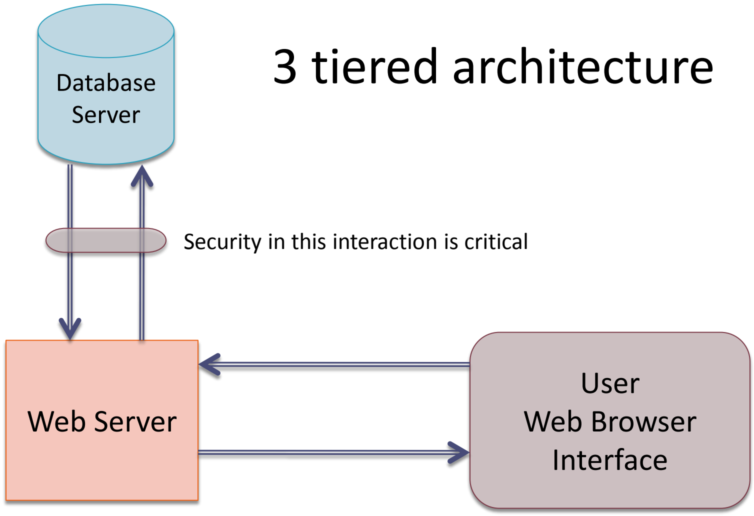
MySQL Database Access with PHP

Fall 2010

CSCI 2910 Server-Side Web Programming

Typical web application interaction

3 tiered architecture



Security in this interaction is critical

MySQL

Most popular open source DB. Free, high performance database engine.

- http://www.mysql.com/
- MySQL Enterprise available as paid annual subscription with additional tools, support, etc.

Part of common open-source LAMP Web deployment architecture—Linux, Apache, MySQL, PHP..

Architecture Components

MySQL Server

- Available for various contemporary server OS. (Requires C++ and support for multi-threading.)

Working with server—3 methods:

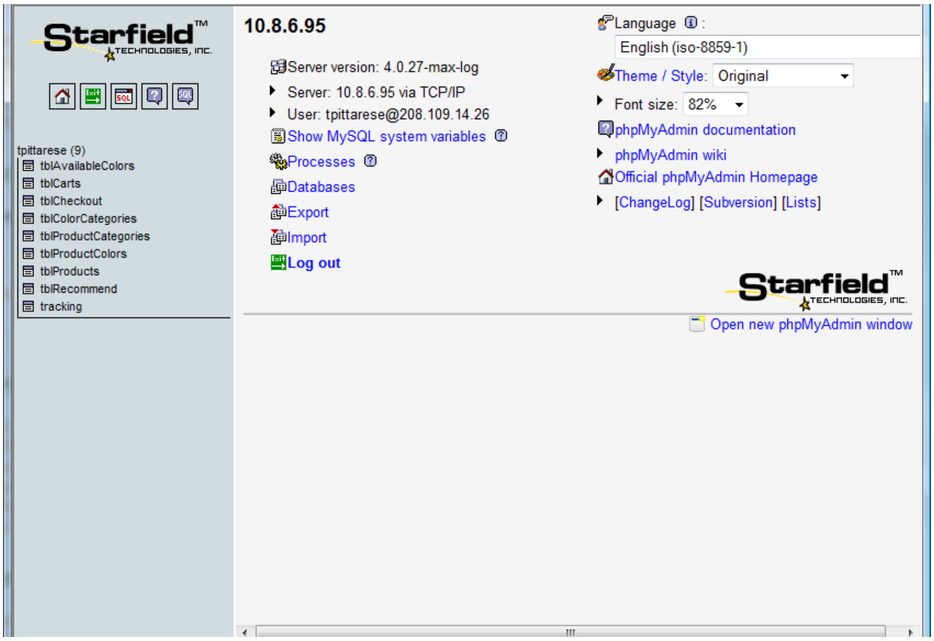
- **MySQL Monitor** executed from terminal window on server (command line tool, part of server install)
- **MySQL Monitor** executed from different server, connecting to target server. (Target server account must permit remote log in.)
- **MySQL GUI Tools**—MySQL Administrator, MySQL Query Browser, and MySQL Migration Toolkit.

Working with server—security issues

Many organizations (and Web hosting companies) disable remote access for security reasons. They only accept localhost logins.

Many Web hosts do not allow MySQL Monitor or other GUI tool access.

Packages like phpMyAdmin allow MySQL database management via a web-based GUI.



10.8.6.95

Server version: 4.0.27-max-log

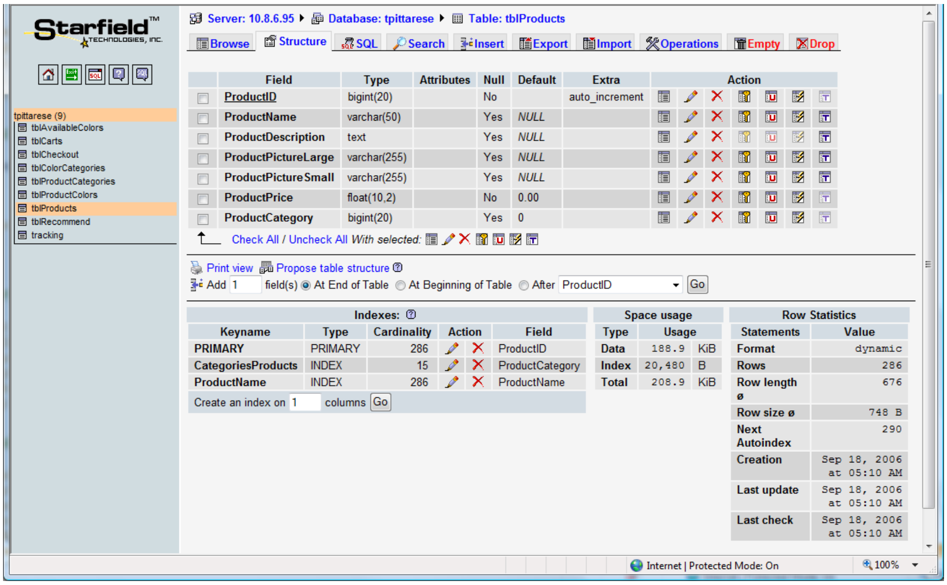
Server: 10.8.6.95 via TCP/IP

User: tpittarese@208.109.14.26

Language: English (iso-8859-1)

Theme / Style: Original

Font size: 82%



tpittarese (9)
- tblAvailableColors
- tblCarts
- tblCheckout
- tblColorCategories
- tblProductCategories
- tblProductColors
- tblProducts
- tblRecommend
- tracking

Server: 10.8.6.95 ▸ Database: tpittarese ▸ Table: tblProducts

	Field	Type	Attributes	Null	Default	Extra	Action
□	ProductID	bigint(20)		No		auto_increment	
□	ProductName	varchar(50)		Yes	*NULL*		
□	ProductDescription	text		Yes	*NULL*		
□	ProductPictureLarge	varchar(255)		Yes	*NULL*		
□	ProductPictureSmall	varchar(255)		Yes	*NULL*		
□	ProductPrice	float(10,2)		No	0.00		
□	ProductCategory	bigint(20)		Yes	0		

Check All / Uncheck All *With selected:*

Print view Propose table structure

Add 1 field(s) ● At End of Table ○ At Beginning of Table ○ After ProductID Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	286	✎ ✕	ProductID
CategoriesProducts	INDEX	15	✎ ✕	ProductCategory
ProductName	INDEX	286	✎ ✕	ProductName

Create an index on 1 columns Go

Space usage

Type	Usage	
Data	188.9	KiB
Index	20,480	B
Total	208.9	KiB

Row Statistics

Statements	Value
Format	dynamic
Rows	286
Row length ø	676
Row size ø	748 B
Next Autoindex	290
Creation	Sep 18, 2006 at 05:10 AM
Last update	Sep 18, 2006 at 05:10 AM
Last check	Sep 18, 2006 at 05:10 AM

Using MySQL monitor

To connect:

1. Use PuTTY to connect to Einstein terminal window
2. At Einstein prompt enter **mysql –u *abc123* -p** substituting your account name for abc123

   -u indicates the field that follows is your username

   -p indicates that you will supply a password
3. Enter your password when prompted. (Initial password is 12345.)
4. An optional –h parameter can be used to specify a remote host.

Working in MySQL monitor

To change your MySQL password:

```
set password = password("mynewpassword");
```

All statements end in a semicolon. If not supplied on enter, prompted to continue entering statement (or semicolon).

Up arrow and down arrow permit scrolling through previous commands.

If you've typed a portion of a command and wish to "abandon" it, type **\c** and press enter.

More about the monitor

You can enter individual queries/commands or create an SQL script in a text editor. (It must be **plain text**.)

- Save the file with an **.sql extension**.
- Use **#** at the beginning of the line for a comment.
- Upload the file to server via sFTP. Place in same directory as where MySQL is launched, typically *account* root (**not** web site root).
- To execute the script, enter **\. scriptname.sql** (slash, period, space, scriptname). (Don't forget the space!)

Working in MySQL monitor

See list of databases: `show databases;`

Select a database to use: `use databasename;`

Database for class is the same as your account name.

See list of tables in database: `show tables;`

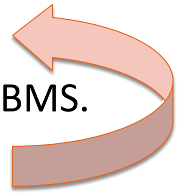
Show columns in a table: `show columns from tablename;` or `describe tablename;`

To exit the monitor: `exit` or `\q`

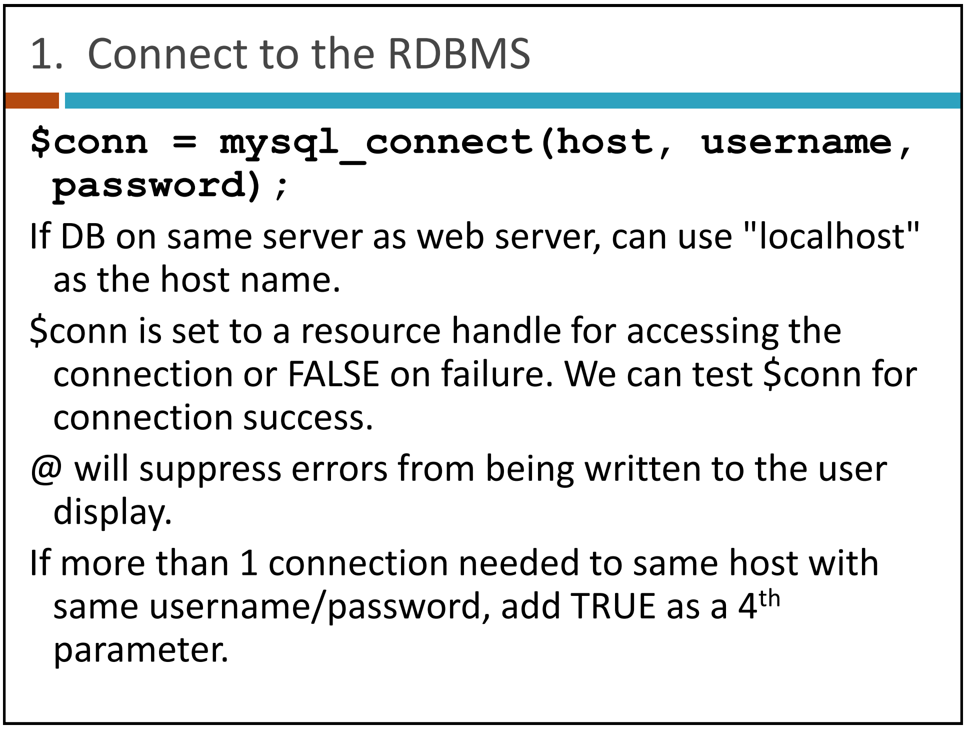
Connecting to MySQL from PHP

Typical steps in database transactions with PHP:

1. Connect to the RDBMS (Relational Database Management System).
2. Select a database to use.
3. Execute query.
4. Receive resultset from RDBMS.
5. Process the resultset.
6. Close the RDBMS connection.



Iterate as needed.



1. Connect to the RDBMS

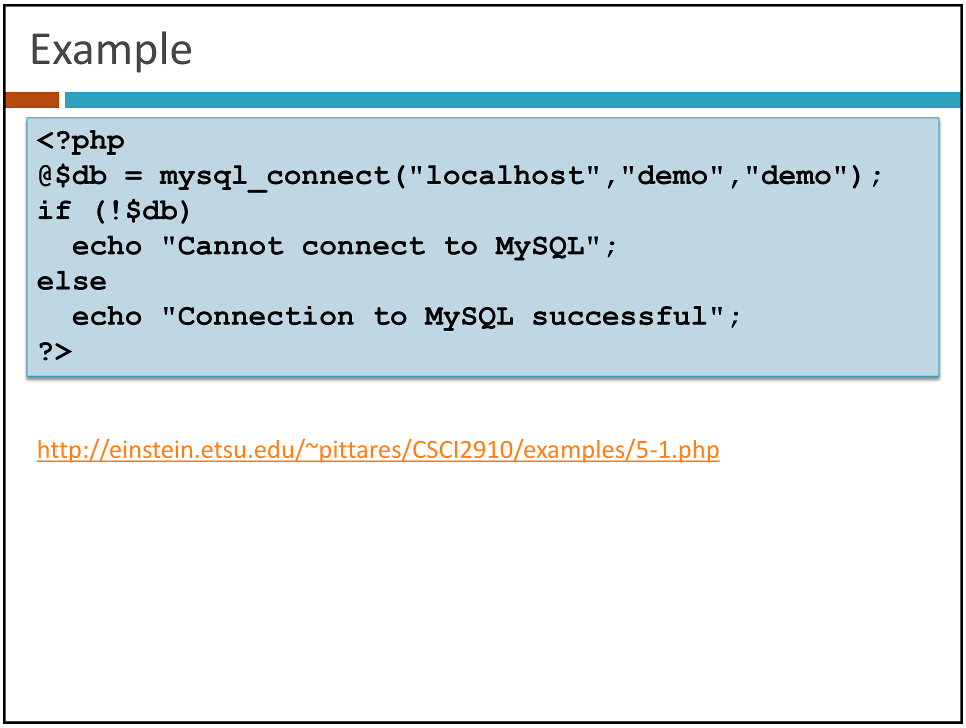
```
$conn = mysql_connect(host, username,
 password);
```

If DB on same server as web server, can use "localhost" as the host name.

$conn is set to a resource handle for accessing the connection or FALSE on failure. We can test $conn for connection success.

@ will suppress errors from being written to the user display.

If more than 1 connection needed to same host with same username/password, add TRUE as a 4th parameter.



Example

```
<?php
@$db = mysql_connect("localhost","demo","demo");
if (!$db)
  echo "Cannot connect to MySQL";
else
  echo "Connection to MySQL successful";
?>
```

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-1.php

die() or exit()

`die($string)` or **`exit($string)`**—terminate script execution writing $string as error message.

```
<?php
@$db = mysql_connect("localhost","demo",
  "demo");
if (!$db)
  die("Cannot connect to MySQL");
echo "Connection to MySQL successful";
?>
```

"or" based construction can be used based on PHP's short-circuit evaluation of logical operators.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-2.php

2. Select a database to use

If only one database to be opened:

```
mysql_select_db('dbname');
```

If more than one database to be opened, list the connection to be used with a particular database.

```
mysql_select_db('dbone', $db1);
mysql_select_db('dbtwo', $db2);
```

Most functions follow the pattern of having you specify the connection as the last parameter if more than 1 connection open.

Returns FALSE if unable to select DB.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-3.php

3. Execute Queries

```
$output = mysql_query($query);
$output = mysql_query($query,$db1);
```

The above allows the execution of *any* raw SQL (Select, Insert, Delete, Update, etc.)

Query string ($query) cannot end in a semicolon.

$output is a "complex" data type—a resource handle to the resultset for Select, etc., boolean for Insert, Update, etc.

6. Close the database connection

If only one (unnamed) connection open

```
mysql_close();
```

If more than one connection open

```
mysql_close($db1);
```

PHP documentation claims close is unnecessary but some report it does affect performance.

Example

```
//Assume form field checking/cleaning completed
$conn = @mysql_connect("localhost", "un", "pass");
if (!$conn)
  die("Unable to connect to DB.<br/>".mysql_error());
if (!@mysql_select_db("mydb"))
  die("Unable to open DB.<br/>".mysql_error());
$query = "insert into students(student_name) value
('$name')";
if (!@mysql_query($query))
  die("Unable to execute insert.<br/>".mysql_error());
echo "New data inserted<br/>";
@mysql_close();
```

Processing Select Resultset

`mysql_num_rows($resultset)` returns number of lines returned by a select statement.

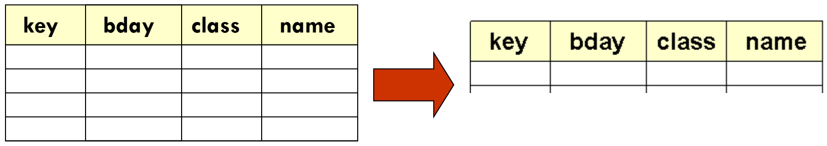
http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-4.php

Although this could be used to process resultset using a for loop, this is not typical.

Recordset visualization

Recordset returned as a complex data type. To display to user or otherwise incorporate into program logic, must parse out into elements.

`mysql_fetch_assoc(recordset)` "pops" row off of recordset into an associative array and advanced internal pointer to next row. Returns false if there are no more rows to process



key	bday	class	name

key	bday	class	name

Processing as an associative array

```
$line = mysql_fetch_assoc($answer);
```

The above retrieves the next line of the result set into an associative array.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-5.php

To handle individual fields:

```
echo $line['fieldname'];
```

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-6.php

Loop through all lines in resultset using while loop.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-7.php

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-8.php

Fetch-ing a line of the resultset

`mysql_fetch_assoc($resultset)`—returns next line of resultset as an associative array.

```
while ($line = mysql_fetch_assoc($results))
  echo $line['title'];
```

`mysql_fetch_row($resultset)`—returns next line of resultset as a numeric (0 based) array.

```
while ($line = mysql_fetch_row($results))
  echo $line[0]; //first field in results
```

`mysql_fetch_array($resultset)`—returns next line of resultset as both an associative and numeric array.

Building a table from query results

1. Verify query returned results. If not, display message and quit.
2. Output a table open tag.
3. Read first row of resultset into associative array.
4. Use a loop to output top row of table containing field names.
5. "Rewind" resultset.
6. Set up a nested loop to process table contents.
7. Output table close tag.

Taking apart a single row of a resultset

A foreach loop can be used to process a single line of a resultset (just like a normal associative array).

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-9.php

Technique can be used to build output in table format.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-10.php

Table can be output with headers using

```
mysql_data_seek($resultset, $row_to_seek);
```

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-11.php

Building a table from query results

1. Verify query returned results.
2. Output a table open tag.
3. Read first row of output into associative array.
4. Use a loop to output table header row containing *field names*.
5. Reset resultset pointer to row 0.
6. Use a loop to process all rows in resultset. Use a nested foreach loop to process individual table cells.
7. Close the table

Freeing Resources

Record sets returned from queries can be large. Use **`mysql_free_result($resultset)`** to conserve memory. (Automatically done at end of script.)

Other Database Interaction

Insert, Update, Delete, etc. feature similar code, but less involved processing of results.

http://einstein.etsu.edu/~pittares/CSCI2910/examples/5-12.htm

Other MySQL Operations

Select, Show, Explain, and Describe return a record set or FALSE. Other operations (Insert, Update, etc.) return TRUE or FALSE.

`mysql_list_dbs()`—returns, as a record set, the list of databases.

`mysql_list_tables($dbname)`—returns, as a recordset, the list of tables in a database.

`mysql_affected_rows()`—returns as an integer the number of rows affected by the last insert, update, etc.

Copyrights

Presentation prepared by and copyright of Dr. Tony Pittarese, East Tennessee State University, Computer and Information Sciences Dept. (pittares@etsu.edu)

Podcast lecture related to this presentation available via ETSU iTunesU.