MySQL, SQL and databases

What is
a database ?

Tables that have a link between them.

Example :

Films		
title	**year**	**category**
Trainspotting	1996	drama
Batman	1989	action
Whiplash	2014	music
Arrival	2016	sci-fi

Actors		
ID	**lastname**	**firstname**
1	Mirren	Helen
2	Asano	Tadanobu
3	Waititi	Taika
4	Lee	Pace

What is MySQL ?

A database management system (DBMS).

That's what makes it possible to manipulate databases.

There are others like SQLite, postgreSQL, mongoDB.

What is phpMyAdmin ?

A web application to manage MySQL databases.

The application is made with PHP.

What is SQL ?

This is the language that allows you to interact with databases.

At first, we will make CRUD :

- **C**reate (add data in the DB)
- **R**ead (get the data from the DB)
- **U**pdate (update the data from DB)
- **D**elete (delete data from DB)

At work !

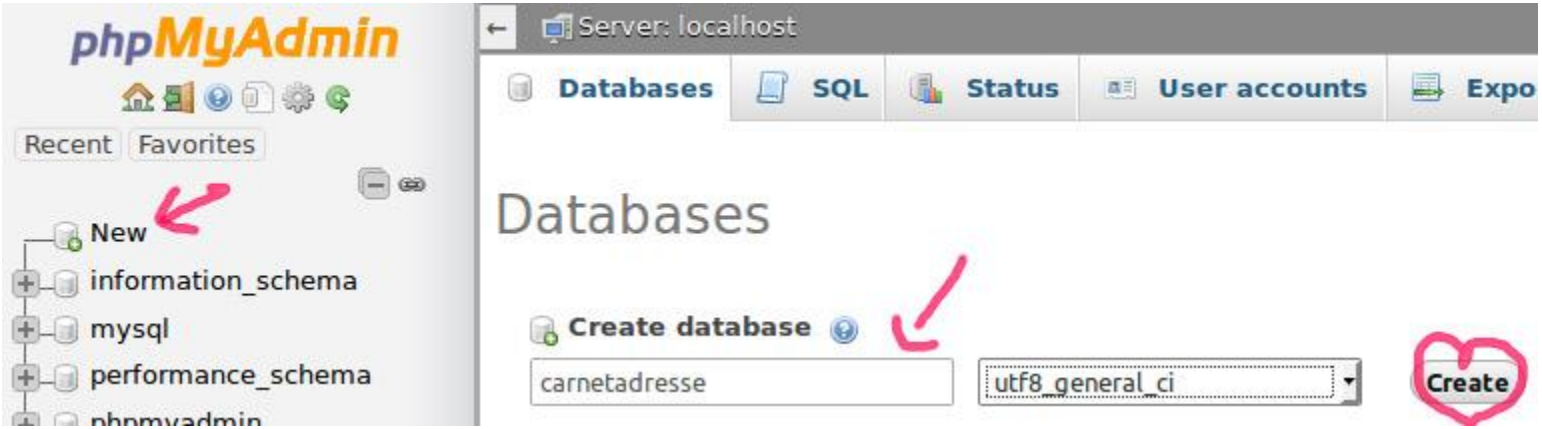
Install
phpmyadmin

Normally, you have already installed LAMP on your machines. You have already done PHP. You lack an interface to create but especially to visualize and play with the databases.

A wiki is available on GitHub at this address.

Otherwise, you can go directly to the official documentation.

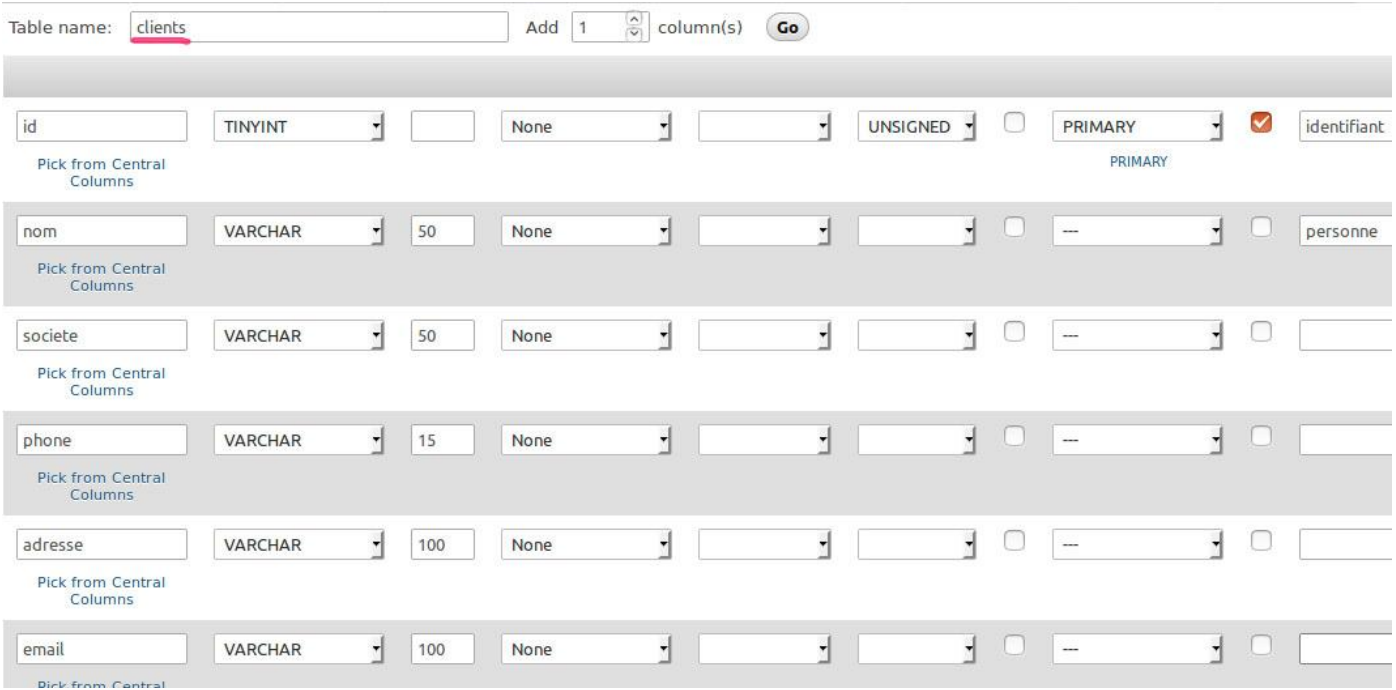
Create a new database



Create a new table

We'll do this in two steps. First we create the table by giving it a name and defining a certain number of columns.

Create a new table



Create a new table

	#	Name	Type	Collation	Attributes	Null	Default	Extra
☐	1	**id** 🔑	tinyint(3)		UNSIGNED	No	*None*	AUTO_INCREMENT
☐	2	**nom**	varchar(50)	utf8_general_ci		No	*None*	
☐	3	**societe**	varchar(50)	utf8_general_ci		No	*None*	
☐	4	**phone**	varchar(15)	utf8_general_ci		No	*None*	
☐	5	**adresse**	varchar(100)	utf8_general_ci		No	*None*	
☐	6	**email**	varchar(100)	utf8_general_ci		No	*None*	

What are those types ?

Define the type of a column to know if we will store text, numbers, dates, ...

Digital	INT, TINYINT, SMALLINT, MEDIUMINT et BIGINT
Alphanumec	CHAR, VARCHAR, TEXT
Temporal	DATE, DATETIME, TIME, TIMESTAMP et YEAR

It is important to correctly define the type of a column. If not :

- Waste of memory
- Performance problem (search faster on a number than a string)
- unable to use features specific to a data type

What are those attributes ?

Binary	
Unsigned	Can not be negative
Unsigned Zerofill	Example: SMALLINT(4) UNSIGNED ZEROFILL means that in the column we can enter numbers with 4 digits and those who have less will register with 0's. 23 => 0023
On update CURRENT_TIMESTAMP	the date will be automatically updated each time the registration is updated

What is this primary key ?

There are two types of keys :

- **Primary key** : is used to uniquely identify a line
- **Foreign key** : is used to manage the relation between the different tables et to ensure data consistency (we will see later).

Let’s play a little bit...

Create a database called becode

Then create a table called students.

Create columns :

- idstudent (primary key, auto_increment, unsigned)
- lastname,
- firstname,
- birthdate,
- gender,
- school

First, fill in the table manually by adding, modifying, deleting data via phpmyadmin.

ATTENTION : make sure that the gender column contains only two choices.
(example: F or M)
For column school, put either Andy or Central.

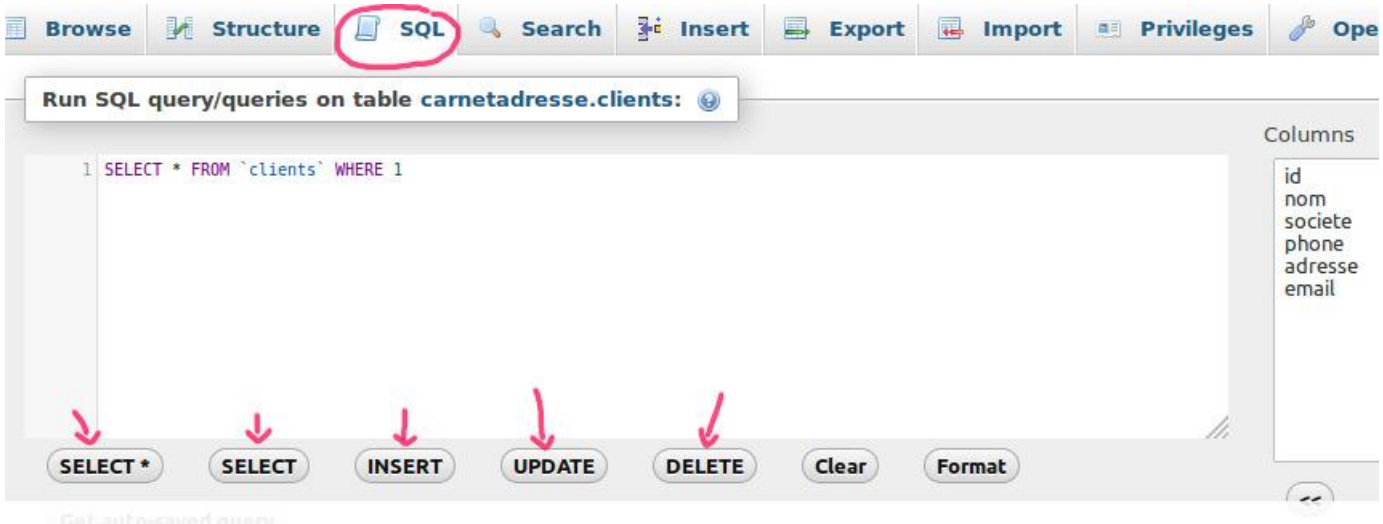
Make sure there's at least 10 people in there.

SQL

How do you speak this magic language?
The basics...

Talking SQL
The basics

You are lucky, in PHPmyAdmin, in the SQL tab, there are already buttons that generate the basic syntax.



Talking SQL
select

Basic structure of the query

```
SELECT
column1, column2, column3, column4
FROM
table_name
WHERE
condition
;
```

To select all columns, we use *.

```
SELECT
*
FROM
table_name
;
```

Talking SQL

For conditions (where)

=	"equal to"
>	"larger than"
<	"smaller than"
>=	"larger or equal to"
<=	"smaller or equal to"
<>	"different from"
LIKE	"that looks like"

Talking SQL

For conditions (like)

LIKE, associated with **%**, will make possible to find resultats in DB according to 3 criteria:

- "which **starts** by *string*". (LIKE 'Er%' : any word beginning with "Er"),
- "which **ends** by *string*" (LIKE '%Er' : any word ending by "Er"),
- or "which **contains** this *string*" (LIKE '%Er%' : any word containing "Er").

Talking SQL

How to sort

With **ORDER BY**.

We will write:
ORDER BY ASC to sort data of a column from A to Z or from the smaller to the higher.
ORDER BY DESC to sort data of a column from Z to A or from the higher to the smaller.

Talking SQL

Limit the results

With **LIMIT**, we will limit the number of results.

Talking SQL
insert

Basic structure of the query

```
INSERT INTO
table_name
(column1, column2, column3...
last_column)
VALUES
(value_column_1, value_column_2, ...
value_last_column);
```

Example

```
INSERT INTO
clients
(name, company, phone, address, mail)
VALUES
('Jen barber', 'Reynholm Industries',
'05 67 89 90 12', 'IT', 'jen@ri.uk');
```

Talking SQL
update

Basic structure of the query

ATTENTION : If we don't put WHERE on, all the rows will be updated. #bigpoop

```
UPDATE table_name
SET column_name = "new value"
WHERE column_name OPERATOR "value"
    [and|or colunm_name OPERATOR
"value"];

[ ] = optional
```

Example

```
UPDATE clients
SET company = "Restauratec"
WHERE company = "COGIP";
```

Talking SQL
delete

Basic structure of the query

ATTENTION : f we don't put WHERE on, all the rows will be deleted. #bigpoop

```
DELETE from
column_name
WHERE column_name OPERATOR "value"
[and|or "column_name" OPERATOR
"valeur"];

[ ] = optional
```

Example

```
DELETE FROM
clients
WHERE
company = "restauratec";
```

Talking SQL
AS (small tip of alias)

Alias on column

```
SELECT too_long_column_name AS c1
FROM `table`
```

Alias on table

```
SELECT *
FROM `ton_long_table_name` AS t1
```

Talking SQL
AS (small tip of alias)

Example

```
SELECT f.invoice_number AS nun, p.people_lastname,
p.people_firstname AS people, c.company_name AS company

FROM invoices AS f, people AS p, companies AS c

WHERE f.fk_company = c.id_company AND f.fk_people =
p.id_people
```

Let’s play a little bit...

Also in phpMyAdmin but via the SQL tab, you will learn how to display, add, modify and delete data in SQL.

- Take over again the ‘becode’ database.
- Display all the data.
- Display only the first names.
- Display first names, birthdates and school for each students
- Display only female students
- Display only students from Andy school.

- Display only first names of students, in reverse order to the alphabet (DESC). Then, the same but with a limit of two results.

- Add Ginette Dalor, born on 01/01/1930 and assigned her to Central, still in SQL.

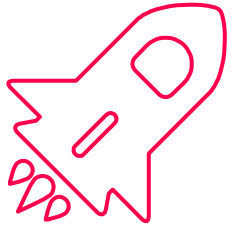
- Update Ginette (still in SQL) and change her gender and her first name to “Omer”.

- Delete student with ID equal to 3

- Doing other manipulations to see if you're understood.

Resources

to start or go further



- Small quizz to check your knowledge
- Become familiar with SQL (Sololearn)
- SQL training (w3schools)
- Date types (W3 Resources)
- Choose the right SQL column types (brandonsavage.net)