

An Access Front-End to MySQL

by
Peter Lavin

June 5, 2004

Overview

Using ODBC, Access can connect with a MySQL database. A proprietary front-end to an open-source database? Microsoft and MySQL? Irreconcilable differences would seem to condemn such a union to a rapid divorce.

Read on to see how, despite expectations to the contrary, this marriage can work.

Why do it?

Even though I have been running a dual-boot Windows/Linux machine for a number of years I still enjoy being able to see my Windows partition from Linux. So, as you might imagine, the voyeuristic pleasure of seeing a MySQL database through the eyes of Access was irresistible. This was all the motivation I needed to attempt to use Access as a front-end for MySQL. However, for the more practically minded, there are many good reasons for doing this; let me enumerate a few of them.

Firstly, MySQL does not come with a graphical interface. The ability to open multiple windows of different tables and queries is a real advantage especially to those who are not used to working from the command line.

How do you go about building a report from a MySQL database? Say what you like about Access but there is no denying that its reporting capabilities are excellent.

While it has its limitations, Access' "Query by Example" window for creating queries can be very useful, especially for those who don't have a strong grasp of SQL.

Finally, maintaining a website is expensive. With a dynamic, database-driven site, clients who are competent computer users can keep their costs down by maintaining the database themselves. In this instance, using Access as an interface to an online database is an attractive option because, for those familiar with Access, there will be no learning curve. Making changes to records in a MySQL database will be no different from doing so in Access.

In sum, using Access in this way might appeal to website administrators, developers and power users.

Preliminary Steps

I'll give a quick overview of how to connect to your MySQL database trying not to repeat information that is readily available elsewhere. For instance, there is already excellent information on how to create an ODBC connection available in the article, "[Using MyODBC To Access Your MySQL Database Via ASP](#)".

Obviously, the first thing you will need is a MySQL database. My primary concerns are with online databases but if you don't have one, a local database will do for testing purposes. However, I should note that to connect to an online

database you will need to be able to access it through the domain name and not, as you might from a script on the server, using “localhost”.

If you don't already have the ODBC driver for MySQL installed you can download it from the URL, <http://www.mysql.com/products/connector/odbc/>. At the time of writing the name and version number is “MySQL ODBC Driver 3.51”. Once this is installed check that a driver with this name shows up in the Drivers' window of the ODBC administrator. You can see the article referenced above for information about how to create a system or a user DSN. You are now ready to use Access as a front-end to MySQL.

Connecting

Let's quickly go through the steps necessary to connect to your MySQL database. Open Access and choose the “More Files” option from the dialogue box. Drop down the “Files of Type” list box, go to the bottom and choose “ODBC Databases”. Click on the tab “Machine Data Source” and choose the connection name you just created with the ODBC administrator. From here on it should be pretty straightforward although you may be asked to create primary keys for any tables without unique identifiers.

Following these steps will create a database in “My Documents” with the name “ODBC1.mdb” or a variant on that name. This database will then open showing all the tables with a globe icon to their left. This icon indicates that the tables are external, linked tables.

Double clicking a table name will open it in that familiar Access tabular format. Records can be added or deleted exactly as they would if the database was a native Access one.

Limitations

First and foremost we know that data can be maintained using Access as a front-end. This is nothing to sniff at but what else can we do?

Let's try out some queries developed for use with a MySQL database. Look at the one below for example:

```
SELECT DISTINCT tblusers.usercode as uc, monthbooked,  
daybooked, fldrate  
FROM tblrates, tblbookings, tblusers  
WHERE tblusers.usercode = tblbookings.usercode  
AND tblbookings.monthbooked = tblrates.fldmonth  
AND tblusers.billing = 1
```

```
AND ISNULL(tblbookings.dateinvoiced)
ORDER BY uc, monthbooked, daybooked
```

This query executes within MySQL but when tried from Access returns an error on the aliased field (“uc”) in the “ORDER BY” clause. This is remedied rather easily by replacing the aliased field with its actual name, “tblusers.usercode”. A promising start but the query;

```
SELECT username FROM tblonline
WHERE fldwhen > (SUBDATE(CURTIME(),
INTERVAL '10' MINUTE))
```

needs to be rewritten to use the Access-specific function “Datediff”. In fact no MySQL-specific functions may be used. Remember, you are dealing with linked tables from within Access and this is going to limit the ability of the driver to support MySQL query syntax. It is for this reason that data definition queries on the linked tables, such as the one below, will not execute.

```
ALTER TABLE tbllinks ADD COLUMN fldtype varchar(35)
```

You can however create local tables from remote ones and in some situations this can prove useful.

Likewise none of the metadata methods of MySQL such as “SHOW” or “DESCRIBE” will work.

It quickly becomes clear that linked tables are treated as though they were Access tables, so knowledge of Access SQL syntax is an absolute must if you wish to create queries.

With large tables there may also be performance issues to consider when using Access as a front-end. Clicking on a linked table and bringing all the data across the network/internet will definitely slow performance.

State of the Union

For the developer who is familiar with the Access dialect of SQL there are certainly times when it may be useful to connect to an online database using Access. However, there are a considerable number of limitations to what can be done. The functions of MySQL are not supported and also, surprisingly, some of the capabilities of Access. For instance, I was disappointed to find out that nested select statements - a capability that production versions of MySQL lack but Access has - won't work. Also, given the restrictions on altering the structure of your database, a developer will most often want to work with a tool such as phpMyAdmin rather than Access. In some ways it looks like we have the worst of

all possible worlds – we've lost functionality on both sides of the equation. But don't start the divorce proceedings just yet.

Access' reporting capabilities are quite a different matter. You can use the report wizard to create professional quality reports quickly. Granted that a report is only as good as the query it is based upon, knowledge of Access' version of SQL will also be necessary in order to use the report writer.

You can also take advantage of some of the other features of Access. You can use Access' documentation capabilities (Under the menu options, "Tools", "Analyze", "Documenter") to produce a quick report on the structure of your database. Note though, that MySQL data types will have been converted to their Access equivalent. If you need an Entity Relationship Diagram of the tables in your database this can also be created and printed from Access. Unfortunately though, cardinality can not be shown. Should you need to convert a MySQL database to Access, using Access as a front-end is probably the quickest way to do this. With a "SELECT ...INTO" statement you can quickly produce a local Access version of any linked table.

What is a disadvantage for the developer is perhaps an advantage for the database user. Because data definition queries are not supported there is no danger of the database structure being accidentally altered. While phpMyAdmin is an excellent piece of software and not difficult to use, lots of damage could be done in a short amount of time by an inexperienced user. If you are a web developer with customers who are willing and able to do their own database updates, Access may be an ideal tool to act as a front-end. If the database is not large and your client is already familiar with Access this may in fact be the ideal solution.

Some web developers might question the wisdom of allowing clients to perform database updates to their site – Isn't that taking money out of your own pocket? I don't think so. A client who is actively involved with his/her website is much more likely to continue needing your services than one who isn't. A customer who recognizes the utility of their website is going to find even more uses for it.

Resources

<http://www.devaricles.com/c/a/ASP/Using-MyODBC-To-Access-Your-MySQL-Database-Via-ASP/>

<http://www.mysql.com/products/connector/odbc/>

<http://builder.com.com/5100-6388-1050135.html>

About the Author

Peter Lavin runs a Web Design/Development firm in Toronto, Canada. For more information visit <http://www.softcoded.com>. He may be reached at peterlavin@sympatico.ca.