

## WampServer (c'est quoi ?) :

WampServer (**WAMP** pour «**W**indows» «**A**pache» «**M**ySQL» «**P**HP») est une plateforme de développement Web permettant de développer localement (sans avoir à se connecter à un serveur externe) des sites internet. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (**Apache** : un serveur web «frontal» qui répond directement aux requêtes du client web « navigateur » ; **MySQL** : système de gestion de base de données relationnelle SGBDR open source qui stocke l'ensemble des données de l'application et **MariaDB** serveur de base de données).

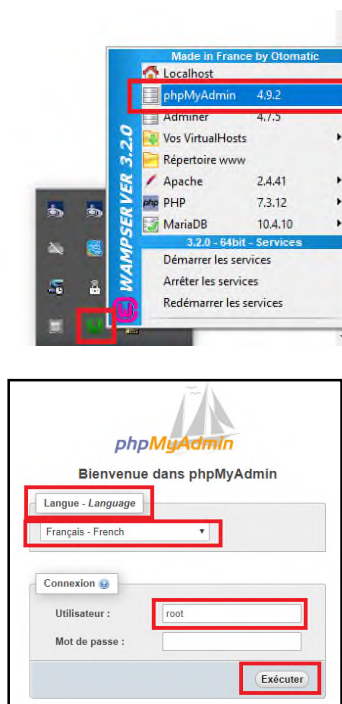
## Démarrage de WampSever :

Après avoir téléchargé et installé WampServer, allez au menu démarrer et exécutez « WampSever » (WampSever est disponible dans la liste des programme du menu démarrer). Vous allez remarquer l'apparition d'une icône dans la barre des tâches de votre bureau (la partie droite du bas de votre écran) :

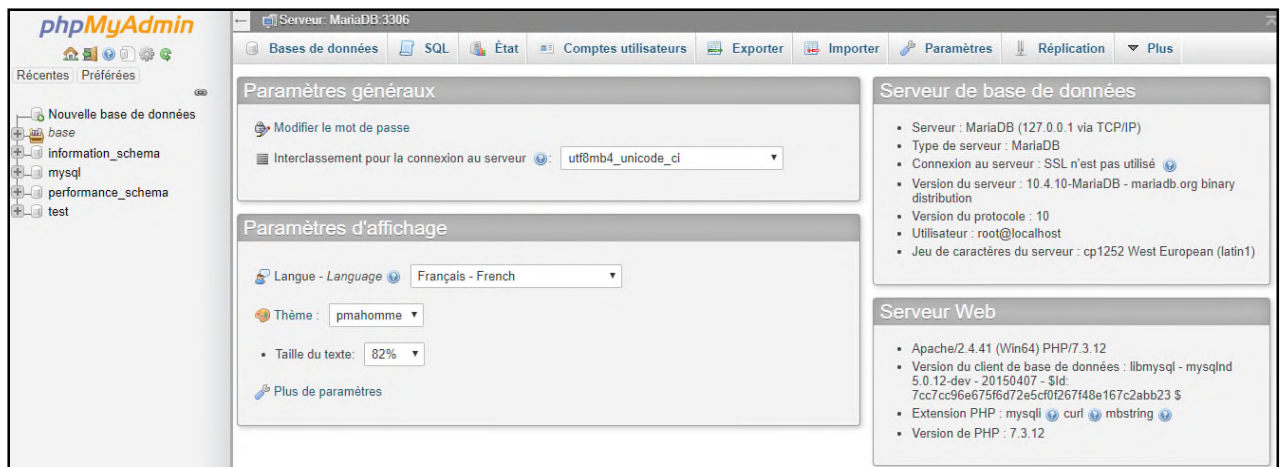


La couleur de cette icône va changer progressivement du rouge au vert en passant par la couleur orange (la couleur verte signifie que Wamp c'est lancé correctement).

Clic-gauche sur Wamp, vous obtenez une fenêtre dans laquelle est incluse plusieurs fonctionnalités. Clic sur PhpMyAdmin (vous obtenez le même résultat si vous allez à votre navigateur internet et vous mettez <http://localhost/phpmyadmin/>) :



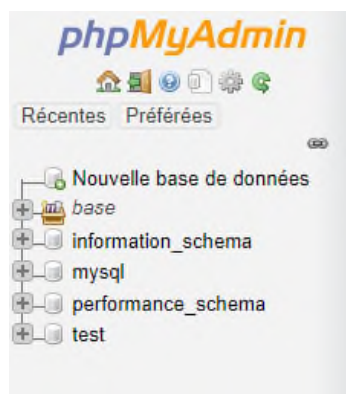
Choisir la langue française, mettre le nom d'utilisateur à « root », laissez le mot de passe vide et cliquez sur « exécuter ». Vous êtes sur la page d'accueil de phpMyAdmin :



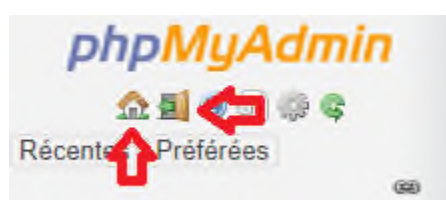
phpMyAdmin est une interface réalisée en PHP qui permet l'administration Web des bases MySQL. Il s'agit d'une interface graphique utilisée pour l'accès et la gestion de bases de données MySQL (phpMyAdmin n'est pas un système de gestion de bases de données relationnelles SGBDR mais une interface graphique qui facilite le travail avec le SGBDR MySQL).

Quand à MySQL, il s'agit d'un système de gestion de bases de données relationnelles SGBDR très largement utilisés qui repose sur langage SQL et fait partie des logiciels de gestion de base de données gratuits « open source » en concurrence avec des systèmes payants comme Oracle, Informix et Microsoft SQL Server.

Depuis la page d'accueil de phpMyAdmin, dans le panneau gauche apparaît 3 bases de données pré-crées ([information\\_schema](#), [performance\\_schema](#), [mysql](#)) qui assure le bon fonctionnement de [phpMyAdmin](#).

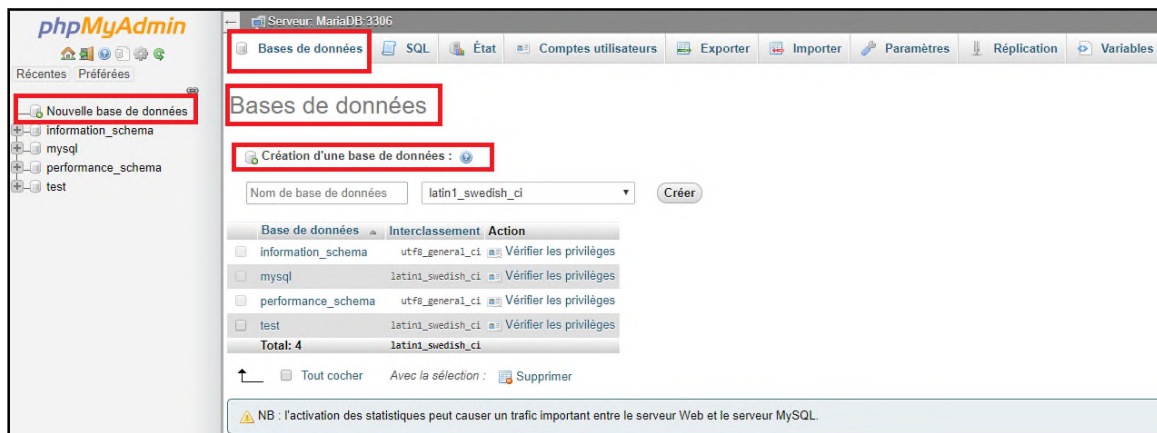


Il est possible de revenir à chaque fois à la page d'accueil de phpMyAdmin en utilisant l'icône « maison » et de sortir de phpMyAdmin en utilisant l'icône « Porte » :

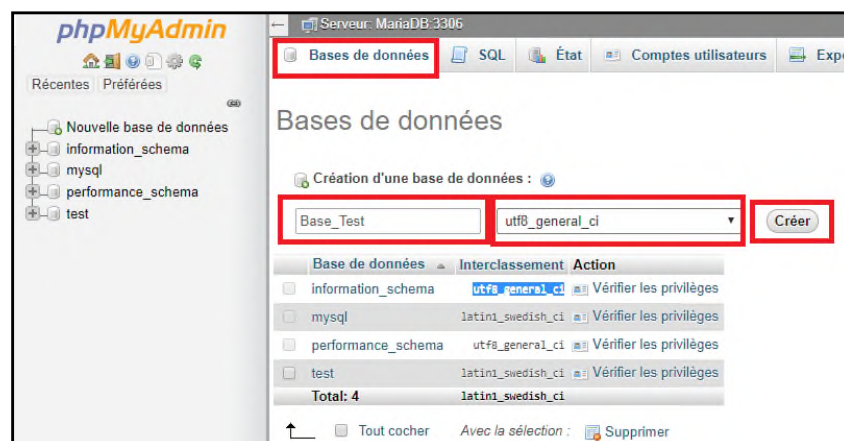


## Création d'une base de données

Pour créer une base de données, la boîte de dialogue apparaît comme indiqué dans la capture d'écran suivante:



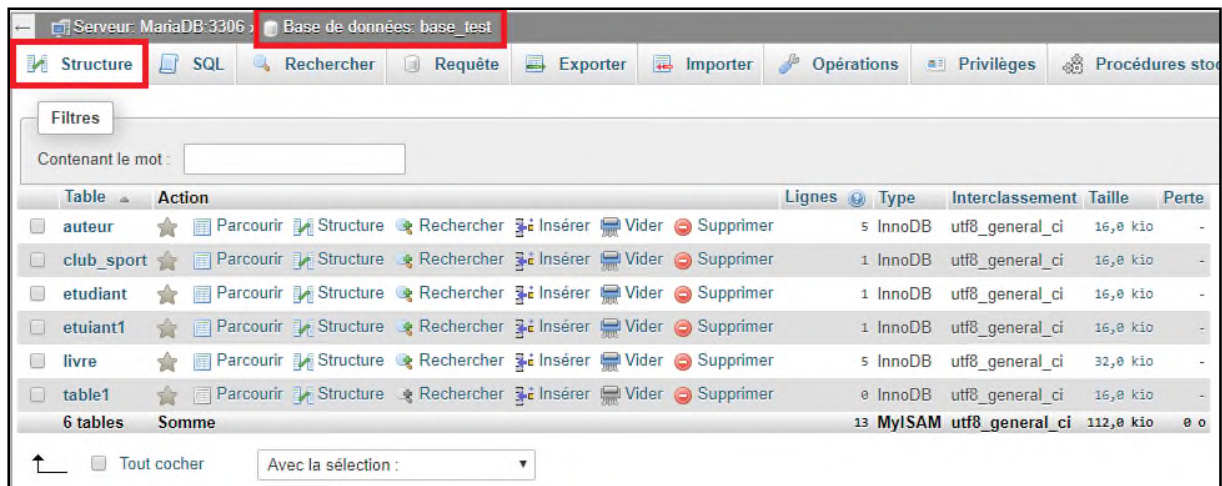
Remplir les champs nom de la base de données. Le premier champ est le nom de la base de données. Le second champ à coté du champ nom de la base est l'interclassement, qui permet de spécifier le type de codage des données (le mettre de préférence à `utf8_general_ci` car il permet gérer les accents et autres caractères propres au français). Une fois ces deux champs remplis, cliquer sur « créer » et la base de données sera créée.



Une fois la base créée, si on clic sur le nom de la base de donnée qui se trouve dans le panneau gauche, une vue global sur la base de donnée s'ouvre sur le panneau droit (allez au volet Structure).



On peut également cliquer sur le nom de la base de données qui se trouve en haut de phpMyAdmin (comme le montre la figure qui suit).



## Création d'une table :

Une fois la base de données créée, il faut créer l'ensemble des tables qui sont présentes dans la base. Avant de créer une table, il faut planifier les informations qu'on veut stocker dans cette table. Cela se fait généralement lors de la conception de la base de données. Dans notre cas nous allons prendre toute au long de ce manuel, l'exemple de deux tables « Auteur » et « Livre » qui sont liés par une liaison « Ecrire ». Les données relatives à un « Auteur » sont :

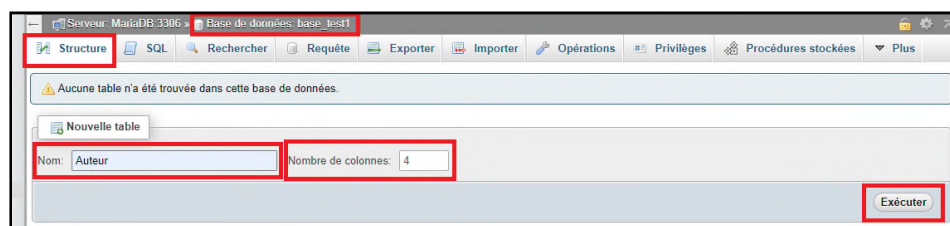
- Identifiant auteur (Id\_Ateur).
- Nom de l'auteur (Nom)
- Prénom de l'auteur (Prénom)
- Date de naissance de l'auteur (Date\_nais)

Les données relatives à un « livre » et qu'il est impérative de conserver sont :

- Numéro international normalisé du livre (ISBN)
- Titre du livre (Titre)
- Nombre de pages du livre (Nb\_page)
- Identification de l'auteur qui a écrit le livre (Id\_Ateur).

Il existe une relation entre ces tables qui relie ces deux dernières à travers l'identification de l'auteur qui a écrit le livre (Id\_Ateur est un identifiant dans la table « Auteur », c'est ce que nous appelons une clé primaire, et un attribut déclaré dans la table « livre » qui permet de faire référence à la table « Auteur », c'est ce que nous appelons une clé étrangère).

Pour la création d'une table il faut aller sur la vue global de la base de données (cliquer sur le nom de la base de données qui se trouve en haut de phpMyAdmin), volet « Structure » puis spécifier le nom de la table et le nombre de colonne qu'elle comporte (le nombre d'attributs que comporte la table). En règle générale, la longueur du nom d'une table et ne doit pas dépasser 64 caractères. La limite de 64 caractères existe également pour les noms de colonne.





Une fois ces informations introduits dans la boîte de dialogue, on clic sur « Exécuter » qui permet d'afficher une nouvelle fenêtre pour remplir les différentes informations liées aux colonnes de la table (panneau spécifiant les informations de colonnes, en d'autre terme, les informations de chaque propriété de la table).

Comme nous avons demandé quatre colonnes, nous obtenons quatre lignes d'entrée. Chaque ligne fait référence à des informations spécifiques à une colonne. La capture d'écran suivante représente une partie de ce panneau :

The screenshot shows the 'Table: Auteur' dialog box in MariaDB. The 'Nom de table' field contains 'Auteur'. The 'Ajouter' button is next to '1 colonne(s)'. The 'Exécuter' button is highlighted. The 'Moteur de stockage' dropdown menu is open, showing options like MyISAM, CSV, MRG\_MyISAM, MEMORY, Aria, MyISAM, SEQUENCE, and InnoDB. The 'InnoDB' option is highlighted.

À ce stade, peu importe si le nombre de colonnes est exactement connu ou pas, il est toujours possible d'ajouter des colonnes lors de la création d'une table à travers l'option [Ajouter ..... colonne\(s\)](#) :

The screenshot shows the 'Table: Auteur' dialog box in MariaDB. The 'Nom de table' field contains 'Auteur'. The 'Ajouter' button is next to '1 colonne(s)'. The 'Exécuter' button is highlighted. The 'Moteur de stockage' dropdown menu is open, showing options like MyISAM, CSV, MRG\_MyISAM, MEMORY, Aria, MyISAM, SEQUENCE, and InnoDB. The 'InnoDB' option is highlighted.

Chaque colonne a un champ « Nom » qui permet d'attribuer un nom à l'attribut (la colonne).

Chaque colonne a un champ « Type » (qui permet de sélectionner le type de données.). Les types les plus couramment utilisés se trouvent au début de la liste déroulante du champ « Type ».

Serveur: MariaDB:3306 » Base de données: base\_test » Table: Livre

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations

Nom de table: Livre Ajouter 1 colonne(s) Exécuter

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs
ISBN	VARCHAR	25	Aucun(e)		
Titre	VARCHAR	70	Aucun(e)		
Nbr_page	INT	10000	Aucun(e)		
Id_Auter	INT		Aucun(e)		

Il est à noter que [phpMyAdmin](#) propose une quantité très importante de types de données. Ceux-ci sont classés par catégories. En bonus, phpMyAdmin a eu la bonne idée de proposer au tout début de la liste déroulante du champ « Type » les quatre types de données les plus courants :

**INT** : ce sont les nombres entiers. On y trouve des types dédiés aux petits nombres entiers (**TINYINT**), aux gros nombres entiers (**BIGINT**), aux nombres décimaux, etc.;

**DATE** and **TIME** : ce sont les dates et les heures. La aussi le champ « Type » propose de nombreux types différents permettant de stocker une date (jour, mois, année), une heure, ou les deux à la fois.

**TEXT** : ce sont les chaînes de caractères et long texte (on peut y stocker un roman toute entier sans problème) ; Là encore, il y a possibilité d'adaptés la taille maximal d'un texte à un besoin spécifique. (on peut y stocker un roman sans problème) ;

Le type **VARCHAR** (caractère de taille variable) est largement utilisé lorsque le contenu de la colonne est alphanumérique. Ce type nécessite une longueur maximale de 255 caractères, ceci dit il est toujours possible de réduire ce nombre en spécifiant le nombre de caractère maximal qu'un utilisateur peut introduire. Si nous oublions de le faire une limite aux nombres de caractères maximal, un petit message contextuel nous rappelle plus tard lorsque de l'enregistrement des informations saisies. **VARCHAR** : texte court (entre 1 et 255 caractères); c'est un type adapté aux textes courts. Sa seule exigence est que vous devez indiquer la taille maximale du champ (entre 1 et 255). Si vous ne le faites pas, vous ne pourrez pas créer la table. Si vous ne savez pas à combien limiter votre champ, vous pouvez mettre la valeur maximale (255). **VARCHAR** : c'est un type adapté aux courts textes, vous devez indiquer la taille maximale du champ (entre 1 et 255). Si vous ne le faites pas, vous ne pourrez pas créer la table.

**SPATIAL** : cela concerne les bases de données spatiales, utiles pour ceux qui font de la cartographie. Ce ne sera pas notre cas, donc nous n'en parlerons pas ici.

**BOOLEAN** : booléen qui ne peut prendre que deux valeurs distinctes « True » et « False ».

Ces types cités couvrent 99 % de nos des besoins des utilisateurs, donc ce sont eux qu'il faut retenir.

## Les Champs à remplir pour chaque attribut de la table :

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Structure Attributs	Null	Index
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---

**Champ « Nom »** : c'est Nom attribué à la colonne (à l'attribut). Les noms de base et de table ne peuvent contenir d'espaces ou des caractères spéciaux, c'est pourquoi nous y insérons des tirets bas « \_ ». Bien qu'ils soient acceptés on évitera les caractères accentués dans les noms de base et de tables. Par contre nous pouvons les utiliser sans problème dans les titres de colonnes.

**Champ « Type »** : INT (nombre entier), VARCHAR (chaîne de caractères qui ne dépasse pas 255 caractères), TEXT (long texte), DATE, DECIMAL, BOOL (le type booléen), BLOB (fichiers), etc.

**Champ « Taille/Valeurs »** : On peut donner une limite de valeur pour chaque champ (limiter une chaîne de caractères (VARCHAR) à 50 caractères par exemple, ou limité les valeurs d'un entier à 100). Avec certains types de données (par exemple des chaînes de caractères **VARCHAR** sont limitées à 255 caractères) vous devez affecter des valeurs pour la longueur maximale d'une colonne. Pour d'autres types ce champ est facultatif.

**Champ « Valeur par défaut »** : On peut donner une valeur par défaut pour chaque champ. Cette option vous permet de définir une valeur standard pour une colonne. Celle-ci est toujours automatiquement ajoutée lorsqu'un paquet de données ne contient pas de valeur pour la colonne en question. Les valeurs par défaut des chaînes de caractères doivent être entre « '.....' ».

**Champ « Interclassement »** : Avec l'option «Interclassement», vous définissez le type de codage des caractères pour une colonne (champ important pour les caractères particuliers). Vous pouvez modifier le type de codage des données pour une table et pour toutes les colonnes. Il est conseillé mettre le champ «Interclassement» de préférence à **utf8\_general\_ci** car il permet gérer les accents et autres caractères propres au français.

**Champ « Attributs »** : Certains types de données peuvent être déterminés de manière plus détaillée par attributs facultatifs. Vous pouvez par exemple ainsi spécifier avec les attributs **signed** et **unsigned**, si un nombre entier ou à virgule flottante peut aussi prendre des valeurs négatives, et non uniquement positives.

**Champ « Null »** : indique si le champ peut être « Null » ou pas. Remarquons que si la colonne **NULL** reste décochée (ce qui se traduit en langage SQL par **NOT NULL**) alors les champs ne doivent pas être vides et doivent être remplis. Si nous avions voulu accepter des champs vides il aurait fallu cocher la case **NULL**. Remarquons après sauvegarde des informations de la table et sa création que la colonne **NULL** (dans le champ structure de la table en question) indique l'information « Non » ce qui se traduit dans les lignes de code SQL par **NOT NULL** et signifie que les champs ne doivent pas être vides. Si nous avions voulu accepter des champs vides il aurait fallu cocher la case **NULL** dans l'écran de création des colonnes.

**Champ « Index » :** Vous pouvez définir des règles d'indexation via l'option « Index ». Sélectionnez un paramètre d'indexation **PRIMARY** pour une colonne indique que cette dernière est considéré comme la clé primaire de la table. Sélectionnez un paramètre d'indexation **INDEX** pour une colonne signifie que cette colonne est une clé étrangère qui fait référence à une table avec laquelle la table actuelle à un lien (cet attribut est une clé primaire dans la table de référence qui est référencée par l'attribut clé étrangère). Le paramètre **UNIQUE** spécifie que les valeurs ne peuvent être stockées qu'une fois dans cette colonne. Ainsi, les duplications peuvent être évitées si besoin. En générale, les paramètres les plus fréquents pour ce champ sont **PRIMARY** et **INDEX**. En générale ce champ est rempli pour quelques attributs, pour les autres ont n'a pas besoins de ce champ.

**Champ « A\_I » :** pour ce champs qui désigne l'abréviation de « **AUTO-INCREMENT** », il spécifie et précise si la valeur d'un attribut doit être auto incrémentée ou non. Cette option est très utile pour produire un numéro d'identification (ID) unique à chaque nouvelle insertion (il est très utile de cocher la case **AUTO\_INCREMENT** afin que ce champ gère lui-même les nouvelles valeurs automatiquement (1, 2, 3,4...)). Cocher la case **AUTO\_INCREMENT** permet à un champ de gérer lui-même les nouvelles valeurs automatiquement à chaque nouvelle saisie d'instances pour la table. Il est important de noter que lors de l'insertion des tuples de la table, les attributs dont le champ **A\_I** est coché ne doivent pas être rempli et doivent rester vide (ils sont rempli automatiquement par système de gestion de base de données).

**Champ « Commentaires » :** permet de donner un descriptif de la colonne (décrire le sens de l'attribut).

En utilisant la barre de défilement vers la droite, vous trouverez encore plus de paramètres (Il existe d'autres champs pour les colonnes tel que : Virtualité, Déplacer une colonne, etc.) que nous avons choisi de ne pas décrire dans ce guide MySQL pour débutants, et que sont laisser vides dans la majorité des cas.

Vous remarquerez peut-être la boîte de dialogue **Ajouter « .... » colonne (s)** en haut de l'écran. Nous pouvons l'utiliser pour ajouter des colonnes à ce panneau de création de table en entrant la valeur appropriée des colonnes supplémentaire nécessaires à la définition de la table et en appuyant sur « Exécuter ». Le nombre de lignes en entrée changerait en fonction du nouveau nombre de colonnes, laissant intactes les informations déjà entrées sur les premières colonnes déjà rempli.

The screenshot shows the MySQL Table Editor window for a table named 'Auteur'. The 'Ajouter' dialog box is open, allowing the user to add new columns. The dialog has a text input for the number of columns to add, currently set to '1', and a button labeled 'Exécuter' (Execute). Below the dialog, the table structure is displayed with columns for Name, Type, Size/Values, Default Value, Index, Attributes, Null, and Index.

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---



Avant d'enregistrer la page de création des colonnes de la table, il convient de remplir le champ « Index » pour au moins un des attributs qu'elle comporte. Chaque table doit avoir une clé primaire (une colonne avec un contenu unique qui représente chaque ligne d'une manière unique. Toute table doit posséder un champ qui joue le rôle de clé primaire. La clé primaire permet d'identifier de manière unique chaque entrée dans la table. Chaque entrée de votre table doit pouvoir être identifiée de manière unique grâce à la clé primaire). Il est nécessaire d'avoir une clé primaire pour l'identification des lignes d'une manière unique. Ainsi, le champ Index doit être positionné sur PRIMARY pour au moins un attribut de la table. D'autres possibilités pour le type d'index peuvent être sélectionnées (INDEX, UNIQUE et FULLTEXT, etc.) pour les autres attributs.

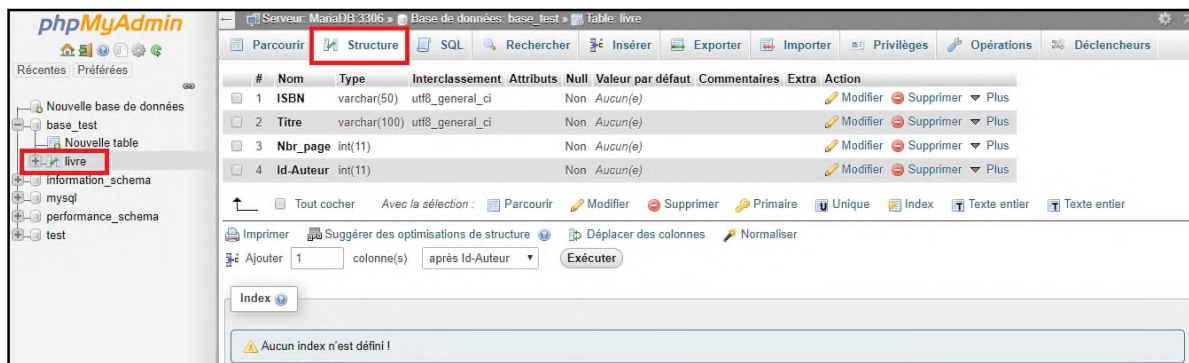
En générale un identifiant Id d'une table (surtouts s'il est artificiel) est de type entier INT, qui servira comme clé primaire de la table (Champ Index positionné sur PRIMARY), et est auto-incrémenté (cocher la case AUTO\_INCREMENT).

Faite bien attention à mettre le champ « Moteur de stockage » à InnoDB (et non pas à MyISAM) lors de la création de toutes vos tables (cela vous sera d'une grande utilité lors de l'élaboration des liaisons entre vos tables).

Après avoir rempli les différents champs de la table, il faut cliquer sur « Sauvegarder » pour sauvegarder toutes les informations saisies de la table.

L'instruction SQL apparaît sous le message encadré de vert pour confirmer d'exécution de la sauvegarde et création de la table (**SHOW CREATE TABLE** [TableName](#); permet d'afficher le code SQL qui a été généré lors de la création de la table [TableName](#)).

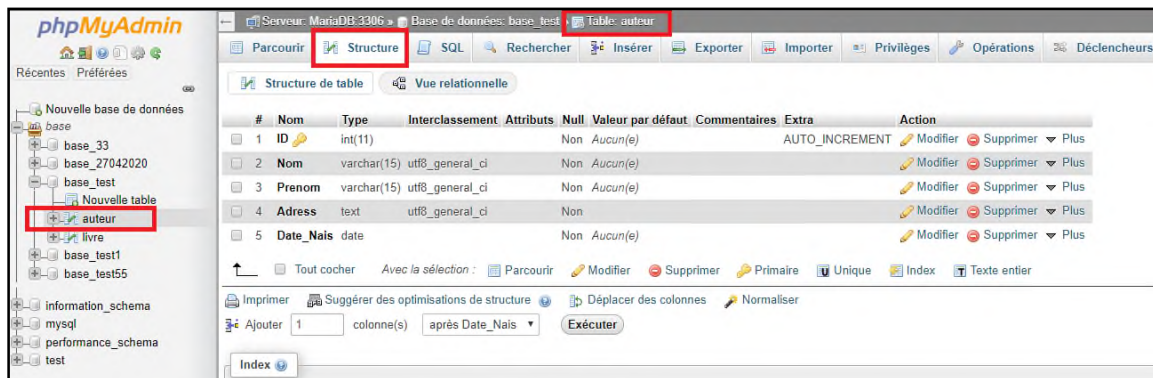
Une fois les informations de la table sauvegarder la fenêtre suivante apparait :



La table créée est automatiquement sélectionnée comme table de travail dans la base de données courante. La page « Structure » de la base de données actuelle confirme que la table a été créée.



Pour afficher la structure de la table, allez au panneau gauche, cliquez sur le nom de la table, et aller au volet « Structure » pour le visionnement de tous les champs qui composent la table :



Les icônes suivants sont visibles pour chaque colonne (attribut) de la table :



Ces icônes à différents endroits de l'interface signifient :

**Modifier** : modifier l'attribut (la colonne) de la table.

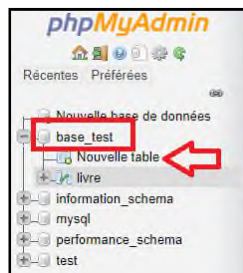
**Supprimer** : supprimer cet attribut (la colonne) de la table.



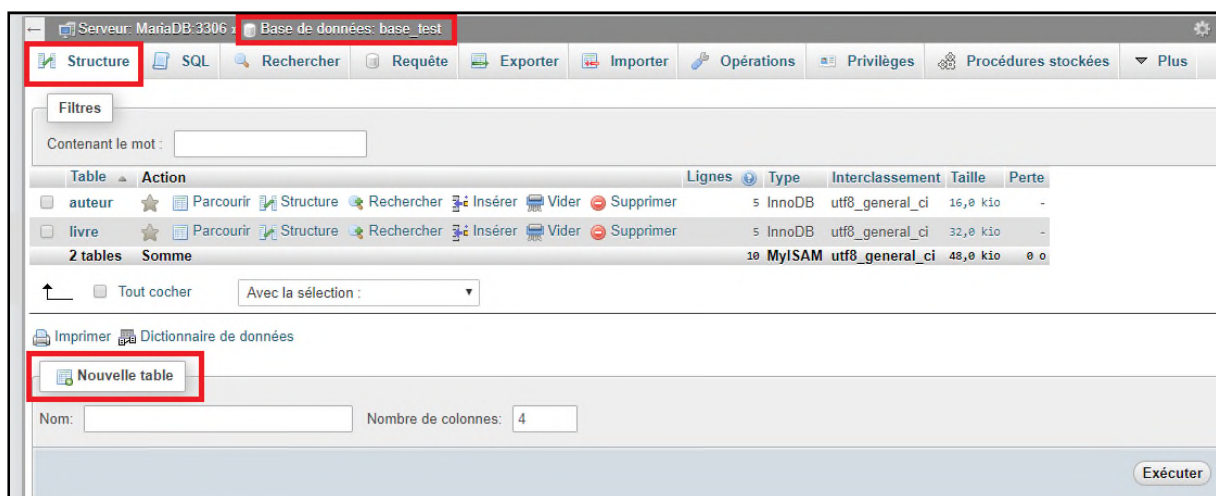
**Plus :**

- **Primaire** : donner à l'attribut la propriété d'être une clé primaire.
- **Unique** : donner à l'attribut la propriété d'être une clé unique
- **Index** : donner à l'attribut la propriété d'être un index (identifier une clé étrangère)

On peut créer une nouvelle table en cliquant sur « Nouvelle table » de panneau gauche de phpMyAdmin.



On peut également cliquer sur le nom de la base de données qui se trouve en haut de phpMyAdmin (comme le montre la figure qui suit) puis créer la nouvelle « Nouvelle table »



## Insérer des tuples (un enregistrement) dans une table « option insérer » :

Pour insérer de nouveaux tuples ou instances (manuellement) dans une table, il faut utiliser l'onglet « insert » de la vue globale de la table. Cette page produit un formulaire pour insérer des valeurs dans chaque champ d'un nouveau tuple. Le volet « insert » sert à insérer des données dans une table. Il convient d'ouvrir la table et de cliquer sur l'onglet « Insérer » pour qu'une fenêtre de saisie des données s'ouvre.

Table: auteur

Colonne	Type	Fonction	Null	Valeur
ID	int(11)			
Nom	varchar(15)			
Prenom	varchar(15)			
Adress	text			
Date_Nais	date			

Exécuter

La colonne « **Function** » permet de réaliser une transformation lors de l'insertion d'un tuple (elle ne vous sera pas utile).

La saisie des valeurs se fait dans le champ valeur de chaque colonne.

Pour les champs qui « s'auto-incrémentent » il faut les laisser vides. Une fois la saisie terminée, cliquer sur « Exécuter » pour insérer les données dans la table.

Table: livre

Colonne	Type	Fonction	Null	Valeur
ISBN	varchar(50)			
Titre	varchar(100)			
Nbr_page	int(11)			
Id-Auteur	int(11)			

Exécuter



La création des tuples est réalisée lorsque l'on appuie sur le bouton « Exécuté ». Notez que les champs pouvant être « Null » peuvent être laissés vides. Les champs d'identification qui ont été sélectionnés « AUTO\_INCREMENT » sont également être laissés vides (le système se charge alors de leur donner un numéro unique et séquentiel).

L'option en bas de la fenêtre [Continuer l'insertion avec « ..... » lignes](#) permet de faire plusieurs insertions de tuples dans une même session de saisie à la fois (le nombre d'instances à saisir à la fois est spécifié également dans ce volet).

Une fois les insertions d'instances terminées (avec le bouton exécuté), la fenêtre qui suit s'affiche, et indique la requête SQL (requête faite en langage SQL) correspondante à l'action d'insertion d'instances:

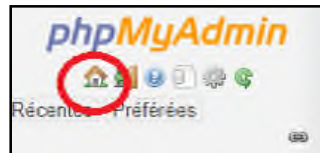
### Afficher les tuples d'une table « option Parcourir » :

On peut afficher les tuples contenus dans la table sélectionnée en utilisant l'onglet « Parcourir » dans la page centrale de la table.

ISBN	Titre	Nbr_page	Id-Auteur
ISS13	titre1	115	1
ISS14	titre2	213	2
ISS15	titre3	119	3
ISS16	titre4	226	4
ISS17	titre7	126	7

Pour changer de table, cliquez sur la table voulue dans le menu de gauche (dans l'illustration, il n'y a qu'une seule table disponible).

Pour retourner à la page d'accueil de phpMyAdmin, utiliser l'icône « maison » (à gauche également).



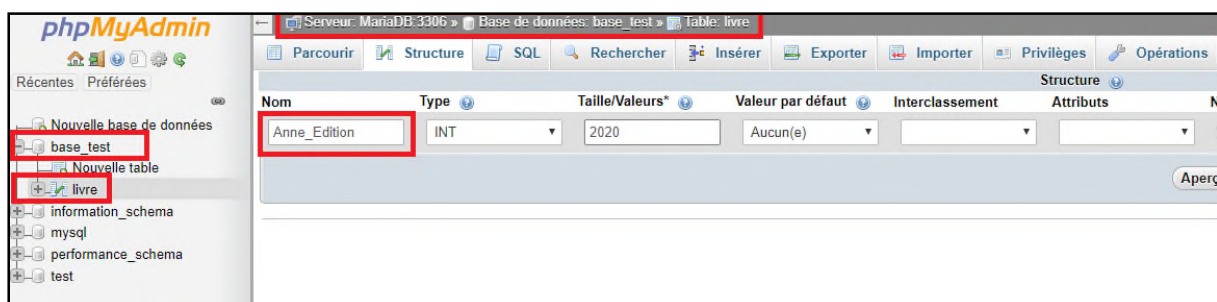
## Modifier les tables

La modification d'une table se produit sous l'onglet « Structure » de la page centrale de la table sélectionnée.

Les différents champs de la table peuvent être modifiés ou supprimés avec les icônes correspondantes. Il est aussi possible d'insérer de nouveau attributs dans la table. Pour la création de nouveaux attributs il faut entrer le nombre de nouveaux attributs qu'on désire insérer dans la table et leurs emplacement dans celle ci (ou insérer le nouveau attribut : au début de la table, à la fin, ou après un autre champ donné) et cliquer sur « Exécuter ».



Lors de l'ajout d'un nouvel attribut ou de sa modification, la page de remplissage de l'attribut s'affiche et on peut remplir les caractéristiques de l'attribut de la même manière que lors de la création d'une table. Une fois les différents champs remplis, sauvegarder les informations saisies.



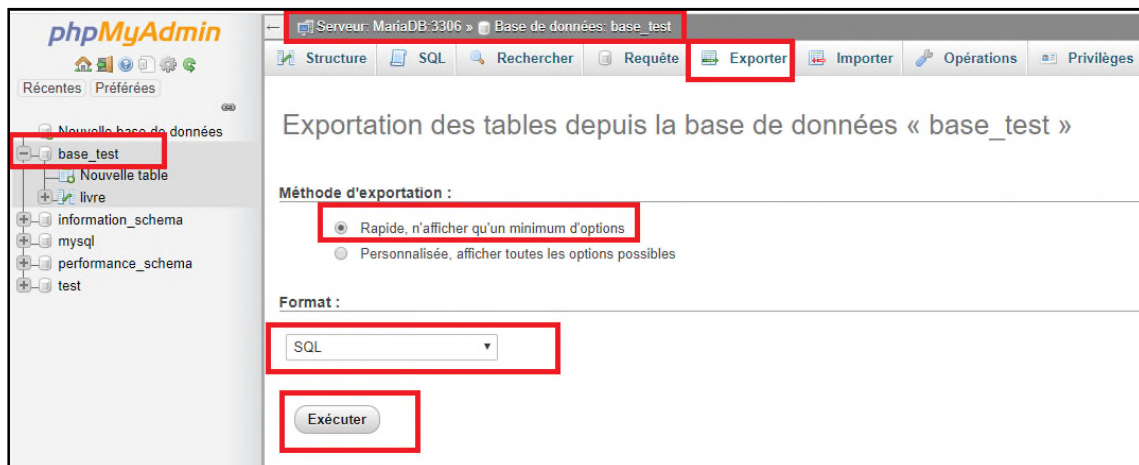
## Sélectionner une table

Lorsqu'une base de données contient plusieurs tables, il faut utiliser le menu de gauche pour changer de table « active » (la table sur laquelle on travaille). Pour changer de table active, cliquez simplement sur son nom de la table ciblée dans la liste de gauche (panneau gauche).

## Exporter une base de données

L'exportation est réalisée dans la page centrale de l'onglet « Exporter ».

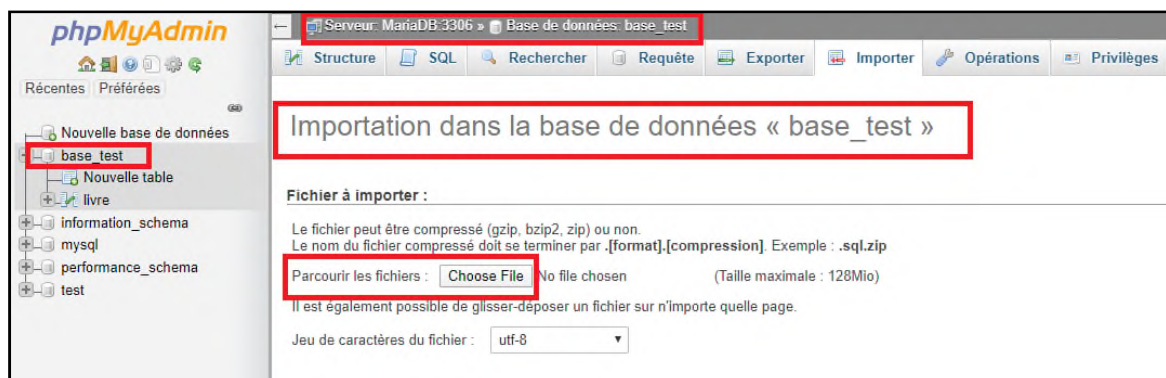
Cette opération permet de « sauvegarder » votre base de données. Pour se faire choisir la base de données à exporter sur le panneau gauche de PhpMyAdmin puis aller vers le volet « Exporter » (garder le format d'exportation SQL et ne touchez pas aux options d'exportation) et cliquez sur « Exécuter ». Vous allez ainsi pouvoir récupérer votre base de données sur le disque dur sous forme de fichier « .sql ».



## Importer une base de données :

L'importation est réalisée dans la page centrale sous l'onglet « Importer ». L'importation d'un fichier « .sql » permet charger toutes les tables s'y trouvant.

Il s'agit simplement de choisir le fichier avec le bouton « Parcourir », puis d'appuyer sur « Exécuter ».



## Opérations

L'onglet « Operations » de la page centrale permet certaines opérations globales sur la base de données ou la table active (permet de créer, modifier, renommer, vider les bases et les tables). L'onglet « Opérations », pour une base de données sélectionnée permet entre autres de renommer la base de données ou en réaliser une copie.

## Entrer des commandes SQL

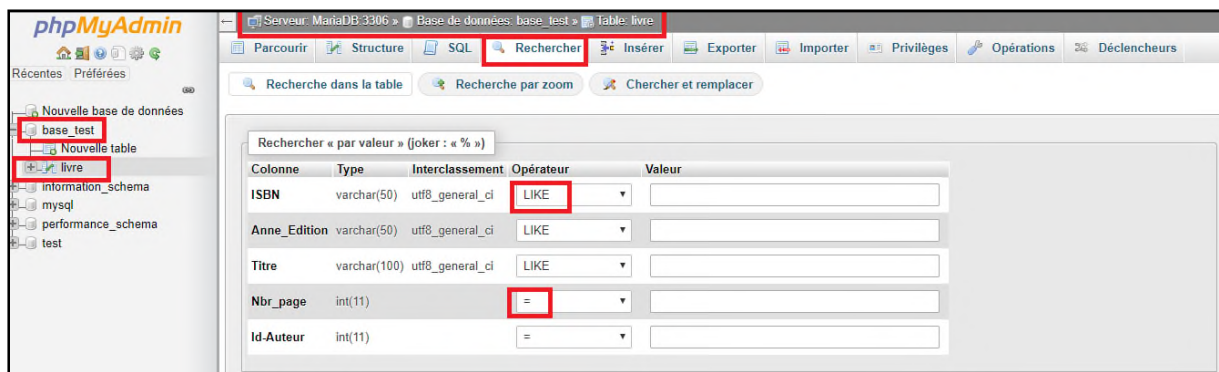
L'utilisateur peut utiliser le langage SQL pour exécuter des commandes, sans utiliser l'interface graphique de PhpMyAdmin. Pour ce faire, il suffit d'aller à l'onglet « SQL » dans la page centrale.

## Requête SQL

Pour précéder à une requête, il suffit d'entrer le code SQL dans la boîte de l'onglet « SQL ». Cliquez ensuite sur « Exécuter » pour l'exécuter.

## Recherche

L'onglet « Search » (« Recherche ») propose une interface pour la recherche dans une table. Pour chaque champ, on peut utiliser un opérateur (exemple l'opérateur LIKE est utilisé pour les chaînes de caractères : il est indifférent à la casse – majuscule / minuscule). Cet onglet permet également de faire des « Requêtes » (une recherche croisée entre les champs de différentes tables).



## Gérer les usagers du système « option Privilèges » :

La gestion des usagers du système est réalisée sur l'onglet « Privilèges », accessible depuis la page d'accueil de phpMyAdmin ou lorsqu'une base de données est sélectionnée.

L'onglet Privilèges permet la gestion des utilisateurs. Généralement on travaille sous l'utilisateur « root » mais il peut être utile de créer plusieurs utilisateurs quand on gère plusieurs bases de données.

Pour cela il faut cliquer sur « Ajouter un utilisateur » et renseigner les champs puis cliquer sur Exécuter. Cet onglet permet également de modifier les privilèges globaux ou locaux (concernant une base de données uniquement) d'un usager.

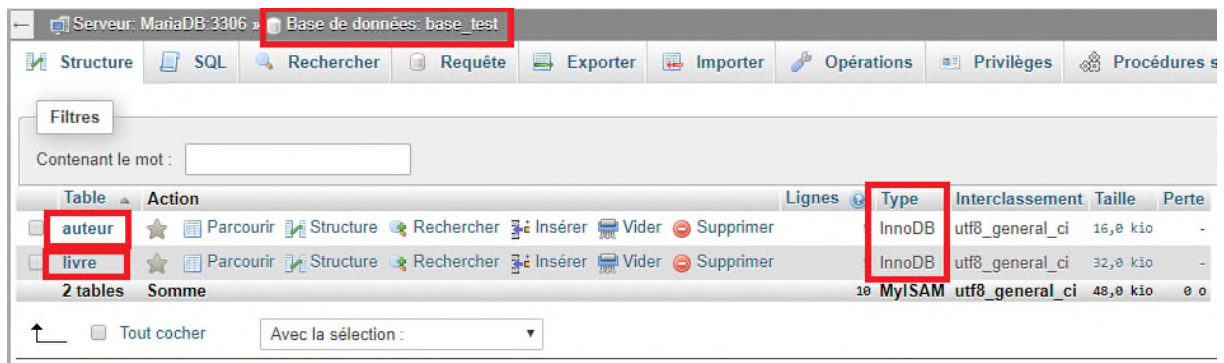
## Définir les relations inter-tables :

Cette section explique comment définir les relations inter-tables.

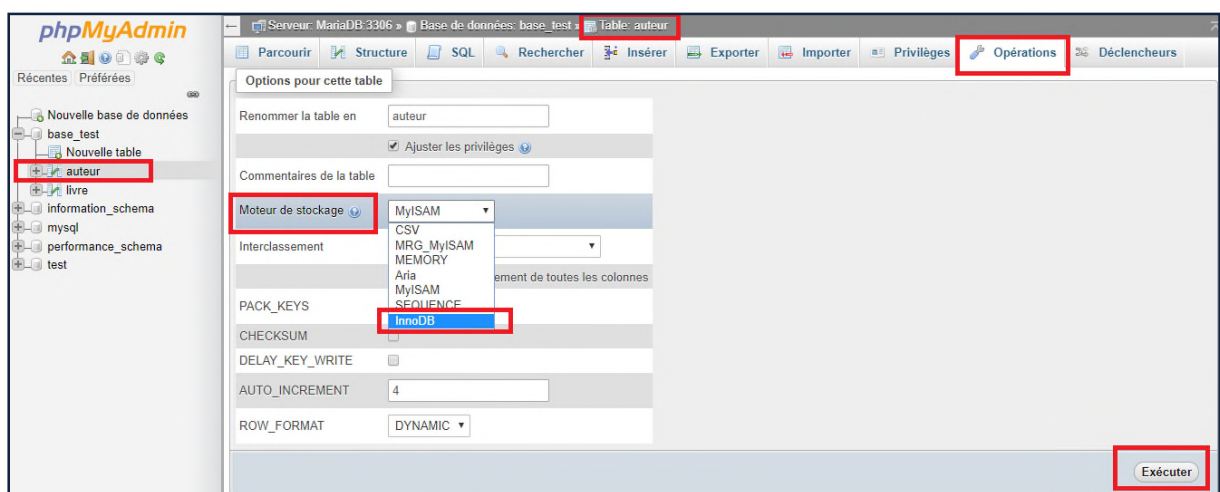
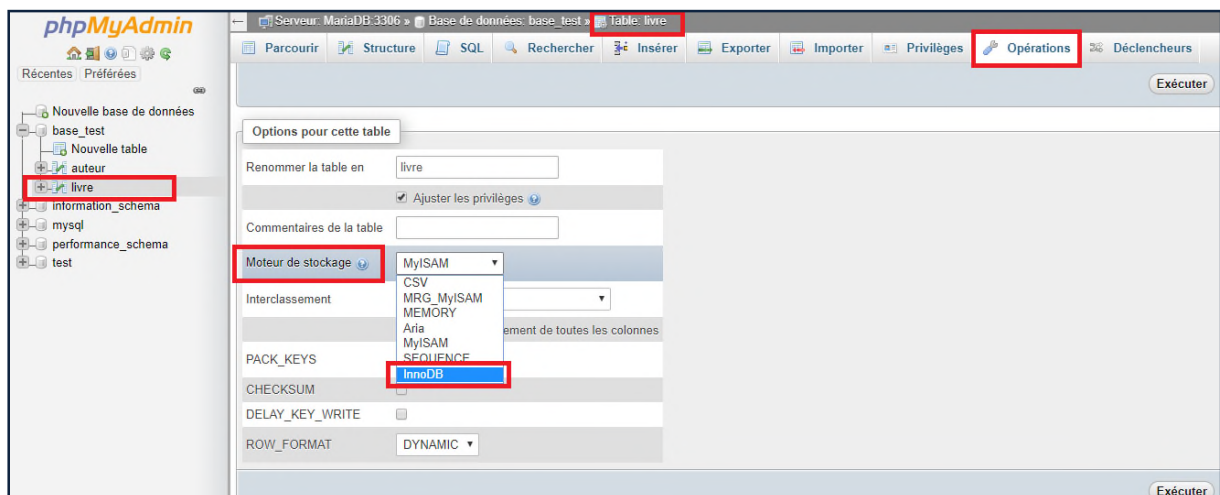
Avant d'exprimer les liens entre les tables, il faut s'assurer que les tables en question ont toutes un moteur de base de données « InnoDB ». Pour faire cette vérification, allez à la vue globale de la base de données dans laquelle ces tables sont créées et assurez-vous que la liste des tables



présentent dans le volet « Structure » de la vue globale de la base de données ont le moteur de stockage « InnoDB ».



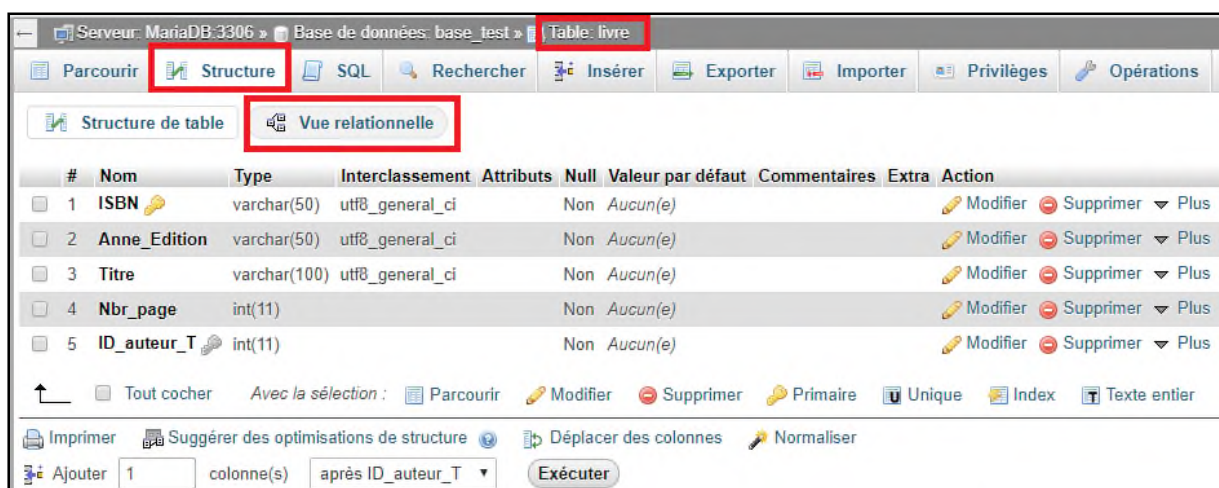
Dans le cas ou une table a le moteur de base de données MyISAM (le moteur de stockage MyISAM ne permet pas de créer des liens entres tables), allez à la vue global de la table en question, volet "opérations" et mettez le moteur de stockage utilisé à « InnoDB » et cliquer sur Exécuté :



Si on prend l'exemple des deux tables « Auteur » et « Livre », nous avons dans la table « Livre » un attribut Id\_Ateur qui fait référence à la table « Auteur » (rappelons que lors de la création de la table « Livre » nous avons mis le champ Index de l'attribut Id\_Ateur à la valeur INDEX pour

dire que cet attribut est une clé étrangère qui fait référence à la table « Auteur ». Nous rappelons également que la table « Auteur » comporte la clé primaire Id\_Ateur. Il est préférable de faire une légère différenciation entre l'attribut Id\_Ateur de la table « Livre » et l'attribut Id\_Ateur de la table « Auteur » à fin d'éviter certaines confusions possibles. Pour ce faire, nous ajoutons une petite lettre « L » à l'attribut Id\_Ateur pour faire allusion à la table « Livre ». De cette façon l'attribut Id\_Ateur\_L nous fait comprendre qu'on parle de la clé étrangère de la table « Livre », tant dis que l'attribut Id\_Ateur nous fait comprendre qu'on parle de la clé primaire de la table « Auteur ». Ceci dit cette légère différenciation ne change rien au fait que Id\_Ateur\_L et Id\_Ateur sont de même type et sont liés. Une seconde remarque importante à retenir est que même si l'attribut Id\_Ateur de la table « Auteur » est auto-incrémentale, l'attribut Id\_Ateur\_L de la table « Livre » ne doit jamais être auto-incrémentale pour la simple raison que le livre d'identifiant 1 peut être écrit par l'auteur d'identifiant 3245).

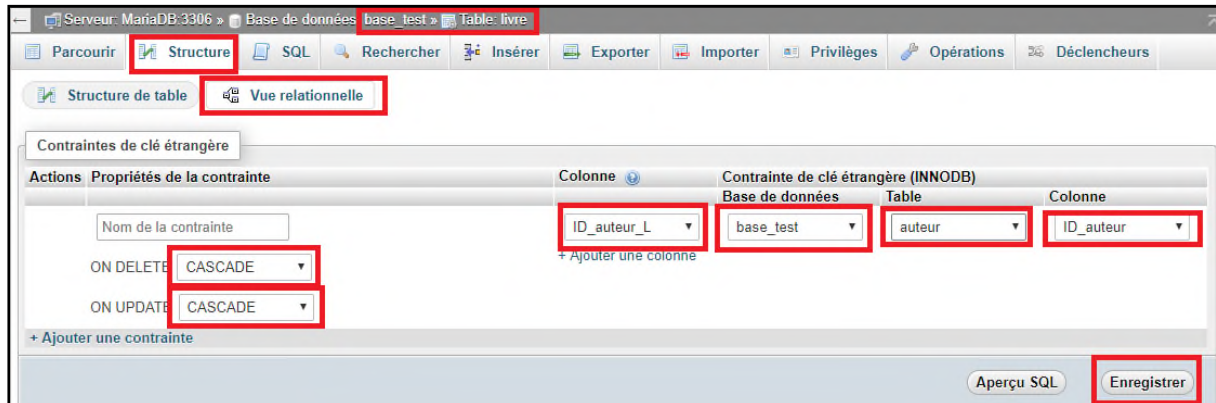
Revenons maintenant à notre objectif qui est de créer une relation entre la table « Livre » (qui comporte l'INDEX Id\_Ateur\_L) et la table auteur « Auteur ». Pour ce faire allez à la vue globale de la table qui comporte la clé étrangère (donc la table « Livre ») volet « Structure » et cliquez sur l'option « Vue Relationnelle » (le lien "vue relationnelle" n'apparaîtra pas sur l'interface du volet Structure de la table Livre si le moteur de base de données de la table en question n'est pas le moteur InnoDB. Donc si vous constatez que le lien de « vue relationnelle » du volet « Structure » de la table n'existe pas, pensez à changer le moteur de base de données).



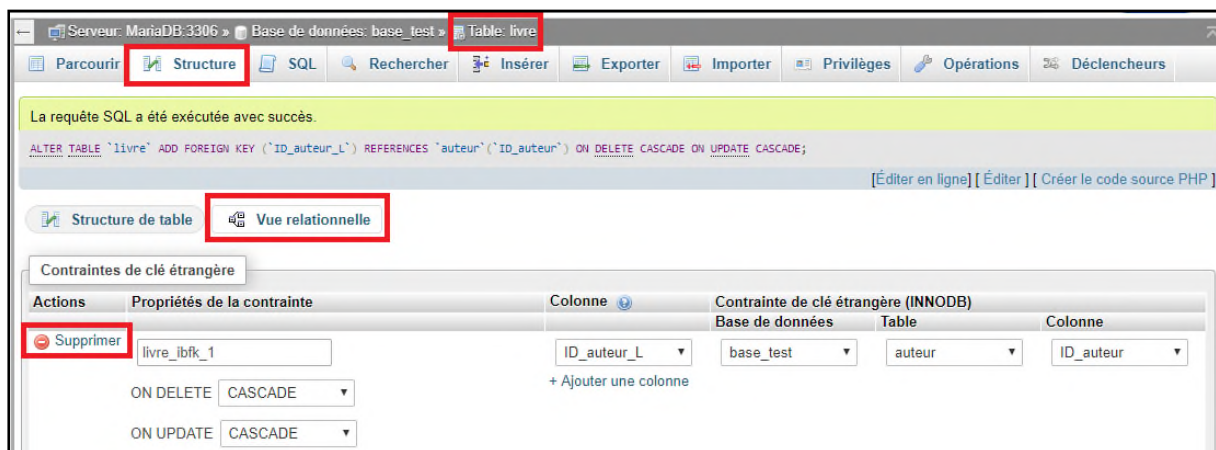
Dans la première liste déroulante « colonne » nous allons désigner le champ (ou l'attribut) qui appartient à la table « Livre » et qui permet de référencer la table « Auteur », ce champ est la clé étrangère « Id\_Ateur\_L » ; dans le champ « base de données », nous allons désigner la base de données ou les tables ont été créés (en d'autre termes désigner la base sur laquelle ont travail), puis nous allons désigner la table référencé avec laquelle nous souhaitons établir cette relation (il s'agit de la table « Auteur » qu'on veut lié à la table Livre). La seconde liste déroulante « colonne » désigne la clé primaire de la table « Auteur » qui permet de concrétiser cette liaison (et qui a été attribué à la table Livre comme clé étrangère pour permettre cette liaison).

Les deux options ON DELETE/ ON UPDATE permettent de définir le comportement des tables pour la relation. Ces deux options (ON DELETE/ ON UPDATE) lorsqu'elles sont à CASCAD, elles permettent d'établir (ou de provoquer) des réactions en chaine pour un changement établie sur l'une des deux tables liées. Pour comprendre la valeur CASCAD, on donne l'exemple de la suppression d'un auteur de la liste des auteurs, cette suppression engendre automatiquement la suppression de toute les livres liés à cet auteur (l'attribut Id\_Ateur\_L est lié à l'attribut Id\_Ateur

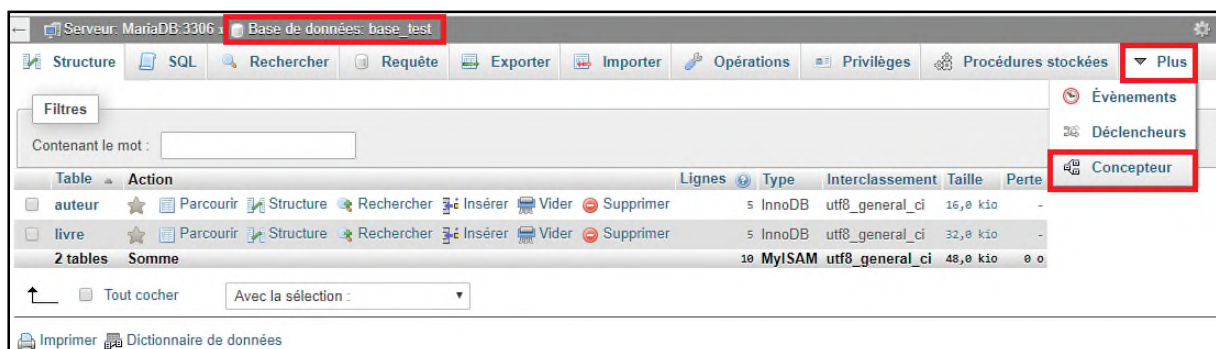
et il ne pourra pas avoir une valeur qui est absente dans la table Auteur. L'identification de clé primaire dans la table Livre permet la modification automatique de tuples dans la table Livre lorsque la clé primaire est modifiée ou effacée dans la table Auteur, c'est ce que nous appelons une suppression en cascade). Ainsi, pour déclencher des réactions en chaîne, nous mettons ces deux options de correspondances de données à CASCAD.



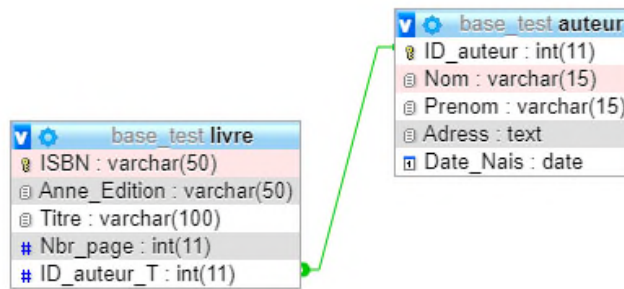
Cliquez ensuite sur « Enregistrer » pour valider cette liaison entre les deux tables Livre et Auteur. Pour supprimer cette relation, il suffit de sélectionner le choix « supprimer » :



Pour voir la concrétisation de cette liaison, cliquer sur le nom de la base de données, « Plus » puis « Concepteur » :



Vous remarquez qu'il existe réellement une liaison entre les deux tables « Auteur » et « Livre » :



Avec cette liaison et dans le cas ou vous insérez livre dont l'identifiant de l'auteur n'existe pas dans la table Auteur (c'est comme si vous dite que le livre est écrit par un auteur qui n'existe pas dans la liste des auteurs) alors une erreur SQL survient.