Unit – 5 Working With Database in PHP

Introduction to MySQL Database with PHP

Overview of the database

- Database: A database is simply an organized collection of related data, typically stored on disk, and accessible by possibly many concurrent users.
- Databases are generally separated into application areas.
- For example, one database may contain Human Resource (employee and payroll) data; another may contain sales data; another may contain accounting data; and so on.
- Databases are managed by a DBMS.
- DBMS: Database Management System (DBMS) is a set of programs that manages any number of databases.

Introduction to MySQL

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use.
- MySQL uses standard SQL.
- MySQL compiles on a number of platforms.
- MySQL is free to download and use.
- MySQL is developed, distributed, and supported by Oracle Corporation

Data type of MySQL

- MySQL Data Types are divided into three main categories as given below :
  (1) Numeric DataType
  (2) String or Text DataType
  (3) Date Time Data Type

- **Numeric Datatype**

Sr. No.	Datatype	Size	Description
1	TINYINT	1	It is used to store integer values.
2	SMALLINT	2	It is used to store integer values.
3	MEDIUMINT	3	It is used to store integer values.

4	INT	4	It is used to store integer values.
5	BIGINT	8	It is used to store integer values.
6	FLOAT	4	It is used to store single precision floating point value.
7	DOUBLE	8	It is used to store double precision floating point value
8	DECIMAL		It is used to store integer values.

- **String Datatype**

Sr. No.	Datatype	Size	Description
1	CHAR	255	It is used to store fixed length string.
2	VARCHAR	255	It is used to store variable length string.
3	TINYBLOB	255	It is used to store short binary data.
4	TINYTEXT	255	It is used to store short text string.
5	BLOB	65535	It is used to store large binary data.
6	TEXT	65535	It is used to store large text string.
7	MEDIUMBLOB	16777215	It is used to store medium binary data.
8	MEDIUMTEXT	16777215	It is used to store medium text string.
9	LONGBLOB	4294967295	It is used to store extremely large binary data.
10	LONGTEXT	4294967295	It is used to store extremely large text string

- **Date and Time Datatype**

Sr. No.	Datatype	Size	Description
1	DATE	3	It is used to store date values in the YYYY – MM – DD Format.

2	TIME	3	It is used to store time values or time duration in the HH : MM : SS format.
3	YEAR	1	It is used to year values in the YYYY format.
4	DATETIME	8	It is used to store Date and Time value in the YYYY – MM – DD HH : MM : SS format.
5	TIMESTAMP	9	It is used to store combined date and time value in the YYYYMMDDHHMMSS format.

Field Modifiers in PHP

- **NULL or NOT NULL** :
  - It allows you to specify weather the field can accept null value or you must have to enter value for that field.
  - You can specify this modifier at the end field definition.
  - By default all the filed of the table having this modifier set to NULL.
  - **Example**
    create table Subject (SubjectCode varchar (5) NOT NULL, SubjectName varchar (10) NOT NULL, SubDesc varchar (20) NULL)

- **DEFAULT**
  - Default modifier allows you to specify default value for the field so if you don't enter any value for that field then default value is used.
  - **Example** :
    create table Subject (SubjectCode varchar (5) NOT NULL, SubjectName varchar (10) NOT NULL, SubDesc varchar (20) DEFAULT 'Computer')

- **AUTO_INCREMENT**
  - This modifier automatically increments the value of field by one.
  - There is no need to insert explicit value for this field.
  - It is widely used in primary key field where unique id is generated automatically by MySQL each time by incrementing previous by 1.
  - However you can specify AUTO_INCREMENT modifier for only INT data type.
  - **Example :**
    create table Subject (SubjectCode INT(5) AUTO_INCREMENT, SubjectName varchar (10) NOT NULL, SubDesc varchar (20) NULL,)

Types of MySQL tables & storage engines

- When we create table using Create Table statement it is also possible to specify type of the table using TYPE attribute as shown below :
  Create table TableName (ColumnNameDataType, ColumnNameDataType) TYPE = TableType
- Following are the various Types of MySQL tables that you can specify at the time of creating new Database Table.

1. MyISAM
2. ISAM
3. HEAP
4. InnoDB
5. BDB

- **MyISAM**
  - While creating Table if you don't specify the type of the Database Table then by default its type is set to MYISAM.
  - This table is portable. Portable means the table created using this type in one OS can also be used in other OS.
  - It supports large table file more than 4GB.
  - It is well suited for faster access then speed.
  - It allows you to reduce the space using compression.
  - It allows you to specify index on BLOB as well as TEXT type.

- **ISAM**
  - This Table Type is similar to the MyISAM table type in the way it also supports fixed size as well as dynamic size table.
  - But it is different from MYISAM table type in following way :
    - It can support table files up to 4GB but not grater then 4GB.
    - It is not portable. Means table created using this type in one OS cannot use in other OS.
    - It does not allow faster accessing and compression.
    - Since maximum key length in this table type is 256 it does not allows you to specify index on BLOB and TEXT type.

- **HEAP**
  - This table type is widely used for temporary tables because it supports incredible speed.
  - It is in-memory table which uses hashed indices.
  - It does not allow BLOB and TEXT type as MYISAM and ISAM type.

- **BDB**
  - BDB is also known as Berkeley DB.

- This table type is transaction safe and widely used for transaction.
- It supports very large applications with more than one users trying to insert and update the same data at the same time.
- It allows you the facility of transaction using commit and rollback statement.
- One important facility provided by this table type is that allows you the facility of recover data from crashes.
- It is not portable because the path of the table is hardwired into the table file while creating the table.

- **InnoDB**
  - This table type is also transaction safe.
  - It supports very large applications with more then one users trying to insert and update the same data at the same with locking mechanism.
  - Locking mechanism prevents users from modifying records while another user is modifying the same record at the same time.
  - It also allows you the facility of recovering data from crashes.
  - This table type is portable.

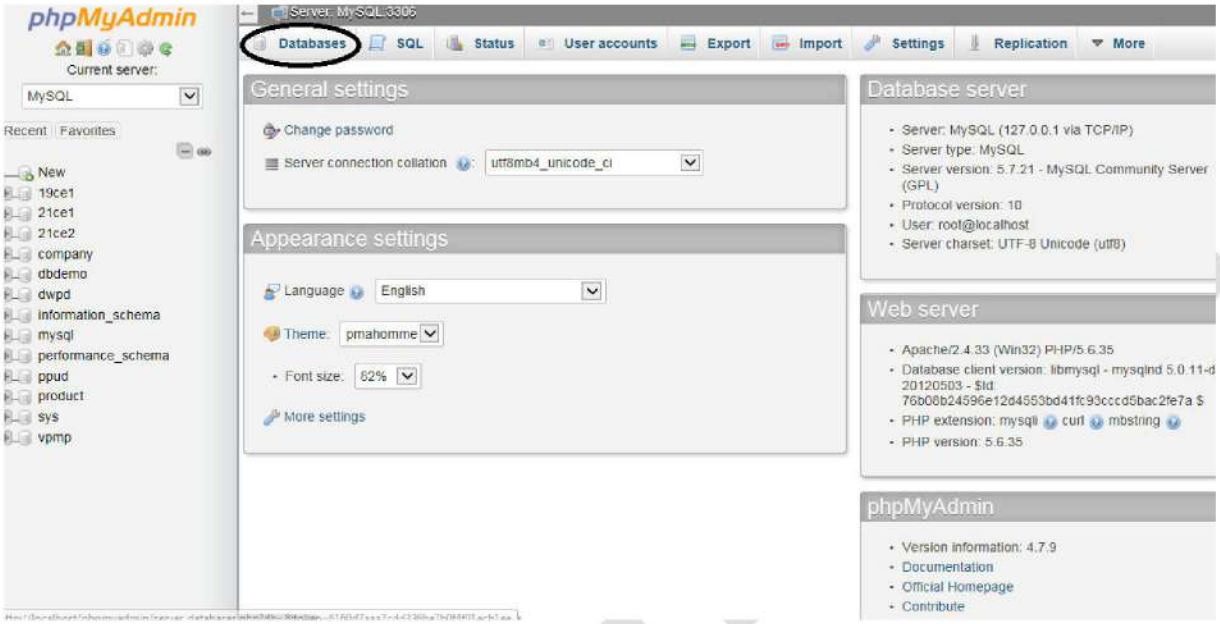
- **Merge**
  - This table type is useful for virtual table.
  - This table type can be created by joining more than one MyISAM table types into single table.
  - However you can join more than one MyISAM table if and only if all the table having same structure.

Creating a database using phpmyadmin & Console

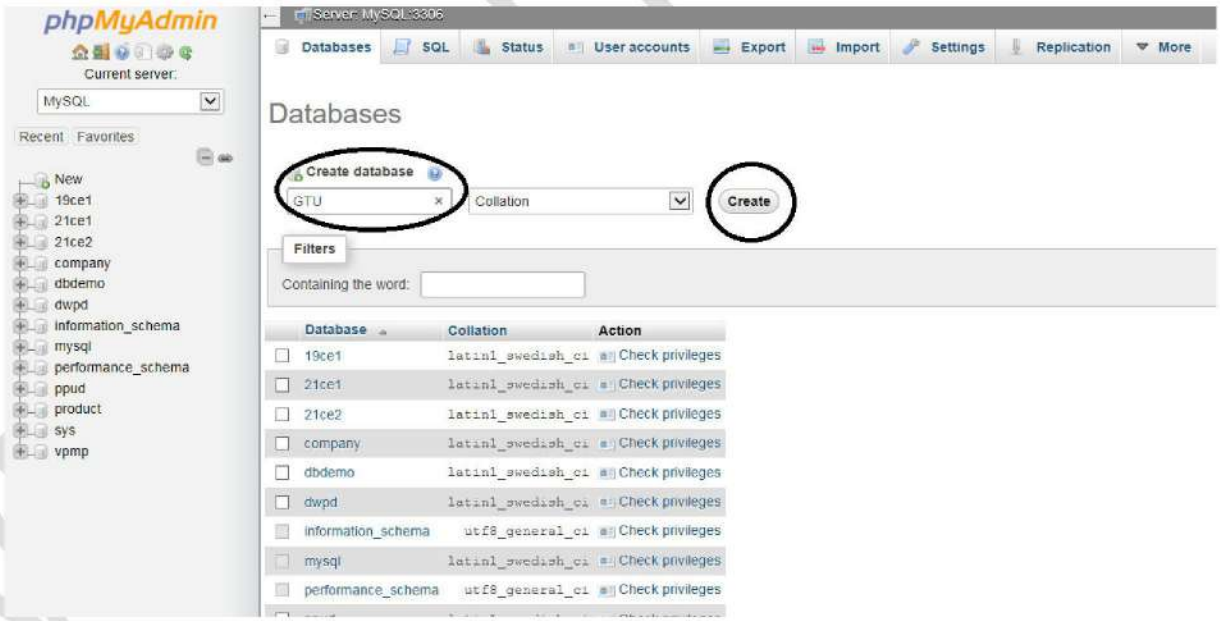
- **Creating a database using phpmyadmin**
  - Run wampserver and Click on wampserver icon

  - Provide "root" username and click on Go button

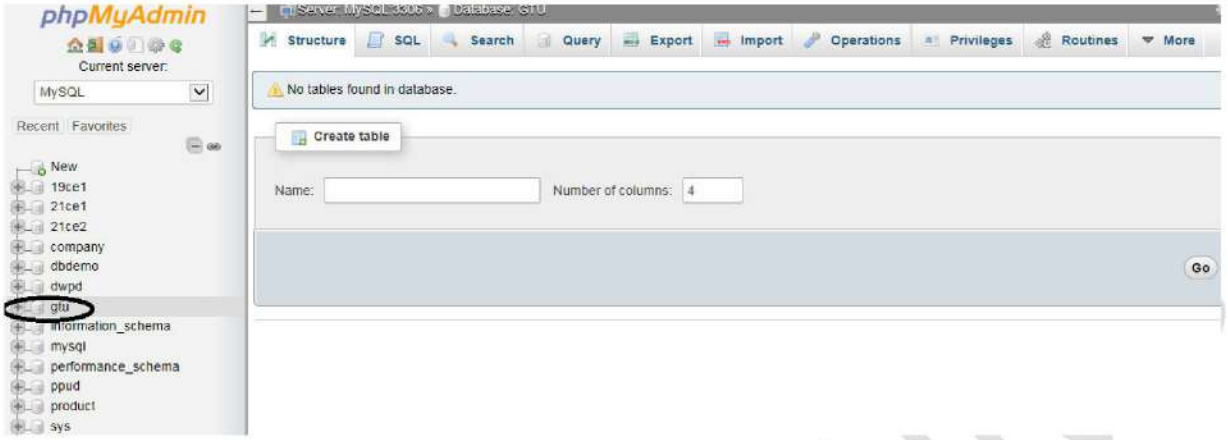
- On phpMyAdmin dashboard, click on **Databases** option



- Now provide database name "GTU" and click on **Create** button



- Now "GTU" database is created.

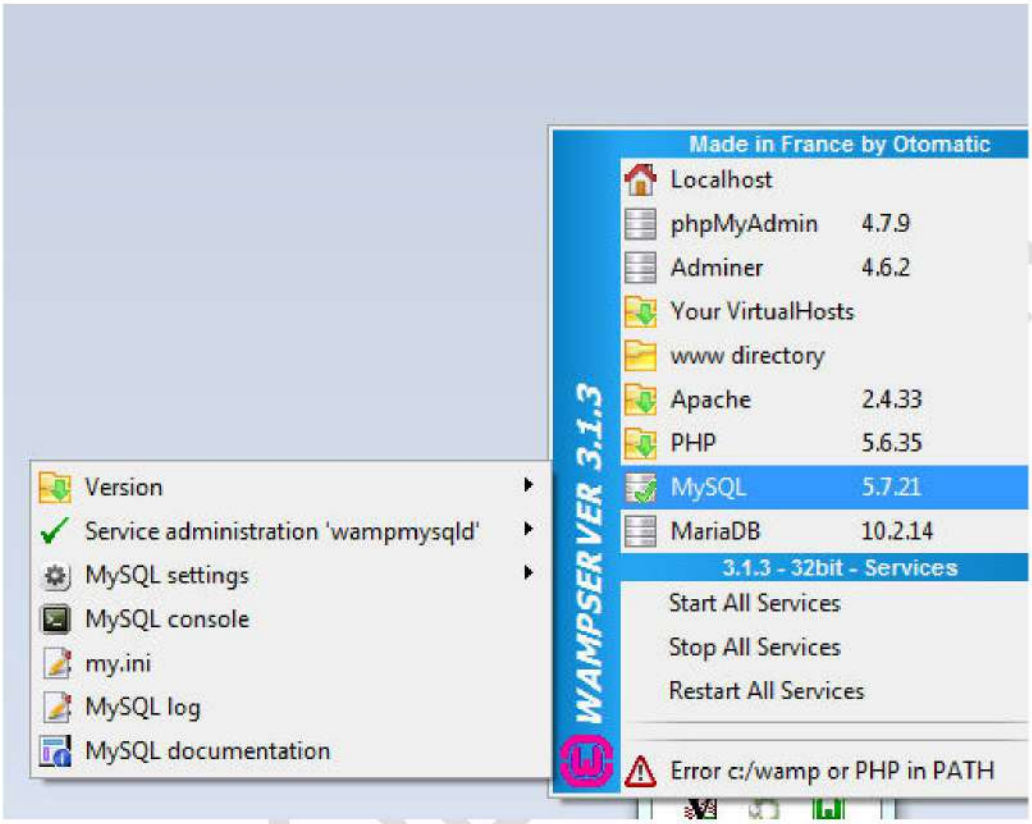


- **Creating a database using console**

  - Run wampserver and Click on wampserver icon

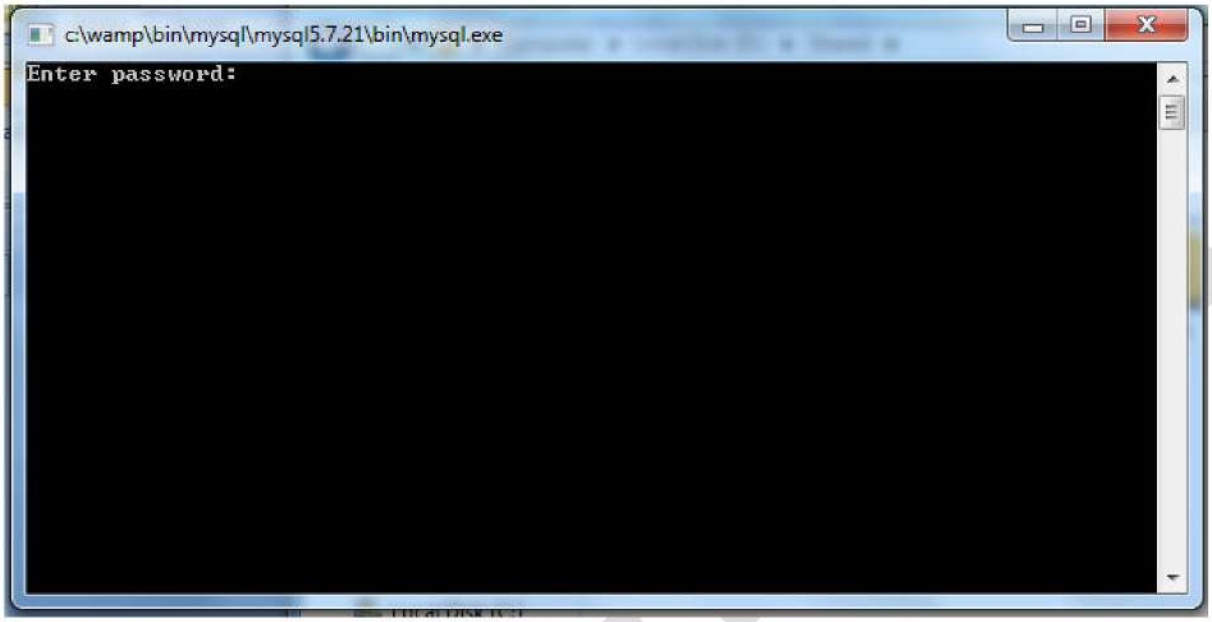
  - Now Click on MySQL option

- Click on MySQL console option

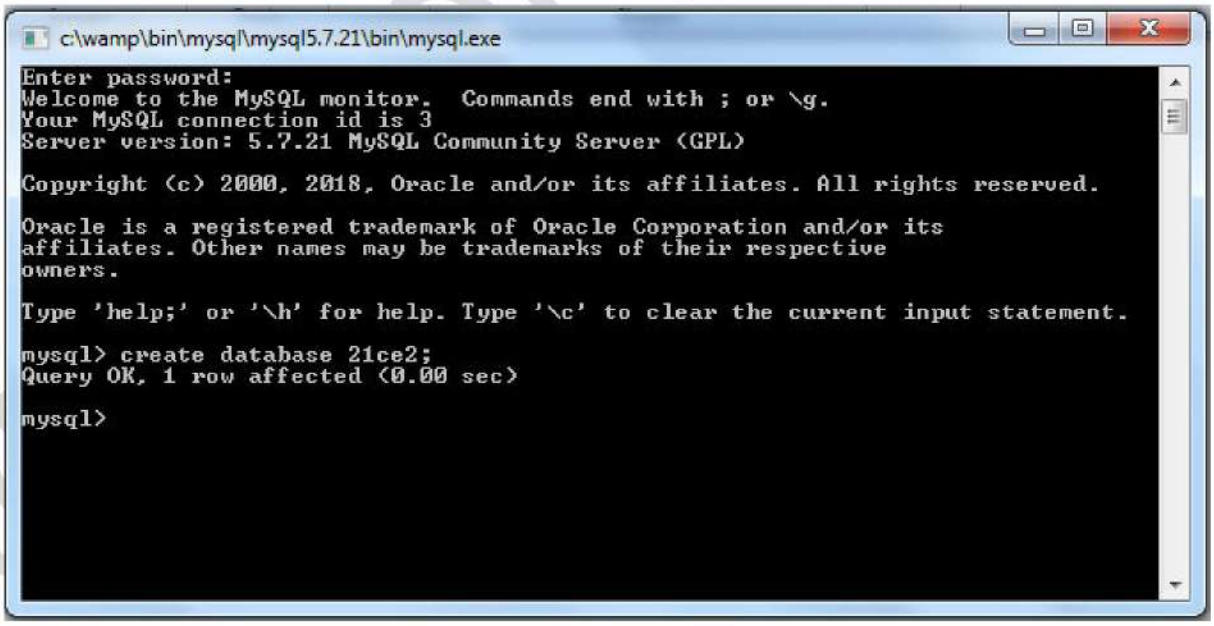


- Provide username “root” and click on OK button

- As "root" username have no password press enter key.



- Now type a sql statement to create an database **"21ce2"**.
  "**create database 21ce2**";



- **Creating a database using PHP script**

```
<?php
        $con=mysqli_connect("localhost", "root");
        $qry="create database vpmp";
        $ans = mysqli_query($con,$qry);
        if($ans)
        {
                echo "Database Created Successfully";
        }
        else
        {
                echo "Database Not Created";
        }
        mysqli_close($con);
?>
```

Connecting with MYSQL Database: mysqli_connect( ) and mysqli_select_db( )

- **mysqli_connect( )**
  - This function allows you to establish connection of PHP application with MySQL server.
  - **Syntax :**
    $VariableName = mysqli_connect (serverName, UserName, Password)
    ServerName: Indicates the name of the MySQL server with which you want to establish connection.
    UserName : Indicates name of the user using which you can logs on to MySQL server.
    Password : Indicates password of the user using which you can logs on to MySQL Server.
  - This Function returns a Boolean value TRUE or FALSE.
  - If connection establish successfully with MySQL Server then this function returns true value otherwise it returns false.
  - **Example :**

```
<?php
$con=mysqli_connect("localhost", "root");
if($con)
{
        echo "Server Connected";
}
else
{
        echo "Server Not Connected";
}
?>
```

- **mysqli_select_db( )**
  - This function is used to select a database.
  - **Syntax:**

    mysqli_select_db(connectionname, "databasename");

    DatabaseName : Indicates the name of the database that you want to select.

    ConnectionName : Indicates the name of the variable that is used at the time of established connection with MySQL server using mysqli_connect ( ) function.
  - This function returns a Boolean value TRUE or FALSE.
  - If Function successfully executed then it returns TRUE otherwise it return false.

  - **Example:**

```
<?php
$con=mysqli_connect("localhost", "root");
$db=mysqli_select_db($con, "vpmp");
if($db)
{
        echo "database selected";
}
else
{
        echo "database not selected";
}
?>
```

Executing MySQL Queries :mysqli_query( )

- **mysqli_query( )**
  - This function allows you to specify and execute the MySQL command on MySQL Server.
  - **Syntax:**

    mysqli_query(connectionname, "Query");

    Query : Indicates the MySQL command to be executed.
  - **ConnectionName** : Indicates the name of the variable that is used at the time of establish connection with MySQL server using mysqli_connect ( ) function.
  - **Example:**

    Write a php script to create a database "vpmp".

```
<?php
        $con=mysqli_connect("localhost", "root");
        $qry="create database vpmp";
```

```
        $ans=mysqli_query($con,$qry);
        if($ans)
        {
                echo "Database created successfully";
        }
        else
        {
                echo "Database not created";
        }
?>
```

Performing database operations

- **mysqli_fetch_rows( )**
  - This function allows you to retrieve a record from the record set that is returned from executing the MySQL query.
  - The record that is returned by this function is in the form of numeric array. Numeric array contains index and value associated with that index.
  - If there is no record in the record set then it returns false value.
  - **Syntax :**

    mysqli_fetch_rows ($VariableName) ;

    Variable Name: Indicates the record set that is returned from executing the MySQL command using mysqli_query ( ) function.
  - **Example:**

    Write a php script to search a record from database.

```
<?php
        $con=mysqli_connect("localhost", "root");
        $db=mysqli_select_db($con, "vpmp");
        $qry="select * from student";
        $ans=mysqli_query($con,$qry);
        while($row=mysqli_fetch_rows($ans))
        {
                echo $row[0] ." ". $row[1];
                echo "<br>";
        }
        mysqli_close($con);
?>
```

- **mysqli_fetch_array( )**
  - This function allows you to retrieve a record from the record set that is returned from executing the MySQL query.
  - The record that is returned by this function is in the form of numeric array, associative array or both.
  - If there is no record in the record set then it returns false value.
  - **Syntax**:

    mysqli_fetch_array ($VariableName, ArrayType) ;

    Here,

    Variable Name : Indicates the record set that is returned from executing the MySQL command using mysqli_query ( ) function.
  - Array Type indicates the type of the array to be returned.
  - It can have one of the following value :

    (1) **MYSQL_ASSOC** : This type of array contains name of the field and the value associated with that field for current record.

    (2) **MYSQL_NUM** : This type array contains index of the field and the value associated with that index for current record.

    (3) **MYSQL_BOTH** : It is combination of both Associative array and Numeric array. It is the default type to be returned by this function.

  - **Example :**

```
<?php
        $con = mysqli_connect ("localhost", "root") ;
        mysqli_select_db ($con,"mydatabase") ;
        $sql = "select * from Package_Master" ;
        $result = mysqli_query ($con,$sql);
        $Ans = mysqli_fetch_array($result,  3) ;
        print_r ($Ans) ;
        mysqli_close ($con)
    ?>
```

- **mysqli_num_rows( )**
  - This function allows you to retrieve number of record available in the record set.
  - **Syntax** :
  - $Number = mysqli_num_rows ($QueryResult) ;

    Here,

    QueryResult is the variable that holds the result returned by mysqli_query ( ) function.

- It returns numeric value which contains number of records available in the record set.

- **Example** :

```
<?php
    $con = mysqli_connect ("localhost", "root") ;
    mysqli_select_db ($con, "vpmp") ;
    $sql = "select * from student" ;
    $result = mysqli_query ($con,$sql);
    $record = mysqli_num_rows($result) ;
    echo $record ."Records" ;
    mysqli_close ($con) ;
   ?>
```

- **mysqli_error( )**
  - This function allows you to retrieve the description of error that is encountered while executing the script.
  - If the script contains more than one errors then it will returns error description of the last statement in which error is encountered.
  - If no error encountered while executing the script then it will returns blank string.
  - Syntax :

```
mysqli_error ( ) ;
```

- **mysqli_close( )**
  - This function allows you to close the connection that is established using mysqli_connect() function.
  - **Syntax :**

```
mysqli_close (ConnectionName)
```

Here,
ConnectionName : Indicates the name of the variable that is used at the time of establish connection with MySQL server using mysqli_connect ( ) function.
It returns a Boolean value TRUE or FALSE. If connection closed successfully then it returns TRUE otherwise it returns FALSE.

- **Create a table**

```
<?php
    $con=mysqli_connect("localhost", "root");
    $db=mysqli_select_db($con, "vpmp");
    $qry="create table student (enrol int, snamevarchar(20), sem int)";
    $ans = mysqli_query($con,$qry);
```

```
        if($ans)
        {
                echo "Table Created Successfully";
        }
        else
        {
                echo "Table Not Created";
        }
        mysqli_close($con);
?>
```

- **Delete a table**

```
<?php
        $con=mysqli_connect("localhost", "root");
        $db=mysqli_select_db($con, "vpmp");
        $qry="delete table student";
        $ans = mysqli_query($con,$qry);
        if($ans)
        {
                echo "Table Deleted Successfully";
        }
        else
        {
                echo "Table Not Deleted";
        }
        mysqli_close($con);
?>
```

- **Insert data into a table**

```
<?php
        $con=mysqli_connect("localhost", "root");
        $db=mysqli_select_db($con, "vpmp");
        $qry="insert into student values(501, 'Hitesh', 5)";
        $ans = mysqli_query($con,$qry);
        if($ans)
        {
                echo "Record Inserted Successfully";
        }
        else
        {
```

```
        echo "Record Not inserted";
    }
    mysqli_close($con);
?>
```

- **Update data into the table**

```
<?php
    $con=mysqli_connect("localhost", "root");
    $db=mysqli_select_db($con, "vpmp");
    $qry="update student set sem=6 where enrol=501";
    $ans = mysqli_query($con,$qry);
    if($ans)
    {
        echo "Record Updated Successfully";
    }
    else
    {
        echo "Record Not updated";
    }
    mysqli_close($con);
?>
```

- **Retrieve data from the table**

```
<?php
    $con=mysqli_connect("localhost", "root");
    $db=mysqli_select_db($con, "vpmp");
    $qry="select * from student";
    $ans=mysqli_query($con,$qry);
    while($row=mysqli_fetch_rows($ans))
    {
        echo $row[0] ." ". $row[1];
        echo "<br>";
    }
    mysqli_close($con);
?>
```

- **Delete data from the table**

```
<?php
        $con=mysqli_connect("localhost", "root");
        $db=mysqli_select_db($con, "vpmp");
        $qry="delete from student where enrol=501";
        $ans = mysqli_query($con,$qry);
        if($ans)
        {
                echo "Record Deleted Successfully";
        }
        else
        {
                echo "Record Not Deleted";
        }
        mysqli_close($con);
?>
```

Displaying data from the database in different formats, including tables

- **Display data from the database in tabular format**

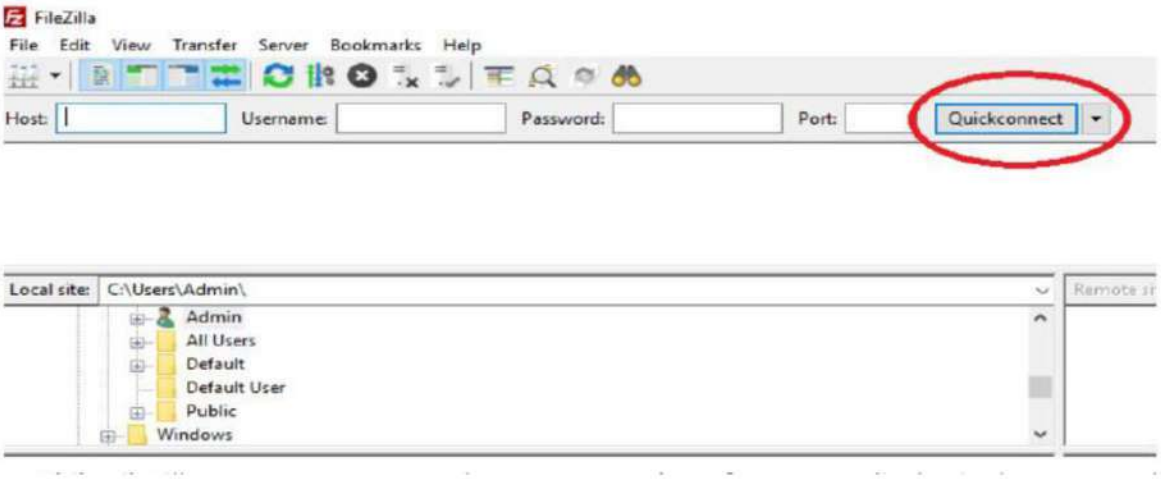
```
<?php
        $con=mysqli_connect("localhost", "root");
        $db=mysqli_select_db($con, "vpmp");
        $qry="select * from student";
        $ans = mysqli_query($con,$qry);
        $no=mysqli_num_rows($ans);
        if($no > 0)
        {
                echo "<table border=1>";
                echo "<tr><td> Enrollment No </td>";
                echo "<td> Student Name </td>";
                echo "<td> Semester </td></tr>";
                while ($res=mysqli_fetch_row($ans))
                {
                        echo "<tr><td>$res[0]</td>";
                        echo "<td>$res[1]</td>";
                        echo "<td>$res[2]</td>";
                        echo "<td>$res[3]</td></tr>";
```

```
                }
        }
        else
        {
                echo "No record Found";
        }
        mysqli_close($con);
?>
```

Hosting a Website

- FTP is stands for File Transfer Protocol.
- It is a system that allow you to log in to a server and upload, download or modify content.
- FileZilla is powerful and free software for transferring file over the internet.
- Once you have the FileZilla client downloaded and activate on your system, enter the domain name in the address field.
- The Username and password you need to type in are the same as the ones you use to log in to your cPanel.

- **Connect to a web server**
    - In the FileZilla QuickConnect bar at the top of the page, enter your website URL in the host field.
    - Enter your FTP username and password.
    - Click QuickConnect.



- While FileZilla connect to your web server, a number of messages display in the message log.
- Once you are successfully connected to web server, the folder structure of your webserver display in the Remote Site pane in FileZilla.

- **Transfer file to website**
  - In the local site pane, navigate to the location of your website.
  - Select the files you want to upload to the website. You can select multiple files.
  - Drag the files you want to transfer from the Local site pane to Remote Site Pane.
  - FileZilla automatically transfer your files to your website.
  - When you are finished uploading your files, close FileZilla and check your website in browser.