

## PHP Getting Data from 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.

### Example

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.

## Example

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.

## Example

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.

## Releasing Memory

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.