ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chapter 6.1.
Using MySQL with PHP

Objectives

- To understand the advantages of using databases to store Web data
- To learn how to prepare a MySQL database for use with PHP
- To learn how to store, retrieve, and update data in a MySQL database

Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

What is a database?

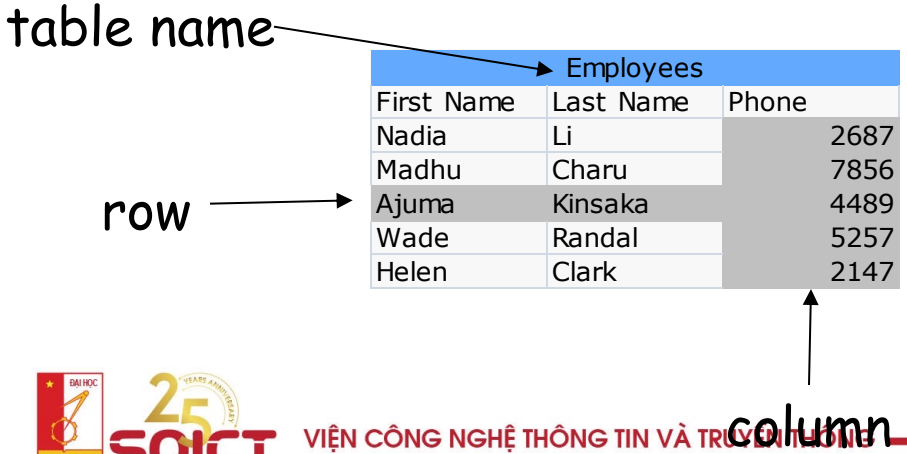
- An organized collection of data, generally stored and accessed electronically from a computer system.
- Database management systems (DBML): like MySQL, SQL Server, etc.

Advantages of Databases Over Files

- Faster access
- Better concurrent access
- Easier changes to data and scripts
- Increased security

Relational Database?

- A database is a collection of tables with defined relationships between them
- Columns define attributes of the data
  - All data in a column must have the same data type
- A record is stored in a row



Employees		
First Name	Last Name	Phone
Nadia	Li	2687
Madhu	Charu	7856
Ajuma	Kinsaka	4489
Wade	Randal	5257
Helen	Clark	2147

Primary key

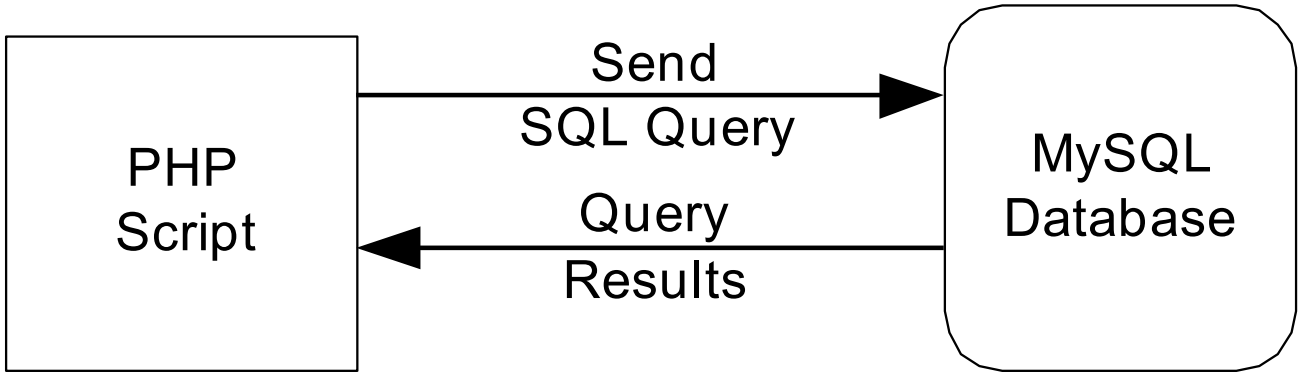
Product Number	Product	Cost	Weight	Number Avail
0	Hammer	$5.00	12	123
1	Screw Driver	$3.00	2	144
2	Wrench	$2.50	1.5	244

Which Database System

- PHP works with a variety of databases that include:
  - Oracle
  - Access
  - Ingres
  - SQL Server
  - MySQL
- Will use MySQL since simple to use, free and very popular.

Using A Query Language

- When using a database, use a separate query language to work with database
- Within MySQL, use Structured Query Language (SQL), to access database



Use MySQL Command line

- Start MAMP/XAMPP/...
- Start the server
- Open Terminal
  - MAMP:

    /Applications/MAMP/Library/bin/mysql --host=localhost -uroot -proot
  - XAMPP:

    cd [path]/xampp/mysql/bin/mysql.exe -u root -p

Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

2. Basic SQL commands

- Connecting to MySQL from the Command Line

```
mysql -u username -p
```

E.g.:

```
mysql -u root
```

- To EXIT MySQL:

```
exit;
```

2. Basic SQL Commands (2)

- SQL statements end with a semicolon
- View databases

```
SHOW DATABASES;
```

- Creating a database

```
CREATE DATABASE trii;
```

- Importing a database:

```
mysql -u username -p password
        databasename < filename.sql
```

E.g.:

```
mysql -u root trii < trii.sql
```

2. Basic SQL Commands (2)

- Use database *databasename*

```
USE databasename;
```

- Display all tables in a database

```
SHOW TABLES;
```

- View column details for a table

```
DESC tablename;
```

Creating a Database Instance

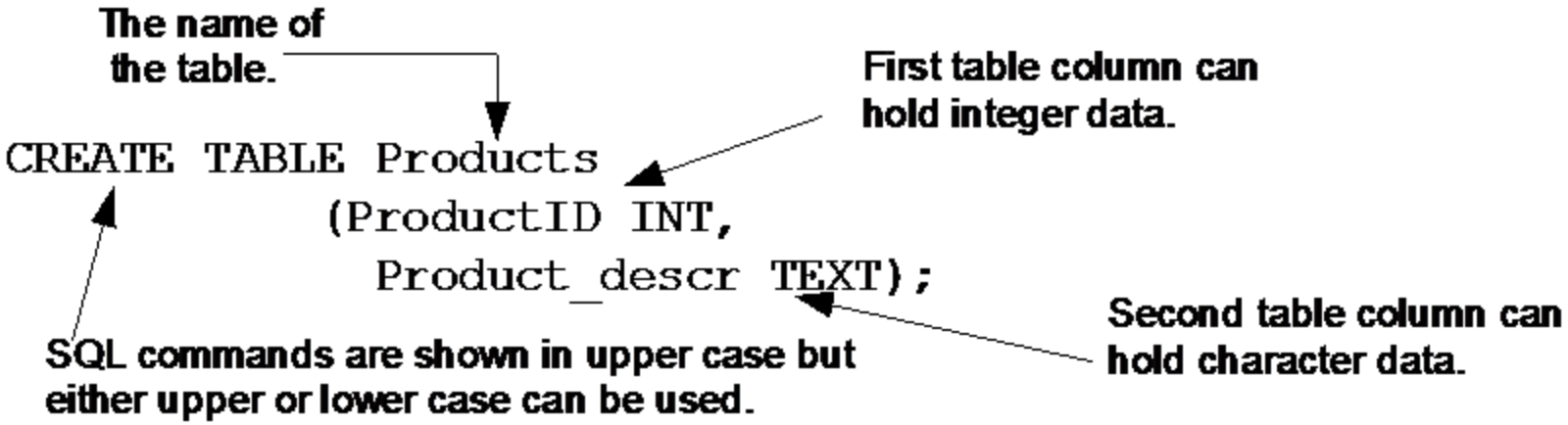
- Once you have access to a server with MySQL installed, need to get a database instance created for you.
  - Usually created by a database administrator
  - Creates a database instance, userid and password.

Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

3. Creating a table

- Once database instance is created need to create your tables.
  - Use SQL CREATE TABLE command



MySQL Data Types

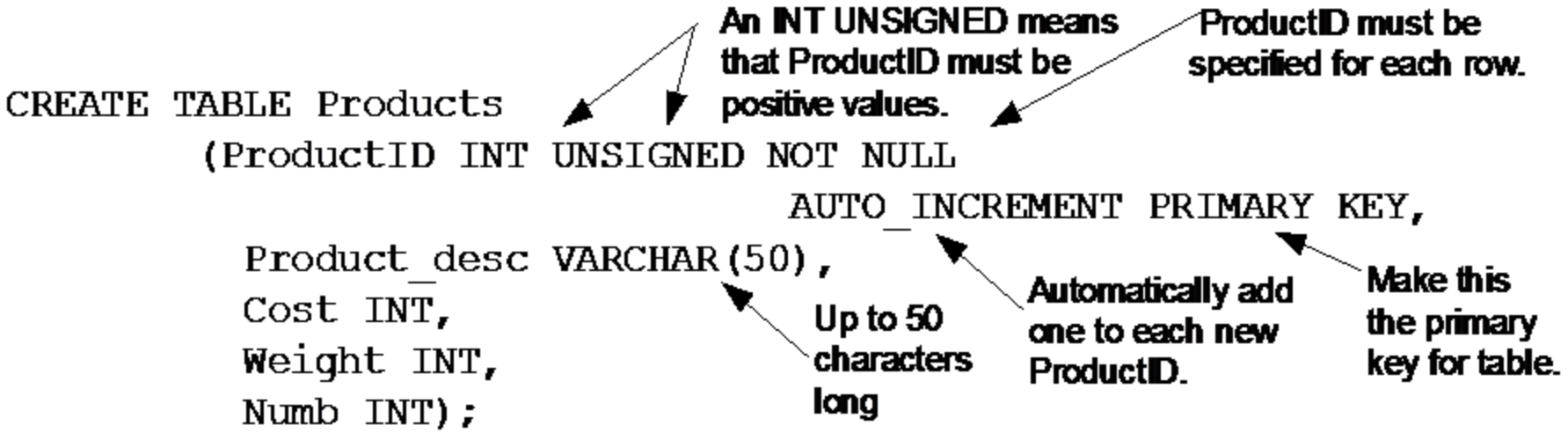
- TEXT
  - hold a large amount of character data
  - Use space inefficiently since it reserves space for up to 65,535 characters.
- CHAR(N)
  - hold a fixed length string of up to N characters (N must be less than 256).
- VARCHAR(N)
  - hold a variable length string of up to N characters
  - removes any unused spaces on the end of the entry.

MySQL Data Types (2)

- INT
  - hold an integer with a value from about –2 billion to about 2 billion.
- INT UNSIGNED
  - hold an integer with a value from 0 to about 4 billion.
- SMALLINT
  - hold an integer with a value from –32,768 to 32,767.
- SMALLINT UNSIGNED
  - hold an integer with a value from 0 to 65,535.
- DECIMAL(N,D)
  - a number that supports N total digits, of which D digits are to the right of the decimal point.

Some additional CREATE TABLE Options

- Can specify some additional options in CREATE TABLE:



Issuing CREATE TABLE From PHP Script Segment

```
1. $connect = mysql_connect($server, $user, $pass);
2. if ( !$connect ) {
3.         die ("Cannot connect to $server using $user");
4. } else {
5.     mysql_select_db('MyDatabaseName');
6.     $SQLcmd = 'CREATE TABLE Products(
                        ProductID INT UNSIGNED NOT NULL
                                AUTO_INCREMENT PRIMARY KEY,
                        Product_desc VARCHAR(50), Cost INT,
                        Weight INT, Numb INT)';
7.     mysql_query($SQLcmd, $connect);
8.     mysql_close($connect);
9. }
```

Connect to MySQL

Issue the SQL query to the database.

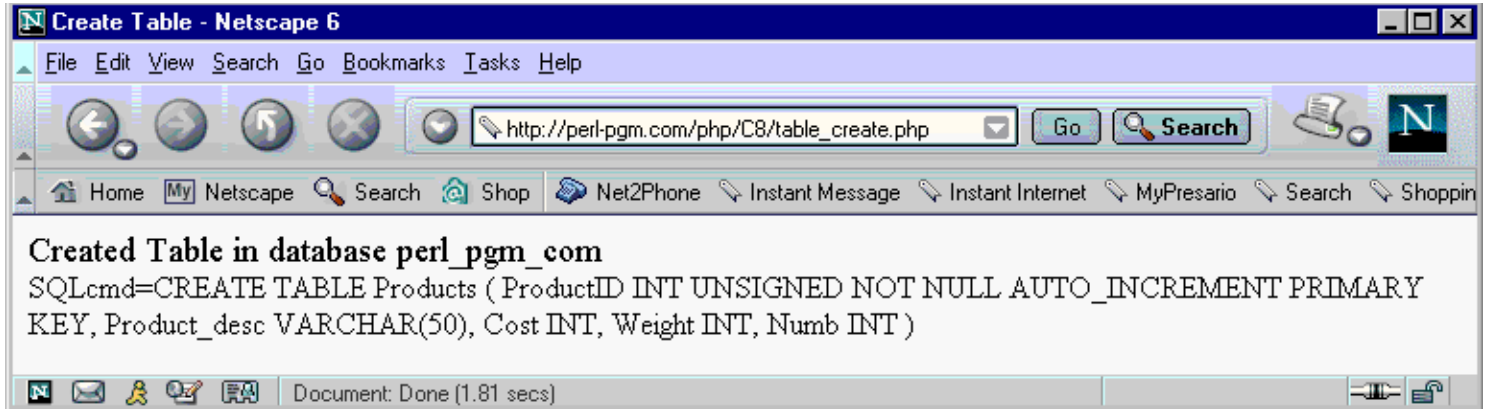
Full Script

```
1. <html><head><title>Create Table</title></head><body>
2. <?php
3. $server = 'localhost';
4. $user = 'phppgm';
5. $pass = 'mypasswd';
6. $mydb = 'mydatabase';
7. $table_name = 'Products';
8. $connect = mysql_connect($server, $user, $pass);
9. if (!$connect) {
10.       die ("Cannot connect to $server using $user");
11. } else {
12.       $SQLcmd = "CREATE TABLE $table_name (
              ProductID INT UNSIGNED NOT NULL
                                AUTO_INCREMENT PRIMARY KEY,
                  Product_desc VARCHAR(50),
                  Cost INT, Weight INT, Numb INT)";
```

Full Script (2)

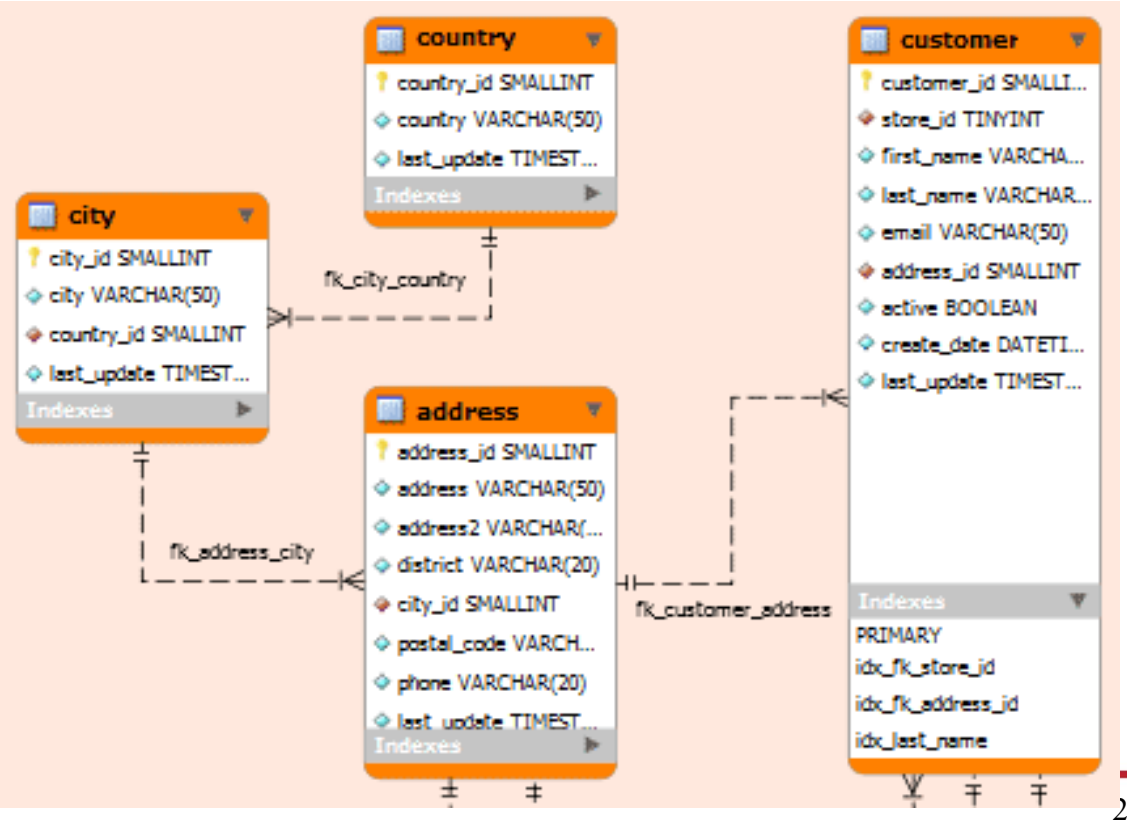
```
13.  mysql_select_db($mydb);
14.  if (mysql_query($SQLcmd, $connect)){
15.     print '<font size="4" color="blue" >Created
  Table';
16.     print "<i>$table_name</i> in
  database<i>$mydb</i><br></font>";
17.     print "<br>SQLcmd=$SQLcmd";
18.  } else {
19.     die ("Table Create Creation Failed
  SQLcmd=$SQLcmd");
20.  }
21.  mysql_close($connect);
22. }
23. ?></body></html>
```

Script Browser Output



MySQL Visual Designer Tools

- phpMyAdmin (web-app)
- MySQL Workbench (Win, Linux, Mac)
- **SQLyog**
- …

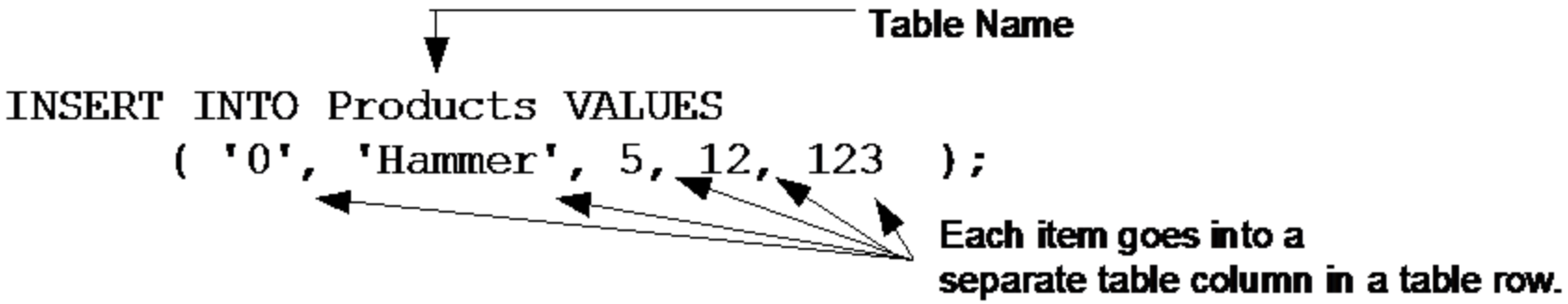


Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

4. Inserting data to a table

- Once database is created will need to insert data
- Use the SQL INSERT command



A Full Example

- Consider an application that allows end-user to enter inventory data:

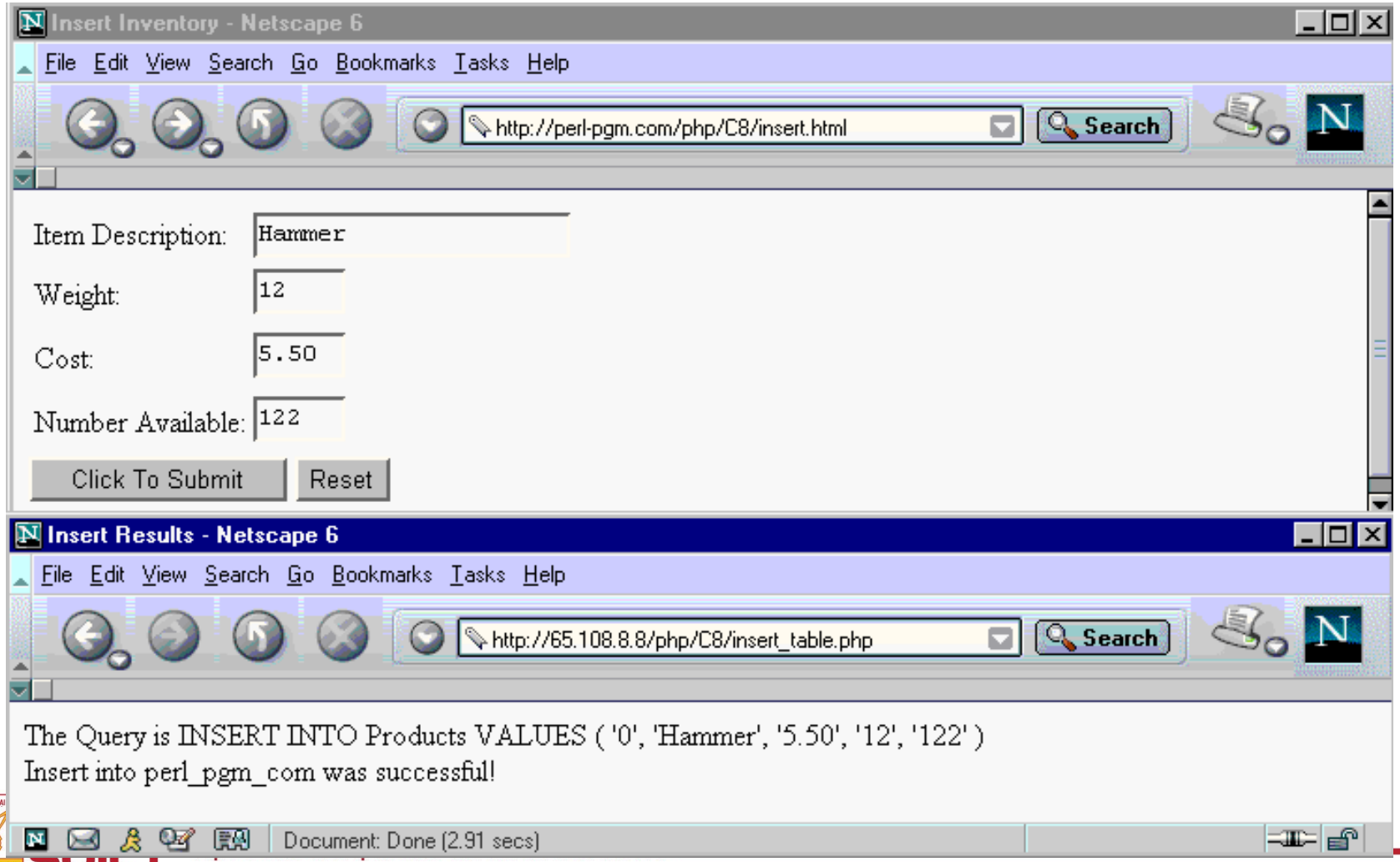
```
Item Description: <input type="text" size="20"
                  maxlength="20" name="Item">
Weight: <input type="text" size="5"
                  maxlength="20" name="Weight">
Cost: <input type="text" size="5"
                  maxlength="20" name="Cost">
Number Available:<input type="text" size="5"
                        maxlength="20"

  name="Quantity">
```

Receiving PHP Script

```
1. <html><head><title>Insert Results</title></head><body>
2. <?php
3. $host = 'localhost';
4. $user = 'phppgm';
5. $passwd = 'mypasswd';
6. $database = 'mydatabase';
7. $connect = mysql_connect($host, $user, $passwd);
8. $table_name = 'Products';
9. $query = "INSERT INTO $table_name VALUES ('0','$Item','$Cost','$Weight','$Quantity')";
10. print "The Query is <i>$query</i><br>";
11. mysql_select_db($database);
12. print '<br><font size="4" color="blue">';
13. if (mysql_query($query, $connect)){
14.    print "Insert into $database was successful!</font>";
15. } else {
16.    print "Insert into $database failed!</font>";
17. } mysql_close ($connect);
18. ?></body></html>
```

Script Output

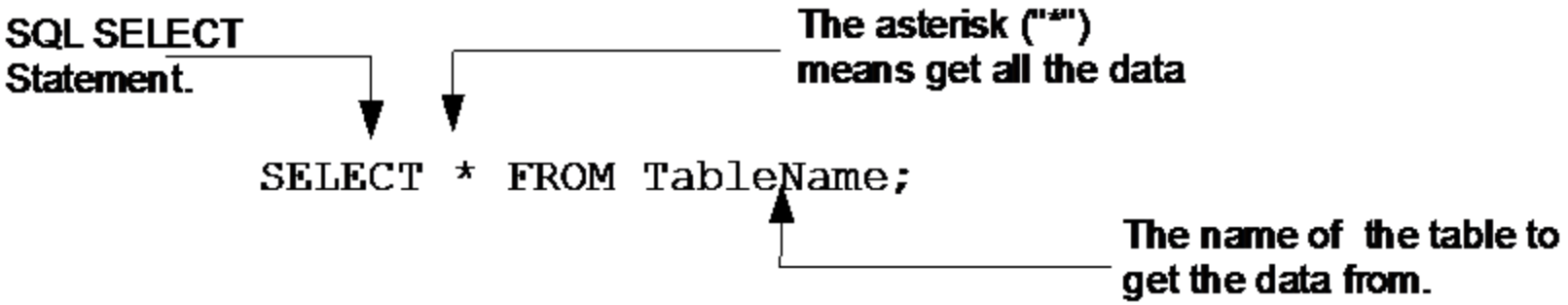


Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

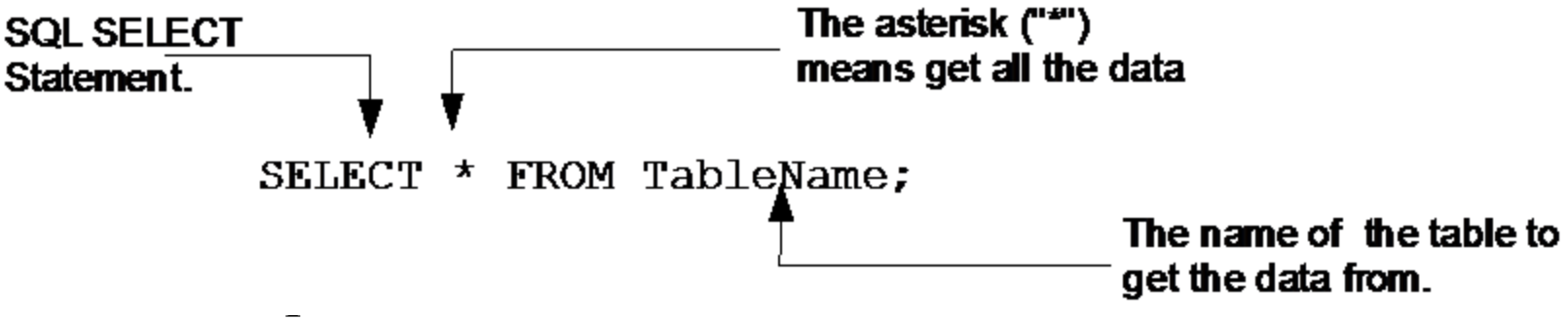
5. Retrieving data from a table

- Two major ways to retrieve data:
  - Retrieving all elements from a table
  - Searching for specific records in a table
- To retrieve all data, use following SQL command



5. Retrieving Data (2)

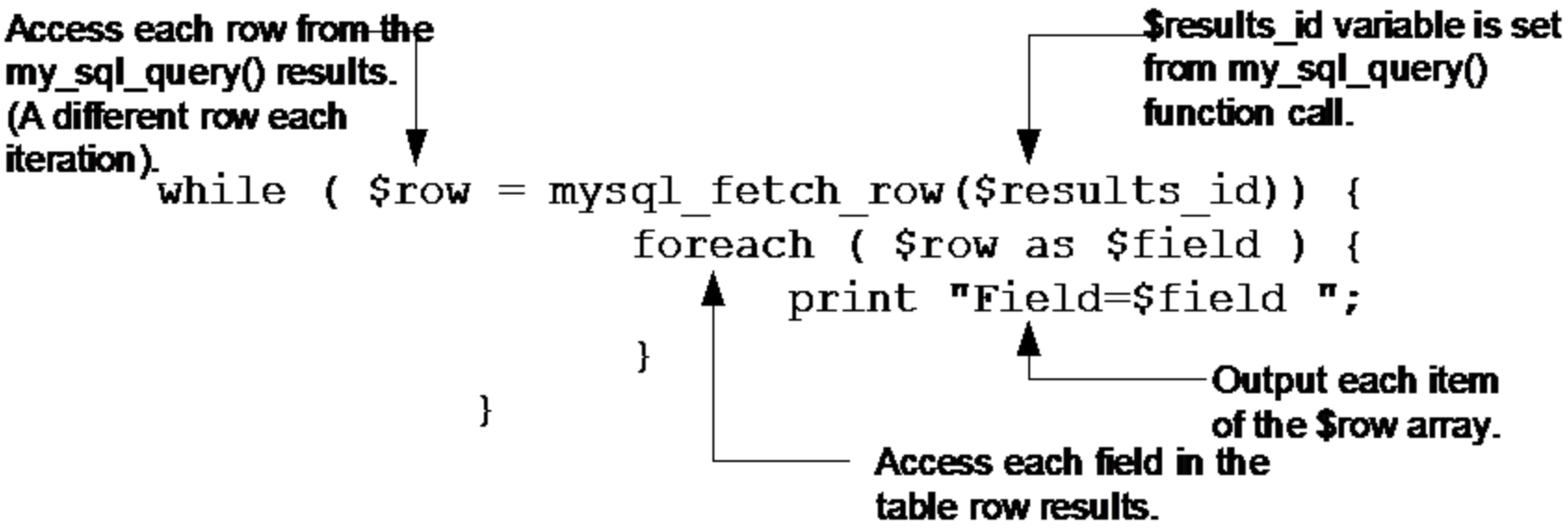
- To retrieve all data, use following SQL command




```
1. $connect = mysql_connect('Localhost', 'phppgm',
   'mypasswd');
2. $SQLcmd = 'SELECT * FROM Products';
3. mysql_select_db('MyDatabase');
4. $results_id = mysql_query($SQLcmd, $connect);
```

5.1. Using mysql_fetch_row()

- Use the mysql_fetch_row() function to retrieve data on row at a time



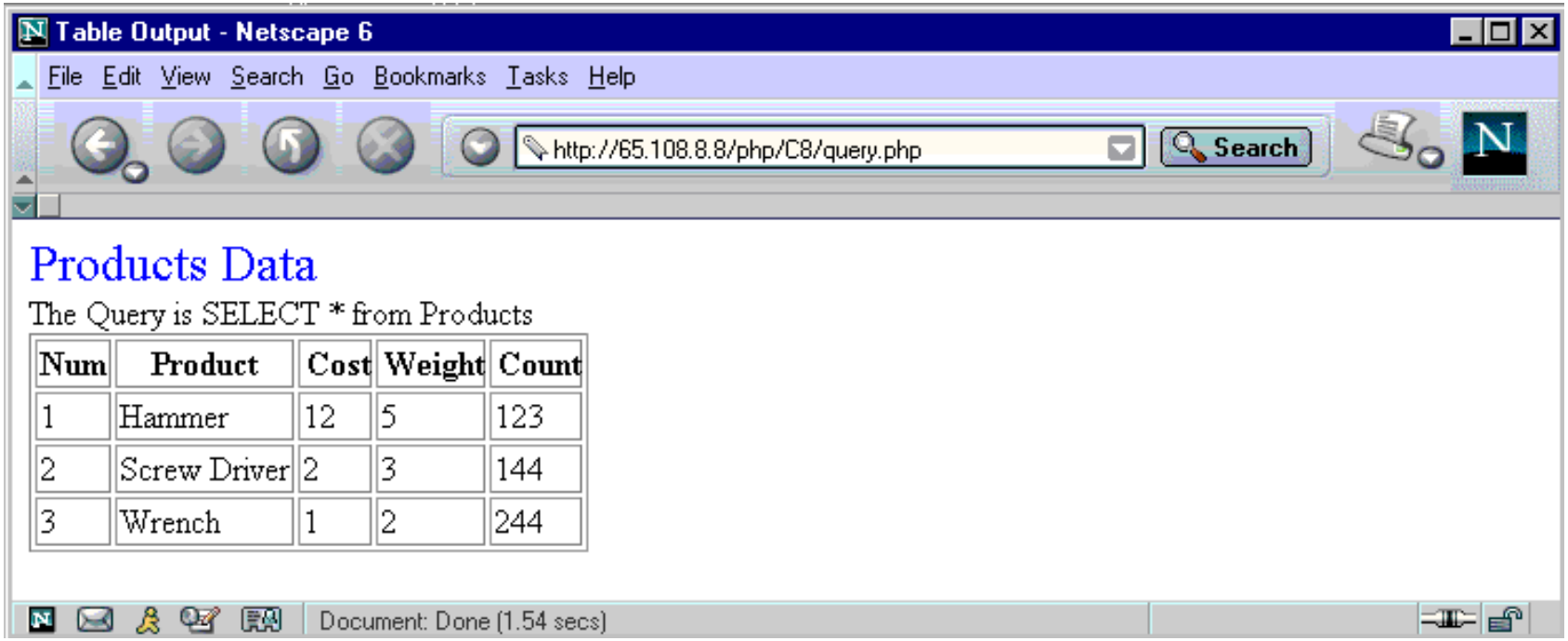
A Script Example

```
1. <html><head><title>Table Output</title></head><body>
2. <?php
3. $host= 'localhost';
4. $user = 'phppgm';
5. $passwd = 'mypasswd';
6. $database = 'phppgm';
7. $connect = mysql_connect($host, $user, $passwd);
8. $table_name = 'Products';
9. print '<font size="5" color="blue">';
10. print "$table_name Data</font><br>";
11. $query = "SELECT * FROM $table_name";
12. print "The query is <i>$query </i><br>";
13. mysql_select_db($database);
14. $results_id = mysql_query($query, $connect);
15. if ($results_id) {
16.       print '<table border=1>';
17.         print '<th>Num<th>Product<th>Cost<th>Weight<th>Count';
```

A Script Example (2)

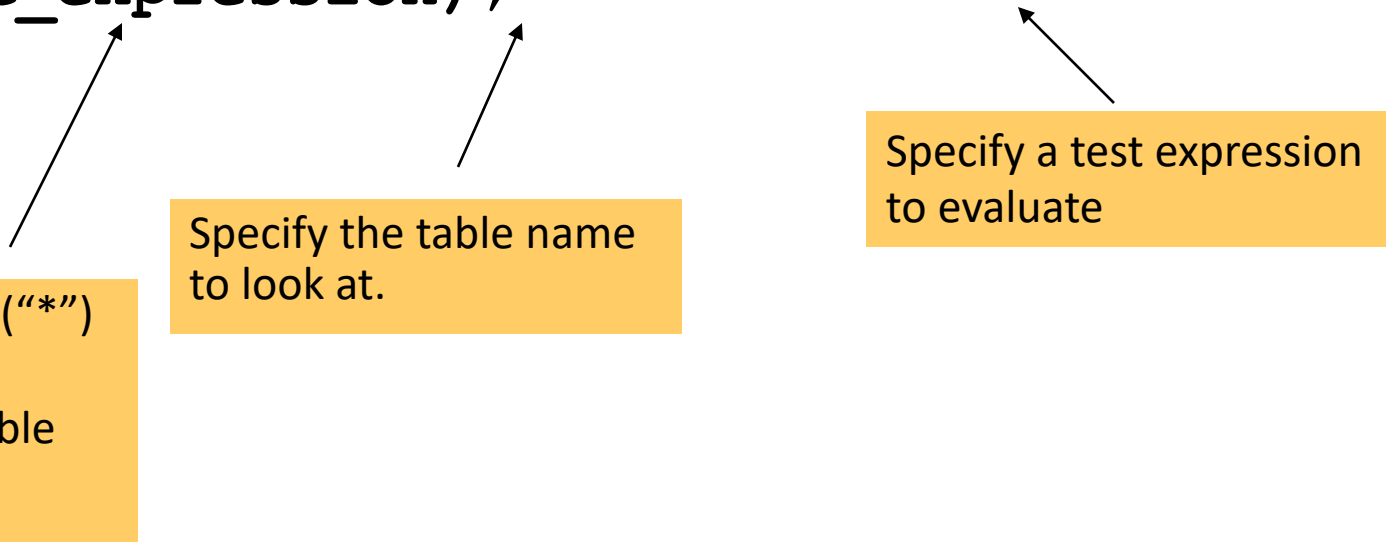
```
18.       while ($row =
  mysql_fetch_row($results_id)){
19.           print '<tr>';
20.           foreach ($row as $field) {
21.                 print "<td>$field</td> ";
22.           }
23.           print '</tr>';
24.       }
25. } else { die ("Query=$query failed!"); }
26. mysql_close($connect);
27. ?> </table></body></html>
```

Script Output



5.2. Searching For Specific Records

- Use the SELECT SQL statement with a WHERE clause
  - **`SELECT * FROM TableName WHERE (test_expression);`**



Selected WHERE CLAUSE Test Operators

Operator	SQL Query Example	Meaning
=	SELECT * FROM Products WHERE (Product_desc = 'Hammer');	Retrieve those rows from the Products table that have a Product_desc column with a value **equal to** Hammer.
>	SELECT * FROM Products WHERE (Cost > '5');	Retrieve those rows from the Products table that have a Cost column with a value **greater than** 5.
<	SELECT * FROM Products WHERE (Numb < '3');	Retrieve those rows from the Products table that have a Numb column with a value **less than** 3.
<=	SELECT * FROM Products WHERE (Cost <= '3');	Retrieve those rows from the Products table that have a Cost column with a value **less than or equal to** 3.
>=	SELECT * FROM Products WHERE (Weight >= '10');	Retrieve those rows from the Products table that have a Weight column with a value **greater than or equal to** 10.

Consider the following example …

- The following example searches a hardware inventory database for a specific part name entered by the user.
- The form uses the following key HTML form element definition.
  - `<input type="text" name="Search" size="20">`

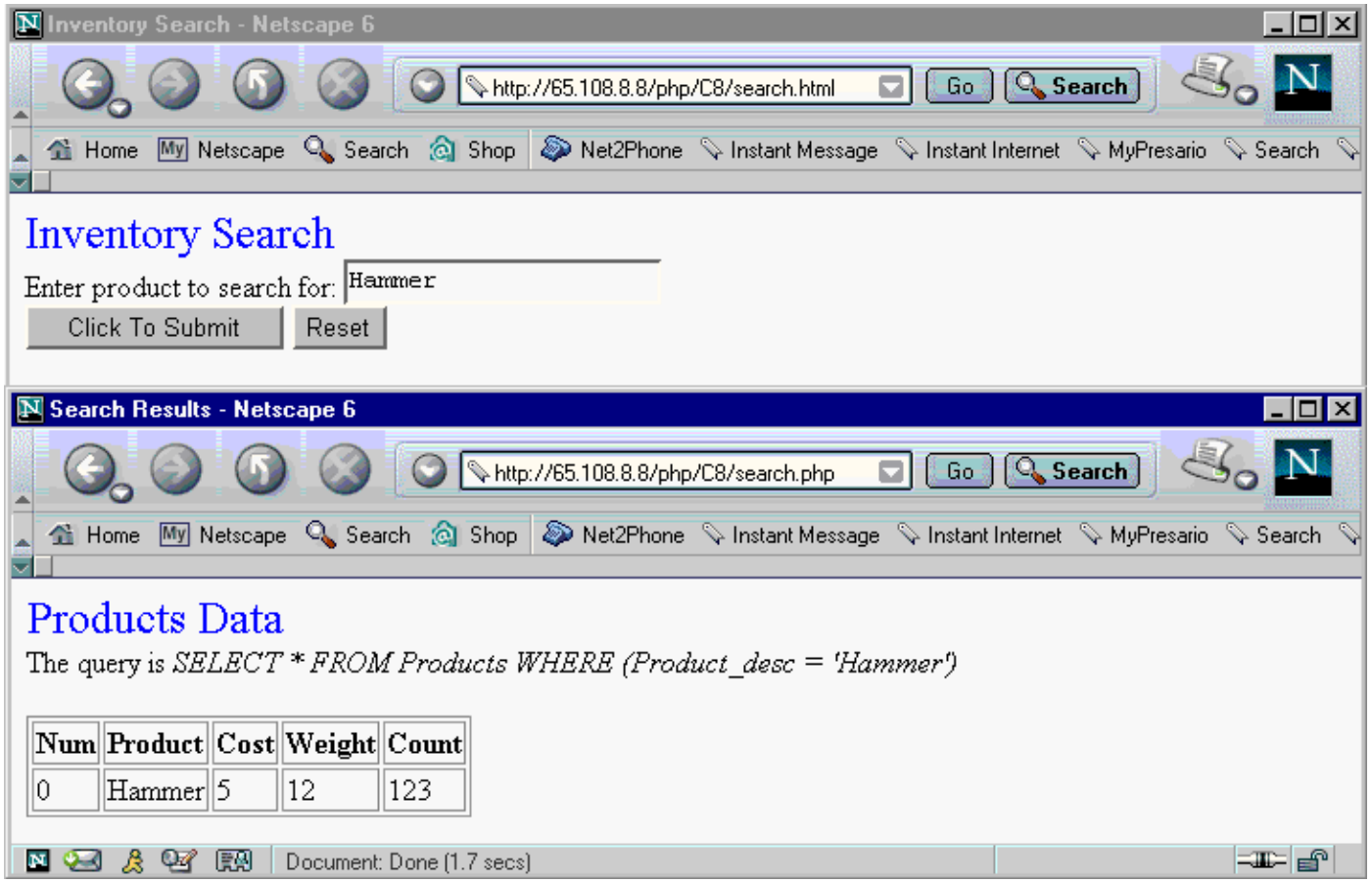
PHP Source

```
1. <html><head><title>Search Results</title></head><body>
2. <?php
3. $host= 'localhost';
4. $user = 'phppgm';
5. $passwd = 'mypasswd';
6. $database = 'phppgm';
7. $connect = mysql_connect($host, $user, $passwd);
8. $table_name = 'Products';
9. print '<font size="5" color="blue">';
10. print "$table_name Data</font><br>";
11. $query = "SELECT * FROM $table_name WHERE
                                (Product_desc = '$Search')";
12. print "The query is <i>$query</i> <br>";
13. mysql_select_db($database);
14. $results_id = mysql_query($query, $connect);
```

PHP Source (2)

```
15. if ($results_id) {
16.  print '<br><table border=1>';
17.  print '<th>Num<th>Product<th>Cost<th>Weight <th>Count';
18.  while ($row = mysql_fetch_row($results_id)) {
19.    print '<tr>';
20.          foreach ($row as $field) {
21.                print "<td>$field</td> ";
22.                }
23.          print '</tr>';
24.    }
25. } else { die ("query=$Query Failed");}
26. mysql_close($connect);
27. ?> </body></html>
```

Would have the following output …

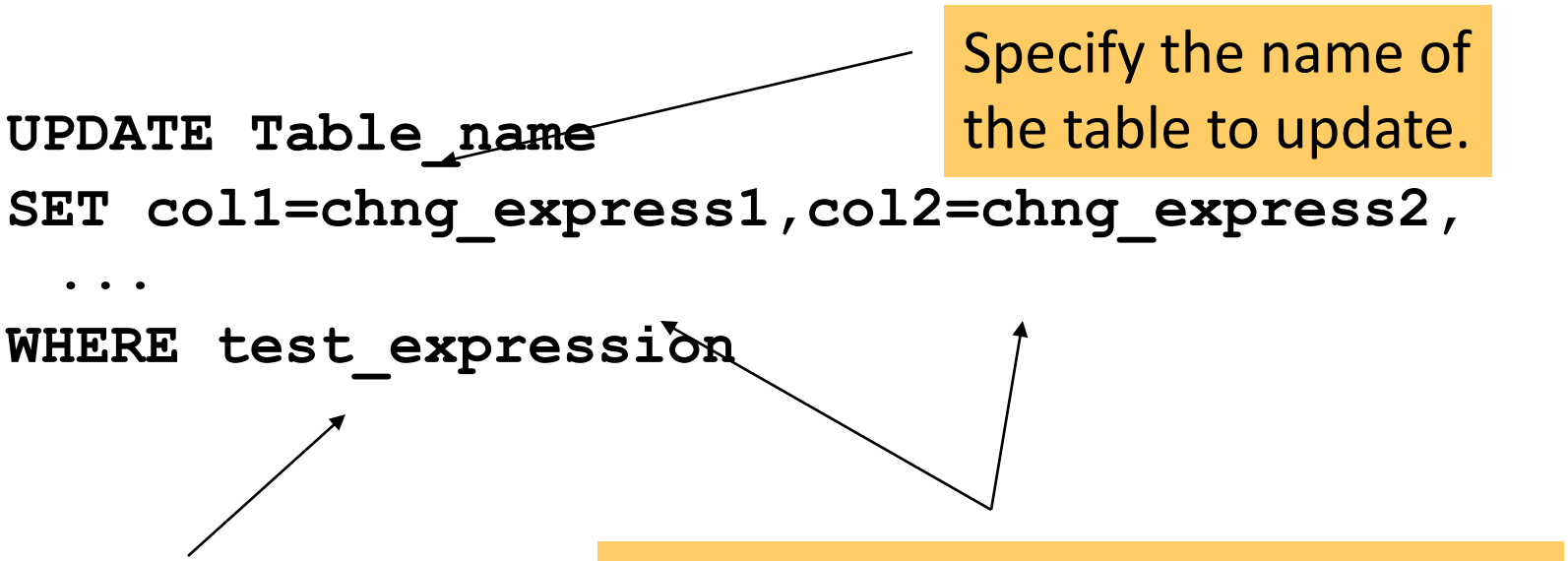


Content

1. Database and MySQL Overview
2. Basic SQL commands
3. Creating a table
4. Inserting data to a table
5. Retrieving data from a table
6. Updating data for a table

6. Updating data for a table

- Use SQL UPDATE command when needing to update a database record:



For Example …

- The following searches the Products table for values of Product_desc equal to Hammer.

```
UPDATE Products
SET Cost=2
WHERE Product_desc = 'Hammer'
```

For Example …

- The following looks through the Products table for values of Product_desc equal to Hammer.
- When it finds it, it decrements the Count column value by 1.

```
UPDATE Products
SET Count=Count-1
WHERE 'Product_desc=Hammer'
```

A Full Example …

- Consider the following example
  - Displays current inventory
  - Asks end-user to decrement value for 1 item
  - Uses the following HTML

```
Hammer: <input type="radio" name="Product"
                                value="Hammer">
Screwdriver: <input type="radio"
          name="Product"
  value="Screwdriver”>
Wrench: <input type="radio" name="Product"

    value="Wrench">
```

Full Example

```
1. <html><head><title>Product Update Results</title></head><body>
2. <?php
3. $host= 'localhost';
4. $user = 'phppgm';
5. $passwd = 'mypasswd';
6. $database = 'phppgm';
7. $connect = mysql_connect($host, $user, $passwd);
8. $table_name = 'Products';
9. print '<font size="5" color="blue">';
10. print "Update Results for Table
                          $table_name</font><br>\n";
11. $query = "UPDATE $table_name
                  SET Numb = Numb-1
                  WHERE (Product_desc = '$Product')";
12. print "The query is <i> $query </i> <br><br>\n";
13. mysql_select_db($database);
```

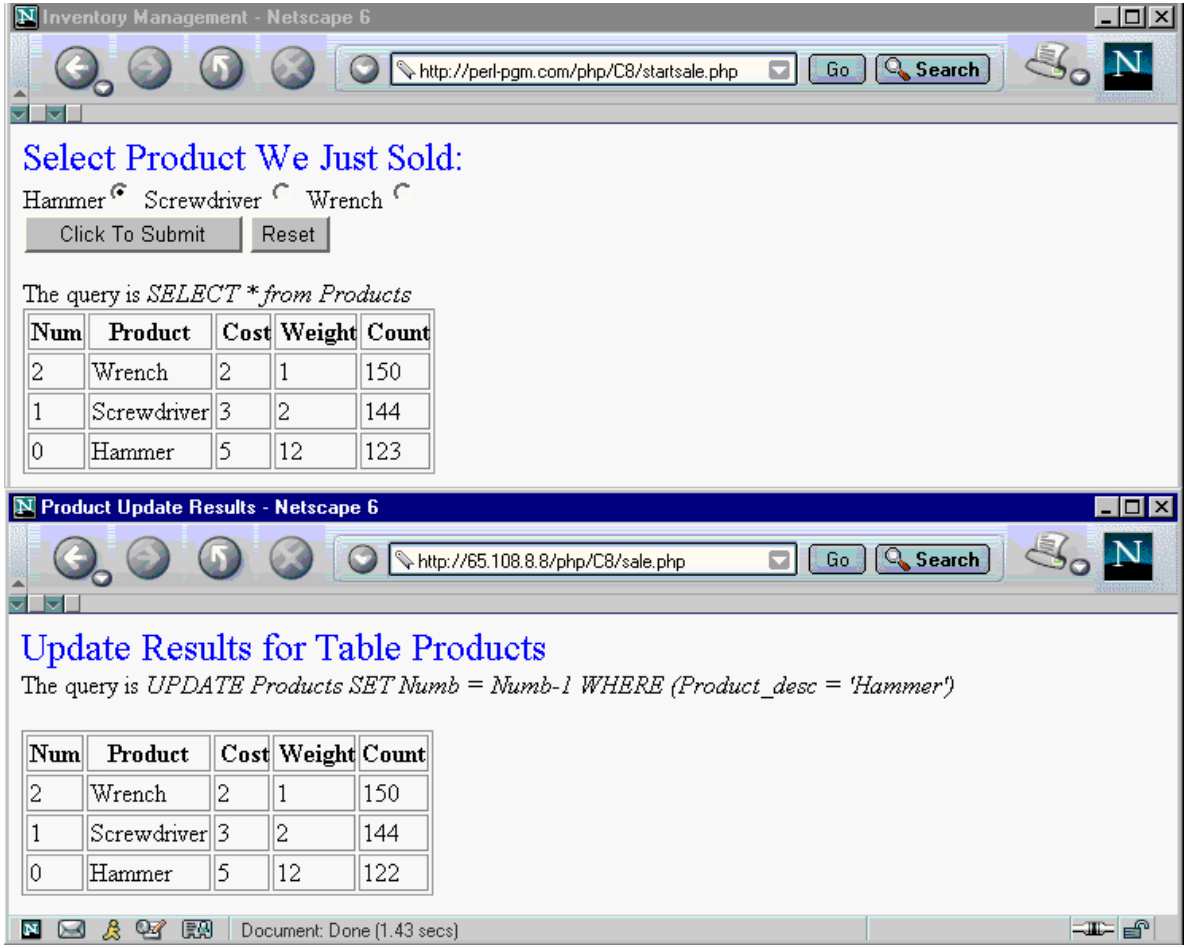
A Full Example (2)

```
14. $results_id = mysql_query($query, $connect);
15. if ($results_id){
16.   Show_all($connect, $database,$table_name);
17. } else {
18.   print "Update=$query failed";
19. }
20. mysql_close($connect);
```

A Full Example (3)

```
21. function Show_all($connect, $database, $table_name){
22.    $query = "SELECT * from $table_name";
23.    $results_id = mysql_query($query, $connect);
24.    print '<table border=1><th> Num </th>
                 <th>Product</th><th>Cost</th>
                 <th>Weight</th><th>Count</th>';
26.    while ($row = mysql_fetch_row($results_id)) {
27.        print '<tr>';
28.       foreach ($row as $field){
29.       print "<td>$field</td> ";
30.       }
31.       print '</tr>';
32.    }
33. }
34. ?> </body></html>
```

Would output the following:



Question?