

CREA DATABASE

Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

CREA TABELLA

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 to 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
)";

if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

CREA CONNESSIONE

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());
}
```

INSERIMENTO DATI

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')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

INSERIMENTO DATI

(LAST INSERTED)

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)";

if (mysqli_query($conn, $sql)) {
    $last_id = mysqli_insert_id($conn);
    echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

INSERIMENTO DATI MULTIPLO

```
<?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);
?>
```

SELECT DATI

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);
?>
```

DELETE DATI

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 to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```


UPDATE DATI

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 = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

LIMIT DATI CON MYSQL

Limit Data Selections From a MySQL Database

MySQL provides a LIMIT clause that is used to specify the number of records to return.

The LIMIT clause makes it easy to code multi page results or pagination with SQL, and is very useful on large tables. Returning a large number of records can impact on performance.

Assume we wish to select all records from 1 - 30 (inclusive) from a table called "Orders". The SQL query would then look like this:

```
$sql = "SELECT * FROM Orders LIMIT 30";
```

When the SQL query above is run, it will return the first 30 records.

What if we want to select records 16 - 25 (inclusive)?

MySQL also provides a way to handle this: by using OFFSET.

The SQL query below says "return only 10 records, start on record 16 (OFFSET 15)":

```
$sql = "SELECT * FROM Orders LIMIT 10 OFFSET 15";
```

You could also use a shorter syntax to achieve the same result:

```
$sql = "SELECT * FROM Orders LIMIT 15, 10";
```

PHP mysqli_num_rows() Function

 [PHP MySQLi Reference](#)

Example

Return the number of rows in a result set:

```
<?php
$con=mysqli_connect("localhost","my_user","my_password","my_db");
// Check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$sql="SELECT Lastname,Age FROM Persons ORDER BY Lastname";

if ($result=mysqli_query($con,$sql))
{
    // Return the number of rows in result set
    $rowcount=mysqli_num_rows($result);
    printf("Result set has %d rows.\n",$rowcount);
    // Free result set
    mysqli_free_result($result);
}

mysqli_close($con);
?>
```

PHP mysqli_query() Function

 [PHP MySQLi Reference](#)

Example

Perform queries against the database:

```
<?php
$con=mysqli_connect("localhost","my_user","my_password","my_db");
// Check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

// Perform queries
mysqli_query($con,"SELECT * FROM Persons");
mysqli_query($con,"INSERT INTO Persons (FirstName,LastName,Age)
VALUES ('Glenn','Quagmire',33)");

mysqli_close($con);
?>
```

PHP mysqli_fetch_array() Function

 [PHP MySQLi Reference](#)

Example

Fetch a result row as a numeric array and as an associative array:

```
<?php
$con=mysqli_connect("localhost","my_user","my_password","my_db");
// Check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$sql="SELECT Lastname,Age FROM Persons ORDER BY Lastname";
$result=mysqli_query($con,$sql);

// Numeric array
$row=mysqli_fetch_array($result,MYSQLI_NUM);
printf ("%s (%s)\n",$row[0],$row[1]);

// Associative array
$row=mysqli_fetch_array($result,MYSQLI_ASSOC);
printf ("%s (%s)\n",$row["Lastname"],$row["Age"]);

// Free result set
mysqli_free_result($result);

mysqli_close($con);
?>
```

PHP mysqli_close() Function

 [PHP MySQLi Reference](#)

Example

Close a previously opened database connection:

```
<?php
$con=mysqli_connect("localhost","my_user","my_password","my_db");

// ....some PHP code...

mysqli_close($con);
?>
```

PHP `mysqli_affected_rows()` Function

 [PHP MySQLi Reference](#)

Example

Print out affected rows from different queries:

```
<?php
$con=mysqli_connect("localhost","my_user","my_password","my_db");
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

// Perform queries and print out affected rows
mysqli_query($con,"SELECT * FROM Persons");
echo "Affected rows: " . mysqli_affected_rows($con);

mysqli_query($con,"DELETE FROM Persons WHERE Age>32");
echo "Affected rows: " . mysqli_affected_rows($con);

mysqli_close($con);
?>
```

PHP 5 MySQLi Functions

Function	Description
<u>mysqli_affected_rows()</u>	Returns the number of affected rows in the previous MySQL operation
<u>mysqli_autocommit()</u>	Turns on or off auto-committing database modifications
<u>mysqli_change_user()</u>	Changes the user of the specified database connection
<u>mysqli_character_set_name()</u>	Returns the default character set for the database connection
<u>mysqli_close()</u>	Closes a previously opened database connection
<u>mysqli_commit()</u>	Commits the current transaction
<u>mysqli_connect_errno()</u>	Returns the error code from the last connection error
<u>mysqli_connect_error()</u>	Returns the error description from the last connection error
<u>mysqli_connect()</u>	Opens a new connection to the MySQL server
<u>mysqli_data_seek()</u>	Adjusts the result pointer to an arbitrary row in the result-set
<u>mysqli_debug()</u>	Performs debugging operations
<u>mysqli_dump_debug_info()</u>	Dumps debugging info into the log
<u>mysqli_errno()</u>	Returns the last error code for the most recent function call
<u>mysqli_error_list()</u>	Returns a list of errors for the most recent function call
<u>mysqli_error()</u>	Returns the last error description for the most recent function call
<u>mysqli_fetch_all()</u>	Fetches all result rows as an associative array, a numeric array, or both
<u>mysqli_fetch_array()</u>	Fetches a result row as an associative, a numeric array, or both
<u>mysqli_fetch_assoc()</u>	Fetches a result row as an associative array
<u>mysqli_fetch_field_direct()</u>	Returns meta-data for a single field in the result set, as an object
<u>mysqli_fetch_field()</u>	Returns the next field in the result set, as an object
<u>mysqli_fetch_fields()</u>	Returns an array of objects that represent the fields in a result set
<u>mysqli_fetch_lengths()</u>	Returns the lengths of the columns of the current row in the result set
<u>mysqli_fetch_object()</u>	Returns the current row of a result set, as an object

<u>mysqli_fetch_row()</u>	Fetches one row from a result-set and returns it as an enumerated array
<u>mysqli_field_count()</u>	Returns the number of columns for the most recent query
<u>mysqli_field_seek()</u>	Sets the field cursor to the given field offset
<u>mysqli_field_tell()</u>	Returns the position of the field cursor
<u>mysqli_free_result()</u>	Frees the memory associated with a result
<u>mysqli_get_charset()</u>	Returns a character set object
<u>mysqli_get_client_info()</u>	Returns the MySQL client library version
<u>mysqli_get_client_stats()</u>	Returns statistics about client per-process
<u>mysqli_get_client_version()</u>	Returns the MySQL client library version as an integer
<u>mysqli_get_connection_stats()</u>	Returns statistics about the client connection
<u>mysqli_get_host_info()</u>	Returns the MySQL server hostname and the connection type
<u>mysqli_get_proto_info()</u>	Returns the MySQL protocol version
<u>mysqli_get_server_info()</u>	Returns the MySQL server version
<u>mysqli_get_server_version()</u>	Returns the MySQL server version as an integer
<u>mysqli_info()</u>	Returns information about the most recently executed query
<u>mysqli_init()</u>	Initializes MySQLi and returns a resource for use with <code>mysqli_real_connect()</code>
<u>mysqli_insert_id()</u>	Returns the auto-generated id used in the last query
<u>mysqli_kill()</u>	Asks the server to kill a MySQL thread
<u>mysqli_more_results()</u>	Checks if there are more results from a multi query
<u>mysqli_multi_query()</u>	Performs one or more queries on the database
<u>mysqli_next_result()</u>	Prepares the next result set from <code>mysqli_multi_query()</code>
<u>mysqli_num_fields()</u>	Returns the number of fields in a result set
<u>mysqli_num_rows()</u>	Returns the number of rows in a result set
<u>mysqli_options()</u>	Sets extra connect options and affect behavior for a connection

<u>mysqli_ping()</u>	Pings a server connection, or tries to reconnect if the connection has gone down
mysqli_prepare()	Prepares an SQL statement for execution
<u>mysqli_query()</u>	Performs a query against the database
<u>mysqli_real_connect()</u>	Opens a new connection to the MySQL server
<u>mysqli_real_escape_string()</u>	Escapes special characters in a string for use in an SQL statement
mysqli_real_query()	Executes an SQL query
mysqli_reap_async_query()	Returns the result from async query
<u>mysqli_refresh()</u>	Refreshes tables or caches, or resets the replication server information
<u>mysqli_rollback()</u>	Rolls back the current transaction for the database
<u>mysqli_select_db()</u>	Changes the default database for the connection
<u>mysqli_set_charset()</u>	Sets the default client character set
mysqli_set_local_infile_default()	Unsets user defined handler for load local infile command
mysqli_set_local_infile_handler()	Set callback function for LOAD DATA LOCAL INFILE command
<u>mysqli_sqlstate()</u>	Returns the SQLSTATE error code for the last MySQL operation
<u>mysqli_ssl_set()</u>	Used to establish secure connections using SSL
<u>mysqli_stat()</u>	Returns the current system status
<u>mysqli_stmt_init()</u>	Initializes a statement and returns an object for use with mysqli_stmt_prepare()
mysqli_store_result()	Transfers a result set from the last query
<u>mysqli_thread_id()</u>	Returns the thread ID for the current connection
<u>mysqli_thread_safe()</u>	Returns whether the client library is compiled as thread-safe
mysqli_use_result()	Initiates the retrieval of a result set from the last query executed using the mysqli_real_query()
mysqli_warning_count()	Returns the number of warnings from the last query in the connection