

Niveau 2 & 3

TP
S4. Développement logiciel
S4.7 - Langages de programmation
Web dynamique : PHP, SQL

Bases de données SQL

TABLE DES MATIÈRES

1 - Présentation.....	2
2 - Préparation de la base de données.....	3
3 - Exportation/Importation de la BdD.....	6
4 - Le site.....	7
4.1 - La page de garde.....	7
4.2 - La page de disponibilité.....	8
5 - Le script de traitement.....	8
5.1 - Les informations personnelles.....	8
5.2 - La base de données.....	8
5.3 - Traitement des résultats obtenus.....	9
6 - Accès à distance.....	9
7 - Bonus.....	10



1 - PRÉSENTATION

Le but de ce TP est la mise en place d'un service de validation d'accès à un site internet de réservation de cours de tennis du complexe sportif de la Cholière.

Pour cela, on utilisera sous Windows, le serveur **WAMP** muni de son moteur **PHP** complété par le SGBD **MariaDB**.

La page **index.html** du site demande l'identification du visiteur pour lui laisser l'accès à la page de réservation des cours de tennis. Elle se présente comme suit :

Bienvenue au

Complexe sportif de la Cholière.

Pour réserver un cours de tennis, veuillez vous identifier.

Nom

Prénom

Mot de passe

L'utilisateur renseigne ses nom, prénom et mot de passe. Le clic sur le bouton « **Effacer** » remet tous les champs de saisie à leur état initial : vide. « **Valider** » lance une requête de recherche de la personne et de la validation du mot de passe au serveur **MariaDB**. Celui-ci recherche dans la base de données la présence de la personne puis l'adéquation du mot de passe.

On peut obtenir alors l'un des écrans suivants en cas d'erreurs :

Erreur !

Vous devez remplir correctement et obligatoirement tous les champs.

Nom

Prénom

Mot de passe

ou bien :

Warning: mysqli_connect(): in Z:\Programmes\Internet_Serveurs\EasyPHP\data\localweb\Acces_Sport\AccesBdD.php on line 17

Connexion au serveur de Base de Données impossible

ou encore :

Cet utilisateur ne m'est pas connu. Veuillez vous identifier à nouveau.

Nom
Prénom
Mot de passe

ou finalement :

Mot de passe incorrect. Veuillez le ressaisir.

Nom
Prénom
Mot de passe

Si l'utilisateur est connu dans la BdD, la page suivante sera affichée :

Bienvenue, Cruzol christophe

PAGE DE DISPONIBILITE DES COURS

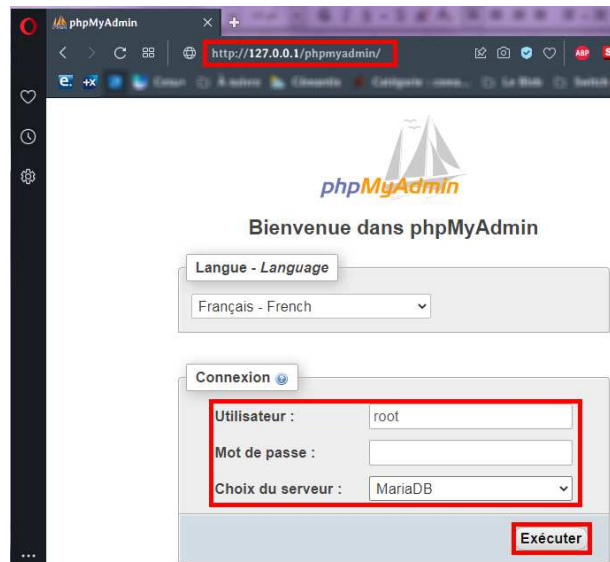
Une aide en ligne est disponible sur : <http://www.php.net/manual/fr>

2 - PRÉPARATION DE LA BASE DE DONNÉES

Respectez exactement les intitulés des bases de données, champs, enregistrements et données qui vous sont indiquées : votre site sera testé et validé sur une autre BdD que la vôtre. Tout changement, même infime, ne permettra pas la validation de votre travail, ce serait dommage !

- 1) Au moyen de [phpMyAdmin](#), cherchez à administrer vos bases de données.

Souvenez-vous que, pour que ce programme puisse se connecter à la base de données, le nom d'utilisateur par défaut de [MariaDB](#) est [root](#) et qu'il n'y a pas de mot de passe.



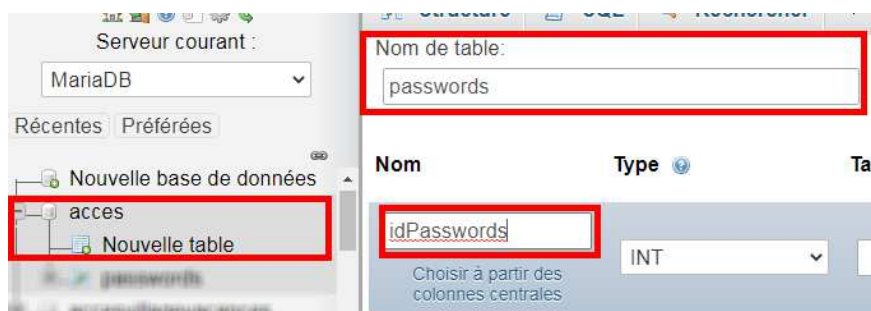
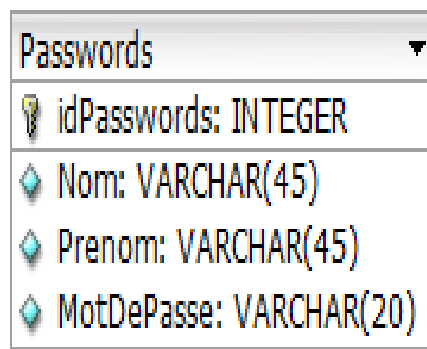
Page de connexion à MariaDB par le site local PhpMyAdmin.

- 2) Commencez par créer une base de données nommée **NOM_Prénom_acces**, sans aucun accent, ni espace. **Attention, dans les captures d'écran suivantes, la BdD s'appelle seulement *acces* !**

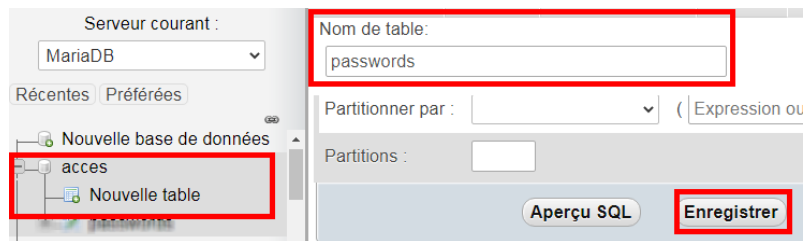


Création d'une nouvelle BdD.

- 3) Ajoutez-y une table nommée **passwords** dont la définition est la suivante :



Création d'une nouvelle table passwords.

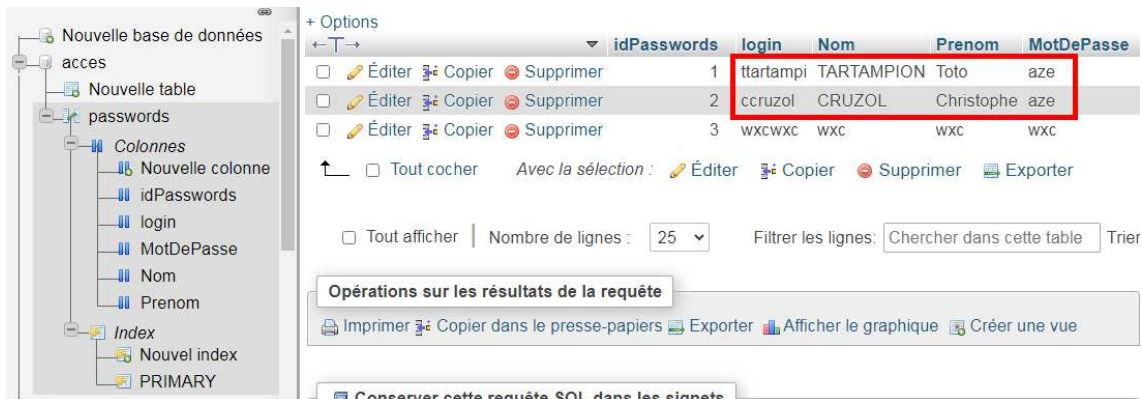


Ne pas oublier d'enregistrer la table !

#	Nom	Type	Interclassement	Attributs	Null	par défaut	Commentaires	Extra	Action
1	idPasswords	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier
2	login	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier
3	Nom	varchar(45)	latin1_swedish_ci		Non	Aucun(e)			Modifier
4	Prenom	varchar(45)	latin1_swedish_ci		Non	Aucun(e)			Modifier
5	MotDePasse	varchar(20)	latin1_swedish_ci		Non	Aucun(e)			Modifier

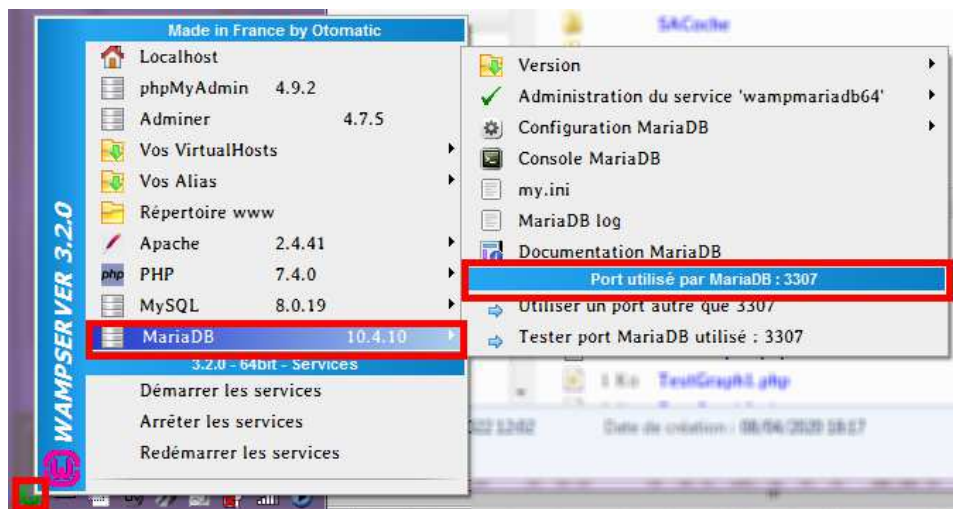
Les champs ont été complétés...

- 4) Ajoutez à cette table l'entrée correspondante à Monsieur **TARTAMPION Toto** et dont le mot de passe est **aze** !
Ajoutez aussi une autre entrée vous correspondant.

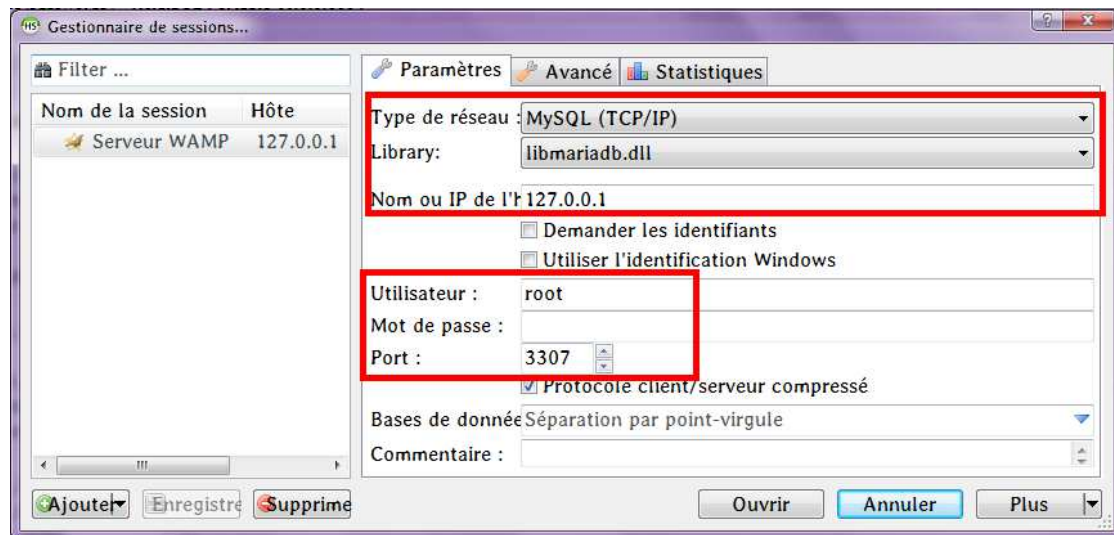


Notre BdD est remplie et fonctionnelle !

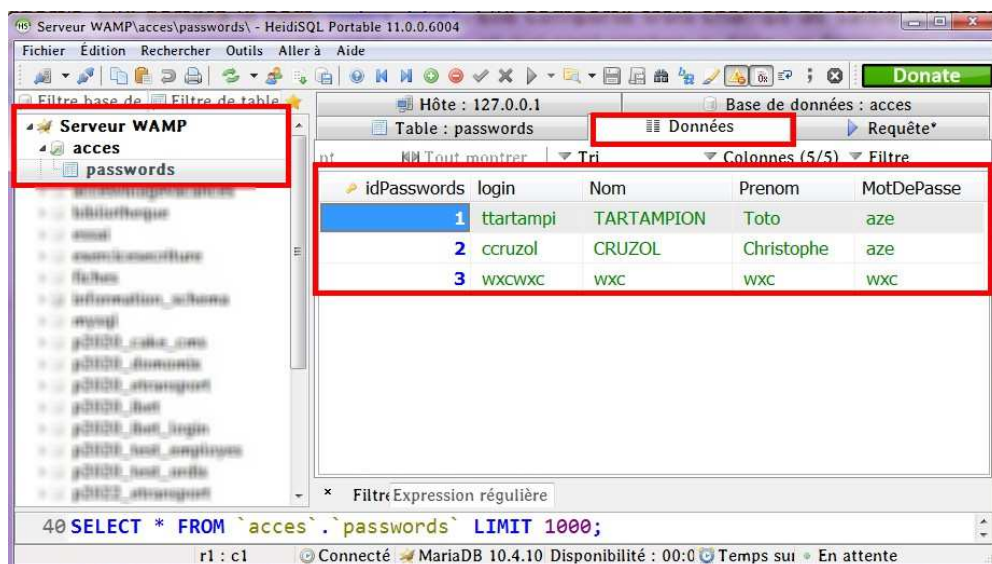
- 5) Validez votre travail en ouvrant la base de donnée **acces** avec **HeidiSQL**.



Récupération du port utilisé par le SGBD MariaDB de WAMP.



Paramètres de connexion à MariaDB par HeidiSQL.



La Bdd acces est disponible !

3 - EXPORTATION/IMPORTATION DE LA BDD

Pour pouvoir transférer votre base de données sur un autre serveur **WAMP**, sur un autre ordinateur, il vous faudra l'exporter grâce à l'option proposée dans **phpMyAdmin**.

- 6) Exportez votre base de données dans un fichier texte, généré automatiquement, qui portera le nom **NOM_PRENOM_Acces.sql** et que vous sauvegarderez dans votre dossier du TP.



L'option Exporter de phpMyAdmin.

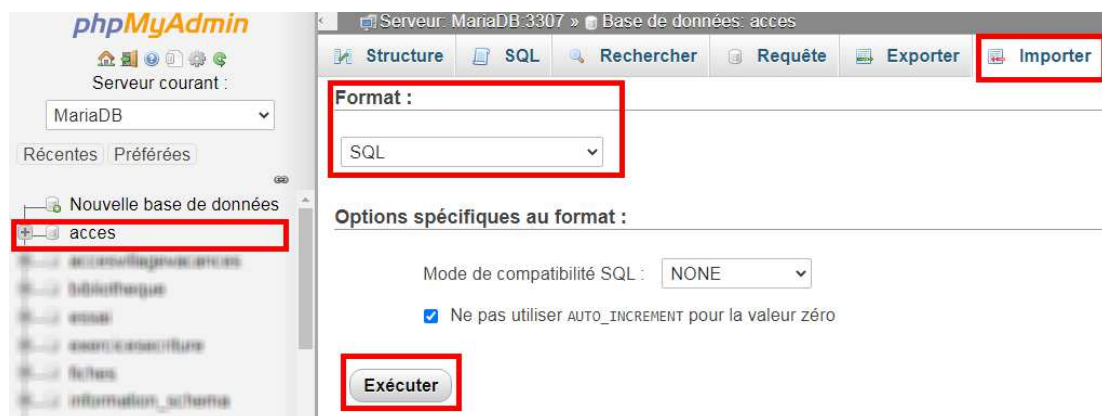


L'exportation en elle-même au format SQL.

Pour l'importer sur le serveur d'un autre ordinateur, lancez ce serveur puis phpMyAdmin. Créez une nouvelle BdD qui porte le même nom que celle d'origine. Sélectionnez-la et activez l'option **Importer** !



Importez le contenu d'une BdD d'un serveur dans un autre...



... puis validez l'importation au format SQL avec le bouton en bas de page.

4 - LE SITE

- 7) Créez un nouveau dossier nommé **NOM_Prénom_Accès_Sportif** dans le dossier **www** de **WAMP**.

4.1 - LA PAGE DE GARDE

- 8) Au moyen d'un éditeur de texte de votre choix, réalisez la page d'accueil du site du complexe sportif. Elle portera le nom **index.html**. Elle comporte trois champs de saisie qui sont référencés en tant que **nom**, **prenom** et **motdepasse**, et qui sont contenus dans un formulaire **HTML**. Le script **AccèsBdD.php** doit être appelé selon la méthode **POST** lorsque le formulaire est validé.
- 9) Validez votre page dans le navigateur en l'adressant par son adresse réseau locale.

4.2 - LA PAGE DE DISPONIBILITÉ

- 10) Créez une nouvelle page HTML, `disponibilites.html`, qui indique les disponibilités des cours de tennis du complexe, comme montrée à la page 3.
Le message de bienvenue ne fait pas partie de cette page : il est ajouté par le script PHP dans le cas de la validation de l'accès d'un utilisateur.

Il n'est pas demandé dans ce TP de gérer les plages horaires des cours, etc. Cette page n'affiche donc (pour le moment) que son titre !

5 - LE SCRIPT DE TRAITEMENT

Les éventuels **Warnings** sont générés par les serveurs eux-mêmes. Il est possible de les supprimer en remplaçant la ligne
`error_reporting = E_ALL | E_STRICT`
par
`error_reporting = E_ALL & ~E_NOTICE | E_STRICT`
dans le fichier `php.ini`.

5.1 - LES INFORMATIONS PERSONNELLES

- 11) Dans l'éditeur de texte, créez un nouveau script PHP, nommé `AccesBdD.php`.
Ce script récupère dans trois nouvelles variables `Nom`, `Prenom` et `MotDePasse`, les valeurs émises par le **POST** lors de la validation du formulaire.
Affichez les valeurs de ces variables.
- 12) Validez la présence de valeurs dans ces trois variables simultanément.
Si ce n'est pas le cas, affichez le message d'erreur montré à la page 2.
On mettra ici en œuvre une structure conditionnelle de type **SI**.

5.2 - LA BASE DE DONNÉES

- 13) Déclarez deux nouvelles variables `ServeurBdD` et `User` et initialisez-les avec les informations adéquates.
Notez que même si Apache permet l'accès distant (serveur Web), le serveur MariaDB est local au serveur Apache et possède ses propres utilisateurs et droits d'accès !

Pour la suite, nous allons utiliser les commandes fournies dans la librairie `mysqli` de PHP.

Elles possèdent généralement des valeurs de retour qui nous permettent de valider leur bonne exécution. Dans le cas contraire, il est obligatoire d'en informer l'utilisateur par un message adéquat puis de quitter le script.

- 14) Réalisez la connexion avec le serveur **MariaDB** et testez la réussite de votre entreprise. Si le serveur n'est pas accessible, affichez le message montré à la page 2 et quittez le script.
- 15) Codez, en toute fin du script, la fermeture de la connexion à la base de données.
- 16) Sélectionnez votre base **accus**.
- 17) Créez la variable **Requete** et initialisez-la avec la requête SQL qui permet de retrouver le mot de passe de l'utilisateur en fonction de son nom et prénom dans la base de données. Exécutez cette requête et affichez les éventuelles erreurs qui en résultent. Le résultat de la requête est placé dans la variable **Resultat**. Dans un premier temps, vous pouvez mettre au point votre requête en entrant la commande en dur. Par la suite, vous intégrerez les valeurs fournies par le formulaire de la page (variables **Nom**, **Prenom** et **MotDePasse**, de la question 11).

Si la personne n'est pas trouvée dans la base de données, affichez le message montré, dans ce cas-là, à la page 3 et quittez le script.

5.3 - TRAITEMENT DES RÉSULTATS OBTENUS

Nous allons rechercher le mot de passe saisi dans tous les résultats potentiels retournés par la requête **SQL**.

- 18) En supposant que tous les mots de passe se trouvent dans la variable **Resultat** (qui est un tableau) et en passant **Enregistrement** par **Enregistrement**, écrivez l'algorithme qui permet de trouver le mot de passe fourni par l'utilisateur (**MotDePasse**) dans ce tableau. Le tableau doit être parcouru mais dès que le bon mot de passe est trouvé, la boucle s'arrête : on ne sait pas combien de boucles doivent être faites. Bien sûr, si le mot de passe n'est pas dans le tableau, la boucle s'arrête aussi ! Pour cela, nous allons utiliser une structure de boucle de type **TANT QUE**.
- 19) Codez cet algorithme en **PHP** dans votre script.
- 20) Affichez soit le message de la page 3 correspondant au fait que le mot de passe saisi par l'utilisateur n'est pas trouvé dans la base de données ; ou celui qui souhaite la bienvenue à l'utilisateur, puisque son mot de passe est validé dans la base de données.
- 21) Pour que la page soit plus conviviale, trouvez un moyen d'afficher à nouveau, à la suite des messages d'erreurs, le formulaire de saisie. Si la validation du mot de passe de l'utilisateur est correcte, affichez à la suite du message de bienvenue (page 3), la page de disponibilités des cours (page 3).

6 - ACCÈS À DISTANCE.

- 22) Connaissant l'adresse IP de votre poste, modifiez la configuration de **WAMP** de façon à permettre l'accès de votre page depuis un ordinateur connecté au réseau.
- 23) Ajoutez, à vos pages et scripts, une marque permettant de reconnaître votre propre page : ce peut être un copyright en bas de page par exemple.

- 24) Sur un poste du réseau, essayez d'accéder à votre site de réservation et de vous faire reconnaître.
- 25) Essayez d'accéder de votre propre poste au site des autres étudiants de la classe.

7 - BONUS

- 26) Trouvez le moyen de cacher la saisie du mot de passe lors de l'identification.
- 27) Cryptez, par un moyen simple, les mots de passe dans la base de données.
- 28) Créez une page permettant l'enregistrement d'un nouvel utilisateur.
Les données saisies sont ajoutées dans la base de données seulement après avoir validé la double saisie du mot de passe.
- 29) Ajoutez les tables nécessaires pour la réservation de cours de tennis (tables relationnelles [cours](#), [reservation...](#)), et proposez les pages de saisie de la réservation.
La base de données est bien sûr mise à jour !