PDO - PHP Data Objects

Connection:

Try

{

$conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  
    echo "Connected successfully";   
    }

catch(PDOException $e)  
    {  
     echo "Connection failed: " . $e->getMessage();  
    }

Close:

$conn = null;

Database Creation:

try {  
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

|  |
| --- |
| $sql = "CREATE DATABASE myDBPDO";     // use exec() because no results are returned     $conn->exec($sql); |

echo "Database created successfully<br>";  
    }  
catch(PDOException $e)  
    {  
    echo $sql . "<br>" . $e->getMessage();  
    }  
  
$conn = null;

|  |
| --- |
| Create Table:  $sql = "CREATE TABLE MyGuests (     id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,      firstname VARCHAR(30) NOT NULL,     lastname VARCHAR(30) NOT NULL,     email VARCHAR(50),     reg\_date TIMESTAMP     )";      // use exec() because no results are returned     $conn->exec($sql); |

|  |
| --- |
| Insert Data:  $sql = "INSERT INTO MyGuests (firstname, lastname, email)     VALUES ('John', 'Doe', 'john@example.com')";     // use exec() because no results are returned     $conn->exec($sql);     echo "New record created successfully"; |

|  |
| --- |
| Get Last ID:  $sql = "INSERT INTO MyGuests (firstname, lastname, email)     VALUES ('John', 'Doe', 'john@example.com')";     // use exec() because no results are returned     $conn->exec($sql);     $last\_id = $conn->lastInsertId(); |

|  |
| --- |
| Multiple insertion:  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; |

|  |
| --- |
| Prepared Statement:  $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)      VALUES (:firstname, :lastname, :email)");     $stmt->bindParam(':firstname', $firstname);     $stmt->bindParam(':lastname', $lastname);     $stmt->bindParam(':email', $email);      // insert a row     $firstname = "John";     $lastname = "Doe";     $email = "john@example.com";     $stmt->execute();      // insert another row     $firstname = "Mary";     $lastname = "Moe";     $email = "mary@example.com";     $stmt->execute();      // insert another row     $firstname = "Julie";     $lastname = "Dooley";     $email = "julie@example.com";     $stmt->execute(); |

|  |
| --- |
| SELCET  $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");      $stmt->execute();      // set the resulting array to associative     $result = $stmt->setFetchMode(PDO::FETCH\_ASSOC);      foreach(new TableRows($stmt->fetchAll() as $k=>$v) {          echo $v;     } |
| Delete  $sql = "DELETE FROM MyGuests WHERE id=3";      // use exec() because no results are returned     $conn->exec($sql); |

|  |
| --- |
| UPDATE:  $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";      // Prepare statement     $stmt = $conn->prepare($sql);      // execute the query     $stmt->execute();      // echo a message to say the UPDATE succeeded     echo $stmt->rowCount() . " records UPDATED successfully"; |