

# PHP - MYSQL

Ing. Bruno Di Caprio

# Mysql

- MySQL è un DBMS, Data Base Management System
- Software per la gestione di basi di dati
- Buone prestazioni su applicazioni di medio livello
- Popolarità indiscussa nonostante la mancanza di diverse caratteristiche avanzate (transazioni, stored procedures, etc.).

# Caratteristiche

- Scritto in C e C+
- Supporta diverse piattaforme: Windows \*, Linux 2.\*, FreeBSD, AIX, Solaris, SunOS,...
- Multi-threaded
- API: C, C++, Java, Perl, PHP, Python, Tcl,...

# Caratteristiche

- B-tree e tabelle temporanee
- Costrutto client/server
- Dimensione massima tabella: dipende dal OS/hardware (Linux intel 32: 2GB)
- Tipi di dato: reali, decimali (1-8), stringhe (0-255), binari , tempo,...

# Metodi di Accesso

Librerie native:

- Pro: veloci, poche dipendenze, portabilità
- Contro: non esiste uno standard
- ODBC:
  - Pro: standardizzato, non-relational data
  - Contro: più lento, necessità driver/provider
- GDA (Data Access Gnome):
  - Pro: tutti quelli di ODBC, scalabilità
  - Contro: in fase di sviluppo

# Gestione Database Mysql

- Accesso tramite client shell
  - Mysql -u root -p
  - Utilizzo di SQLite o PhpMyadmin
- Lista dei database presenti
  - Show db;
- Creazione di un database
  - Create db <nomedatabase>

# Gestione delle Tabelle

- Creazione delle tabelle
  - CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name [(create\_definition,...)] [table\_options]
  - Definizione delle colonne:  
col\_name type [NOT NULL | NULL] [DEFAULT default\_value] [AUTO\_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY] [COMMENT 'string']

# Gestione delle Tabelle

- Esempio

```
CREATE TABLE libri (  
  id INT(10) NOT NULL AUTO_INCREMENT,  
  autore VARCHAR(255) NOT NULL,  
  titolo VARCHAR(255) NOT NULL,  
  costo FLOAT(10,2) NOT NULL,  
  PRIMARY KEY (id)  
) TYPE=MyISAM
```



# Gestione delle Tabelle

- Rinominare una o più tabelle  
`RENAME TABLE tbl1 TO tmp_tbl1,  
tbl2 TO tbl1,  
tmp_tbl1 TO tbl2;`
- Eliminare una o più tabelle  
`DROP TABLE tbl1, tbl2, tbl3`

# Modifica di una Tabella Esistente

```
ALTER [IGNORE] TABLE tbl_name(alter_specification)
```

alter\_specification:

```
ADD [COLUMN] column_definition [FIRST | AFTER col_name ]
```

```
| ADD [COLUMN] (column_definition,...)
```

```
| ADD INDEX [index_name] (index_col_name,...)
```

```
| ADD PRIMARY KEY (index_col_name,...)
```

```
| ADD UNIQUE [index_name] (index_col_name,...)
```

```
| CHANGE [COLUMN] old_col_name column_definition
```

```
[FIRST|AFTER col_name]
```

```
| DROP [COLUMN] col_name
```

```
| DROP PRIMARY KEY
```

```
| DROP INDEX index_name
```

```
| RENAME [TO] new_tbl_name
```

```
| ORDER BY col_name
```

# Inserimento dati in tabella

Supponendo di avere una tabella chiamata rubrica  
(Cognome, Nome, Telefono)

```
insert into rubrica values ('Di Caprio', 'Bruno',  
    '349xxxxxxx')
```

# PHP & Mysql

- L'accesso ad un database MySQL avviene mediante autenticazione; questo vuol dire che, prima di poter effettuare qualsiasi operazione su di esso, dobbiamo disporre dei privilegi necessari. In particolare, le informazioni di cui abbiamo bisogno sono: il nome dell'host su cui è in esecuzione il server MySQL, il nome del nostro database, il nome utente che ci è stato assegnato e la relativa password.

# Utilizzo PHP per le interrogazioni

Definiamo quindi in php le variabili di interesse per utilizzarle nella connessione con Mysql

```
// Il nome dell'host (hostname) su cui si trova MySQL
```

```
$dbhost = "localhost";
```

```
// Il nome del nostro database
```

```
$dbname = "esempio";
```

```
// Il nostro nome utente (username)
```

```
$dbuser = "root";
```

```
// La nostra password
```

```
$dbpass = "root";
```

# Utilizzo PHP per le interrogazioni

- La prima funzione che utilizziamo è `mysql_connect()`, che ci servirà per instaurare la connessione con il server MySQL.

```
// Funzione mysql_connect()  
$conn = mysql_connect($dbhost,$dbuser,$dbpass)  
or die("Impossibile collegarsi al server MySQL.");
```

*Nell'impossibilità di collegarsi al server (valore nullo ritornato dalla funzione) viene invocata die, altrimenti in \$conn vi è l'identificativo della connessione da utilizzare nelle interrogazioni*

# Utilizzo PHP per le interrogazioni

- Selezione del Database su cui operare

```
// Funzione mysql_select_db()  
mysql_select_db($dbname,$conn)  
or die("Impossibile selezionare il database $dbname");
```

# PHP & Mysql

- Qualsiasi interazione con la base dati avviene tramite il linguaggio sql e quindi con una query quindi creiamo una variabile \$sql a cui assegnamo la query

```
$res = mysql_query($sql,$conn)  
or die( "Errore: " . mysql_error() );
```



# PHP & Mysql

- Estrazione del risultato con `mysql_fetch_row()`  
// Estrazione del risultato  
`$info = mysql_fetch_row($res);`  
`echo "Il mio numero di telefono è " . $info[0];`
- Chiusura della connessione  
// Funzione `mysql_close()`  
`mysql_close($conn);`

# Ulteriori Strumenti per l'interazione utente

- Variabili di sessione
  - La sessione è il modo più semplice per riconoscere un utente da quando si collega al server a quando interrompe il collegamento.
  - Per essere più chiari immaginiamo di fornire una password per un'area protetta e poi di navigare. Per ogni nuova pagina dovremo ridigitare la password, ma non è così.

# Variabili di Sessione

- Il server apre una sessione e gli assegna un nome univoco. Analizza l'utente, raccoglie i dati che vogliamo e li salva in un file che porta lo stesso nome della sessione.
- Nello stesso tempo deposita un cookie nella macchina dell'utente.

# Variabili di Sessione

Ci sono alcuni semplici punti da seguire quando si crea una sessione PHP:

- `session_start()`; Funzione che apre una sessione.
- `session_destroy()`; Funzione che distrugge una sessione.

# Variabili di Sessione

- `session_register("...");` Salva una variabile nella sessione. (Si scrive solo il nome senza \$. Il valore della variabile va inserito nel codice sottostante, in pratica la macchina lo salva da sola.)
- `session_unset();` Distrugge le variabile nella sessione.

# Variabili di Sessione

- `$PHPSESSID` Il nome della variabile globale che contiene il codice univoco di 32 caratteri che distingue la sessione. (Se scriviamo `echo $PHPSESSID;` scriviamo 32 caratteri a video se la sessione è attiva.)
- Altrimenti è possibile inserire la funzione `session_id()`;  
esempio:  
`$PHPSESSID=session_id();`

# Esempio di Utilizzo

```
<?  
session_start();//Apriamo la sessione  
session_register("nomevariabile");//Salviamo una variabile nella sessione  
....//Calcoli o funzioni con la sessione  
....  
session_register("nomevariabile2");//Salviamo una variabile nella sessione  
session_register("....");//Possiamo salvare quante variabili vogliamo  
  
$nomevariabile=.....;//Definiamo la variabile  
$nomevariabile2=.....;//Definiamo la variabile2  
....//Calcoli o funzioni con la sessione  
....  
?>
```

# Cookie

I Cookie sono il metodo più comune per la memorizzazione dello stato nel web.

```
<?php
setcookie ("test_cookie","niente di particolare",time()+43200,"/");
# cookie.php
echo "<HTML>";
echo "<BODY>";
if (isset($test_cookie)){
echo "Ciao cookie, i tuoi contenuti sono: $test_cookie";
} else {
echo "Non ho trovato alcun cookie con il nome test_cookie";
}
echo "</BODY>";
echo "</HTML>";
```



# Esempio riepilogativo

```
<?php
$conn = mysql_connect('www.myserver.it', 'giovanni', 'segretissima');
mysql_select_db('cinema');

$cur = mysql_query ("
SELECT film.anno, film.titolo, recita.personaggio
FROM persone, film, recita
WHERE Nome = '$Nome' AND Cognome = '$Cognome'
      AND persone.codice = recita.attore
      AND recita.film = film.codice
");
```

# Esempio riepilogativo

```
echo "  
<table border=  
<tr><th colspan='3'>Carriera di $Nome $Cognome</tr>  
<tr><th>Anno<th>Titolo<th>Personaggio</tr>  
";  
while ($riga = mysql_fetch_row($cur)){  
echo "<tr><td>$riga[0]<td>$riga[1]<td>$riga[2]";  
}  
echo "</table>";  
  
mysql_close($conn);
```

# PHP & Mysql

- Wamp
- EasyPHP
- Apache + Mysql + mod php per Apache
- PhpMyAdmin