

Memento MYSQL LINUX

Pour installer le serveur Mysql

Installer les paquets suivants :

- Mysql-server (moteur de bases de données)
- Apache2 (serveur web si besoin)
- Php5 (langage de programmation permettant entre autre d'interroger un serveur mysql)
- Libapache2-mod-php5 (Plugin permettant la communication entre PHP et Apache)
- Php5-mysql (plugin permettant la communication entre PHP et Mysql)
- Openssh-server (permettant d'administrer à distance un serveur linux et donc mysql en console)
- Smbfs (eventuellement pour accéder à des ressources sur le réseau local)
- Php5-mcrypt (utile pour phpmyadmin)

Après cela il est utile d'installer phpmyadmin qui permet d'administrer le serveur mysql via interface web (plus convivial mais moins incomplet)

Pour cela

- Télécharger la dernière version sur le site de l'éditeur puis rapatrier cette dernière sur votre serveur (via le réseau local par exemple)
- Extraire le contenu du tar.gz dans le dossier /var/www
- Si besoin renommer le dossier phpmyadmin-3.4.xxxx-All-languages en phpmyadmin
- Relancer le service apache2 (/etc/init.d/apache2 restart)

A partir de là vous devez avoir une fenêtre de connexion lorsque vous saisissez l'url suivante dans votre navigateur préféré : http://ip_du_serveur/phpmyadmin :



Langue - Language

Français - French



Connexion

Utilisateur :

Mot de passe :

Exécuter

Si vous avez ceci c'est bon signe ! Continuons.

Quelques infos utiles :

Chemin par défaut des bases de données : /var/lib/mysql

Chemin du fichier de configuration : /etc/mysql/my.cnf

Nom du service : mysql ou mysqld

Commandes de base :

Connexion au serveur mysql :

```
mysql -u root -p
```

Enter password : (Saisir le mot de passe entré lors de l'installation de mysql)

Afficher les bases de données présentes :

```
show databases ;
```

Utiliser une base de données :

```
use nom_de_la_base ;
```

Créer une base de données :

```
create database nom_de_la_base ;
```

Créer une table :

```
create table nom_de_la_table (nom_du_champ1 type_de_champ1, nom_du_champ2  
type_de_champ2, nom_du_champ3 type_de_champ3, ... PRIMARY KEY  
(nom_du_champ_clé_primaire)) ;
```

Renommer une table :

```
alter table nom_de_la_table RENAME TO nouveau_nom_de_la_table ;
```

Modifier le nom d'un champ :

```
alter table nom_de_la_table CHANGE ancien_nom_de_champ nouv_nom_de_champ  
type_de_champ ;
```

Modifier le type d'un champ :

```
alter table nom_de_la_table MODIFY nom_de_champ type_de_champ ;
```

Ajouter un champ à une table :

```
alter table nom_de_la_table ADD nouv_champ type_de_champ ;
```

Déplacer un champ dans une table :

```
alter table nom_de_la_table MODIFY COLUMN champ_a type_de_champ AFTER champ_b ;
```

Supprimer un champ à une table :

```
alter table nom_de_la_table DROP nom_du_champ ;
```

Vider une table :

```
truncate table nom_de_la_table ;
```

Les types de champs :

Les champs numériques :

TINYINT	Entier très petit	1 octet
SMALLINT	Entier petit compris entre -32768 et 32767, si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 65535.	2 octets
MEDIUMINT	Entier moyen compris entre -8388608 et 8388607, si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 16777215	3 octets
INT	Entier standard compris entre -2 147 483 648 et 2 147 483 647. Si l'option UNSIGNED est utilisée, ce nombre sera compris entre 0 et 4 294 967 295	4 octets
BIGINT	Entier grand	8 octets
FLOAT	Décimal de simple précision	4 octets
DOUBLE, REAL	Décimal de double précision	8 octets
DECIMAL (entier,décimal)	Réel, définissez la longueur de chacune des deux parties.	variable

Les chaînes de caractères :

CHAR (n)	Chaîne de n caractère, taille fixe	
VARCHAR(M)	Chaîne de caractères variable. M peut être compris entre 1 et 255.	255 caractères. maximum
TINYBLOB, TINYTEXT	Petite zone de texte. Objet d'une longueur maximale de 255 caractères, TINYTEXT aura un contenu de type ASCII (casse insensible) et TINYBLOB aura un contenu de type binaire (casse sensible).	255 caractères. maximum
BLOB, TEXT	Zone de texte standard. Objet d'une longueur maximale de 65535 caractères, TEXT aura un contenu de type ASCII (casse insensible) et BLOB aura un contenu de type binaire (casse sensible).	65 535 caractères. maximum
MEDIUMBLOB, MEDIUMTEXT	Zone de texte moyenne. Objet d'une longueur maximale de 16777216 caractères, MEDIUMTEXT aura un contenu de type ASCII (casse insensible) et MEDIUMBLOB aura un contenu de type binaire (casse sensible).	16 millions caractères. maximum
LOB, LONGTEXT	Grande zone de texte. Objet d'une longueur maximale de 4294967295 caractères, LONGTEXT aura un contenu de type ASCII (casse insensible) et LOBBLOB aura un contenu de type binaire (casse sensible).	4 milliards caractères. maximum
ENUM('valeur','valeur2',...)	Une valeur parmi plusieurs. Objet texte qui ne peut avoir qu'une des valeurs 'valeur','valeur2',...	65535 valeurs max.
SET('valeur','valeur2',...)	Une ou plusieurs valeurs parmi plusieurs. Objet texte qui peut avoir une ou plusieurs des valeurs 'valeur','valeur2',...	64 valeurs max.

Les champs de Types date et heure :

DATE	Date (ex: 2000-08-24)	3 octets
TIME	Heure (ex: 23:44:05)	3 octets
DATETIME	Date et heure (ex: 2000-08-24 23:44:05)	8 octets
YEAR	Année (ex: 2000)	1 octet

Peupler une table :

Ajouter des données à une table :

```
insert into nom_de_la_table (champ1,champ2,champ3) values (valeur_champ1, valeur_champ2, valeur_champ3);
```

Attention : Si les valeurs de champs doivent être de type texte elles doivent être mises entre quotes (')

Importer un fichier texte avec champs séparés par des points virgules dans une table en vue de la remplir :

```
load data local infile 'chemin_du_fichier/fichier_texte' into table nom_de_la_table FIELDS TERMINATED BY ';' ;
```

ATTENTION : Pour que l'option fonctionne sur mysql server 5.5 ajouter l'option --local-infile lors de la connexion au serveur (mysql -u root -p --local-infile)

ATTENTION : Pour cette opération le fichier texte doit avoir la même structure que la table qui doit l'accueillir.

Modification du format des dates lors du chargement d'un fichier :

Si vous souhaitez importer un fichier texte comportant des dates au format français de type JJ/MM/AAAA, vous serez obligé de transformer ces dates au format AAAA/MM/JJ qui est celui de Mysql.

Si votre fichier source n'est pas très volumineux, il est possible de travailler sur celui-ci avec Excel.

Pour des bases plus conséquentes, la solution peut être de faire la modification lors de l'importation :

Il faut passer par la création d'une variable intermédiaire :

```
load data infile 'Adherents.txt' into table adherents fields terminated by ';' lines terminated by '\n'
```

```
(codead, civilite, nomad, prenom, adresse, cp, ville, numtel, Ici les champs que je ne touche pas
```

```
>@datenaiss, @datead), Ici les champs sur lesquels je souhaite faire une modification
```

```
>set Début de la déclaration des variables pour la modification du champ
```

```
>datenaiss = date_format(str_to_date(@datenaiss,'%d/%m/%Y'),'%Y/%m/%d'), Pour le champ
```

datenaiss je convertis le format d'origine d/m/Y en Y/m/d

```
>datead = date_format(str_to_date(@datead,'%d/%m/%Y'),'%Y/%m/%d'); Pour le champs datead je
```

convertis le format d'origine d/m/Y en Y/m/d

L'ensemble des lignes surlignées, peuvent s'écrire en une seule ligne

Exporter le résultat d'une requête vers un fichier texte :

```
select * into outfile '/tmp/fichier_export.txt' fields terminated by ';' from vente;
```

Dans cet exemple le fichier aura des champs séparés par des points virgule.

Gérer les droits d'accès à une base de données :

Lister les utilisateurs mysql :

```
use mysql ;  
select user, host from mysql.user ;
```

Ajouter un utilisateur pour mysql :

```
create user utilisateur identified by 'mot_de_passe';
```

Supprimer un utilisateur dans mysql :

```
drop user utilisateur ;
```

Changer le mot de passe d'un utilisateur mysql :

```
update mysql.user SET password = PASSWORD ('nouveau_mot_passe') where user = 'utilisateur' ;  
Flush privileges ;
```

Pour ajouter des droits :

```
grant type_privilège ON base_de_données.* to 'utilisateur'@'etendue_des_postes_distants'  
identified by 'mot_de_passe_de_utilisateur' ;
```

Il est possible d'ajouter les droits d'ajout de droits en ajoutant l'option WITH GRANT OPTION à la fin de la commande.

Pour l'étendue des postes, il s'agit là de spécifier quels sont les postes distants qui peuvent accéder au serveur mysql. Les options possibles sont variées telles que :

'%' : tous les postes

'192.168.10.5' : uniquement le poste ayant l'ip 192.168.10.5

'192.168.10.0/255.255.255.0' : Tous les postes ayant une IP dans le réseau 192.168.10.0

'%.mydom.dom' : Tous les postes ayant un nom dans le domaine mydom.dom

Pour supprimer des droits :

```
revoke type_privilège ON base_de_données.* FROM 'utilisateur';
```

ATTENTION : Pour accéder à une base de données depuis un autre poste voir un serveur web différent du serveur mysql, il faudra modifier le fichier /etc/mysql/my.cnf (fichier de configuration de mysql)

Pour cela il faudra adapter la ligne bind-address en fonction des besoins sinon la commenter.

Types de droits GRANT

Droit	Signification
ALL [PRIVILEGES]	Tous les droits sauf WITH GRANT OPTION .
ALTER	Autorise l'utilisation de ALTER TABLE .
CREATE	Autorise l'utilisation de CREATE TABLE .
CREATE TEMPORARY TABLES	Autorise l'utilisation de CREATE TEMPORARY TABLE .
DELETE	Autorise l'utilisation de DELETE .
DROP	Autorise l'utilisation de DROP TABLE .

EXECUTE	Autorise l'utilisateur à exécuter des procédures stockées (pour MySQL 5.0).
FILE	Autorise l'utilisation de SELECT ... INTO OUTFILE et LOAD DATA INFILE .
INDEX	Autorise l'utilisation de CREATE INDEX et DROP INDEX .
INSERT	Autorise l'utilisation de INSERT .
LOCK TABLES	Autorise l'utilisation de LOCK TABLES sur les tables pour lesquelles l'utilisateur a les droits de SELECT .
PROCESS	Autorise l'utilisation de SHOW FULL PROCESSLIST .
REFERENCES	Réservé pour le futur.
RELOAD	Autorise l'utilisation de FLUSH .
REPLICATION CLIENT	Donne le droit à l'utilisateur de savoir où sont les maîtres et esclaves.
REPLICATION SLAVE	Nécessaire pour les esclaves de réplication (pour lire les historiques binaires du maître).
SELECT	Autorise l'utilisation de SELECT .
SHOW DATABASES	SHOW DATABASES affiche toutes les bases de données.
SHUTDOWN	Autorise l'utilisation de mysqladmin shutdown .
SUPER	Autorise une connexion unique même si max_connections est atteint, et l'exécution des commandes CHANGE MASTER , KILL thread , mysqladmin debug , PURGE MASTER LOGS et SET GLOBAL .
UPDATE	Autorise l'utilisation de UPDATE .
USAGE	Synonyme de ``pas de droits``.
GRANT OPTION	Synonyme pour WITH GRANT OPTION

Administrer un serveur distant avec PHPMYADMIN.

En effet il peut être utile de vouloir administrer un serveur mysql sans pour autant installer apache et tous les outils liés au langage PHP.

Il faut donc commencer par renommer le fichier config.sample.inc.php présent à la racine du dossier phpmysqladmin en config.inc.php (ceci permettra l'utilisation de ce dernier)

Une fois renommé, il faut éditer ce fichier puis modifier la ligne

```
$cfg['Servers'][$i]['host'] = 'localhost';
```

Et la remplacer par

```
$cfg['Servers'][$i]['host'] = 'adresse_ip_du_serveur_mysql';
```

Il faut maintenant ajouter les droits du compte root mysql sur le serveur mysql afin qu'il soit possible de se connecter depuis un poste distant :

Dans mysql taper la commande :

```
grant all privileges on *.* to 'root'@'%' identified by 'mot_de_passe' with grant option;
```

Créer un lien ODBC entre Microsoft Access et Mysql

Il faut commencer par installer un connecteur ODBC, dans notre cas il s'appelle souvent MySQL Connector et il existe en version 32 ou 64 bits (choisir la version de votre suite office).

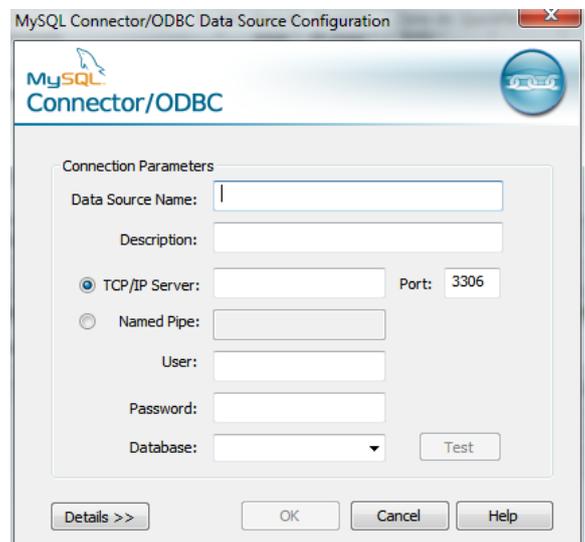
Dans Access, créer une nouvelle base de données, puis cliquer sur "Données Externes" puis base de données ODBC...

- Lier à la source de données en créant une table attachée
- Source de données machine
 - Nouveau
 - Source de données utilisateurs
 - MySQL ODBC Unicode driver

On arrive alors à cette fenêtre :

Dans le champ Data Source Name, il faut saisir un nom explicite pour cette connexion comme serveur_base.
Saisir l'adresse IP du serveur Mysql
Saisir l'utilisateur ayant les droits sur la base de données et son mot de passe.
Puis choisir la base de données.

Si aucune base n'apparaît, cliquer sur le bouton Test, si la connexion échoue, c'est certainement un problème d'accès au serveur mysql, il faut donc vérifier que le fichier my.cnf est bien configuré et que l'utilisateur spécifié a bien les droits sur la base depuis « n'importe où »



Sélectionner les tables désirées :



Et valider :

codead	civilite	nomad	prenom	adresse	codepos
1	Madam	MARIN	Anne	25 Rue des Fle	56740
2	Monsi	LE GOFF	André	9 Rue Amiral-C	56400
3	Monsi	LE GUEN	Denis	1 Rue Sablen	56400
4	Monsi	MALECK	Guy	3 Rue Ty Mad	56400
5	Madem	DURUEL	Véronique	13 rue Paul-Ga	56400
6	Monsi	GUEGUEN	Alphonse	35 Rue d'Anjou	35000
7	Madam	LUCAS	Elisabeth	3 Place de la p	56740
8	Monsi	BERLEAND	Jean-Luc	71 Bd de la Ré	35000

Les requêtes :

Les requêtes consistent à afficher les éléments d'une base de données selon un certain nombre de critères. Traduit en français une requête SQL pourrait s'écrire comme suit :

Sélectionne pour l'affichage les clients de la table ventes ayant acheté du service

En SQL :

```
select ventes.clients from ventes where libelle='service' ;
```

Il est possible d'utiliser les caractères génériques pour élargir le champ de recherche :

% : équivalent de *

_ : équivalent de ?

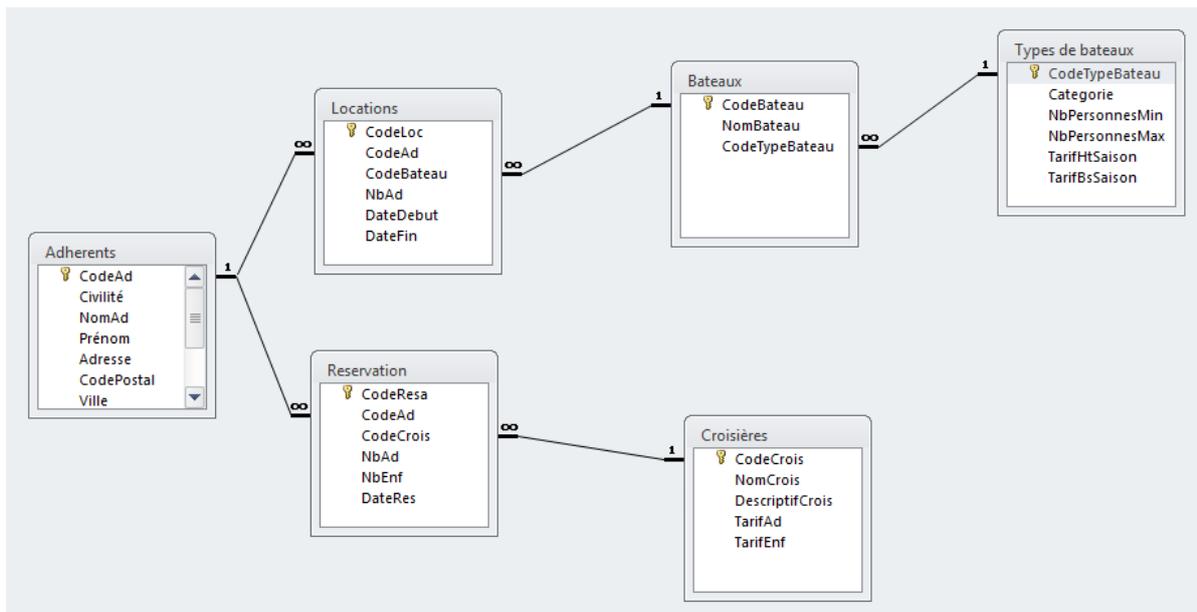
```
select ventes.clients from ventes where libelle like 's%' ;
```

Cette requête retournera les enregistrements clients de la table ventes ayant un libellé commençant par (like) un s

Si nous souhaitons trier les résultats par nom de clients :

```
SELECT ventes.clients from ventes where libelle='service' order by ventes.clients ;
```

Soit la structure de base de données suivante pour la base Auray



Si je souhaite afficher le nom, le prénom et la date de réservation que certains clients ont déjà effectués, j'ai besoin dans ce cas d'interroger 2 tables :

La table Adhérents dans laquelle se trouvent les champs nom et prénom des adhérents

La table Réservation dans laquelle se trouve le champ DateRes

On constate que les deux tables comportent un champ commun : CodeAd ce champs établi d'ailleurs une jointure entre les tables.

La requête permettant d'afficher les informations demandées devra donc faire appel à cette jointure.

Voici comment on écrira la requête :

```
SELECT Adhérents.NomAd, Adhérents.Prénom, Réservation.DateRes
FROM Adhérents INNER JOIN Réservation ON Adhérents.CodeAd=Réservation.CodeAd ;
```

Si maintenant je souhaite afficher en plus le nom de la croisière, je fait appel à une table de plus je dois donc imbriquer une jointure dans une jointure (inner join), j'obtiens donc la syntaxe suivante (âmes sensibles s'abstenir !) :

```
SELECT Adhérents.NomAd, Adhérents.Prénom, Croisières.NomCrois
FROM Croisières INNER JOIN (Adhérents INNER JOIN Réservation ON Adhérents.CodeAd =
Réservation.CodeAd) ON Croisières.CodeCrois = Réservation.CodeCrois;
```

Sauvegarde et restauration de bases de données MYSQL :

Sauvegarde d'une bdd :

```
Mysqldump -uroot -pmdp_root --databases nom_de_bdd > fichier_de_sauvegard_bdd.sql
```

Sauvegarde de toutes les bdd :

```
Mysqldump -uroot -pmdp_root -A > fichier_de_sauvegarde_globale.sql
```

Restauration d'une bdd à partir d'une sauvegarde d'une base :

```
Mysql -uroot -pmdp_root < fichier_de_sauvegarde_bdd.sql
```

Restauration d'une bdd à partir d'une sauvegarde globale :

Recréer manuellement au préalable (si nécessaire) la base de données

```
Mysql -uroot -pmdp_root --one-database base_a_restaurer < fichier_de_sauvegarde_globale.sql
```