
**Abstract**

It is not widely known that the OpenOffice.org suite contains a very capable database front-end component that can be used to access a number of Database Management Systems (DBMS's). In this paper the use of OpenOffice Base (as it is called) to access various DBMS's is explained and its use as a teaching tool is explored.

**Keywords**

Database administration, open source software

**1. Introduction and Background**

"Database Administration involves the overall design and management of the database. Administration tasks include archiving, consistency checks, developing/maintaining indexing and retrieval functionality, migration, monitoring, performance issues, replication issues, and database sizing/space management." (Wikipedia, n.d.)

Students are confronted with various software tools, each with its own interface and idiosyncrasies (for a selection of tools, see for example http://www.tdan.com/i005fe04.htm).

This situation has had the unfortunate side effect that students tend to get bogged down in the complexities of the particular DBMS and in the process struggle to understand the basic concepts of database design. In addition, students tend to become proficient in a particular commercial product, rather than to be able to extend their knowledge across products.

" Without explicit instructions, students do not master the art of using available tools to solve problems; therefore, teachers need to question whether their teaching practices provide explicit instruction to develop students' problem solving skills" (Chen & Ray, 2004).

In the past this situation was accepted as being "how it is" -- students simply had to learn these tools before they could do any work on a particular DBMS. Academic textbooks often cover only a particular software tool (Rob & Corronel, 2004, pp. 741 - 755). However using software to illustrate the concepts of database management and design is not uncommon - see for example (Douglas & Barker, 2004). This report will attempt to evaluate the usefulness of OpenOffice software as a simplified database teaching tool.

Sun Microsystems donated the Staroffice 5.2 code base to the open source community in 2000 and a major rewrite of the whole suite followed (see http://www.sun.com/software/star/openoffice/index.xml ). In 2003 Sun released Staroffice 7 and the OpenOffice.org community released OpenOffice 1.0. The next major release of OpenOffice was planned for October 2005, coinciding with the release of Staroffice 8. This office suite comprises of a word processor (named "OpenOffice.org Writer"), a spreadsheet package ("Calc"), a presentation manager ("Impress"), a graphics editor ("Draw"), a scripting language ("Basic") and a database front-end ("Base") (see http://about.openoffice.org/index.html#history ).
During 2004 a number of large organisations have indicated that they will be using OpenOffice.org as a standard office suite; the latest (January 2005) being the French Police Department. Many observers are now considering OpenOffice.org as a viable suite for modern businesses. For example, the Australian PC Authority magazine considers OpenOffice.org to be the "Editor's Choice" in the January 2005 issue, in preference to products such as Microsoft Office, Lotus Smartsuite, Corel Office and Ability Office (Ellis, 2005). It is clear that OpenOffice.org is becoming a major player, and the fact that it is free is a big consideration.

There are therefore two good reasons to use OpenOffice.org as a teaching tool: a simple and unified front-end to various DBMS's, and cost to students. For example: setting up OpenOffice.org and MySQL on any operating system has no financial costs, and students can build a fairly sophisticated database application on this platform.

The report is organised as follows: after a brief literature review, the installation process is described. The results of the functionality testing are discussed and an evaluation including student feedback is provided.

2. Literature Review

Documentation on the OpenOffice database front-end was (and still is) very limited and sparse. (Koch, Murray, Roth, 1999) covers StarOffice 5.1, which differs so drastically from Staroffice 7 that it simply isn't usable. The StarOffice 5.2 User's Guide (2000) devoted 20 pages out of 490 on this topic; (Haugland & Jones, 2001) uses 47 pages out of 1121, duplicating it in their 2002 book. In these books the emphasis is purely on using OOB to set up tables for simple office tasks, such as a mail merge, but the technical detail of connecting to a DBMS was skipped.

Not surprisingly there is more recent and usable information available on the Internet. (See "Additional Sources" ). The OpenOffice documentation website at http://documentation.openoffice.org has some up-to-date material. The best technical source was found to be John McCreesh's document on the Unixodbc.org website, in which he describes the whole process of installing all the software and how to link OOB to a database. (John McCreesh is the Project Leader for the OOB project).

ODBC, on the other hand, is well documented in numerous books and articles, but none of these are relevant in the discussion of linking OpenOffice with a DBMS via ODBC, and are therefore not covered in the literature review.

3. Some Notes on Software Installation and Configuration

OpenOffice is available from http://www.openoffice.org but is widely being distributed by many organisations. Almost all modern Linux distributions now include OpenOffice.org as part of the standard package, and it is installed by default. Even so, installation on Windows, Linux or Mac X is very simple and straightforward. OpenOffice.org provides clear instructions and a comprehensive installer.

The OOB has built-in access to ODBC, JDBC, MySQL and Adabas. In this paper the ODBC functionality will be explored, as all modern DBMS's have ODBC drivers. Once OpenOffice.org is installed, the user needs to set up the ODBC connections to the various DBMS's.

In MS Windows this is a simple matter:

1. Install the DBMS (e.g. MySQL MS SQL Server, Oracle, Caché, if not already installed)
2. Install the ODBC driver for the particular DBMS, e.g. MyODBC for MySQL
3. Create a database on the DBMS and set up user permissions.
4. Use the ODBC Data Source Administrator in Administrative Tools from the Control Panel to create a new ODBC connection to the newly created database.

(This has been tested on Windows XP Professional connecting to MS SQL Server, MS Access, Oracle 9i, Caché 5 and MySQL 4.0.)
In Linux it is slightly more complex, but still well within the capability of students. Steps 1, 2 and 3 are the same as above, then:

4. Install unixODBC and libmyodbc (if not already installed)

5. Create a new ODBC connection to the newly created database. This is done by ensuring the files odbc.ini and odbcinst.ini contains the correct parameters. Please note: some Linux distributions put these files in /etc -- others put it in /etc/unixODBC ; others in /etc/unixODBC/unixODBC . (See McCreesh, 2002 for examples of MySQL parameters). A number of tools exist to set up the ODBC connections, but the easiest is still to simply edit these files with a text editor. Some Linux distributions contain ODBCConfig, which is functionally the same as the Windows ODBC Data Source Administrator and can be used to set up the ODBC connections. (See the Appendix for examples).

6. OOB looks for a library libodbc.so. As root user, set up a symbolic link in a terminal window:

   ln -s /usr/lib/libodbc.so.1 /usr/lib/libodbc.so

   (This has been tested on Suse 9.1, RedHat 9, Fedora Core 3 and Xandros 2.0 using MySQL.)

   Update: OpenOfficeBase Version 2.0 now (November 2005) contains a JDBC driver that can connect directly with a MySQL DBMS, thus bypassing steps 3, 4, 5 and 6 above. OOB 2.0 can now also create a new database, as opposed to version 1 which could only connect to an existing database.

4. Using OOB as a Database Administration Tool

A digital asset is an image or media element included with your paper. As this is a web format it is possible to include sound and video if this is appropriate, but it should be noted that these will only be included if the size is small. Also, authors should be aware that many corporate firewalls block elements other than images.

4.1 Setting up a "Data Source"

1. OpenOffice.org Base calls the connection to a DBMS a "data source", which can be accessed via F4 or by selecting Tools / Data Source. (F4 is preferred, as it opens a window that shows a tree-like structure - see Figure 1.)

2. The first time a user accesses this function, OpenOffice.org Base will show only one "database Bibliography", which is in fact a simple dbaseIV file. Right click on Bibliography and select "Administrate Data Sources".

3. Click "New Data Source"

4. Under the "General" tab, enter an appropriate name, e.g. "Invoices". Select the ODBC connection from the drop-down list, then click on the button to the right of "Data source URL". Provided the ODBC connectors have been set up correctly as discussed above, the window will show these. Select the correct connector.

5. Under the "ODBC" tab, set up a userid and tick the box if a password is required by the DBMS. (Users are normally set up as part of the DBMS install process. Clearly, if the user doesn't have the correct permissions to access the DBMS it isn't going to work!)

6. Click the "Tables" tab. At this point OpenOffice.org Base will try to connect via ODBC to the DBMS specified in step 4 above. Any errors in connecting will be shown at this stage. If no errors are shown, then the connection was successful. This window will also show any existing tables in the database being connected to.

4.2 Creating Tables

1. Click on the plus sign next to the database name in the left, tree-like panel. This will expand the tree and show "Links", "Queries" and "Tables".
2. Right click on "Tables" and select "New table design". This opens a new window where the student can simply define the field names and field types ("Attributes").

3. By right-click on the button to the left of a field and selecting "Primary key", the student can define the primary key for this table. (If no Primary Key was defined, OpenOffice.org Base will create a new numeric field as the key, if the DBMS allows this activity).

3. When finished, click the save button. OpenOffice.org Base will ask for a name for this table. After closing the Table Design window, OpenOffice.org Base will show the new table in the tree in the left panel and the empty table in the right panel. Data can now be entered directly into the right panel, i.e. directly into the table in the DBMS.

Figure 1 illustrates the creation of a table using MySQL DBMS under Xandros Linux. Note the tree structure of connected databases in the top left of the graphic. Also note the navigation buttons below the view of the table, middle right. These buttons can be used to go forward and backwards in a table; add or delete records etc.

![Figure 1. Creating a table](image)

### 4.3 SQL Queries

A very similar approach is used by OpenOffice.org Base for the design of queries:

1. Right click the "Queries" tab in the left panel, and select "New Query (Design view)"

2. A new window opens, and the student will be asked to "Add tables" to the query. For a simple query using only one table, select the correct table. If a more complex query is needed more than one table can be selected. Tables can be "joined" by simply dragging and dropping field names from one table to the next.

3. At the bottom of the Query Design form, the student can now set up the various conditions for the query, such as Field Name; Table name; Sort sequence, various functions (average, count, sum, group, min, max); and Criteria.

4. When finished, the query is saved by giving it a name. (In effect, OpenOffice.org Base saves the query as SQL statements.) The query can be run by simply clicking on it once.

Figure 2 demonstrates designing a query by using filters. A fairly complex query can be constructed by simply using the drop down lists to set up the conditions of the "SELECT" statement, such as specifying the table(s) and field(s) to be used; the ORDERED BY clause; functions (if any) and any OR or AND statements as needed. For DBMS's that allow JOINS, OOB also allows SIMPLE, INNER and OUTER JOINS to be performed.
5. Complex queries can be set up using the simple Query Designer. However, the “New Query (SQL view)” tool enables the student to set up an SQL query in the SQL format of the DBMS. In other words, if a DBMS has a particular SQL syntax, the student can bypass the standard SQL statements generated by OpenOffice.org Base and execute the specific SQL query directly on the DBMS. This is particularly useful for students getting to grips with the details of a particular flavor of SQL. (Some general information can be found at: http://documentation.openoffice.org/HOW_TO/data_source/link_tablehtml/link.html)

4.4 Reports

OpenOffice allows the creation of reports that may contain SQL queries. (Select File > Autopilot > Report). The actual query is contained inside the report, which makes it easily transportable. For example, a report can be kept on the main database server and accessed from any client that has OpenOffice installed; effectively giving the DBA a set of “roving” queries that can be run from anywhere on the LAN. As OpenOffice documents are stored as zipped XML files, it is quite feasible to open the XML file with a text editor and see the actual data. The advantage of this is that the SQL query is visible, so the DBA can spot mistakes and fix the report. In other words, the result of the query is open and visible.

(OpenOffice.org has announced that they will support the new OpenDocument file formats. For details on the new standards, see http://www.oasis-open.org/specs/index.php#opendocumentv1.0.)

Figure 3 shows the actual OpenOffice document, showing the embedded query, using Notepad on Windows XP. The query is highlighted. The DBA should be able to see exactly what the query did, and can therefore verify the accuracy of the data on the report. This is very useful in situations where data has been corrupted; clearly if the query is correct but the report delivers “funny” results, then the data is suspect.
The output of running a report is illustrated in Figure 4. Note that this report was generated by connecting OOB to a Microsoft Access database; in particular, the venerable "Northwind" database. This figure also illustrates one of the very useful aspects of OOB's report writer: the ability to create ad-hoc reports.

5. Evaluation and Conclusion

Rob and Coronel (2004, pp. 712 - 757) give a comprehensive view of the tasks of the DBA. They cover topics such as database creation, configuration, security, storage and capacity management, backup and restore; all of which are handled by the DBA tools of the particular DBMS. Clearly the DBMS specific tools provide the "heavy duty" functionality needed by the DBA. Rob and Coronel go on and explain the various DBA functions by using the Oracle Enterprise Manager tools. This of course means that the DBA needs to have these tools installed on a workstation, which in turn has certain licensing implications.

However, clearly OOB cannot be used as a comprehensive DBA tool, replacing the particular DBMS tools. OOB is useful for the "quick-and-dirty" type of activities is easy and straightforward, but for detailed and "heavy duty" work the DBMS tools are obviously superior. The big advantage of OOB is that it can be installed on any client anywhere on a LAN and used for these simple tasks. For example, a database table can be "dragged and dropped" into an OpenOffice spreadsheet for further analysis: Figure 5
shows dropping an MS Access table in Windows XP into an OpenOffice spreadsheet. For situations where the DBA needs to analyze data in a table or do consistency checking, this facility is not only easy to use, but also very convenient.

Figure 5. Converting an MS Access table

Of course, one could achieve a similar effect by using Microsoft Access as a front-end to SQL Server, but the cost to do this makes it impractical for students. For students, using OOB at home with (say) MySQL has the added benefit that they can practice their skills without incurring any costs. (Note: OOB contains its own programming language, called OO Basic. It is quite feasible to build a database application system using OO Basic + OOB + MySQL, but that activity falls outside the scope of this report. For more, see http://www.ooomacros.org/ ). Figure 6 shows an example of StarBasic code.

Figure 6. StarBasic code

Students found using OpenOffice Base very easy to learn and were surprised by the capabilities of OOB. Doing simple tasks, such as creating tables, setting attributes of fields, creating SQL queries and creating reports were found to be much quicker in OOB than in the particular DBMS tools. Using an OOB report to extract data and use that as a backup was one technique discussed.

Two students used OOB in their Industry Projects to extend the usability of commercial packages. One student discovered that he could access a commercial DBMS via OOB and actually fix some erroneous data. In addition, the student developed some reports and added to the functionality of the commercial package in a short space of time.
Another student used OOB to understand the structure of a commercial package, and this helped him develop an interface with some home-grown code. In both cases the students were able to enhance the commercial software and provide users with extra functionality, something which the students and the stakeholders greatly appreciated. Whilst OOB should not be seen some sort of universal DBA tool, it is clearly very useful.

As is the case with many open source packages, development of the OOB package is quite fast. During the course of this research the new version of OOB became available, which includes a number of enhancements. A JAVA based connector (JDBC) is included that can be used with many DBMS's rather than ODBC. More significantly is the inclusion of the HSQLDB database engine which will enable developers to build database applications directly from OpenOffice; much like Microsoft (R) Access. (See http://hsqldb.org/ for details). OpenOffice 2.0 is now -- November 2005 -- available from http://www.openoffice.org as a free download. But discussing the new capabilities calls for a new article!

Acknowledgements

This article represents a reviewed and revised version of a paper presented at the 18th Annual NACCQ Conference (van Aardt, 2005).

References


Additional Sources

http://www.openoffice.org/welcome/support.html
http://user-faq.openoffice.org/faq/ar01s10.html
http://documentation.openoffice.org/HOW_TO/data_source/link_tablehtml/link.html
http://dba.openoffice.org/FAQ/FAQ.html
http://www.oooauthors.org/FAQs/Database

Appendix

odbcinst.ini:

```
[MySQL]
Description = MySQL driver
Driver = /usr/lib/libmyodbc.so
FileUsage = 1
```

don.ini:

```
[MySQL-test]
```
Description = MySQL database test
Driver = MySQL
Server = localhost
Database = test
Port = 3306

Note: On a typical Linux box the MySQL documentation is in

Copyright © 2005 van Aardt, A.

The author(s) assign to NACCQ and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The author(s) also grant a non-exclusive licence to NACCQ to publish this document in full on the World Wide Web (prime sites and mirrors) and in printed form within the Bulletin of Applied Computing and Information Technology. Authors retain their individual intellectual property rights.

Copyright ©2005 NACCQ.
Krassie Petrova, Michael Verhaart & Tim Hunt (Eds.)
An Open Access Journal, DOAJ # 11764120 , ( ✓zotero)