



PROGRAMME

- Historique
- MySQL
- Installation
- Base de donnée
- Un peu de pratique
- SQL
- Un peu de pratique
- Projet

Un peu d'histoire ...



Créé 23 mai 1995 (date de sortie)



David Axmark

Son nom vient du prénom de la fille du co-créateur Michael Widenius (My)

SQL fait référence au *Structured Query Language*, le langage de requête structurée.



Michael Widenius

MySQL





Qu'est ce que MySQL ?



MySQL est un système de gestion de bases de données relationnelles (SGBDR). qui utilise le langage SQL

— Système de gestion de base de donnée

MySQL

PostgreSQL

SQLite

Oracle

Microsoft SQL server



Mais avant tout !

C'est un SGBD (système de gestion de base de données)



— Qu'est ce qu'une base de donnée ?



Une base de données permet de stocker et de retrouver des données structurées

Base de données relationnelle ?



Une **base de données relationnelle** est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés des *relations* ou *tables*

		0	1	2
0	20	20	16	14
1	17	17	21	15
	un dimension		deux dimensions	

exemple :

ID	prenom	nom	tel
1	Florent	Dugommier	01 44 77 21 33
2	Patrick	Lejeune	03 22 17 41 22
3	Michel	Doussand	04 11 78 02 00



Qu'est ce que le langage SQL ?



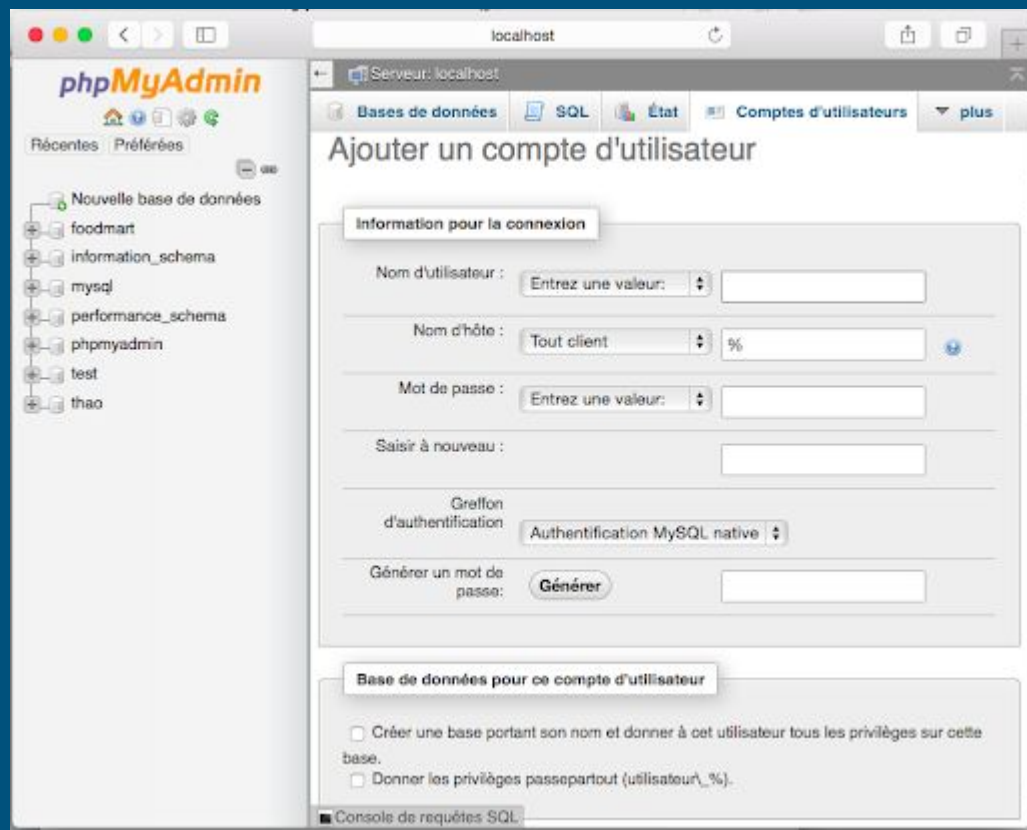
Langage de requête qui sert à exploiter les BDD relationnelles



Qu'est ce que phpMyAdmin ?



c'est un outil qui a été développé avec php et qui va vous permettre de gérer votre base de donnée.



il existe plusieurs logiciels

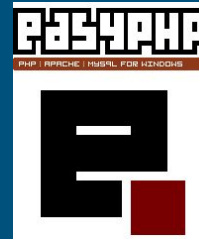
XAMPP



WAMP



EASYPHP



MAMP
(mac)



Installation



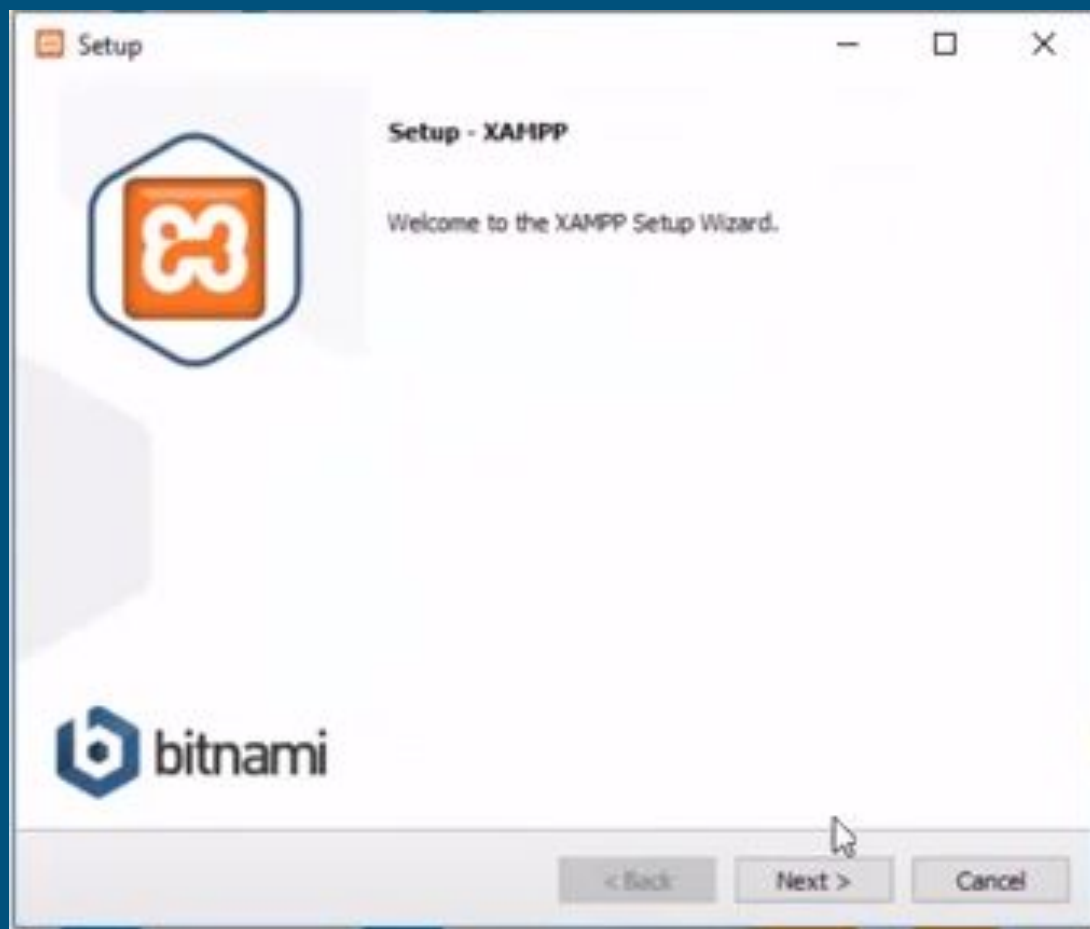


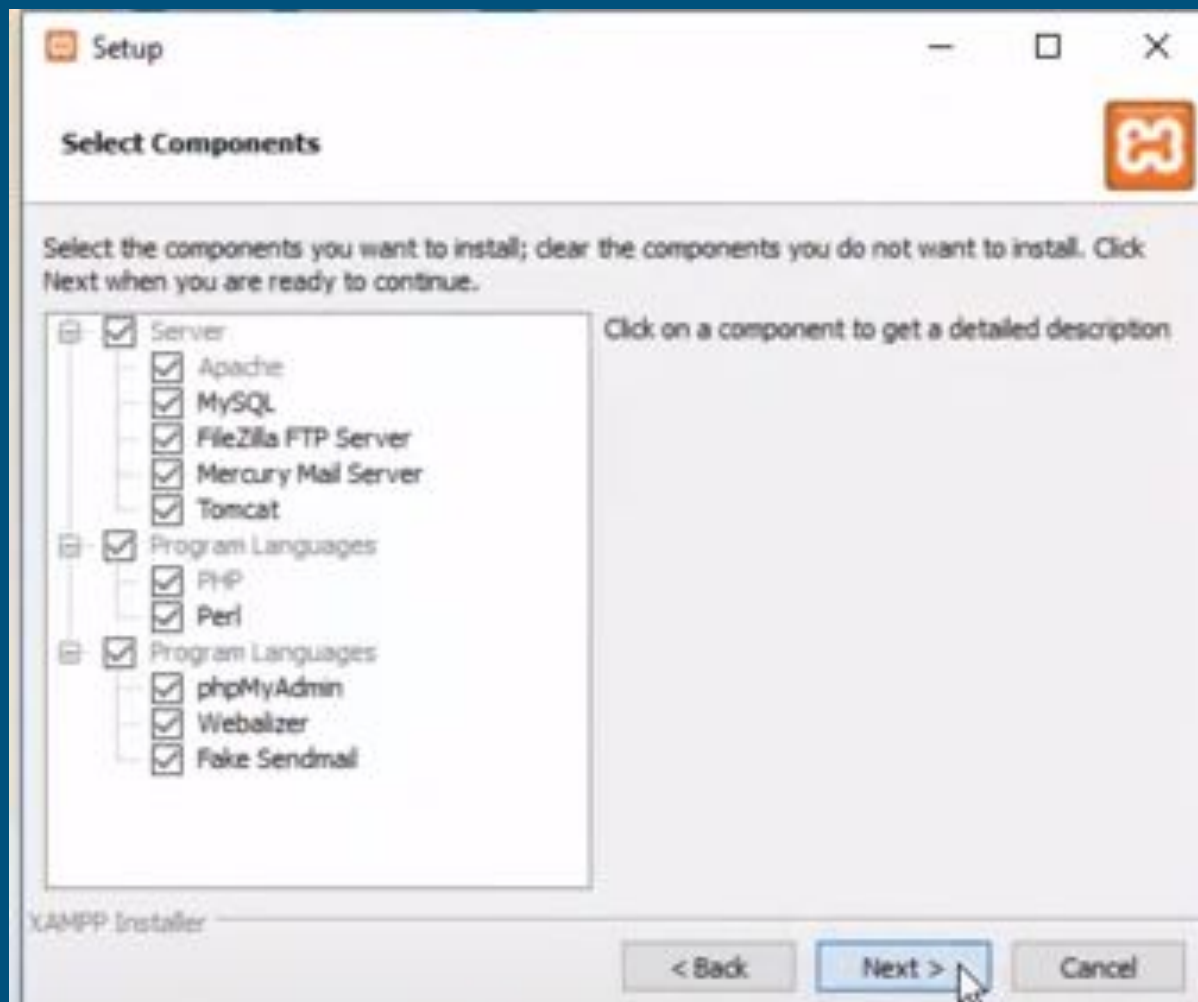
Installation

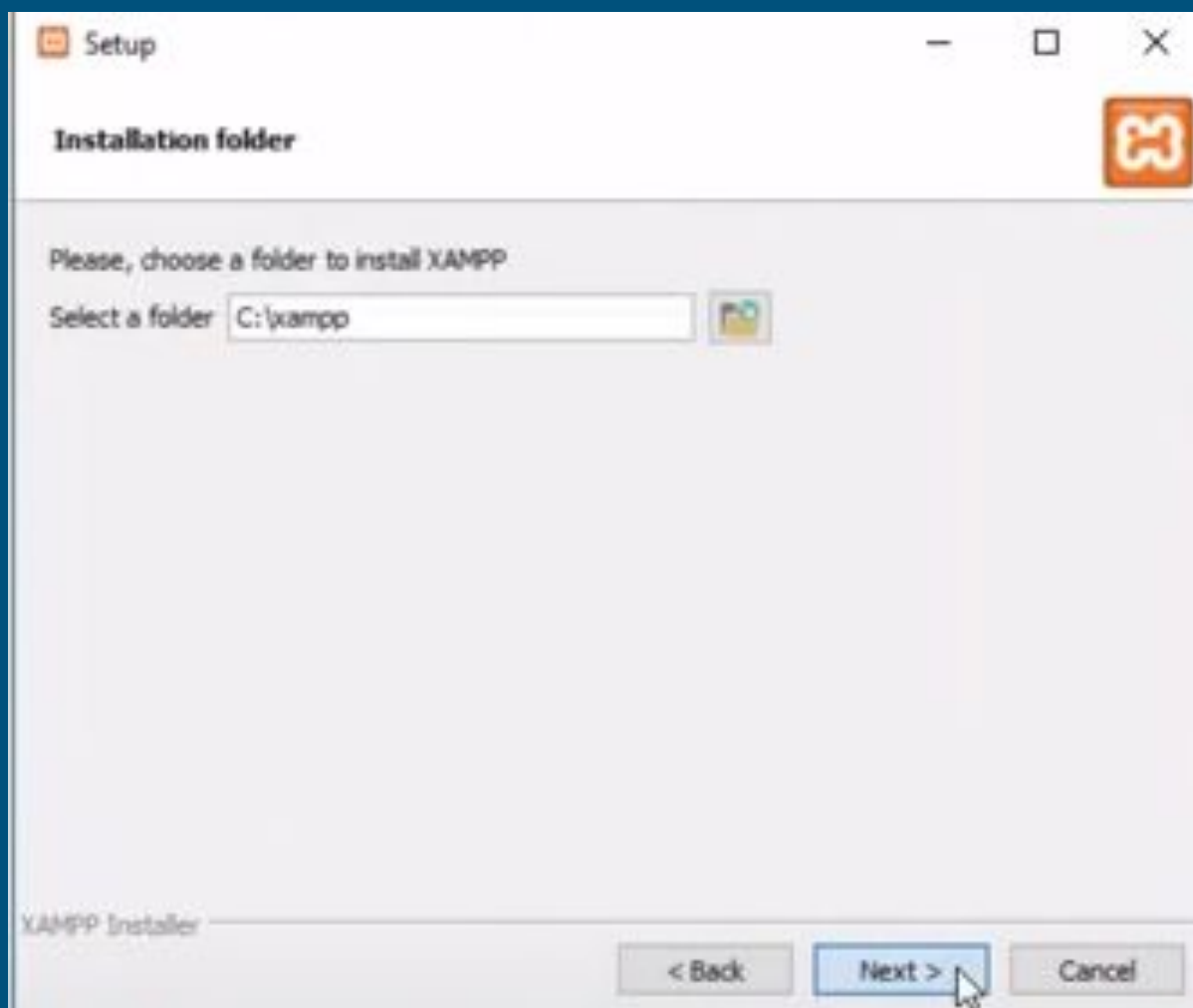
Installation d'un serveur web local



<https://www.apachefriends.org/fr/download.html> (dernière version)

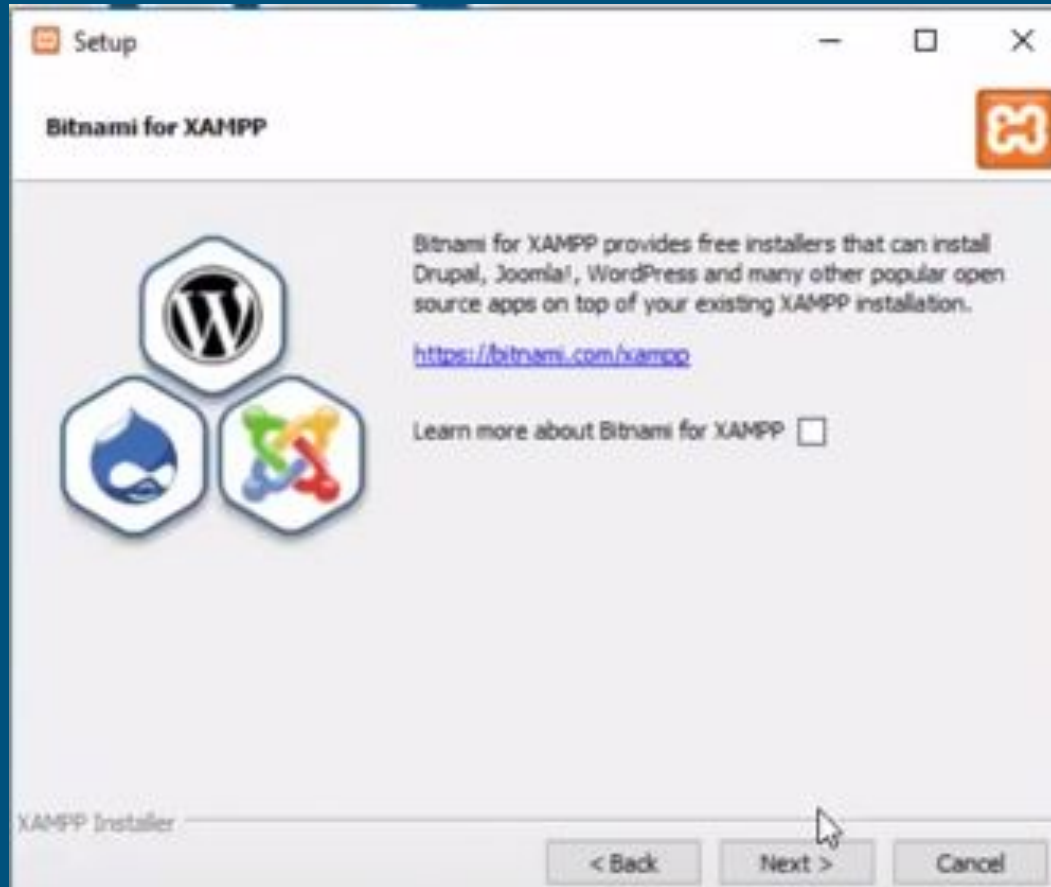


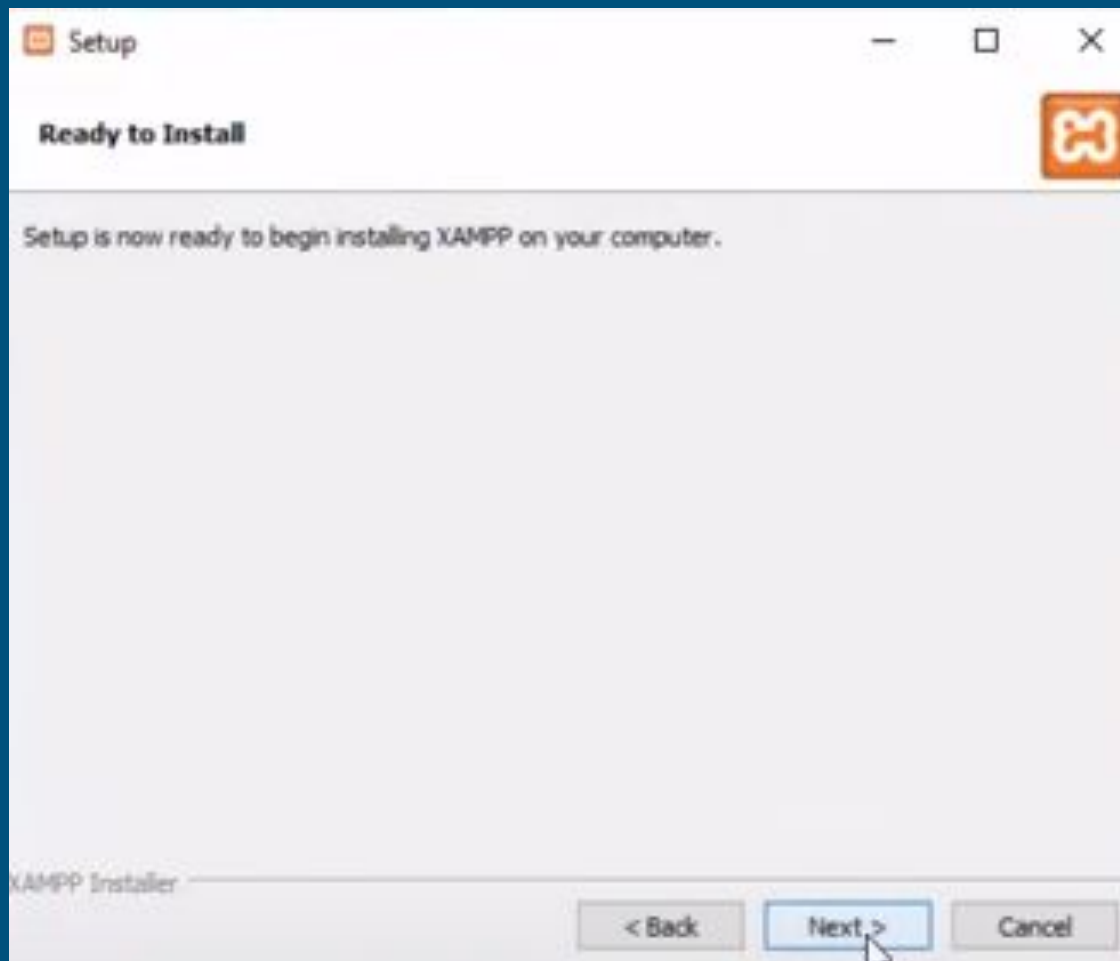


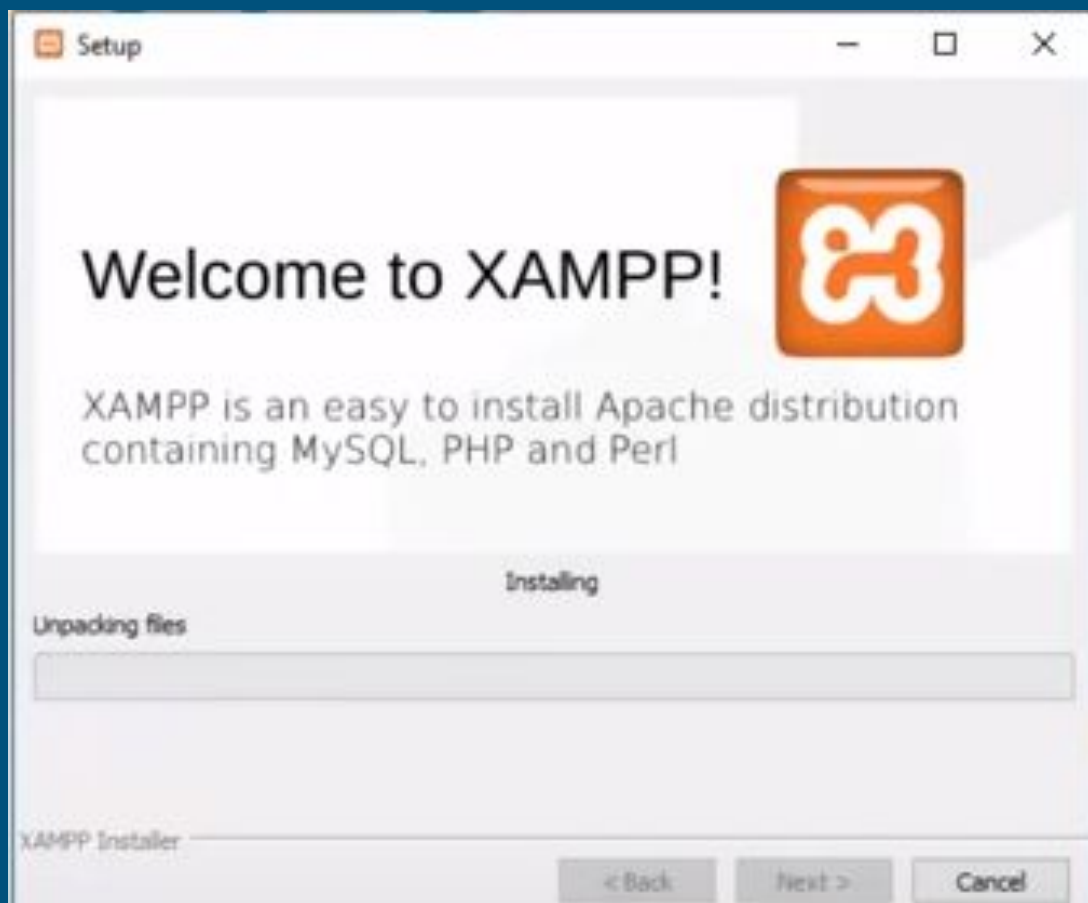


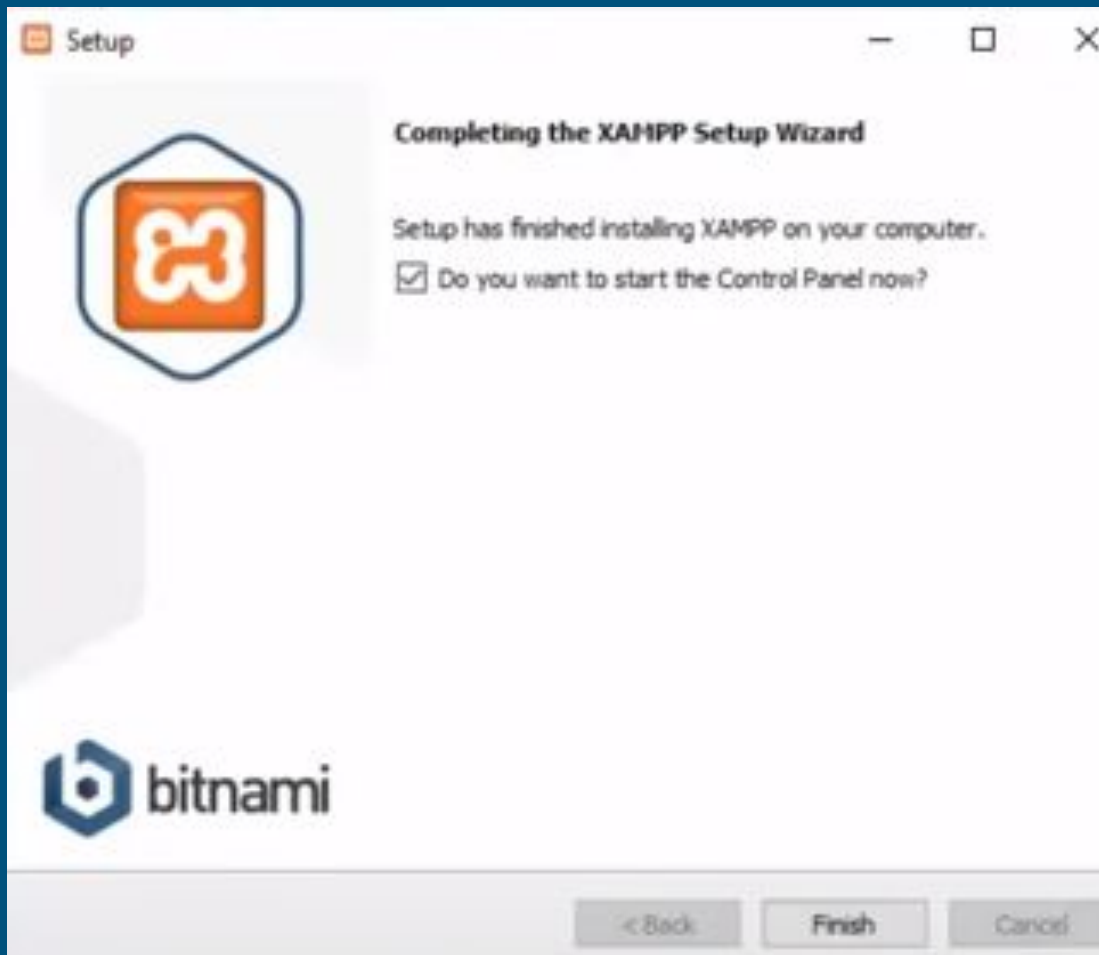


Décocher la croix

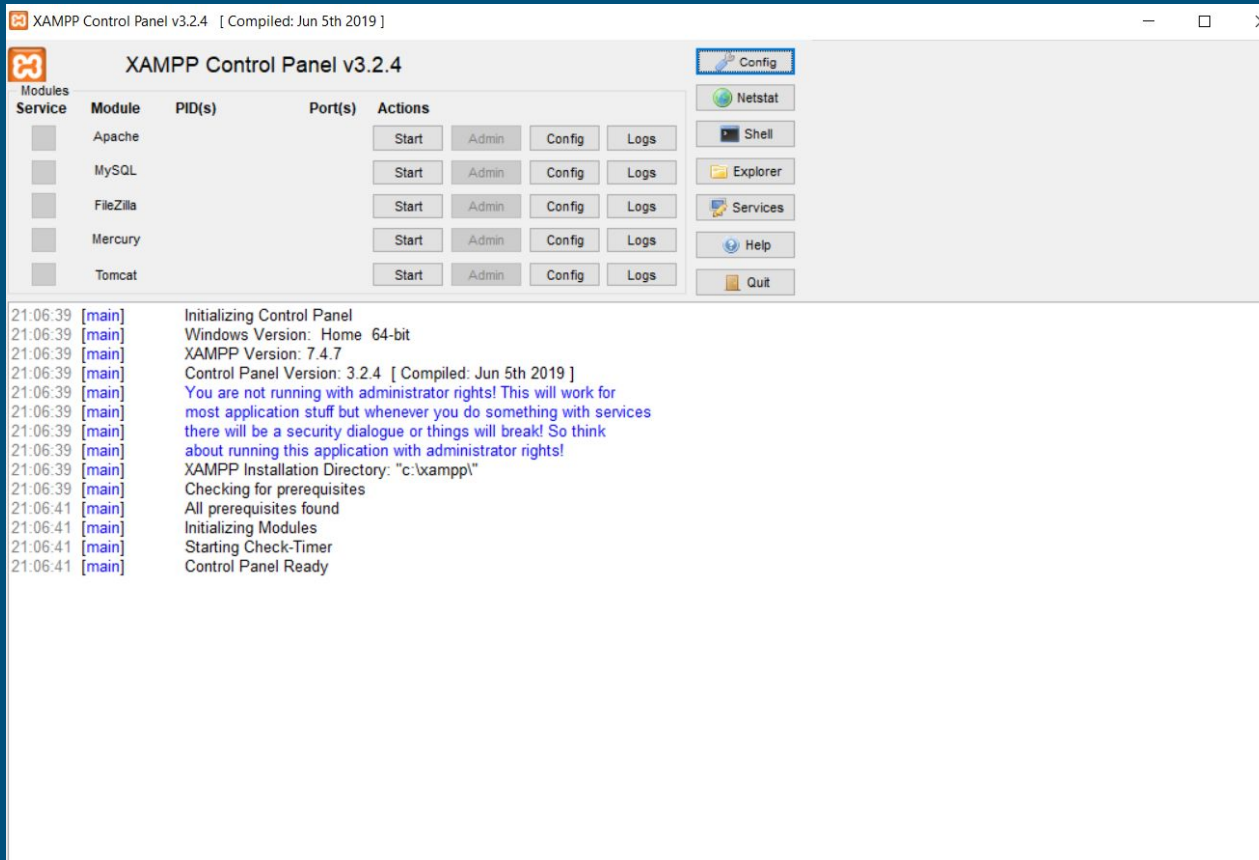








Le contrôle panel



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

21:06:39 [main] Initializing Control Panel
21:06:39 [main] Windows Version: Home 64-bit
21:06:39 [main] XAMPP Version: 7.4.7
21:06:39 [main] Control Panel Version: 3.2.4 [Compiled: Jun 5th 2019]
21:06:39 [main] You are not running with administrator rights! This will work for
21:06:39 [main] most application stuff but whenever you do something with services
21:06:39 [main] there will be a security dialogue or things will break! So think
21:06:39 [main] about running this application with administrator rights!
21:06:39 [main] XAMPP Installation Directory: "c:\xampp\
21:06:39 [main] Checking for prerequisites
21:06:41 [main] All prerequisites found
21:06:41 [main] Initializing Modules
21:06:41 [main] Starting Check-Timer
21:06:41 [main] Control Panel Ready

Configuration

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

Config

Netstat
Shell
Explorer
Services
Help
Quit

```
21:06:39 [main] Initializing Control Panel
21:06:39 [main] Windows Version: Home 64-bit
21:06:39 [main] XAMPP Version: 7.4.7
21:06:39 [main] Control Panel Version: 3.2.4 [ Compiled: Jun 5th 2019 ]
21:06:39 [main] You are not running with administrator rights! This will work for
21:06:39 [main] most application stuff but whenever you do something with services
21:06:39 [main] there will be a security dialogue or things will break! So think
21:06:39 [main] about running this application with administrator rights!
21:06:39 [main] XAMPP Installation Directory: "c:\xampp\"
21:06:39 [main] Checking for prerequisites
21:06:41 [main] All prerequisites found
21:06:41 [main] Initializing Modules
21:06:41 [main] Starting Check-Timer
21:06:41 [main] Control Panel Ready
```

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Modules

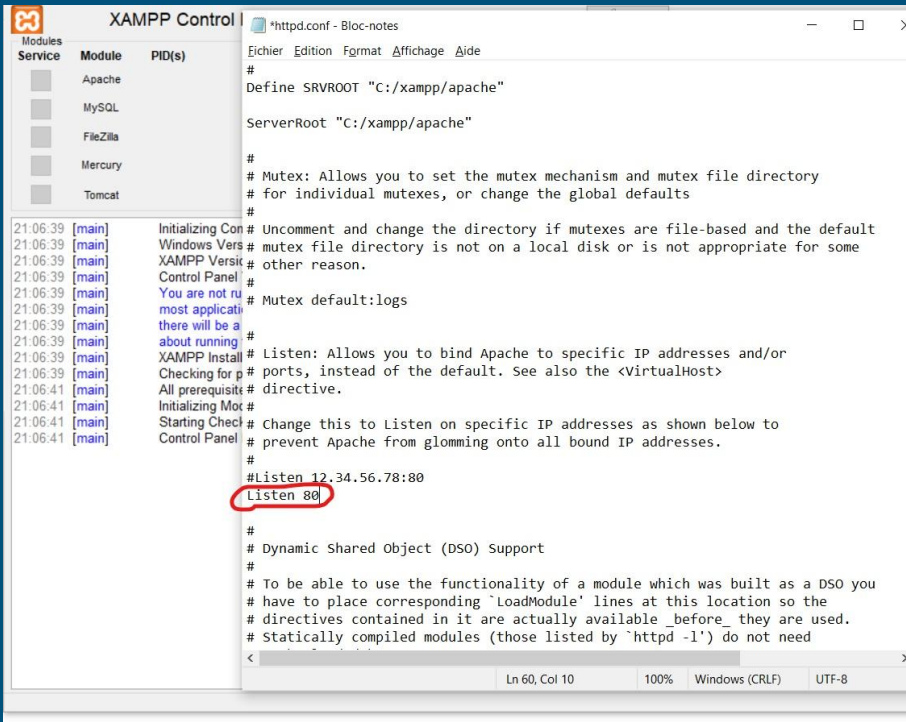
Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config
<input type="checkbox"/>	FileZilla			Start Admin Config
<input type="checkbox"/>	Mercury			Start Admin Config
<input type="checkbox"/>	Tomcat			Start Admin Config

Config Netstat Shell

- Apache (httpd.conf)
- Apache (httpd-ssl.conf)
- Apache (httpd-xampp.conf)
- PHP (php.ini)
- phpMyAdmin (config.inc.php)
- <Browse> [Apache]
- <Browse> [PHP]
- <Browse> [phpMyAdmin]

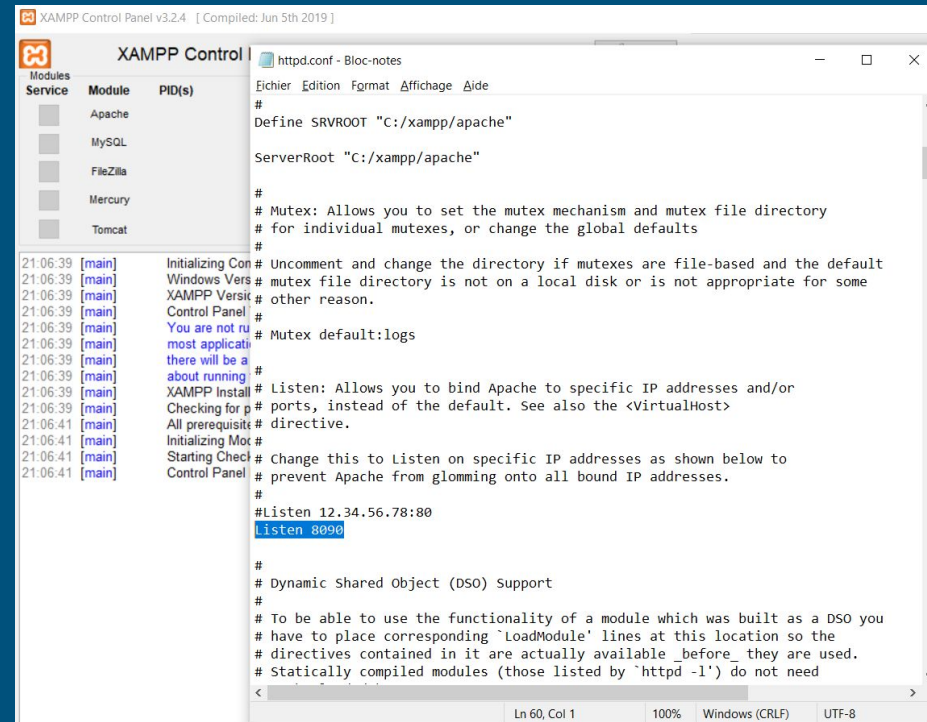
21:06:39 [main] Initializing Control Panel
21:06:39 [main] Windows Version: Home 64-bit
21:06:39 [main] XAMPP Version: 7.4.7
21:06:39 [main] Control Panel Version: 3.2.4 [Compiled: Jun 5th 2019]
21:06:39 [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!
21:06:39 [main] XAMPP Installation Directory: "c:\xampp\
21:06:39 [main] Checking for prerequisites
21:06:41 [main] All prerequisites found
21:06:41 [main] Initializing Modules
21:06:41 [main] Starting Check-Timer
21:06:41 [main] Control Panel Ready

Descendre : changer le Listen 80 → Listen 8090



The screenshot shows the XAMPP Control Panel on the left and the httpd.conf file in a text editor on the right. The Control Panel lists the status of various services: Apache, MySQL, FileZilla, Mercury, and Tomcat. The httpd.conf file is open to the section for listening on connections. The line `#Listen 12.34.56.78:80` is highlighted with a red circle, and the status bar at the bottom indicates the cursor is at line 60, column 10.

```
*httpd.conf - Bloc-notes
Eichier  Edition  Frmat  Affichage  Aide
#
Define SRVROOT "c:/xampp/apache"
ServerRoot "C:/xampp/apache"
#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
21:06:39 [main] Initializing Con # Uncomment and change the directory if mutexes are file-based and the default
21:06:39 [main] Windows Vers # mutex file directory is not on a local disk or is not appropriate for some
21:06:39 [main] XAMPP Versi # other reason.
21:06:39 [main] Control Panel #
21:06:39 [main] You are not ru # Mutex default:logs
21:06:39 [main] most applicati #
21:06:39 [main] there will be a #
21:06:39 [main] about running #
21:06:39 [main] XAMPP Install # Listen: Allows you to bind Apache to specific IP addresses and/or
21:06:39 [main] Checking for p # ports, instead of the default. See also the <VirtualHost>
21:06:41 [main] All prerequisite # directive.
21:06:41 [main] Initializing Mot #
21:06:41 [main] Starting Check # Change this to Listen on specific IP addresses as shown below to
21:06:41 [main] Control Panel # prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
<
Ln 60, Col 10    100%    Windows (CRLF)    UTF-8
```



The screenshot shows the XAMPP Control Panel on the left and the httpd.conf file in a text editor on the right. The Control Panel lists the status of various services: Apache, MySQL, FileZilla, Mercury, and Tomcat. The httpd.conf file is open to the section for listening on connections. The line `#Listen 12.34.56.78:80` is highlighted with a blue selection, and the status bar at the bottom indicates the cursor is at line 60, column 1.

```
XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]
*httpd.conf - Bloc-notes
Eichier  Edition  Frmat  Affichage  Aide
#
Define SRVROOT "c:/xampp/apache"
ServerRoot "C:/xampp/apache"
#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
21:06:39 [main] Initializing Con # Uncomment and change the directory if mutexes are file-based and the default
21:06:39 [main] Windows Vers # mutex file directory is not on a local disk or is not appropriate for some
21:06:39 [main] XAMPP Versi # other reason.
21:06:39 [main] Control Panel #
21:06:39 [main] You are not ru # Mutex default:logs
21:06:39 [main] most applicati #
21:06:39 [main] there will be a #
21:06:39 [main] about running #
21:06:39 [main] XAMPP Install # Listen: Allows you to bind Apache to specific IP addresses and/or
21:06:39 [main] Checking for p # ports, instead of the default. See also the <VirtualHost>
21:06:41 [main] All prerequisite # directive.
21:06:41 [main] Initializing Mot #
21:06:41 [main] Starting Check # Change this to Listen on specific IP addresses as shown below to
21:06:41 [main] Control Panel # prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
#Listen 8090
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
<
Ln 60, Col 1    100%    Windows (CRLF)    UTF-8
```

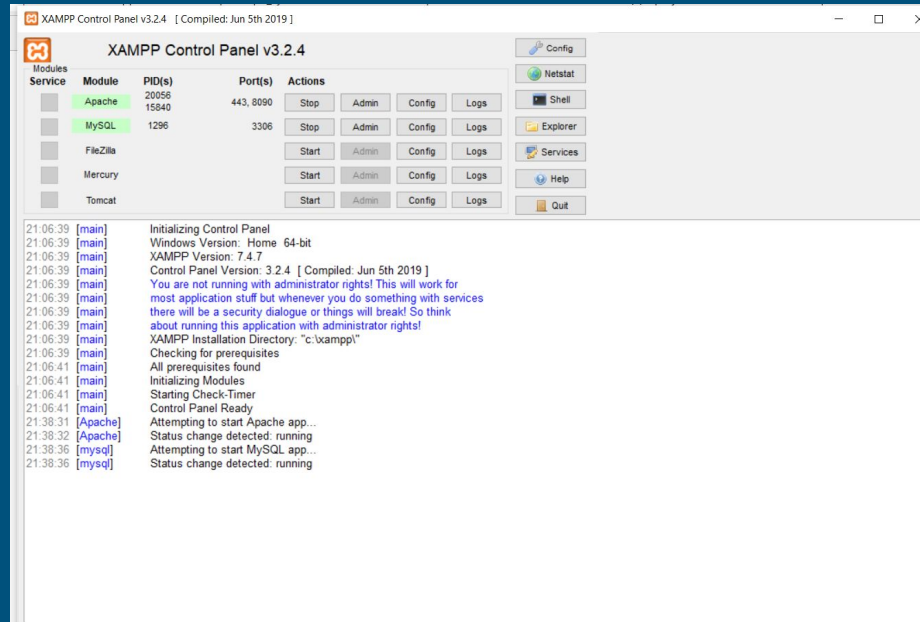
Aller sur → C:\xampp

2 dossiers :

- phpmyAdmin
- htdocs : (contient les fichiers que l'on veut mettre en ligne)

Accéder à phpMyAdmin

Start : Apache & MySQL





Dans la barre de navigation : <http://localhost:8090/phpmyadmin/>



Base de donnée



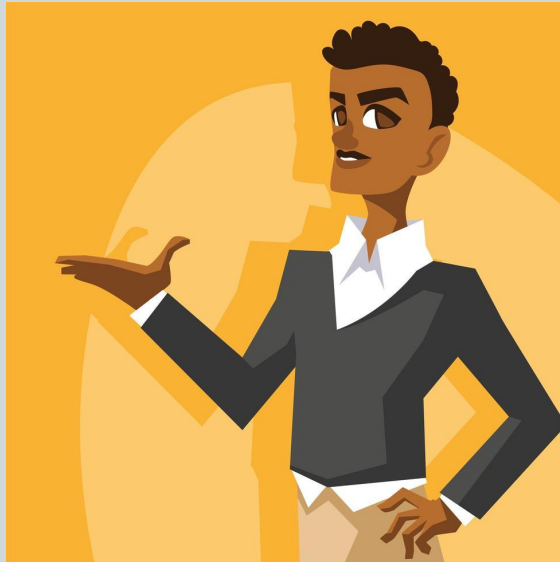
Création d'une base de donnée

Avec l'interface phpMyAdmin



concept de clé

— Place à la pratique !



Créer la table sans vous soucier des doublons

Table : Clients

codecli	prenomcli	nomcli	ruecli	cpcli	villecli
1	Alberto	Dubois	3 Rue du Pont	57500	Saint-Avold
2	Mi	Volond	4 rue de la liberté	57500	Saint-Avold
3	Roger	Botas	5 place du marché	57500	Saint-Avold
4	Edouard	Noulas	41 rue de l'église	57600	Forbach
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming
6	Eric	Pondier	14, rue des Agates	57600	Forbach
7	Thomas	Malon	12, rue des lapins	57600	Forbach
8	Rénato	Point	451, rue de légalité	57500	Saint-Avold
9	Michel	Botas	17, rue des hochets	57500	Saint-Avold
10	David	Collague	14, rue Utrillo	57600	Forbach
11	Simon	Potillon	17, rue des marguerittes	57800	Freyming



Correction

Questions:

1. Combien de requête sont présentées dans cette exemple ?
2. Regardez la base obtenue. A quoi ont servi ces requêtes ?
3. Observez la première requête. Quelle instruction permet de créer la table ?

Réponses:

1. 2. Une requête CREATE TABLE et une requête INSERT INTO
2. Créer une table et y insérer des données.
3. CREATE TABLE suivi du nom de la table. Ici la table créée s'appelle Client.

SQL





A quoi sert le langage SQL ?

Permet à notre système de base de donnée de manipuler les données



Création d'une base de donnée

Avec le langage SQL

Aller dans phpMyadmin pour taper du langage SQL ---> onglet SQL

Mots clé

les mots clé sont écrits en majuscule : (convention)

“ ; “ toujours en fin de commande



CREATE : permet de créer

```
CREATE DATABASE mydatabase;
```

Diagram illustrating the SQL command for creating a table with annotations:

```
CREATE TABLE mytable (id int(11), field1 varchar(100));
```

Annotations:

- champs id (points to `id`)
- champs (points to `field1`)
- nbr caractères (points to `100`)
- type du champs (points to `int`)
- nbr caractères (points to `11`)
- type du champs (points to `varchar`)

— Place à la pratique !



Saisissez la requête qui permettra de créer la table Jeux qui contient les champs "identifiant" qui sera de type integer et "nom" qui sera du type VARCHAR(40).
Ajouter les jeux n°1 "Monopoly" et n°2 "Trivial".

Correction

```
CREATE TABLE Jeux (identifiant INT, nom  
VARCHAR(40), PRIMARY KEY(identifiant));  
INSERT INTO Jeux VALUES ('1', 'Monopoly'), ('2',  
'Trivial');
```




Select



SELECT :

- récupère les informations de la base de donnée

sélectionne tous les champs



```
SELECT * FROM arme
```



De quelle table on va sélectionner

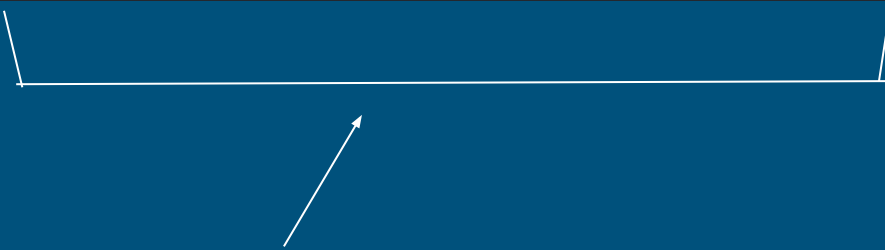


Table choisie

Sélection de champs spécifique

ex :

```
SELECT idArme , nom, levelMin FROM arme
```



Sélection des champs spécifique



SELECT AS

AS (Alias) : est un mot clé permettant de renommer des colonnes

Afficher le nom et le niveau de tous les personnages en modifiant les titres colonnes en “ Pseudo ” et “ Niveau”.

Afficher le type des armes en renommant le type en “ Types d’armes du jeu ”

```
SELECT libelle AS " Types armes du jeu"  
FROM typearme;
```




Requête de sélection avec des calculs

Récupérer le nombre d'armes existante

```
SELECT count(*)  
FROM arme
```

```
SELECT count(*) AS "Nombre d'armes "  
FROM arme
```

— Place à la pratique !



Afficher le nombre de personnages du jeu
(Attention au nom de la colonne)

Correction

```
SELECT count(*) AS "nombre de personnages "  
FROM personnage ;
```



Calcul de la moyenne avec : **AVG**

Calcul Pour additionner avec : **SUM**

Récupérer les plus petites valeurs avec : **MIN**

Récupérer les plus grandes valeurs avec : **MAX**

Calculs basiques avec : **(+, -, *, /)**



Récupérer les moyennes des niveaux des personnages

Correction

```
SELECT AVG(level) AS "Moyenne de niveau"  
FROM personnage ;
```

Récupérer la somme des points de force, d'agilité et d'intelligence de toutes les classes

Correction

```
SELECT SUM(baseForce), SUM(baseAgi), SUM(baseIntel)
FROM classe;
```

En renommant

```
SELECT SUM(baseForce) AS "Ponts de force du jeu ",
SUM(baseAgi) AS "Points d'agilité du jeu", SUM(baseIntel)
AS "Points d'intel du jeu"
FROM classe;
```

Récupérer le “level ” Minimum et Maximum des armes du jeu

Correction

```
SELECT MIN(levelMin), MAX(levelMin)  
FROM arme;
```



Afficher le nom et le nombre de points de chaque caractéristique (classe)

Correction

```
SELECT nom, baseForce + baseAgi + baseIntel AS "Nombre de  
caractéristique"  
FROM classe ;
```



SELECT DISTINCT

Permet de sélectionner

```
SELECT DISTINCT nom FROM arme;
```



SELECT : (LIMIT)

Limit au 3 premières lignes

```
SELECT * FROM `personnage` LIMIT 3;
```



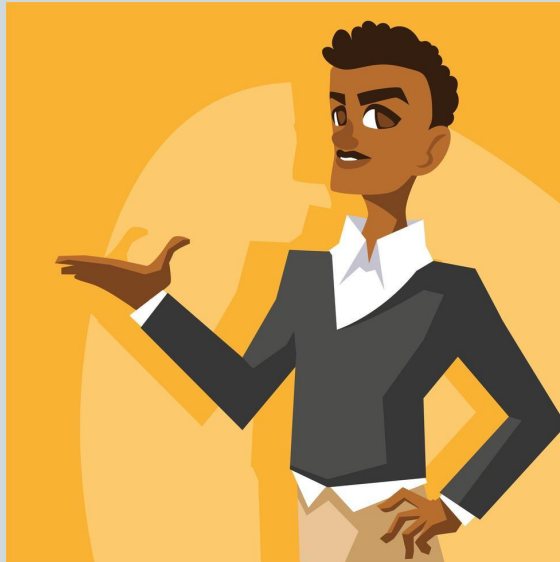

SELECT: ORDER BY ---- DESC

ORDER BY : Permet de les sélectionner dans un ordre précis

ordre décroissant

```
SELECT * FROM `personnage` ORDER BY `idPersonnage` DESC
```

— Place à la pratique !



Sélectionner uniquement les noms de la table NomCli

Correction

```
SELECT NomCli FROM Clients;
```

Requête 1
nomcli
Dubois
Volond
Botas
Noulas
Lontague
Pondier
Malon
Point
Botas
Collague
Potillon

Questions

1 - De quelle table provient cette colonne ?

2 - Quelle est l'instruction qui indique dans quelle table il faut chercher la colonne à afficher ?

3 -Quelle est l'instruction qui permet de choisir la colonne à afficher ?

Réponses

De la table Clients.

L'instruction FROM.

L'instruction SELECT.

Sélectionner uniquement les noms ruecli et villecli

Correction

```
SELECT ruecli, villecli FROM Clients;
```

Requête 1	
ruecli	villecli
3 Rue du Pont	Saint-Avoid
4 rue de la liberté	Saint-Avoid
5 place du marché	Saint-Avoid
41 rue de l'église	Forbach
21 Boulevard des oiseaux	Freyming
14, rue des Agates	Forbach
12, rue des lapins	Forbach
451, rue de légalité	Saint-Avoid
17, rue des hochets	Saint-Avoid
14, rue Utrillo	Forbach
17, rue des marguerittes	Freyming

Questions

1 - Comment affiche-t-on plusieurs colonnes ?

2 - Par quoi se termine la requête ?

Réponses

On ajoute après le SELECT toutes les colonnes que l'on souhaite afficher en les séparant par des virgules.

Par un point virgule.



Trier les cpcli par ordre croissant

Correction

```
SELECT * FROM clients ORDER BY cpcli;
```

Trier les nomcli par ordre décroissant

Correction

```
SELECT * FROM Clients ORDER BY nomcli DESC;
```



Insert



INSERT

Ajoute des lignes dans les tables

nom des champs

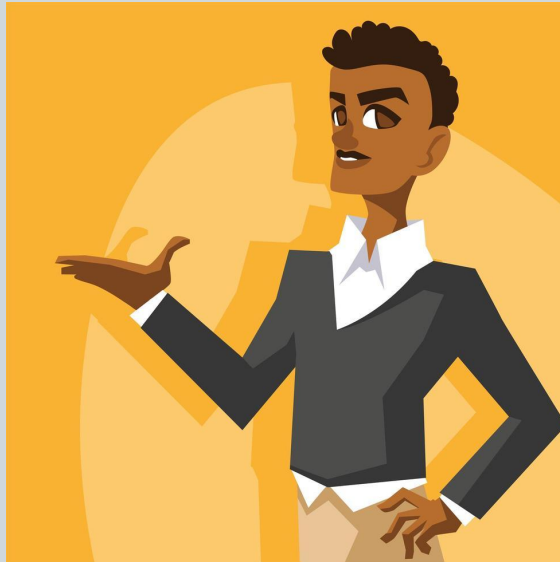


```
INSERT INTO personnage (nom, surnom, level,  
idArmeUtilise, idClasse )  
VALUES ('peter', 'spider', '9', '5', '3');
```



valeur à insérer

— Place à la pratique !



Saisissez la requête qui permettra d'insérer le client :

Prénom: scott

nom: Bis

rue : 3, rue des capucins

CP: 67500

ville : Haguenau

Résultat

Clients					
codecli	prenomcli	nomcli	ruecli	cpcli	villecli
1	Alberto	Dubois	3 Rue du Pont	57500	Saint-Avold
2	Mi	Volond	4 rue de la liberté	57500	Saint-Avold
3	Roger	Botas	5 place du marché	57500	Saint-Avold
4	Edouard	Noulas	41 rue de léglise	57600	Forbach
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming
6	Eric	Pondier	14, rue des Agates	57600	Forbach
7	Thomas	Malon	12, rue des lapins	57600	Forbach
8	Rénato	Point	451, rue de légalité	57500	Saint-Avold
9	Michel	Botas	17, rue des hochets	57500	Saint-Avold
10	David	Collague	14, rue Utrillo	57600	Forbach
11	Simon	Potillon	17, rue des marguerittes	57800	Freyming
12	Scott	BIS	3, rue des Capucins	67500	Haguenau

Correction

```
INSERT INTO Clients VALUES(NULL, 'Scott', 'BIS', '3, rue  
des Capucins', '67500', 'Haguenau');
```



Update



UPDATE

Permet de mettre à jour des données déjà enregistrées

```
UPDATE personnage SET nom = ' james '
```

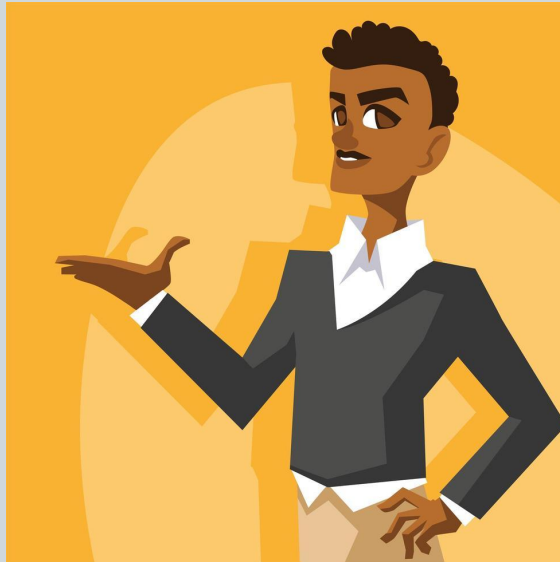
(change tous les noms de la table en james)

```
UPDATE personnage SET nom=' james ' WHERE  
surnom=' spider '
```

(change 1 seul nom dans la table)

(=> met à jour dans la table personnage le nom , là ou se trouve le surnom spider

— Place à la pratique !



Saisissez la requête qui permettra de modifier

villecli : haguenu

Résultat

Clients					
codecli	prenomcli	nomcli	ruecli	cpcli	villecli
1	Alberto	Dubois	3 Rue du Pont	57500	Haguenau
2	Mi	Volond	4 rue de la liberté	57500	Haguenau
3	Roger	Botas	5 place du marché	57500	Haguenau
4	Edouard	Noulas	41 rue de léglise	57600	Haguenau
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Haguenau
6	Eric	Pondier	14, rue des Agates	57600	Haguenau
7	Thomas	Malon	12, rue des lapins	57600	Haguenau
8	Rénato	Point	451, rue de légalité	57500	Haguenau
9	Michel	Botas	17, rue des hochets	57500	Haguenau
10	David	Collague	14, rue Utrillo	57600	Haguenau
11	Simon	Potillon	17, rue des marguerittes	57800	Haguenau

Correction

```
UPDATE Clients SET villecli='Haguenau';
```

Saisissez la requête qui permettra de modifier villecli pour un seul client

villecli : PARIS

codecli: 1

Résultat

Clients					
codecli	prenomcli	nomcli	ruecli	cpcli	villecli
1	Alberto	Dubois	3 Rue du Pont	57500	Saint-Avoid
2	Mi	Volond	4 rue de la liberté	57500	Saint-Avoid
3	Roger	Botas	5 place du marché	57500	Saint-Avoid
4	Edouard	Noulas	41 rue de l'église	57600	Forbach
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming
6	Eric	Pondier	14, rue des Agates	57600	Forbach
7	Thomas	Malon	12, rue des lapins	57600	Forbach
8	Rénato	Point	451, rue de légalité	57500	Saint-Avoid
9	Michel	Botas	17, rue des hochets	57500	Saint-Avoid
10	David	Collague	14, rue Utrillo	57600	Forbach
11	Simon	Potillon	17, rue des marguerittes	57800	Freyming
12	Scott	BIS	3, rue des Capucins	67500	Haguenau



Clients					
codecli	prenomcli	nomcli	ruecli	cpcli	villecli
1	Alberto	Dubois	3 Rue du Pont	57500	Haguenau
2	Mi	Volond	4 rue de la liberté	57500	Saint-Avoid
3	Roger	Botas	5 place du marché	57500	Saint-Avoid
4	Edouard	Noulas	41 rue de l'église	57600	Forbach
5	Paul	Lontague	21 Boulevard des oiseaux	57800	Freyming
6	Eric	Pondier	14, rue des Agates	57600	Forbach
7	Thomas	Malon	12, rue des lapins	57600	Forbach
8	Rénato	Point	451, rue de légalité	57500	Saint-Avoid
9	Michel	Botas	17, rue des hochets	57500	Saint-Avoid
10	David	Collague	14, rue Utrillo	57600	Forbach
11	Simon	Potillon	17, rue des marguerittes	57800	Freyming

Correction

```
UPDATE Clients SET villecli= 'PARIS' WHERE codecli=1;
```



Delete



DELETE

Supprime des lignes de la table

```
DELETE FROM personnage WHERE nom = 'james';
```

(supprime une ligne)

```
DELETE FROM personnage;
```

(supprime toute la table)



Drop



DROP

```
DROP TABLE dispose;
```

(supprime toute la table)

```
DROP DATABASE shop;
```

(supprime la base de donnée)



Where

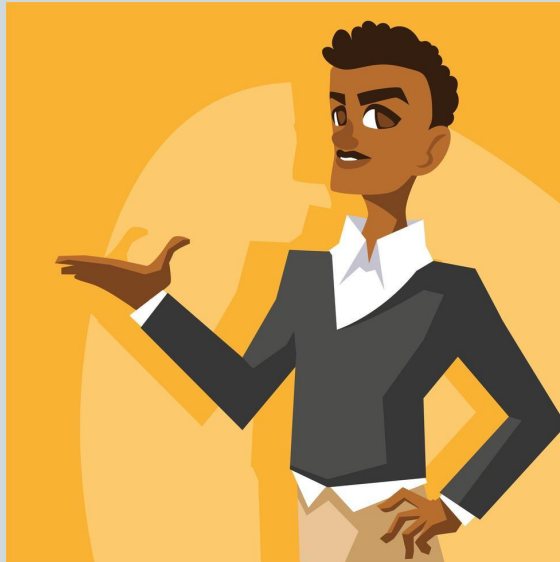


WHERE

Permet de choisir où on veut faire la requête

```
SELECT * FROM `personnage` WHERE surnom = 'baba';
```

— Place à la pratique !





WHERE



1

Récupérer toutes les armes ayant un nombre de dégâts inférieur à 25

CORRECTION

```
SELECT *  
FROM arme  
WHERE degat < 25;
```

Récupérer tous les personnages ayant le “level ” 10 et n’afficher que leur nom et leur surnom

CORRECTION

```
SELECT nom, surnom  
FROM personnage  
WHERE LEVEL = 10;
```

3

Récupérer tous les types armes à distance

CORRECTION

```
SELECT *
```

```
FROM typearme
```

```
WHERE estDistance = 1
```

3

Récupérer toutes les armes à distance

CORRECTION

```
SELECT `nom` AS 'armes à distances' FROM arme WHERE  
`idTypeArme`= 2 or `idTypeArme`= 4;
```



DROP

Saisissez la requête qui permettra de supprimer la table
location

Correction

```
DROP TABLE Locations;
```



Where AND

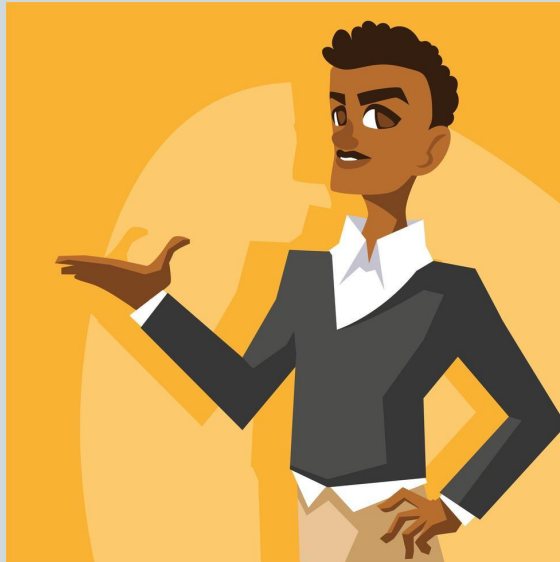
Récupérer toutes les armes ayant un “level” minimum compris entre 4 et 8 (inclus)

```
SELECT *
```

```
FROM arme
```

```
WHERE levelMin >= 4 AND levelMin <=8;
```

— Place à la pratique !



Récupérer tous les personnages ayant un identifiant inférieur à 3 et un “level” égal à 10.

correction

```
SELECT *  
FROM personnage  
WHERE idPersonnage < 3 AND level = 10;
```

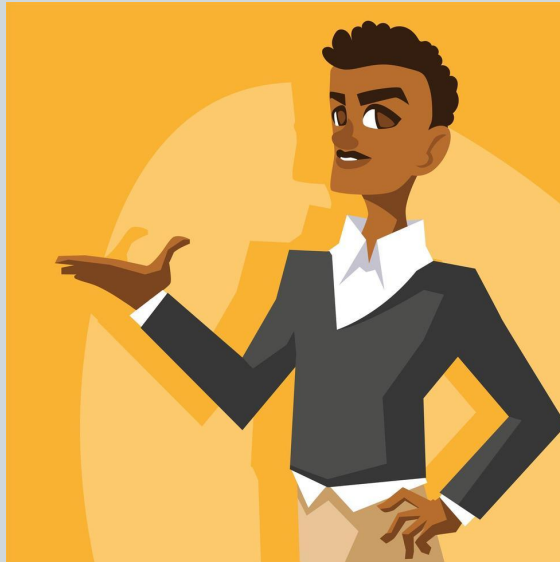


Where OR

Récupérer toutes les armes ayant un “ level ” minimum , inférieur a 8 (inclus)

```
SELECT `nom`, `levelMin` FROM arme WHERE levelMin <= 8;
```

— Place à la pratique !



Récupérer les armes 1, 2 et toutes les armes ayant un nombre de dégâts supérieur à 30

```
SELECT *
```

```
FROM arme
```

```
WHERE idArme= 1 OR idArme = 2 OR degat > 30;
```

recupérer l'arme 1 et 2 , et les armes ayant un nombre de dégâts compris entre 25 et 40

```
SELECT *
```

```
FROM arme
```

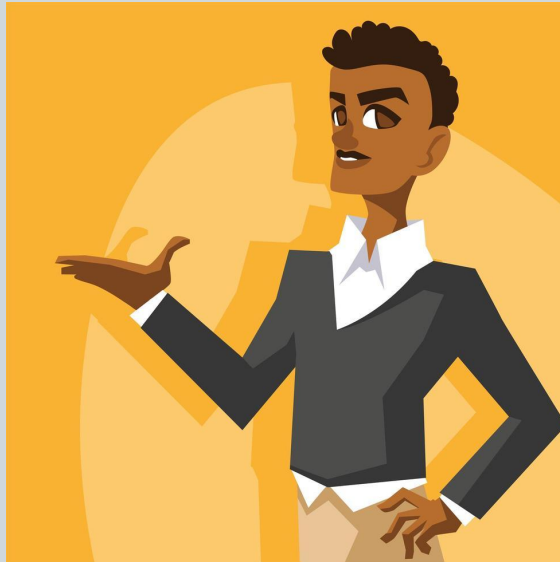
```
WHERE idArme = 1 OR idArme = 2 OR (degat >= 25 AND degat <= 40);
```

— Les fonctions sur les “string”

Afficher le nom et le surnom des personnages dans une seule colonne (concaténation)

```
SELECT CONCAT (nom, " ", surnom ) AS "personnage"  
FROM personnage;
```

— Place à la pratique !



Afficher le nom des classes avec les points de
caractéristique dans une seule colonne

Correction

```
SELECT CONCAT(nom, "-F: ", baseForce, " -A: ", baseAgi, "  
-I: ", baseIntel) AS "Classe"  
FROM classe;
```

SUBSTRING

Affiche les 6 premières lettres des noms des personnages

```
SELECT SUBSTRING(nom, 1, 6)  
FROM personnage;
```

SUBSTRING → Découpe la colonne "nom" à partir de la position 1, et prend les 6 premiers caractères

— Place à la pratique !



Afficher les 5 premières lettres du nom des classes
concaténées au 20 premières lettres de la description

Correction

```
SELECT CONCAT (SUBSTRING (nom, 1, 5) , "-" ,  
SUBSTRING (description, 1, 20))  
AS "classe"  
FROM classe;
```

Autres fonctions possible

LENGTH() : retourne la taille d'une chaîne de caractères

REPLACE() : permet de remplacer une chaîne de caractère par une autre dans une colonne

REVERSE() : permet de retourner une chaîne de caractère : toto \Rightarrow otot

ETC...

— **LIKE / IN / OFFSET / IS NULL**



LIKE

Récupérer les personnages qui ont un nom commençant par la lettre "l"

```
SELECT *
```

```
FROM personnage
```

```
WHERE nom LIKE "l%";
```

ceci est un L minuscule

LIKE

Récupérer les personnages qui ont un nom commençant par la lettre "l" et se termine par "er"

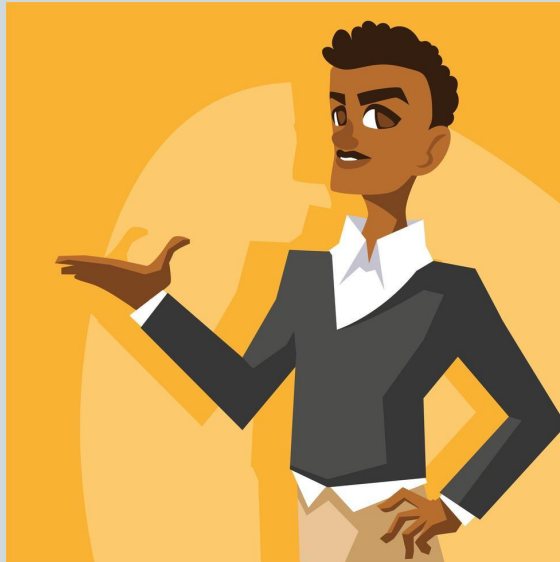
```
SELECT *
```

```
FROM personnage
```

```
WHERE nom LIKE "l%er";
```

ceci est un L minuscule

— Place à la pratique !



1

Récupérer les armes contenant le mot “ bois ”

Correction

```
SELECT *
```

```
FROM arme
```

```
WHERE nom LIKE "%bois%";
```



Récupérer les armes commençant par A et ayant
en troisième lettre un B

Correction

```
SELECT *
```

```
FROM arme
```


```
WHERE nom LIKE "a_  b%";
```

Le “ _ ” correspond à 1 caractère



IN

Récupérer les armes ayant comme identifiant
1,2,4,5 ou 7



```
SELECT *  
FROM arme  
WHERE idArme IN (1,2,4,5,7);
```


— Place à la pratique !



Récupérer les personnage ayant l'identifiant 2,3,4,
6 et qui sont " level " 10

Correction

```
SELECT *
```

```
FROM personnage
```

```
WHERE idPersonnage IN (2,3,4,6) AND level = 10 ;
```



IS NULL

Récupérer les personnages n'ayant pas un surnom



```
SELECT *
```


```
FROM personnage
```

```
WHERE surnom IS NULL;
```



IS NOT NULL

Récupérer les personnages ayant un surnom



```
SELECT *
```


```
FROM personnage
```

```
WHERE surnom IS NOT NULL;
```



OFFSET

Récupérer les lignes 4 et 5 de la table
“personnage”



```
SELECT *
```

```
FROM personnage LIMIT 2 OFFSET 3;
```

—

NoSQL

Le **NoSQL** est un type de bases de données, dont la spécificité est d'être non relationnelles.

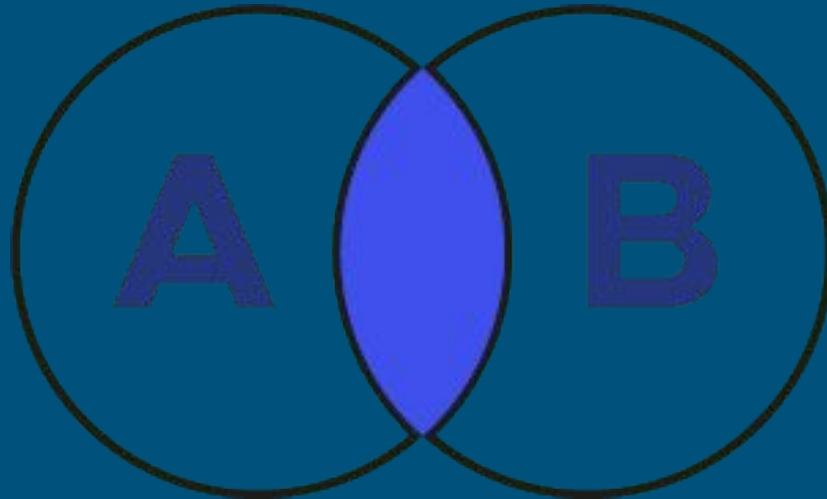
Ces systèmes permettent le stockage et l'analyse du Big Data. ... Ainsi, **NoSQL** est utilisé pour le Big Data et les applications web en temps réel.

Join

- Une jointure permet d'assembler des tables afin de réaliser des sélections sur plusieurs tables dans une même requête
- C'est l'un des principaux avantages des bases de données relationnelles

INNER JOIN (jointure interne)

permet de retourner les données quand la condition est vraie dans les deux tables.



Exemple

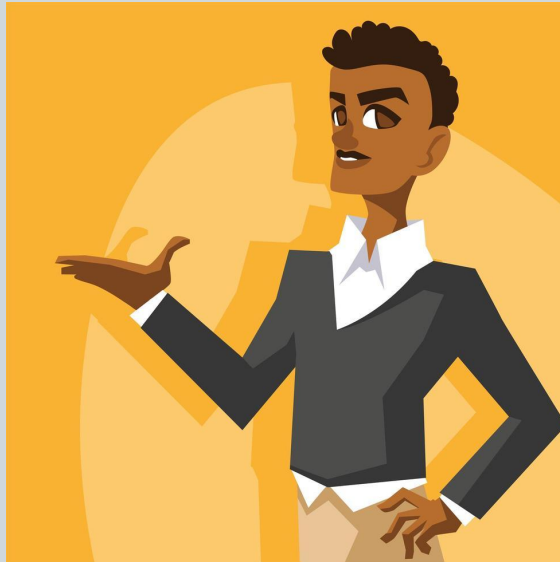
```
SELECT *
```

```
FROM personnage
```

```
INNER JOIN classe
```

```
ON personnage.idClasse = classe.idclasse
```

— Place à la pratique !



Exercice 1

Récupérer toutes les armes et leur type.

(dans database rpg)

réponse 1

```
SELECT nom, levelMin, libelle, estDistance  
FROM arme  
INNER JOIN typearme ON arme.idTypeArme =  
typearme.idTypeArme;
```

Exercice 2

Récupérer le nom des personnages et le nom de leur classe
(dans database rpg)

réponse 2

```
SELECT personnage.nom , classe.nom
```

```
FROM personnage
```

```
INNER JOIN classe ON personnage.idClasse = classe.idClasse
```

Exercice 3

Récupérer l'arme qui est utilisée par chaque personnage
(dans database rpg)

réponse 3

```
SELECT p.nom, a.nom, a.levelMin, a.degat  
FROM personnage p  
INNER JOIN arme a ON p.idArmeUtilise = a.idArme;
```

(ici on utilise un surnom au niveau des tables pour faciliter l'écriture)

Exercice 4

Récupérer l'arme qui est utilisée par chaque personnage et
le type d'arme

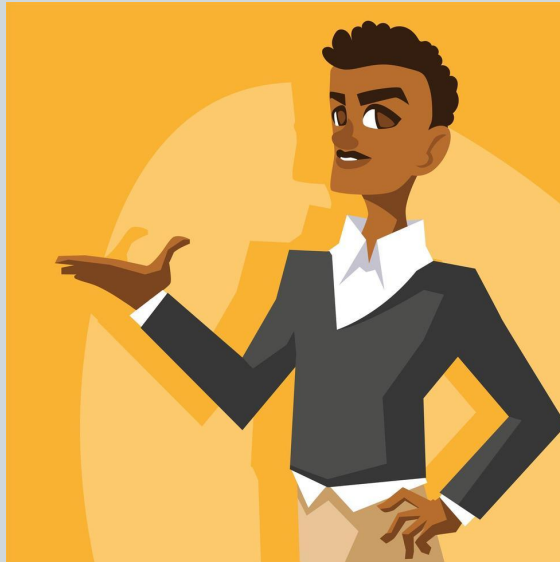
(dans database rpg)

réponse 4

```
SELECT p.nom, a.levelMin, a.degat, t.libelle AS "Type Arme", t.estDistance  
FROM personnage p  
INNER JOIN arme a ON a.idArme = p.idArmeUtilise  
INNER JOIN typearme t On t.idTypeArme = a.idTypeArme;
```

(ici on utilise un surnom au niveau des tables pour faciliter l'écriture)

— Place à la pratique !



a

Récupérer toutes les armes de tous les personnages
(afficher les armes , levelMin, level et les personnages)
(dans database rpg)

a

Correction

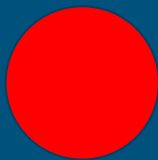
```
SELECT p.nom AS "nom des personnages", level, a.nom AS  
"arme ", levelMin  
FROM personnage p  
INNER JOIN dispose d ON p.idPersonnage = d.idPersonnage  
INNER JOIN arme a ON a.idArme = d.idArme
```


b

Correction

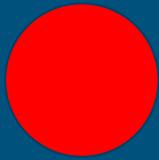
```
SELECT a.nom, a.levelMin, a.degat, t.libelle AS "Types  
d'armes "  
FROM arme a  
INNER JOIN typearme t ON a.idTypeArme = t.idTypeArme  
WHERE t.estDistance = false;
```

C



Récupérer l'arme utilisé par chaque guerrier
(afficher les personnages / arme utilisé / type d'arme)
(dans database rpg)

C



Correction

```
SELECT p.nom AS "personnage", a.nom AS "arme utilisée",  
t.libelle AS "type d'arme"  
FROM personnage p  
INNER JOIN classe c ON p.idClasse = c.idClasse  
INNER JOIN arme a ON a.idTypeArme = p.idArmeUtilise  
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme  
WHERE C.nom = "guerrier"
```

— Jointure calcule et trie

Correction

```
SELECT p.idPersonnage, p.nom AS "Personnage", a.nom AS  
"Armes", t.libelle AS "Type"  
FROM personnage p  
INNER JOIN dispose d ON d.idPersonnage = p.idPersonnage  
INNER JOIN arme a ON a.idArme = d.idArme  
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme  
WHERE p.level = 10
```


Correction

```
SELECT AVG(degat) AS "Moyenne de degat des armes a distance "  
FROM arme a  
INNER JOIN typearme t ON a.idTypeArme = t.idTypeArme  
WHERE t.estDistance = true ;
```

Récupérer tous les personnages disposant d'une arme dont le
type (libelle) commence par "a"

(nom des personnages)

(dans database rpg)

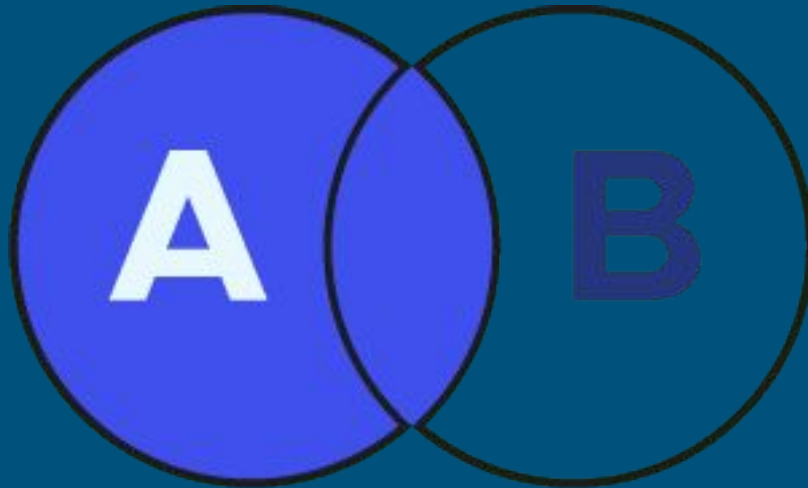
3

```
SELECT DISTINCT p.nom  
FROM personnage p  
INNER JOIN dispose d ON p.idPersonnage = d.idPersonnage  
INNER JOIN arme a ON a.idArme = d.idArme  
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme  
WHERE t.libelle LIKE "A%"
```

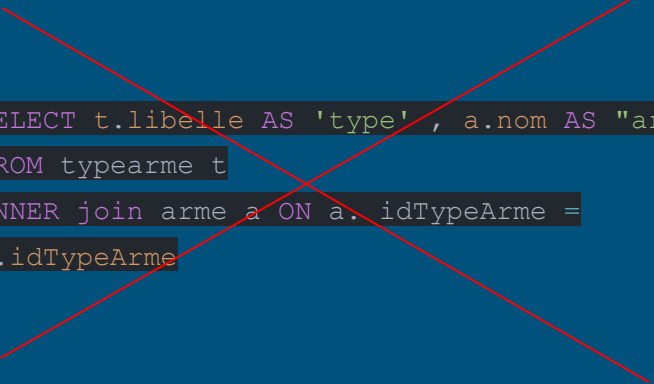
— Jointures Externes

LEFT JOIN (jointure à gauche)

Permet de retourner tous les enregistrements de la table de gauche même s'il n'y a pas de correspondance avec la table de droite.



Récupérer tous les types d'armes même ceux qui n'ont pas
d'arme associé



```
SELECT t.libelle AS 'type' , a.nom AS "arme"  
FROM typearme t  
INNER join arme a ON a. idTypeArme =  
t.idTypeArme
```



```
SELECT t.libelle AS 'type' , a.nom AS "arme"  
FROM typearme t  
LEFT join arme a ON a. idTypeArme =  
t.idTypeArme
```

```
SELECT t.libelle AS 'type' , a.nom AS "arme"  
FROM arme a  
RIGHT join typearme t ON a. idTypeArme =  
t.idTypeArme
```

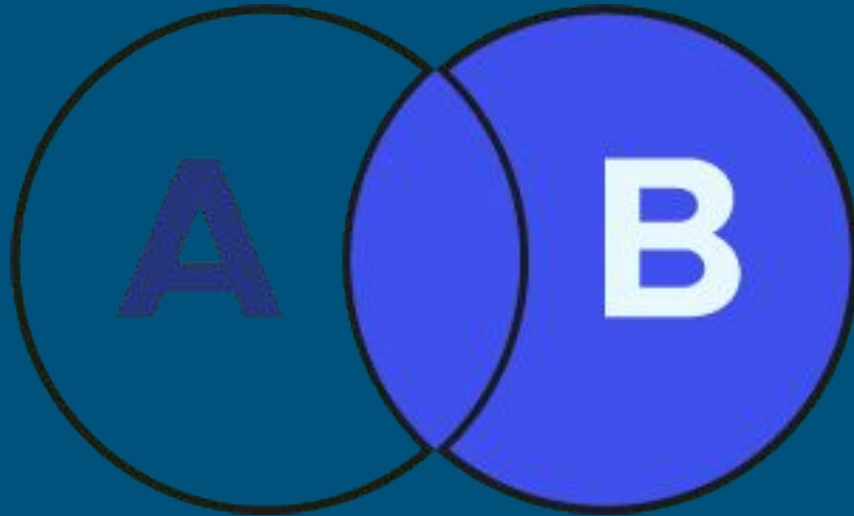
Récupérer toutes les armes et afficher le personnage qui les utilise, ordonnées par level minimum

Correction

```
SELECT *  
FROM arme a  
LEFT JOIN personnage p ON a.idArme = p.idArmeUtilise  
ORDER BY a.levelMin ;
```

RIGHT JOIN (jointure à droite)

permet de retourner tous les enregistrements de la table de droite même s'il n'y a pas de correspondance avec la table de gauche.



Récupérer toutes les armes et voir les personnages qui les ont en leur possession (table dispose).

ordonnées par id.Arme

(Utiliser RIGHT JOIN)

(idArme / arme / Personnage)

```
SELECT *
```

```
FROM personnage p
```

```
RIGHT JOIN dispose d ON d.idPersonnage = p.idPersonnage
```

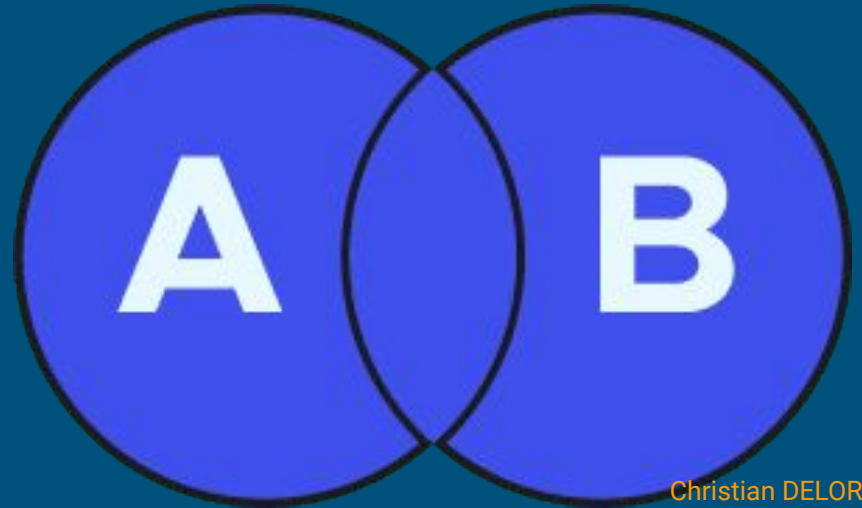
```
RIGHT JOIN arme a ON a.idArme = d.idArme
```

```
ORDER BY a.idArme ;
```

FULL JOIN

Permet de retourner tous les enregistrements des deux tables même s'il n'y a pas de correspondance avec l'autre.

Malheureusement le FULL JOIN n'existe pas en MySQL



NATURAL JOIN

Permet de retourner les enregistrements des deux tables de façon « naturelle ».

-INNER JOIN : Permet de récupérer les informations entre deux tables lorsque la condition sur le ON est respectée sur les deux tables,

-LEFT / RIGHT JOIN : Permet de récupérer les informations entre deux tables en gardant toutes les informations de l'une des deux tables.

-FULL JOIN: Récupère toutes les informations des deux tables sans perte d'information (mais elle peut afficher beaucoup de NULL),

-NATURAL JOIN: Effectue une jointure naturelle entre les deux tables, on n'a pas besoin de spécifier le ON.

— les groupements

exemple

Afficher le nombre d'arme par type d'arme

```
SELECT t.libelle AS "type arme ", COUNT(*) as "nbr d'arme"  
from arme a  
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme  
GROUP BY t.libelle
```

Afficher le nombre de personnage par classe
(classe / description / nbr personnage)

```
SELECT c.nom AS "classe", c.description, count(*) as "nbr  
personnage"  
FROM classe c  
INNER JOIN personnage p ON c.idClasse = p.idClasse  
GROUP BY c.nom ,c.description
```

Afficher le nombre d'arme dont dispose chaque personnages

```
SELECT p.nom, count(*)
```

```
FROM personnage p
```

```
INNER JOIN dispose d ON d.idPersonnage = p.idPersonnage
```

```
INNER JOIN arme a ON a.idArme = d.idArme
```

```
GROUP BY p.nom
```

afficher le nombre d'armes dont dispose chaque personnage mais seulement les guerriers

```
SELECT c.nom AS "classe", p.nom , count(*)  
FROM personnage p  
INNER JOIN dispose d ON d.idPersonnage = p.idPersonnage  
INNER JOIN arme a ON a.idArme = d.idArme  
INNER JOIN classe c ON c.idClasse = p.idClasse  
WHERE c.nom = "guerrier"  
GROUP BY p.nom
```



Afficher le nombre de personnages par arme

```
SELECT a.nom, count(p.idPersonnage)
```

```
FROM arme a
```

```
LEFT JOIN dispose d ON d.idArme = a.idArme
```

```
LEFT JOIN personnage p ON p.idPersonnage = d.idPersonnage
```

```
GROUP BY a.nom
```



Afficher le niveau moyen de chaque classe

```
SELECT c.nom AS "classe", AVG(p.level) AS "Niveau moyen"  
from classe c  
INNER JOIN personnage p ON c.idClasse = p.idClasse  
GROUP by c.nom
```



HAVING

Requête précédente

Afficher le niveau moyen de chaque classe

```
SELECT c.nom AS "classe", AVG(p.level) AS "Niveau moyen"  
from classe c  
INNER JOIN personnage p ON c.idClasse = p.idClasse  
GROUP by c.nom  
  
HAVING AVG(p.level) > 9
```



Permet de filtrer sur les fonctions de calcul
(AVG / COUNT / SUM / MIN / MAX)

Afficher le nbr de personnage par arme et ne garder que les armes ayant entre 1
et 2 utilisateurs
(table dispose)

```
SELECT a.nom, count(p.idPersonnage)
```

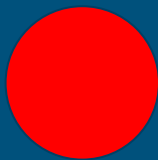
```
FROM arme a
```

```
INNER JOIN dispose d ON d.idArme = a.idArme
```

```
INNER JOIN personnage p ON p.idPersonnage = d.idPersonnage
```

```
GROUP BY a.nom
```

```
HAVING count(p.idPersonnage) <= 2
```



Afficher le nombre d'armes présents (par type) mais ne prendre que les armes de corps à corps présent au maximum 1 fois

— Les requêtes imbriquées

exemple

Récupérer les armes ayant un nombre de dégât supérieur à la moyenne du nombre de dégât de toutes les armes

(nom des armes / dégâts)

```
SELECT a.nom, a.degat
```

```
FROM arme a
```

```
WHERE a.degat > (SELECT AVG(degat) FROM arme);
```

Récupérer les personnages ayant un level inférieur à la moyenne

```
select *  
from personnage p  
WHERE p.level <  
(SELECT avg(level)  
FROM personnage)
```

Récupérer les personnages ayant un level supérieur à la moyenne des archers

```
SELECT *  
from personnage p  
WHERE p.level >  
(SELECT avg(level)  
FROM personnage p  
INNER JOIN classe c ON p.idClasse = c.idClasse  
WHERE c.nom = "archer")
```

— Les requêtes imbriquées (complexe)

exemple 1

Pour récupérer les armes à distances,
récupérer le nombre maximum d'occurrence du type d'arme

Dans une entité, une **occurrence** correspond à l'ensemble des valeurs des propriétés rattachées à un seul identifiant.

Dans une relation, une **occurrence** correspond à l'ensemble des valeurs des propriétés de la relation (représenté par les clés de chaque entité liée) : on l'appelle alors une **occurrence** de relation.

```
SELECT MAX(valeur) AS "nbr d'arme max a distance"
```

```
from (
```

```
    SELECT t.libelle COUNT(*) AS "valeur"
```

```
        FROM arme a
```

```
        INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme
```

```
        WHERE t.estDistance = true
```

```
        GROUP BY t.libelle
```

```
) v
```

Récupérer les types d'armes ayant le nombre égal d'occurrence de la précédente requête

```
SELECT t.libelle AS "type arme", COUNT(*) AS "nombre d'arme"
FROM typearme t
INNER JOIN arme a ON t.idTypeArme = a.idTypeArme
GROUP BY t.libelle
HAVING COUNT(*) =
    (SELECT MAX(valeur) AS "nbr d'are max a distance"
    from (
        SELECT COUNT(*) AS "valeur"
        FROM arme a
        INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme
        WHERE t.estDistance = true
        GROUP BY t.libelle
    ) v
)
```

Récupérer les armes ayant un nombre de dégâts supérieur au nombre de dégâts
des arc

Récupère toutes les occurrences supérieur à celle qui sont retournées par la sous-requête

```
SELECT *  
from arme a  
WHERE a.degat > ALL  
(  
SELECT a.degat  
FROM arme a  
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme  
WHERE t.libelle = "Arc"  
)
```

Récupérer les armes au corps à corps sans utiliser de jointure

```
SELECT *  
FROM arme  
WHERE idTypeArme IN  
(  
    SELECT t.idTypeArme  
    FROM typearme t  
    WHERE t.estDistance = false  
)
```

Ecrire l'équivalent avec des jointures

```
SELECT a.idArme, a.nom, a.levelMin, a.degat, a.idTypeArme
FROM arme a
INNER JOIN typearme t ON t.idTypeArme = a.idTypeArme
WHERE t.estDistance = false
```

— **Modifier les données**

Modifier la table personnage pour ajouter une dat de naissance ,
définir ensuite une valeur pour chaque personnage



Les dates



Récupérer les personnages nés après 2002

```
SELECT *
```

```
FROM personnage P
```

```
WHERE p.dateNaissance > "2002-01-01";
```

Récupérer l'année de naissance de tous les personnages

```
SELECT nom, YEAR(dateNaissance) AS "année Naissance"  
FROM personnage;
```



Récupérer l'âge de chaque personnage

```
SELECT nom, DATEDIFF(now(),dateNaissance ) / 365 AS "Age"  
FROM personnage;
```



Calculer la moyenne d'âge des personnes

```
SELECT AVG(date1.AGE)
```

```
from(SELECT nom, DATEDIFF(now(),dateNaissance)/365
```

```
AS "Age"
```

```
FROM personnage) date1
```




Récupérer les personnages ayant plus de 15ans

```
SELECT date1.nom, date1.AGE
```

```
from(SELECT nom, DATEDIFF(now(),dateNaissance)/365
```

```
AS "Age"
```

```
FROM personnage) date1
```

```
WHERE date1.AGE > 15
```

LES VIEWS

En SQL, une vue est une table virtuelle basée sur le jeu de résultats d'une instruction SQL.

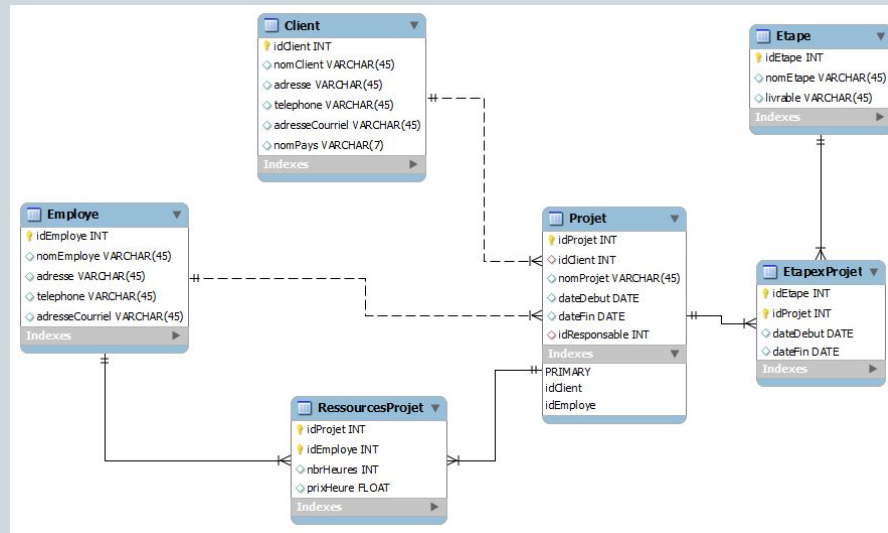
Une vue contient des lignes et des colonnes, tout comme une vraie table. Les champs d'une vue sont des champs d'une ou plusieurs tables réelles de la base de données.

Vous pouvez ajouter des instructions et des fonctions SQL à une vue et présenter les données comme si elles provenaient d'une seule table.

Une vue est créée avec l'instruction `CREATE VIEW`.

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

PROJET



Créer une base de donnée

(Envoie des fichiers en pdf)

Table : Auteurs

aid	nom
1	Victor Hugo
2	Michel MonBlanc
3	Dan Simmons
4	Jean Dorian
5	Alphonse Daudet

Table : Editeurs

edid	nom	adresse
1	Campus Presse	10, rue de la Paix, 75015 Paris
2	Flammarion	12, avenue des Champs Elysées, 75010 Paris
3	Larousse	20, rue du Peuple Belge, 59000 Lille
4	Plein Air	20, rue du Ciel, 66666 Paradiis

Table : Livres

isbn	titre	prix	edid	date_edition
0x0	Journal Interne	3	3	2016-04-03
215	Hypérion	7	1	2016-02-01
214	La chute d'Hypérion	7	1	2016-01-03
548	Les misérables	7	2	2016-02-11
547	Notre dame de Paris	7	2	2016-04-04
744	JavaScript 1.3	55	1	2016-03-02
855	Dictionnaire	44	3	2016-05-05
856	Les noms propres	43	3	2016-05-04

Table : Livres_auteurs

aid	isbn
2	744
4	744
1	547
1	548
3	214
3	215

Table : stock

id	isbn	id_librairie	nbre_exemplaire
1	0x0	1	1500
2	0x0	2	2000
3	214	1	1400
4	214	2	1879
5	215	1	1366
6	215	2	714
7	547	1	295
8	547	2	1340
9	548	1	1441
10	548	2	1700
11	744	1	900
12	744	2	300
13	855	1	1700
14	855	2	1600

Table Auteurs (aid, nom)

Clé primaire : aid

Table Editeurs (edid, nom, adresse)

Clé primaire : edid

Table Livres (isbn, titre, prix, edid, date_edition)

Clé primaire : isbn

Clé étrangère : edid de la table Editeurs

Table Livres_auteurs (aid, isbn)

Clé étrangère : isbn de la table Livres, aid de la table Auteurs

Table stock (id, isbn, id_librairie, nbre_exemplaire)

Clé primaire : id

Clé étrangère : isbn de la table Livres

MERCI

