

6.5/10

Notation :

- Quelques fautes d'orthographe et de style
- Le document manque de structure (titres et sous-titres)
- Les concepts sont présentés maladroitement
- La création des relations aurait pu être plus approfondie

Tutoriel "Comment créer une base de donnée MySQL", par Zaidou Tafridja

1. Les théories de bases de données :

Une base de données est un entité permettant de stocker des données de façon structurée et avec le moins de redondance possible.

Elle contient des renseignements tels que des chiffres, des dates, des mots etc. qui sont stockés dans des tables. Chaque ligne d'une table est un enregistrement.

Le MCD (Modèle Conceptuel des données):

Le modèle conceptuel des données est l'organisation des données. Il structure et renseigne toutes les caractéristiques de chaque type de données qui se trouvent dans la base de données, Chaque entité est lié à une autre, cette association est le lien qui existe entre ses 2 entités, par exemple entre un étudiant et une classe.

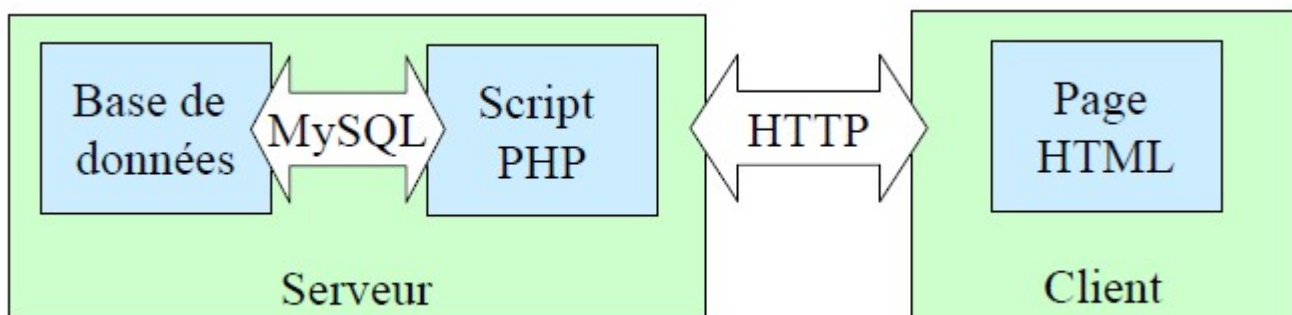
2. Création de la base de données MySQL:

Présentation :

MySQL est un système de gestion de base de données (SGBD). Il dérive de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel.

Le serveur de base de données MySQL est très souvent utilisé avec des langages de création sites web dynamiques (exemple: PHP).

Exemple:

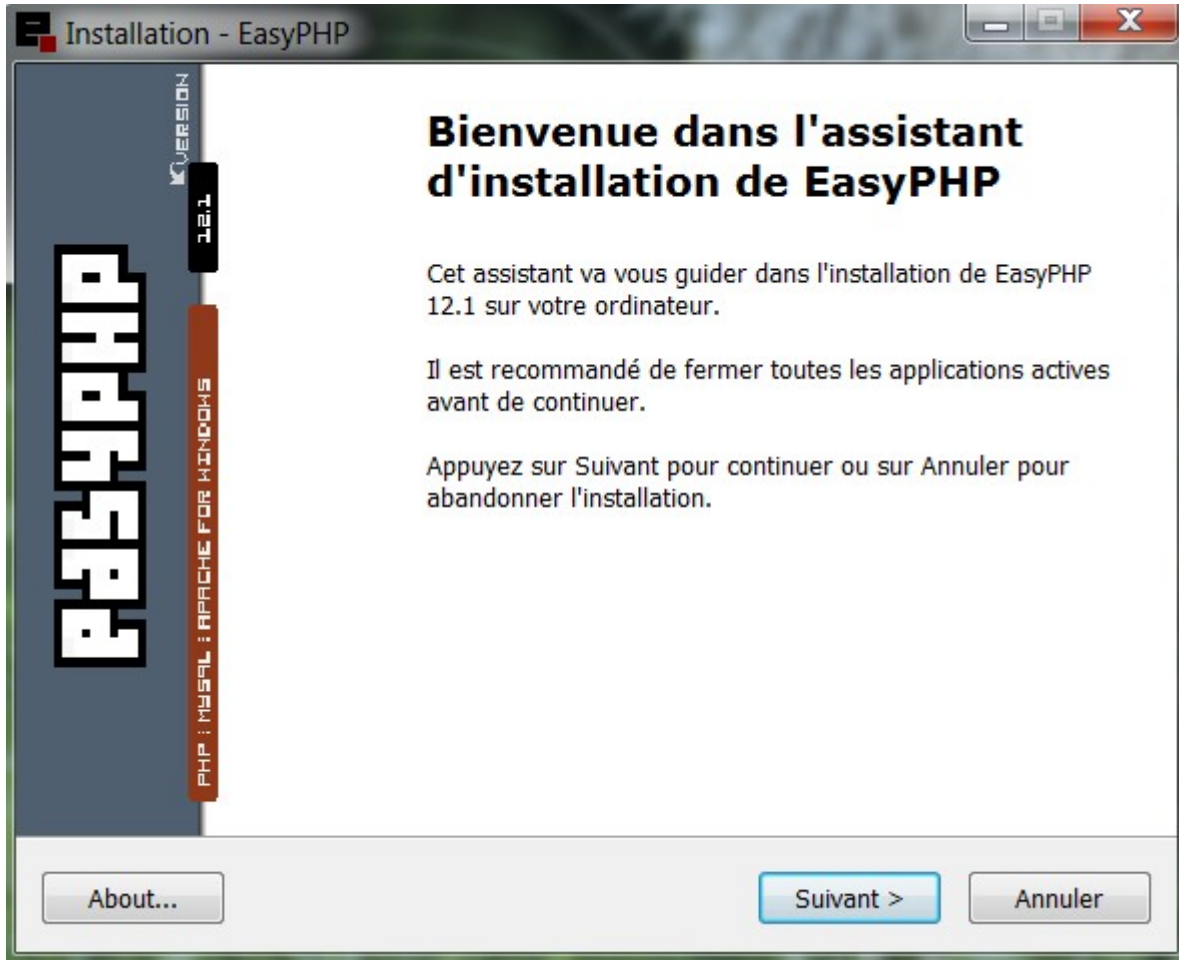


Nous allons pouvoir créer notre première base de données sur un système d'exploitation Windows 7. Pour cela nous avons besoin de mettre en place un serveur de base de données MySQL.

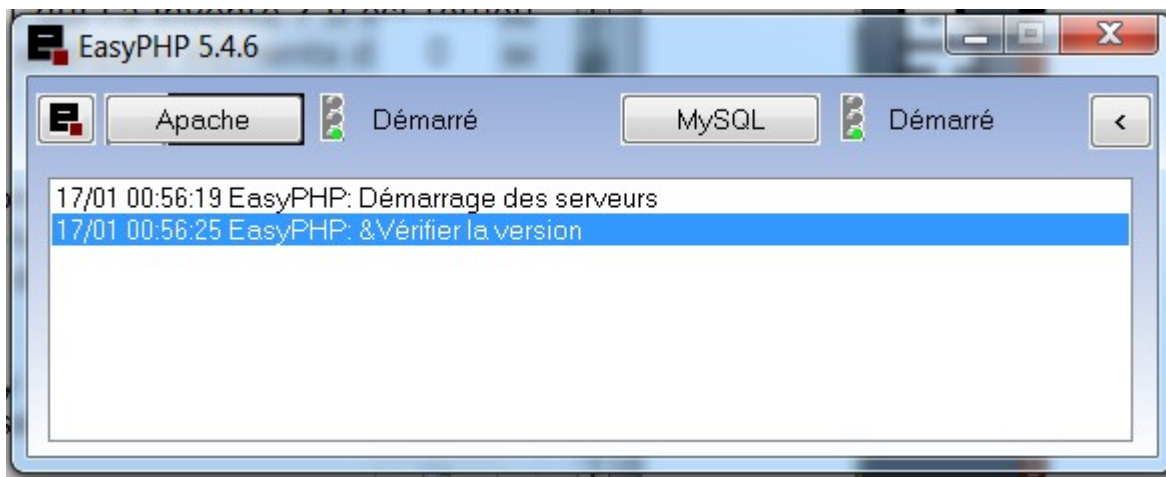
Nous allons installer donc EasyPHP, c'est un environnement de travail comprenant le serveur web Apache, le

système de base de données MySQL et le support du langage PHP.

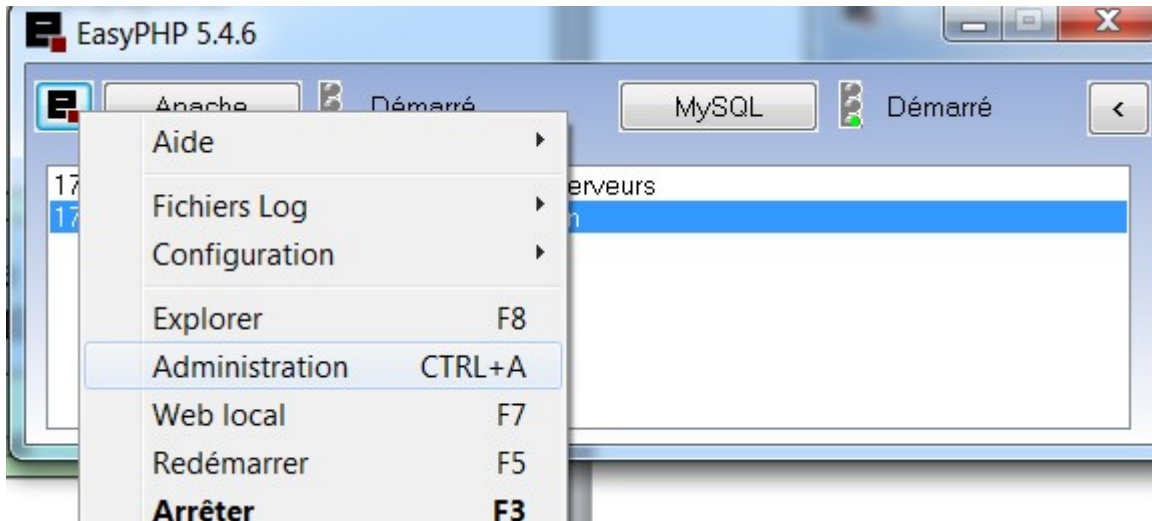
L'avantage de ce logiciel, c'est qu'il fournit phpMyAdmin, une interface permettant de manipuler très facilement les enregistrements de la base de données MySQL.



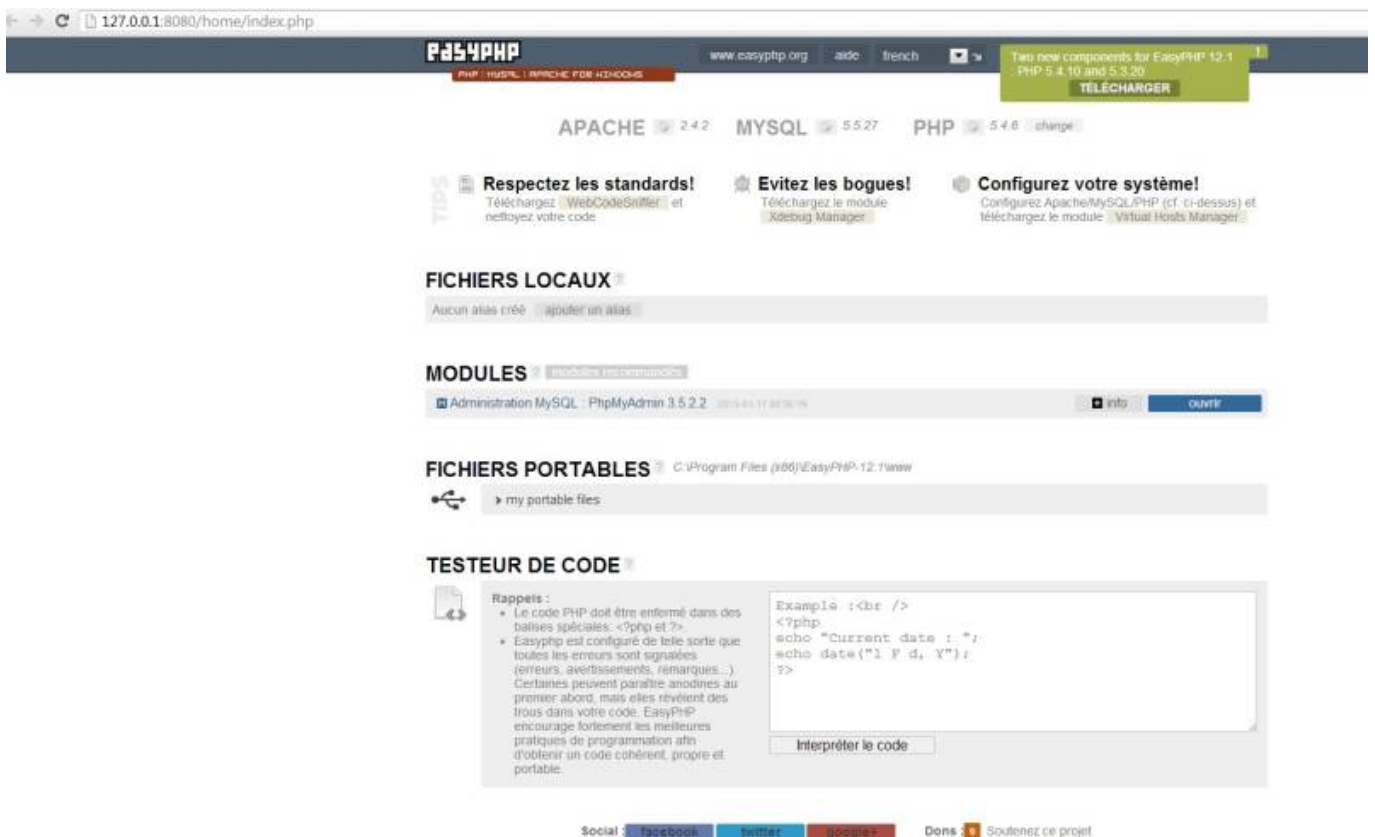
Une fois l'installation terminée, on exécute le raccourci « easy php.exe » et on se retrouve sur cette fenêtre.



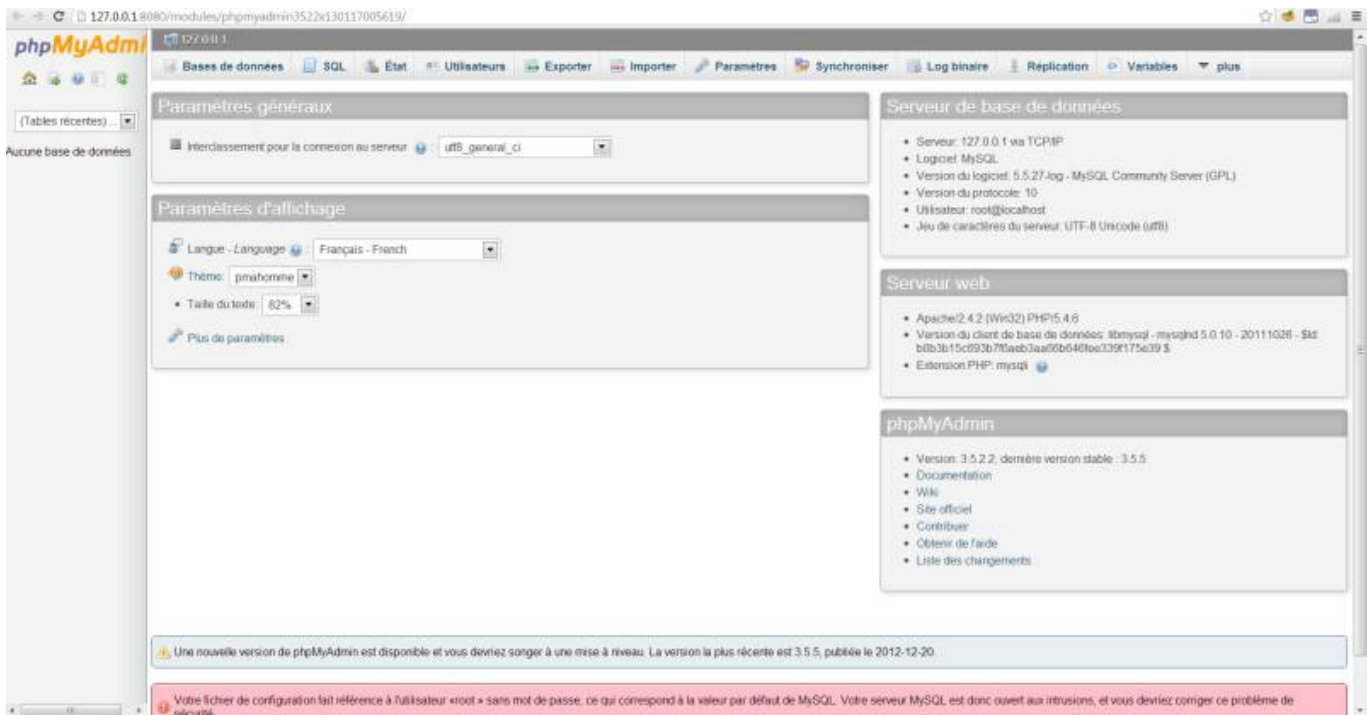
Les 2 serveurs (apache et mysql) sont lancés, cliquez sur « Administration »



Cliquez ensuite sur « ouvrir » l'environnement « phpMyAdmin »



Nous arrivons sur la page d'accueil de « phpMyAdmin »



Entrez dans le champ le nom de la base de données et cliquez sur créer, pour l'exemple nous allons créer une base de données qui contiendra les informations des étudiants de chaque classe.

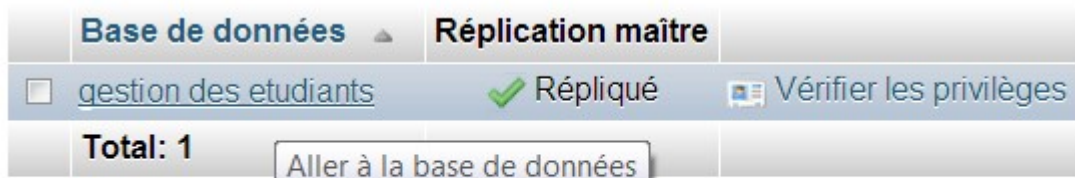


Bases de données

Créer une base de données


Aucune base de données

A présent cliquez sur la base de données que vous venez de créer.



/ Pour la sélection :

Entrez dans le champ « Nom » le nom de la première table « étudiant » et entrez « 8 » dans le nombre de colonnes puis cliquez sur « exécuter ». Nous allons voir pourquoi.

 **Nouvelle table**

Nom: Nombre de colonnes:

Une fois toutes les informations remplies, cliquez sur sauvegardé.

Nom de la table: Ajouter colonne(s)

Structure										
Nom	Type	Taille/Valeurs* 1	Défaut 2	Interclassement	Attributs	Null	Index	A_I	Commentaires	Type MIME
id_etudiant	INT	5	Aucune			<input checked="" type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>		
nom	VARCHAR	20	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
prenom	VARCHAR	20	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
date_de_naissance	DATE		Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
adresse	VARCHAR	100	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
code_postal	VARCHAR	5	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
ville	VARCHAR	50	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		
id_classe	INT	3	Aucune			<input type="checkbox"/>	---	<input type="checkbox"/>		

Commentaires sur la table:

Moteur de stockage: InnoDB

Interclassement:

Définition de PARTITION:

Dans la première colonne tout à gauche, nous allons rentrer les noms des champs de la table. Le premier champ est généralement l'ID, il permet d'obtenir avec certitude les données de l'enregistrement associé. Il garantit l'intégrité de ce dernier. Les autres champs seront utilisé pour stocker ce que nous avons besoin :

- Le nom de l'étudiant
- Son prénom
- Sa date de naissance
- Son adresse
- Son code postal
- Sa ville

« id_classe » est la clé étrangère de la table classe.

Une clé étrangère est un attribut qui point vers la clé primaire d'une autre table. L'objectif de cette clé étrangère est d'assurer l'intégrité référentielle des données.

Nous créons le champ maintenant et nous établirons par la suite la relation entre la table « etudiant » et la table « classe ». Dans la 2ème colonne se trouve les types de données possible pour chaque champ.

Les principaux types de données dans MySQL :

- Les types numériques
- Les dates et heures
- Les textes et chaînes de caractères

C'est ces 3 types que nous allons d'ailleurs utiliser.

La colonne « Null » permet au champ de contenir des valeurs nulles.

La colonne « index » possède plusieurs valeurs. Nous allons l'utiliser juste une fois et seulement pour l'id_etudiant, comme étant la clé primaire, « **Toute table doit avoir une clé primaire** ».

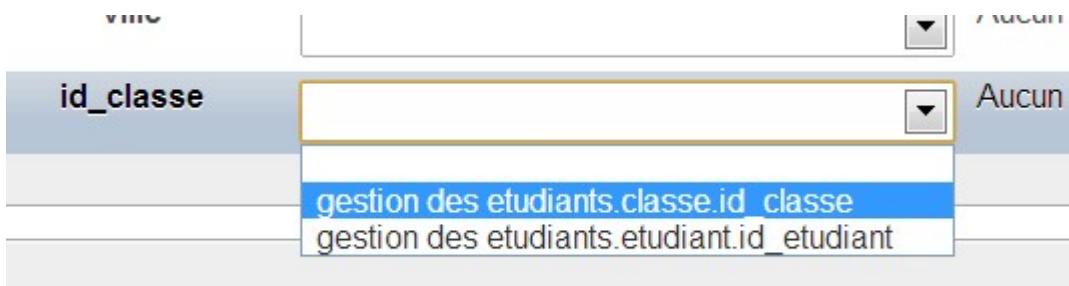
Nous allons répéter la même procédure pour créer la table « classe », elle est identifiée par un « id_classe » et un nom.

Nos 2 tables sont maintenant créées mais pas encore liées.

Pour créer la relation entre ces 2 tables, cliquez sur la structure de la table « etudiant » puis « vue relationnelle »



Enfin sélectionner l'id_classe de la table « classe » pour effectuer la relation et valider.



Nous avons terminé la création complète de notre base de données. La relation entre les 2 tables est maintenant effective. Nous allons pouvoir insérer des données. Nous commençons par insérer 2 classes par exemples.

Colonne	Type	Fonction	Null	Valeur
id_classe	int(11)	<input type="text"/>		<input type="text"/>
nom	varchar(20)	<input type="text"/>		BTS 1 SIO

Ignorer

Colonne	Type	Fonction	Null	Valeur
id_classe	int(11)	<input type="text"/>		<input type="text"/>
nom	varchar(20)	<input type="text"/>		BTS 2 SIO

Ensuite on insère des étudiants, en précisant pour chaque étudiant la classe dont il se trouve. (« id_classe » est la clé étrangère qui point vers la clé primaire de la table « classe », seules les valeurs existants dans la table « classe » apparaissent ici dans la liste déroulante du champ « id_classe »)

Colonne	Type	Fonction	Null	Valeur
id_etudiant	int(5)	<input type="text"/>		<input type="text"/>
nom	varchar(20)	<input type="text"/>		Zaidou
prenom	varchar(20)	<input type="text"/>		Tafridja
date_de_naissance	date	<input type="text"/>		1992-07-30
adresse	varchar(100)	<input type="text"/>		av de la valeuse
code_postal	varchar(5)	<input type="text"/>		14200
ville	varchar(50)	<input type="text"/>		Hérouville St Clair
id_classe	int(3)	<input type="text"/>		<input type="text"/>

- 1
- 2

Maintenant, grâce au langage SQL nous allons vérifier si notre relation est bien faite.

Exécuter une ou des requêtes SQL sur la base gestion des étudiants: 

```
1 SELECT *
2 FROM etudiant, classe
3 WHERE etudiant.id_classe = classe.id_classe
4 AND classe.nom = "BTS 1 SIO"
```

Résultat de la requête :

+ Options

id_etudiant	nom	prenom	date_de_naissance	adresse	code_postal	ville	id_classe	id_classe	nom
3	Zaidou	Tafridja	1992-07-30	av de la valeuse	14200	Hérouville St Clair	1	1	BTS 1 SIO
4	J	Audrey	1992-02-05	monadresse	14000	Caen	1	1	BTS 1 SIO

Conclusion : La base de données est maintenant créée et fonctionnelle, on pourrait ainsi l'utiliser dans une page php.

From:
<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:
<http://slamwiki2.kobject.net/si3/etudiants/td7/zaidou>

Last update: **2019/08/31 14:21**

