

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 (DBMS): 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

table name →

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

row →

column ↑

Primary key →

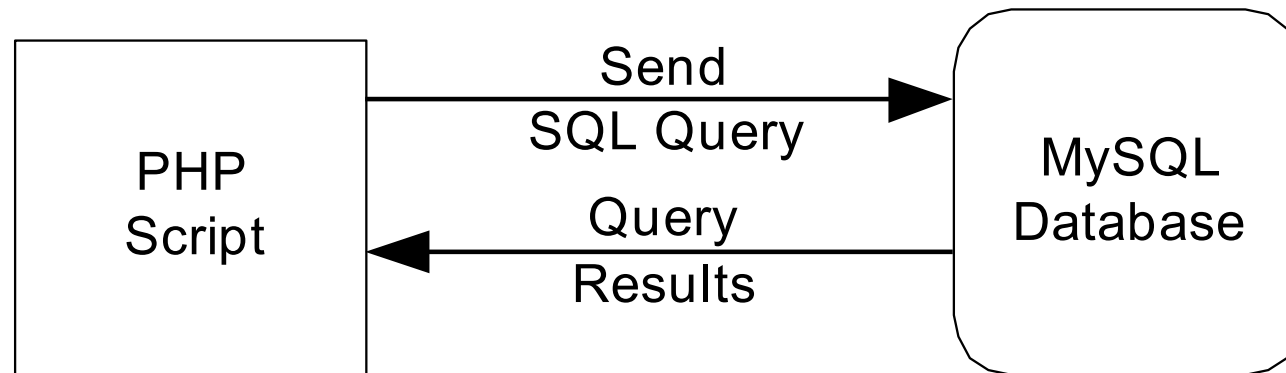
Product Number	Product	Cost	Weight	Number Avail
0	Hammer	\$5.00	12	123
1	Screw Driver	\$3.00	2	144
2	Wrench	\$2.50	1.5	244

Which Database System

- PHP works with a variety of databases that include:
 - SQL Server
 - MySQL
 - Oracle
 - Access
- 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

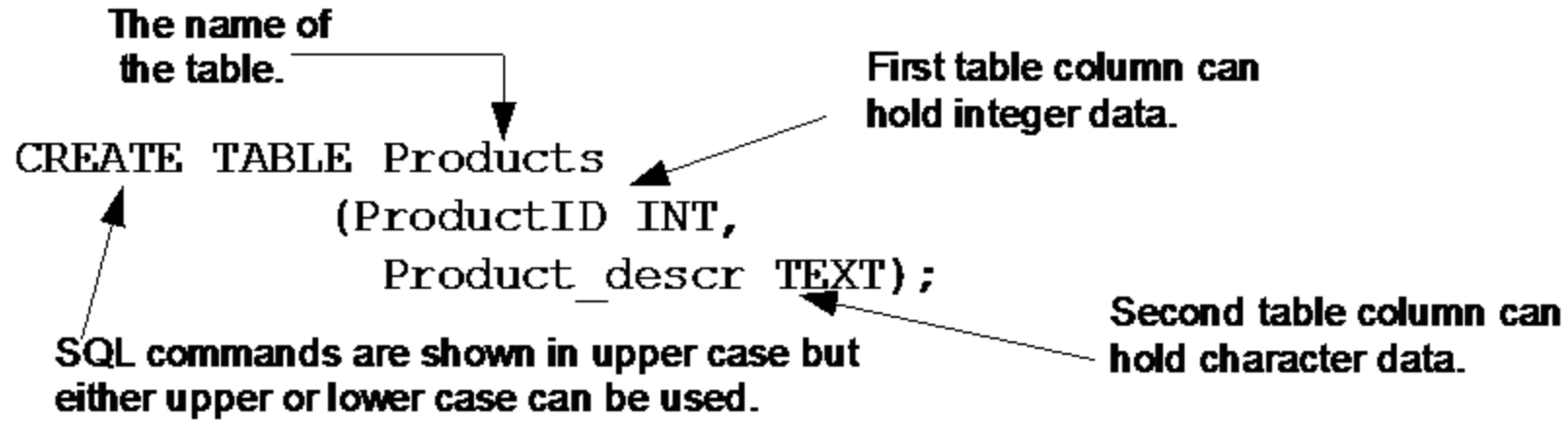
- Once you have access to a server with MySQL installed, need to get a database instance created for you.
 - Usually created by a database administrator
 - Creates a database instance, userid and password.

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



The diagram shows the SQL command `CREATE TABLE Products (ProductID INT, Product_descr TEXT);` with four annotations: 1. An arrow from 'The name of the table.' points to 'Products'. 2. An arrow from 'First table column can hold integer data.' points to 'INT'. 3. An arrow from 'Second table column can hold character data.' points to 'TEXT'. 4. An arrow from 'SQL commands are shown in upper case but either upper or lower case can be used.' points to 'CREATE'.

The name of the table.

First table column can hold integer data.

Second table column can hold character data.

SQL commands are shown in upper case but either upper or lower case can be used.

```
CREATE TABLE Products
(ProductID INT,
Product_descr TEXT);
```


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:

```
CREATE TABLE Products
(ProductID INT UNSIGNED NOT NULL
Product_desc VARCHAR(50),
Cost INT,
Weight INT,
Numb INT);
```

An INT UNSIGNED means that ProductID must be positive values.

ProductID must be specified for each row.

AUTO INCREMENT PRIMARY KEY,

Up to 50 characters long

Automatically add one to each new ProductID.

Make this the primary key for table.

Issuing CREATE TABLE From PHP Script Segment

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

Connect to
MySQL

Issue the SQL query
to the database.

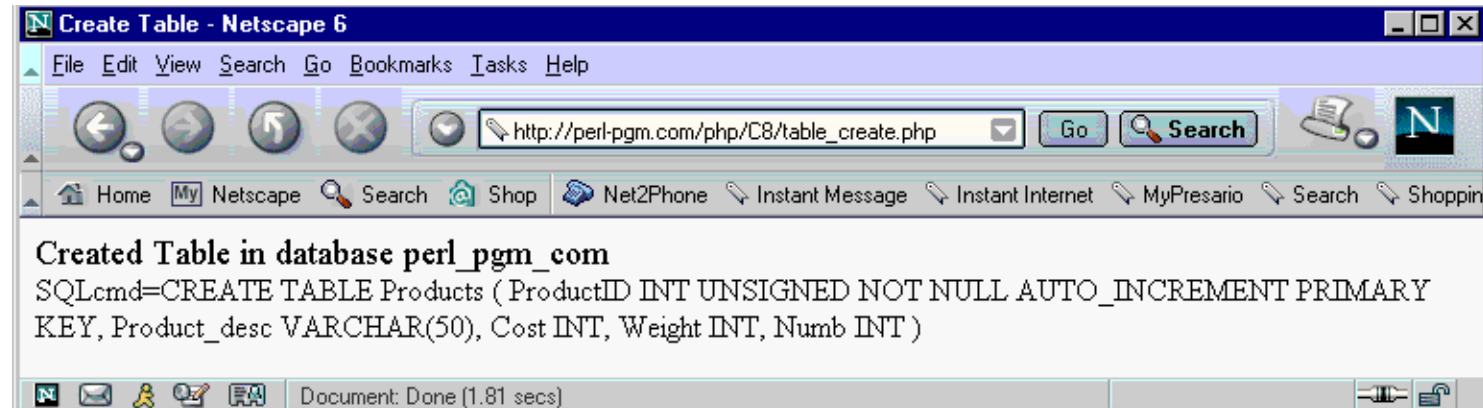
Full Script

```
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. $connect = mysqli_connect($server, $user, $pass, $mydb);
9. if (!$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)

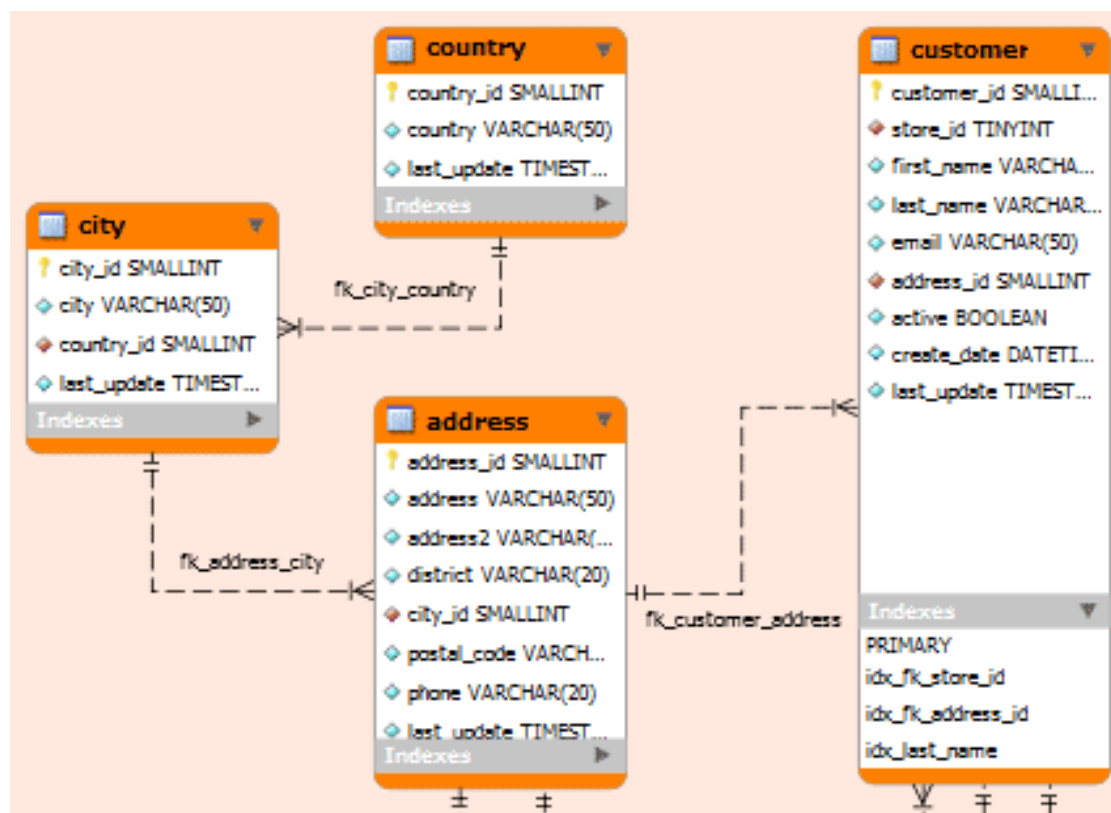
```
14.  if (mysqli_query($connect, $SQLcmd)) {
15.      print '<font size="4" color="blue" >Created Table';
16.      print "<i>$table_name</i> in database<i>$mydb</i><br></font>";
17.      print "<br>SQLcmd=$SQLcmd";
18.  } else {
19.      die ("Table Create Creation Failed SQLcmd=$SQLcmd");
20.  }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

Script Browser Output



MySQL Visual Designer Tools

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



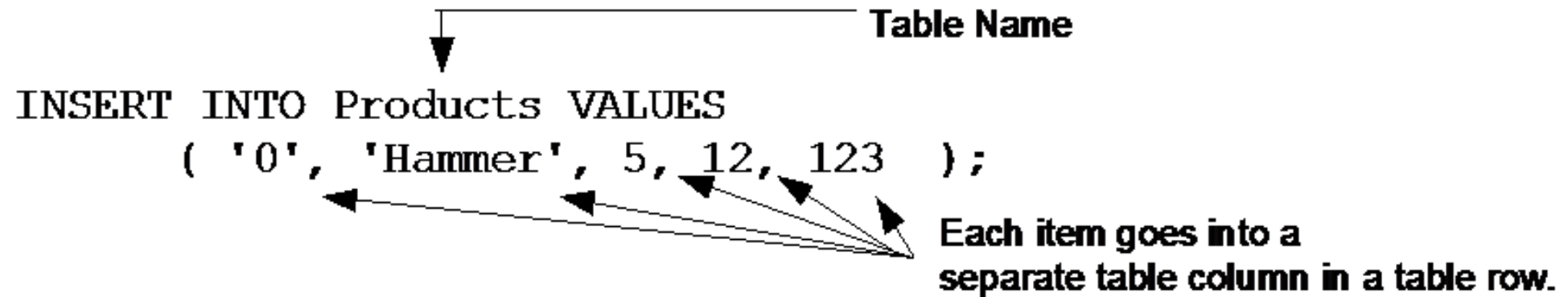
Content

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

The diagram shows an SQL INSERT statement with annotations. A horizontal line with a downward arrow points from the text 'Table Name' to the word 'Products' in the statement. Another set of arrows points from the text 'Each item goes into a separate table column in a table row.' to each of the five values in the VALUES clause: '0', 'Hammer', 5, 12, and 123.

```
INSERT INTO Products VALUES  
( '0', 'Hammer', 5, 12, 123 );
```

Table Name

Each item goes into a separate table column in a table row.

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">`

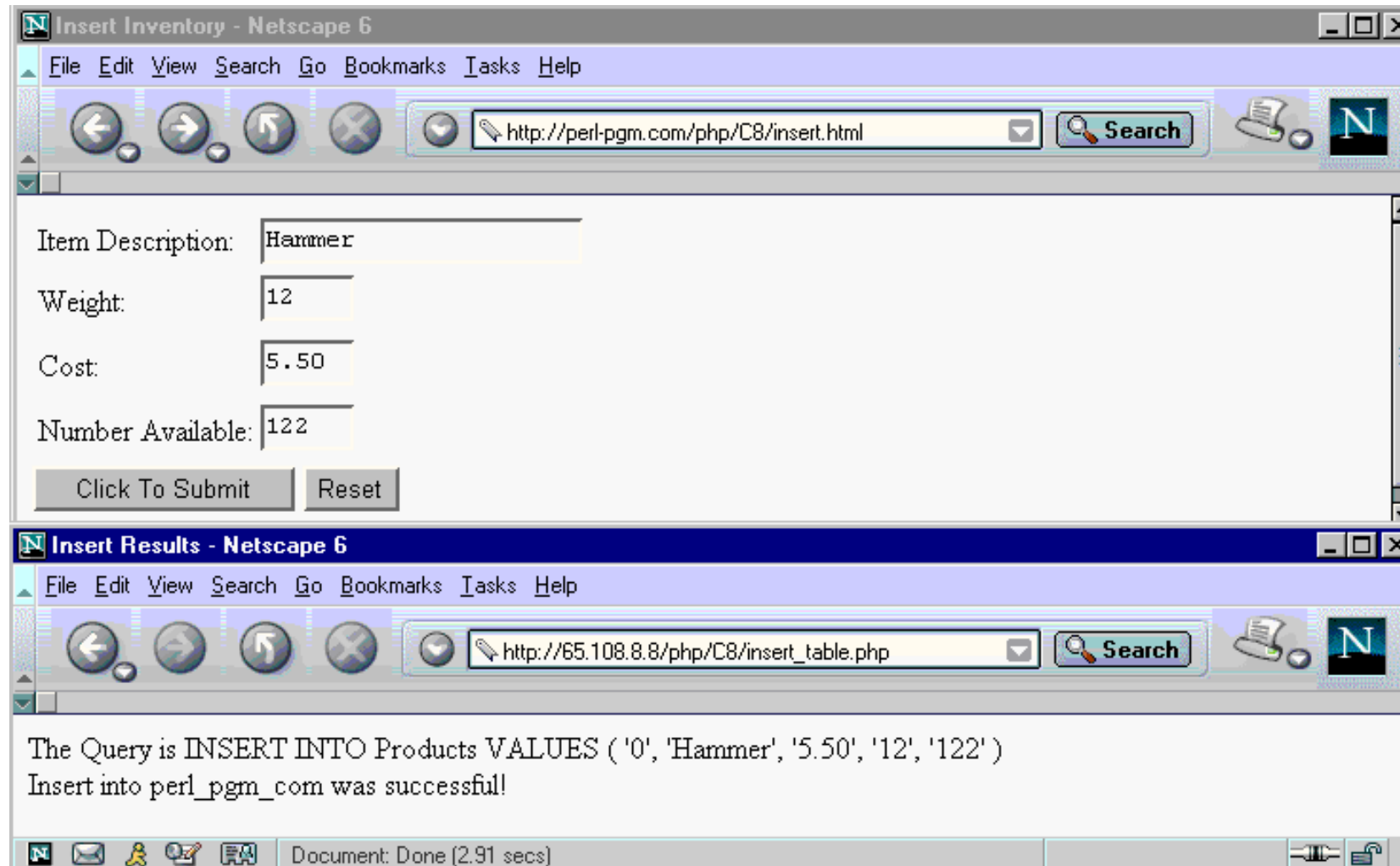
Cost: `<input type="text" size="5"
maxlength="20" name="Cost">`

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. $connect = mysqli_connect($host, $user, $passwd, $database);
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>";
12. print '<br><font size="4" color="blue">';
13. if (mysqli_query($connect, $query)){
14.     print "Insert into $database was successful!</font>";
15. } else {
16.     print "Insert into $database failed!</font>";
17. } mysqli_close ($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

SQL SELECT Statement.

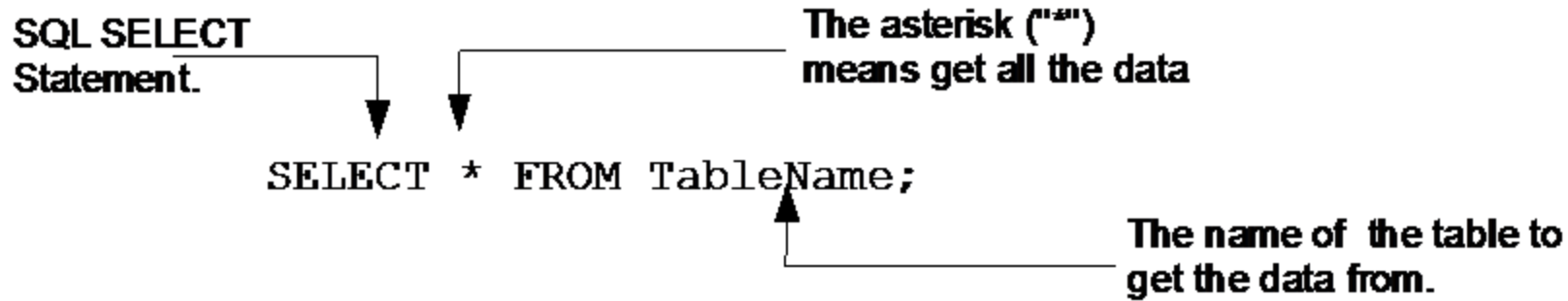
`SELECT * FROM TableName;`

The asterisk (*) means get all the data

The name of the table to get the data from.

5. Retrieving Data (2)

- To retrieve all data, use following SQL command



- For example

```
1. $connect = mysqli_connect('localhost', 'phppgm', 'mypasswd', 'mydatabase');  
2. $SQLcmd = 'SELECT * FROM Products';  
3. $result = mysqli_query($connect, $SQLcmd);
```


5.1. Using mysql_fetch_row()

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

```
if ($result = mysqli_query($con, $sql)) {  
    // Fetch one and one row  
    while ($row = mysqli_fetch_row($result)) {  
        printf ("%s (%s)\n", $row[0], $row[1]);  
    }  
    mysqli_free_result($result);  
}
```

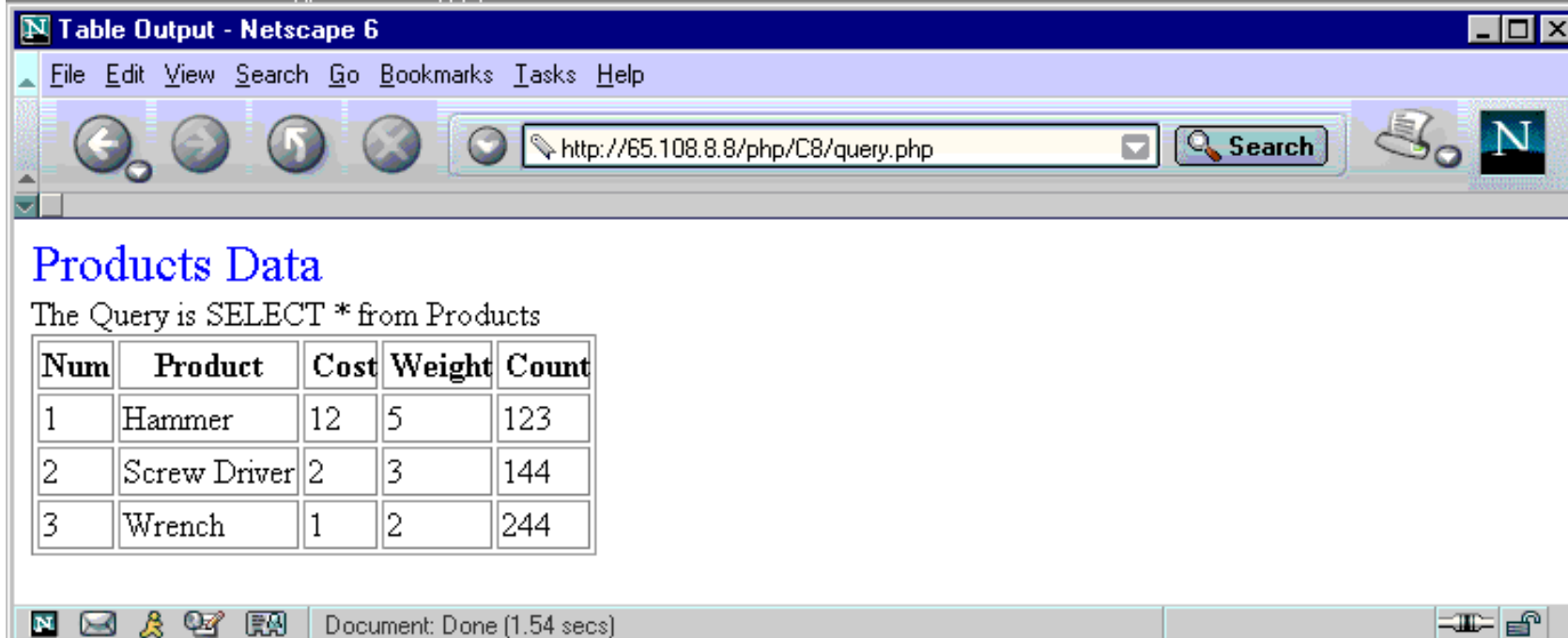
A Script Example

```
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, $database);
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>";
14. $results_id = mysqli_query($connect, $query);
15. if ($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 ($row = mysqli_fetch_row($results_id)) {
19.     print '<tr>';
20.     foreach ($row as $field) {
21.         print "<td>$field</td> ";
22.     }
23.     print '</tr>';
24. }
25.     mysqli_free_result($results_id);
25. } else { die ("Query=$query failed!"); }
26. mysqli_close($connect);
27. ?> </table></body></html>
```

Script Output



Products Data

The Query is SELECT * from Products

Num	Product	Cost	Weight	Count
1	Hammer	12	5	123
2	Screw Driver	2	3	144
3	Wrench	1	2	244

Document: Done (1.54 secs)

5.2. Searching For Specific Records

- Use the SELECT SQL statement with a WHERE clause

- **SELECT * FROM TableName WHERE (test_expression) ;**

The asterisk
("*") means
look at all
table
columns.

Specify the table
name to look at.

Specify a test expression
to evaluate

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">`

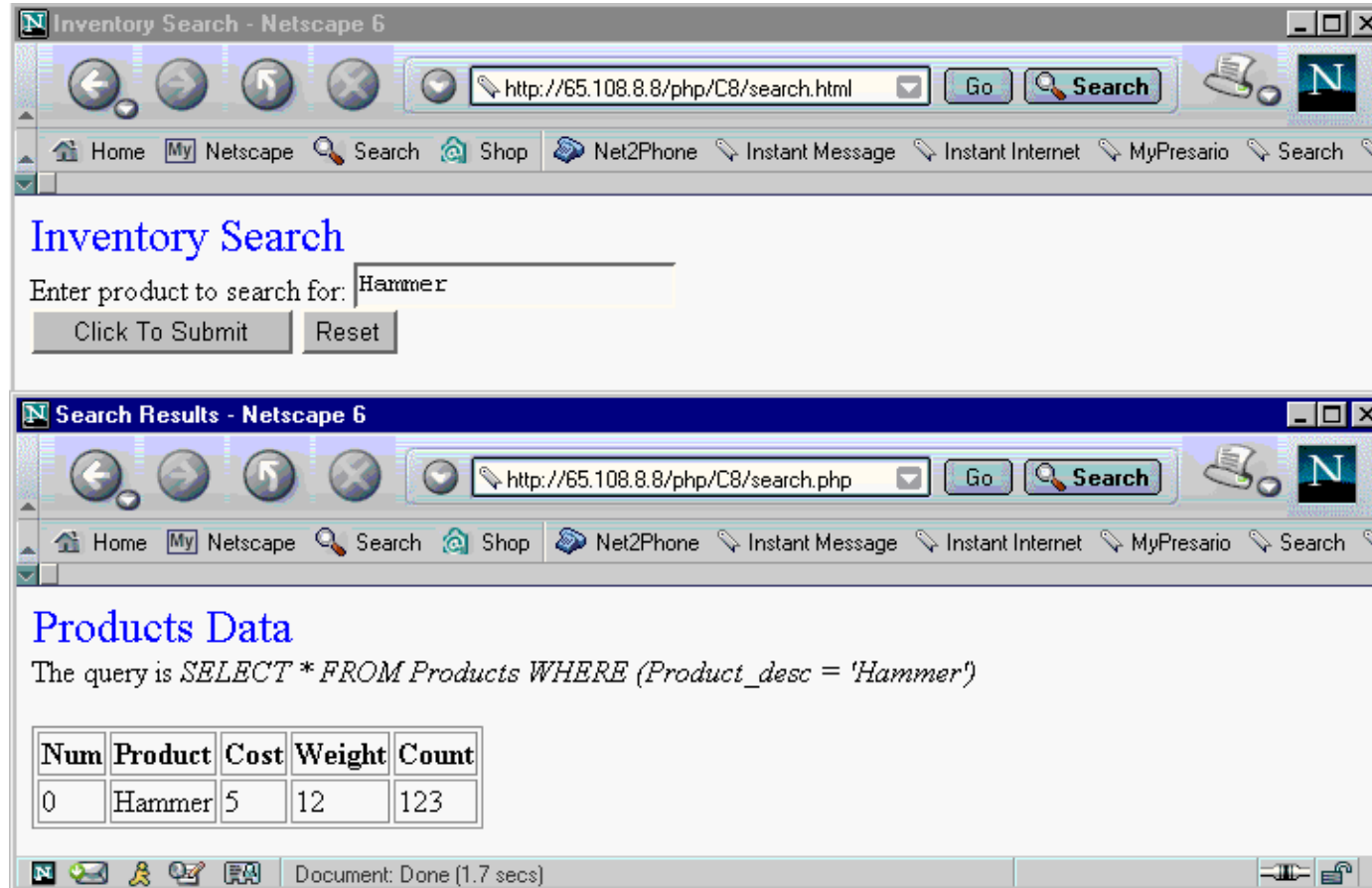
PHP Source

```
1. <html><head><title>Search Results</title></head><body>
2. <?php
3. $host= 'localhost';
4. $user = 'phppgm';
5. $passwd = 'mypasswd';
6. $database = 'phppgm';
7. $connect = mysqli_connect($host, $user, $passwd, $database);
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>";
14. $results_id = mysqli_query($connect, $query);
```


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 = mysqli_fetch_row($results_id)) {
19.         print '<tr>';
20.         foreach ($row as $field) {
21.             print "<td>$field</td> ";
22.         }
23.         print '</tr>';
24.     }
25. mysqli_free_result($results_id);
25. } else { die ("query=$Query Failed"); }
26. mysqli_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:

```
UPDATE Table_name  
SET col1=chng_express1,col2=chng_express2, ...  
WHERE test_expression
```

Specify the name of the table to update.

Optionally specify a WHERE clause and test expression.

Specify one or more table column to receive the results of an expression. Optionally specify a WHERE

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. $connect = mysqli_connect($host, $user, $passwd, $database);
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";
```


A Full Example (2)

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

A Full Example (3)

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

Would output the following:

The image shows two Netscape 6 browser windows. The top window, titled 'Inventory Management - Netscape 6', displays a form for selecting a product. The form has three radio buttons: 'Hammer' (selected), 'Screwdriver', and 'Wrench'. Below the buttons are 'Click To Submit' and 'Reset' buttons. The text 'The query is *SELECT * from Products*' is shown above a table. The table has columns: Num, Product, Cost, Weight, and Count. The data rows are: (2, Wrench, 2, 1, 150), (1, Screwdriver, 3, 2, 144), and (0, Hammer, 5, 12, 123).

The bottom window, titled 'Product Update Results - Netscape 6', displays the text 'Update Results for Table Products' and 'The query is *UPDATE Products SET Numb = Numb-1 WHERE (Product_desc = 'Hammer')*'. Below this is a table with the same columns as the first window. The data rows are: (2, Wrench, 2, 1, 150), (1, Screwdriver, 3, 2, 144), and (0, Hammer, 5, 12, 122).

Num	Product	Cost	Weight	Count
2	Wrench	2	1	150
1	Screwdriver	3	2	144
0	Hammer	5	12	123

Num	Product	Cost	Weight	Count
2	Wrench	2	1	150
1	Screwdriver	3	2	144
0	Hammer	5	12	122

Question?

