

MySQL

Table des matières

Vocabulaire.....	3
Une base de données et SGBD.....	3
Une table	4
Une colonne	4
Une contrainte	4
<i>La contrainte : NOT NULL</i>	<i>5</i>
<i>La contrainte : UNIQUE</i>	<i>5</i>
<i>La contrainte : PRIMARY KEY.....</i>	<i>5</i>
<i>La contrainte : CHECK</i>	<i>6</i>
<i>La contrainte : FOREIGN KEY</i>	<i>6</i>
Définition d'une table.....	7
Une table possède un NOM	7
Une table possède des COLONNES	7
Les colonnes ont un TYPE.....	7
Utiliser phpMyAdmin	9
Créer une base de données.....	9
Supprimer une base de données.....	9
Créer une table.....	9
Supprimer une table.....	9
Vider une table	9
Créer les différentes colonnes de la table.....	10
Modifier une table.....	10
Insérer manuellement les données.....	11
Voir le contenu d'une table.....	11
Exercice 1 :.....	12
<i>Consigne</i>	<i>12</i>
<i>Résultat.....</i>	<i>12</i>
Rechercher des données	13
<i>Dans phpMyAdmin</i>	<i>13</i>
<i>SELECT ... FROM.....</i>	<i>13</i>
<i>Sélectionner toutes les colonnes et toutes les lignes.....</i>	<i>13</i>
Exercice 2.....	14



<i>Sélectionner toutes les lignes de certaines colonnes uniquement</i>	14
<i>Exercice 3</i>	14
<i>Changer le nom de la colonne</i>	14
<i>Exercice 4</i>	14
<i>Opérations arithmétiques</i>	14
<i>Exercice 5</i>	15
<i>Ordonner les opérations</i>	15
<i>Concaténation</i>	15
<i>Exercice 6</i>	15
<i>Distinct</i>	16
<i>Exercice 7</i>	16
<i>Limitier et ordonner les résultats</i>	16
<i>WHERE : où la condition est vraie</i>	16
<i>Exercice 8</i>	17
<i>BETWEEN ... AND ... : entre ... et ...</i>	17
<i>Exercice 9</i>	17
<i>IN ... : égal à ...</i>	17
<i>Exercice 10</i>	17
<i>Exercice 11</i>	18
<i>LIKE ... : comme ...</i>	18
<i>Exercice 12</i>	18
<i>Exercice 13</i>	18
<i>NOT ... : pas</i>	18
<i>Exercice 14</i>	19
<i>IS (NOT) NULL : est (pas) null</i>	19
<i>AND : et</i>	19
<i>Exercice 15</i>	19
<i>OR : ou</i>	19
<i>Exercice 16</i>	19
<i>Ordonner les résultats</i>	20
<i>Exercice 17</i>	20
<i>Exercice 18</i>	20

Vocabulaire

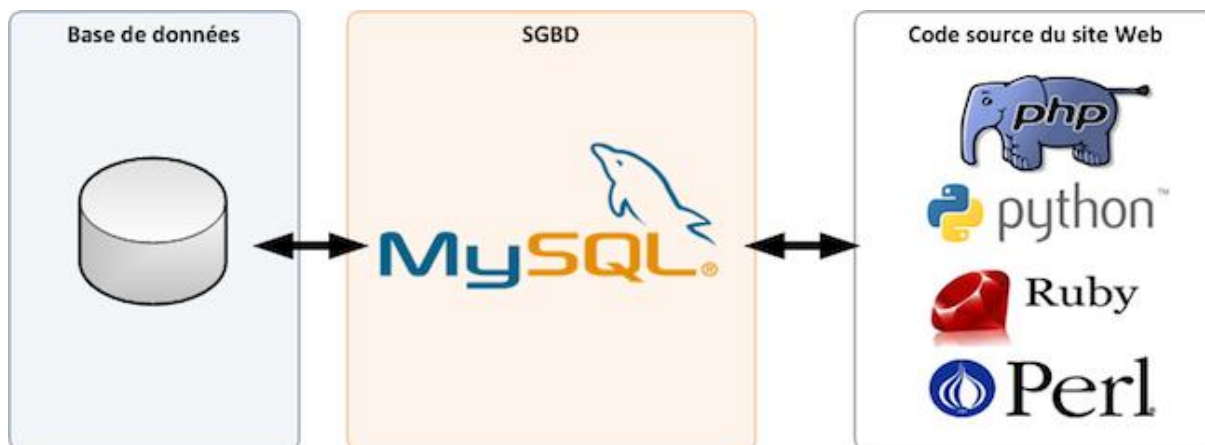
Une base de données et SGBD



Une **base de données** est une structure, le plus souvent informatique, permettant le stockage et l'exploration de données.

Passé	Présent
	
http://blogs.getty.edu/iris/what-is-the-future-of-the-photo-archive/	https://techcrunch.com/2016/02/19/optimizing-analytics-on-time-series-databases-a-supply-chain-perspective/

Un **Système de Gestion de Base de Données (SGBD)** est un programme informatique permettant de gérer des bases de données.

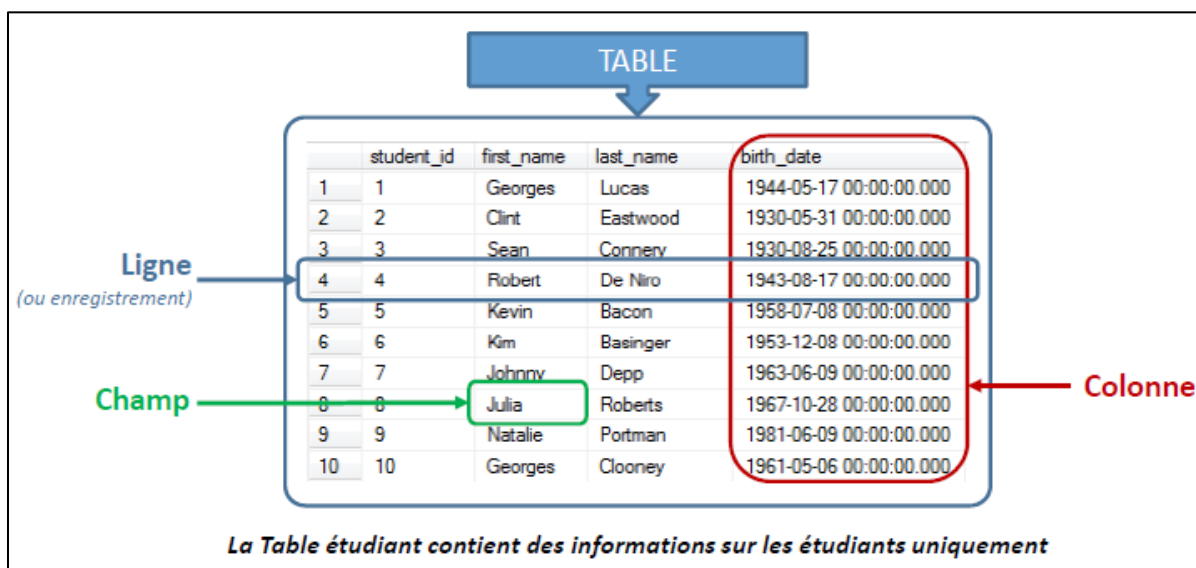


Source : <http://www.blog-nouvelles-technologies.fr/7605/mysql-pour-les-debutants/>

Les données de la base de données sont répertoriées dans des tables.

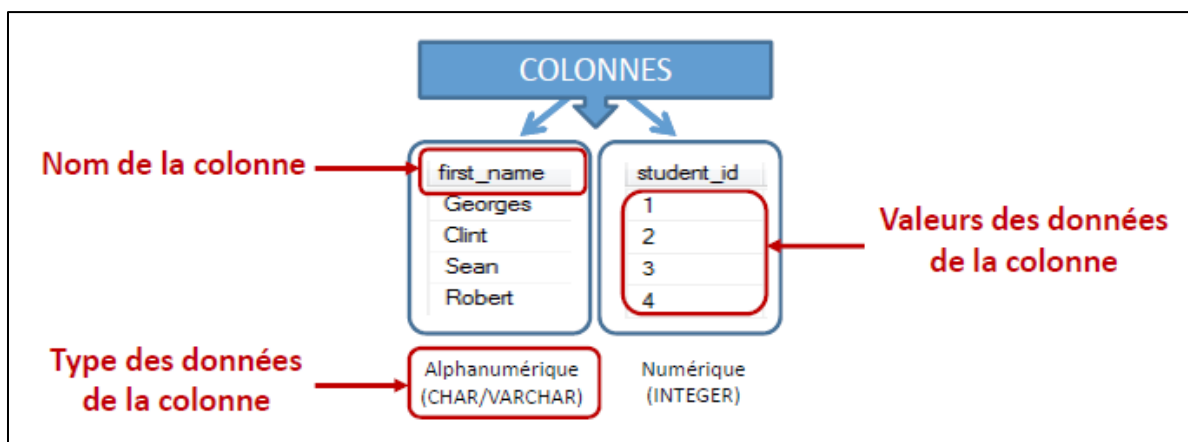
Une table

Une **table** regroupe des **ensembles de données** (d'informations) stockés de façon permanente et décrivant chacun un acteur de système réel modélisé (Entité).



Une colonne

Une **colonne** est un attribut d'une table, elle représente une caractéristique particulière de l'objet réel représenté.



Une contrainte

Une **contrainte** est un objet intégré à une table (comme les colonnes). Une contrainte possède un **nom**, un **type** et concerne au moins une colonne de la table.

	Contraintes	Description
Clé primaire	NOT NULL	Force la présence d'une valeur
	UNIQUE	Empêche les valeurs-doublons
	CHECK	Conditionne les valeurs (expression conditionnelle)
	FOREIGN KEY	Conditionne les valeurs par rapport à une autre table

La contrainte : NOT NULL

Ajouter la **contrainte « NOT NULL »** à la colonne d'une table obligera cette colonne à contenir **une valeur pour chacune des lignes** de la table.

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	3	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0

La contrainte : UNIQUE

La **contrainte d'unicité « unique »** a la particularité d'empêcher une colonne ou une combinaison de colonnes d'accepter deux valeurs identiques pour deux lignes différentes de la table.

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	nportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

La contrainte : PRIMARY KEY

La **contrainte de clé primaire** d'une table désigne un ensemble (souvent minimal) de colonnes qui **identifient** de manière **unique** les enregistrements d'une table. La combinaison des colonnes qui la composent doit être **UNIQUE** et **NON NULL**.

Il ne peut exister **qu'une et une seule clé primaire** par table MAIS celle-ci peut être composée d'une combinaison de plusieurs colonnes.

Une clé primaire est dite « **naturelle** » si la ou les colonnes qui la composent sont des attributs **représentant réellement une caractéristique** de l'objet que la table décrit.

Une clé primaire sera « **artificielle** » si elle est représentée par une colonne dont les valeurs ne **représentent rien dans le monde réel**. Ces colonnes artificielles sont **régulièrement utilisées** car elles sont faciles à manipuler et, comme elles ont le plus souvent comme valeurs des nombres, leur contenu peut être géré directement par le SGBD.

On dira alors, qu'en plus d'être une clé primaire artificielle, les valeurs de la colonne sont **auto-incrémentées**, le plus souvent de **1 à l'infini**.

Combinaison **non candidate**
(contient des valeurs **NULL**)

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	rportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

PRIMARY KEY
dite « artificielle »
(PEUT être auto-incrémentée)

Combinaison **UNIQUE ET NOT NULL**
candidate à la création d'une
PRIMARY KEY « naturelle »

La contrainte : CHECK

La **contrainte CHECK** d'une table permet de poser une **condition spécifique** sur les colonnes de la table, afin d'y empêcher l'insertion de n'importe quelle valeur.

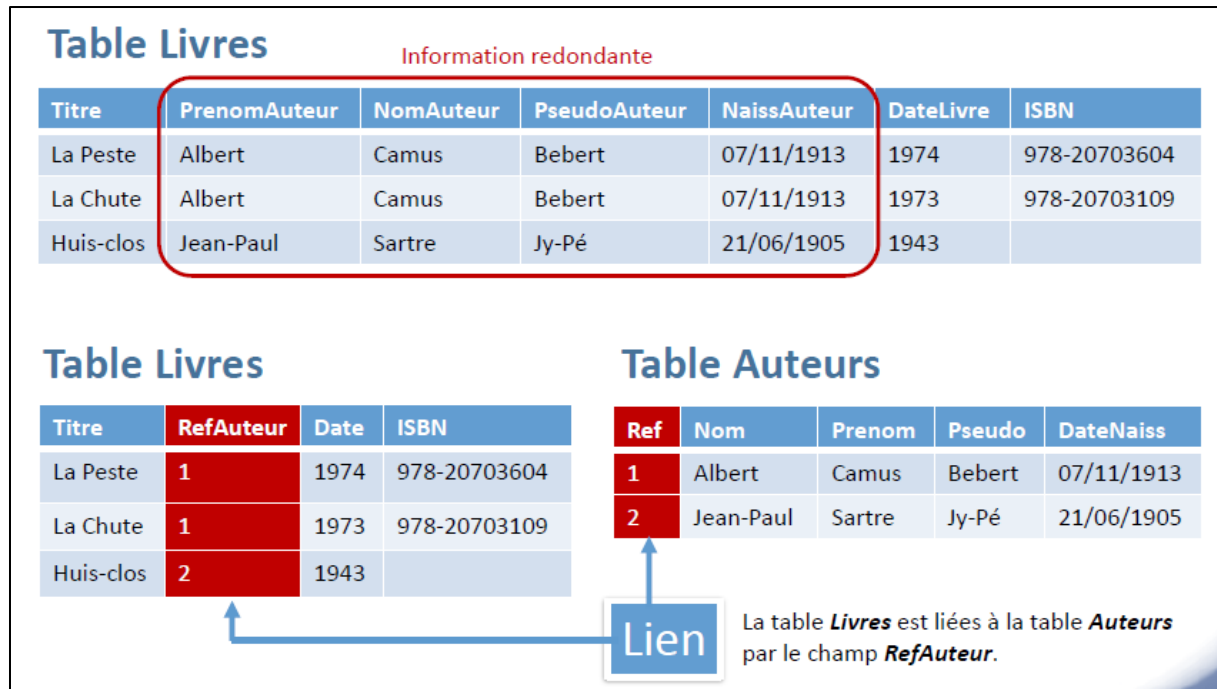
	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	rportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

$0 \leq \text{year_result} \leq 20$

Birth_date > 01-01-1930

La contrainte : FOREIGN KEY

La **contrainte de clé étrangère** d'une table (enfant) permet **d'éviter la redondance d'information** au niveau de cette table par le biais d'une colonne de référence **pointant** vers une colonne identifiante d'une autre table (parent) contenant le détail de l'objet référencé.



Définition d'une table

Une table possède un NOM

✓ Ce nom peut être composé de **128 caractères** et est choisi par l'utilisateur.

✓ Il est conseillé de choisir des noms **clairs** et **concis**.

× Les **caractères spéciaux** comme les accents et les **espaces blancs** sont interdits.

Une table possède des COLONNES

↑ Le nom de ces colonnes est limité par les mêmes contraintes que celles énoncées pour le nom de la table.

Les colonnes ont un TYPE

Le type de la colonne définit le type qu'auront les **valeurs de cette colonne**.

Les types principaux sont :

Symbole	Signification
INT INTEGER	entiers de -214743648 à 2147483647
DECIMAL(X,Y)	nombre décimal dont X est le nombre de chiffres au total et Y le nombre après la virgule de 0 jusqu'à 999.999
VARCHAR(taille)	chaines de caractères (de 1 à 8000 non-Unicode)
DATE	date du 01/01/1001 au 31/12/9999
BIT	Valeurs 1, 0 ou NULL

Les types de données numériques :

Symbole	Signification
TINYINT	Valeurs entières de -128 à 127
SMALLINT	Valeurs entières de -32768 à 32767
BIGINT	Valeurs entières de -9223372036854775808 à 9223372036854775807
NUMERIC(X,Y)	(idem DECIMAL)

Les types de données textuelles :

Symbole	Signification
CHAR(x)	Maximum 255 caractères.
VARCHAR(x)	Chaine de caractères d'une taille variable (1 à 255 caractères)
TEXT	Plus de 255 caractères.

Les types de données de dates et heures :

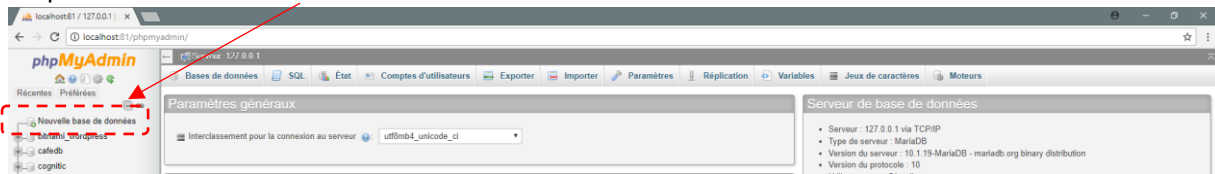
Symbole	FORMAT	Signification
TIME	'00:00:00'	Heures allant de -838:59:59 à 838:59:59
DATE	'0000-00-00'	Dates allant du 01/01/1001 à 31/12/9999
DATETIME	'0000-00-00 00:00:00'	Dates allant du 01/01/1001 au 31/12/9999 et Heures allant de 00:00:00 à 23:59:59
YEAR	0000	Années de 1901 à 2155
TIMESTAMP	0000000000000000	C'est la date du format AAAAMMJJHHMMSS

Il y a d'autres types mais nous ne les aborderons pas dans ce cours d'initiation.

Utiliser phpMyAdmin

Créer une base de données

Cliquez sur « Nouvelle base de données ».



Choisissez un nom pour votre nouvelle base de données. Nous allons l'appeler « messagerie ».

Cliquez sur « Créer ».



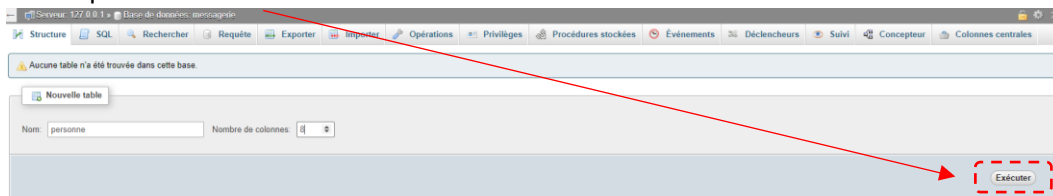
Supprimer une base de données

Aller dans l'onglet « Opérations » et cliquez sur « Supprimer la base de données (DROP) ». Attention cette action est irréversible !



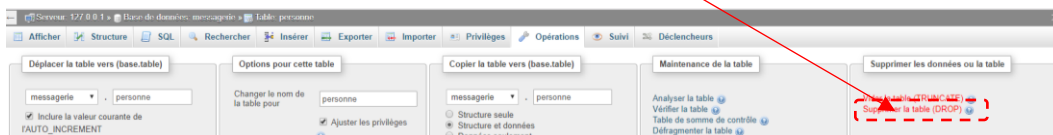
Créer une table

Maintenant que la base de données est créée, on peut ajouter une table « personne » avec 8 colonnes. Cliquez sur « Exécuter »



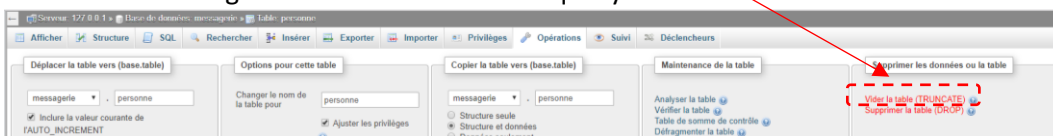
Supprimer une table

Aller dans l'onglet « Opérations » et cliquez sur « Supprimer la table (DROP) ». Attention cette action est irréversible !



Vider une table

Aller dans l'onglet « Opérations » et cliquez sur « Vider la table (TRUNCATE) ». Attention cette action est irréversible ! Vider signifie « retirer les données qu'il y a à l'intérieur ».



Créer les différentes colonnes de la table

Quelles sont les données relatives à la personne ?

N°	Nom	Type	NOT NULL	UNIQUE	PRIMARY KEY	Auto Incrémenté
1	Id	Int	X	X	X	X
2	nom	varchar(60)	X			
3	prenom	varchar(60)	X			
4	sexe	varchar(1)	X			
5	ad_rue	varchar(60)	X			
6	ad_num	varchar(10)	X			
7	ad_cp	int	X			
8	ad_ville	varchar(60)	X			

Ajoutons des noms et des types aux colonnes et leurs caractéristiques. Lorsque tout est complété, cliquez sur « Sauvegarder ».

Ce qui nous donne :

Notre base de données et notre première table sont créées. Maintenant, il nous reste à ajouter des données à l'intérieur.

Modifier une table

Allez dans « Structure ». Vous pouvez modifier et supprimer chaque colonne soit individuellement ou en cochant les colonnes à modifier/supprimer. La manipulation se déroule comme dans une boîte mail.

Insérer manuellement les données

Il est possible d'insérer des données manuellement en cliquant sur « Insérer ».

Colonne	Type	Fonction	Null	Valeur
Id	int(11)			
nom	varchar(60)			Lucky
prenom	varchar(60)			Luke
sexe	varchar(1)			M
ad_rue	varchar(60)			Rue du Western
ad_num	varchar(10)			77
ad_cp	int(11)			8500
ad_ville	varchar(60)			Courtrai

Entrez les données dans les différents champs puis cliquez sur « Exécuter ».

Voir le contenu d'une table

Allez dans l'onglet « Afficher », chaque personne est représentée dans une ligne.

✓ Affichage des lignes 0 - 0 (total de 1, Traitement en 0.0005 secondes.)

`SELECT * FROM `personne``

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

	Id	nom	prenom	sexe	ad_rue	ad_num	ad_cp	ad_ville
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	Lucky	Luke	M	Rue du Western	77	8500	Courtrai

☐ Tout cocher | Pour la sélection : ☐ Modifier ☐ Copier ☐ Effacer ☐ Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Chaque ligne peut être modifiée, copiée, effacée individuellement ou alors comme dans une boîte mail, vous pouvez en sélectionner plusieurs et faire ces mêmes actions.

Exercice 1 :

Consigne

Entrez les différentes personnes ci-dessous dans la table « personne »

Nom	Prénom	Sexe	Rue	Numéro	CP	Ville
Lucky	Luke	M	Rue du Western	77	8500	Courtrai
Le Gaulois	Astérix	M	Rue de la Bretagne	48	7500	Tournai
César	Jules	M	Rue de Rome	23	6000	Charleroi
D'Egypte	Cléopâtre	F	Rue de la Pyramide	14	5000	Namur
Curie	Marie	F	Rue de la Pologne	36	4000	Liège
Mère	Theresa	F	Rue de l'Albanie	58	6700	Arlon
Dion	Céline	F	Rue du Québec	3	8400	Oostende
De Belgique	Mathilde	F	Rue de Bruxelles	86	1000	Bruxelles
Einstein	Albert	M	Rue des Etats-Unis	147	9000	Gent
Armstrong	Neil	M	Rue de la Lune	985	2000	Antwerpen

Résultat

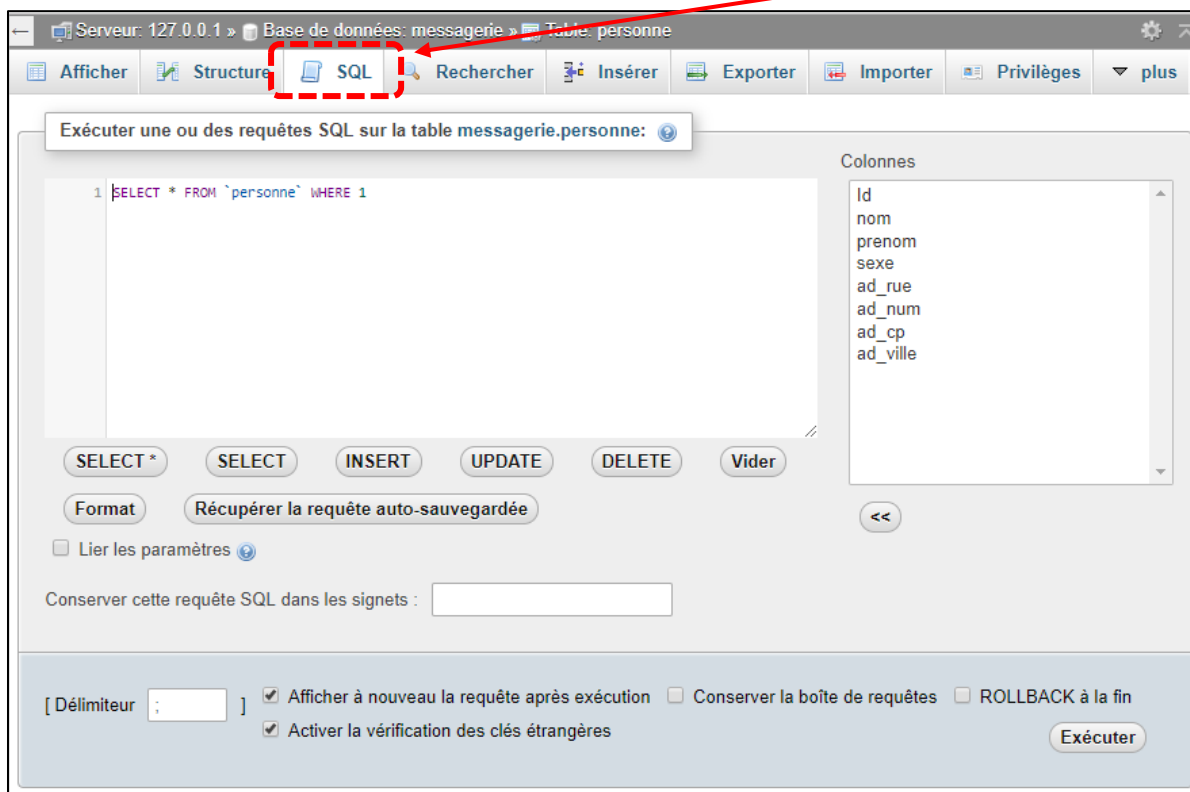
	id	nom	prenom	sexe	ad_rue	ad_num	ad_cp	ad_ville
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	Lucky	Luke	M	Rue du Western	77	8500	Courtrai
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	Le Gaulois	Astérix	M	Rue de la Bretagne	48	7500	Tournai
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	César	Jules	M	Rue de Rome	23	6000	Charleroi
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	D'Egypte	Cléopâtre	F	Rue de la Pyramide	14	5000	Namur
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	Curie	Marie	F	Rue de la Pologne	36	4000	Liège
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	Mère	Theresa	F	Rue de l'Albanie	5	6700	Arlon
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	7	Dion	Céline	F	Rue du Québec	3	8400	Oostende
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	8	De Belgique	Mathilde	F	Rue de Bruxelles	86	1000	Bruxelles
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	9	Einstein	Albert	M	Rue des Etats-Unis	147	9000	Gent
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	10	Armstrong	Neil	M	Rue de la Lune	985	2000	Antwerpen

Rechercher des données

L'objectif d'une base de données est de pouvoir trouver des résultats rapidement. Nous allons maintenant faire cette action de recherche en code SQL. Ce code sera placé dans notre code PHP, ce qui nous permettra d'afficher les résultats sur notre page web.

Dans phpMyAdmin

Dans un premier temps, nous allons tester notre code, dans l'onglet « SQL » :



Le code que nous allons écrire se placera dans la zone blanche. Une fois le code entré, cliquez sur « Exécuter » pour exécuter le code.

SELECT ... FROM

Pour trouver ces résultats, nous allons **sélectionner** les données depuis une **table**.

```
SELECT colonne1, colonne2, colonne 3
FROM nom_table
```

La casse n'a pas d'importance, mais on prendra l'habitude d'écrire les mots-clés du langage en majuscules.

On écrira généralement les clauses (« SELECT », « FROM », ...) sur des lignes différentes afin d'indenter au mieux le code.

Sélectionner toutes les colonnes et toutes les lignes

```
SELECT *
FROM nom_table
```

Exercice 2

Consigne

Quel sera le code pour afficher tous les résultats de notre table « personne » ?

Sélectionner toutes les lignes de certaines colonnes uniquement

```
SELECT nom_col1, nom_col2
FROM nom_table
```

nom	prenom	sexe
Le Gaulois	Astérix	M
César	Jules	M
D'Egypte	Cléopâtre	F
Curie	Marie	F
Mère	Theresa	F
Dion	Céline	F
De Belgique	Mathilde	F
Einstein	Albert	M
Armstrong	Neil	M

Exercice 3

Consigne

Quel sera le code pour afficher le nom, le prénom et le sexe des personnes de notre table « personne » ?

Changer le nom de la colonne

```
SELECT nom_col1 as "Nom1", nom_col2 'Nom2', nom_col3 Nom3, nom_col4 as
"Nom4"
FROM nom_table
```

Il est possible de changer le nom de la colonne en utilisant les alias. Cela ne change rien au niveau des données contenues dans les tables. Le mot-clé « as » n'est pas obligatoire.

Exercice 4

Consigne

Nom de famille	Prénom	Sexe	Ville
Le Gaulois	Astérix	M	Tournai
César	Jules	M	Charleroi
D'Egypte	Cléopâtre	F	Namur
Curie	Marie	F	Liège
Mère	Theresa	F	Arlon
Dion	Céline	F	Oostende
De Belgique	Mathilde	F	Bruxelles
Einstein	Albert	M	Gent
Armstrong	Neil	M	Antwerpen

Quel sera le code pour afficher le nom, le prénom, le sexe et la ville des personnes de notre table « personne » en changeant le nom des colonnes respectivement « Nom de famille », « Prénom », « Sexe » et « Ville » ?

Opérations arithmétiques

Il est possible de faire des opérations pour afficher un résultat en particulier.

```
SELECT nom_col1, nom_col2, (nom_col3/20)*100 "Résultat"
FROM nom_table
```

Opérateurs autorisés

Opérateur	Signification
/	Division
DIV	Division entière
*	Multiplication
+	Addition
-	Soustraction

Exercice 5

Consigne

prenom	Province
Astérix	7
Jules	6
Cléopâtre	5
Marie	4
Theresa	6
Céline	8
Mathilde	1
Albert	9
Neil	2

Quel sera le code pour afficher le prénom et les villes par province en divisant celle-ci par 1000 (division entière) ?

Ordonner les opérations

Il est conseillé d'utiliser des parenthèses afin de forcer le système à tester les conditions dans l'ordre souhaité.

Concaténation

Pour fusionner 2 colonnes en une, on utilise la fonction CONCAT().

```
SELECT CONCAT(nom_col1, nom_col2)
FROM nom_table
```

Attention ! Les 2 mots vont se coller. Il faut donc ajouter un espace blanc comme ceci :

```
SELECT CONCAT(nom_col1 , ' ', nom_col2)
FROM nom_table
```

Nom et prénom
Le Gaulois Astérix
César Jules
D'Egypte Cléopâtre
Curie Marie
Mère Theresa
Dion Céline
De Belgique Mathilde
Einstein Albert
Armstrong Neil

Exercice 6

Consigne

Quel sera le code pour afficher le nom et le prénom dans une seule colonne qui sera nommée « Nom et prénom » ?

Distinct

Distinct efface les doublons.

```
SELECT DISTINCT nom_col1
FROM nom_table
```

Exercice 7

Consigne

sexe

Quel sera le code pour afficher les différents sexes possibles ?

M

F

Limitier et ordonner les résultats

Il est possible de sélectionner des données qui respectent une condition et également d'ordonner les données.

```
SELECT nom_col1
FROM nom_table
WHERE conditions
ORDER BY liste_colonnes
```

La clause « WHERE » permet de limiter le nombre de lignes sélectionnées.

La clause « ORDER BY » permet de trier les résultats affichés, selon une ou plusieurs colonnes données.

Comme vu précédemment, ces clauses ne sont pas obligatoires mais si elles sont présentes, elles apparaissent dans cet ordre.

Il n'y a qu'une seule clause de chaque type. Il n'y aura donc jamais deux « WHERE » dans une même requête.

WHERE : où la condition est vraie

Si le champ respecte la condition, il sera affiché.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 >= 6
```

Opérateurs de comparaison

Opérateur	Signification
>	Plus grand
<	Plus petit
>=	Plus grand ou égal
<=	Plus petit ou égal
<>	Différent
!	Négation

nom	prenom
Lucky	Luke
De Belgique	Mathilde
Einstein	Albert
Armstrong	Neil

Exercice 8

Consigne

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le numéro de maison est supérieur à 50 ?

BETWEEN ... AND ... : entre ... et ...

Pour sélectionner les champs dont la valeur est comprise entre X et Y, ajoutez BETWEEN X AND Y.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 BETWEEN 10 AND 20
```

Cela revient à demander la liste dont les champs de nom_col2 est **plus grand ou égal** à 10 et **plus petit ou égal** à 20.

nom	prenom
Lucky	Luke
Le Gaulois	Astérix
César	Jules
D'Egypte	Cléopâtre
Curie	Marie
De Belgique	Mathilde

Exercice 9

Consigne

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le numéro de maison contient 2 chiffres ?

IN ... : égal à ...

Pour lister les champs dont la valeur est égale à X **ou** Y **ou** Z, ajoutez IN (X, Y, Z)

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 IN (10, 20, 30)
```

Exercice 10

Consigne

nom	prenom
Curie	Marie
De Belgique	Mathilde

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le numéro de maison est 36 ou 86 ?

C'est également possible avec des chaînes de caractères.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 IN ('mot1', 'mot2', 'mot3')
```

L'opérateur « IN » permet de comparer tous types de données, tant que les valeurs entre parenthèses sont bien du même type que la valeur comparée. **La casse n'a pas d'importance.**

Exercice 11

Consigne

nom	prenom
César	Jules
De Belgique	Mathilde

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont la ville est « Charleroi » ou « Bruxelles » ?

LIKE ... : comme ...

L'opérateur « LIKE » est utilisé pour comparer des chaînes de caractères entre elles. Le symbole « % » peut être utilisé pour remplacer de 0 à N caractères.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 LIKE 'm%'
```

« m% » signifie : commence par la lettre « m » peu importe la casse.

Exercice 12

Consigne

nom	prenom
D'Egypte	Cléopâtre
Dion	Céline
De Belgique	Mathilde

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le nom commence par la lettre « D » ?

La position du « % » signifie :

- positionné avant le caractère : le mot doit se terminer par ce caractère,
- positionné après le caractère : le mot doit commencer par ce caractère.

Le caractère « _ » signifie : un caractère indéfini.

Exercice 13

Consigne

nom	prenom
César	Jules
Mère	Theresa

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le code postal commence par le chiffre « 6 » ?

NOT ... : pas ...

L'opérateur « NOT » marque la négation des opérateurs « BETWEEN », « IN » et « LIKE ».

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 NOT LIKE 'm%'

/ WHERE nom_col2 NOT IN (4, 6, 8)

/ WHERE nom_col2 NOT BETWEEN 10 AND 20
```

nom	prenom
Lucky	Luke
Le Gaulois	Astérix
César	Jules
Curie	Marie
Mère	Theresa
De Belgique	Mathilde
Einstein	Albert
Armstrong	Neil

Exercice 14

Consigne

Quel sera le code pour afficher la liste des personnes (nom et prénom) dont le prénom ne commence pas par la lettre « C » ?

IS (NOT) NULL : est (pas) null

Afin de déterminer si une valeur est « NULL » ou non, il faudra utiliser la syntaxe « IS NULL » dont la négation sera « IS NOT NULL ».

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 IS NULL
```

AND : et

L'opérateur « AND » permet de combiner plusieurs conditions en même temps. Une ligne doit répondre simultanément à toutes les conditions pour faire partie du résultat.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 LIKE 'c%'
AND nom_col1 >= 10
```

Exercice 15

Consigne

nom	prenom
Curie	Marie

Qui sont les personnes dont le nom commence par « c » et qui sont des femmes ?

OR : ou

Tout comme son compère « AND », l'opérateur « OR » permet également de combiner plusieurs conditions en même temps.

Il suffit qu'un champ réponde à l'une des conditions pour faire partie du résultat.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 LIKE 'c%'
OR nom_col1 >= 10
```

Exercice 16

Consigne

nom	prenom
Lucky	Luke
Le Gaulois	Astérix
César	Jules
D'Egypte	Cléopâtre
Dion	Céline
De Belgique	Mathilde
Einstein	Albert
Armstrong	Neil

Qui sont les personnes dont le nom commence par « d » ou qui sont des hommes ?

Ordonner les résultats

La clause « ORDER BY » permet de trier le résultat d'une requête selon une ou plusieurs colonnes.

Le tri peut se faire de façon :

- croissante (« ASC » - ascendant) (valeur par défaut)
- décroissante (« DESC » - descendant) sur les valeurs.

Il est possible de trier en fonction d'une colonne qui n'est pas affichée.

```
SELECT nom_col1, nom_col2
FROM nom_table
WHERE nom_col2 LIKE 'c%'
ORDER BY nom_col1 ASC
```

Exercice 17

Consigne

Afficher la liste entière des personnes dans l'ordre alphabétique.

id	nom	1	prenom	sexe	ad_rue	ad_num	ad_cp	ad_ville
10	Armstrong		Neil	M	Rue de la Lune	985	2000	Antwerpen
3	César		Jules	M	Rue de Rome	23	6000	Charleroi
5	Curie		Marie	F	Rue de la Pologne	36	4000	Liège
4	D'Egypte		Cléopâtre	F	Rue de la Pyramide	14	5000	Namur
8	De Belgique		Mathilde	F	Rue de Bruxelles	86	1000	Bruxelles
7	Dion		Céline	F	Rue du Québec	3	8400	Oostende
9	Einstein		Albert	M	Rue des Etats-Unis	147	9000	Gent
2	Le Gaulois		Astérix	M	Rue de la Bretagne	48	7500	Tournai
1	Lucky		Luke	M	Rue du Western	77	8500	Courtrai
6	Mère		Theresa	F	Rue de l'Albanie	5	6700	Arlon

Il est possible d'ordonner **en fonction d'un alias** de colonne ainsi que sur une **combinaison de plusieurs colonnes**.

```
SELECT nom_col1, nom_col2 as 'Résultat'
FROM nom_table
WHERE nom_col2 LIKE 'c%'
ORDER BY nom_col1, 'Résultat' DESC
```

Exercice 18

Consigne

Nom complet	Ville
Le Gaulois Astérix	Tournai
Dion Céline	Oostende
D'Egypte Cléopâtre	Namur
Curie Marie	Liège
Einstein Albert	Gent
Lucky Luke	Courtrai
César Jules	Charleroi
De Belgique Mathilde	Bruxelles
Mère Theresa	Arlon
Armstrong Neil	Antwerpen

Afficher la liste des personnes (nom et prénom étant 'Nom complet') et la Ville en triant par ville par ordre non alphabétique.