

Connettersi ad un database MySQL

Per prima cosa vediamo come fa PHP a connettersi al MySQL Server. Allo scopo soccorre la funzione **mysql_connect()** che si utilizza con la seguente sintassi:

```
mysql_connect(server, utente, password);
```

Ad esempio:

```
$myconn = mysql_connect('localhost', 'pippo', 'xxxxxx') or die('Errore...');
```

si noti l'utilizzo del comando *die()* il cui scopo è, in caso di errore, bloccare l'elaborazione dello script e stampare a video un messaggio.

Questa operazione - la connessione al server MySQL - è da considerare preliminare ad ogni altra operazione sui database. Una volta concluse le operazioni sul database è possibile chiudere la connessione al server in modo esplicito mediante la funzione **mysql_close()** in questo modo:

```
mysql_close($myconn);
```

Selezionare un database

Una volta stabilita la connessione è necessario selezionare uno specifico db sul quale lavorare. A questo scopo PHP ci fornisce la funzione **mysql_select_db()** da utilizzarsi con la seguente sintassi:

```
mysql_select_db(database, connessione);
```

Ad esempio:

```
mysql_select_db('mio_database', $myconn) or die('Errore...');
```

Effettuare una query

Per prima cosa vediamo come è possibile recuperare (leggere) dei dati presenti nel nostro database. Per fare questo dobbiamo formulare ed eseguire una *query*, la quale consiste in una interrogazione che lo sviluppatore rivolge al database utilizzando il [linguaggio SQL](#). Per fare ciò si fa ricorso alla funzione **mysql_query()** con la seguente sintassi:

```
mysql_query(query, connessione);
```

Ad esempio:

```
$query = "SELECT * FROM tabella WHERE id > 100";  
$result = mysql_query($query, $myconn) or die('Errore...');
```

Si noti che l'indicazione della connessione è facoltativa; se omessa lo script utilizzerà l'ultima connessione aperta.

Leggere i record restituiti da una query di SELECT

Una volta "recuperati" i dati dal database mediante una *SELECT* dovremo preoccuparci di ciclarli, ad esempio, per stamparli a video. A tal fine ci serviranno una serie di altre funzioni come, ad esempio, `mysql_num_rows()` e `mysql_fetch_rows()`.

Vediamo quindi un esempio completo: poniamo di voler recuperare dalla tabella "amici" una serie di dati (nome, cognome e telefono) e di volerli stampare a video per ogni occorrenza trovata nel nostro database. Ecco il codice completo del nostro script PHP opportunamente commentato:

```
<?php
// mi connetto al MySql Server
$myconn = mysql_connect('localhost', 'pippo', 'xxxxxx') or die('Errore...');

// seleziono il database degli amici
mysql_select_db('database_degli_amici', $myconn) or die('Errore...');

// imposto ed eseguo la query
$query = "SELECT nome, cognome, telefono FROM amici ORDER BY cognome ASC";
$result = mysql_query($query, $myconn) or die('Errore...');

// conto il numero di occorrenze trovate nel db
$numrows = mysql_num_rows($result);

// se il database è vuoto lo stampo a video
if ($numrows == 0){
    echo "Database vuoto!";
}
// se invece trovo delle occorrenze...
else
{
    // avvio un ciclo for che si ripete per il numero di occorrenze trovate
    for ($x = 0; $x < $numrows; $x++){

        // recupero il contenuto di ogni record trovato
        $resrow = mysql_fetch_row($result);
        $nome = $resrow[0];
        $cognome = $resrow[1];
        $telefono = $resrow[2];

        // stampo a video il risultato
        echo "nome: <b>" . $nome . "</b><br/>";
        echo "cognome: <b>" . $cognome . "</b><br/>";
        echo "telefono: <b>" . $telefono . "</b>";
    }
}

// chiudo la connessione
mysql_close($myconn);
?>
```

Quello che abbiamo fatto qui sopra dovrebbe esservi abbastanza chiaro (ricordate la lezione sui cicli?)... Resta pertanto da chiarire solo il significato di due funzioni specifiche che abbiamo utilizzato:

- **mysql_num_rows()** - Serve per conteggiare il numero di records trovati all'interno del nostro db sulla base di una data query;
- **mysql_fetch_row()** - Recupera il contenuto dei records trovati. Più precisamente restituisce una array contenente i valori di ogni campo riscontrato nel recordset che potremo poi richiamare specificando il relativo indice numerico.

Di seguito, per completezza, alcune considerazioni in merito al codice del nostro esempio.

In alternativa a *mysql_fetch_row()* avremmo potuto utilizzare **mysql_fetch_assoc()**, in tal caso i dati dal recordset sarebbero stati recuperabili mediante l'indicazione del nome del campo (invece che dell'indice numerico). Ad esempio:

```
// usando mysql_fetch_rows() abbiamo scritto...
$nome = $resrow[0];

// ...usando mysql_fetch_assoc() avremmo scritto
$nome = $resrow['nome'];
```

In alternativa a questi due, infine, avremmo potuto usare **mysql_fetch_array()** che supporta, indistintamente, entrambe le tecniche di chiamata.

Ancora, nel nostro esempio abbiamo ciclato il recordset utilizzando un comune *ciclo for* ma avremmo potuto usare, forse più correttamente, anche *while* in questo modo:

```
// ...

if ($numrows == 0){
    echo "Database vuoto!";
}
else
{
    while ($resrow = mysql_fetch_row($result)) {
        $nome = $resrow[0];
        $cofnome = $resrow[1];
        $telefono = $resrow[2];

        // Stampo a video il risultato
        // ...
    }
}

// ...
```

Cenni di SQL: INSERT INTO, UPDATE e DELETE

Per finire vediamo brevemente come eseguire altre importanti operazioni con i database attraverso i più comuni comandi del linguaggio SQL.

Con [INSERT INTO](#) si inseriscono nuovi dati nel db, con [UPDATE](#) si aggiornano dei dati già presenti, con [DELETE](#) si cancellano dei dati.

Dal punto di vista di PHP queste operazioni non differiscono tra loro, l'unica cosa che cambia è la query che viene eseguita, ma questo discorso attiene al linguaggio SQL. Facciamo degli esempi:

Per INSERT INTO useremo:

```
mysql_query("INSERT INTO tabella VALUES('valore1','valore2','valore3')");
```

Per UPDATE useremo:

```
mysql_query("UPDATE tabella SET campo1='valore1', campo2='valore2',  
campo3='valore3' WHERE id = 1");
```

Per DELETE useremo:

```
mysql_query("DELETE FROM tabella WHERE id = 1");
```

Attenzione! se nelle query di UPDATE e DELETE non usiamo la clausola "WHERE" verranno aggiornati/eliminati tutti i record del db!

Facciamo un esempio di utilizzo di queste query; vediamo come cancellare con PHP un record dal nostro database MySQL:

```
<?php  
// mi connetto al server MySQL  
$myconn = mysql_connect('localhost', 'pippo', 'xxxxxx') or die('Errore...');  
  
// mi connetto al database degli amici  
mysql_select_db('database', $myconn) or die('Errore...');  
  
// imposto ed eseguo la query  
$query = "DELETE FROM tabella WHERE id = 1";  
$result = mysql_query($query, $myconn) or die('Errore...');  
  
// chiudo la connessione  
mysql_close($myconn);  
?>
```

Per le altre operazioni basterà sostituire la query mantenendo inalterata la struttura del codice PHP.