

## PHP & MySql – estensione MySQLi

Collegarsi ad un database e interagire con i dati in esso archiviati è indubbiamente uno dei compiti principali di un sito Web dinamico costruito con PHP.

PHP ci dà la possibilità di connetterci con un numero elevatissimo di database server (MySql, PostgreSQL, Oracle, Microsoft Sql Server, Access, Sybase, Informix, mSql ecc.).

Prenderemo in considerazione **MySql**, che è il database server che si è affermato prepotentemente negli ultimi anni per la sua velocità e la sua stabilità, oltre che per il fatto di essere open source.

Lo sviluppatore interessato a costruire delle applicazioni che possano colloquiare con MySQL, il DBMS (Database Management System) Open Source per la gestione di database relazionali più utilizzato in Rete con PHP, ha a sua disposizione principalmente tre scelte corrispondenti ad altrettante estensioni:

### 1) estensione nativa per MySQL

N.B. presente dalla prima versione 2.0 di PHP ma deprecata a partire dalla versione 5.5

### 2) estensione MySQLi (MySQL improved)

N.B. utilizzabile a partire dalla versione PHP 5.0 sia con l'approccio procedurale sia con l'approccio Object Oriented

### 3) estensione PHP Data Objects (PDO)

N.B. completamente orientata alla programmazione Object Oriented

**MySQLi** è l'abbreviazione di **MySQL improved** ("perfezionato") ed è il nome di un'estensione per il linguaggio PHP che è stata realizzata per mettere a disposizione degli sviluppatori delle funzionalità più avanzate rispetto a quelle messe a disposizione dall'estensione **MySQL nativa** per l'interazione tra le applicazioni Web based e il noto DBMS Open Source MySQL.

Può essere utilizzata sia all'interno di applicazioni realizzate seguendo il paradigma procedurale sia all'interno di applicazioni realizzate seguendo il paradigma Object Oriented;

- fornisce il supporto nativo al protocollo binario di MySQL introdotto con la versione 4.1;
- supporta **stored procedures, query multiple e transactions**;
- permette impostazioni avanzate per la connessione tramite **mysqli\_init()**;
- supporta le **prepared statements** per il caching delle interrogazioni;
- garantisce prestazioni più elevate dell'estensione nativa MySQL, è ritenuta più sicura e fornisce in genere migliori risposte in fase di debugging.

**Noi useremo l'estensione MySQLi con approccio procedurale (punto 2)**

## A Eseguire una query su un db MySQL utilizzando l'estensione MySQLi

```
<?php // INIZIO script PHP
```

Apertura di una connessione tra PHP ed il DBMS MySQLi

```
$conn = mysqli_connect (hostname, username, password, database) OR  
die ("Proprio_messaggio_di_errore " . mysqli_connect_error() . " " . mysqli_connect_errno() );
```

### Passo 1) Apertura di una connessione con il db

- **hostname**: il nome di rete della macchina ospitante il Database server, ad esempio un indirizzo IP o "localhost" nel caso di installazioni locali;
- **username**: il nome dell'utente da utilizzare per la connessione, esso dovrà avere i privilegi necessari per l'accesso e la manipolazione dei dati;
- **password**: la parola chiave necessaria per l'autenticazione dell'utente che si desidera utilizzare in connessione;
- **database**: il nome del database che si desidera interfacciare alla propria applicazione

(N.B. DA OMETTERE IN CASO DI CREATE DATABASE o DROP DATABASE)

Fase1: Costruzione di una query (caso CREATE DATABASE oppure DROP DATABASE)

```
//Fase 1: Costruzione di una stringa contenente il comando SQL da eseguire (caso CREATE DATABASE)  
$query = "CREATE DATABASE X;";
```

Fase 2: Esecuzione di una query (caso CREATE DATABASE) oppure DROP DATABASE)

```
// Fase 2: Invio del comando SQL al DBMS  
$risultato = mysqli_query ($conn, $query) OR  
die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );
```

Chiusura della connessione

```
mysqli_close($conn) OR  
die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );
```

```
?> // FINE script PHP
```

## B Eseguire una query (tutte le altre) su un db MySQL utilizzando l'estensione MySQLi

```
<?php // INIZIO script PHP
```

Apertura di una connessione tra PHP ed il DBMS MySQLi

```
$conn = mysqli_connect (hostname, username, password, database) OR  
die ("Proprio_messaggio_di_errore " . mysqli_connect_error() . " " . mysqli_connect_errno() );
```

N.B. In caso di CREATE o DROP DATABASE ovviamente non occorrerà passare il parametro **database** (chiamata con 3 soli parametri)

### Passo 1) Apertura di una connessione con il db

- **hostname**: il nome di rete della macchina ospitante il Database server, ad esempio un indirizzo IP o "localhost" nel caso di installazioni locali;
- **username**: il nome dell'utente da utilizzare per la connessione, esso dovrà avere i privilegi necessari per l'accesso e la manipolazione dei dati;
- **password**: la parola chiave necessaria per l'autenticazione dell'utente che si desidera utilizzare in connessione;
- **database**: il nome del database che si desidera interfacciare alla propria applicazione

Fase1: Costruzione di una query di inserimento/modifica/cancellazione/altro (INSERT/UPDATE/DELETE/etc.)

```
//Fase 1: Costruzione di una stringa contenente il comando SQL da eseguire (caso INSERT)
```

```
$query = "INSERT INTO Tabella (id, password) VALUES ('pippo', '1234');";      query a valori fissi  
oppure
```

```
$query = " INSERT INTO Tabella (id, password) VALUES ('$id', '$pswd');";      query a valori variabili (con uso di un form HTML) ottenuti  
attraverso l'utilizzo degli array globali $_GET[] o $_POST[]
```

Fase 2: Esecuzione di una query di inserimento/modifica/cancellazione/altro (INSERT/UPDATE/DELETE/etc.)

```
// Fase 2: Invio del comando SQL al DBMS, generazione e ricezione del result set
```

```
$risultato = mysqli_query ($conn, $query) OR  
die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );
```

Fase 3: valutazione esito di una query di inserimento/modifica/cancellazione/altro (INSERT/UPDATE/DELETE/etc.)

```
// Fase 3: Valutazione dell'esito dell'operazione di inserimento/modifica/cancellazione/etc. (OPZIONALE)
```

```
$numrighe = mysqli_affected_rows ($conn) ;
```

Il risultato della chiamata a **mysqli\_affected\_rows ( )** è un **numero:che indica le righe** coinvolte dall'operazione di inserimento/aggiornamento/cancellazione proposto

Chiusura della connessione

```
mysqli_close($conn) OR  
die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );
```

```
?> // FINE dello script PHP
```

## C Eseguire una query (SELECT) su un db MySQL utilizzando l'estensione MySQLi

**<?php** // INIZIO script PHP

Apertura di una connessione tra PHP ed il DBMS MySQLi

```
$conn = mysqli_connect (hostname, username, password, database) OR  
die ("Proprio_messaggio_di_errore " . mysqli_connect_error( ) . " " . mysqli_connect_errno( ) );
```

### Passo 1) Apertura di una connessione con il db

- **hostname**: il nome di rete della macchina ospitante il Database server, ad esempio un indirizzo IP o "localhost" nel caso di installazioni locali;
- **username**: il nome dell'utente da utilizzare per la connessione, esso dovrà avere i privilegi necessari per l'accesso e la manipolazione dei dati;
- **password**: la parola chiave necessaria per l'autenticazione dell'utente che si desidera utilizzare in connessione;
- **database**: il nome del database che si desidera interfacciare alla propria applicazione

Fase1: Costruzione di una query di lettura (caso SELECT)

//Fase 1: Costruzione di una stringa contenente il comando SQL da eseguire (caso SELECT)

```
$query = "SELECT colonna1, ..colonna n FROM Tabella WHERE id = 'pippo';";  
oppure
```

query a valori fissi

```
$query = "SELECT colonna1, ..colonna n FROM Tabella WHERE id = '$id';";
```

query a valori variabili (con uso di un form HTML)ottenuti attraverso l'utilizzo degli array globali `$_GET[ ]` o `$_POST[ ]`

Fase 2: Esecuzione di una query di lettura (caso SELECT)

// Fase 2: Invio del comando SQL al DBMS, generazione e ricezione del result set

```
$risultato = mysqli_query ($conn, $query) OR  
die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );
```

Fase 3: Estrazione dei dati da una query di lettura (caso SELECT)

// Fase 3: Estrazione dei dati presenti nel result set (N.B. si utilizza un CICLO)

Scelta 1

```
while ($riga = mysqli_fetch_array ($risultato, MYSQLI_NUM | MYSQLI_ASSOC | MYSQLI_BOTH )
```

```
{
```

```
echo $riga[0] . $riga[1] . ..... . $riga[m]; //con indice 0 per prima colonna, 1 per seconda colonna,....., m per m-esima colonna
```

```
oppure
```

```
echo $riga['colonna_1'] . $riga['colonna_2'] . ..... . $riga['colonna_m']; //con colonna_1 per nome 1° colonna, colonna_2 per nome 2° colonna, etc.....
```

```
.....
```

```
}
```

Il risultato della chiamata a **mysqli\_fetch\_array ( )** è un **array**:

- **solo numerico** (se il 2° parametro è = **MYSQLI\_NUM**),
- **solo associativo** (se il 2° parametro è = **MYSQLI\_ASSOC**),
- **oppure entrambi** (se il 2° parametro è = **MYSQLI\_BOTH**) **default**
- **oppure NULL** quando è terminato il result set

### Scelta 2

```
$n = mysqli_num_rows ($risultato);
```

```
for ($i = 1; $i <= $n; $i++)
```

```
{
```

```
    $riga = mysqli_fetch_array ($risultato, MYSQLI_NUM | MYSQLI_ASSOC | MYSQLI_BOTH);
```

```
    $campo_1 = $riga[0] oppure $campo_1= $riga['colonna_1'];
```

```
    $campo_2 = $riga[1] oppure $campo_2= $riga['colonna_2'];
```

```
    .....
```

```
    $campo_n = $riga[m] oppure $campo_n= $riga['colonna_m'];
```

```
    .....
```

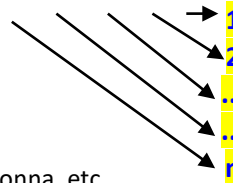
```
}
```

//OVVIAMENTE MENO EFFICIENTE DELLA PRECEDENTE – CI SONO 2 CHIAMATE A FUNZIONE

//N.B. ATTENZIONE MATRICE DEI RISULTATI – VETTORI PARALLELI – SI PARTE DA INDICE 1

*\$risultato* è la seguente **MATRICE n righe x m colonne**

*\$riga* è il vettore  
corrente  
monodimensionale



	colonna_1	colonna_2	.....	colonna_m
F_01	GIALLI	ERNESTO	.....	1998-12-12
F_02	BIANCHI	MARIO	.....	2000-03-04
F_03	ROSSI	DAVIDE	.....	1999-11-07
.....	.....	.....	.....	.....
F_0n	VERDI	ANTONIO	.....	1997-08-25
	0	1		m

N.B. Con indice 0 per prima colonna, 1 per seconda colonna,....., n-1 per n-esima colonna, etc

Con *colonna\_1* per nome 1° colonna, *colonna\_2* per nome 2° colonna, etc

### Scelta 3

```
while ($riga = mysqli_fetch_array ($risultato, MYSQLI_NUM | MYSQLI_ASSOC))
```

```
{
```

```
    foreach ($riga as $valore)
```

```
    {
```

```
        echo $valore . " ";
```

```
    }
```

```
    .....
```

```
}
```

N.B Se si usa la costante **MYSQLI\_BOTH** i dati letti dalla tabella appariranno raddoppiati in quanto *\$riga* sarà un array che dovrà contenere sia gli indici numerici, sia gli indici stringa per ogni singolo valore per garantire così entrambi i tipi di accesso

Valore 1	Valore 1	Valore 2	Valore 2	.....	.....	Valore m	Valore m
0	'colonna 1'	1	'colonna 2'	.....	.....	m	'colonna m'
<i>\$riga</i>							

N.B. QUALUNQUE SIA LA SCELTA FATTA TRA LE 3 PROPOSTE, in alternativa alla funzione `mysqli_fetch_array (...)` posso utilizzare la funzione

`mysqli_fetch_assoc (...)` il cui risultato è **ESCLUSIVAMENTE** un array **ASSOCIATIVO** dove gli indici sono uguali ai nomi dei campi o attributi

oppure **NULL** quando è terminato il result set

Fase 4: Formattazione dei risultati della query di lettura (caso SELECT semplice)

*// Fase 4: Costruzione del codice HTML per visualizzare i risultati nella pagina (OPZIONALE – formattazione dei risultati)*

Consiste nell'utilizzare durante la fase precedente di estrazione dei dati all'interno del ciclo utilizzato gli opportuni TAG del linguaggio HTML per formattare i dati ottenuti creando anche tabelle o report riepilogativi (Esempio TAG TABLE, TR e TD)

Fase 5: Liberazione della memoria (caso SELECT semplice)

*// Fase 5: Liberare la memoria allocata per i dati restituiti dalla query (utile solo per query che restituiscono una grossa mole di dati)*

`mysqli_free_result ($risultato);`

`//N.B. Non restituisce alcun valore....NON VA TESTATA con il meccanismo della funzione die()`

Chiusura della connessione

`mysqli_close($conn) OR`

`die ("Proprio_messaggio_di_errore " . mysqli_error($conn) . " " . mysqli_errno($conn) );`

`?> // FINE script PHP`