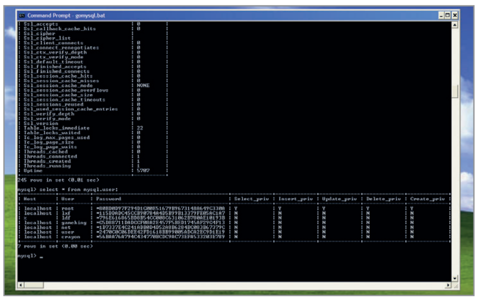
Project 5: Using MySQL with Base

Base is capable of interfacing with remote database servers such as the popular and free MySQL. Try it!

You've already seen how *Base* can help you organise your data with the minimum of fuss, but although it's easy, pretty and powerful, *Base* does have some limitations. The most noticeable of these is the fact that your data is stored in a file on your computer, which makes it rather hard to share with other people. Even if you did share it with other people, how would you make sure that everyone can make changes at the same time? The answer is that you can't – *Base* files are only really meant to be used by a single person at a time, which is fine for things like CD collections and other simple tables, but rather useless for important things such as company-wide systems or websites.

For these kinds of solutions you need a real database server like *MySQL*, *Oracle* or *Microsoft SQL Server*. Usually these are much harder to use than *Base*, because you need to delve into the world of SQL and often use task-specific tools such as *Oracle*'s form designer. But *Base* is able to work as a front-end to all of these databases and many others, which means you get all the attractive and easy user interface help that *Base* brings, while also getting the advantages of storing your data on a dedicated database server.

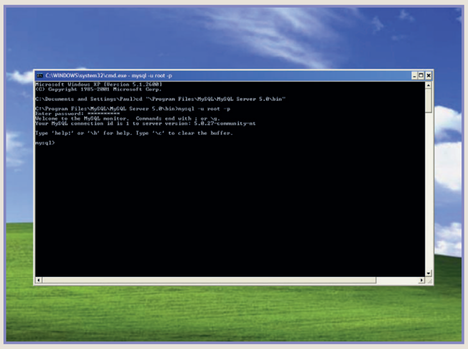
In this project you'll learn how to hook *MySQL* up to *Base* so that you can work with your current tables, but also create new tables using *Base*. You can also add some *Base*-specific stuff to the table, such as saved queries, forms and reports – all relying on your *MySQL* database. The nice thing about this link up is that the *MySQL* database server is still able to respond to normal requests – if you have a PHP site up, for example, that can be writing to the database while you're working with it in *Base*.



```
| Ssl_accepts                    | 0    |
| Ssl_callback_cache_hits        | 0    |
| Ssl_cipher                     |      |
| Ssl_cipher_list                |      |
| Ssl_client_connects            | 0    |
| Ssl_connect_renegotiates       | 0    |
| Ssl_ctx_verify_depth           | 0    |
| Ssl_ctx_verify_mode            | 0    |
| Ssl_default_timeout            | 0    |
| Ssl_finished_accepts           | 0    |
| Ssl_finished_connects          | 0    |
| Ssl_session_cache_hits         | 0    |
| Ssl_session_cache_misses       | 0    |
| Ssl_session_cache_mode         | NONE |
| Ssl_session_cache_overflows    | 0    |
| Ssl_session_cache_size         | 0    |
| Ssl_session_cache_timeouts     | 0    |
| Ssl_sessions_reused            | 0    |
| Ssl_used_session_cache_entries | 0    |
| Ssl_verify_depth               | 0    |
| Ssl_verify_mode                | 0    |
| Ssl_version                    |      |
| Table_locks_immediate          | 22   |
| Table_locks_waited             | 0    |
| Tc_log_max_pages_used          | 0    |
| Tc_log_page_size               | 0    |
| Tc_log_page_waits              | 0    |
| Threads_cached                 | 0    |
| Threads_connected              | 1    |
| Threads_created                | 1    |
| Threads_running                | 1    |
| Uptime                         | 5707 |
245 rows in set (0.01 sec)

mysql> select * from mysql.user;
| Host      | User     | Password                                  | Select_priv | Insert_priv | Update_priv | Delete_priv | Create_priv |
| localhost | root     | *80BD0D97F294D1C00E51679B9673148A649C330A | Y           | Y           | Y           | Y           | Y           |
| localhost | lxf      | *115DDADC45CCB90784A4D5B99B13379FE05AC1A7 | N           | N           | N           | N           | N           |
| %         | ldf      | *796E616065BD0B54CC008C631062B780AE10193B | N           | N           | N           | N           | N           |
| localhost | gameking | *C5D88711DADCCF0BA2E457F5BED1745AF29C4F13 | N           | N           | N           | N           | N           |
| localhost | net      | *1D7337E4C241ADB0D4D52A0B6204DC003D67379C | N           | N           | N           | N           | N           |
| localhost | user     | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N           | N           | N           | N           | N           |
| localhost | crayon   | *56BAA76A794C434770BCDC9AC73EFA533203E789 | N           | N           | N           | N           | N           |
7 rows in set (0.00 sec)

mysql> _
```

› ***MySQL* monitor: text-based, tricky to learn and easy to make mistakes – use *Base* instead?**

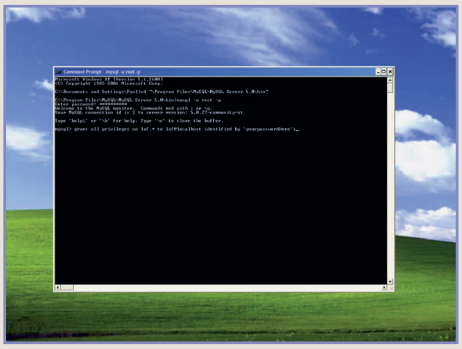
Base: change your style



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Paul>cd "\Program Files\MySQL\MySQL Server 5.0\bin"

C:\Program Files\MySQL\MySQL Server 5.0\bin>mysql -u root -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

1 User account

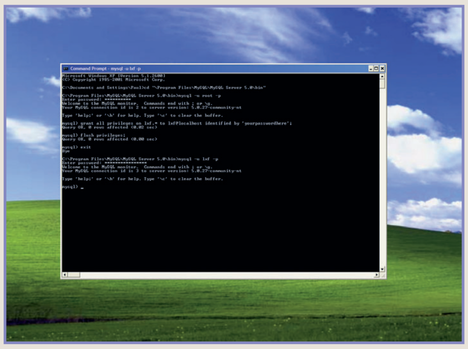
Base needs a user account on your *MySQL* server in order to connect, read and write data, so fire up your *MySQL* monitor as the root user – **mysql -u root -p** ought to do it if you're hosting locally.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Paul>cd "\Program Files\MySQL\MySQL Server 5.0\bin"

C:\Program Files\MySQL\MySQL Server 5.0\bin>mysql -u root -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant all privileges on lxf.* to lxf@localhost identified by 'yourpasswordhere';_
```

2 Grant privileges

If you don't have a database already, run the command **create database mydb**. Regardless, run this command: **grant all privileges on yourdatabasehere.* to youruserhere@localhost identified by 'yourpasswordhere';**



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Paul>cd "\Program Files\MySQL\MySQL Server 5.0\bin"

C:\Program Files\MySQL\MySQL Server 5.0\bin>mysql -u root -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant all privileges on lxf.* to lxf@localhost identified by 'yourpasswordhere';
Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

C:\Program Files\MySQL\MySQL Server 5.0\bin>mysql -u lxf -p
Enter password: ****************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

3 Flush it away

Now run **flush privileges** to ensure your account data is saved, then type **exit**. Now double-check everything worked by running **mysql -u youruserhere -p** – if you can connect, then you're now ready to go!

Base: understanding MySQL

MySQL connects to *Base* using the JDBC access layer, which is short for Java Database Connectivity. This layer allows any Java-enabled program to connect to any Java-enabled database, which might sound like it opens up a world of possibilities, but actually it's pretty limited. There are quite a few programs out there that can read databases using Java, of which *Base* is one of the better examples. But there are remarkably few databases around that are open to Java as standard, which is why you usually need to install a special connector: something that sits between your database and Java, translating JDBC instructions into database-specific instructions. *MySQL* makes one of these for its own database, known as *MySQL Connector/J*, which translates JDBC instructions into *MySQL* server protocol in the same way that *MySQL for PHP* translates PHP functions into *MySQL* equivalents.

If you'd rather use a database other than *MySQL*, you'll need the specific JDBC connector for the database you choose. Alternatively, if you're on Windows, you can use *ODBC* to join *Base* and your database together – this is a non-Java database connection layer that has been around for quite some time. However, using ODBC requires you to know the precise name of your data source, so you will need some previous ODBC experience to get it working.

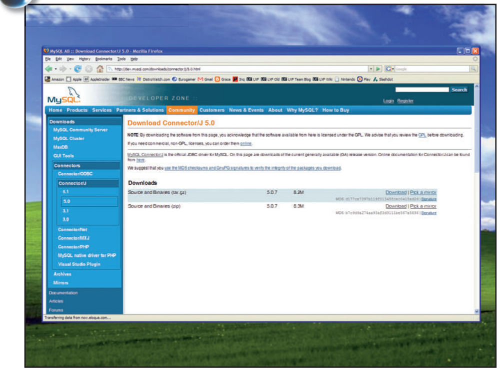
Choosing your data source

Apart from *MySQL*, *Base* is also capable of hooking up to *Microsoft Access, Oracle, Adabas B* and *dBASE* databases, as well as Microsoft's proprietary ADO data access system and many others using JDBC/ODBC. For more specialist uses, Base can also connect directly to various apps that store address books, including *Mozilla/Thunderbird, Microsoft Outlook, Windows Address Book* and – for corporate address books – LDAP servers. But keep in mind that these have specialist uses: we wouldn't recommend trying to write changes back to your LDAP server if you're accessing it from *Base*.

Once you're hooked up to your data source of choice using *Base*, you have two options: use *Base* as a dumb front-end reader for the data, or create customised local access methods. The first option means that anyone with *Base* can connect to the server, and read data as much as they like along with countless other users – *Base* just acts like a fancy version of the *MySQL* monitor. But if you want to use custom access methods such as queries, forms and reports, you should know that these are saved into the ODB file rather than into *MySQL* – if you make changes to those using *Base*, they won't be synchronized to the other users unless you give them a copy of your ODB file. So, if you intend to create a little database access system for desktop users to read from your database server, create it on your machine first, get it right, then copy it to everyone who needs it.

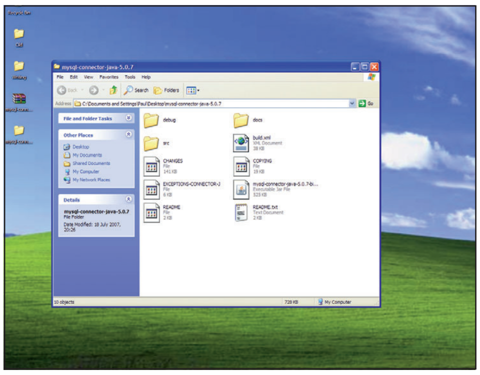
»

Base: how to link to MySQL



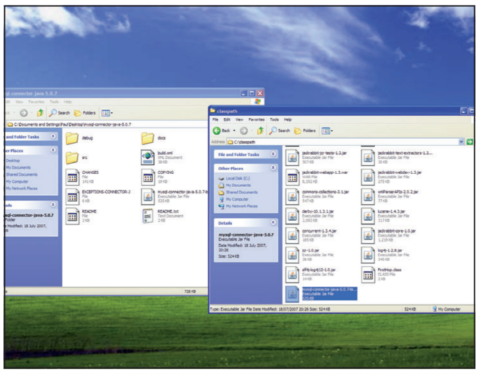
1 Get Connector/J

MySQL hosts the download for *MySQL Connector/J* itself at **http://tinyurl.com/ymt6rb** – download the "Source and binaries" .zip file from there.



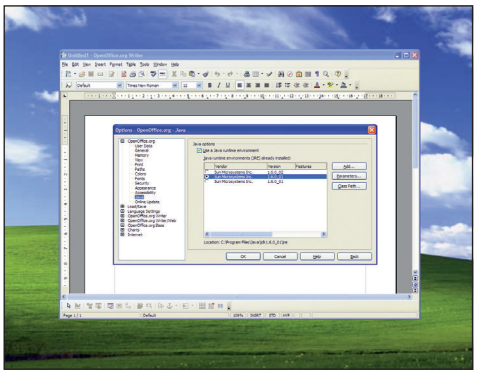
2 Extract to directory

Once the zip file is down, extract it into a directory of its own. There are lots files in there, but the one you're interested in is called **mysql-connector-java-5.0.7-bin.jar**.



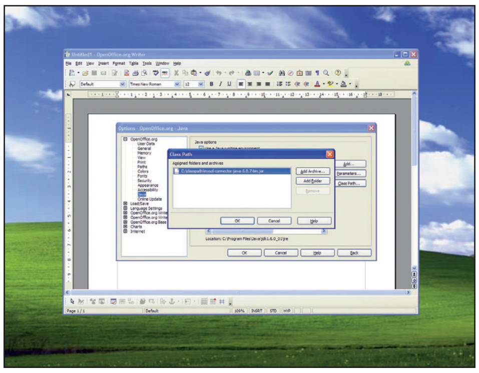
3 Basic functionality

That .jar file contains the *MySQL* functionality to connect *Base* to *MySQL*, so you'll need to create a directory on your hard drive called "classpath" (*eg* **c:\classpath**) and copy the file there.



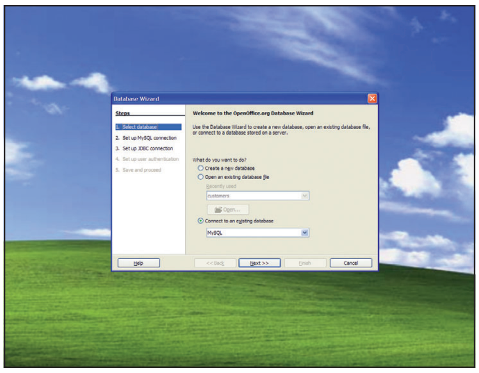
4 Java option

Now start *Writer*, and choose Tools > Options from the menu, then look in the *OpenOffice.org* settings and click the Java option. On the right of this window is a Class Path button – click that.



5 Customisation

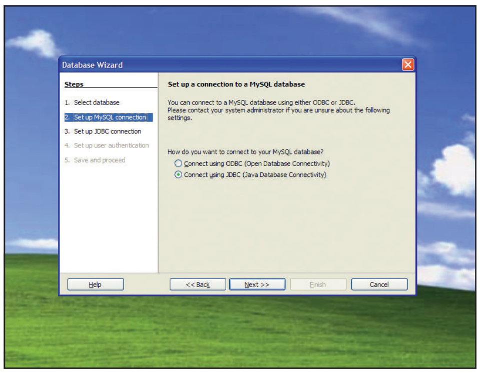
This screen lets you customise which .jar files are loaded into *Open Office.org*. First of all, click the Add Archive button, then select the .jar file you copied into **c:\classpath**.



6 Supported databases

Shutdown *Writer* and any other *OOo* apps, then launch *Base*. When the wizard appears, choose Connect To A Different Database, then choose *MySQL* from the list of supported databases.

Project 4: Base & MySQL



Set up a connection to a MySQL database

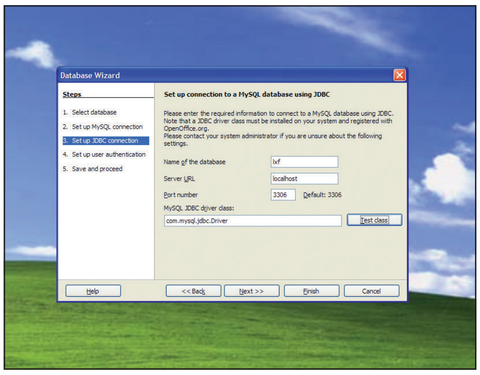
You can connect to a MySQL database using either ODBC or JDBC.
Please contact your system administrator if you are unsure about the following settings.

How do you want to connect to your MySQL database?

○ Connect using ODBC (Open Database Connectivity)
● Connect using JDBC (Java Database Connectivity)

7 Use default

This screen lets you choose between JDBC and ODBC – JDBC is the default, so leave it selected and click Next to continue on.



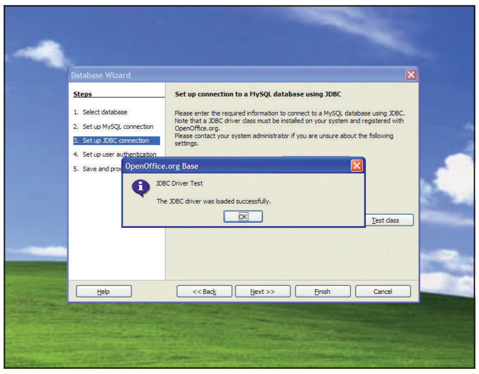
Set up connection to a MySQL database using JDBC

Please enter the required information to connect to a MySQL database using JDBC. Note that a JDBC driver class must be installed on your system and registered with OpenOffice.org.
Please contact your system administrator if you are unsure about the following settings.

Name of the database: lxf
Server URL: localhost
Port number: 3306 Default: 3306
MySQL JDBC driver class: com.mysql.jdbc.Driver

8 Enter name

Enter the name of your database and server here, but leave the MySQL JDBC Driver Class field set to its default value. If your *MySQL* server is on the same machine as base, use **localhost** for the server URL.



Set up connection to a MySQL database using JDBC

Please enter the required information to connect to a MySQL database using JDBC. Note that a JDBC driver class must be installed on your system and registered with OpenOffice.org.
Please contact your system administrator if you are unsure about the following settings.

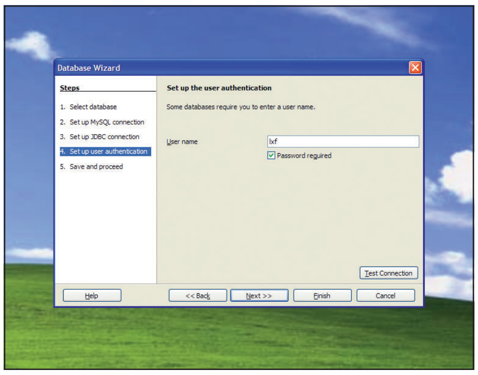
OpenOffice.org Base

JDBC Driver Test

The JDBC driver was loaded successfully.

9 Test class

Before you continue, make sure you click the Test Class – this ensures that the *MySQL/Java* connection system is working. If all is well, you should see this message!



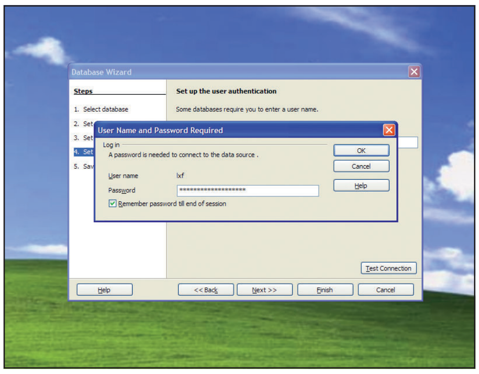
Set up the user authentication

Some databases require you to enter a user name.

User name: lxf
☑ Password required

10 Username entry

Now enter the username you set up on the first page, and make sure you tick the Password Required box. As with before, make sure you click the Test button to make sure everything is working



Set up the user authentication

Some databases require you to enter a user name.

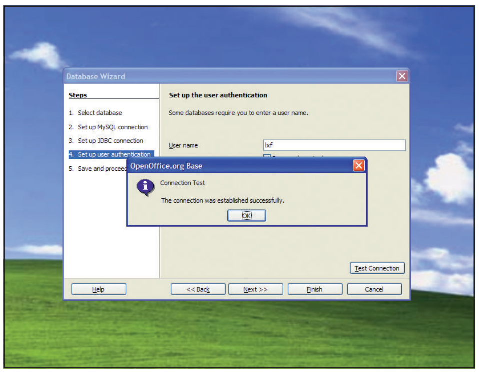
User Name and Password Required

Log in
A password is needed to connect to the data source .

User name: lxf
Password: ********************
☑ Remember password till end of session

11 Remember your password!

MySQL will prompt you for the password for your user account – make sure you tick the Remember Password option, otherwise it will harass you for the password again later.



Set up the user authentication

Some databases require you to enter a user name.

User name: lxf

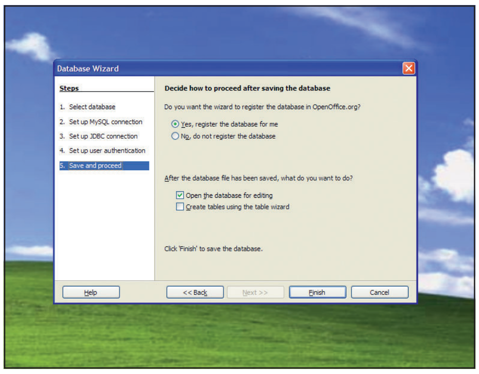
OpenOffice.org Base

Connection Test

The connection was established successfully.

12 Start MySQL

If *Base* connected to *MySQL* using the username you enter, you'll get a confirmation message here. If not, make sure *MySQL* is started and, if the server is remote, that it is configured to allow remote connections.



Decide how to proceed after saving the database

Do you want the wizard to register the database in OpenOffice.org?

● Yes, register the database for me
○ No, do not register the database

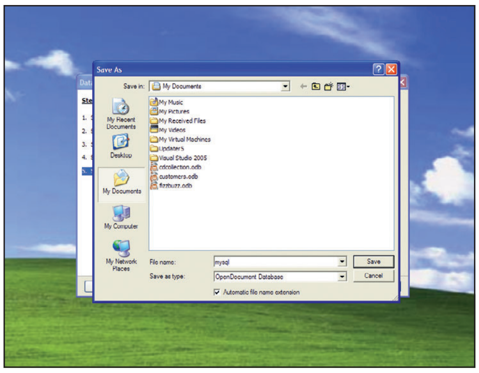
After the database file has been saved, what do you want to do?

☑ Open the database for editing
☐ Create tables using the table wizard

Click 'Finish' to save the database.

13 Back to Base

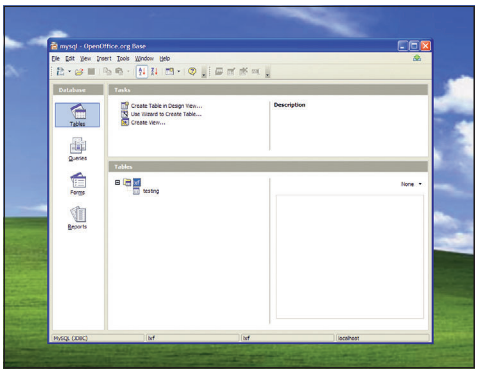
Now you come back to the normal *Base* database configuration screen – you can safely ignore these options and click Finish.



Save in: My Documents
File name: mysql
Save as type: OpenDocument Database
☑ Automatic file name extension

14 Store information

Base will prompt you to save your database now. This is because *Base* stores all non-table information in its ODB file, as well as connection information to *MySQL* – you're not actually saving the table data.



MySQL (JDBC) | lxf | lxf | localhost

15 Connect to MySQL server

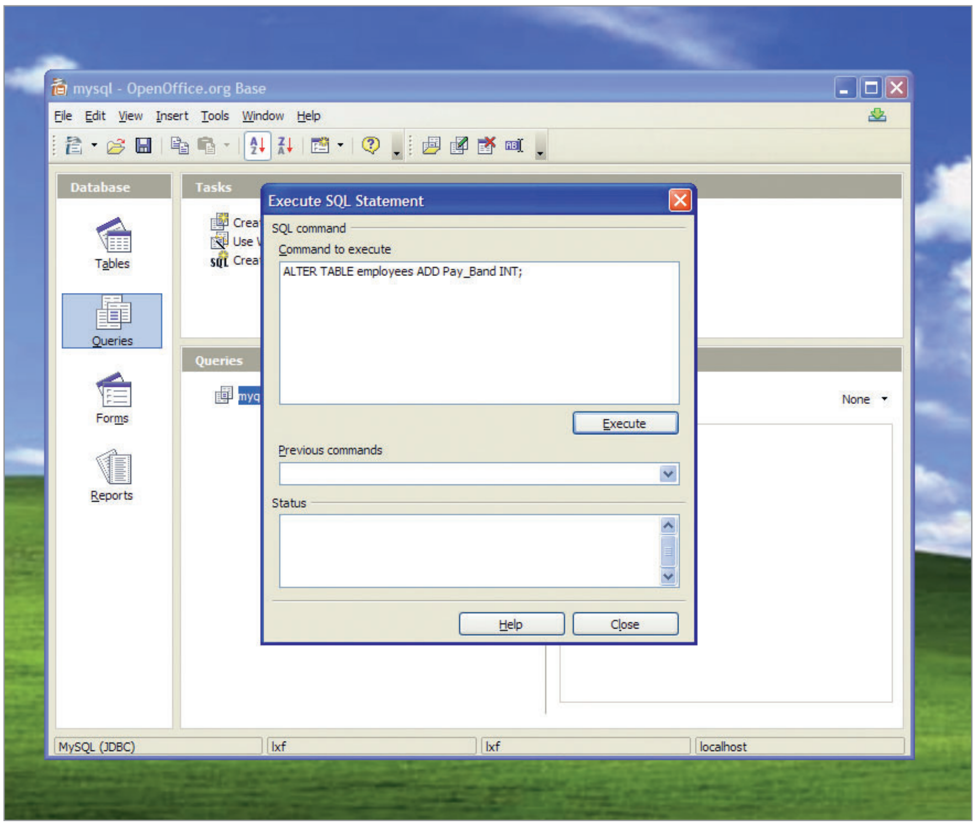
After all that work, you should finally be able to connect to your *MySQL* server in *Base*. Here you can see we have one table already created. In the future, just open the ODB file you saved.

Base: SQL primer

Now that Base and MySQL are linked together as one, you can push your database skills even further...

Just so that you don't overwrite any data by accident, we would recommend you only *read MySQL* tables rather than write them, which restricts your uses to queries, forms and reports. The idea here is that your existing data access systems – *eg* a PHP web site, or custom-coded data access programs – continue to write data normally, and *Base* can only read. If this mental safeguard weren't in place, you might open up a table for viewing while someone else makes changes, then overwrite those changes by accident when you save the table! *Base* does let you write data if you want to, but be careful!

If you're already a veteran *MySQL* user, you'll almost certainly be used to work inside the command-line *MySQL* monitor application – it lets you "talk" to the server and get responses almost immediately. Well, don't fret: *Base* has an equivalent system both for creating queries and for typing in general commands.



```
ALTER TABLE employees ADD Pay_Band INT;
```

› ***Base* lets you enter your own SQL queries directly into the interface, but don't try it with SELECT queries because you'll get no results back!**

Multiple masters

If you've been using *MySQL* for any length of time, you'll know that any SQL statement in *MySQL* is called a "query" - whether it reads data, writes data, deletes data, or whatever. In *MySQL*, these queries are one-off affairs that you can run as you please by typing in some SQL and seeing the results, but in Base a query must be saved then viewed. Working the *MySQL* is nice because you can rattle off queries as and when you need them, making changes where needed just by amending your SQL.

But equally, the *OpenOffice.org Base* way of working has its own big advantage: any query you write gets saved so that running it again is just a matter of double-clicking it. Of course, it also helps that you don't need to understand SQL to read results from *Base* queries!

Saved SQL queries

OpenOffice.org lets you type queries freely by choosing the Create Query In SQL View option in the Queries tab. In fact, this is the same query editor we used back on p66 (the *Advanced Base* tutorial), just with the wizard parts hidden. For instance, let's for the moment say you have the database "employees" with fields FirstName, LastName and Age. You could write a quick query in Base with this SQL, thus:

```
SELECT FirstName, LastName FROM employees
ORDER BY Age;
```

When you save that query (yes, you have to give it a name), you can right-click on it in the queries view and choose Edit. Except when you back to the editing view you'll be in the standard query designer view where you can see all the employees table in the top part and the fields you selected in the bottom. What's happened is that *OOo* has parsed your query and converted it into its wizard display. In fact, any changes you make to the query in this design mode will be written back to your SQL – try choosing View > Switch Design Mode On/Off from the menu, and you'll see your modified SQL query.

Free-form SQL queries

Base also lets you run SQL queries in the fly, but this is (usefully) limited to queries that write data rather than read. To try it out, go to Tools > SQL from the menu bar, then enter a query. If your query executes without error, you'll see a confirmation message in the lower part of the screen. But no matter what kind of query you run, *Base* won't print out the results for you – which is why the free-form query window is rather pointless for SELECT queries! LXF