

# Programmer une base de données

## Table des matières

1.	CONNEXION A UNE BASE DE DONNEES. ....	3
2.	UTILISATION D'UNE BASE DE DONNEES.....	3
2.1.	Créer un enregistrement .....	3
2.2.	Modifier un enregistrement.....	4
2.3.	Lecture d'un enregistrement et utilisation mysql_fetch.....	4
▪	Important. ....	4
2.4.	Suppression d'un enregistrement .....	4
2.5.	Récupérer le numéro du dernier enregistrement mysql_insert_id();.....	5
2.6.	Traitement de champ WHERE dans un SELECT.....	5
▪	Une expression .....	5
▪	Une variable .....	5
▪	Un champ .....	5
2.7.	COMPTE.....	5
▪	Compter tous les éléments d'un tableau COUNT .....	5
▪	Exemples : .....	5
▪	Faire la somme d'un champ SUM.....	5
▪	Faire la moyenne obtenue d'un champ AVG.....	5
▪	Connaître la meilleure moyenne obtenue d'un champ MAX ( <i>Peut être utiliser pour rechercher une valeur texte</i> ).....	6
▪	Connaître la plus petite moyenne obtenue d'un champ MIN.....	6
2.8.	Lecture dans deux tables simultanément.....	6
2.9.	COMPARER deux chaînes avec STRCMP .....	6
2.10.	TABLE TEMPORAIRE.....	6
3.	INTERROGATION PHP MYADMIN.....	7
3.1.	Connexion à la base de données.....	7
3.2.	Choisir et ouvrir une table.....	7
3.3.	Ouvrir l'onglet SQL. ....	7
3.4.	Autres exemples : .....	7
3.5.	Récupérer un fichier CLS.....	8
3.6.	Ouvrir avec Excel.....	9
3.7.	Récupération des inscrits d'un atelier. ....	10
3.8.	Récupération des inscrits d'une conférence. ....	10
3.9.	Récupération des inscrits suivant tarif .....	11
3.10.	Récupération des stagiaires impayés .....	11
3.11.	Récupération des stagiaires d'une année .....	11

3.12.	Récupération avec deux critères premier caractère et année.....	11
3.13.	Modifier la valeur d'un champ .....	11
3.14.	Recherche et classement par code postal.....	11
3.15.	Recherche de la date de modification des tables .....	11
4.	TABLE ACCESS TRANSFORME EN TABLE <i>MYSQL</i> .....	11
4.1.	Dans Access sauvegarder la table sous Excel. ....	11
4.2.	Sous Excel.....	12
4.3.	Sous <i>MYSQL</i> . ....	12
4.4.	Renommer à votre convenance. ....	13

## 1. CONNEXION A UNE BASE DE DONNEES.

Dossier : **include**

Fichier : **connexion.inc.php**

//Version localhost

```
<?php
//Nom d'utilisateur
$user="root";
//Autre exemple : $user « non d'utilisateur chez votre hébergeur »
//Nom de l'hôte de la base de données
$host="localhost";
//Autre exemple chez votre hébergeur, $host=" sql5 ";
//Nom du mot de passe
$password="";
//Autre exemple : $password="42424242";
//Active tous les warnings. Utile en phase de développement, en phase de production, remplacer E_ALL par 0
error_reporting(0);
//Nom de la base de données
$dbname="atelier_site2";
//Connexion au serveur Mysql
$connexion = mysqli_connect($host,$user, $password, $dbname) or die("Impossible de se connecter au serveur: ");
//Connexion à de la base de données
$_SESSION['connexion'] = $connexion; // sauvegarder la connexion dans une variable de session on a besoin
//Forcer l'écriture avec accents dans la base de données
mysqli_set_charset($connexion,"utf8");?>
```

//version NUXIT

```
<?php
//Nom d'utilisateur
$user="dechavanne";
//Autre exemple : $user « non d'utilisateur chez votre hébergeur »
//Nom de l'hôte de la base de données
$host="sql5";
//Autre exemple chez votre hébergeur, $host=" sql5 ";
//Nom du mot de passe
$password="2a[szGii";
//Autre exemple : $password="42424242";
//Active tous les warnings. Utile en phase de développement, en phase de production, remplacer E_ALL par 0
error_reporting(0);
//Nom de la base de données
$dbname="dechavanne";
//Connexion au serveur Mysql
$connexion = mysqli_connect($host,$user, $password, $dbname) or die("Impossible de se connecter au serveur: ");
//Connexion à de la base de données
$_SESSION['connexion'] = $connexion; // sauvegarder la connexion dans une variable de session
// Change caractère pour utf8
mysqli_set_charset($connexion,"utf8");
?
```

**PHP 5:**

```
//Connexion au serveur Mysql
$connexion=mysqli_connect($host,$user,$password) or die('Impossible de se connecter au serveur: ' . mysqli_error());
//Connexion à de la base de données
$db=mysqli_select_db($dbname,$connexion) or die ("Erreur de sélection de la base de données");
```

## 2. UTILISATION D'UNE BASE DE DONNEES

### 2.1. Créer un enregistrement

INSERT INTO **table** (id,nom) VALUES ('', 'Dupond')

Exemple :

```
$$sql = "INSERT INTO recettes (Prenom,Titre,Familles,Descriptions,Image,Personnes,Ingredients,Observations)
VALUES('$Prenom','$Titre','$Familles','$Descriptions','$Image','$Personnes','$Ingredients','$Observations)";
```

```
mysqli_query($_SESSION['connexion'],$sql) or die ("Exécution requête impossible.sql :".mysqli_error($_SESSION['connexion'])."(__FILE__)");
```

## 2.2. Modifier un enregistrement

```
UPDATE Clients_tbl SET prenom='Jacques' WHERE id=1
```

Mise à jour de la table **Clients\_tbl**

Modifier le champ **prenom** pour la valeur **Jacques** **SET prenom='Jacques'**

Quand le champ **id** est égal à **1** **WHERE id=1**

Exemple :

```
$set = prenom= 'paul',nom='duvers',adresse='12 rue machin',codepostal=42100' ;
```

```
$sql="UPDATE membres SET ".$set." WHERE id=".$svisiteur;
```

```
mysqli_query($_SESSION['connexion'],$sql) or die ("Exécution requête impossible.sql
: ".mysqli_error($_SESSION['connexion'])."(__FILE__)");
```

## 2.3. Lecture d'un enregistrement et utilisation mysqli\_fetch\_array

```
//Ouverture de l'enregistrement sélectionné
```

```
$req="SELECT * FROM recettes WHERE ID_recettes=$ID_recettes";
```

```
//Si non défini (ouverture de la page avant le choix) ne pas afficher de message d'erreur
```

```
$res= mysqli_query($_SESSION['connexion'],$req) or die ("Exécution requête impossible.sql
: ".mysqli_error($_SESSION['connexion'])."(__FILE__)");
```

```
$infos=mysqli_fetch_array($res);
```

```
$Titre=$infos[Titre];
```

## Lecture d'un enregistrement et utilisation mysqli\_fetch\_row

- Important.

(mysqli\_fetch\_row() retourne un tableau à indices numériques. L'ordre de ce tableau (ici \$resultat[]) est donné par l'ordre de sélection des champs dans la requête. Ici le premier champ sélectionné est mail, ensuite signature et ensuite config.

```
//Ouverture de l'enregistrement sélectionné
```

```
$req="SELECT * FROM recettes WHERE ID_recettes=$ID_recettes";
```

```
//Si non défini (ouverture de la page avant le choix) ne pas afficher de message d'erreur
```

```
$res= mysqli_query($_SESSION['connexion'],$req) or die ("Exécution requête impossible.sql
: ".mysqli_error($_SESSION['connexion'])."(__FILE__)");
```

```
("__FILE__");
```

```
$infos= mysqli_fetch_row ($res);
```

```
$Titre=$infos[5];
```

**mysqli\_fetch\_array()** est légèrement moins rapide que **mysqli\_fetch\_row()** sur certains calculs. Elle retourne un tableau avec le nom des champs sélectionnés comme indice.

## Suppression d'un enregistrement

```
DELETE FROM table WHERE id=1
```

Effacer l'enregistrement de la table

Quand l'id de l'enregistrement est égal à **1**

## Récupérer le numéro du dernier enregistrement `mysql_insert_id()`;

```
En php5 : $id=mysql_insert_id();
$id= mysql_insert_id($_SESSION['connexion']);
echo "login : $id ";
```

### Variante.

```
$result = mysqli_query("SELECT ID FROM benevoles ORDER BY ID DESC LIMIT 0,1");
$final = mysqli_fetch_array($result);
echo $final["ID"];
```

## Traitement de champ WHERE dans un SELECT.

- Une expression

```
"SELECT cheque FROM comptes WHERE cheque='SG'");
WHERE dates='2014-2015'
```

- Une variable

```
"SELECT cheque FROM comptes WHERE cheque='$cheque'");
```

- Un champ

```
"SELECT * civilite FROM benevoles WHERE SUBSTR(dates, -4)='$choix' GROUP BY civilite";
```

## COMPTER

On sélectionne **SELECT** une table et un champ (ou la totalité avec **\***)

- Compter tous les éléments d'un tableau **COUNT**

```
//NOMBRE DE RECETTES CONCERNEES
```

```
//On selectionne une table et des champs
```

```
$selection="SELECT COUNT(*) AS ID_recettes FROM recettes where $phrase order by Titre,
Titre ASC";
```

```
// on crée la requête SQL
```

```
$requete= mysqli_query($_SESSION['connexion'],$selection) or die ("Exécution requête
impossible.sql :".mysqli_error($_SESSION['connexion'])."
```

```
("._FILE_.")");
```

```
// $data est un tableau associatif
```

```
while ($data= mysqli_fetch_array($requete)) {
```

```
// les éléments sont « indicés » par les noms
```

```
$nbs_recettes=$data['ID_recettes'];
```

```
}
```

Exemples :

```
SELECT codepostal, COUNT(*) FROM membres GROUP BY codepostal ORDER BY codepostal
ASC
```

```
SELECT codepostal, COUNT(*) FROM membres GROUP BY codepostal ORDER BY COUNT(*)
DESC
```

```
SELECT commune,codepostal, COUNT(*) FROM membres GROUP BY codepostal ORDER BY
COUNT(*) DESC
```

```
SELECT commune,codepostal, COUNT(*) FROM membres WHERE dernier='2014-2015' GROUP BY
codepostal ORDER BY COUNT(*) DESC
```

```
SELECT commune,codepostal, COUNT(*)AS total FROM membres WHERE dernier='2014-2015'
GROUP BY codepostal ORDER BY total DESC
```

- Faire la somme d'un champ **SUM**

```
SUM(reglement)AS total
```

- Faire la moyenne obtenue d'un champ **AVG**

```
AVG( moyenne ) AS moyenneclasse FROM etudiant ;
```

- **Connaître la meilleure moyenne obtenue d'un champ MAX** (*Peut être utiliser pour rechercher une valeur texte*)

**MAX**(comptes.observations)AS comptes\_tot

- **Connaître la plus petite moyenne obtenue d'un champ MIN**

**MIN**(comptes.observations)AS comptes\_tot

### Lecture dans deux tables simultanément.

**SELECT** les champs que tu veux récupérer

Toujours préciser les colonnes retourner par la recherche, c'est vrai pour chacune de vos requêtes « SQL » mais ça l'est davantage avec une jointure. Non seulement le "**SELECT \* FROM**" peut devenir très lent, particulièrement si votre table contient de longs textes, mais la confusion peut devenir ingérable voir impossible avec plusieurs tables qui contiennent des colonnes de mêmes noms.

Profitez-en pour ajouter le nom de la table. Un bon petit truc pour optimiser vos requêtes est de placer les textes le plus à droite possible dans la requête.

**FROM** la TABLE que tu veux interroger

**INNER JOIN** la table à laquelle elle est liée

Jointure par défaut qui compare deux tables et retourne tous les enregistrements comportant une concordance.

Exemple **SELECT ... FROM** <table de gauche> **INNER JOIN** <table de droite> **ON** <Condition de jointure>

```
$rqSql = "SELECT comptes.adherent, membres.id, membres.nom, membres.prenom FROM comptes INNER JOIN membres ON comptes.adherent=membres.id ";
```

```
$result = mysql_query($rqSql) or die("Exécution requête impossible.");
```

**WHERE** les criteres de filtre des résultats

```
while ($ligne= mysql_fetch_array($result)) {
```

\$lignes est un tableau associatif

Les éléments sont « indicés » par les noms

Des colonnes. Je préfère cette technique à celle

Des indices numériques. On ajoute une colonne.

```
$nom=$ligne["adherent"];
```

```
$nom1=$ligne["id"];
```

```
$nom2=$ligne["nom"];
```

```
$nom3=$ligne["prenom"];
```

```
echo "N°adherent sur table comptes : " . $nom;
```

```
echo "<br>";
```

```
echo "N°adherent sur table membres:" . $nom1;
```

```
echo "<br>";
```

```
echo "Nom de l'adhérent : " . $nom2;
```

```
echo "<br>";
```

```
echo "Prenom de l'adhérent : " . $nom3;
```

```
echo "<br>";
```

### COMPARER deux chaines avec STRCMP

```
$sql = "SELECT ID,ville FROM ville";
```

```
$req = mysql_query($sql) or die('Erreur SQL !<br>'.$sql.'<br>'.mysql_error());
```

```
while($infos=mysql_fetch_row($req)) {
```

La ligne est retournée sous la forme d'un tableau. Chaque colonne est enregistrée sous la forme d'un tableau commençant à la position 0.

```
$infos=mysql_fetch_row($req);
```

```
$identifiant=$infos[0];
```

```
$ville=$infos[1];
```

```
$choix="Roanne";
```

```
if (strcmp($choix, $ville) == 0) {
```

```
    echo '<b>'. $identifiant.' </b> ('.$ville.);    } }
```

### TABLE TEMPORAIRE.

Une table temporaire est une table qui n'existe que dans la session qui l'a créée.

Pour créer une table temporaire, on peut utiliser tout simplement **CREATE TABLE** en ajoutant **TEMPORARY**, pour préciser qu'il s'agit d'une table temporaire.

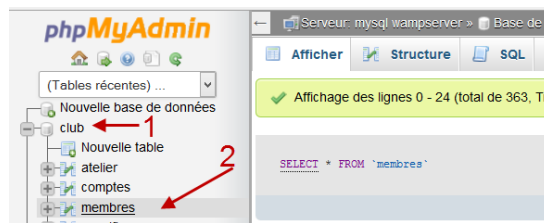
```
$sql_temporaire = " create temporary table commande_temp  
as  
select *  
from comptes  
WHERE dates='2014-2015' ";  
$rep_temporaire = mysql_query($sql_temporaire)  
or die ('Erreur sql_temporaire:$sql_temporaire'.mysql_error())
```

## INTERROGATION PHP MYADMIN

### Connexion à la base de données.

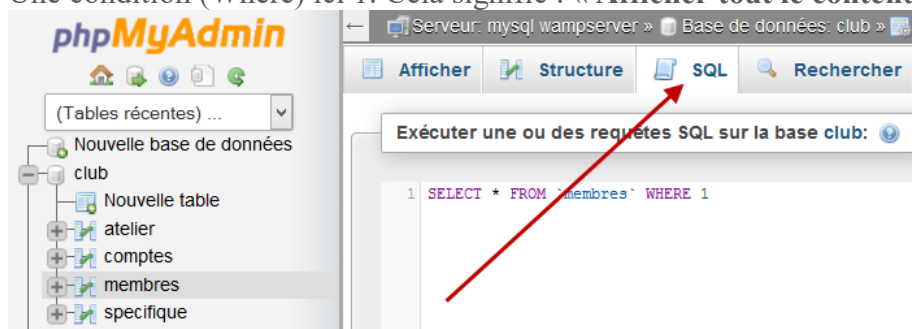
```
-- INFORMATIONS MYSQL --  
Hôte de connexion : sql4  
Nom d'utilisateur : clubinformatiqu  
Nom de la base : clubinformatiqu  
Mot de passe : qnSTFMkr  
Accès PhpMyAdmin : http://phpmyadmin.club-informatique-spj.fr
```

### Choisir et ouvrir une table.

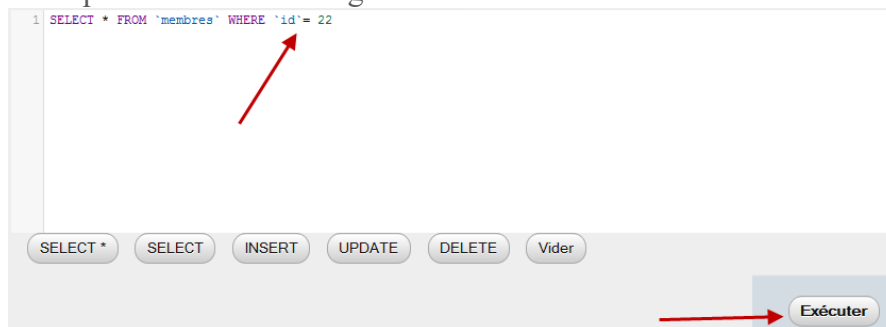


### Ouvrir l'onglet SQL.

Par défaut tous les champs de la table (ici membres) sont sélectionnés.  
Une condition (Where) ici 1. Cela signifie : « **Afficher tout le contenu de la table 'membres'** »



Exemple rechercher l'enregistrement 22 :



Autres exemples :

- Rechercher les animateurs.  
Cliquez sur Animateur dans la liste des champs, ajouter la condition entre guillemets (Ici le mot oui)

1 SELECT \* FROM 'membres' WHERE 'Animateur'='oui'

Colonnes  
id  
courriel  
nom  
prenom  
adresse  
codepostal  
commune  
telephone  
portable  
fonction  
CA  
Animateur  
Resp\_atelier

[ Délimiteur : ]  Afficher à nouveau la requête après exécution  Conserver la boîte de requêtes **Exécuter**

- Rechercher les adhérents avec une adresse email.  
Sélectionnez la zone avec \* puis cliquez sur le champ à rechercher ici courriel

1 SELECT \* FROM 'membres' WHERE 'courriel'

Colonnes  
id  
courriel  
nom  
prenom  
adresse  
codepostal  
commune  
telephone  
portable  
fonction  
CA  
Animateur  
Resp\_atelier

[ Délimiteur : ]  Afficher à nouveau la requête après exécution  Conserver la boîte de requêtes **Exécuter**

Pour une meilleure lisibilité on peut ajouter les champs nom et prénom.

1 SELECT 'nom','prenom','courriel' FROM 'membres' WHERE 'courriel'

Colonnes  
id  
courriel  
nom  
prenom  
adresse  
codepostal  
commune  
telephone  
portable  
fonction  
CA  
Animateur  
Resp\_atelier

[ Délimiteur : ]  Afficher à nouveau la requête après exécution  Conserver la boîte de requêtes **Exécuter**

Rechercher les membres sans courriel et non null.

1 SELECT 'nom','prenom','courriel' FROM 'membres' WHERE 'courriel'=''

Colonnes  
id  
courriel  
nom  
prenom  
adresse  
codepostal  
commune  
telephone  
portable  
fonction  
CA  
Animateur  
Resp\_atelier

[ Délimiteur : ]  Afficher à nouveau la requête après exécution  Conserver la boîte de requêtes **Exécuter**

Rechercher les membres avec courriel.

Exécuter une ou des requêtes SQL sur la base club: [?](#)

1 SELECT \* FROM 'membres' WHERE 'courriel' is not null

Colonnes  
id  
courriel  
nom  
prenom  
adresse  
codepostal  
commune  
telephone  
portable  
fonction  
CA  
Animateur  
Resp\_atelier

[ Délimiteur : ]  Afficher à nouveau la requête après exécution  Conserver la boîte de requêtes **Exécuter**

## Récupérer un fichier CLS.

En bas de la liste sélectionnée choisir Exporter



DE LAVERGNE	Carmen	cetavernicarmen@gmail.com
ROUCHON	Rosette	rosette.rouchon@gmail.com
DE MORI	Daniel	danieldomori@orange.fr

Nombre de lignes : 500

**Opérations sur les résultats de la requête**

Version imprimable Version imprimable (avec textes complets) Exporter Afficher le graphique Créer une vue

Puis CVS

**Méthode d'exportation :**

- Rapide - n'afficher qu'un minimum d'options
- Personnalisée - afficher toutes les options p

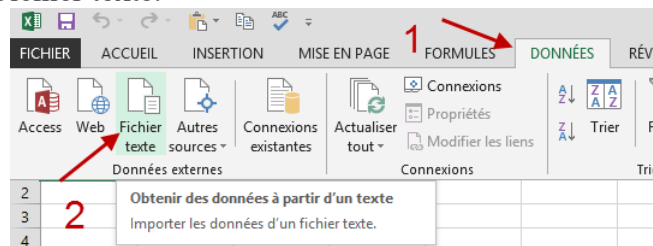
**Format :**

- SQL
- CodeGen
- CSV
- CSV for MS Excel
- Microsoft Word 2000
- JSON

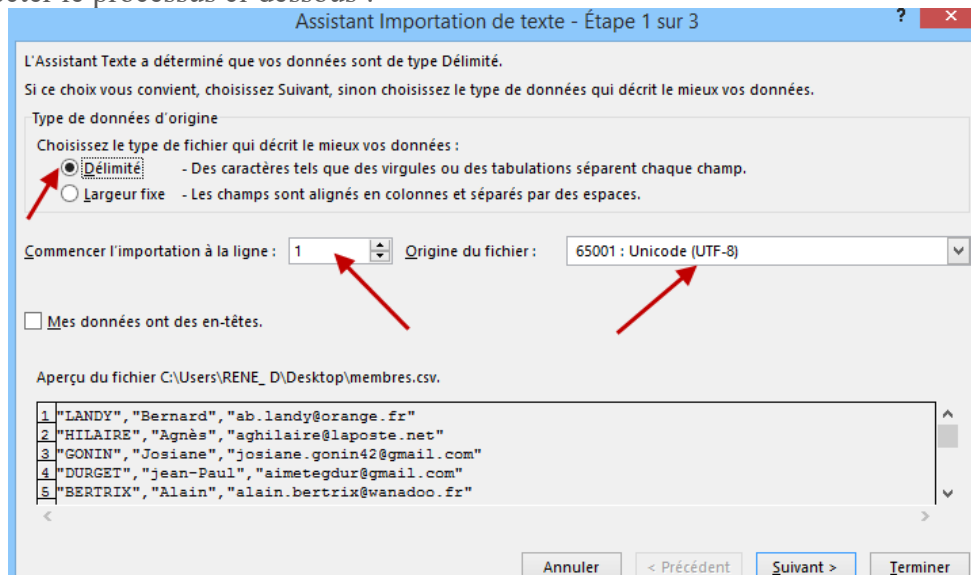
Puis enregistrer sur votre ordinateur.

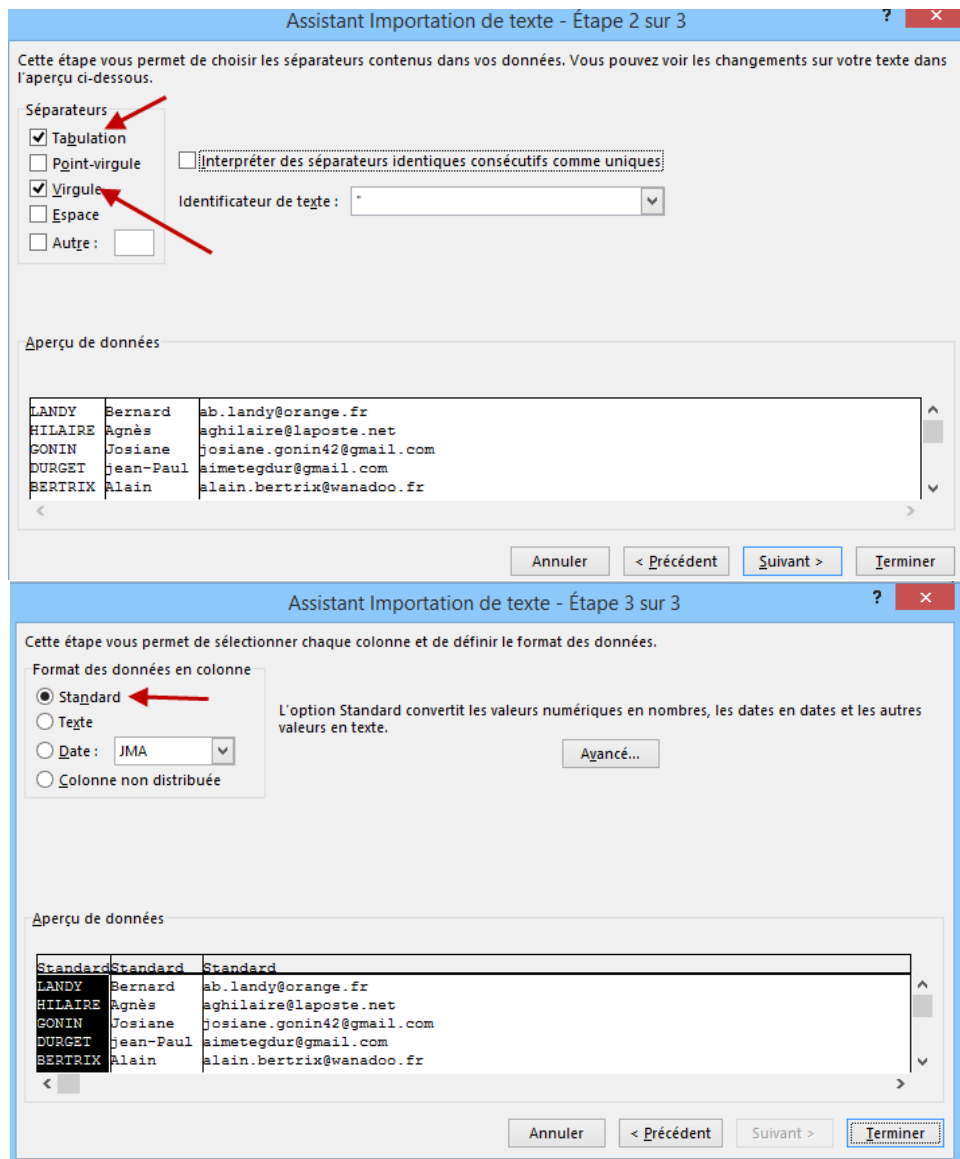
**Ouvrir avec Excel.**

Choisir Données puis Fichier texte.



Bien respecter le processus ci-dessous :





Sauvegarder avec l'option :

CSV (séparateur: point-virgule) (\*.csv)

### Récupération des inscrits d'un atelier.

```
SELECT membres.id, membres.nom, membres.prenom, membres.courriel, comptes.ateliers10
FROM comptes
INNER JOIN membres
ON comptes.adherent= membres.id
WHERE substr(comptes.ateliers10, 1, 4) = 'RP01'
ORDER BY membres.nom ASC
```

### Récupération des inscrits d'une conférence.

```
SELECT membres.id, membres.nom, membres.prenom, membres.courriel, comptes.ateliers10 FROM comptes
INNER JOIN membres ON comptes.adherent= membres.id WHERE substr(comptes.ateliers10, 1, 6) = 'C01-01' ORDER BY membres.nom ASC
```

## Récupération des inscrits suivant tarif

```
SELECT membres.id, membres.nom, membres.prenom, membres.courriel, comptes.reglement
FROM comptes INNER JOIN membres ON comptes.adherent= membres.id WHERE
reglement='90' OR reglement='155' ORDER BY membres.nom ASC
```

OU

```
SELECT membres.id, membres.nom, membres.prenom, membres.courriel, comptes.reglement
FROM comptes INNER JOIN membres ON comptes.adherent= membres.id WHERE
reglement>'70' ORDER BY membres.nom ASC
```

## Récupération des stagiaires impayés

### Liste des adhérents avec enregistrement de la formule de paiement :

```
SELECT membres.id, membres.nom, membres.prenom, comptes.observations,
comptes.ateliers10, membres.dernier FROM comptes INNER JOIN membres ON comptes.adherent=
membres.id WHERE substr(comptes.ateliers10, 1, 6) = 'Formul' AND dates='2016-2017' ORDER BY
membres.nom ASC
```

### Puis listes des adhérents de l'année :

```
SELECT `nom`, `prenom` FROM `membres` WHERE `dernier`='2016-2017' ORDER BY membres.nom ASC
```

## Récupération des stagiaires d'une année

```
SELECT `courriel` FROM `membres` WHERE `dernier`='2015-2016'
```

## Récupération avec deux critères premier caractère et année.

```
SELECT * FROM `atelier` WHERE `exercice`='2016-2017' AND substr(atelier,1,6)= 'C17-01'
```

```
SELECT * FROM `enregistrements` WHERE substr(id_asso,1,16)= 'ASSOCIATION POUR' ORDER BY `id_asso` ASC
```

## Modifier la valeur d'un champ

Remplacer la valeur 14 par 2014

```
UPDATE `cotisations` SET annee_cotisation='2014' Where annee_cotisation='14'
```

```
UPDATE `cotisations` SET `remarque`='Soldé' Where `remarque`='7'
```

## Recherche et classement par code postal

```
SELECT commune, codepostal, COUNT(*) AS total FROM membres WHERE dernier='2015-
2016' GROUP BY codepostal ORDER BY total DESC
```

## Recherche de la date de modification des tables

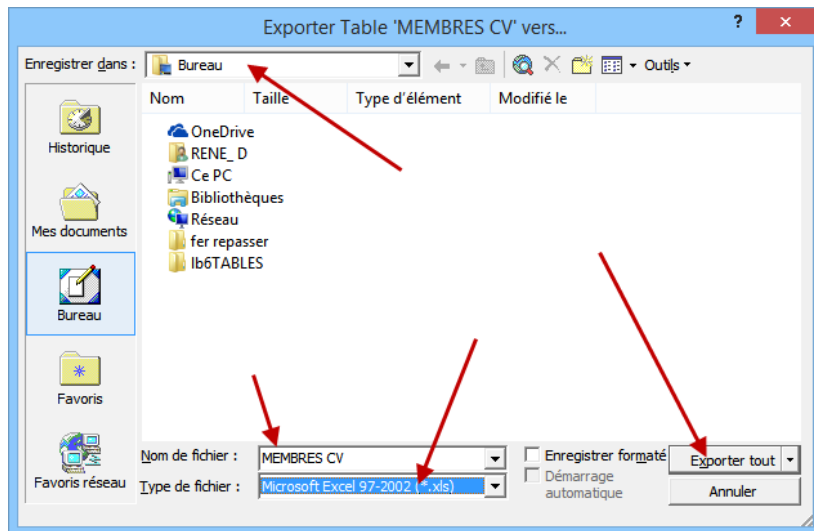
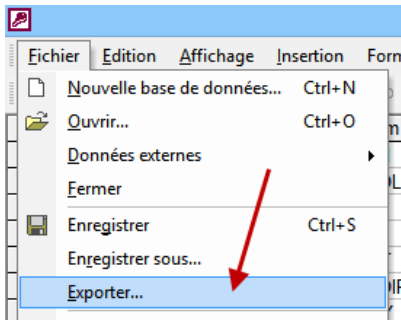
```
SHOW TABLE STATUS
```

Au 13/01/2016

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Max_data_length	Index_length	Data_free	Auto_increment	Create_time	Update_time	Check_time	Collation	Checksum	Create_options	Comment
atelier	InnoDB	10	Compact	78	210	16384		0	0	93	2015-12-12 14:45:49		NULL	utf8mb4_unicode_ci	NULL		
comptes	MyISAM	10	Dynamic	1075	86	92712	281474976710655	16384	0	1152	2015-10-17 10:49:01	2016-01-13 08:15:12		latin1_swedish_ci	NULL		comptes
membres	MyISAM	10	Dynamic	466	143	66868	281474976710655	34816	0	477	2015-04-09 17:22:51	2016-01-13 06:41:29	2015-04-12 12:58:39	latin1_swedish_ci	NULL		Table des membres
specifique	InnoDB	10	Compact	1	16384	16384		0	0	2	2015-12-12 14:45:49		NULL	latin1_swedish_ci	NULL		

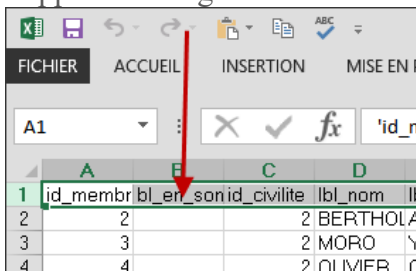
## TABLE ACCESS TRANSFORME EN TABLE MYSQL

Dans Access sauvegarder la table sous Excel.

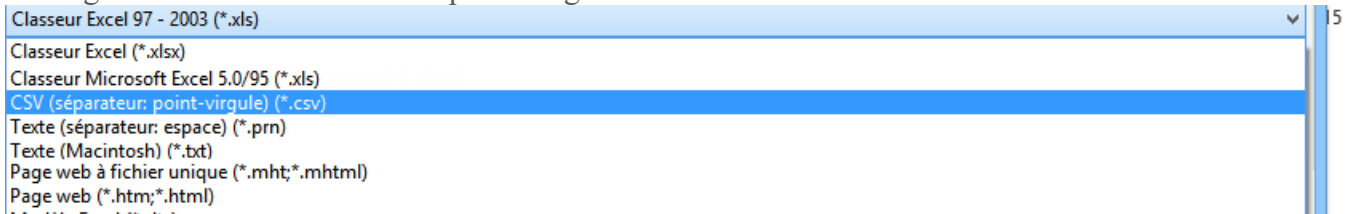


## Sous Excel.

Supprimer la ligne des identifiant.

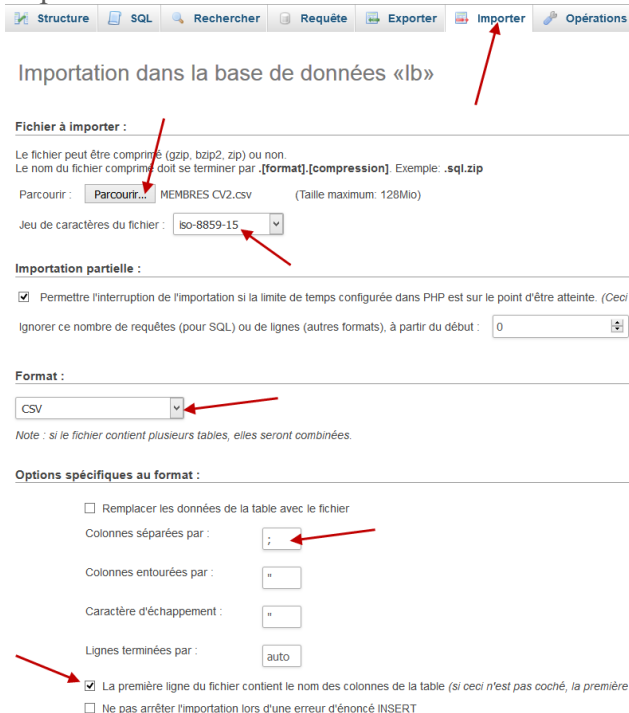


Sauvegarde un fichier en csv avec point virgule



## Sous MYSQL.

Importer la table et renommer à votre convenance.



## Renommer à votre convenance.

The screenshot displays the phpMyAdmin interface for a MySQL server. The left sidebar shows a tree view of databases and tables, with a red arrow pointing to 'table 3' under the 'ib' database. The main area features several panels: 'Nouvelle table' (New table) with fields for 'Nom' (Name) and 'Nombre de colonnes' (Number of columns); 'Renommer la base de données comme suit:' (Rename database as follows:) with a text input containing 'membres' and an 'Exécuter' (Execute) button; 'Supprimer la base de données' (Delete database) with a red arrow pointing to the 'Opérations' tab and a red arrow pointing to the 'Exécuter' button; and 'Copier la base de données vers:' (Copy database to:). The 'Copier la base de données vers:' panel includes radio buttons for 'Structure seule', 'Structure et données', and 'Données seulement', and checkboxes for 'Faire CREATE DATABASE avant la copie', 'Ajouter DROP TABLE / DROP VIEW', 'Inclure la valeur courante de l'AUTO\_INCREMENT', and 'Inclure les contraintes de clés étrangères'. The top navigation bar includes tabs for 'Structure', 'SQL', 'Rechercher', 'Requête', 'Exporter', 'Importer', 'Opérations', 'Privilèges', 'Procédures stockées', 'Événements', and 'Déclarer'.