Previously

In Unit 3 we saw how to write a web page (GrahamsComputers.html) that contained a form allowing a user to order computer hardware. The order details were passed to another file (ProcessOrder.php) which displayed details of the order, and saved the order to a file, in that case a text file. These were the relevant lines of code (taken from ProcessOrder.php):

```
$outputstring=$date."\t".$cpuqty." cpus\t".$vduqty."vdus\t".$printerqty."printers\t".
$total_2."\t".$street."\t".$suburb."\t".$state."\t".$pcode."\n";
$orderfile = fopen("orders.txt","a");                    // open the file – the new record will be appended
        if(!($orderfile))                                // check the file exists
        {
                echo("sorry file not found");
                exit;
        }
   fputs($orderfile, "$outputstring");                   // write the string
   fclose($orderfile);                                   // close the file
```

Here we are setting up a primitive sort of database. Primitive because it's a flat file database with records separated by a carriage return (new line), and fields separated by a tab character. Also, the details of the order are stored (mixed) with the date and delivery address.
Nowadays, such data should be stored using a relational database management system (RDBMS) such as MS Access, MySQL or Oracle. A RDBMS (in our case, MySQL) is a program that can store large amounts of information in an organized format that is easily accessible using scripting languages like PHP. With a RDBMS, data is stored in tables. A table consists of rows and columns. Later we will use 3 tables viz Customers, Stock and Orders to store details of the orders made to Grahams Computers. Using the appropriate PHP commands we will be able to access data in any one of the tables, or data from a combination of the tables eg which customer ordered what item of stock, and when they ordered it.
Before we do that we will commence by setting up a MySQL database consisting of a single table. We will then use a series of PHP programs to interact with this database.

Introduction to MySQL

For many people, the main reason for learning a scripting language like PHP is because of the interaction with databases it can offer. In this tutorial we will see how to use PHP and the MySQL database to store information on the web and include it into your website. Before you read this tutorial you should have at least a basic knowledge of how to use PHP.

'On the Web today, content is king. After you've mastered HTML and learned a few neat tricks in JavaScript and Dynamic HTML, you can probably build a pretty impressive-looking Web site. But then comes the time to fill that fancy page layout with some real information. Any site that successfully attracts repeat visitors has to have fresh and constantly updated content. In the world of traditional site building, that means HTML files--and lots of 'em.
The problem is that, more often than not, the people providing the content for a site are not the same people handling its design. Often the content provider doesn't even know HTML. How, then, is the content to get from the provider onto the Web site? Not every company can afford to staff a full-time Webmaster, and most Webmasters have better things to do than copying Word files into HTML templates anyway.
Maintenance of a content-driven site can be a real pain, too. Many sites appear locked into a dry, outdated design because rewriting those hundreds of HTML files to reflect a new design would take forever. Server-side includes (SSI's) can help alleviate the burden a little, but you still end up with hundreds of files that need to be maintained should you wish to make a fundamental change to your site.
The solution to these headaches is database-driven site design. By achieving complete separation between your site's design and the content you are looking to present, you can work with each without disturbing the other. Instead of writing an HTML file for every page of your site, you only need to write a page for each kind of information you want to be able to present. Instead of endlessly pasting new content into your tired page layouts, create a simple content management system that allows the writers to post new content themselves without a lick of HTML!'[1]

Why Would I Want A Database?[2]

It is actually surprising how useful a database can be when used with a website. There are a huge variety of things you can do when you interact the two, from displaying simple lists to running a complete website from a database. Some examples of PHP and MySQL being used together are:

- Banner Rotation. On this site, where each banner is, a PHP script is called. This opens a database and picks a random banner from it to show the visitor. It also counts the number of times the banner has been viewed and could, with a few changes, track clicks too. To add, change or edit the banners all I have to do is change the database and the script will pick the correct banners for all the pages on the site.
- Forums. Hundreds of forums (message boards) on the internet are run using PHP and MySQL. These are much more efficient than other systems that create a page for each message and offer a wide variety of options. All the pages in the forum can be updated by changing one script.
- Databases. One quite obvious example is sites which get all there information from a database. For example Script Avenue is run by a few scripts, which gain all their information from a large database. All the different script categories can be accessed in one script by just changing the URL to access a different part of the database.
- Websites. If you have a large website and you want to change the design it can take a very long time to update and upload all the pages. With PHP and MySQL your whole website could be just one or two PHP scripts. These would access a MySQL database to get the information for the pages. To update the website's design you would just have to change one page.

What Do I Need?

You only really need three things to run PHP scripts which access MySQL databases. Firstly, you will, of course, need a webserver. This can either be on a computer of your own or on a web host. Any web server software should work with PHP and MySQL but the best to use is Apache, which is free.

PHP also needs to be installed on the server.

Finally, you will also require MySQL. This is the actual database software. You can also use most other types of database (SQL, Oracle etc.) but as this is a PHP/MySQL tutorial I will deal just now with the MySQL database (although the commands used here will also work with SQL databases).

If you cannot install (or your web host won't allow) PHP and MySQL you can still use another web host. Freedom2Surf are a free (banner supported) web host and support PHP and have MySQL installed.

Creating A Table

Before you can do anything with your database, you must create a table. A table is a section of the database for storing related information. In a table you will set up the different fields which will be used in that table. Because of this construction, nearly all of a site's database needs can be satisfied using just one database.

Creating a table in PHPMyAdmin is simple, just type the name, select the number of fields and click the button. You will then be taken to a setup screen where you must create the fields for the database. If you are using a PHP script to create your database, the whole creation and setup will be done in one command.

Fields

There are a wide variety of fields and attributes available in MySQL and only a few will be covered here:

Field Type	Description
TINYINT	Small Integer Number
SMALLINT	Small Integer Number
MEDIUMINT	Integer Number
INT	Integer Number
VARCHAR	Text (maximum 256 characters)
TEXT	Text

These are just a few of the fields which are available. A search on the internet will provide lists of all the field types allowed.

The Jokes Database

In our first example we will use a PHP script to look in a database for a joke that we want to appear on our Web site. A different joke will appear each day. In this example, the jokes will be stored entirely in the database. The advantage of this would be twofold. First, instead of having to write a different HTML file for each day, we write a single PHP script designed to fetch the joke out of the database and display it. Second, to add a new joke to our Web site would just be a matter of adding the joke to the database. The PHP code would take care of the rest by automatically displaying the new joke on the allocated date when fetched from the database.
First we need to look at how data is stored in a database. A database is composed of one or more 'tables', each of which contains a list of 'things'. For our joke database, we would probably start with a table called "jokes" which would contain a list of jokes. Each table in a database has one or more columns, or fields. Each column holds a certain piece of information about each "thing" in the database. Returning to our example, our "jokes" table might have columns for the text of the jokes and the dates the jokes were added to the database. Each joke that we stored in this table would then be said to be a 'row' in the table. To see where all this terminology comes from, have a look at what this table actually looks like:

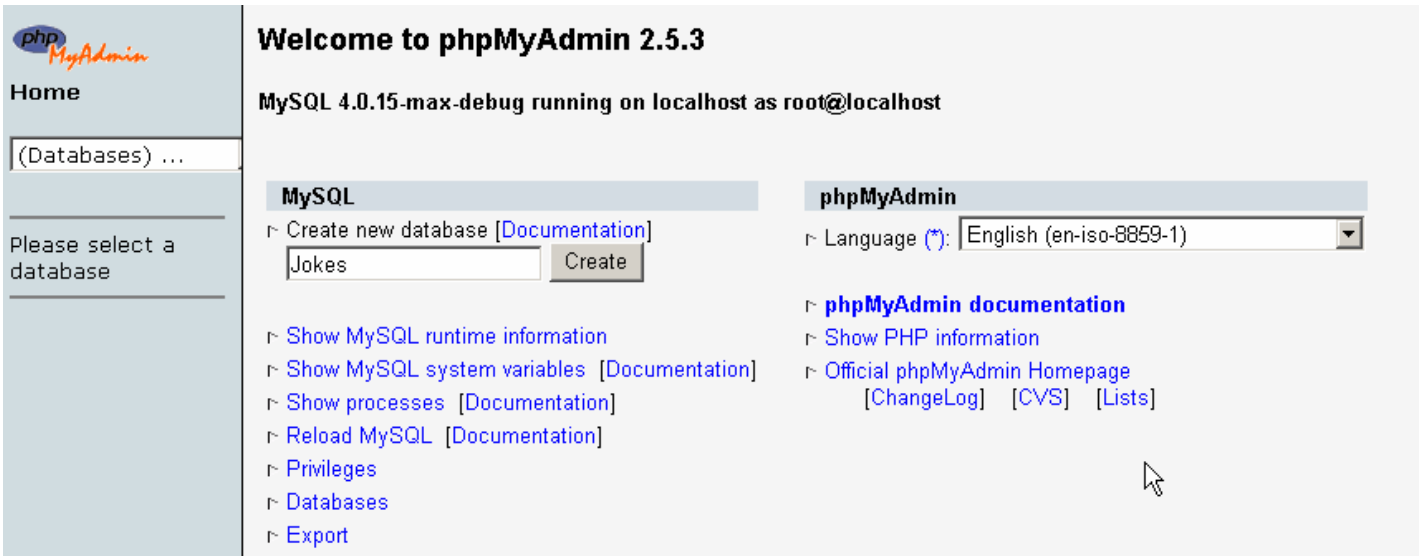
	column	column	column
row	ID	JokeText	JokeDate
row	1	A horse walks into a bar. The ..	2004-11-24
row	2	How do you make …	2004-11-25

Notice that, in addition to columns for the joke text ("JokeText") and the date of the joke ("JokeDate"), there is a column named "ID". The function of this column is to assign a unique number to each joke so we have an easy way to refer to them and to keep track of which joke is which.
So to review, the above is a three-column table with two rows (or entries). Each row in the table contains a joke's ID, its text, and the date the joke is to be displayed. With this basic terminology under our belts, we're ready to get started using MySQL.

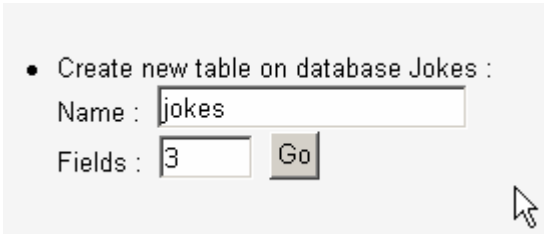
Setting up a database

After starting Foxserv and selecting Visit localhost, select phpmyadmin

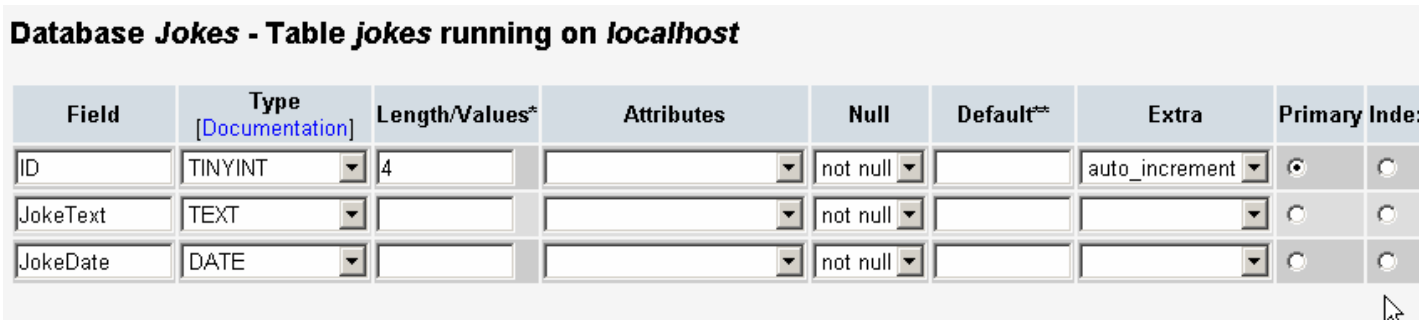
You should see a screen similar to this:



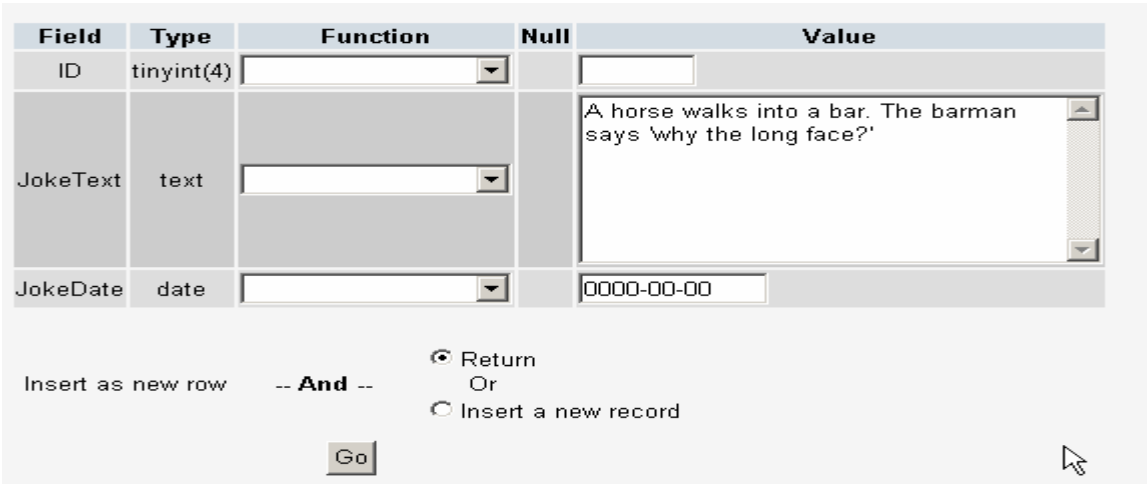
Create a new database called Jokes, and within that database a table called jokes with 3 fields.



The 3 fields should be set out as follows:



Click on Insert and insert a joke with the current date. Do not enter a value for ID.



We will now write the HTML to create a web page similar to that below. In our example only jokes that have a date equal to current date will be displayed.

The script for the page above is

```
<! ------------------------------------------------------------------------------------------------------------------------------------------------------
Program web_page_with_joke.php written by Graham Travers July 2004.
This program illustrates how PHP can be used to access a MySQL database. The web page
content is generated from fields within the database. The database is called Jokes and consists
of one table called jokes which has 3 fields. The fields are ID, JokeText, and JokeDate.
The current date is compared to the date in the jokes table and a joke/s displayed if a match is found.
This program is an adaptation of an idea from http://dev.mysql.com/tech-resources/articles/ddws/
------------------------------------------------------------------------------------------------------------------------------------------------------>
<html>
<head>
<title>My joke page</title>
</head>
<body>
<H2><CENTER>Using a database to generate web page content</CENTER></H2><BR>
<?php
        $date=date("Y-m-d");
        echo "<B>Today's date is ".$date."</B><P>";
        $db = mysql_connect("localhost", "root");           //connect to the database server
        if (!$db)
        {
                echo "Error: Could not connect to database.  Please try again later.";
                exit;
        }
        mysql_select_db("Jokes");         //select the database
        $query="SELECT * FROM jokes where JokeDate='$date'";
        $result = mysql_query($query); //set up the query
        $num_results=mysql_num_rows($result);   //find number of results
        if ($num_results!=null){       //check that number of results is not null
                for ($i=0;$i<=$num_results;$i++){   //loop for all jokes that match date
                $row = mysql_fetch_row($result);   //fetch the row which is an array
                echo $row[1]."<BR><BR>";      //print the joke - the second column
                }   //end for loop
        }  //end if statement
        else
        {
                echo "Sorry, no joke today!";
        }
?>
</body>
</html>
```

Click here to run this program.

Adding jokes to our database

We could use phpMyAdmin to add jokes to our database, but this would be 'fiddly' for someone other than the webmaster. Let's create a web page that will do the job (actually 2 web pages are needed – the first passes data to the second).

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name new_jike.php written by Graham Travers July 2004.
This program adds a joke to the jokes table which is part of the Jokes database.
This program is an adaptation of an idea from http://dev.mysql.com/tech-resources/articles/ddws/
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Jokes Database - New Entry</title>
</head>
<body>
<h1>Jokes Database - New Entry</h1>
<form action="insert_joke.php" method="post">
<table border=0>
     <tr><td>JokeText</td><td> <textarea name=JokeText rows=3 cols=20></textarea></td></tr>
     <tr><td>Joke Date</td><td> <input type=text name=JokeDate maxlength=10 size=10><br></td></tr>
     <tr><td colspan=2><input type=submit value="Insert"></td></tr>
</table>
</form>
</body>
</html>
```

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name insert_joke.php written by Graham Travers July 2004.
This program takes values passed from the file new_joke.php and will add a new joke to the Jokes database.
This program is an adaptation of an idea from http://dev.mysql.com/tech-resources/articles/ddws/
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
 <title>New Joke Entry</title>
</head>
<body>
<h1>New Joke Entry</h1>
<?
     $JokeText=$_POST['JokeText'];$JokeDate=$_POST['JokeDate'];
     if (!$JokeText || !$JokeDate) //check that the necessary data has been passed from the calling program
     {
          echo "You have not entered all the required details.<br>.Please go back and try again.";
          exit;
      }
     $JokeText = addslashes($JokeText); //we need to addslashes to any user (string) input being saved to a database
     $JokeDate = addslashes($JokeDate);
     $db = mysql_connect("localhost", "root");
     //we need to tell the program the name of the host where the MySQL server is running
     mysql_select_db("Jokes");//we need to select the database we plan to use
     if (!$db)
     {
          echo "Error: Could not connect to database.  Please try again later.";
           exit;
     }
     $query = "insert into jokes values ('NULL','".$JokeText."', '".$JokeDate."')";
     $result = mysql_query($query);
     if ($result)
     echo  mysql_affected_rows()." joke/s inserted into database."; //informs the user of the number of rows affected
?>
</body>
</html>
```

Click here to run this program

Adding fields

Use phpMyAdmin to edit your Jokes database, adding two new fields after the JokeText field. Add a field with the name Proverb (type text length 40) and a field called foto (type char length 20). Add a proverb for the current day and the name of the image you want displayed that day eg cartoon1.jpg. Then edit your web page so that the joke, proverb, and image are all displayed. You will need a folder called images in your php folder which holds your cartoons. The program below produces the web page which follows.

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name web_page_with_joke2.php written by Graham Travers July 2004.
This program illustrates how PHP can be used to access a MySQL database. The web page
content is generated from fields within the database. The database is called Jokes and consists
of one table called jokes which has 5 fields. The fields are ID, JokeText, Proverb and JokeDate.
The current date is compared to the date in the jokes table and a field contents displayed if a match is found.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>My web page</title>
</head>
<body>
<H2><CENTER>Welcome to my web page</CENTER></H2>
<?php
    $date=date("Y-m-d");
    $db = mysql_connect("localhost", "root"); //connect to the database server
    if (!$db)
    {
        echo "Error: Could not connect to database.  Please try again later.";
        exit;
    }
    mysql_select_db("Jokes");   //select the database
    $result = mysql_query("SELECT * FROM jokes where JokeDate='$date'");     //set up the query
    if ($result!=null) {     //fetch the row & check to see that it is not null
        $row= mysql_fetch_row($result);
        echo "<B>Today's joke is:</B><BR>";
        echo $row[1]."<P>";       //print the joke
        echo "<B>The proverb for the day is:</B><BR>";
        echo $row[2]."<P>";       //print the proverb
        $foto=$row[3];         //$row[3] holds the path of the image
        echo "<CENTER><img src=images/$foto border=0></CENTER>"; //display the image
    }
    else
    {
        echo "Sorry, no joke, proverb or cartoon today!";
    }
?>
</body>
</html>
```

Welcome to my web page

Today's joke is:
A grasshopper goes into a bar. The barman says 'we've got a drink named after you'. The grasshopper says 'what? Eric'

The proverb for the day is:
All that glitters is not gold.

The Contacts Database

The contacts database will contain all the contact information for the people you enter The information will be able to be edited and viewed on the internet. The following fields will be used in the database:

Name	Type	Length	Description
contact_ID	INT	6	A unique identifier for each record
surname	CHAR	15	The person's last name
first_name	CHAR	15	The person's first name
street	CHAR	25	The person's street number & name
suburb	CHAR	25	The person's suburb
postcode	INT		The person's post code

The contactID field will also be set as PRIMARY, and will be set to auto_increment (found under Extra in PHPMyAdmin). The reason for this is that this will be the field identifier (primary and index) and so must be unique. The auto increment setting means that whenever you add a record, as long as you don't specify an id, it will be given the next number.

Create a new database called Contacts.

Create a new table in this database called Names consisting of 6 fields:

Your Names table should be constructed exactly as follows:

Database *Contacts* - table *Names* running on *localhost*

Field	Type	Length/Values*	Attributes	Null	Default	Extra	Primary
contact_ID	INT	10		not null		auto_increment	☑
surname	CHAR	15		not null			☐
first_name	CHAR	15		not null			☐
street	CHAR	25		not null			☐
suburb	CHAR	20		not null			☐
postode	INT			not null			☐

Your options should appear as follows. Select Insert

table Names has been created.

SQL-query : [Edit]

```
CREATE TABLE `Names` (
`contact_ID` INT(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
`surname` CHAR(15) NOT NULL,
`first_name` CHAR(15) NOT NULL,
`street` CHAR(25) NOT NULL,
`suburb` CHAR(20) NOT NULL,
`postode` INT NOT NULL
);
```

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	
☐	contact_ID	int(10)		No		auto_increment	Chan
☐	surname	char(15)		No			Chan
☐	first_name	char(15)		No			Chan
☐	street	char(25)		No			Chan
☐	suburb	char(20)		No			Chan
☐	postode	int(11)		No	0		Chan

With selected: Change *Or* Drop

Insert 2 records into your table. For future purposes have the same postcode for both contacts. Do not enter a value for the contact_ID as it is auto_increment

Field	Type	Function	Null	Value
contact_ID	int(10)			
surname	char(15)			Finster
first_name	char(15)			Mervyn
street	char(25)			21 Geelong Rd
suburb	char(20)			Grovedale
postode	int(11)			326

Now we will write a program in PHP to read the Names table and output the results (the records or rows).

Reading from a database

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File name read_contacts.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program passes 2 values to the file search_results_contacts.php which will search the Contacts database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Contacts - Read List of Contacts</title>
</head>
<body>
<h1>List of current contacts</h1>
<?php
        $db = mysql_connect("localhost", "root");   //connect to the database
        mysql_select_db("Contacts");                                    //select the database
        $query = "SELECT * FROM names order by contact_ID";    //set the query as a variable
        $result = mysql_query($query);              //perform the query and store the details of the result
        $num_results=mysql_num_rows($result); //establish the number of matching results
        if($num_results!=null){   //check that there are results
                echo "<table border=1>\n";
                echo "<tr><td>CONTACT ID</td><td>SURNAME</td><td>FIRST
NAME</td><td>STREET</td><td>SUBURB</td><td>POSTCODE</td></tr>\n";
        for ($i=0; $i<$num_results;$i++){             //begin loop
                $row=mysql_fetch_row($result);   //fetch a row
        echo"<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row[3]</td><td>$row[4]</td><td>$row[5]</td
></tr>\n";          //output the 6 fields
        }  //end for loop
                echo "</table>\n";
        } //end if statement
        else
        {
                echo "Sorry, no records were found!";
        }
?>
</body>
</html>
```

Click here to run this program

When run, the output should appear something like this:

CONTACT ID	SURNAME	FIRST NAME	STREET	SUBURB	POSTCODE
1	Nerk	Fred	21 High St	Belmont	3216
2	Finster	Mervyn	21 Geelong Rd	Grovedale	3216

Next we will write programs to add a contact to the Names table, search the Names table, and delete a contact from the Names table.

Adding rows (records) to a database table

You will need the following two programs. The first (I've called it new_contact.php) calls the second program (called insert_contact.php). The first program is all HTML and uses text boxes to obtain input. This data is then passed to the second program.

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File new_contact.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program adds a contact to the names table which is part of the Contacts database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Contacts - New Contact Entry</title>
</head>
<body>
<h1>Contacts - New Contact Entry</h1>
<form action="insert_contact.php" method="post">
<table border=0>
    <tr><td>Surname</td><td> <input type=text name=surname maxlength=20 size=30><br></td></tr>
    <tr><td>First Name</td><td> <input type=text name=first_name maxlength=20 size=30><br></td></tr>
    <tr><td>Address</td><td> <input type=text name=street maxlength=30 size=30><br></td></tr>
    <tr><td>Suburb</td><td><input type=text name=suburb maxlength=20 size=20><br></td></tr>
    <tr><td>Postcode</td><td><input type=text name=postcode maxlength=4 size=4><br></td></tr>
    <tr><td colspan=2><input type=submit value="Insert"></td></tr>
</table>
</form>
</body>
</html>
```

This program is called by the previous program.

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File name insert_contact.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program takes values passed from the file insert_contact.php and will add a new contact to the Contacts database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
 <title>New Contact Entry</title>
</head>
<body>
<h1>New Contact Entry</h1>
<?
     $surname=$_POST['surname'];$first_name=$_POST['first_name'];street=$_POST['street'];
     $suburb=$_POST['suburb']; $postcode=$_POST['postcode'];
     if (!$surname || !$first_name || !$street|| !$suburb|| !$postcode)
     //check that the necessary data has been passed from the calling program
     {
          echo "You have not entered all the required details.<br>.Please go back and try again.";
          exit;
      }
     $surname = addslashes($surname); //we need to addslashes to any user (string) input being saved to a database
     $first_name = addslashes($first_name);
     $street = addslashes($street);
     $suburb = addslashes($suburb);
     $postcode = addslashes($postcode);
     $db = mysql_connect("localhost", "root");
     //we need to tell the program the name of the host where the MySQL server is running
     mysql_select_db("contacts");               //we need to select the database we plan to use
     if (!$db)
     {
          echo "Error: Could not connect to database.  Please try again later.";
           exit;
     }
     $query = "insert into names values ('NULL','".$surname."', '".$first_name."','".$street."', '".$suburb."','".$postcode."')";
     $result = mysql_query($query);
     if ($result)
     echo  mysql_affected_rows()." contact inserted into database."; //informs the user of the number of rows affected
?>
</body>
</html>
```

Use the above programs to add two more contacts to the Names table. Check that the new records were successfully added by reading the database.

Searching a database table

The following program (called search_contacts.php) calls the program following it (called search_contacts_results.php).

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name search_contacts.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program passes 2 values to the file search_ contacts_results.php which will search the Contacts database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Contacts Name Search</title>
</head>
<body>
<h1>Contacts Name Search</h1>
<form action="search_ contacts_results.php" method="post">
Choose Search Type:<br>
<select name="searchtype"> //searchtype is the field that will be searched
       <option value="contact_ID">Contact ID
       <option value="surname">Surname
       <option value="first_name">First name
       <option value="street">Street
       <option value="suburb">Suburb
      <option value="postcode">Postcode
 </select>
 <br>
 Enter Search Term:<br>
<input name="searchterm" type=text>//searchterm is the text that will be searched for
<br>
<input type=submit value="Search">
</form>
</body>
</html>
```

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File name search_contacts_results.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program takes 2 values to it from the file search_contacts.php which will search the Contacts database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
 <title>Contacts Database Search Results</title>
</head>
<body>
<h1>Contacts Database Search Results</h1>
<?
     $searchtype=$_POST['searchtype']; $searchterm=$_POST['searchterm'];
     if (!$searchtype || !$searchterm)
     //searchtype is the name of the field we're going to search and searchterm the data we're searching for
     {
          echo "You have not entered search details.  Please go back and try again.";
          exit;
     }
     $searchtype = addslashes($searchtype);//data inserted into a database has  hadd slashes added at each end
     $searchterm = addslashes($searchterm);
     $db = mysql_connect("localhost", "root");//connect to where the database is located
     if (!$db)
     {
          echo "Error: Could not connect to database.  Please try again later.";
          exit;
     }
  mysql_select_db("Contacts");   //select the database
  $query = "select * from names where ".$searchtype." like '%".$searchterm."%'"; //using like is 'safer' than using =
  $result = mysql_query($query); //the result variable has a number of parts we can access
  $num_results = mysql_num_rows($result); //the number of rows affected by our search
  echo "<p>Number of contacts found: ".$num_results."</p>";
  for ($i=0; $i <$num_results; $i++)    //iterate through a loop writing each row affected by the search
  {
    $row = mysql_fetch_array($result);//this line gets the details from the row affected
    echo "<p><strong>".($i+1).". Contact ID: ";
    echo stripslashes($row["memberID"]);//we need to strip slashes from the data prior to displayiing
    echo "</strong><br>Surname: ";
    echo stripslashes($row["surname"]);
    echo "</strong><br>First name: ";
    echo stripslashes($row["first_name"]);
    echo "<br>Street: ";
    echo stripslashes($row["street"]);
    echo "<br>Suburb: ";
    echo stripslashes($row["suburb"]);
    echo "<br>Postcode: ";
    echo stripslashes($row["postcode"]);
      echo "</p>";
  }
?>
</body>
</html>
```

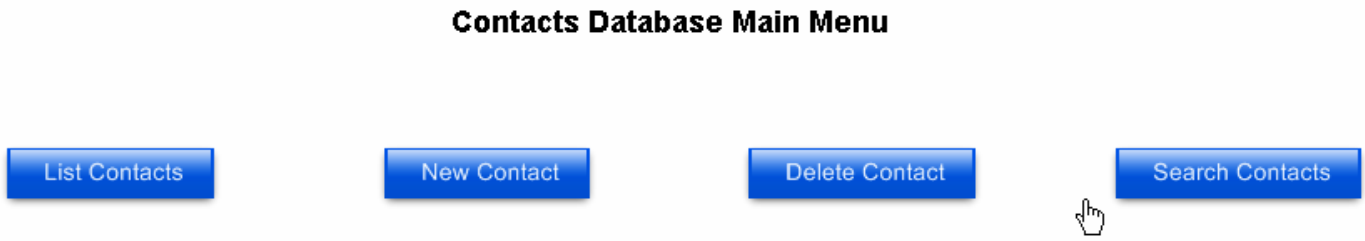
Deleting a contact

Similar to the preceding programs, a program I've called delete_contact.php (all HTML) requests the contact_ID to be deleted and passes this information to the program that follows it (called delete_contact_result.php)

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name delete_contact.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program passes the contacID to the file delete_contact_results.php which will delete a contact from the Contacts
database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Delete a Contact</title>
</head>
<body>
<h1>Delete a contact</h1>
 <form action="remove_contact.php" method="post">
 <table border=0>
      <tr><td>Contact ID</td><td> <input type=text name=contact_ID maxlength=20 size=30><br></td></tr>
      <tr><td colspan=2><input type=submit value="Delete"></td></tr>
</table>
</form>
</body>
</html>

<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name remove_contact.php written by Graham Travers June 2004.
Adapted from the text - PHP and MySQL Web Development by Welling and Thomson.
This program passes the contacID to the file delete_contact_results.php which will delete a contact from the Contacts
database.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
  <title>Contacts - Delete Contact </title>
</head>
<body>
<h1>Contact Deletion</h1>
<?
 $contact_ID=$_POST['contact_ID'];
 $db = mysql_connect("localhost", "root");
mysql_select_db("contacts");
if (!$db)
  {
     echo "Error: Could not connect to database.  Please try again later.";
     exit;
  }
  mysql_select_db("contacts");
  $query = "delete from name where contact_ID=$contact_ID";
  $result = mysql_query($query);
  if ($result)
      echo  mysql_affected_rows()." name deleted from database.";
?>
</body>
```

Main menu for the Contacts database:

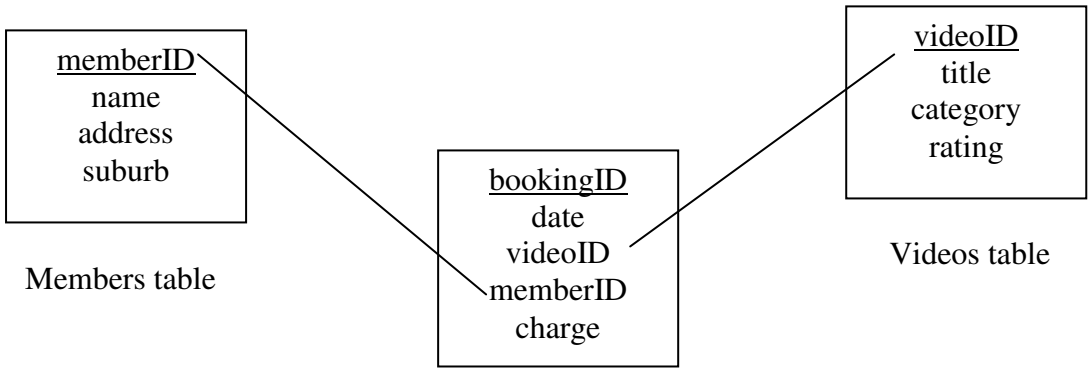


Click here to run the menu above

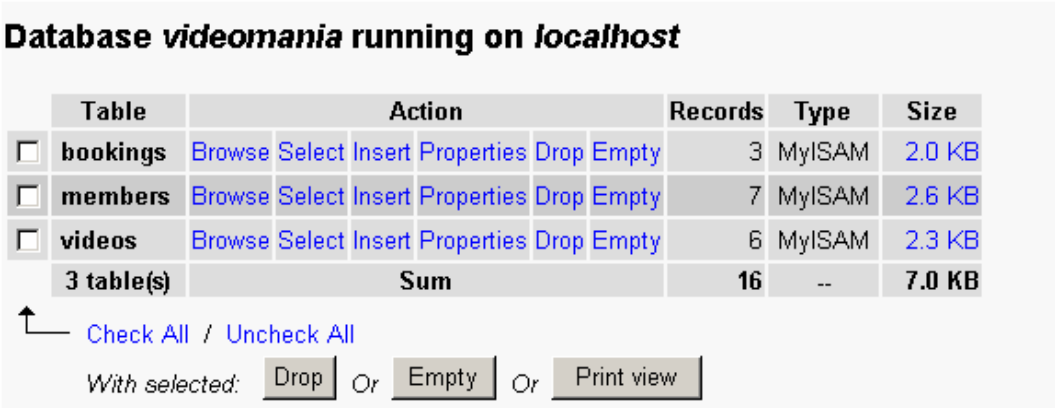
A database with multiple tables

Case Study - VIDEOMANIA

Videomania have recently bought out Graham's Video Hire. Graham's Video Hire had stores in Geelong, Belmont and Highton, and used an database (MSAccess) to keep track of customers, videos, and bookings. The new manager wishes to switch to MySQL as a database for security reasons and because, as a database, it is more 'web-friendly'. The MySQL database has been set up with 3 tables – members, videos and bookings. The schema is shown below, with the key fields (primary fields) underlined. The straight lines indicate links.



The following screen dumps show how this MySQL database was set up.



Database *videomania* - table *members* running on *localhost*

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	Action					
☐	memberID	int(10)	UNSIGNED	No		auto_increment	Change	Drop	Primary	Index	Unique	Fulltext
☐	name	char(30)		No			Change	Drop	Primary	Index	Unique	Fulltext
☐	address	char(40)		No			Change	Drop	Primary	Index	Unique	Fulltext
☐	suburb	char(20)		No			Change	Drop	Primary	Index	Unique	Fulltext

With selected: Change Or Drop

Database *videomania* - table *videos* running on *localhost*

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	Action					
☐	videoID	int(10)	UNSIGNED	No		auto_increment	Change	Drop	Primary	Index	Unique	Fulltext
☐	title	char(20)		No			Change	Drop	Primary	Index	Unique	Fulltext
☐	category	char(20)		No			Change	Drop	Primary	Index	Unique	Fulltext
☐	rating	char(10)		No			Change	Drop	Primary	Index	Unique	Fulltext

With selected: Change Or Drop

Database *videomania* - table *bookings* running on *localhost*

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	Action					
☐	bookingID	int(10)		No		auto_increment	Change	Drop	Primary	Index	Unique	Fulltext
☐	date	date		No	0000-00-00		Change	Drop	Primary	Index	Unique	Fulltext
☐	videoID	int(10)		No	0		Change	Drop	Primary	Index	Unique	Fulltext
☐	memberID	int(10)		No	0		Change	Drop	Primary	Index	Unique	Fulltext
☐	charge	float(6,2)		No	0.00		Change	Drop	Primary	Index	Unique	Fulltext

With selected: Change Or Drop

A main menu has been written:

Videomania Staff Main Menu

New Member | New Video | New Booking

Search Members | Search Videos | Search Bookings

List Members | List Videos | List Bookings

Delete Member | Delete Video | Delete Booking

Click here to run this menu

Programs have been written in PHP to add a new member, search the member table, read and display the member table, and delete a member.
Your task is to write the remaining 8 programs. The existing programs are listed below.

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File read_members.php written by Graham Travers June 2003.
Adapted from Welling and Thomson text - PHP and MySQL Web Development.
This program reads the members table. The videomania database was created in MySQL and has 3 tables
- members, videos, bookings. This program is an adaptation of the Book-O-Rama example in chapter 10 of the text
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Videomania - Read Members</title>
</head>
<body>
<?php
      $db = mysql_connect("localhost", "root");//connect to the database
      mysql_select_db("videomania");//select the database
      $query = “SELECT * FROM members order by memberID”;  //set up the query
      $result = mysql_query($query);
      //perform the query and store the details of the result
      $num_results=mysql_num_rows($result); //establish the number of matching results
       if($num_results!=null){   //check that there are results
            echo "<table border=1>\n";
            echo "<table border=1>\n";//print the table headings
            echo "<tr><td>MEMBER ID</td><td>NAME</td><td>ADDRESS</td><td>SUBURB</td></tr>\n";
      for($i=0;$i<$num_results;i++){
            $row = mysql_fetch_array($result)) //fetch the row
            echo”<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>row[3]</td></tr>\n";
            } //end for loop
      echo "</table>\n";
      } //end if statement
      else
      {
            echo "Sorry, no records were found!";
      }
?>
</body>
</html>

<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File new_member.php written by Graham Travers
Adapted from Welling and Thomson text - PHP and MySQL Web Development.
June 2003. This program will calls the insert_member.php file. The videomania database was created in MySQL and
has 3 tables
- members, videos, bookings. This program is an adaptation of the Book-O-Rama example in chapter 10 of the text
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Videomania - New Member Entry</title>
</head>
<body>
<h1>Videomania - New Member Entry</h1>
<form action="insert_member.php" method="post">
<table border=0>
      <tr><td>Name</td><td> <input type=text name=name maxlength=20 size=30><br></td></tr>
      <tr><td>Address</td><td> <input type=text name=address maxlength=30 size=30><br></td></tr>
      <tr><td>Suburb</td><td><input type=text name=suburb maxlength=20 size=20><br></td></tr>
      <tr><td colspan=2><input type=submit value="Insert"></td></tr>
</table>
</form>
</body>
</html>
```

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File insert_member.php written by Graham Travers June 2003.
Adapted from Welling and Thomson text - PHP and MySQL Web Development.
This program will insert records into the members table which is part of the db called videomania.
This db was created in MySQL and has 3 tables- members, videos, bookings.
This program is an adaptation of the Book-O-Rama eg from the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
  <title>Videomania Member Entry </title>
</head>
<body>
<h1>Videomania Member Entry</h1>
<?
      $name=$_POST['name']; $address=$_POST['address']; $suburb=$_POST['suburb'];
      if (!$name || !$address || !$suburb)    //check that the necessary data has been passed rom the calling program
       {
            echo "You have not entered all the required details.<br>" ."Please go back and try again.";
           exit;
       }
      $name = addslashes($name); //we need to addslashes to any user (string) input being saved to a database
      $address = addslashes($address);
      $suburb = addslashes($suburb);
      $db = mysql_connect("localhost", "root");
      //we need to tell the program the name of the host where the MySQL server is running
      mysql_select_db("videomania");//we need to select the database we plan to use
       if (!$db)
      {
           echo "Error: Could not connect to database.  Please try again later.";
           exit;
      }
      $query = "insert into members values ('NULL','".$name."', '".$address."', '".$suburb."')";
      $result = mysql_query($query);
      if ($result)
           echo  mysql_affected_rows()." member inserted into database."; //informs the user of the number of rows
affected
?>
</body>
</html>

<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File search_members.php written by Graham Travers June 2003.
Adapted from Welling and Thomson text - PHP and MySQL Web Development.
This program will call the search_members_results.php program. The videomania database was created in MySQL and
has 3 tables- members, videos, bookings. This program is an adaptation of the Book-O-Rama example in chapter 10 of
the text
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Videomania Member Search</title>
</head>
<body>
<h1>Videomania Member Search</h1>
<form action="search_members_results.php" method="post">
Choose Search Type:<br>
<select name="searchtype"> //searchtype is the field that will be searched
      <option value="memberID">Member ID
      <option value="name">Name
      <option value="address">Address
       <option value="suburb">Suburb
</select><br>
Enter Search Term:<br>
<input name="searchterm" type=text><!-searchterm is the text that will be searched for-><br>
<input type=submit value="Search">
</form>
</body></html>
```

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File search_members_results.php written by Graham Travers June 2003.
Adapted from Welling and Thomson text - PHP and MySQL Web Development
This program is called by the program search_members.php and will search the member table. The videomania
database was created in MySQL and has 3 tables - members, videos, bookings. This program is an adaptation of the
Book-O-Rama example in chapter 10 of the text
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
  <title>Videomania Search Results</title>
</head>
<body>
<h1>Videomania Search Results</h1>
<?
     $searchtype=$_POST['searchtype']; $searchterm=$_POST['searchterm'];
     if (!$searchtype || !$searchterm)//searchtype is the name of the field we're going to search and searchterm the data
we're searching for
     {
          echo "You have not entered search details.  Please go back and try again.";
          exit;
     }
     $searchtype = addslashes($searchtype);//data added to a database has had slashes added at each end
     $searchterm = addslashes($searchterm);
     $db = mysql_connect("localhost", "root");//connect to where the database is located
     if (!$db)
     {
          echo "Error: Could not connect to database.  Please try again later.";
          exit;
     }
     mysql_select_db("videomania");//select the database
     $query = "select * from members where ".$searchtype." like '%".$searchterm."%'";//using like is 'safer' than using =
     $result = mysql_query($query);//the result variable has a number of parts we can access
     $num_results = mysql_num_rows($result);//the number of rows affected by our search is one part we can access
     echo "<p>Number of customers found: ".$num_results."</p>";
     for ($i=0; $i <$num_results; $i++)//iterate through a loop writing each row affected by the search
     {
          $row = mysql_fetch_array($result);//this line gets the details from the row affected
          echo "<p><strong>".($i+1).". Member ID: ";
          echo stripslashes($row["memberID"]);//we need to strip slashes from the data prior to displayiing
          echo "</strong><br>Name: ";
          echo stripslashes($row["name"]);
          echo "<br>Address: ";
          echo stripslashes($row["address"]);
          echo "<br>Suburb: ";
          echo stripslashes($row["suburb"]);
          echo "</p>";
     }
?>
</body>
</html>
```

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File remove_member.php written by Graham Travers June 2003.
Adapted from Welling and Thomson text - PHP and MySQL Web Development.
This program will calls the delete_member.php file. The videomania database was created in MySQL and has 3 tables
- members, videos, bookings. This program is an adaptation of the Book-O-Rama example in chapter 10 of the text
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Videomania - Delete Member</title>
</head>
<body>
<h1>Videomania - Delete Member</h1>
<form action="delete_member.php" method="post">
<table border=0>
      <tr><td>Member ID</td><td> <input type=text name=memberID maxlength=20 size=30><br></td></tr>
      <tr><td colspan=2><input type=submit value="Delete"></td></tr>
</table>
</form>
</body>
</html>

<! ----------------------------------------------------------------------------------------------------------------------------------------------------------------
File delete.member.php by Graham Travers June 2003.
Adapted from text  PHP and MySQL Web Development by Welling and Thomson.
This program will delete a record from the members table which is part of the db called videomania.
This db was created in MySQL and has 3 tables- members, videos, bookings.
This program is an adaptation of the Book-O-Rama eg from the text.
---------------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
  <title>Videomania - Delete Member </title>
</head>
<body>
<h1>Videomania Member Deletion</h1>
<?
$memberID=$_POST['memberID'];
 $db = mysql_connect("localhost", "root");
  if (!$db)
  {
     echo "Error: Could not connect to database.  Please try again later.";
     exit;
  }
  mysql_select_db("videomania");
  $query = "delete from members where memberID=$memberID";
  $result = mysql_query($query);
  if ($result)
      echo  mysql_affected_rows()." member deleted from database.";
?>
</body>
</html>
```

Searching multiple tables

The following program displays all the bookings in detail. Whilst the bookings table stores only the member ID and video ID, this program locates the corresponding member name from the member table, and the corresponding video title from the videos table.

```
<! ----------------------------------------------------------------------------------------------------------------------------------------------------------
File name read_bookings.php written by Graham Travers, June 2003.
This program is an adaptation of the Book-O-Rama example in chapter 10 of the text 'PHP and MySQL Web Development' by Welling and Thomson
 The videomania database was created in MySQL and has 3 tables- members, videos, bookings.
This program reads the 3 tables and finds matches in the bookings table.
---------------------------------------------------------------------------------------------------------------------------------------------------------->
<html>
<head>
<title>Videomania - List of Bookings</title>
</head>
<body>
<?php
      $db = mysql_connect("localhost", "root");            //connect to the database
      mysql_select_db("videomania");                  //select the database
      $query="SELECT * FROM bookings,members,videos where members.memberID =    bookings.memberID and videos.videoID = bookings.videoID order by date";
     $result = mysql_query($query);
      //perform the query and store the details of the result
      if ($row = mysql_fetch_array($result))     //check to see if $row is null, that is, no values satisfy the query
      {
            echo "<table border=1>\n";
            echo "<tr><td>BOOKING ID</td><td>DATE</td><td>VIDEO ID</td><td>VIDEO TITLE</td><td>MEMBER ID</td><td>MEMBER NAME</td><td>CHARGE</td></tr>\n"; //print heading
      do
      {
      echo "<tr><td>".$row["bookingID"]."</td><td>".$row["date"]."</td><td>".$row["videoID"]."</td><td>".$row["title"]."</td><td>".$row["memberID"]."</td><td>".$row["name"]."</td><td>".$row["charge"]."</td></tr>\n";
      }
      while ($row = mysql_fetch_array($result));
      //$row is an associative array. This loop will continue as long as $row is not null
      echo "</table>\n";
      }
      else
      {
      echo "Sorry, no records were found!";
      }
?>
</body>
</html>
```

The result of running the program above is shown here:

BOOKING ID	DATE	VIDEO ID	VIDEO TITLE	MEMBER ID	MEMBER NAME	CHARGE
4	2003-07-05	5	Sniper	4	Greg Lauder	15.00
3	2003-07-08	3	Goodbye Mr Chips	6	Aristude Schumacher	9.00
11	2004-06-04	6	Dinner Rush	12	Greg Stafford	4.00
12	2004-07-07	9	The Producers	12	Greg Stafford	5.00

Your tasks

1. Create a new database called Grahams_Computers and within this database create 3 tables – Customers, Stock, Orders. You'll need to determine the fields necessary.
2. Write the programs necessary to interact with this database
3. The main menu shown here will be provided.

4. Write programs to delete jokes from the Jokes database

Review[1]

It's worth a brief look back to remind ourselves of the goal we were working toward. We have two powerful, new tools at our disposal: the PHP scripting language, and the MySQL database engine. It's important to understand how these two fit together.
The whole idea of a database-driven Web site is to allow the content of the site to reside in a database, and for that content to be dynamically pulled from the database to create Web pages featuring it for people using a regular Web browser to view. So on one end of the system you have a visitor to your site using a Web browser, loading http://www.yoursite.com/, and expecting to view a standard HTML Web page. On the other end you have the content of your site sitting in one or more tables in a MySQL database that only understands how to respond to SQL queries (commands).
The PHP scripting language is the go-between that speaks both languages. Using PHP, you can write the presentation aspects of your site (the fancy graphics and page layouts) as "templates" in regular HTML. Where the content belongs in those templates, you use some PHP code to connect to the MySQL database and -- using SQL queries just like those used to create a table of jokes - retrieve and display some content in its place.
Just so it's clear and fresh in your mind, this is what will happen when someone visits a page on our database-driven Web site:

- The visitor's Web browser asks for the Web page using a standard URL.
- The Web server software (Apache, IIS, or whatever) recognizes that the requested file is a PHP script, and so interprets it using its PHP plug-in before responding to the page request.
- Some PHP commands connect to the MySQL database and request the content that belongs in the Web page.
- The MySQL database responds by sending the requested content to the PHP script.
- The PHP script stores the content into one or more PHP variables, then uses the now-familiar `echo` function to output it as part of the Web page.
- The PHP plug-in finishes up by handing a copy of the HTML it has created to the Web server.
- The Web server sends the HTML to the Web browser as it would a plain HTML file, except instead of coming directly from an HTML file, the page is the output provided by the PHP plug-in.

Appendix

1. From: http://dev.mysql.com/tech-resources/articles/ddws/