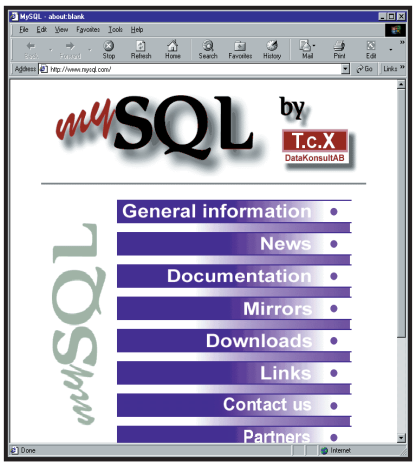
Uniquely Unix

Tim Anderson takes the Unix approach to data-driven websites with open-source applications.

In the last few columns I have looked mainly at using Active Server Pages (ASP) to create data-driven websites. ASP is ideal for Windows developers, but less useful on Unix-like operating systems. Fortunately there are other options. A great solution is the combination of PHP (which in some curious way stands for Hypertext Preprocessor) and a database such as MySQL.

These products are children of the Internet era. PHP started off as a parser, written by Rasmus Lerdorf for his home page. It has been worked on since by many others and is now available under the GPL (GNU General Public License). MySQL is mainly written by Monty Widenius, with support from the Internet community. Both are open-source products. MySQL has its own licence, which is complex because the terms for Windows are different (and less favourable) than for other operating systems. However, MySQL is free for use on web servers running on non-Microsoft operating systems. PHP and MySQL are available for Windows, but ASP is the natural choice on Windows.



MySQL is proving popular as a web database

FIG 1

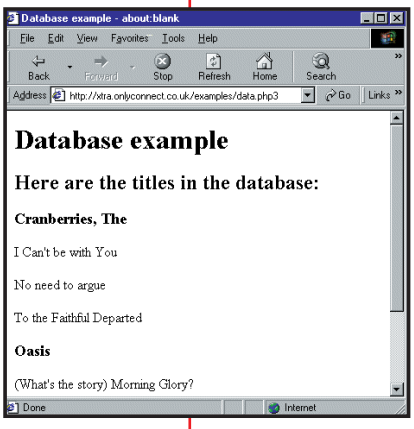
PHP code to query a MySQL database

```
<HTML>
<HEAD>
 <TITLE>Database example</TITLE>
</HEAD>
<BODY>
<H1>Database example</H1>
<H2>Here are the titles in the database:</H2>
<?php
$conn = mysql_connect();
$db = mysql_select_db("test",$conn);
$result = mysql_query("select * from titles, artists where
titles.artistid
= artists.id order by name, title",$conn);
if ($result) $numrows =
mysql_num_rows($result);
if ($numrows == 0) {
echo "<p>No results available</p>";
exit;
}
$artistname = "";
for ($i=0; $i < $numrows; $i++) {
 $newname = mysql_result($result,$i,"name");
 if ($newname != $artistname) {
 echo "<h3>" . $newname . "</h3>\n";
 $artistname = $newname;
 }
 echo "<p>" . mysql_result($result,$i,"title") . "</p>\n";
 }
?>
</BODY>
</HTML>
```

(*Key:* ↙ *code string continues*)

Getting started

The PHP/MySQL combination could prove useful both on an intranet and the Internet. It is worth getting an intranet solution working, if only to make it easier to design an application before moving it to a remote ISP. My suggestion is you install Linux on a spare computer, hook it up to your network, and install Apache, PHP and MySQL. I use SuSE Linux, which comes with MySQL and Apache pre-configured with PHP support, and the same may be true of most recent Linux distributions. The other option is to visit the PHP and MySQL websites and download the latest versions. You can download binary or source installations, with the latter recommended if you want to be right up-to-date. The websites and documentation offer beginners plenty of tutorials and advice, so even compiling your own source is usually not too challenging.



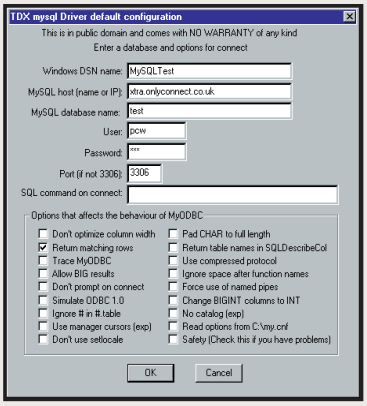
Output from PHP/MySQL running on a Linux intranet server

If you don't have a spare machine, you can dual-boot Linux and Windows. In this case, however, stay with Windows and install the Windows versions instead.

Access to successful data transfer

While there is nothing wrong with MySQL's command-line tools, it is also useful to work with a program such as Microsoft Access, to get things such as graphical report tools and the ability to view tables in scrolling grids. Using the MyODBC driver, you can attach MySQL tables to an Access database. First, download and install the driver from the MySQL site. Next, configure the driver (see screenshot right), ensuring that the Return Matching Rows option is checked.

You should also check that the named database exists on the MySQL server. If you are connecting across a network, MySQL needs to be configured to accept the specified user connecting from the host on which the ODBC driver is located. You do this by adding an entry to the MySQL user table for the host, user, password and permissions combination.

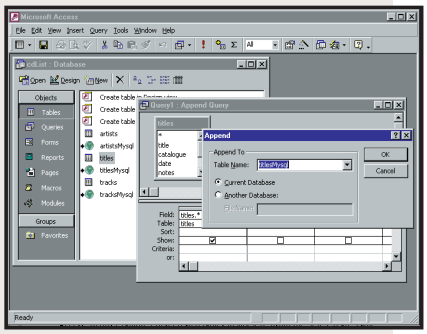


Set up the MySQL ODBC driver to use it with other Windows apps

On an intranet, you would typically use %.yourdomain.co.uk to allow connections from any internal host.

Once the driver is set up, you can get at the MySQL data from any ODBC client. MsQuery, which comes with Office, is a useful general ODBC client from which you can execute SQL statements. In Access, open a database and choose File, Get External Data, Link Tables to attach MySQL tables. As long as the tables have a primary key defined, you should get a read-write connection.

The screenshot above shows how to use this feature to copy data between MySQL and Access. In this example I first created tables in MySQL using the SQL Create Table command, with field names and types to match those in the Access database. Next, open a query and add the table you want to copy. From the Query menu choose Append Query, and select the matching MySQL table as the target. As a further refinement, choose



Once a MySQL table is attached to Access through ODBC, it is easy to exchange data. In this example, MySQL is running on Linux

View, SQL view and add the clause:

```
'WHERE yourtable.ID
NOT IN (SELECT ID
from yourtableMsql)'
```

In this example, the ID field is the primary key. The advantage is that records will only be added if they do not already exist. You can do the same in reverse, to copy data from MySQL to Access.

This way you can run MySQL at the same time as other Windows applications, making it easier to transfer data (see panel above).

MySQL is not as sophisticated as SQL Server or Oracle. For example, it does not yet support transactions, stored procedures or subqueries. But it is fast and stable. As its name implies, it is an SQL database server, and the supplied administration tools are command-line utilities that accept SQL statements. Much MySQL documentation is an SQL reference. You need to learn SQL to work with MySQL. The ODBC driver is a handy route to more user-friendly data management.

Using PHP

PHP works on similar principles to Active Server Pages. In both cases, the source HTML is processed on the server before being delivered to the browser. The big advantage is you can include dynamic content without needing to run scripts or applications on the browser client.

The first thing to know about PHP is that, like ASP, it uses script delimiters. The safest delimiter to use is like this:

```
<?php echo "Greetings" ?>
```

The command echo simply outputs the following string to the browser.

The next thing to know is that the scripting language is loosely based on C. For example, '=' is the assignment operator while '==' is for comparisons. Variable names are case-sensitive and begin with a $ character. Download the manual in HTML format, pop it on your local web server, and you have a ready online reference.

Figure 1 is an example of how you might use PHP to query a database. It used PHP's built-in support for MySQL. The function mysql_connect sets up the connection, but may need arguments for username and password. The next two commands select a database and perform a query. The command mysql_result retrieves the value of a particular field in a row. There is a mysql_close function, but no need to use it as the connection is automatically closed when the script ends.

Here's a PHP tip: include the line:

```
<?php echo phpinfo(); ?>
```

to retrieve a comprehensive table of how PHP and the web server are configured.

There will be a further look at how to make PHP/MySQL interactive using forms and searches next month.

CONTACTS

Tim Anderson welcomes your web development queries, at **webdev@pcw.co.uk**. You can find this PHP example in the Hands On section on this month's CD-ROM. The home page for PHP is at **www.php.net**. The home page for MySQL is **www.mysql.com**. There are numerous mirror sites. If your ISP supports PHP and MySQL, check for specific information on how it is set up as this varies.