



Database concepts (CBD)

Class 01

Database

A database is a collection of information that is structured in such a way that it can easily be accessed, managed and updated. Computer databases typically contain data gathered in records or files, containing information about a specific theme such as sales transactions or interactions with specific customers.

In a *relational database*, information about a specific customer, for instance, is structure into rows, columns and tables which are indexed to make it easier to find using SQL or NoSQL queries.

Types of databases

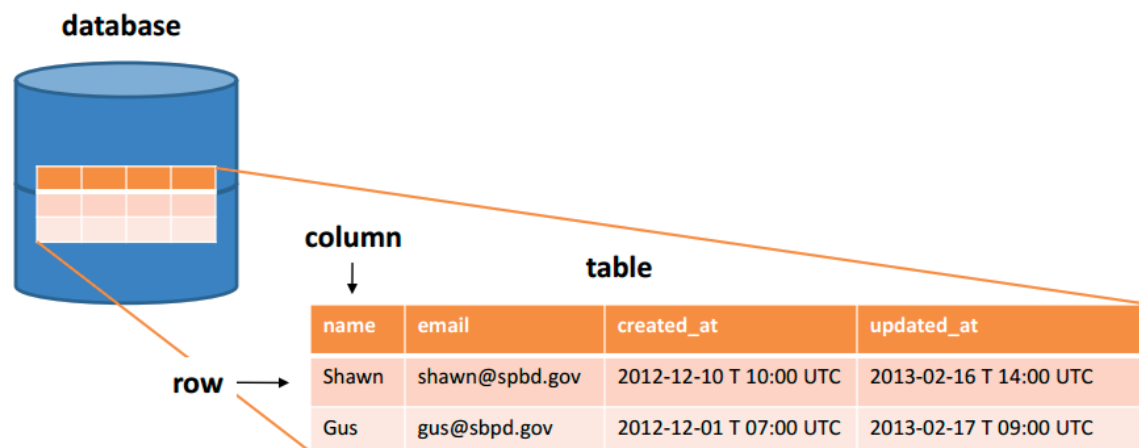
Databases greatly evolved since their invention in the 1960s, beginning with hierarchical and network databases, through the 1980s with object-oriented databases, and today with SQL, NoSQL databases and cloud databases.

In a certain way, databases could simply be classified based on content type: bibliographic, full text, numeric and images. Although, in computing, databases are more often classified based on their organizational approach.

Relational database

Invented by *E.F. Codd* (IBM, 1970), a relational database is a *tabular database* in which data is defined so that it can be reorganized and accessed in a number of different ways, often using *Structured Query Language* (SQL) as the standard program interface used for a relational database.

Relational databases are made up of a collection of tables containing data of predefined categories. Each table has at least one data category in a column (*field*), and each row (*entries*) has a certain data instance for the categories which are defined in the columns.

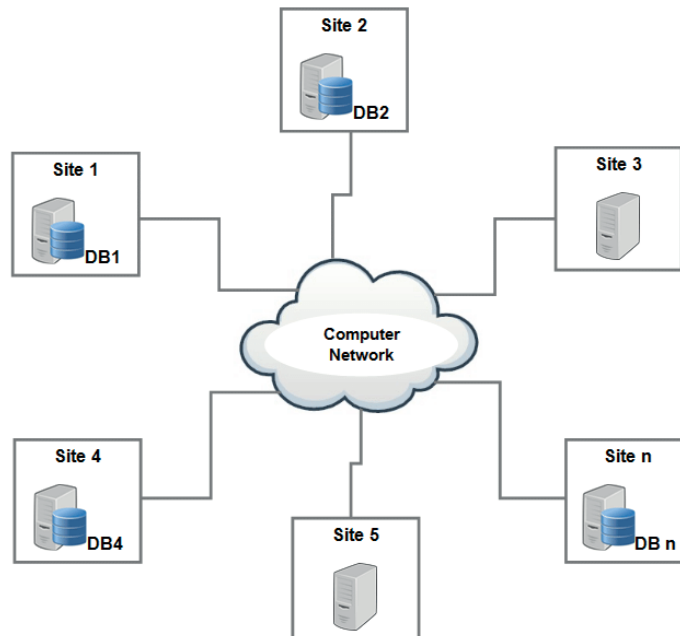


A database consists of one or multiple tables containing information made of entries (rows) containing fields (columns).

Distributed database

In a distributed database, portions of the database are stored in multiple physical locations, and processing is dispersed or replicated among different points in a network.

Distributed databases can be homogeneous or heterogeneous. In a homogeneous distributed database, all the physical locations system have the same underlying hardware and run the same operating systems and database applications. In a heterogeneous distributed database, the hardware, operating systems or database applications may be different at each of the locations.



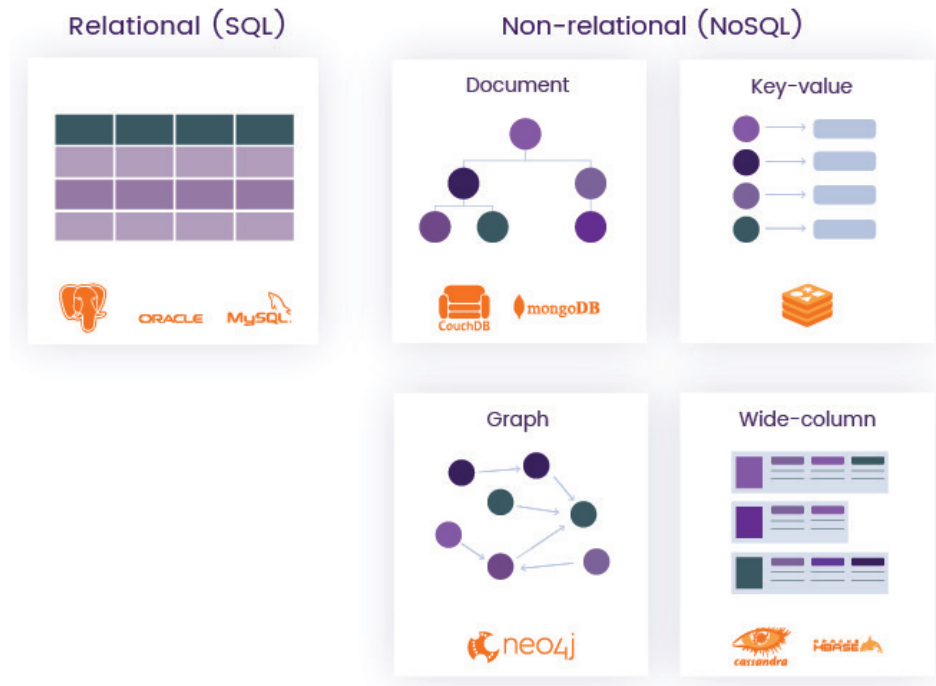
Cloud database

A cloud database consists in a database that has been optimized or built for a virtual environment (hybrid, public or private cloud). It provides benefits such as the ability to pay for storage capacity and bandwidth based on usage, and they provide scalability on demand, along with high availability.



NoSQL database

NoSQL databases are useful for large sets of distributed data. They are especially effective for big data performance where relational databases can't always manage efficiently.



Object-oriented database

Although items created using object-oriented programming languages may be stored in relational databases, object-oriented databases are well-suited for those items.

An object-oriented database is structured around objects instead of being structured around actions, and around data instead of logic.

For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value.

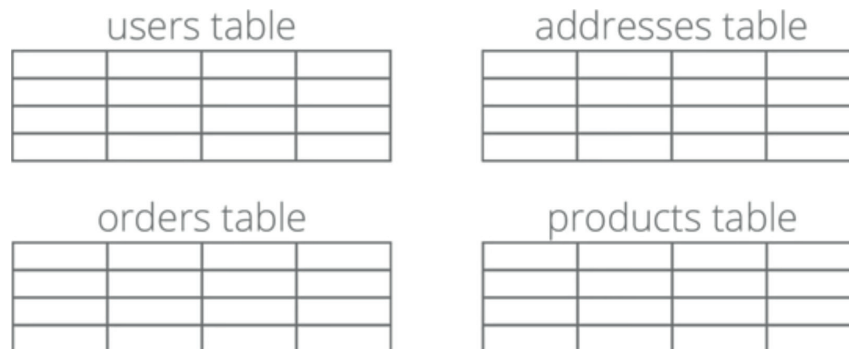
Graph database

Growing in popularity, graph-oriented databases (graph databases) are NoSQL databases using graph theory to store, map and query relationships. Basically, a graph databases consists in collections of nodes and edges, where each node represents an entity, and each edge represents a connection between nodes.

Graph databases often employ SPARQL, a declarative programming language and protocol for graph database analytics which allows to perform all the analytics that SQL can perform, plus it can be used for semantic analysis, the examination of relationships.

MySQL database

MySQL is a *Structured Query Language* which makes it possible to communicate and interact with a database. A database is made of different tables containing entries (rows) and fields containing values (columns).



Table

In the example of a table below, each row represents an entry for different individuals, and each column represent different types of information (field) contained in every entry. In this table, each entry contains the name, age, phone number and email of every users.

Name	Age	Phone	email
John Smith	52	514 123-4567	him@fakemail.com
Jane Doe	25	514 123-4567	her@fakemail.com
Ludwig Roth	33	514 123-4567	him@fakemail.com
Susan Cohen	41	514 123-4567	her@fakemail.com

MySQL

MySQL is a coding language that allows searching and interacting with a database by modifying table's informations, adding or removing entries, etc. To use MySQL and test it, you need to work online or to use a virtual server such as MAMP.

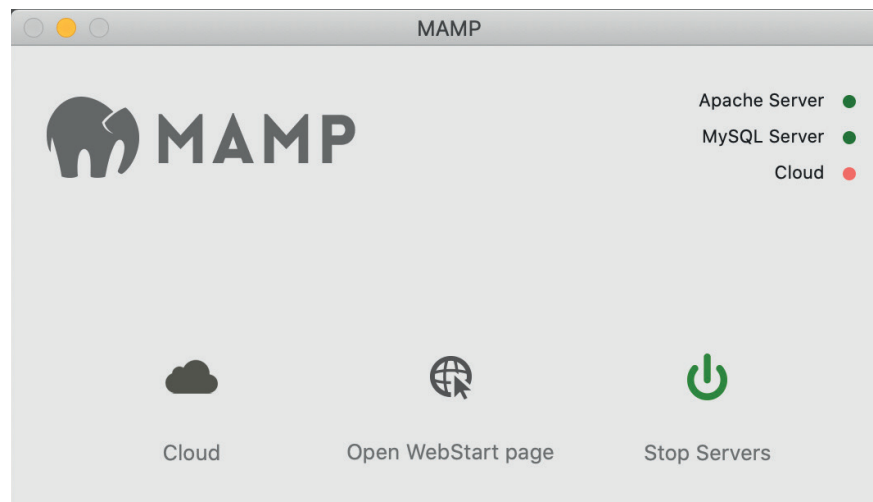
Primary keys

A primary key is used just as an ID number. For instance, in our users table, it is possible that more than one user would be named John Smith. To avoid such a situation, a primary key can be associated to each entry. This way it is possible to retrieve an entry using the ID number instead of any other information.

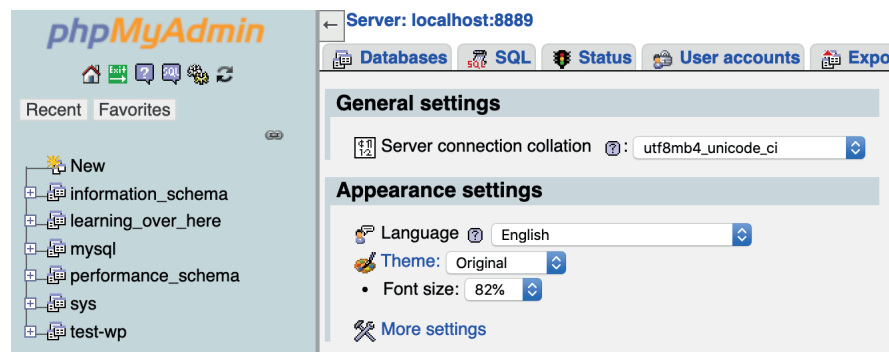
ID	Name	Age	Phone	email
1	John Smith	52	514 123-4567	him@fakemail.com
2	John Smith	29	514 123-1212	him@fakemail.com
3	Jane Doe	25	514 123-4567	her@fakemail.com
4	Ludwig Roth	33	514 123-4567	him@fakemail.com

Importing a SQL database to MAMP

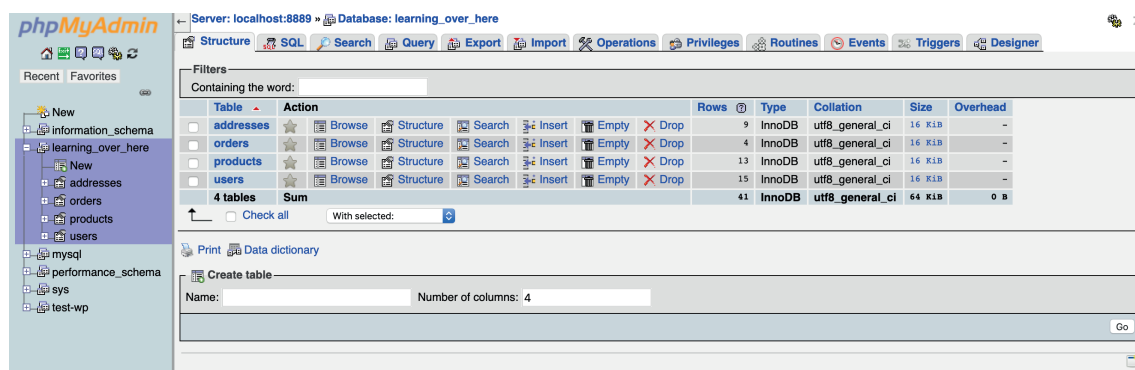
After downloading the proper SQL file, run MAMP and make sure PHP and MySQL are running (green buttons should show).



Click *Open WebStart Page* and from the web page menu, select **TOOLS/PHPADMIN**, which will open the PHPADMIN page.



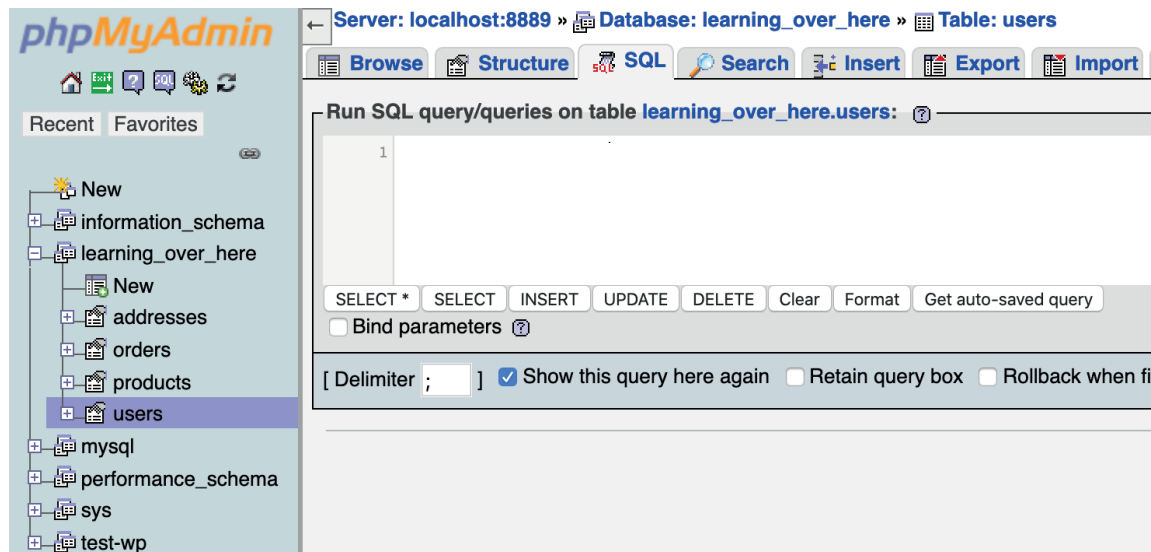
When in PHPADMIN page, click the **IMPORT** tab and select the SQL file you want to import¹ to MAMP. Once the file has been imported, select it from the left panel so it shows like below. The purple background indicates a certain database has been selected.



1. File "learning_over_here.sql", from DEV MOUNTAIN's Quentin Watt's tutorials.

SQL queries

Once a database has been selected in the left panel of PHPADMIN, the list of the database's tables should show in the main panel. In order to make a query using SQL, it is needed to first click the SQL tab and to type the query in the white field.



Selecting a table

In order to select one of the database's table, it is needed to use the SELECT command, followed by the star ("*") which means "all", and finally the name of the table you want to select between grave accents (not quotes). When this is done, press the GO button and wait for loading to succeed. Any error would be displayed in the main panel.

```
SELECT * FROM `users`
```

Note :

In MySQL, all commands must be written in capital letters.

Selecting specific fields from a table

In order to select one or several specific fields from a given table, under SQL tab, the query needed would have to tell to select specific fields using grave accents (separated with commas if needed) from a certain table.

```
SELECT `username`, `password` FROM `users`
```

Depending on how many entries a database contains, you might want to limit the number of results to show using the LIMIT command. The OFFSET command may be used to skip some entries from the results.

```
SELECT * FROM `users` LIMIT 5 OFFSET 5
```

Sorting the results

When selecting entries from a table, it is convenient to be able to show the result in a certain order (e.g.: ID number or alphabetical order). To do so, the ORDER BY command is used. The DESC command (standing for "descending") may be used to display the entries from the highest to the smallest id.

```
SELECT * FROM `users` ORDER BY `id` DESC LIMIT 10
```

Nota:

LIMIT must be written after ORDER BY or the query would generate an error.

				id	username	password	user_email	first_name	last_name
<input type="checkbox"/>	Edit	Copy	Delete	15	wreckitralph	imgonnawreck!t	ralph@wreck.it	Ralph	
<input type="checkbox"/>	Edit	Copy	Delete	14	thejedimasterluke	thef0rc3	luceskywalker@thejedicouncil.galaxy	Luke	Skywalker
<input type="checkbox"/>	Edit	Copy	Delete	13	greg0812	sup3rstr0ngp455!	greg@fakemail.com	Greg	Coltman
<input type="checkbox"/>	Edit	Copy	Delete	12	cartman	thecoon!	cartman@southpark.show	Cartman	
<input type="checkbox"/>	Edit	Copy	Delete	11	kenny	foreverdead	kenny@southpark.show	Kenny	
<input type="checkbox"/>	Edit	Copy	Delete	10	james	jamesbond007	james@mi6.gov.uk	James	Bond
<input type="checkbox"/>	Edit	Copy	Delete	9	thezuckmeister	zucker!@9)BNa	markzuckerberg@facebook.com	Mark	Zuckerberg
<input type="checkbox"/>	Edit	Copy	Delete	8	homer	doh!	homer@thesimpsons.com	Homer	Simpson
<input type="checkbox"/>	Edit	Copy	Delete	7	jackie	therealjackiechan	jackie@therealjackiechan.com	Jackie	Chan
<input type="checkbox"/>	Edit	Copy	Delete	6	SuzelleDIY	suzzles1989	suzelle@thediychannel.com	Suzelle	

☐ Check all
 With selected:
 ☐ Edit
 ☐ Copy
 ☐ Delete
 ☐ Export

Refining queries

Let's say you want to check within PRODUCTS table, but you want to know which of the items you have 10 or less of in stock. You then have to use the WHERE command to refine the query. In the example below, we use the *product_name* and *product_stock* from the *products* table to show all of the items we have 10 or less of in stock.

```
SELECT `product_name`, `product_stock` FROM `products` WHERE `product_stock` <= 10
```

Another example of refining search would be to search for a specific entry using "id".

```
SELECT `first_name`, `last_name` FROM `users` WHERE `id` = 4
```