

4.3 DATABASE CONNECTIVITY

PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.

Opening Database Connection:

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.

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

- **Server:** The host name running database server. If not specified then default value is localhost:3306. This is optional.
- **User:** The username accessing the database. If not specified then default is the name of the user that owns the server process. This is optional.
- **Password:** The password of the user accessing the database. If not specified then default is an empty password.
- **new_link:** If a second call is made to `mysql_connect()` with the same arguments, no new connection will be established; instead, the identifier of the already opened connection will be returned. This is optional.
- **client_flags:** This is optional. A combination of the following constants:
 1. `MYSQL_CLIENT_SSL` - Use SSL encryption
 2. `MYSQL_CLIENT_COMPRESS` - Use compression protocol
 3. `MYSQL_CLIENT_IGNORE_SPACE` - Allow space after function names
 4. `MYSQL_CLIENT_INTERACTIVE` - Allow interactive timeout seconds of inactivity before closing the connection

➤ Closing Database Connection:

PHP uses `mysql_close` to close a database connection. This function takes connection resource returned by `mysql_connect` function. It returns TRUE on success or FALSE on failure. If a resource is not specified then last opened database is closed.

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

➤ Creating a Database:

To create and delete a database the users should have admin privilege. PHP uses `mysql_query` function to create a MySQL database. This function takes two parameters and returns TRUE on success or FALSE on failure.

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

- **Sql:** SQL query to create a database
- **Connection:** if not specified then last opened connection by `mysql_connect` will be used.

➤ **Selecting a Database:**

Once the user establishes a connection with a database server then it is required to select a particular database where all the tables are associated. This is required because there may be multiple databases residing on a single server. PHP provides function `mysql_select_db` to select a database. It returns `TRUE` on success or `FALSE` on failure.

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

- **db_name:** Database name to be selected
- **connection:** if not specified then last opened connection by `mysql_connect` will be used.

➤ **Creating Database Tables:**

To create tables in the new database the user 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.

Creating and selecting a database table

```
<?php
$dbhost = 'localhost:3036';
$dbuser = 'root';
$dbpass = 'rootpassword';
$conn = mysql_connect($dbhost, $dbuser, $dbpass); //Creating a connection
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(14) NOT NULL, ' .
      'primary key ( emp_id ) )'; //Creating a table
mysql_select_db('test_db'); //Seeting a table
$retval = mysql_query( $sql, $conn );
if(! $retval )
```

```
{ die('Could not create table: ' . mysql_error()); }
echo "Table employee created successfully\n";
mysql_close($conn); //Closing a connection ?>
```

In case the user need to create many tables then it is 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 execute those commands.

➤ **Deleting a Database:**

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

➤ **Deleting a Table:**

It is 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 the users can delete some important information the user has in the table.

Deleting 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'; //Deleting a table
$retval = mysql_query( $sql, $conn );
if(! $retval )
{ die('Could not delete table employee: ' . mysql_error()); }
echo "Table deleted successfully\n";
mysql_close($conn); ?>
```

➤ **Insert Data into MySQL Database**

Data can be entered into MySQL tables by executing SQL INSERT statement through PHP function mysql_query. 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.

Inserting values

```
<?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); ?>
```

➤ Getting Data From MySQL Database

Data can be fetched from MySQL tables by executing SQL SELECT statement through PHP function `mysql_query`. The user 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.

Fetching data from tables

```
<?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> "; }
echo "Fetched data successfully\n";
mysql_close($conn); ?>

```

➤ Deleting Data from MySQL Database

Data can be deleted from MySQL tables by executing SQL DELETE statement through PHP function mysql_query. To delete a record in any table it is required to locate that record by using a conditional clause.

Deleting data from tables

```

<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 employee WHERE emp_id = $emp_id" ; //Query to delete a record
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>

```

➤ Updating Data into MySQL Database

Data can be updated into MySQL tables by executing SQL UPDATE statement through PHP function `mysql_query`. To update a record in any table it is required to locate that record by using a conditional clause.

Updating data

```

<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" 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">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>

```