# SQL Tutorial 1 – Basic Select Statements

This tutorial assumes you know who to open *phpMyAdmin*, navigate your way around it and have a database created that you can work with.

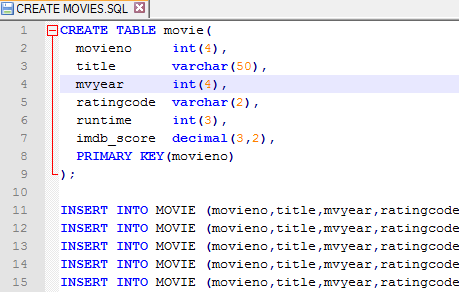
## SETUP

This tutorial requires the file ‘**CREATE MOVIES.SQL**’. This file contains a table with data that all questions require you to work with. There are two ways to import this data into your database.

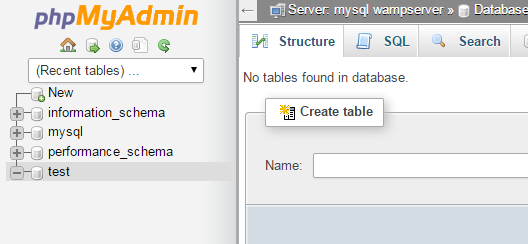
(Please note this file was error free at the time of writing this tutorial, let’s hope your teacher has checked it just in case.)

**Copy and Paste Method**

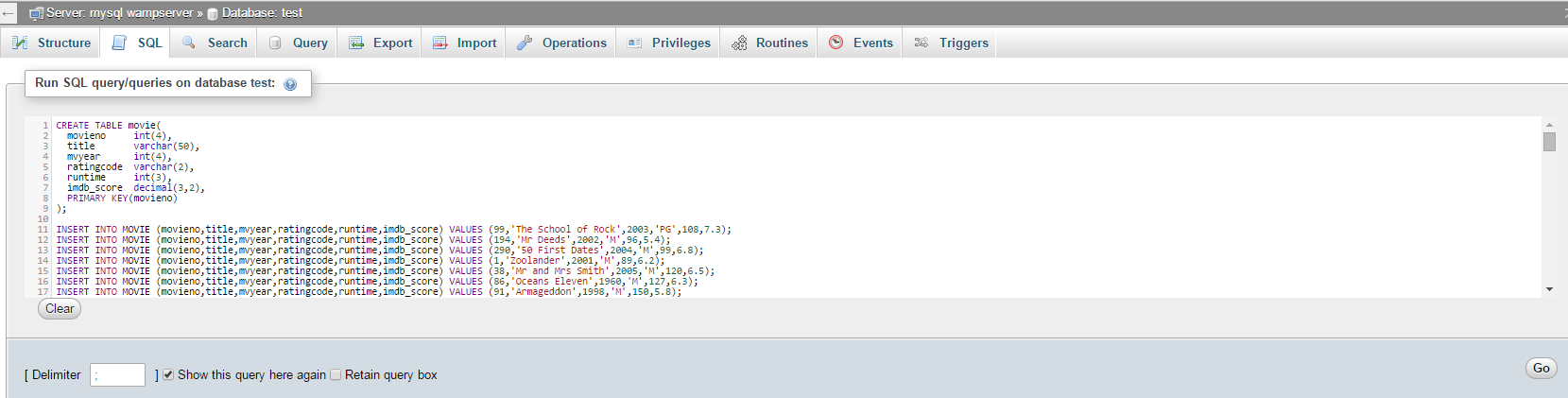
Open up the .sql file using a text editor of your choice. (I prefer notepad++.) In this file you will see one create statement and 126 insert statements. (More on those statements in later tutorials.)



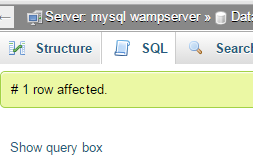
Select and copy all of the SQL staments, go to *phpMyAdmin* and select your database. In my case I have the *test* database selected. Click on the SQL tab.



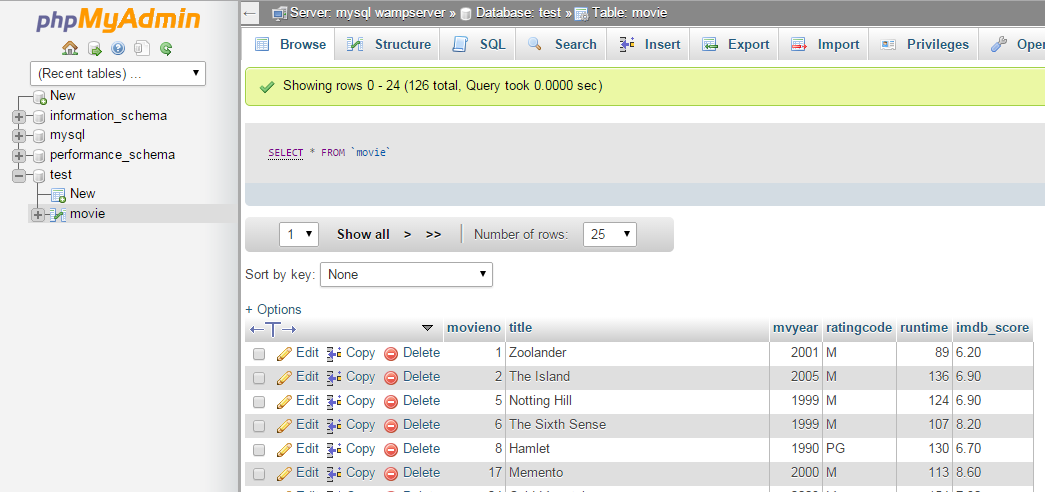
Copy the contents of the .sql file into the text box provided and then click go.



You should see a message in a green box showing everything was successful.

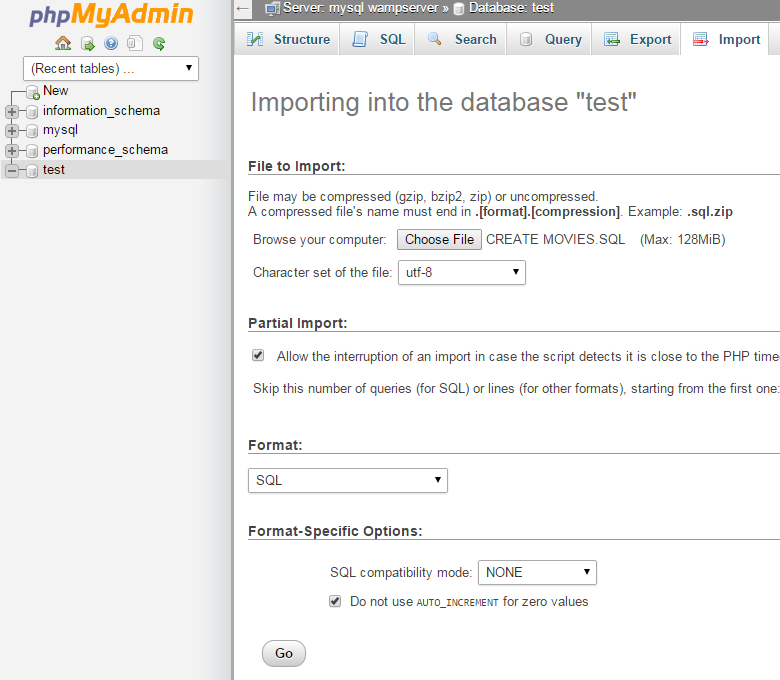


To see if it was successful refresh the navigation tab. You should see a new table called *movie* in your database. If you click on the browse tab you can see the contents of the table. (Make sure you have the table selected first.)



**Import Method.**

*phpMyAdmin* also has a handed Import function that allows you to import .sql files. Go o *phpMyAdmin* and select your database. In my case I have the *test* database selected. Click on the Import tab.

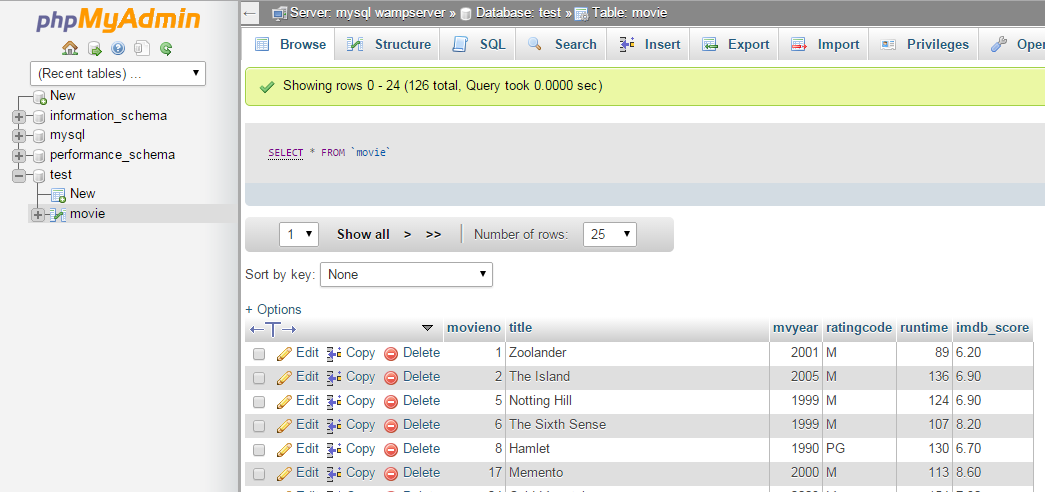


A new page will load. Click on chose file and navigate to the .sql file you wish to import. Once the .sql file is selected click go. A green success message should appear.



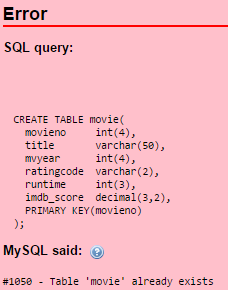
To see if it was successful refresh the navigation tab. You should see a new table called *movie* in your database. If you click on the browse tab you can see the contents of the table. (Make sure you have the table selected first.)

See picture on next page.

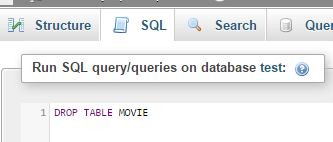


**In case you want to re import the data. (Dropping tables, watch your feet.)**

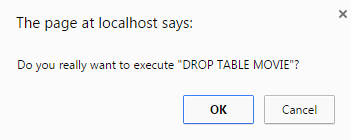
The great thing about these tutorials is if you feel that you have messed up completely you can reimport the table and start from fresh. But if you import the data a second time using any of the methods above you will get a nasty error message.



To avoid this message you will need to drop the *movie* table before you try to import it again. Dropping tables is deleting the table. To drop a table go to *phpMyAdmin* and select your database that has the movie table in it and click on the SQL statement. In the text box enter, DROP TABLE MOVIE, and then click on go.



A warning message will appear. Click OK.



A success message will appear.

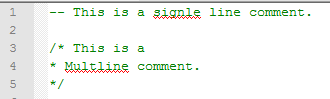


**WARNING:** RDBMS have no undo button. Once you drop a table it is gone for good.

## A few notes before we get started.

I highly recommend you using a text editor that comes with syntax highlighting, like notepad++, this will make reading SQL statements a lot easier. You should also be typing your SQL statements into the text editor and then copy it over to *phpMyAdmin* that way if something goes wrong and you lose the statement you still have a copy of it you can edit instead of having to type it out all over again (Some later statements get big.)

Lastly you can put comments in a .sql file. If your teachers allows you to bring resources into the SAC you could use your .sql files as a handy reference guide if you have written proper comments.

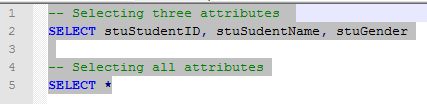


Now on to the actual tutorial.

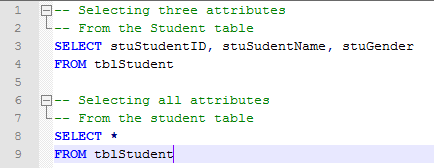
## Basic SQL SELECT Statements

A basic SQL SELECT statement consists of three parts.

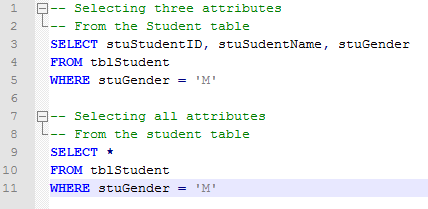
1. First is the key word **SELECT**. In this part of the stamen we state which attributes we want to see once the query has run. In the example below I am selecting the attributes **stuStudentId**, **stuStudentName, stuGender.** If you wanted to select all attributes you can use the \*



1. Second is the key word **FROM** this allows us to state which tables the attributes are to be selected from. For this tutorial you will only be selecting from one table. In the example below I am selecting from the table **tblStudent.**



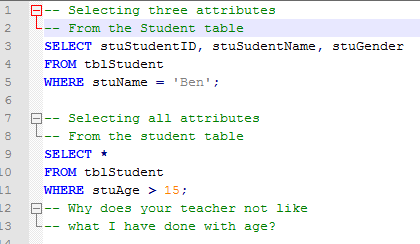
1. At the moment the examples will return all data from the student table. What happens if we wanted to see only the data for male students or for students under the age of 15. We can use the key word **WHERE** for that. In the example below I am only selecting students where the gender is equal to ‘M’



Like in programming we have a selection of comparison operators that we can use. For this tutorial you will only need.

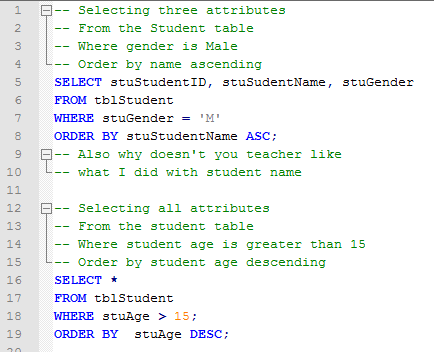
|  |  |
| --- | --- |
| Operator | What it does |
| = | Equals to |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal to |
| <= | Less than or equal to |

When comparing text we need to put single quotation marks around the text. For numbers we don’t add anything around the number. See examples below.



**Ordering SQL statements**

If we want to have the results from the SQL statement ordered in a particular way we can use the ORDER BY key word. In the example below you can see in the first statement that I am ordering by **stuStudentName** in ascending order. In the second statement I am ordering by **stuAge** in descending order.



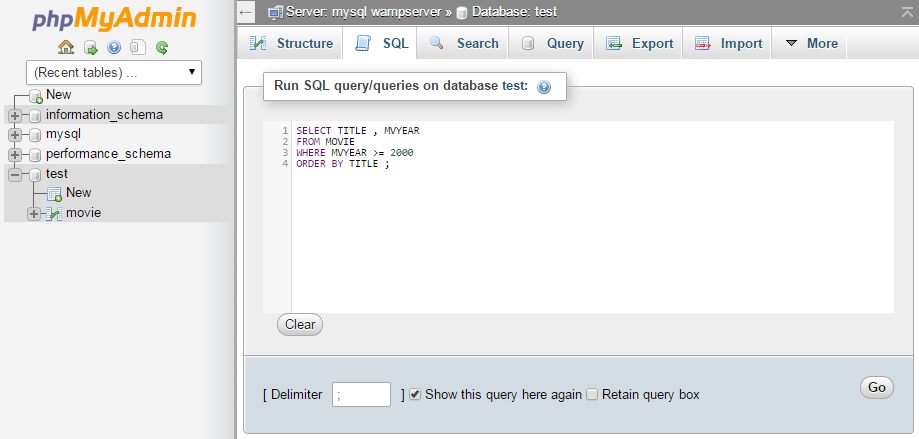
**A few notes.**

If you want to bulk execute SQL statements you will need to end each stamen with a semicolon ‘;’.

Also you do not need to put the key word on a new line but it does make it easier for reading. You may also note that the comments I have written are pesudocode for the actual SQL statements, this is one way of designing SQL statements.

**To execute an SQL statement in *myPhpAdmin.***

Select the database you want to run the SQL statements on. Select the SQL table, enter in the SQL statement and then push go.



## Tutorial Questions.

To do these question you must have imported the ‘**CREATE MOVIES.SQL**’ file. See instructions at the start of this tutorial.

Write and execute the SQL statement to solve the following problems

1. Type the following to display all the column names within the MOVIE table.

Type DESCRIBE MOVIE into SQL stamen box and click go.

1. Write the statement to list only the TITLE and MVYEAR columns of every row.
2. List the MVYEAR and TITLE columns of all rows in ascending MVYEAR sequence.
3. List the MVYEAR and TITLE columns of all rows in ascending MVYEAR sequence.
4. List the MOVIE NO, MVYEAR and TITLE of every row in descending MVYEAR sequence.
5. List TITLE and RATING of all movies that have a PG rating.
6. List TITLE, MVYEAR and IMDB\_SCORE of all movies that are titled HAMLET in descending IMDB\_SCORE sequence
7. List TITLE and MVYEAR of all movies that been made this century. The list must be in ascending Title sequence.
8. List TITLE and RUNTIME of all movies that are shorter than 100 minutes in ascending title sequence.
9. List all details of movies that have an IMDB\_SCORE of 7.8 or higher in ascending MVYEAR sequence.
10. List all columns of every row. Ensure that rows are displayed in MVYEAR / TITLE sequence I.e. The result set will show movies in ascending MVYEAR sequence. If a number of movies were made in the same MVYEAR, then those movies must be listed in title sequence.

|  |  |
| --- | --- |
| MVYEAR | TITLE |
| 1939 | Gone with the wind |
| 1939 | Wizard of Oz |
| 1960  All Rows in MVYEAR sequence | Oceans 11  Within 1960, all Titles are shown in ascending title sequence |
| 1960 | Psycho |
| 1960 | Spartacus |
| 1960 | The Time Machine |
| 1963 | Charade |
| 1963 | Pink Panther |

Etc.