# PHP MySQL Database

With PHP, you can connect to and manipulate databases.

MySQL is the most popular database system used with PHP.

## What is MySQL?

1. MySQL is a database system used on the web
2. MySQL is a database system that runs on a server
3. MySQL is ideal for both small and large applications
4. MySQL is very fast, reliable, and easy to use
5. MySQL uses standard SQL
6. MySQL compiles on a number of platforms
7. MySQL is free to download and use

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Databases are useful for storing information categorically. A company may have a database with the following tables:

1. Employees
2. Products
3. Customers
4. Orders

## PHP + MySQL Database System

1. PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

## Database Queries

A query is a question or a request.

We can query a database for specific information and have a recordset returned.

Look at the following query (using standard SQL):



The query above selects all the data in the "LastName" column from the "Employees" table.

# PHP Connect to MySQL

PHP 5 and later can work with a MySQL database using:

* MySQLi extension (the "i" stands for improved)
* PDO (PHP Data Objects)

## Should I Use MySQLi or PDO?

Both MySQLi and PDO have their advantages:

PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.

So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.

Both are object-oriented, but MySQLi also offers a procedural API.

Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

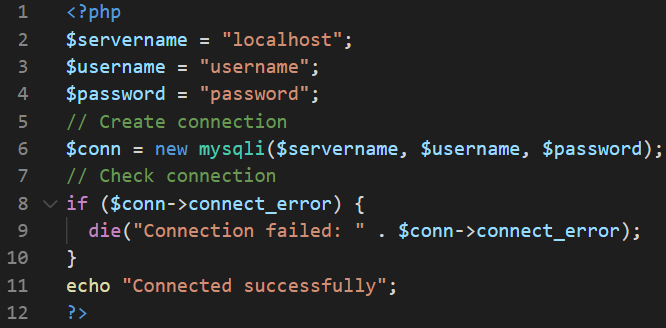
## MySQLi Installation

For Linux and Windows: The MySQLi extension is automatically installed in most cases when php5 mysql package is installed.

## Open a Connection to MySQL

Before we can access data in the MySQL database, we need to be able to connect to the server.

Example:



## Close the Connection

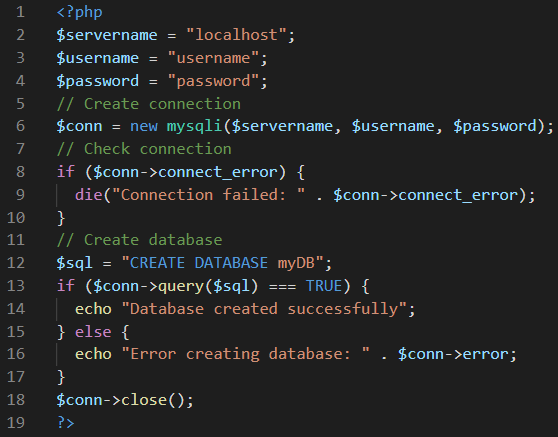
The connection will be closed automatically when the script ends.

# PHP Create a MySQL Database

## Create a MySQL Database Using MySQLi

The CREATE DATABASE statement is used to create a database in MySQL.

The following examples create a database named "myDB":



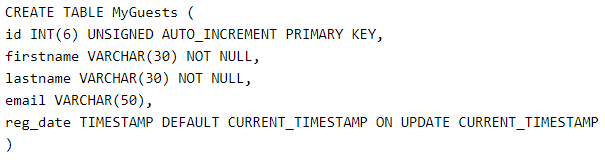
# PHP MySQL Create Table

A database table has its own unique name and consists of columns and rows.

## Create a MySQL Table Using MySQLi

The CREATE TABLE statement is used to create a table in MySQL.

We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg\_date":



Notes on the table above:

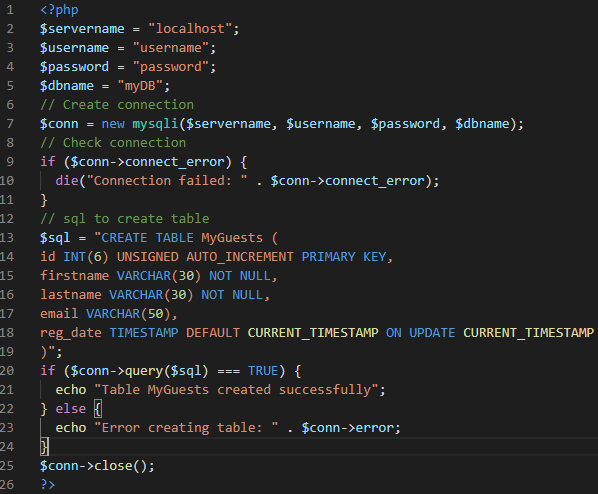
The data type specifies what type of data the column can hold.

After the data type, you can specify other optional attributes for each column:

1. NOT NULL - Each row must contain a value for that column, null values are not allowed
2. DEFAULT value - Set a default value that is added when no other value is passed
3. UNSIGNED - Used for number types, limits the stored data to positive numbers and zero
4. AUTO INCREMENT - MySQL automatically increases the value of the field by 1 each time a new record is added
5. PRIMARY KEY - Used to uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number, and is often used with AUTO\_INCREMENT

Each table should have a primary key column (in this case: the "id" column). Its value must be unique for each record in the table.

The following examples shows how to create the table in PHP:



# PHP MySQL Insert Data

## Insert Data Into MySQL Using MySQLi

After a database and a table have been created, we can start adding data in them.

Here are some syntax rules to follow:

1. The SQL query must be quoted in PHP
2. String values inside the SQL query must be quoted
3. Numeric values must not be quoted
4. The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table.

The following examples add a new record to the "MyGuests" table:

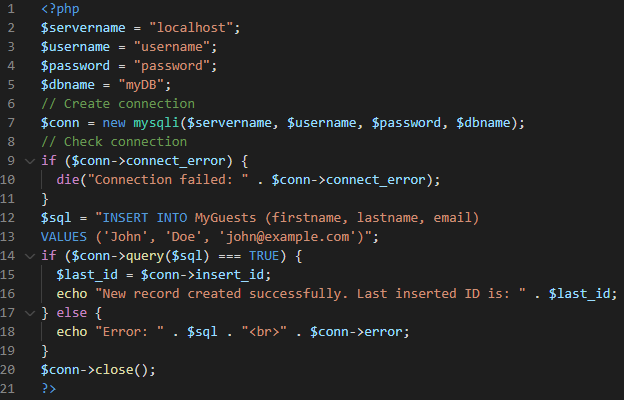


# PHP MySQL Get Last Inserted ID

## Get ID of The Last Inserted Record

If we perform an INSERT or UPDATE on a table with an AUTO\_INCREMENT field, we can get the ID of the last inserted/updated record immediately.

Example:

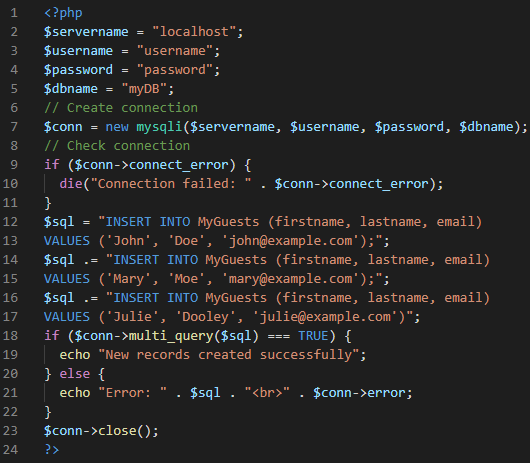


# PHP MySQL Insert Multiple Records

## Insert Multiple Records Into MySQL Using MySQLi

Multiple SQL statements must be executed with the mysqli\_multi\_query() function.

The following examples add three new records to the "MyGuests" table:



# PHP MySQL Prepared Statements

## Prepared Statements and Bound Parameters

A prepared statement is a feature used to execute the same (or similar) SQL statements repeatedly with high efficiency.

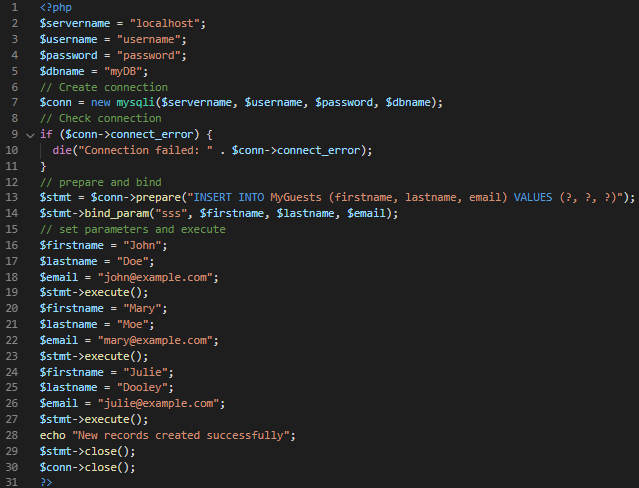
Prepared statements basically work like this:

1. Prepare: An SQL statement template is created and sent to the database. Certain values are left unspecified, called parameters (labeled "?"). Example: INSERT INTO MyGuests VALUES(?, ?, ?)
2. The database parses, compiles, and performs query optimization on the SQL statement template, and stores the result without executing it
3. Execute: At a later time, the application binds the values to the parameters, and the database executes the statement. The application may execute the statement as many times as it wants with different values

Compared to executing SQL statements directly, prepared statements have three main advantages:

1. Prepared statements reduce parsing time as the preparation on the query is done only once (although the statement is executed multiple times)
2. Bound parameters minimize bandwidth to the server as you need send only the parameters each time, and not the whole query
3. Prepared statements are very useful against SQL injections, because parameter values, which are transmitted later using a different protocol, need not be correctly escaped. If the original statement template is not derived from external input, SQL injection cannot occur.

The following example uses prepared statements and bound parameters in MySQLi:



Example Explained

1. In our SQL, we insert a question mark (?) where we want to substitute in an integer, string, double, or blob value. Then, have a look at the bind\_param() function
2. This function binds the parameters to the SQL query and tells the database what the parameters are. The "sss" argument lists the types of data that the parameters are. The s character tells mysql that the parameter is a string.
3. The argument may be one of four types:
   1. i - integer
   2. d - double
   3. s - string
   4. b - BLOB
4. By telling mysql what type of data to expect, we minimize the risk of SQL injections.

# PHP MySQL Select Data

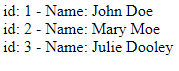
## Select Data From a MySQL Database

The SELECT statement is used to select data from one or more tables.

The following example selects the id, firstname and lastname columns from the MyGuests table and displays it on the page:



The output of the above code is as follows:



Code lines to explain from the example above:

First, we set up an SQL query that selects the id, firstname and lastname columns from the MyGuests table. The next line of code runs the query and puts the resulting data into a variable called $result.

Then, the function num\_rows() checks if there are more than zero rows returned.

If there are more than zero rows returned, the function fetch\_assoc() puts all the results into an associative array that we can loop through. The while() loop loops through the result set and outputs the data from the id, firstname and lastname columns.

# PHP MySQL Use The WHERE Clause

## Select and Filter Data From a MySQL Database

1. The WHERE clause is used to filter records.
2. The WHERE clause is used to extract only those records that fulfill a specified condition.

The following example selects the id, firstname and lastname columns from the MyGuests table where the lastname is "Doe", and displays it on the page:



id: 1 - Name: John Doe will be displayed as the output for the above code.

Code lines to explain from the example above:

First, we set up the SQL query that selects the id, firstname and lastname columns from the MyGuests table where the lastname is "Doe". The next line of code runs the query and puts the resulting data into a variable called $result.

Then, the function num\_rows() checks if there are more than zero rows returned.

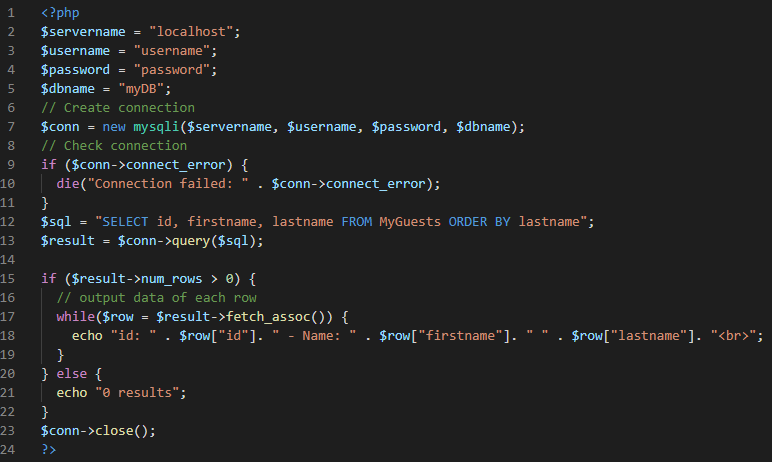
If there are more than zero rows returned, the function fetch\_assoc() puts all the results into an associative array that we can loop through. The while() loop loops through the result set and outputs the data from the id, firstname and lastname columns.

# PHP MySQL Use The ORDER BY Clause

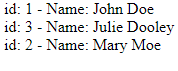
## Select and Order Data From a MySQL Database

1. The ORDER BY clause is used to sort the result-set in ascending or descending order.
2. The ORDER BY clause sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

The following example selects the id, firstname and lastname columns from the MyGuests table. The records will be ordered by the lastname column:



The output for the above code is as follows:



Code lines to explain from the example above:

First, we set up the SQL query that selects the id, firstname and lastname columns from the MyGuests table. The records will be ordered by the lastname column. The next line of code runs the query and puts the resulting data into a variable called $result.

Then, the function num\_rows() checks if there are more than zero rows returned.

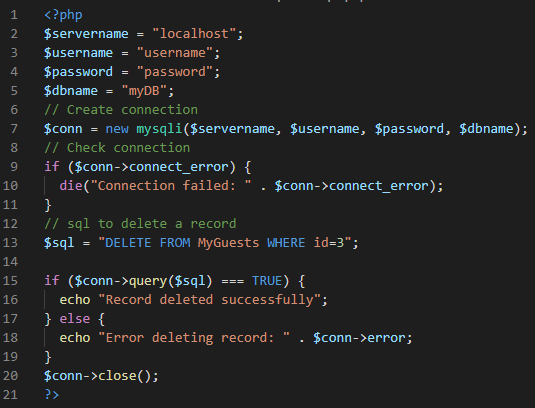
If there are more than zero rows returned, the function fetch\_assoc() puts all the results into an associative array that we can loop through. The while() loop loops through the result set and outputs the data from the id, firstname and lastname columns.

# PHP MySQL Delete Data

## Delete Data From a MySQL Table Using MySQLi

The DELETE statement is used to delete records from a table.

The following examples delete the record with id=3 in the "MyGuests" table:

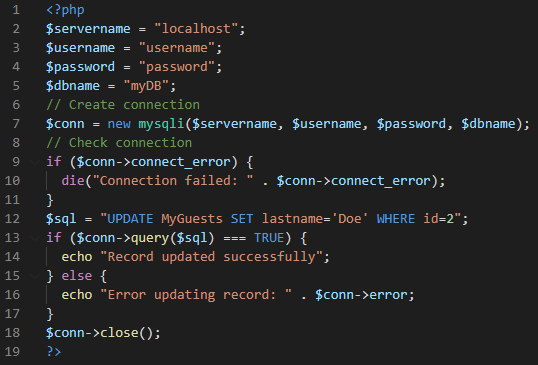


# PHP MySQL Update Data

## Update Data In a MySQL Table Using MySQLi

The UPDATE statement is used to update existing records in a table.

The following examples update the record with id=2 in the "MyGuests" table:



# PHP MySQL Limit Data Selections

## Limit Data Selections From a MySQL Database

MySQL provides a LIMIT clause that is used to specify the number of records to return.

The LIMIT clause makes it easy to code multi page results or pagination with SQL, and is very useful on large tables. Returning a large number of records can impact on performance.

-----------------------Reference credits: w3schools,Tutorispoint, [https://github.com/hegdenaveen1/Become-A-Full-Stack-Web-Developer#start-here](https://www.google.com/url?q=https://github.com/hegdenaveen1/Become-A-Full-Stack-Web-Developer%23start-here&sa=D&source=hangouts&ust=1599916002593000&usg=AFQjCNFqAEgym0xUC_MkOX8mYy5AtiOv1g) -------------------