***Basic Concept***

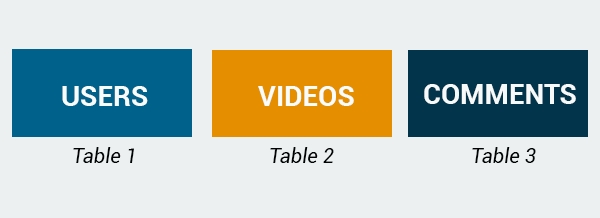
***Introduction to Databases***

**A Database**

A **database** is a collection of data that is organized in a manner that facilitates ease of access, as well as efficient management and updating.

A database is made up of **tables** that store relevant information.

For example, you would use a database, if you were to create a website like YouTube, which contains a lot of information like videos, usernames, passwords, comments.



*In this course we will learn how to create and query databases using SQL!*

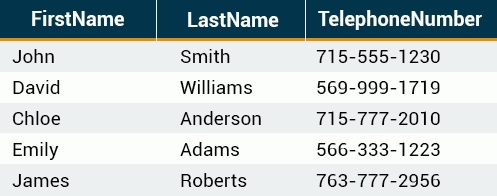
**Database Tables**

A table stores and displays data in a structured format consisting of **columns** and **rows** that are similar to those seen in Excel spreadsheets.

Databases often contain multiple tables, each designed for a specific purpose. For example, imagine creating a database table of names and telephone numbers.

First, we would set up columns with the titles FirstName, LastName and TelephoneNumber.

Each table includes its own set of fields, based on the data it will store.



*A table has a specified number of columns but can have any number of rows.*

**Primary Keys**

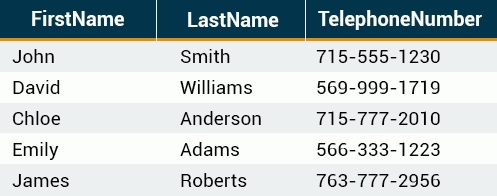
A primary key is a field in the table that uniquely identifies the table records.

The primary key's main features:

- It must contain a **unique value** for each row.

- It cannot contain **NULL** (an absent of value) values.

For example, our table contains a record for each name in a phone book. The unique **ID** number would be a good choice for a primary key in the table, as there is always the chance for more than one person to have the same name.



*- Tables are limited to ONE primary key each.*

*- The primary key's value must be different for each row.*

**What is SQL?**

Once you understand what a database is, understanding SQL is easy. **SQL** stands for Structured Query Language.

**SQL** is used to access and manipulate a **database**.

**MySQL** is a **program** that understands **SQL**.

SQL can:

- insert, update, or delete records in a database.

- create new databases, table, stored procedures, views.

- retrieve data from a database, etc.

*SQL is an ANSI (American National Standards Institute) standard, but there are different versions of the SQL language.*

*Most SQL database programs have their own proprietary extensions in addition to the SQL standard, but all of them support the major commands.*

***SQL Statement: SELECT***

**Basic SQL Commands**

The SQL SHOW statement displays information contained in the database and its tables. This helpful tool lets you keep track of your database contents and remind yourself about the structure of your tables.

For example, the **SHOW DATABASES** command lists the databases managed by the server.

SHOW DATABASES: Returns the list of database on the server.

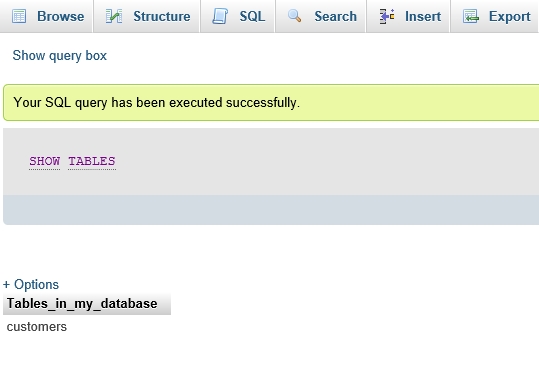
Throughout the course we will be using the MySQL engine and the PHPMyAdmin tool to run SQL queries.

The easiest way to get MySQL and PHPMyAdmin is to install free tools like **MAMP**, **XAMPP**, or **WAMP**, which include all necessary installers.

**Basic SQL Commands**

The SHOW TABLES command is used to display all of the tables in the currently selected MySQL database.

For this example, we have created a database, **my\_database**, with a table called **customers**.



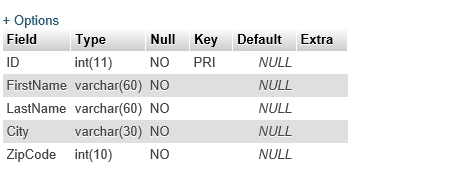
**Basic SQL Commands**

**SHOW COLUMNS** displays information about the columns in a given table.

The following example displays the columns in our **customers** table:

**SHOW COLUMNS FROM** customers

Result:



SHOW COLUMNS displays the following values for each table column:

**Field**: column name

**Type**: column data type

**Key**: indicates whether the column is indexed

**Default**: default value assigned to the column

**Extra**: may contain any additional information that is available about a given column

*The columns for the customers table have also been created using the PHPMyAdmin tool.*

**SELECT Statement**

The **SELECT** statement is used to select data from a database.

The result is stored in a result table, which is called the **result-set**.

A **query** may retrieve information from selected columns or from all columns in the table.

To create a simple SELECT statement, specify the name(s) of the column(s) you need from the table.

Syntax of the SQL SELECT (is used to select data from a database) Statement:

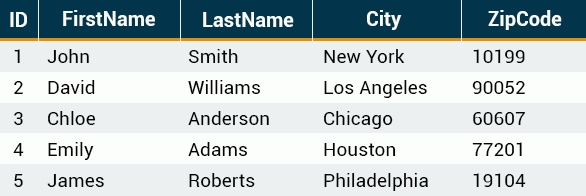
SELECT column\_list

FROM table\_name

- **column\_list** includes one or more columns from which data is retrieved

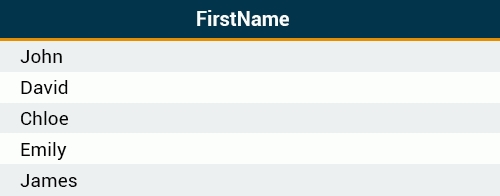
- **table-name** is the name of the table from which the information is retrieved

Below is the data from our customers table:



The following SQL statement selects the **FirstName** from the **customers** table:

**SELECT** FirstName **FROM** customers



*A SELECT statement retrieves zero or more rows from one or more database tables.*

***SQL Syntax Rules***

**Multiple Queries**

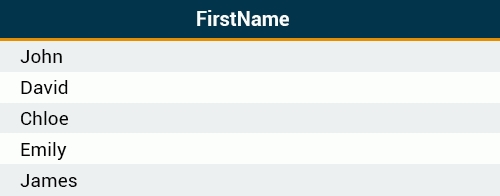
SQL allows to run multiple queries or commands at the same time.

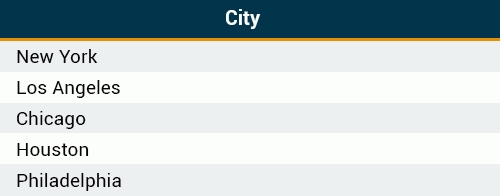
The following SQL statement selects the **FirstName** and **City** columns from the customers table:

SELECT **FirstName** FROM customers;

SELECT **City** FROM customers;

Result:





*Remember to end each SQL statement with a semicolon to indicate that the statement is complete and ready to be interpreted.*

**Case Sensitivity**

SQL is case insensitive.

The following statements are equivalent and will produce the same result:

select City from customers;

SELECT City FROM customers;

sElEct City From customers;

*It is common practice to write all SQL commands in* ***upper-case****.*

**Syntax Rules**

A single SQL statement can be placed on one or more text lines. In addition, multiple SQL statements can be combined on a single text line.

White spaces and multiple lines are ignored in SQL.

For example, the following query is absolutely correct.

SELECT City

FROM customers;

However, it is recommended to avoid unnecessary white spaces and lines.

*Combined with proper spacing and indenting, breaking up the commands into logical lines will make your SQL statements much easier to read and maintain.*

***Selecting Multiple Columns***

As previously mentioned, the SQL SELECT statement retrieves records from tables in your SQL database.

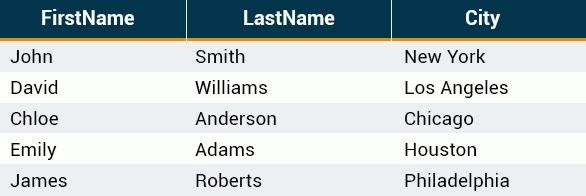
You can select multiple table columns at once.

Just list the column names, separated by commas:

SELECT FirstName, LastName, City

FROM customers;

Result:

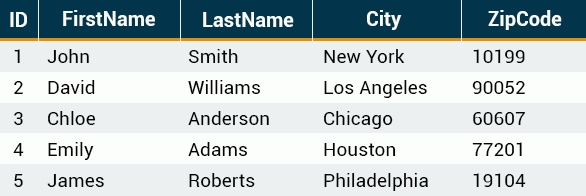


*Do not put a comma after the* ***last*** *column name.*

**Selecting All Columns**

To retrieve all of the information contained in your table, place an **asterisk (\*)** sign after the SELECT command, rather than typing in each column names separately.

The following SQL statement selects all of the columns in the **customers** table:



*In SQL, the asterisk means all.*

***DISTINCT and LIMIT***

**The DISTINCT Keyword**

In situations in which you have multiple duplicate records in a table, it might make more sense to return only unique records, instead of fetching the duplicates.

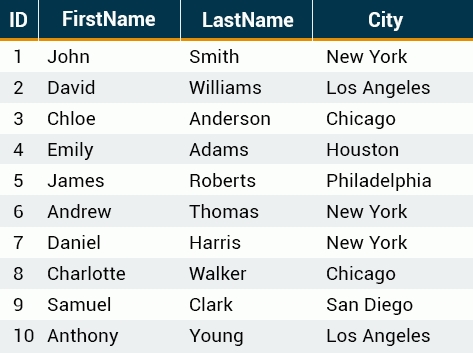
The SQL DISTINCT keyword is used in conjunction with SELECT to eliminate all duplicate records and return only unique ones.

The basic syntax of DISTINCT is as follows:

SELECT DISTINCT column\_name1, column\_name2

FROM table\_name;

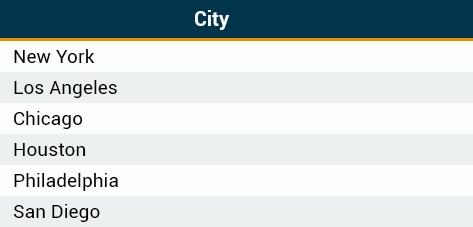
See the **customers** table below:



Note that there are duplicate City names. The following SQL statement selects only distinct values from the City column:

SELECT DISTINCT City FROM customers;

This would produce the following result. Duplicate entries have been removed.



*The DISTINCT keyword only fetches the unique values.*

**The LIMIT Keyword**

By default, all results that satisfy the conditions specified in the SQL statement are returned. However, sometimes we need to retrieve just a subset of records. In MySQL, this is accomplished by using the LIMIT keyword.

The syntax for LIMIT is as follows:

SELECT column list

FROM table\_name

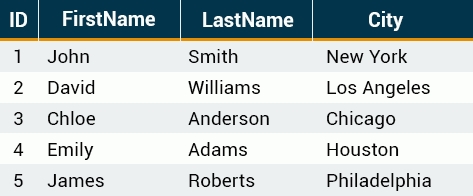
**LIMIT** [number of records];

For example, we can retrieve the first five records from the customers table.

SELECT ID, FirstName, LastName, City

FROM customers LIMIT 5;

This would produce the following result:



*By default, all results that satisfy the conditions specified in the SQL statement are returned.*

**The LIMIT Keyword**

You can also pick up a set of records from a particular **offset**.

In the following example, we pick up **four** records, starting from the **third** position:

SELECT ID, FirstName, LastName, City

FROM customers **LIMIT 3, 4**;

This would produce the following result:



*The reason that it produces results starting from ID number four, and not three, is that MySQL starts counting from* ***zero****, meaning that the offset of the first row is 0, not 1.*

***Sorting Results***

Fully Qualified Names

In SQL, you can provide the table name prior to the column name, by separating them with a **dot**.

The following statements are equivalent:

SELECT City FROM customers;

SELECT **customers**.City FROM customers;

The term for the above-mentioned syntax is called the **"fully qualified name"** of that column.

*This form of writing is especially useful when working with multiple tables that may share the same column names.*

**Order By**

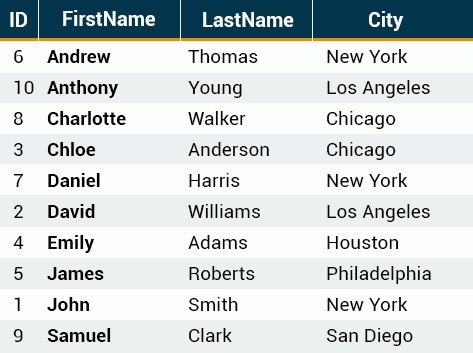
ORDER BY is used with SELECT to sort the returned data.

The following example sorts our **customers** table by the FirstName column.

SELECT \* FROM customers

ORDER BY FirstName;

Result:



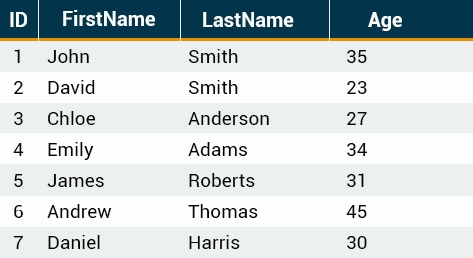
As you can see, the rows are ordered **alphabetically** by the **FirstName** column.

*By default, the ORDER BY keyword sorts the results in ascending order.*

**Sorting Multiple Columns**

ORDER BY can sort retrieved data by multiple columns. When using ORDER BY with more than one column, separate the list of columns to follow ORDER BY with commas.

Here is the customers table, showing the following records:

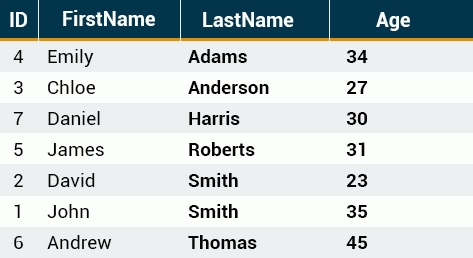


To order by LastName and Age:

SELECT \* FROM customers

ORDER BY LastName, Age;

This ORDER BY statement returns the following result:



As we have two **Smiths**, they will be ordered by the **Age** column in ascending order.

*The ORDER BY command starts ordering in the same sequence as the columns. It will order by the first column listed, then by the second, and so on.*