Texte——"

VARCHAR :

- ▶ maximum 65 535 caractères
- ▶ pas de remplissage avec des blancs : VARCHAR(10) → "Texte"

**Attention**, si vous ne mettez pas de taille pour les CHAR et VARCHAR, vous risquez d'avoir une erreur (ou une taille de 1, selon les version de MySQL)

## Exemple

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

- ▶ ID\_Client : SMALLINT (probablement)
- ▶ Nom, Prénom, Adresse : de quelle taille de texte ai-je besoin ?
- ▶ Num\_Tel : de quelle taille de texte ai-je besoin ?



## Exemple

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

- ▶ ID\_Client : SMALLINT (probablement)
- ▶ Nom, Prénom, Adresse : de quelle taille de texte ai-je besoin ?  
→ VARCHAR(50) pour un nom, VARCHAR(20) pour un prénom, TINYTEXT pour une adresse ( ?)
- ▶ Num\_Tel : de quelle taille de texte ai-je besoin ? → VARCHAR(20) ( ?)

## Système de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

Types numériques

Chaînes de caractères

**Autres types disponibles**

Créer une BD

phpMyAdmin

Pour finir

# Chaînes binaires

pour stocker des données "brutes" (par ex. des images)

- ▶ `VARBINARY(x)` et `BINARY(x)` : fonctionnent comme `VARCHAR` et `CHAR` (mais pour des données binaires)
- ▶ `TINYBLOB`, `BLOB`, `MEDIUMBLOB` et `LONGBLOB` : même fonctionnement que `TINYTEXT`, `TEXT`, `MEDIUMTEXT` et `LONGTEXT`

## DATE, TIME, DATETIME

- ▶ DATE : AAAA-MM-JJ (2020-03-20, par exemple)
- ▶ DATETIME (date et heure) : AAAA-MM-JJ HH :MM :SS
- ▶ TIME (heure et intervalle de temps) : HH :MM :SS (peut être supérieur à 24h)

# ENUM et SET

n'existent que dans MySQL

- ▶ ENUM : liste de valeurs autorisées, de type "chaîne de caractères"
  - ▶ espece ENUM ('chat', 'chien', 'tortue') → on pourra stocker soit 'chat', soit 'chien', soit 'tortue'
- ▶ SET : idem, mais
  - ▶ espece SET ('chat', 'chien', 'tortue') → on pourra stocker toutes les combinaisons 'chat', 'chien', 'tortue', 'chat,tortue', 'chien,tortue', etc.

## Systeme de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

#### Créer une BD

- Conventions à respecter

- Autres champs à renseigner (autre les types)

### phpMyAdmin

### Pour finir

Systeme de Gestion de Bases de Données (SGBD)

Principaux types de données disponibles

Créer une BD

Conventions à respecter

Autres champs à renseigner (autre les types)

phpMyAdmin

Pour finir

# Erreurs à ne pas commettre

Noms de bases, tables, champs :

- ▶ **jamais** d'espaces ou d'accents
- ▶ **ne pas utiliser** des mots réservés par MySQL (par exemple, "date", "text", "type")



# Conseils

Nommage :

- ▶ mettez tous les noms de tables au singulier, ou au contraire au pluriel
- ▶ séparez-les toujours avec ' \_ ' (*Nom\_ Coureur*) ou bien toujours avec une majuscule (*NomCoureur*)

## Système de Gestion de Bases de Données (SGBD)

### Principaux types de données disponibles

### Créer une BD

Conventions à respecter

Autres champs à renseigner (autre les types)

phpMyAdmin

Pour finir

# NULL

Autorise à ne **pas** stocker de valeur.

Si NOT NULL, alors on ne peut pas laisser le champ vide.

## Exemple

Client (ID\_Client, Nom, Prénom, Adresse, Num\_Tel)

- ▶ Nom est sans doute obligatoire → NOT NULL
- ▶ Num\_Tel est facultatif → NULL

## Default

Définit une valeur par défaut pour le champ

## A\_I (AUTO\_INCREMENT)

surtout pour les clés

Les valeurs seront auto-incrémentées à l'ajout, par MySQL (qui prend la dernière valeur insérée dans le champ et l'incrémente de 1)

# PRIMARY

clé primaire

Définit une clé primaire, qu'il faut décrire comme `AUTO_INCREMENT` (sera incrémentée automatiquement) et *unsigned* (toujours positive)

# Remarque concernant les moteurs de tables

pour MySQL seulement

## MyISAM

moteur par défaut, commandes d'insertion et sélection de données plus rapides sur les tables utilisant ce moteur, mais **ne gère pas** les clés étrangères (entre autres).

## InnoDB

Plus lent et plus gourmand en ressources que MyISAM, mais gère les clés étrangères.

Système de Gestion de Bases de Données (SGBD)

Principaux types de données disponibles

Créer une BD

**phpMyAdmin**

Présentation

Créer une base de données via phpMyAdmin

Pour finir



Systeme de Gestion de Bases de Données (SGBD)

Principaux types de données disponibles

Créer une BD

phpMyAdmin

Présentation

Créer une base de données via phpMyAdmin

Pour finir



## phpMyAdmin

outil libre, écrit en PHP, permettant d'administrer des bases de données MySQL via le Web et une [interface graphique](#) (cliquette).

**Inconvénient** : il faut installer, en plus de MySQL, un serveur Web (Apache) et PHP (le trio s'installe ensemble et s'appelle [XAMPP](#)).

Il est évidemment possible de n'installer que MySQL, mais dans ce cas, pas d'interface graphique de manipulation.

# Accès

À distance, et pour vous éviter d'installer phpMyAdmin, je vous propose d'utiliser le serveur fourni :

<http://krik.paris-sorbonne.fr/phpmyadmin/index.php>  
(login : BDD\_L3, mot de passe : Bddl3\_ve)

Pour ceux-celles qui voudraient l'installer sur leur machine, la doc est ici : <https://docs.phpmyadmin.net/fr/latest/setup.html#installing-on-windows>

Système de Gestion de Bases de Données (SGBD)

Principaux types de données disponibles

Créer une BD

**phpMyAdmin**

Présentation

Créer une base de données via phpMyAdmin

Pour finir

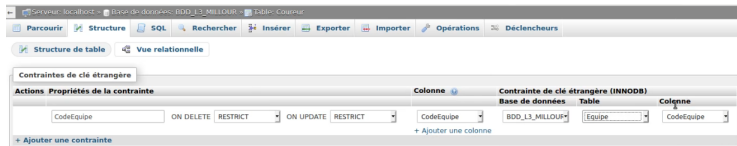
# Vidéo

Alice Millour a préparé une vidéo présentant les manipulations de base pour la création d'une base de données en ligne sur phpMyAdmin : voir le lien sur mon site.

# Attention (1) : création de clé étrangère

Créer un lien (contrainte) entre les deux tables

3 :35 de la vidéo (environ) :



Attention aux **types** (doivent être les mêmes) !

# Création de clé étrangère : résultat

On doit obtenir une clé grise :



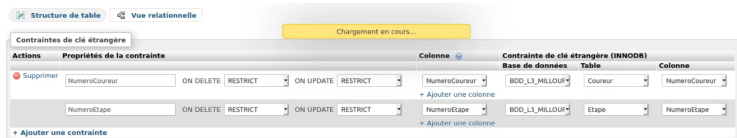
The screenshot shows a database management interface with a toolbar at the top containing buttons for 'Parcourir', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', 'Opérations', and 'Déclencheurs'. Below the toolbar, there are two tabs: 'Structure de table' (selected) and 'Vue relationnelle'. The main area displays a table structure for a table named 'Coureur'. The table has four columns: '#', 'Nom', 'Type', and 'Action'. The data rows are as follows:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	NumeroCoureur	int(100)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/> 2	NomCoureur	varchar(100) latin1_swedish_ci			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/> 3	CodeEquipe	int(11)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/> 4	CodePays	int(11)			Non	Aucun(e)			Modifier  Supprimer  Plus

# Attention (2) : création de clé double

Composition de 2 clés étrangères

6 :47 de la vidéo (environ) :



Puis :



Attention aux **types** (doivent être les mêmes) !



Systeme de Gestion de Bases de Données (SGBD)

Principaux types de données disponibles

Créer une BD

phpMyAdmin

Pour finir

CQFR : Ce Qu'il Faut Retenir

TD



- ▶ les types de données
- ▶ les conventions

## Créer une BD "Tour de France"

- ▶ créer sur PhpMyAdmin la base données correspondant au modèle logique proposé pour le tour de France (rappelé sur la diapo suivante)
- ▶ nommer cette base de données : BDD\_L3\_VotreNom
- ▶ sélectionner utf8\_general\_ci (encodage UTF8, insensible à la casse)
- ▶ peupler cette base avec des données réelles du Tour 2019 :
  - ▶ 3 coureurs, dont 2 de la même équipe (voir [https://fr.wikipedia.org/wiki/Liste\\_des\\_coureurs\\_du\\_Tour\\_de\\_France\\_2019](https://fr.wikipedia.org/wiki/Liste_des_coureurs_du_Tour_de_France_2019))
  - ▶ les 3 premières étapes du Tour 2019 (voir <https://www.letour.fr/fr/classements/etape-1>)

# MLD Tour de France

Coureur (NuméroCoureur, NomCoureur, #CodeEquipe,  
#CodePays)

Equipe (CodeEquipe, NomEquipe, DirecteurSportif)

Pays (CodePays, NomPays)

Etape (NuméroEtape, Date, VilleDépart, VilleArrivée, NbKm)

Participer (NuméroCoureurNuméroEtape, TempsRéalisé)