

WEB ENGINEERING

By

Adnan Amin

Institute of Management Sciences, Peshawar

OVERVIEW

- Connection to the database (MySQL from PHP)
- Separate statement for connecting to server and database.
- Connecting with different databases.
- Sending a query to the database using PHP.

CONNECTION TO THE DATABASE (1/3)

- The first step in a database interaction is connecting to the database. You use a PHP function to connect to the database. To make the connection, you need to supply the function with four things:
- **1. location:** Specify the server name where the database is installed. Such as IP address, localhost or domainname .com
- **2. Account name:** Specify the correct user name, if you are using web hosting company, you will be given a valid user name and password. Such as “root”.
- **3. Password:** You have to enter a valid password to access the database. Such as by default root user password is blank.
- **4. Database name:** On your MySql server, there may be created many databases, so you need to tell it which database you want to use.

CONNECTION TO THE DATABASE (2/3)

- Connects to the MySQL server.

```
mysql_connect ( "hostname", "user", "pass" )  
    or die ( " Could not connect to localhost ");
```

Example # 1

```
$host = "localhost";  
$user = "root";  
$pass = " ";  
$db    = "University";  
$connect= mysql_connect ( $host, $user, $pass, $db) or  
    die ( "Could not connect to the server ");
```

Example # 2

```
$connect= mysql_connect ("localhost","root","", $db)  
    or die ( " Could not connect to the server ");
```

CONNECTION TO THE DATABASE (3/3)

- You'll need database connection in many places of website. therefore, Create an include file called info.php with the following statements.

```
$host = "localhost";  
$user = "root";  
$pass = " ";  
$db = "University";  
$connect = mysql_connect( $host, $user, $pass, $db )  
or die ("Could not connect to the server or database  
");
```

- Whenever, you need the database connection just type single line statement. Such as

<?php include ("info.php"); ?>

SEPARATE STATEMENT FOR CONNECTING TO SERVER AND DATABASE (1/2)

- `mysql_select_db (“ database name “) or
die (“ Could not connect the database”
);`
- The above statement is equivalent to the MySQL command **USE**; makes the selected database the active one.

SEPARATE STATEMENT FOR CONNECTING TO SERVER AND DATABASE (2/2) EXAMPLE

- <?php
- \$host="localhost";
- \$user="root";
- \$pass=" ";
- \$db="mydb";
- \$connect= mysql_connect(\$host,\$user,\$pass) or
die ("Could not connect to the server");
- echo "connect to server";
- \$usedb=mysql_select_db(\$db) or
 die("Could not connect to database");
- echo "
Use database db";
- ?>

CONNECTING WITH DIFFERENT DATABASES

Connect to the MySQL database.

```
$connect=mysql_connect ($host, $user, $pass);  
$db=mysql_select_db( $db, $connect);
```

Connect to the postgresQL database.

```
$connect=pg_connect ("host-$host user-$user  
password- $pass dbname- $db ");
```

Connect to the sybase database.

```
$connect=sybase_connect ($host, $user, $pass);  
$db=sybaseselect_db( $db, $connect);
```

Connect to the Oracle database.

```
$connect= OCILogon ( $user, $pass);
```


SENDING A QUERY TO THE DATABASE USING PHP (1/5)

- After PHP has established a connection to the database, you can perform whatever action you desire, such as get data, change data, or insert new data.
- The SQL query tells the database what action you want to perform.
- See next page...

```
$sql = "SELECT * FROM students" ;  
$result = mysql_query ( $sql , $connect );
```

- The first statement stores the SQL query in a variable `$sql`. The SQL statement gets all the data in the table `students`.
- The `mysql_query` function sends the query in `$sql` to the database over the database connection established earlier and stored in `$connect`. The data temporary stored in rows and columns into `$result`.

```
$sql = "SELECT * FROM students" ;  
$result = mysql_query ( $sql , $connect );
```

- The data is stored in the temporary table in rows and columns. You can use PHP functions to retrieve one row from the table and store it in an array.

```
$sql = "SELECT * FROM students" ;  
$result = mysql_query ( $sql , $connect );  
  
$row= mysql_fetch_array( $result) ;  
echo "$row[0] $row[1]";
```

- The mysql_fetch_array returns one row of data from the temporary table specified by \$result while \$row is an array containing all the fields in the temporary table.

- `$row = mysql_fetch_array ($result);`
- `echo $row[0];`
- 0 means display the data of first field of the table. You can also use to mention the field name such as
- `echo $row['regno'];`
- You can use loop to display many records.

```
while($row = mysql_fetch_array($result))  
{  
    echo "Regno: $row[0]";  
    echo "Name: $row[1]";  
}
```

- You can display the MySQL error message by using the `mysql_error` function.

```
$sql = "SELECT * FROM student";  
$result = mysql_query($sql) or die("Query failed: ".mysql_error());  
$row = mysql_fetch_array($result);
```

- Shows the error message that has been returned directly from the MySQL server.

Query failed: Table 'student' doesn't exist With these statements, when the query fails to execute,

```
<?php include('info.php'); ?>
<?php
    $sql="select * from table2";
    $result=mysql_query($sql);

    while($row=mysql_fetch_array($result))
    {
        echo "<br>". $row['regno'];
        echo $row['name'];
    }
?>
```

Syntax:

```
INSERT INTO table_name(field1,field2,.....)  
VALUES(value1,value2.....);
```

```
<?php include('info.php'); ?>
```

```
<?php
```

```
$insert="insert into student(Regno,Name)  
values(10,'Atif Amin')";
```

```
$results= mysql_query($insert) or  
die(mysql_error());
```

```
echo "Data inserted successfully";
```

```
?>
```

ERASES THE DATA FROM YOUR TABLE

Syntax:

**UPDATE table_name SET field1=value1,
field2=value2..... WHERE field=value;**

```
<?php include('info.php'); ?>
```

```
<?php
```

```
    $update="UPDATE students SET  
    name='Bilal'
```

```
        WHERE Regno=10";
```

```
$results= mysql_query($update) or  
die(mysql_error());
```

```
echo "Data Updated successfully";
```

```
?>
```


THANK YOU!

Referances

- PHP5,MySql,Apache By Wrox