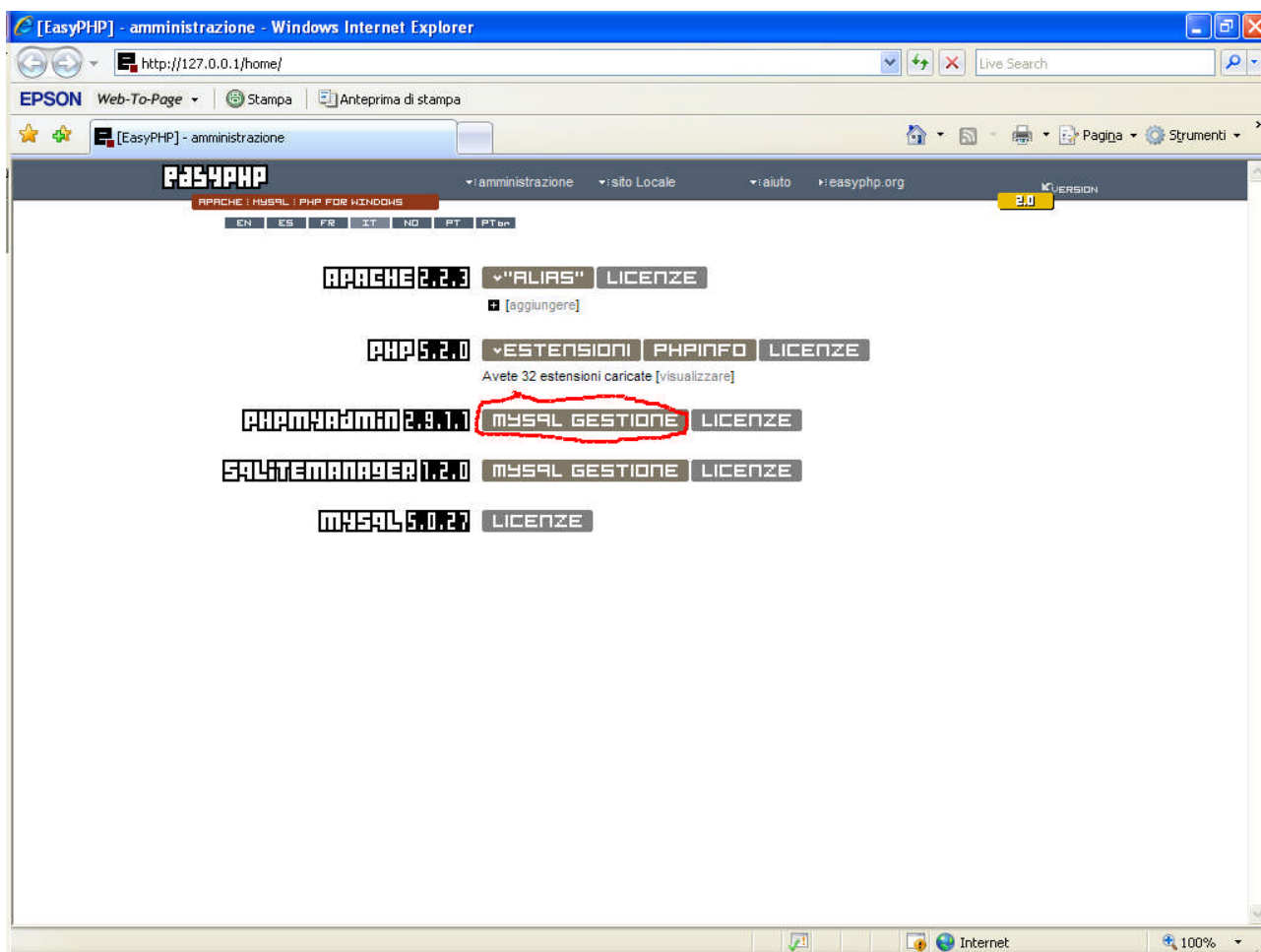


PHP E MYSQL

CREAZIONE DI UN NUOVO DATABASE DAL PHPMYADMIN

Il metodo migliore per creare nuovi DB è quello di usare l'interfaccia del PHPMYADMIN (che viene installata automaticamente con il pacchetto "easy php": www.easyphp.org)

Dalla HomePage che si apre lanciando easyphp selezionare la gestione di phpmyadmin:



Seguiremo ora lo sviluppo di un esempio pratico: la creazione di un DB per la gestione di un'ipotetica area "NEWS" all'interno di un sito.

Per prima cosa ci serve ragionare sui campi di cui abbiamo bisogno per il nostro Database: nel nostro caso ci serve una tabella contenente i seguenti campi:

- id: chiave primaria che identifica in maniera univoca i nostri record
- titolo
- abstract
- testo
- data

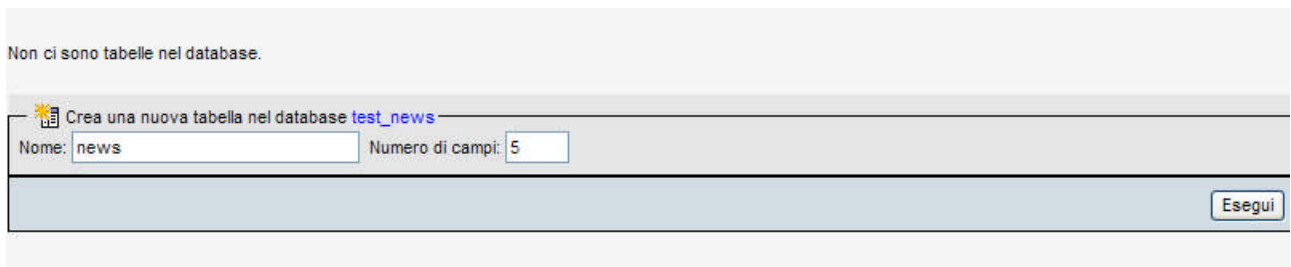
Per prima cosa creiamo un nuovo Database che chiamiamo “test_news”:



Nota: selezionare come tipo di caratteri “latin1_swedish_ci” per gestire al meglio caratteri accented

Dopo aver immesso i dati, cliccare sul pulsante “crea”.

A questo punto possiamo creare la tabella (che chiameremo “news”) del nostro DB e indicare quanti campi ci servono (5 nel nostro esempio):



Dopo aver cliccato su “esegui” inseriamo i nomi e le proprietà dei nostri campi:

Server: localhost Database: test_news Tabella: news

Campo	Tipo	Lunghezza/Set	Collation	Attributi	Null	Predefinito	Extra			
id	INT				not null		auto_increment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
titolo	TEXT				not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
abstract	TEXT				not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
testo	LONGTEXT				not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
data	DATE				not null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Commenti sulla tabella:

Motore di Memorizzazione: MyISAM Collation:

Salva Oppure Aggiungi 1 campo(i) Esegui

Nel nostro esempio abbiamo impostato i campi nel seguente modo:

- id: tipo INT (numero intero); EXTRA: auto_increment (ad ogni nuovo record viene assegnato un nuovo ID incrementale); CHIAVE PRIMARIA
- titolo: tipo TEXT
- abstract: tipo TEXT
- testo: tipo LONGTEXT
- data: tipo DATE




































Cliccare su "SALVA": apparirà una schermata con il riassunto della tabella creata:







Tabella news è stato creato.

```

query SQL:
CREATE TABLE `news` (
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `titolo` TEXT NOT NULL ,
  `abstract` TEXT NOT NULL ,
  `testo` LONGTEXT NOT NULL ,
  `data` DATE NOT NULL
) ENGINE = MYISAM ;



```

	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
<input type="checkbox"/>	id	int(11)			No		auto_increment	      
<input type="checkbox"/>	titolo	text	latin1_swedish_ci		No			      
<input type="checkbox"/>	abstract	text	latin1_swedish_ci		No			      
<input type="checkbox"/>	testo	longtext	latin1_swedish_ci		No			      
<input type="checkbox"/>	data	date			No			      

Seleziona tutti / Deseleziona tutti Se selezionati:      

[Visualizza per stampa](#) [Proponi la struttura della tabella](#)

1 campo(i) Alla fine della tabella All'inizio della tabella Dopo id

Indici					Spazio utilizzato		Statistiche righe	
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore
PRIMARY	PRIMARY	0	 	id	Dati	0 B		
Crea un indice su 1 columns <input type="button" value="Esegui"/>					Indice	0 B		
					Totale	0 B		

Ovviamente è possibile modificare/aggiungere/cancellare i campi anche successivamente.

A questo punto il nostro DB è pronto per essere utilizzato.

INTERFACCIAMENTO AL DATABASE DA PHP

Per prima cosa è utile preparare un file php di connessione al nostro database, che potremo utilizzare nel nostro sito ogni volta che abbiamo bisogno di utilizzare il DB.

Prepariamo un file che chiamiamo "connect.php" in cui inseriamo il seguente codice:

FILE "connect.php":

```

<?php
//definisco i parametri della connessione: nomeserver,username,password e nome db
$server="localhost";
$username="root";
$password="";

```



```
$nomedb="test_news";  
  
//mi connetto al server mysql  
$db=mysql_connect('localhost', 'root', '');  
if ($db == FALSE) {  
die ("Errore nella connessione.");  
}  
  
//mi connetto al database specificato nella variabile $nomedb  
mysql_select_db($nomedb, $db)  
or die ("Errore nella selezione del database");  
?>
```

```
<?php include('connect.php');?>
```

INSERIMENTO RECORD PRENDENDO I DATI DA UN FORM

Nella stragrande maggioranza dei casi l'aggiornamento del nostro DB sarà effettuato prendendo i dati da un form di inserimento. La prima cosa da fare è quindi di creare un semplice form in html che richiede l'inserimento dei campi relativi al nostro DB: creiamo un file html che chiamiamo "form_news.htm"

FILE: form_news.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head> <title>Inserisci News</title> </head>  
<body bgcolor=white>  
<div align="center">INSERIMENTO NUOVA NEWS</div>  
<form method=post action="inserisci_news.php">  
Giorno: <input type="text" name="giorno" size=2>  
Mese: <input type="text" name="mese" size=2>  
Anno: <input type="text" name="anno" size=4>  
<br><br>  
Titolo:<br>  
<input type="text" name="titolo" size=50><br><br>  
Abstract: <br>  
<textarea cols="30" rows="4" name="abstract"></textarea><br><br>  
Testo: <br>  
<textarea cols="30" rows="4" name="testo"></textarea>  
</form>  
</body>  
</html>
```

Da notare nell'apertura del form che è stato specificato come method "POST" verso la pagina "inserisci_news.php". La sostanziale differenza tra il metodo "POST" e quello "GET" è che con il metodo "POST" i dati vengono inviati in maniera invisibile da una pagina all'altra (e vanno recuperati nella pagina di destinazione tramite il comando "\$_POST". Utilizzando il metodo GET i dati vengono passati alla pagina di destinazione "accodati" all'indirizzo della pagina, e vanno recuperati tramite il comando "\$_GET".

Ora creiamo il file "inserisci_news.php" contenente il codice PHP per l'inserimento dei dati nel nostro DB:

FILE “inserisci_news.php”

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head> <title>INSERIMENTO NEWS</title></head>
<body>

<?php
// CONNESSIONE AL DATABASE
include('connect.php');
// RECUPERA DATI DAL FORM
$giorno = $_POST['giorno'];
$mese = $_POST['mese'];
$anno = $_POST['anno'];
// CREO DATA NEL FORMATO ANNO/MESE/GIORNO
$data= $anno."-".$mese."-".$giorno;
// RECUPERA DATI DAL FORM
$title = $_POST['titolo'];
$title = addslashes(stripslashes($title));
$abstract = $_POST['abstract'];
$abstract = addslashes(stripslashes($abstract));
$testo = $_POST['testo'];
$testo = addslashes(stripslashes($testo));

// INSERISCI NUOVO ARTICOLI IN DATABASE
$query = "INSERT INTO `news` ( `id` , `titolo` , `abstract` , `testo` , `data` )
VALUES (NULL , '$title', '$abstract', '$testo', '$data')";

if (@mysql_query($query)) {
echo ("<div align=center class=testo> <b>Record inserito con successo.</b></div>");
} else {
echo ("<div align=center class=testo> <b>Errore di inserimento record: ". mysql_error()
."</b></div>");
}
?>
</body>
</html>
```

Analizziamo ora il codice PHP per capire i vari passaggi.

Per prima cosa serve connettersi al Database, richiamando il file “connect.php” che contiene i parametri x la connessione al DB “test_news”:

```
include('connect.php');
```



Poi bisogna recuperare i dati inseriti e passati dal form. Il comando (avendo utilizzato il metodo “POST” del form) è nella forma:

```
$nomevariabile= $_POST['nome_campo_form'];
```

Per prima cosa vengono recuperati i dati per la data.

Le date nei DB mysql vengono memorizzate nel formato “Anno/Mese/Giorno”

Per prima cosa recuperiamo i dati (che abbiamo richiesto in campi separati nel form, per evitare errori nel formato della data):

```
$giorno = $_POST['giorno'];
```

```
$mese = $_POST['mese'];
```

```
$anno = $_POST['anno'];
```

Poi, andiamo a creare la data nel formato corretto per il DB, creando la nuova variabile \$data (composta dalla variabile \$anno + il carattere “-“ + \$mese...etc...):

```
$data= $anno."-".$mese."-".$giorno;
```

Fatto questo andiamo a recuperare gli altri campi di testo. Per questi campi dobbiamo prevedere che l’utente può inserire delle parole con apostrofi o virgolette. Ciò rischierebbe di confondere questi caratteri con la virgoletta di chiusura nell’inserimento del record. Per evitare ciò è necessario specificare che le eventuali virgolette (o apostrofi) inseriti sono caratteri da considerare tali e quali.

Ciò viene fatto aggiungendo prima di ogni virgoletta inserita un “back slash” (\).

Per far questo si usa il seguente comando:

```
$variabile = addslashes(stripslashes($variabile));
```

Quindi per ogni campo del nostro form, recuperiamo i dati e aggiungiamo i back slash:

```
$titolo = $_POST['titolo'];
```

```
$titolo = addslashes(stripslashes($titolo));
```

```
$abstract=$_POST['abstract'];
```

```
$abstract = addslashes(stripslashes($abstract));
```

```
$testo=$_POST['testo'];
```

```
$testo = addslashes(stripslashes($testo));
```

A questo punto abbiamo tutti i nostri dati e non ci rimane che inserirli nel nostro DB.

Per farlo si utilizza una “query” SQL di inserimento, che ha la seguente sintassi:

```
INSERT INTO `nome_tabella` ( `campo1` , `campo2` , ....., `campoN` )
```

```
VALUES ( `valore1` , `valore2` , ....., `valoreN` )
```

Nel nostro esempio il primo campo (“id”) deve essere un numero che viene incrementato automaticamente. Poiché abbiamo specificato nelle caratteristiche del DB come proprietà EXTRA “AUTO INCREMENT” (e lo abbiamo



indicato come campo numerico e chiave primaria), basterà nella query dargli valore “NULL” e in automatico verrà inserito un numero incrementale.

In php quindi creiamo una variabile \$query contenente la QUERY SQL per il nostro inserimento:

```
$query = "INSERT INTO `news` (`id`, `titolo`, `abstract`, `testo`, `data`)
VALUES (NULL, '$titolo', '$abstract', '$testo', '$data)";
```

Poi, diciamo di eseguire la query contenuta nella variabile \$query e di dare messaggio di conferma se la query viene eseguita correttamente o di errore in caso contrario:

```
if (@mysql_query($query)) {
echo "<div align=center class=testo> <b>Record inserito con successo.</b></div>");
} else {
echo "<div align=center class=testo> <b>Errore di inserimento record: ". mysql_error() ."</b></div>");
}
```

MODIFICA/CANCELLAZIONE RECORD

La prima pagina da preparare è una pagina contenente l’elenco dei record esistenti nel DB per scegliere quello da modificare/cancellare:

File: modifica.php

```
<?php
// CONNESSIONE AL DATABASE
include('connect.php');
// CREAZIONE QUERY SQL
$rs = mysql_query("SELECT * FROM news ORDER BY id");
// CONTROLLA QUANTI RECORD SODDISFANO LA QUERY E CREA UN CICLO IN CUI LEGGE I CAMPI DAI
RECORD
$nr = mysql_num_rows($rs);
if ($nr != 0){
for($x = 0; $x < $nr; $x++){
$row = mysql_fetch_assoc($rs);
$data=$row['data'];
$titolo=$row['titolo'];
$id=$row['id'];

// CREA CODICE HTML
```



```

?>
<table width=80% border=1><tr><td>data</td><td>titolo</td></tr>
<tr><td><?php echo $data;?></td><td><?php echo $titolo;?></td><td><a
href="modifica2.php?id=<?php echo $id;?>">Modifica</a></td><td><a href="cancella.php?id=<?php
echo $id;?>">Cancella</a></td></tr>
</table>
<?php
// PASSA A RECORD SUCCESSIVO
}
}
?>

```

Con il codice mostrato sopra viene creato un elenco di tutti i record presenti nel DB:

la query:

```
$rs = mysql_query("SELECT * FROM news ORDER BY id");
```

seleziona tutti I record (“*”) dalla tabella news, ordinandoli per id

```
$nr = mysql_num_rows($rs);
```

Con questa riga viene specificato di contare I record selezionati

Se la variabile nr risulta maggiore di zero, viene creato un ciclo che si ripete per “nr” volte, in cui vengono letti i dati dal DB:

```

if ($nr != 0){
    for($x = 0; $x < $nr; $x++){
        $row = mysql_fetch_assoc($rs);
        $data=$row['data'];
        $titolo=$row['titolo'];
        $id=$row['id'];
    }
}

```

Per ultima cosa viene creata una tabella html, in cui si scrivono i dati recuperati e si crea il link per la modifica o cancellazione del DB, passando alla pagina linkata la variabile “id”:

```
<a href="modifica2.php?id=<?php echo $id;?>">Modifica</a>
```

Vediamo ora la pagina per la MODIFICA del record selezionato:

File: modifica2.php

```

<?php
// CONNESSIONE AL DATABASE
include('connect.php');
$id=$_GET['id']; // RECUPERA VARIABILE ID DA PAGINA PRECEDENTE
$rs = mysql_query("SELECT * FROM news where id='$id'"); // QUERY SQL

```

```
$row = mysql_fetch_assoc($rs); // DEFINISCE VARIABILE $row
?>

// CREA CODICE HTML

<form method=post action="modifica3.php">

Data: <input type="text" name="data" size=10 value="<?php echo $row['data']?>">
<br><br>
Titolo:<br>
<input type="text" name="titolo" size=50 value="<?php echo $row['titolo']?>"><br><br>

Abstract: <br>
<textarea cols="30" rows="4" name="abstract" ><?php echo
$row['abstract']?></textarea><br><br>
Testo: <br>
<textarea cols="30" rows="4" name="testo"><?php echo $row['testo']?></textarea><br><br>
<input type="hidden" name="id" value="<?php echo $row['id']?>">
<input type="submit">
</form>
```

Il codice qui sopra è molto semplice e intuitivo: viene letto il record dalla tabella news, corrispondente al id scelto:

```
("SELECT * FROM news where id='$id');
```

dopodichè viene create un form html con all'interno I dati presi dal DB (notare la riga in cui viene passato il valore \$id tramite un campo nascosto del form)

L'ultima pagina è quella con la raccolta dei dati dal form, e l'aggiornanto del record:

File: modifica3.php

```
<?php
// CONNESSIONE AL DATABASE
include('connect.php');
// RECUPERA DATI DAL FORM
$id=$_POST['id'];
$data=$_POST['data'];
$titolo = $_POST['titolo'];
$titolo = addslashes(stripslashes($titolo)); // AGGIUNGE BACKSLASHES PRIMA DI
APOSTROFI/VIRGOLETTE
$abstract = $_POST['abstract'];
$abstract = addslashes(stripslashes($abstract)); // AGGIUNGE BACKSLASHES PRIMA DI
APOSTROFI/VIRGOLETTE
$testo = $_POST['testo'];
$testo = addslashes(stripslashes($testo)); // AGGIUNGE BACKSLASHES PRIMA DI
APOSTROFI/VIRGOLETTE
```

```
// AGGIORNA RECORD IN DATABASE
$query = "UPDATE news set `data`='$data' , `titolo`='$titolo' , `abstract`='$abstract' ,
`testo`='$testo' WHERE id='$id' " ;
if (@mysql_query($query)) {
echo ("<div align=center class=testo> <b>Record modificato con successo.</b></div>");
} else {
echo ("<div align=center class=testo> <b>Errore di inserimento record: ". mysql_error()
."</b></div>");
}
?>
```

Anche questo codice risulta semplice. Da notare:

la rielaborazione delle stringhe per la gestione di virgolette e apostrofi:

```
$testo = $_POST['testo'];
$testo = addslashes(stripslashes($testo));
```

La sintassi della query SQL per fare l'UPDATE del record:

```
$query = "UPDATE news set `data`='$data' , `titolo`='$titolo' , `abstract`='$abstract' , `testo`='$testo'
WHERE id='$id' " ;
```

L'ultima pagina che rimane è quella per la cancellazione di un record:

File: cancella.php

```
<?php
// CONNESSIONE AL DATABASE
include('connect.php');
$id=$_GET['id'];
$sql = "DELETE FROM news WHERE id=$id";
if (@mysql_query($sql)) {
echo ("<div align=center class=testo> <b>Record eliminato con successo.</b></div>");
} else {
echo ("<div align=center class=testo> <b>Errore: ". mysql_error() ."</b></div>");
}
?>
```