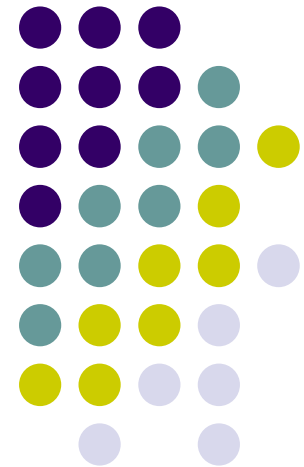


UD19: Connessione a database **MySQL**

Fabio Cantaro





Introduzione

- Operazioni sui database, fasi:
 1. Apertura connessione
 2. Selezione del db
 3. Impostazione/Esecuzione query
 4. Lettura risultato query (**Fetching**)
 5. Chiusura connessione



Apertura/chiusura CONNESSIONE

- Tramite la funzione **MySQL_connect()**
- È l'inizio di ogni script di gestione di database tramite PHP
- Sintassi Apertura Connessione:

```
$connessione=mysql_connect($hostname, $username, $password);
```

\$connessione conterrà il puntatore, ritornato da **mysql_connect**, che ci permette di identificare la connessione => da usare per la selezione del DB

Chiusura Connessione:

```
mysql_close($connessione)
```

Controllo Connessione



- Occorre verificare se la connessione è andata a buon fine:

```
$hostname="localhost";
$username="root";
$password="";
$database="esamestato2006";
$connessione=mysql_connect($hostname, $username, $password);
if(! $connessione)
{
    $messaggio="Siamo Spiacenti, ci sono Problemi di connessione
                al Database!<br>Riprovare più tardi. ";
    echo $messaggio);
    return;
    //exit("Problemi di connessione al Database!");
}
else
{
    //echo("<h1>Connessione ok</h1>");    evito di fare scrivere cose ovvie
}
}
```

Selezione del db

Fabio Cantaro



- Tramite la funzione `MYSQL_SELECT_DB()`
- Ritorna:
 - **True** in caso di successo
 - **False** se errore (esempio: “nome db errato”, oppure “server non attivo”)
- Sintassi: **`mysql_select_db($database, $connessione)`**

esempio

```
if(!mysql_select_db($database, $connessione) )
{
    //exit("Errore nella scelta del DB");
    $messaggio="Siamo Spiacenti, Problemi nella selezione del Database!";
    $messaggio.="<br>Riprovare più tardi. ";
    echo $messaggio;
    return;
}
//Else ok
```

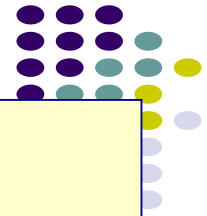


Esecuzione query

Sintassi:

`mysql_query($Query)`

- Fa eseguire la query `$Query` (stringa con la query espressa in SQL) al server
- Ritorna un array associativo contenente il risultato, ossia il **RECORD SET** ossia i record risultato della query
- È possibile verificare l'esecuzione tramite la clausola **OR DIE**
- OSS: per non fare comparire gli errori di MySQL => fare precedere ogni funzione dalla **@**:
ad esempio: **@mysql_connect(...)**



Esempio

```
<?
//connessione ad un database di nome prova2

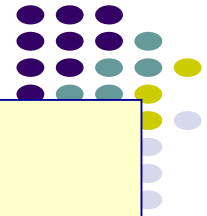
$db = mysql_connect("localhost","pippo","pippo")
    or die ("Non riesco a creare la connessione");
mysql_select_db("prova2") or die ("Non trovo il DB");
$sql = "SELECT utente,password,conta_pres,ID FROM users
        WHERE conta_pres <> 0";
$ris = mysql_query($sql) or die ("Query fallita!");

VEDREMO DOPO:
while ($riga = mysql_fetch_array($ris)) {
    echo $riga["utente"] . " , ";
    echo $riga["password"] . " , ";
    echo $riga["conta_pres"] . "<BR>";
}
mysql_close();
?>
```



Funzioni di FETCHING

- La `mysql_query($sql)` ritorna il recordset che deve essere processato UNA RIGA ALLA VOLTA => tramite le funzioni di fetching:
 1. **`MySQL_fetch_row($recSet)`:**
 - Prende il Record Set come parametro
 - Ritorna una riga alla volta
 2. **`MySQL_fetch_array($recSet)`:**
 - Prende il Record Set come parametro
 - Ritorna un Array Associativo in cui i nomi fungono da chiavi



Esempio

```
<?
//connessione ad un database di nome prova2

$db = mysql_connect("localhost","pippo","pippo")
    or die ("Non riesco a creare la connessione");
mysql_select_db("prova2") or die ("Non trovo il DB");
$sql = "SELECT utente,password,conta_pres,ID FROM users
        WHERE conta_pres <> 0";
$ris = mysql_query($sql) or die ("Query fallita!");

while ($riga = mysql_fetch_array($ris))
{
    echo $riga["utente"] . " , ";
    echo $riga["password"] . " , ";
    echo $riga["conta_pres"] . "<BR>";
}
mysql_close();
?>
```



Tipi di query

- Per la select che ritorna tante righe occorre un while (esempio precedente).
- Insert: la query viene eseguita => occorre solo verificare che sia andata a buon fine
- Similmente per Update e Delete



Esempio 1: Insert

```
<?  
//inserimento record in database prova2  
  
$db = mysql_connect("localhost","generico","child")  
      or die ("Non riesco a creare la connessione");  
  
mysql_select_db("prova2") or die ("Non trovo il DB");  
$sql = "INSERT INTO users (utente, password, ID, conta_pres)  
      VALUES ('utente n','abibo',NULL,0)";  
$ris = mysql_query($sql) or die ("Query fallita!");  
mysql_close();  
?>
```



Esempio 2: UPDATE

```
<?
//aggiornamento record database Prova2
$valore=1;
$db = mysql_connect("localhost","generico","child")
    or die ("Non riesco a creare la connessione");

mysql_select_db("prova2") or die ("Non trovo il DB");

$sql = "UPDATE users SET conta_pres='$valore' WHERE conta_pres=0 ";

$ris = mysql_query($sql) or die ("Query fallita!");
mysql_close();
?>
```



Esempio 3: DELETE

```
<?  
//cancellazione record database prova2  
$db = mysql_connect("localhost","generico","child")  
    or die ("Non riesco a creare la connessione");  
  
mysql_select_db("prova2") or die ("Non trovo il DB");  
  
$sql = "DELETE FROM users WHERE conta_pres=1";  
  
$ris = mysql_query($sql) or die ("Query fallita!");  
mysql_close();  
?>
```

Verificare i risultati di una query



- Una query corretta potrebbe non produrre alcun risultato => se la where non è soddisfatta!
- => occorre sapere n° righe ritornate dalla query => **N=MySQL_num_row(\$ris_query)**
- Ove **N** = numero righe ritornate

ESERCIZIO: esame Stato 2006

Fabio Cantaro



Una rete di scuole chiede che sia progettato e realizzato un database per l'organizzazione e la

gestione del portale di una community di apprendimento sul Web.

L'organizzazione della community prevede che:

- l'accesso sia consentito ai soli utenti registrati;
- gli utenti siano distribuiti in tre gruppi: *amministratore*, *docente*, *studente* tali che:
 - un amministratore abbia accesso a tutte le aree protette del portale;
 - un docente abbia accesso a tutte le aree protette tranne che all'area di amministrazione;
 - uno studente abbia accesso alla propria area protetta e non abbia accesso né all'area di amministrazione né all'area riservata al gruppo *docente*;

Implementare almeno le seguenti query:

n. 1: I docenti che hanno un account presso la community con la rispettiva collocazione geografica ed i moduli formativi scelti.

n. 2: I dati relativi agli studenti e ai documenti che essi hanno inviato in remoto sul portale della community mediante upload.

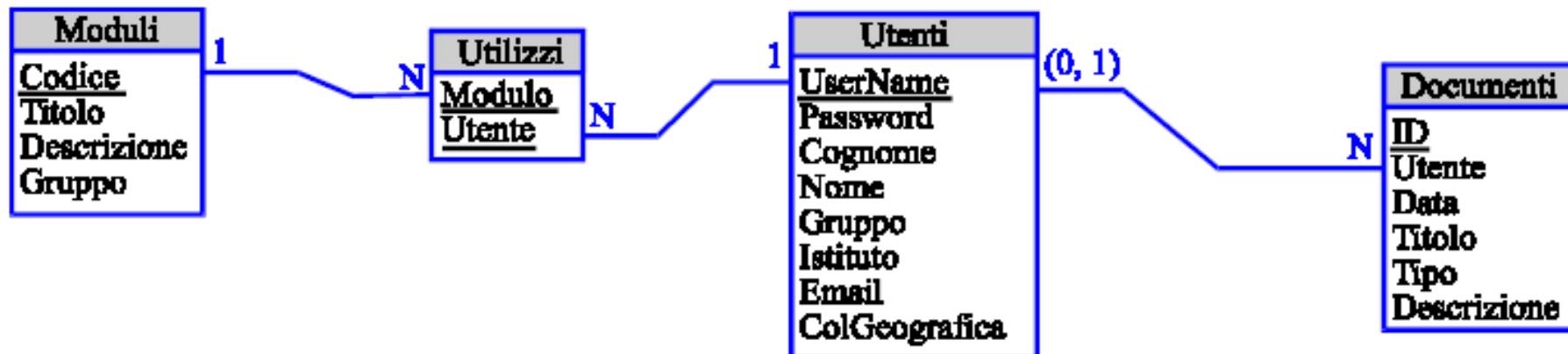
Scrivere, in php, il codice di almeno una delle seguenti pagine del portale:

- con accesso riservato all'amministratore, il report che trae i dati dalla query n. 1
- con accesso riservato agli studenti, il report che trae i dati dalla query n. 2
- registrazione di un nuovo utente con eventuale invio automatico dei dati registrati mediante

e-mail diretta all'utente appena registrato e ad un amministratore.



Schema E/R





Tabelle

```
CREATE TABLE Utenti
(
  UserName CHAR(15),
  Password CHAR(10) NOT NULL,
  Cognome CHAR(15) NOT NULL,
  Nome CHAR(15) NOT NULL,
  Gruppo CHAR(14) NOT NULL,
  Istituto CHAR(30) NOT NULL,
  Email CHAR(30) NOT NULL,
  ColGeografica CHAR(20) NOT NULL,
  CHECK (Gruppo IN ('amministratore',
'docente', 'studente') ),
  PRIMARY KEY (UserName)
)
```

```
CREATE TABLE Documenti
(
  ID INTEGER IDENTITY(1, 1),
  Utente CHAR(15),
  Data DATE NOT NULL,
  Titolo CHAR(30) NOT NULL,
  Tipo CHAR(8) NOT NULL,
  Descrizione CHAR(255),
  CHECK (Gruppo IN ('testo', 'audio', 'immagine',
'video') ),
  PRIMARY KEY (ID),
  FOREIGN KEY (Utente) REFERENCES
Utenti(UserName)
)
```

```
CREATE TABLE Moduli
(
  Codice INTEGER,
  Titolo CHAR(30) NOT NULL,
  Descrizione CHAR(255),
  Gruppo CHAR(8) NOT NULL,
  CHECK (Gruppo IN ('docente', 'studente') ),
  PRIMARY KEY (Codice)
)
```

```
CREATE TABLE Utilizzi
(
  Modulo INTEGER,
  Utente CHAR(15),
  PRIMARY KEY (Modulo, Utente),
  FOREIGN KEY (Modulo) REFERENCES
Moduli(Codice),
  FOREIGN KEY (Utente) REFERENCES
Utenti(UserName)
)
```



Query

Prima operazione

```
SELECT Utenti.Cognome, Utenti.Nome, Utenti.ColGeografica, Moduli.Titolo  
FROM Utenti, Utilizzi, Moduli  
WHERE Utenti.UserName = Utilizzi.Utente AND  
        Utilizzi.Modulo = Moduli.Codice AND  
        Utenti.Gruppo = 'docente'
```

Seconda operazione

```
SELECT Utenti.Cognome, Utenti.Nome, Documenti.Descrizione, Documenti.Tipo  
FROM Utenti, Documenti  
WHERE Utenti.UserName = Documenti.Utente AND  
        Utenti.Gruppo = 'studente'
```



PRIMA OPERAZIONE

DUE FILE:

- **logon.html**: form HTML in cui viene effettuato il login del possibile amministratore
- **Query1.php**:
 - qui, prima viene verificato se l'utente è un amministratore (prima query)
 - Poi, se la precedente è verificato, una query relativa ai docenti e alla loro collocazione geografica. Il risultato viene formattato in una tabella



Logon.html

```
<FORM ACTION="query1.php" METHOD="POST">
<FIELDSET><LEGEND><B>Report riservato all'amministratore </B></LEGEND>
User name <INPUT TYPE="TEXT" NAME="txtUsername" SIZE="15">
Password <INPUT TYPE="PASSWORD" NAME="txtPassword"
SIZE="10"><BR><BR>
<INPUT TYPE="SUBMIT" VALUE="Logon amministratore" NAME="cmdInvia">
</FIELDSET>
</FORM>
```

Query1.php

```
<?php
    $hostname="localhost";
    $username="root";
    $password="";
    $database="esamestato2006";

    //connessione
    $connessione=mysql_connect ($hostname, $username, $password);
    if(! $connessione)
    {
        $messaggio="Siamo Spiacenti, ci sono Problemi di connessione al Database!<br>Riprovare più tardi. ";
        funzione_messaggio($messaggio);
        return;
        //exit("Problemi di connessione al Database!");
    }
    else
    {
        //echo("<h1>Connessione ok</h1>");    evito di fare scrivere cose ovvie
    }

    if( ! mysql_select_db($database, $connessione) )
    {
        //exit("Errore nella scelta del DB");
        $messaggio="Siamo Spiacenti, ci sono Problemi nella selezione de l Database!";
        $messaggio.="<br>Riprovare più tardi. ";
        echo $messaggio;
        return;
    }
    else
    {
        //echo("Selezione Ok");    evito messaggi inutili
    }
}
```

CONTINUA PROSSIMA SLIDE

Query1.php (2)

```
//ricavo i parametri passati dal FORM
$username=$_POST["txtUsername"];
$password=$_POST["txtPassword"];

//formattazione query inserimento

//Impostazione della Query per verificare username e password dell'amministratore
$query="SELECT gruppo, username, password FROM utenti WHERE username=\"".$username.\" AND ";
$query.="password=\"".$password.\" ";
//eseguo la query
$res=mysql_query($query);
//prendo, se esiste, la riga che viene fuori:
$data=mysql_fetch_array($res);

//se la query non mi restituisce alcuna riga => l'utente non fa parte della t abella utenti => non è registrato!
if(!$data)
{
    echo "\nNon è un utente registrato";
    return;
}
else //se la query restituisce delle righe
{
    $gruppo=$data["gruppo"];
    if($gruppo=='amministratore')
    {
        //Essendo veramente l'amministratore => Impostazione della Query richiesta
        $query="SELECT utenti.cognome, utenti.nome, utenti.colgeografica, moduli.titolo ";
        $query.="FROM utenti, utilizzi, moduli WHERE utenti.username=utilizzi. utente ";
        $query.="AND utilizzi.modulo=moduli.codice AND utenti .gruppo='docente'";

        //esecuzione della query
        $res=mysql_query($query);
    }
}
```

CONTINUA PROSSIMA SLIDE

Query1.php (3)

```
//esecuzione della query
$Res=mysql_query($Query);
if(!$Res)
{
    $messaggio="Query fallita!";
    funzione_messaggio($messaggio);
}
else //la query è andata a buon fine
{
    //formatto la tabella che conterrà i valori estratti
    print("<table border=1 bgcolor=YELLOW>");
    //intestazione della tabella
    print("<tr>");
    print("<th>Cognome</th>"); print("<th>nome</th>"); print("<th>Collocazione geografica</th>");
    print("<th>Titolo Modulo</th>");print("</tr>");

    //con un while inizio a leggere il risultato della query e la metto nelle varie righe della tabella
    while ($data=mysql_fetch_object($Res)) //per ogni riga stampare:
    {
        print("<tr>");
        //inserisco i valori dei campi nella tabella
        print("<TD>".$data->cognome."</TD>");
        print("<TD>".$data->nome."</TD>");
        print("<TD>".$data->colgeografica."</TD>");
        print("<TD>".$data->titolo."</TD>");
        print("</tr>");
    }
    print("</table>");
}

} // fine if amministratore
else {echo "Non sei un Amministratore!!"; return;}

} // fine ELSE "la query restituisce delle righe"
mysql_close($connessione);
```

?>

Esempio 2: Registro dei voti (448)

Fabio Cantaro



- File:
 - Esempio2.html
 - Esempio2.php
 - Esempio2_1.php
 - Database: registro_classe.sql
- Mediate Esempio2.html si fa il login e la validazione (Esempio2.php) del login del docente (administrator)
- Se ok => i dati trasmessi assieme ai dati dello studente a Esempio2_1.php