

Study Unit 14: MYSQL DATABASE

Outline

Student will be able to learn about MySQL database.

- PHP DB Connectivity
- Introduction to MYSQL.
- Insert Form Data into database
- Select From Data From Database
- Update From data from database
- Delete data from Database

Learning Outcomes of Study Unit 14

Upon completion of this study unit, Student will learn about MySQL database and its Queries related to insert, update, delete, select and the PHP connectivity with MySQL database.

MySQL

1. Connecting to MySQL database – Learn how to use PHP to open and close a MySQL database connection.
2. Create MySQL Database Using PHP – This part explains how to create MySQL database and tables using PHP.
3. Delete MySQL Database Using PHP – This part explains how to delete MySQL database and tables using PHP.
4. Insert Data To MySQL Database – Once you have created your database and tables then you would like to insert your data into created tables. This session will take you through real example on data insert.
5. Retrieve Data From MySQL Database – Learn how to fetch records from MySQL database using PHP.
6. Updating Data Into MySQL Database – This part explains how to update existing records into MySQL database using PHP.
7. Deleting Data From MySQL Database – This part explains how to delete or purge existing records from MySQL database using PHP.

(1)

PHP provides **mysql_connect** function to open a database connection. This function takes five parameters and returns a MySQL link identifier on success, or FALSE on failure.

Syntax

```
connection mysql_connect(server,user,passwd,new_link,client_flag);
```

Sr.No	Parameter & Description
1	Server Optional – The host name running database server. If not specified then default value is localhost:3306 .
2	User Optional – The username accessing the database. If not specified then default is the name of the user that owns the server process.
3	passwd Optional – The password of the user accessing the database. If not specified then default is an empty password.

4	<p>new_link</p> <p>Optional – If a second call is made to <code>mysql_connect()</code> with the same arguments, no new connection will be established; instead, the identifier of the already opened connection will be returned.</p>
5	<p>client_flags</p> <p>Optional – A combination of the following constants –</p> <ul style="list-style-type: none"> • MYSQL_CLIENT_SSL – Use SSL encryption • MYSQL_CLIENT_COMPRESS – Use compression protocol • MYSQL_CLIENT_IGNORE_SPACE – Allow space after function names • MYSQL_CLIENT_INTERACTIVE – Allow interactive timeout seconds of inactivity before closing the connection

NOTE – You can specify server, user, passwd in **php.ini** file instead of using them again and again in your every PHP scripts. Check [php.ini file](#) configuration.

Its simplest function **mysql_close** PHP provides to close a database connection. This function takes connection resource returned by `mysql_connect` function. It returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_close ( resource $link_identifier );
```

If a resource is not specified then last opened database is closed.

Example

Try out following example to open and close a database connection –

```
<?php

$dbhost = 'localhost:3036';
$dbuser = 'guest';
$dbpass = 'guest123';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

echo 'Connected successfully';
mysql_close($conn);
?>
```

(2)

To create and delete a database you should have admin privilege. Its very easy to create a new MySQL database. PHP uses **mysql_query** function to create a MySQL database. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_query( sql, connection );
```

Sr.No	Parameter & Description
1	sql Required - SQL query to create a database
2	connection Optional - if not specified then last opened connection by mysql_connect will be used.

Example

Try out following example to create a database –

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
```

```
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

echo 'Connected successfully';

$sql = 'CREATE Database test_db';
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not create database: ' . mysql_error());
}

echo "Database test_db created successfully\n";
mysql_close($conn);
?>
```

Once you establish a connection with a database server then it is required to select a particular database where your all the tables are associated.

This is required because there may be multiple databases residing on a single server and you can do work with a single database at a time.

PHP provides function **mysql_select_db** to select a database. It returns TRUE on success or FALSE on failure.

Syntax

```
bool mysql_select_db( db_name, connection );
```

Sr.No	Parameter & Description
1	db_name Required - Database name to be selected
2	connection Optional - if not specified then last opened connection by mysql_connect will be used.

Example

Here is the example showing you how to select a database.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'guest';
$dbpass = 'guest123';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

echo 'Connected successfully';

mysql_select_db( 'test_db' );
```

```
mysql_close($conn);
```

```
?>
```

To create tables in the new database you need to do the same thing as creating the database. First create the SQL query to create the tables then execute the query using `mysql_query()` function.

Example

Try out following example to create a table –

```
<?php

$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

echo 'Connected successfully';

$sql = 'CREATE TABLE employee( '.
    'emp_id INT NOT NULL AUTO_INCREMENT, '.
    'emp_name VARCHAR(20) NOT NULL, '.
    'emp_address VARCHAR(20) NOT NULL, '.
    'emp_salary INT NOT NULL, ';
```



```
'join_date    timestamp(4) NOT NULL, '.
'primary key ( emp_id ));
mysql_select_db('test_db');
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not create table: ' . mysql_error());
}

echo "Table employee created successfully\n";

mysql_close($conn);
?>
```

In case you need to create many tables then its better to create a text file first and put all the SQL commands in that text file and then load that file into \$sql variable and excute those commands.

Consider the following content in sql_query.txt file

```
CREATE TABLE employee(
    emp_id INT NOT NULL AUTO_INCREMENT,
    emp_name VARCHAR(20) NOT NULL,
    emp_address  VARCHAR(20) NOT NULL,
    emp_salary   INT NOT NULL,
    join_date    timestamp(14) NOT NULL,
    primary key ( emp_id ));
```

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
```

```
if(! $conn ) {  
    die('Could not connect: ' . mysql_error());  
}  
  
$query_file = 'sql_query.txt';  
  
$fp = fopen($query_file, 'r');  
$sql = fread($fp, filesize($query_file));  
fclose($fp);  
  
mysql_select_db('test_db');  
$retval = mysql_query( $sql, $conn );  
  
if(! $retval ) {  
    die('Could not create table: ' . mysql_error());  
}  
  
echo "Table employee created successfully\n";  
mysql_close($conn);  
?>
```

(3)

If a database is no longer required then it can be deleted forever. You can use pass an SQL command to **mysql_query** to delete a database.

Example

Try out following example to drop a database.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'DROP DATABASE test_db';
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not delete database db_test: ' . mysql_error());
}

echo "Database deleted successfully\n";

mysql_close($conn);
?>
```

WARNING – its very dangerous to delete a database and any table. So before deleting any table or database you should make sure you are doing everything intentionally.

Its again a matter of issuing one SQL command through **mysql_query** function to delete any database table. But be very careful while using this command because by doing so you can delete some important information you have in your table.

Example

Try out following example to drop a table –

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'DROP TABLE employee';
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not delete table employee: ' . mysql_error());
}

echo "Table deleted successfully\n";

mysql_close($conn);
?>
```

(4) Insert Data into MySQL Database

Data can be entered into MySQL tables by executing SQL INSERT statement through PHP function **mysql_query**. Below a simple example to insert a record into **employee** table.

Try out following example to insert record into employee table.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'INSERT INTO employee '.
    '(emp_name,emp_address, emp_salary, join_date) '.
    'VALUES ( "guest", "XYZ", 2000, NOW() )';

mysql_select_db('test_db');
```

```
$retval = mysql_query( $sql, $conn );  
  
if(! $retval ) {  
    die('Could not enter data: ' . mysql_error());  
}  
  
echo "Entered data successfully\n";  
  
mysql_close($conn);  
?>
```

In real application, all the values will be taken using HTML form and then those values will be captured using PHP script and finally they will be inserted into MySQL tables.

While doing data insert its best practice to use function **get_magic_quotes_gpc()** to check if current configuration for magic quote is set or not. If this function returns false then use function **addslashes()** to add slashes before quotes.

TXT FILE

INSERT INTO employee

(emp_name,emp_address, emp_salary, join_date)

VALUES ("guest", "XYZ", 2000, NOW());

Try out this example by putting this code into add_employee.php, this will take input using HTML Form and then it will create records into database.

```
<html>
```

```
<head>
  <title>Add New Record in MySQL Database</title>
</head>

<body>
  <?php
    if(isset($_POST['add'])) {
      $dbhost = 'localhost:3036';
      $dbuser = 'root';
      $dbpass = 'rootpassword';
      $conn = mysql_connect($dbhost, $dbuser, $dbpass);

      if(! $conn ) {
        die('Could not connect: ' . mysql_error());
      }

      if(! get_magic_quotes_gpc() ) {
        $emp_name = addslashes ($_POST['emp_name']);
        $emp_address = addslashes ($_POST['emp_address']);
      }else {
        $emp_name = $_POST['emp_name'];
        $emp_address = $_POST['emp_address'];
      }

      $emp_salary = $_POST['emp_salary'];

      $sql = "INSERT INTO employee ". "(emp_name,emp_address, emp_salary,
        join_date) ". "VALUES('$emp_name','$emp_address',$emp_salary, NOW())";
```

```
mysql_select_db('test_db');
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not enter data: ' . mysql_error());
}

echo "Entered data successfully\n";

mysql_close($conn);
}else {
    ?>

    <form method = "post" action = "<?php $_PHP_SELF ?>">
        <table width = "400" border = "0" cellpadding = "1"
            cellspacing = "2">

            <tr>
                <td width = "100">Employee Name</td>
                <td><input name = "emp_name" type = "text"
                    id = "emp_name"></td>
            </tr>

            <tr>
                <td width = "100">Employee Address</td>
                <td><input name = "emp_address" type = "text"
                    id = "emp_address"></td>
            </tr>

            <tr>
```



```
        <td width = "100">Employee Salary</td>
        <td><input name = "emp_salary" type = "text"
            id = "emp_salary"></td>
    </tr>

    <tr>
        <td width = "100"> </td>
        <td> </td>
    </tr>

    <tr>
        <td width = "100"> </td>
        <td>
            <input name = "add" type = "submit" id = "add"
                value = "Add Employee">
        </td>
    </tr>

</table>
</form>

<?php
}
?>

</body>
</html>
```

(5) Getting Data From MySQL Database

Data can be fetched from MySQL tables by executing SQL SELECT statement through PHP function `mysql_query`. You have several options to fetch data from MySQL.

The most frequently used option is to use function **`mysql_fetch_array()`**. This function returns row as an associative array, a numeric array, or both. This function returns FALSE if there are no more rows.

Below is a simple example to fetch records from **employee** table.

Try out following example to display all the records from employee table.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';

$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'SELECT emp_id, emp_name, emp_salary FROM employee';
mysql_select_db('test_db');
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not get data: ' . mysql_error());
}
```

```
}

while($row = mysql_fetch_array($retval, MYSQL_ASSOC)) {
    echo "EMP ID :{$row['emp_id']} <br> ".
        "EMP NAME : {$row['emp_name']} <br> ".
        "EMP SALARY : {$row['emp_salary']} <br> ".
        "-----<br>";
}

echo "Fetched data successfully\n";

mysql_close($conn);
?>
```

The content of the rows are assigned to the variable \$row and the values in row are then printed.

NOTE – Always remember to put curly brackets when you want to insert an array value directly into a string.

In above example the constant **MYSQL_ASSOC** is used as the second argument to `mysql_fetch_array()`, so that it returns the row as an associative array. With an associative array you can access the field by using their name instead of using the index.

PHP provides another function called **mysql_fetch_assoc()** which also returns the row as an associative array.

Try out following example to display all the records from employee table using `mysql_fetch_assoc()` function.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
```

```
$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'SELECT emp_id, emp_name, emp_salary FROM employee';
mysql_select_db('test_db');
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not get data: ' . mysql_error());
}

while($row = mysql_fetch_assoc($retval)) {
    echo "EMP ID :{$row['emp_id']} <br> ".
        "EMP NAME : {$row['emp_name']} <br> ".
        "EMP SALARY : {$row['emp_salary']} <br> ".
        "-----<br>";
}

echo "Fetched data successfully\n";

mysql_close($conn);
?>
```

You can also use the constant **MYSQL_NUM**, as the second argument to `mysql_fetch_array()`. This will cause the function to return an array with numeric index.

Try out following example to display all the records from employee table using MYSQL_NUM argument.

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';

$conn = mysql_connect($dbhost, $dbuser, $dbpass);

if(! $conn ) {
    die('Could not connect: ' . mysql_error());
}

$sql = 'SELECT emp_id, emp_name, emp_salary FROM employee';
mysql_select_db('test_db');
$retval = mysql_query( $sql, $conn );

if(! $retval ) {
    die('Could not get data: ' . mysql_error());
}

while($row = mysql_fetch_array($retval, MYSQL_NUM)) {
    echo "EMP ID :{$row[0]} <br> ".
        "EMP NAME : {$row[1]} <br> ".
        "EMP SALARY : {$row[2]} <br> ".
        "-----<br>";
}

echo "Fetched data successfully\n";
```

```
mysql_close($conn);  
?>
```

All the above three examples will produce same result.

Its a good practice to release cursor memory at the end of each SELECT statement. This can be done by using PHP function **mysql_free_result()**. Below is the example to show how it has to be used.

Example

Try out following example

```
<?php  
$dbhost = 'localhost:3036';  
$dbuser = 'root';  
$dbpass = 'rootpassword';  
  
$conn = mysql_connect($dbhost, $dbuser, $dbpass);  
  
if(! $conn ) {  
    die('Could not connect: ' . mysql_error());  
}  
  
$sql = 'SELECT emp_id, emp_name, emp_salary FROM employee';  
mysql_select_db('test_db');  
$retval = mysql_query( $sql, $conn );  
  
if(! $retval ) {
```

```
        die('Could not get data: ' . mysql_error());
    }

    while($row = mysql_fetch_array($retval, MYSQL_NUM)) {
        echo "EMP ID :{$row[0]} <br> ".
            "EMP NAME : {$row[1]} <br> ".
            "EMP SALARY : {$row[2]} <br> ".
            "-----<br>";
    }

    mysql_free_result($retval);
    echo "Fetched data successfully\n";

    mysql_close($conn);
?>
```

While fetching data you can write as complex SQL as you like. Procedure will remain same as mentioned above.

(6) Updating Data into MySQL Database

Data can be updated into MySQL tables by executing SQL UPDATE statement through PHP function **mysql_query**.

Below is a simple example to update records into **employee** table. To update a record in any table it is required to locate that record by using a conditional clause. Below example uses primary key to match a record in employee table.

Try out following example to understand update operation. You need to provide an employee ID to update an employee salary.

```
<html>

<head>
  <title>Update a Record in MySQL Database</title>
</head>

<body>
  <?php
    if(isset($_POST['update'])) {
      $dbhost = 'localhost:3036';
      $dbuser = 'root';
      $dbpass = 'rootpassword';

      $conn = mysql_connect($dbhost, $dbuser, $dbpass);

      if(! $conn ) {
        die('Could not connect: ' . mysql_error());
      }

      $emp_id = $_POST['emp_id'];
      $emp_salary = $_POST['emp_salary'];

      $sql = "UPDATE employee ". "SET emp_salary = $emp_salary ".
        "WHERE emp_id = $emp_id" ;
      mysql_select_db('test_db');
      $retval = mysql_query( $sql, $conn );

      if(! $retval ) {
        die('Could not update data: ' . mysql_error());
      }
    }
  }
}
```



```
echo "Updated data successfully\n";

mysql_close($conn);
}else {
    ?>
    <form method = "post" action = "<?php $_PHP_SELF ?>">
        <table width = "400" border = " 0" cellpadding = "1"
            cellspacing = "2">

            <tr>
                <td width = "100">Employee ID</td>
                <td><input name = "emp_id" type = "text"
                    id = "emp_id"></td>
            </tr>

            <tr>
                <td width = "100">Employee Salary</td>
                <td><input name = "emp_salary" type = "text"
                    id = "emp_salary"></td>
            </tr>

            <tr>
                <td width = "100"> </td>
                <td> </td>
            </tr>

            <tr>
                <td width = "100"> </td>
                <td>
                    <input name = "update" type = "submit"
```

```
        id = "update" value = "Update">
    </td>
</tr>

</table>
</form>
<?php
}
?>

</body>
</html>
```

(7)Deleting Data from MySQL Database

Data can be deleted from MySQL tables by executing SQL DELETE statement through PHP function **mysql_query**.

Below is a simple example to delete records into **employee** table. To delete a record in any table it is required to locate that record by using a conditional clause. Below example uses primary key to match a record in employee table.

Try out following example to understand delete operation. You need to provide an employee ID to delete an employee record from employee table.

```
<html>

<head>
```

```
<title>Delete a Record from MySQL Database</title>
</head>

<body>
    <?php
        if(isset($_POST['delete'])) {
            $dbhost = 'localhost:3036';
            $dbuser = 'root';
            $dbpass = 'rootpassword';
            $conn = mysql_connect($dbhost, $dbuser, $dbpass);

            if(! $conn ) {
                die('Could not connect: ' . mysql_error());
            }

            $emp_id = $_POST['emp_id'];

            $sql = "DELETE FROM employee WHERE emp_id = $emp_id" ;
            mysql_select_db('test_db');
            $retval = mysql_query( $sql, $conn );

            if(! $retval ) {
                die('Could not delete data: ' . mysql_error());
            }

            echo "Deleted data successfully\n";

            mysql_close($conn);
        }else {
            ?>
```

```
<form method = "post" action = "<?php $_PHP_SELF ?>">
  <table width = "400" border = "0" cellspacing = "1"
    cellpadding = "2">

    <tr>
      <td width = "100">Employee ID</td>
      <td><input name = "emp_id" type = "text"
        id = "emp_id"></td>
    </tr>

    <tr>
      <td width = "100"> </td>
      <td> </td>
    </tr>

    <tr>
      <td width = "100"> </td>
      <td>
        <input name = "delete" type = "submit"
          id = "delete" value = "Delete">
      </td>
    </tr>

  </table>
</form>
<?php
}
?>

</body>
```

```
</html>
```

TEXT file

```
UPDATE employee SET emp_salary = '10000'  
WHERE emp_id = 2;
```

Self-Review Questions (SRQ) For Study Session 14

Now that you have completed this study unit, you can assess how well you have achieved its Learning Outcomes by answering these questions. Write your answers in your Study Diary and discuss them with your Tutor at the next Study Support Meeting or Online interactive sessions. You can also check your answers at the Self-Review Answers section which is located at the end of this Module.

- | |
|---|
| 1. How to make database connection to MySQL in PHP? |
| 2. Create an application and Use a MySQL database to perform CRUD (create, read, update, and delete) operations with PHP. |
| 3. What is database? |

Self-Review Answers (SRA) for Study Unit 14

1. How to make database connection to MySQL in PHP?

Syntax

Object oriented style:

`$mysqli -> new mysqli(host, username, password, dbname, port, socket)`

Example - Object Oriented style

Open a new connection to the MySQL server:

```
<?php
$mysqli = new mysqli("localhost", "my_user", "my_password", "my_db");

// Check connection
if ($mysqli -> connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
    exit();
}
?>
```

2. Create an application and Use a MySQL database to perform CRUD (create, read, update, and delete) operations with PHP.

https://www.w3schools.com/php/php_mysql_connect.asp

<https://www.tutorialrepublic.com/php-tutorial/php-mysql-crud-application.php>

<https://code.tutsplus.com/tutorials/how-to-work-with-mysql-in-php--cms-32222>

3. What is database?

A database is an organized collection of structured information, or data, typically stored electronically in a computer system.

Quires of MySQL database

- INSERT into student (name ,email) VALUES (xyz, 'xyz@email.com')
- SELECT * FROM `student` WHERE name = 'xyz'
- UPDATE student SET name = 'xyz' WHERE id = 1
- DELETE FROM student WHERE id = 2

References and Additional Reading Materials

<https://www.w3schools.com/sql/default.asp>

<https://www.techbeamers.com/sql-query-questions-answers-for-practice/>