



Utilizzare MySQL attraverso PHP

Matteo Manzali
INFN - Università degli Studi di Ferrara



Introduzione

- Tutto il materiale presente in queste slides (e molto altro) lo potete trovare su www.w3schools.com

w3schools.com

- Sono disponibili diverse modalità per accedere e lavorare su un database MySQL utilizzando PHP:
 - **MySQL extension**: deprecata dal 2012 e non più presente a partire dal versione 7 di PHP.
 - **MySQLi extension**: offre due API differenti (Object Oriented e procedurale).
 - **PDO**: offre una API Object Oriented e permette di astrarre la tipologia del database (supporta MySQL, SQLite, PostgreSQL, Microsoft SQL Server, etc...)
- In questo corso verranno utilizzate le MySQLi extensions seguendo l'approccio procedurale.

mysqli_connect

- La funzione `mysqli_connect` permette di connettersi ad un database MySQL:

`mysqli_connect(host, username, password, dbname, port, socket);`

- Esempio di utilizzo:

```
<?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();
}
?>
```

- La funzione `mysqli_connect_errno()` ritorna l'error code dell'ultimo errore di connessione (se c'è stato un errore).
- In alternativa si può controllare che `$con` sia non nullo (se è nullo allora si è verificato un errore).

mysqli_close

- La funzione `mysqli_close` permette di connettersi ad un database MySQL:

`mysqli_close(connection);`

- Esempio di utilizzo:

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

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

mysqli_close($con);
?>
```

- La funzione ritorna un valore boolean, può quindi essere controllato per verificare se la connessione è stata chiusa correttamente.

mysqli_query

- Tutte le operazioni di inserimento, rimozione, lettura e modifica su un database MySQL vengono effettuate attraverso la funzione:

mysqli_query(connection, query, resultmode);

- Dove:
 - connection: rappresenta la connessione al server
 - query: una stringa che contiene la query SQL da eseguire
 - resultmode: una costante che di default vale MYSQLI_STORE_RESULT, ma può venire settata a MYSQLI_USE_RESULT in caso di una grande quantità di dati (per recuperare un risultato alla volta).
- Ritorna FALSE in caso di errore. Ritorna un “result set” in caso di SELECT, SHOW, DESCRIBE o EXPLAIN. Ritorna TRUE negli altri casi.

leggere un “result set”

- La funzione `mysqli_num_rows` ritorna il numero di righe di un “result set” ottenuto da una `mysqli_query`:

```
mysqli_num_rows(result);
```

- La funzione `mysqli_fetch_assoc` recupera una riga rimuovendola da un “result set” e ritorna un array associativo:

```
mysqli_fetch_assoc(result);
```

- La funzione `mysqli_free_result` rimuove tutte le righe di un “result set” (i.e. libera la memoria):

```
mysqli_free_result(result);
```

- N.B. Se si legge **tutto** il “result set” utilizzando `mysqli_fetch_assoc` non serve chiamare `mysqli_free_result` in quanto il “result set” è già stato eliminato.

esempio: create db

```
1  <?php
2  $servername = "localhost:8889";
3  $username = "root";
4  $password = "root";
5
6  // Create connection
7  $conn = mysqli_connect($servername, $username, $password);
8  // Check connection
9  ▼ if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error()."\n");
11 }
12
13 // Create database
14 $sql = "CREATE DATABASE MyDB";
15 ▼ if (mysqli_query($conn, $sql)) {
16     echo "Database created successfully\n";
17 ▼ } else {
18     echo "Error creating database: " . mysqli_error($conn)."\n";
19 }
20
21 mysqli_close($conn);
22 ?>
```

esempio: create table

```
1  <?php
2  $servername = "localhost:8889";
3  $username = "root";
4  $password = "root";
5  $dbname = "MyDB";
6
7  // Create connection
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9  // Check connection
10 ▼ if (!$conn) {
11      die("Connection failed: " . mysqli_connect_error()."\n");
12  }
13
14  // sql to create table
15  $sql = "CREATE TABLE MyGuests (
16      id INT(6) UNSIGNED PRIMARY KEY,
17      firstname VARCHAR(30) NOT NULL,
18      lastname VARCHAR(30) NOT NULL,
19      email VARCHAR(50),
20      reg_date TIMESTAMP
21  )";
22
23 ▼ if (mysqli_query($conn, $sql)) {
24      echo "Table MyGuests created successfully\n";
25 ▼ } else {
26      echo "Error creating table: " . mysqli_error($conn)."\n";
27  }
28
29  mysqli_close($conn);
30  ?>
```


esempio: insert row

```
1  <?php
2  $servername = "localhost:8889";
3  $username = "root";
4  $password = "root";
5  $dbname = "myDB";
6
7  // Create connection
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9  // Check connection
10 ▼ if (!$conn) {
11     die("Connection failed: " . mysqli_connect_error()."\n");
12 }
13
14 $sql = "INSERT INTO MyGuests (id, firstname, lastname, email) VALUES ('0', 'John', 'Doe',
        'john@example.com')";
15
16 ▼ if (mysqli_query($conn, $sql)) {
17     echo "New record created successfully\n";
18 ▼ } else {
19     echo "Error: " . $sql . "\n" . mysqli_error($conn)."\n";
20 }
21
22 mysqli_close($conn);
23 ?>
```

esempio: select rows

```
1  <?php
2  $servername = "localhost:8889";
3  $username = "root";
4  $password = "root";
5  $dbname = "myDB";
6
7  // Create connection
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9  // Check connection
10 ▼ if (!$conn) {
11      die("Connection failed: " . mysqli_connect_error()."\n");
12  }
13
14  $sql = "SELECT id, firstname, lastname, email FROM MyGuests";
15  $result = mysqli_query($conn, $sql);
16
17 ▼ if (mysqli_num_rows($result) > 0) {
18      // output data of each row
19 ▼      while($row = mysqli_fetch_assoc($result)) {
20          echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
21              $row["lastname"]. " - Mail: " . $row["email"] . "\n";
22      }
23 ▼ } else {
24     echo "0 results\n";
25 }
26 mysqli_close($conn);
27 ?>
```