# 7.11 PHP Insert Multiple Records Into MySQL

## Insert Multiple Records Into MySQL Using MySQLi and PDO

Multiple SQL statements must be executed with the mysqli\_multi\_query() function.

The following examples add three new records to the "MyGuests" table:

### Example (MySQLi Object-oriented)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

**// Create connection**
$conn = new mysqli($servername, $username, $password, $dbname)

;
**// Check connection**if ($conn->connect\_error) {
    die("Connection failed: " . $conn->connect\_error); }
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";
if ($conn->multi\_query($sql) === TRUE) {
    echo "New records created successfully"; } else {
    echo "Error: " . $sql . "<br>" . $conn->error; }
$conn->close();
?>

Note that each SQL statement must be separated by a semicolon.

###  Example (MySQLi Procedural)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

**// Create connection**$conn = mysqli\_connect($servername, $username, $password, $dbname);

**// Check connection**if (!$conn) {
    die("Connection failed: " . mysqli\_connect\_error());
}
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";
if (mysqli\_multi\_query($conn, $sql)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli\_error($conn); }
mysqli\_close($conn);
?>

The PDO way is a little bit different:

### Example (PDO)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);

    **// set the PDO error mode to exception**
    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    **// begin the transaction**
    $conn->beginTransaction();

**// our SQL statements**    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')");
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Mary', 'Moe', 'mary@example.com')");
    $conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('Julie', 'Dooley', 'julie@example.com')");

**// commit the transaction**    $conn->commit();
    echo "New records created successfully";
    }
catch(PDOException $e)
    {
    **// roll back the transaction if something failed**
    $conn->rollback();
    echo "Error: " . $e->getMessage();
    }
$conn = null;
?>

# 7.12 PHP Select Data From MySQL

## Select Data From a MySQL Database

The SELECT statement is used to select data from one or more tables:

SELECT column\_name(s) FROM table\_name

or we can use the \* character to select ALL columns from a table:

SELECT \* FROM table\_name

## Select Data With MySQLi

The following example selects the id, firstname and lastname columns from the MyGuests table and displays it on the page:

### Example (MySQLi Object-oriented)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

**// Create connection**
$conn = new mysqli($servername, $username, $password, $dbname);

**// Check connection**if ($conn->connect\_error) {
    die("Connection failed: " . $conn->connect\_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num\_rows > 0) {

    **// output data of each row**    while($row = $result->fetch\_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

d: 1 - Name: John Doe
id: 2 - Name: Mary Moe
id: 3 - Name: Julie Dooley

Code lines to explain from the example above:

First, we set up an SQL query that selects the id, firstname and lastname columns from the MyGuests table. The next line of code runs the query and puts the resulting data into a variable called $result.

Then, the function num\_rows() checks if there are more than zero rows returned.

If there are more than zero rows returned, the function fetch\_assoc() puts all the results into an associative array that we can loop through. The while() loop loops through the result set and outputs the data from the id, firstname and lastname columns.

The following example shows the same as the example above, in the MySQLi procedural way:

### Example (MySQLi Procedural)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

**// Create connection**$conn = mysqli\_connect($servername, $username, $password, $dbname);

**// Check connection**if (!$conn) {
    die("Connection failed: " . mysqli\_connect\_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli\_query($conn, $sql);
if (mysqli\_num\_rows($result) > 0) {

    **// output data of each row**    while($row = mysqli\_fetch\_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
mysqli\_close($conn);
?>

id: 1 - Name: John Doe
id: 2 - Name: Mary Moe
id: 3 - Name: Julie Dooley

You can also put the result in an HTML table:

### Example (MySQLi Object-oriented)

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

**// Create connection**$conn = new mysqli($servername, $username, $password, $dbname);

**// Check connection**if ($conn->connect\_error) {
    die("Connection failed: " . $conn->connect\_error);
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num\_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";

    **// output data of each row**    while($row = $result->fetch\_assoc()) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["firstname"]." ".$row["lastname"]."</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}
$conn->close();
?>

|  |
| --- |
|  |
| 1 | **John Doe** |
| 2 | **Mary Moe** |
| 3 | **Julie Dooley** |

## Select Data With PDO (+ Prepared Statements)

The following example uses prepared statements.

It selects the id, firstname and lastname columns from the MyGuests table and displays it in an HTML table:

### Example (PDO)

<?php
echo "<table style='border: solid 1px black;'>";
echo "<tr><th>Id</th><th>Firstname</th><th>Lastname</th></tr>";
class TableRows extends RecursiveIteratorIterator {
    function \_\_construct($it) {
        parent::\_\_construct($it, self::LEAVES\_ONLY);
    }
    function current() {
        return "<td style='width:150px;border:1px solid black;'>" . parent::current(). "</td>";
    }
    function beginChildren() {
        echo "<tr>";
    }
    function endChildren() {
        echo "</tr>" . "\n";    }
}
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
    $stmt->execute();

**// set the resulting array to associative**
    $result = $stmt->setFetchMode(PDO::FETCH\_ASSOC);
    foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {
        echo $v;
    }
}
catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
echo "</table>";
?>

|  |
| --- |
|  |
| 1 | John | Doe |
| 2 | Mary | Moe |
| 3 | Julie | Doole |