

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 DDL o DCL 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( ) );
```

```
echo "Apertura connessione OK! <BR>";
```

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) );
```

```
echo "Esecuzione query OK! <BR>";
```

Chiusura della connessione

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

```
echo "Chiusura connessione OK!";
```

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

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)**

B Eseguire una query **DML** (INSERT/UPDATE/DELETE) 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 questo caso occorrerà passare il parametro **database** (chiamata con 4 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 cui si desidera collegarsi dalla propria applicazione

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

```
//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 parametrici (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**)

```
// 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) );
```

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

```
// Fase 3: Valutazione dell'esito dell'operazione di inserimento/modifica/cancellazione (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 **QL** (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( ) );
```

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 colonna_1, ..colonna_n FROM tabella WHERE id = 'pippo'";  
oppure
```

```
$query = "SELECT colonna_1, ..colonna_n FROM tabella WHERE id = '$id'";
```

query a valori fissi

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.....  
    .....  
}
```

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 cui si desidera collegarsi dalla propria applicazione

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

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`