APPUNTI DI PHP : V INFORMATICA SEZIONE G

mysql_connect()

Apre una connessione verso un DB MySQL.

```
$db = mysql_connect([$db_host[, $db_username[,
$db_password[,$newlink[,$clientflags]]]]]);
```

$db Restituisce un identificatore della connessione col DB (da utilizzare nelle chiamate alle successive funzioni di interrogazione) e **FALSE** in caso di fallimento
$db_host Nome host del database server e relativa porta. Default: localhost:3306
$db_username Nome dell'utente per il login al database. Di default corrisponde all'utente che possiede il processo corrente.
$db_password Password per il login al database. Di default è una stringa vuota
$newlink Di default, se viene effettuata una seconda chiamata a **mysql_connect** con gli stessi parametri, non viene aperta una nuova connessione ma la funzione restituisce l'ID della connessione esistente. Se questo parametro è impostato a **TRUE** viene aperta una nuova connessione ad ogni chiamata.
$clientflags Combinazione dei flag **MYSQL_CLIENT_COMPRESS** (usa la compressione dei dati), **MYSQL_CLIENT_IGNORE_SPACE** (permette gli spazi dopo i nomi delle funzioni) o **MYSQL_CLIENT_INTERACTIVE** (lascia trascorrere interactive_timeout secondi - anziché wait_timeout - di inattività prima di chiudere la connessione).

Compatibilità: il paramentro **$newlink** è stato indrodotto dal PHP 4.2.0; mentre **$clientflags** dal PHP 4.3.0
La connessione verrà chiusa al termine dell'esecuzione dello script, a meno che non venga chiusa esplicitamente da una chiamata a **mysql_close()**.

mysql_close()

Autore: **mario-online** - (Revisione: **bluviolin**) - Ultimo Aggiornamento: **2004-09-13 10:54:30** - Data di creazione: **2004-09-13 10:54:30**
Tipo Infobox: **DESCRIPTION** - Skill: **2- JUNIOR**

Chiude una connessione con un DB MySQL

```
$success = mysql_close ([$db])
```

$success La funzione restituisce **TRUE** in caso di successo e **FALSE** in caso di fallimento.

$db Identificatore della connessione da chiudere (restituito da **mysql_connect()**. Se non specificato, chiude l'ultima connessione aperta.

mysql_query()

Indica al database server MySQL di eseguire la query passata come argomento.

mysql_query($query, $link)
il parametro opzionale **$link** serve ad indicare quale connessione attiva utilizzare per eseguire il codice SQL presente nella stringa **$query**. se non viene indicato il parametro **$link** allora PHP tenta di utilizzare l'ultima connessione aperta, se non ce ne è nessuna verrà tentata una connessione senza nessun parametro: mysql_connect().
In caso di errore, sia legato all'impossibilità di connessione ad un db sia a errori nel codice SQL, mysql_query() ritornerà FALSE.
Nel caso di risultato positivo il valore ritornato sarà un identificatore di risorsa nel caso di query "descrittived" come ad esempio SELECT, SHOW etc. sarà invece un qualsiasi valore diverso da FALSE per query del tipo UPDATE, INSERT e altro. Attenzione che il valore ritornato non è necessariamente legato al numero di righe risultati dalla query.

Come gestire i dati ottenuti da una query

Una volta eseguita una query di SELECT è possibile eseguire delle istruzioni di codice per ognuno dei record risultanti.

PHP offre diverse funzioni che permettono di estrarre i valori dei campi dei record risultanti da una query di selezione. La differenza principale tra le varie possibilità offerte è data dalla tipologia di dato che viene associata al record.
mysql_fetch_row($result) estrae il record successivo nel record-set rappresentato da ***$result*** in forma di array. E' possibile cioè accedere ai valori dei singoli campi estratti utilizzando codice del tipo

```
$sql = "SELECT codice, descrizione FROM prodotti";
$result = mysql_query($sql);
$array = mysql_fetch_row($result);
print "CODICE: ".$array[0]."<br>";
print "DESCRIZIONE: ".$array[1]."<br>";
```

Verranno stampati il codice e la descrizione del primo prodotto trovato dalla query ***$sql***.
Se non esiste nessun record successivo il valore ritornato da mysql_fetch_row() sarà FALSE.

Una modo più comodo per accedere ai campi, soprattutto in presenza di query con un gran numero di dati estratti è **mysql_fetch_array($result)**
il comportamento di fetch dei dati è simile a quello di mysql_fetch_row(), il valore aggiunto è dato dal fatto che sarà possibile accedere al particolare campo indicando esattamente il suo nome, vediamo lo stesso esempio:

```
$sql = "SELECT codice, descrizione FROM prodotti";
$result = mysql_query($sql);
$array = mysql_fetch_array($result);
print "CODICE: ".$array['codice']."<br>";
print "DESCRIZIONE: ".$array['descrizione']."<br>";
```

come si può notare mysql_fetch_array() rende il codice più leggibile senza per questo influire sulle perfomance del recupero di dati.
Una possibile fonte di errore può essere invece il fatto che può capitare, lavorando con query su diverse tabelle, di dovere estrarre campi con lo stesso nome, in questo caso sarà necessario utilizzare degli alias all'interno del codice SQL perchè altrimenti il valore di ***$array[$nome_campo]*** sarà relativo solamente all'ultimo campo di nome ***$nome_campo*** presente nella query come nel caso di

```
$sql = "
SELECT prodotti.codice, prodotti.descrizione,
categoria.descrizione
FROM prodotti, descrizione
WHERE descrizione.id = prodotti.id_descrizione";
$result = mysql_query($sql);
$array = mysql_fetch_array($result);
print "CODICE Prod: ".$array['codice']."<br>";
print "DESCRIZIONE Prod: ".$array['descrizione']."<br>";
print "DESCRIZIONE Cat: ".$array['descrizione']."<br>";
```

mentre

```
$sql = "
SELECT prodotti.codice, prodotti.descrizione,
categoria.descrizione AS desc_cat
FROM prodotti, descrizione
WHERE descrizione.id = prodotti.id_descrizione";
$result = mysql_query($sql);
$array = mysql_fetch_array($result);
print "CODICE Prod: ".$array['codice']."<br>";
print "DESCRIZIONE Prod: ".$array['descrizione']."<br>";
print "DESCRIZIONE Cat: ".$array['desc_cat']."<br>";
```

ci mostra un output più consono alle nostre aspettative.

E' possibile utilizzare le funzioni di fetch ricorsivamente per estrarre tutti i record risultati da una query SQL:

```
$sql = "
SELECT prodotti.codice, prodotti.descrizione,
categoria.descrizione AS desc_cat
FROM prodotti, descrizione
WHERE descrizione.id = prodotti.id_descrizione";
$result = mysql_query($sql);
while ($array = mysql_fetch_array($result))
```

```
{
print "CODICE Prod: ".$array['codice']." - ";
print "DESCRIZIONE Prod: ".$array['descrizione']." - ";
print "DESCRIZIONE Cat: ".$array['desc_cat']."<br><br>";
}
```
la condizione del ciclo while sarà TRUE fino a che sarà possibile estrarre record dalla risorsa ottenuta dall'esecuzione della query, dopo l'ultimo record la condizione diventa FALSE e il ciclo termina.

Verificare il numero di record riscontrati o modificati dalla query

Autore: **mario-online** - Ultimo Aggiornamento: **2004-10-28 22:54:07** - Data di creazione: **2004-10-28 22:54:07**
Tipo Infobox: **DESCRIPTION** - Skill: **3- INTERMEDIATE**

Possiamo, senza necessariamente scorrere tutto il record-set, avere indicazioni sul numero di record che la nostra query ha estratto o modificato utilizzando la funzione **`mysql_num_rows($result).`**

Essa ritorna, solamente nel caso di query di tipo SELECT, il numero di record che sono stati estratti:
```
$result = mysql_query("SELECT * FROM prodotti");
$num_prodotti = mysql_num_rows($result);
```
`$num_prodotti` conterrà il numero totale di prodotti presenti nel database.
Si noti come lo stesso risultato si potrebbe ottenere abbassandosi di livello e facendo effettuare il conteggio direttamente al database:
```
$result = mysql_query("SELECT COUNT(*) AS tot, * FROM prodotti");
$array = mysql_fetch_array($result);
$num_prodotti = $array['tot'];
```
`$num_prodotti` conterrà nuovamente il numero totale di prodotti presenti nel database.

Nel caso invece di statement SQL che modificano in qualche modo i record presenti nella nostra tabella, ad esempio con
istruzioni di INSERT, UPDATE e di DELETE, per conoscere il numero di record "toccati" si utilizza la funzione
`mysql_affected_rows()`
Si porga la dovuta attenzione però ad alcune eccezioni che potrebbe indurre ad errori difficilmente debuggabili:
- nel caso in cui la query sia del tipo **`DELETE FROM`** ***`tabella`***, quindi una cancellazione di tutti i record presenti in ***`tabella`*** senza nessun vincolo di WHERE, la funzione ritornerà sempre e comunque 0.
- quando si effettua uno statement SQL di update, il motore MySQL non "tocca" i record che già contengono lo stesso valore che si vorrebbe inserire, quindi potrebbe esserci il caso in cui mysql_affected_rows() non ritorni esattamente il numero di record riscontrati dalla query ma solo quelli effettivamente modificati.

Gestione degli errori

Autore: **mario-online** - Ultimo Aggiornamento: **2004-10-28 23:39:56** - Data di creazione: **2004-10-28 23:39:56**
Tipo Infobox: **DESCRIPTION** - Skill: **2- JUNIOR**

Buona norma di programmazione è verificare e gestire sempre gli eventuali errori che possono nascere durante l'esecuzione, soprattutto quando i comandi contengono parti variabili come nel caso di query da fare eseguire al database.

In questo caso, o meglio ancora dopo ogni chiamata a funzione che interagisce con MySQL è bene mostrare, nel caso la funzione ritorni un valore FALSE spesso sintomo di un errore, le cause dell'errore.
Per fare ciò si utilizza la funzione **`mysql_error($link)`** al solito non specificando l'identificativo della connessione ***`$link`*** verrà considerata l'ultima connessone aperta come attiva.
In questo modo possiamo delineare una forma standard per l'esecuzione di query in questo modo:

```
$result = mysql_query($sql)
or die("<br>Invalid query: $query\n<BR>\n" . mysql_error());
```

In questo modo nel caso ci siano errori ritornati dal motore MySQL lo script si interromperà e ci verrà mostrato l'errore stesso e la query responsabile.

Ottenere l'ID dell'ultimo record inserito

Autore: **mario-online** - Ultimo Aggiornamento: **2004-10-28 23:15:45** - Data di creazione: **2004-10-28 23:15:45**
Tipo Infobox: **DESCRIPTION** - Skill: **3- INTERMEDIATE**

PHP interfaccia l'API di MySQL last_insert_id() per ritornare l'indice associato ad un campo AUTO_INCREMENT del record che abbiamo appena inserito, la funzione che lo permette è **`mysql_inserted_id($link)`**.

In pratica capita spesso di dover inserire in una tabella un record di definizione di una insieme di oggetti che verranno inseriti in un'altra tabella, ad esempio una nuova categoria di prodotti ed un elenco di prodotti legati alla nuova categoria.
Per fare questo in un'unica pagina diventa necessario conoscere l'ultimo valore che è stato generato da MySQL nel campo AUTO_INCREMENT della tabelle, consideriamo questo semplice esempio in cui supponiamo di ricevere in input la descrizione di una nuova categoria e di un prodotto ad essa associato, in pratica il risultato di una print_r dei valori che ci arrivano in POST sia qualcosa del tipo

```
Array
(
    [desc_cat] => 'cibi'
    [desc_prod] => 'pasta'
)
```

Possiamo inserire la nuova categoria e associarvi i nuovi prodotti direttamente in un unica pagina, tramite codice del tipo

```
$sql = "INSERT INTO categoria (id, descrizione) VALUES (NULL,
'".$_POST['descrizione']."')"; // sia id di tipo AUTO_INCREMENT
```

```
$result = mysql_query ($sql);
$new_id_cat = mysql_inserted_id();

$sql = "INSERT INTO prodotto (id, id_categoria, descrizione)
VALUES (NULL, '$new_id_cat', '".$_POST['descrizione']."')"; // sia
id di tipo AUTO_INCREMENT
$result = mysql_query ($sql);
```

Connessione a MySQL con PHP

Per stabilire una connessione ad un database MySQL dobbiamo avere a disposizione alcuni dati necessari:

- **Nome Host**
- **Nome Utente**
- **Password di accesso al DB**
- **Nome del Database**

Creiamo quindi un file .php per registrare i dati della connessione al DB MySQL.

```
1 2 $db_host     = "localhost";
3 $db_user     = "user_name";
4 $db_password = "password";
5 $db_database = "database_name";
6 ?>
```

In questo modo non abbiamo fatto altro che memorizzare i dati della connessione al DB all'interno di variabili. Le variabili saranno utilizzate nell'istruzione che permette la connessione al database.

In PHP l'istruzione per connettersi al database è la seguente:

```
1 $connessione=mysql_connect($db_host,$db_user,$db_password);
2 mysql_select_db($db_database,$connessione);
```

La prima istruzione stabilice una connessione ad un server MySQL e restituisce un *identificativo* di connessione MySQL in caso di successo oppure *FALSE* in caso di fallimento.

```
1 mysql_connect ( [string server
2                 [, string nome_utente
3                 [, string password
4                 [, bool nuova_connessione
5                 [, int client_flags]]]]] )
```

I seguenti valori predefiniti sono assunti per i parametri opzionali mancanti: server = '*localhost:3306*', nome_utente = *nome dell'utente* proprietario del processo server e **password** = *password vuota*.

Il parametro **server** può anche includere un numero di porta. Es. "*hostname:porta*" o un percorso ad un socket es. "*percorso/al/socket*" per il localhost.

La seconda istruzione specifica il nome del database al quale ci vogliamo connettere. Restituisce *TRUE* in caso di successo, *FALSE* in caso di fallimento.

```
1 bool mysql_select_db ( string nome_database
2                        [, resource identificativo_connessione])
```

mysql_select_db() imposta il database attualmente attivo sul server associato all'identificativo di connessione specificato. Se nessun identificativo di connesione è specificato, viene considerata l'ultima connessione aperta.

A questo punto è possibile scrivere le nostre query, con le quali possiamo interrogare o modificare il contenuto delle tabelle.

Il metodo appena descritto è solo uno dei modi per accedere a un database MySQL con PHP.

```
CREATE TABLE `articoli` (
`id` INT( 2 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`descrizione` VARCHAR( 200 ) NOT NULL
) ENGINE = innodb;
```

Creiamo lo script di connessione con PHP e HTML:

Ora veniamo alla creazione dello script che ci permetterà di inserire i dati nella tabella da noi creata denominata "articoli" come potete capire dal codice SQL abbiamo due campi: "id" e "descrizione". Il primo è la nostra chiave primaria ed è rappresentata da campo numerico autoincrementale, significa che ad ogni nuovo inserimento il numero si incrementa in automatico, perciò noi non dobbiamo mai inserirlo. Dobbiamo solamente inserire il secondo campo, "descrizione", che è rappresentato da un piccolo tempo esplicativo dell'articolo.

Ecco la prima pagina in HTML, che chiameremo **inserisci_descrizione.html**:

```
<html>

<head>
<meta http-equiv="Content-Language" content="it">

<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252">
<title>Inserisci descrizione Articolo</title>
</head>

<body>
<center>
<form method="POST" action="inserisci_database.php">
```

```
<p align="center">INSERISCI UN NUOVO ARTICOLO</p>
<input type="text" name="descrizione" size="100"></td>
<p align="center"><input type="submit" value="Inserisci" name="B1"></p>
</form>
<p> </p>
</center>
</body>

</html>
```

Semplicemente una piccola casella di testo che invia il dato inserito, nel nostro caso la descrizione, alla pagina inserisci_database.php in cui è incluso lo script PHP di inserimento nel database.

Ecco il codice della pagina "**inserisci_database.php**":

```
<?
$db_host="localhost";
$db_user="USER";
$db_password="PASS";
$db_name="test_connessione";
$descrizione = $_POST['descrizione'];
$db=mysql_connect($db_host,$db_user,$db_password);
if ($db==FALSE) die("errore nella connessione");
mysql_select_db($db_name,$db) or die("errore nella selezione del database");
$query="Insert into `articoli` ( `id` , `descrizione` ) VALUES ( NULL ,
'$descrizione');";
$result=mysql_query($query,$db);
if ($result==FALSE) die("Errore inserimento articolo,<br> la seguente query è
sbagliata: $query");
echo "Articolo inserito con successo: ".$descrizione;
mysql_close($db);
?>
```

Ecco fatto, ora posizioniamoci sulla pagina **inserisci_descrizione.html** ed inseriamo la descrizione del nuovo articolo che vogliamo inserire, poi clicchiamo su "Inserisci", i dati verranno inviato alla pagina **inserisci_database.php** che ci restituirà la conferma o l'errore di inserimento!