# Comparison of Selection Criteria of Open Source DBMS Against Proprietary DBMS

**By: Sohail Zaffar Khan**

**Master's Thesis in Advanced Financial Information Systems**

**Swedish School of Economics and Business Administration**

**2006**

**HANKEN - Swedish School of Economics and Business Administration**

| | |
|---|---|
| **Department:** Accounting | **Type of Work:**<br>Master of Science Thesis |
| **Author:** Sohail Zaffar Khan | **Date:** 06.11.2006 |
| **Title of Thesis:**<br>**COMPARISON OF SELECTION CRITERIA OF OPEN SOURCE DBMS AGAISNT PROPRIETARY DBMS** | |

**Abstract:**

The purpose of this research paper is to compare the selection criteria of Open Source Database Management System (OS DBMS) against proprietary DBMS.

The theoretical part of thesis shows that DBMS is the important part of organizational IT strategy. It does not accordingly denote which DBMS product can be regarded to be the absolute best product, but instead the actual needs which organizations could be considered to have. In this paper, twelve important factors are identified which are influential/decisive when choosing OS DBMS/Proprietary DBMS software based on the relevant literature. The research also explains the concept of Open Source Software (OSS) and proprietary software and compared pros and cons of both types.

Main subject of the empirical study is to analyze and compare data statistically, identify difference in DBMS selection criteria practices and examine the validity of the DBMS selection criteria practices in the case of Pakistani companies. The study was conducted in a form of an online survey, which was targeted toward the companies who implemented DBMS in Pakistan.

Several criteria were chosen to be important to the companies and these criteria were given different rank according to their significance from each company. All companies' response are analyzed and ranked by these criteria. The results indicate that importance of some selection criteria practices differ in the case of the open source DBMS against proprietary DBMS.

**Keywords:** DBMS, FOSS, OS, FS, proprietary software, important factors in DBMS selection criteria, online questionnaire, Pakistan.

**Acknowledgment**

The study was not possible without the idea and guidance of **Mr. Niranjan Rajhani**, System Administrator, Hanken School of Economics and Business Administration.

**Mr. Omer Farooq**, Research Assistant, Economics department has supported and encouraged me in conducting the present studies. I pay thanks for his professional support and guidance.

Linux Pakistan and Pakistan Open Source organization provided me valuable literature, data and guidance on the subject required for the completion of the project.

My advisor **Mr. Anders Tallberg** was a source of inspiration, knowledge, guidance and moral support for me in conducting the present study. I pay thanks for his professional support and guidance.

The thesis is dedicated to my father **Mr. Muzaffar Khan.**

**TABLE OF CONTENTS**

**LIST OF FIGURES**

## LIST OF TABLES

# 1. INTRODUCTION

## 1.1 Research Background

In the early days of computing, software was generally free, and it was something that was shared among researchers and developers, who were usually eager to improve it. However, that situation changed as computers became more common, and the production of proprietary software became an excellent business model for many companies. In recent years Open Source Software (OSS) has become a major interest both for the software industry and for economic theory. Some companies have begun to realize that Open Source Software (OSS) can also be highly profitable. The most outstanding example of this is IBM, which continues to reap high returns from its approximately one billion-dollar investment in Linux (Linux Information Project, 2005).

Some industry observers think that the role of proprietary software will decrease in the future because of the growing competition from Open Source Software (OSS). Business today demands a different kind of data management solution. Performance, scalability, and reliability are essential, but businesses now expect more from their key IT investments. This means that when selecting software organizations must focus more on surrounding services such as support, specific service, implementation and migration. This entails a more function oriented procurement of the software. The organizations strive to find the most cost-efficient alternative, no matter if it is proprietary software or not (Linux Information Project, 2005).

According to Redners Tryckeri (2003) report “Free and Open Source Software”, situation in the developing countries like Pakistan can be summed up in the following manner: There is a lack of resources. Information Communication and Technology (ICT) is an important area, not the most prioritized, but without a development in IT, the digital gulf between the developed and developing world will contribute to an increase in poverty. In order to get development started it is therefore necessary to use the free resources which can be found on the Internet.

Open Source Software (OSS) is the only possibility for developing countries to have legal access to software products. Another argument for Open Source Software (OSS) is that developing countries benefit by favoring their own domestic software industry. Even the anti-piracy policies of proprietary software producers, contribute to support for OSS. Open Source Software (OSS) is subsequently an important strategic question for the development of these countries (Redners tryckeri, 2003)

When directly comparing Open Source (OS)/Proprietary Database Management System (DBMS), the former stands out of course as a much more mature product with more functions and with a more well thought through user interface (Redners tryckeri, 2003). It is important to recognize that each organization has a unique set of requirements for a Database Management System (DBMS) software application and it is not possible to provide a single list of requirements appropriate for every organization. Organizations differ in terms of size, IT infrastructure, communication style, required level of security, cash position, scorecard design, IT literacy, in-house capabilities, etc. All these aspects affect the selection criteria of a software solution. But then organizations have to weigh each of the criteria to reflect their unique set of requirements. Before organizations start considering the software solution they should already have successfully developed a robust balanced scorecard (Neely et al., 2002; Anonymous, 2002; Kaplan and Norton, 2000a).

However, various authors draw attention to the fact that software is only a tool and not a substitute for the initial hard work of strategic analysis (Marr, 2001; Sharman and Kavan, 1999). Organizations implement the software with one precondition that employees will use it. Therefore it is important to pick a solution that meets the requirements of the organization in order to ensure the usage and subsequent success of software (Marr et al., 2000).

## 1.2 Research Objective

The study will compare selection criteria practices for Open Source Database Management System against proprietary Database Management Systems in developing countries. This study is not against proprietary DBMS software. The ground realities of developing countries point to OSS being more appropriate solution. The following question was posed to the participants:

- Define the importance of criteria which are influential/decisive when choosing OS DBMS/Proprietary DBMS software.

## 1.3 Structure of the Thesis

This thesis is composed of 6 chapters. Chapter 1 describes the research background, the research objective(s) and the structure of thesis.

Chapter 2 is; Database Management Systems. Chapter 2 starts by giving overall picture what is Database Management System and its importance in organizational IT strategy. Further it explores major characteristics, components, architecture, functionality and current and emerging trends.

Chapter 3 is; Free and Open Source software vs Proprietary software. This chapter elaborates the concept of Free and Open Source and proprietary software. Advantages and disadvantages of both kind of software are compared. Some of them are common to both, others are specific to one. It also briefly discusses different DBMS in both categories.

Chapter 4 is; Important Factors in Selection Criteria of DBMS. This chapter looks at the relevance of important factors in Database Management System decision. It also includes in-depth information to evaluate each factor. All database software selection goes through evaluation phase in spite of open source/proprietary software.

Chapter 5 is the Methodology. It is composed of a hypothesis, research methodology and analysis of data. It describes the survey's methodology as well as its effectiveness. It also contains important information about how the survey results can be interpreted.

Chapter 6, the Conclusion; Summarizes key points of the research and provides suggestions for further research.

# 2. DATABASE MANAGEMENT SYSTEMS

## 2.1 Objective and Structure

The objective of this chapter is to explain Database Management Systems, identify major characteristics, components, architecture, functionality, trends and what a DBMS must furnish. This chapter contains general information on the concepts and principles needed to understand the structure of a database that is created and maintained. The opening section of this chapter starts with the definition of DBMS (section 2.2) and then the section that follows (section 2.3) describes the components of DBMS. Section 2.4 explores in-depth information about the DBMS architecture and structure. Section 2.5 lists functionality that should be provided by full scale DBMS and section 2.6 and 2.7 briefly explain the current and emerging trends of DBMS.

## 2.2 Database Management System

A Database Management System (DBMS) is the software that enables users to define, create, maintain and control access to the database. DBMS is the software that interacts with the users' application program and the database. Typically, a DBMS provides the following facilities (Connolly, Begg, ch: 1):

- It allows users to define the database, usually through a Data Definition Language (DDL). The DDL allows users to specify the data types and structures and constraints on the data to be stored in the database.

- It allow user to insert, update, delete and retrieve data from the database, usually through a Data Manipulation Language (DML). Having a central repository for all data and data descriptions allow the DML to provide a general inquiry facility to this data, called a query language. The most common query language is the Structured Query Language (SQL).

- It provides controlled access to the database. For example it may provide:

- A security system, which prevents unauthorized users accessing the database and also prevent unauthorized transactions.
- An integrity system, which maintains the consistency and validity of stored data.
- A concurrency control system, which allows multiple users to access the database simultaneously.
- A recovery control system, which provides a method for backing up and restoring the data in the database, in the event of some type of system failure.
- A user accessible catalog, which contains descriptions of the data in the database.

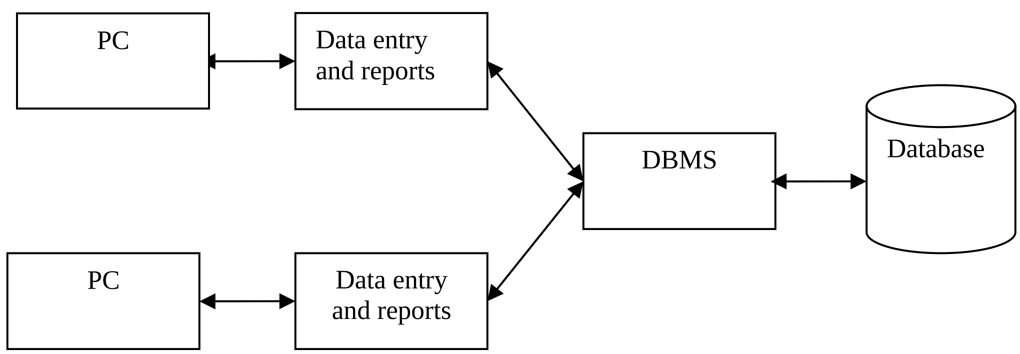


**Figure 2.1: Database Processing**

## 2.3 Components of DBMS Environment

### 2.3.1 Hardware

The DBMS and the application require hardware to run. The hardware can range from a single personal computer, to a single mainframe, to a network of computers. The particular hardware depends on the organization requirements and the DBMS used. Some DBMSs run only on particular hardware or operating systems, while others run on a wide variety of hardware and operating systems. A DBMS requires a minimum amount of main memory and disk space to run, but this minimum configuration may not necessarily give acceptable performance (Connolly, Begg, ch: 1).

### 2.3.2 Software

The software component comprises the DBMS software itself and the application program, together with the operating system, including network software if the DBMS is being used over a network. Typically application programs are written in a third-generation programming language (3GL), such as ‘C’, C++, Java, Visual Basic, COBOL, Fortran, Ada, or Pascal, or using a fourth-generation language (4GL), such as SQL, embedded in a third generation language. The target DBMS may have its own fourth-generation tools that allow rapid development of applications through the provision of non-procedural query languages, reports generators, forms generators, graphics generators, and application generators. The use of fourth generators tools can improve productivity significantly and produce programs that are easier to maintain (Connolly, Begg, ch: 1).

### 2.3.3 Data

The most important component of the DBMS environment certainly from the end users’ point of view is the data. Figure 2.2 shows that data act as bridge between the machine components and the human components. The databases contain the operational data and the meta data, the ‘data about data’ (Connolly, Begg, ch: 1).

### 2.3.4 Procedure

Procedure refers to the instruction and rules that govern the design and use of the database. The user of the system and the staff that manage the database require documented procedures on how to use or run the system. These may consist of instructions how to (Connolly, Begg, ch: 1):

- Log on to the DBMS.
- Use particular DBMS facility or application program.

- Start and stop the DBMS.
- Retrieve, insert, delete and modify information stored in the database
- Make backup copies of the database.
- Handle hardware or software failures. This may include procedures on how to identify the failed component, how to fix the failed component and following the repair of the fault, how to recover the database.
- Change the structure of a table, reorganize the database across multiple disks, improve performance, or archive to secondary storage.

### 2.3.5 Database Administrator

The database administrator (DBA) is responsible for overall control of the database system. Responsibilities include (Connolly, Begg, ch: 1):

- Installing new DBMS and upgrading existing DBMS. The DBA must be conversant with installation and upgrade issues, i.e. problems, requirements, etc.
- Deciding the information content of the database, i.e. identifying the entities of interest to the enterprise and the information to be recorded about those entities.
- Deciding the storage structure and access strategy, i.e. how the data is to be represented by writing the storage structure definition. A DBA must be equipped to eliminate and reduce outages planned and unplanned.
- Liaising with users, i.e. data must be available to all who need it when they need it.
- Defining authorization checks and validation procedures. Authorization checks and validation procedures are extensions to the conceptual schema and can be specified using the DDL.
- Defining a strategy for backup and recovery. DBA needs to implement recovery procedures that will reduce downtime associated with the failure.
- Monitoring performance and responding to changes in requirements, i.e. DBA maintains the information an organization needs to be successful.

- Assigning users to databases and determining the proper security and access level for each user.

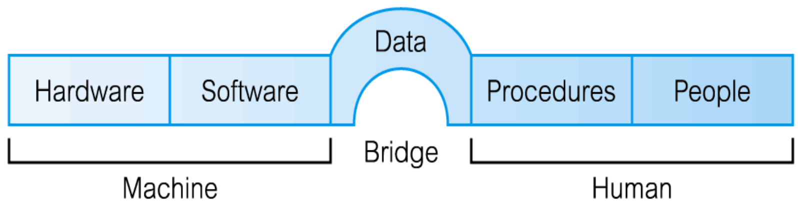


**Figure 2.2 DBMS environment**

*Source: Connolly, Begg, ch: 1*

### 2.4 Database Architecture

There are two different ways to look at the architecture of a DBMS: the logical DBMS architecture and the physical DBMS architecture. The logical architecture deals with the way data is stored and presented to users, while the physical architecture is concerned with the software components that make up a DBMS.

#### 2.4.1 Logical DBMS Architecture

All DBMSs do not confirm to the same architecture. Since a database is a shared resource, each user may require a different view of the data held in the database. To satisfy these needs, the architecture of most commercial DBMSs available today is based to some extent on the American National Standards Institute (ANSI) Standards Planning and Requirement Committee (SPARC) so called ANSI-SPARC architecture. The ANSI-SPARC architecture is divided into three levels: external level, conceptual level and internal level. (Connolly, Begg, ch: 2).

The three-level architecture forms the basis of modern database architectures. The objective of the three-level architecture is to separate each user's view of the database from the way the database is physically represented. The diagram below shows the logical architecture for a typical DBMS.

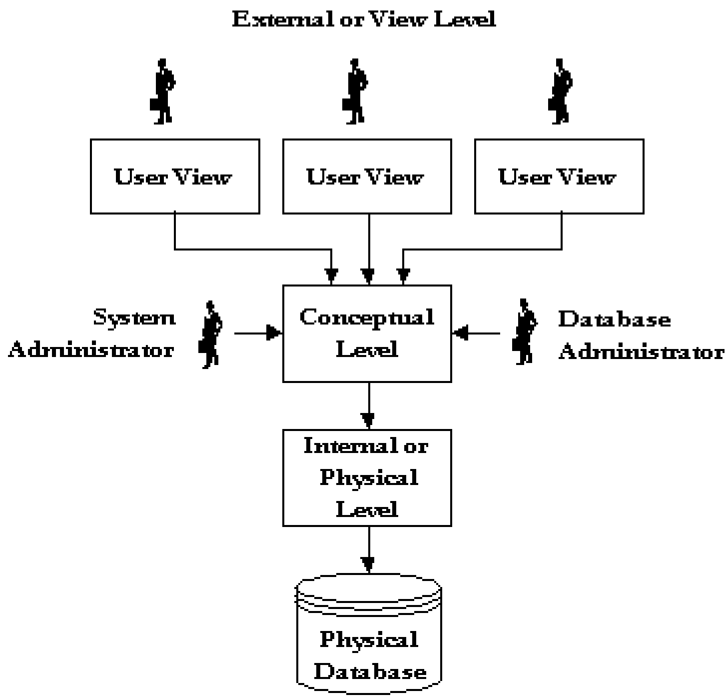


**Figure 2.3 Logical DBMS Architecture**

*Source: www.dbmaker.com.tw*

### 2.4.1.1 External Level

The external level is concerned with the way individual users see the data. Each user has a view of the data represented in a form that is familiar for that user. The external view includes only those entities, attributes, and relationships in the 'real world' that the user is interested in. Other entities attribute, or relationships that are not of interest may be represented in database, but many users of the database system will not be concerned with all this information.

In addition, system may provide many views for the same database. For example, one user may view dates in the form (day, month, year), while another may view dates as (year, month, day). Some view might include derived or calculated data i.e. data not actually stored in the database as such, but created when needed (Connolly, Begg, ch: 2).

#### 2.4.1.2 Conceptual Level

The conceptual level can be regarded as community users view a formal description of data of interest to the organization, independent of any storage considerations. The first stage in the design of a database is to define the conceptual view, and a DBMS provides a data definition language for this purpose. It is the conceptual level that allows a DBMS to provide data independence. There is only one conceptual view per database. The conceptual level represents (Connolly, Begg, ch: 2).

- All entities, their attributes, and their relationships
- Multiple occurrences of multiple types of conceptual record
- A general a view of the data as it actually is
- Semantic information about the data
- Authorization and validation procedures

One conceptual view represents the entire database. The conceptual level supports each external view, in that any data available to a user must be contained in, or derivable form, the conceptual level. However this level must not contain any storage dependent details.

#### 2.4.1.3 Internal Level

The internal level describes how the data are stored in the database, and what relationships exist among that data. The internal level covers the physical implementation of the database to achieve optimal runtime performance and storage space utilization. The internal level is the one closest to physical storage, and it provides a low level description of the physical database, and an interface between the operating system's file system and the record structures used in higher levels of abstraction. It covers the data structures and file organizations used to store data on storage devices. It interfaces with the operating system access methods to place the data on the storage devices, build the indexes, and retrieve the data and so on. The internal level is concerned with such things as (Connolly, Begg, ch: 2):

- Storage space allocation for data and indexes
- Defines the various types of stored record (with stored sizes for data items).
- Record placement.
- Data compression and data encryption techniques.

### 2.4.2 Physical DBMS Architecture

The physical architecture describes the software components used to enter and process data, and how these software components are related and interconnected. Although it is not possible to generalize the component structure of a DBMS, it is possible to identify a number of key functions that are common to most DBMS. The components that normally implement these functions are shown in the figure 2.4, which depicts the physical architecture for a typical DBMS. Physical DBMS architecture can be broken down into two parts: the back end and the front end (www.dbmaker.com.tw).

The back end is responsible for managing the physical database and providing the necessary support for the internal, conceptual, and external levels. Other benefits of a DBMS, such as security, integrity, and access control, are also the responsibility of the back end.

The front end is just any application that runs on top of the DBMS. These may be applications provided by the DBMS vendor, the user, or a third party. The user interacts with the front end, and may not even be aware that the back end exists.

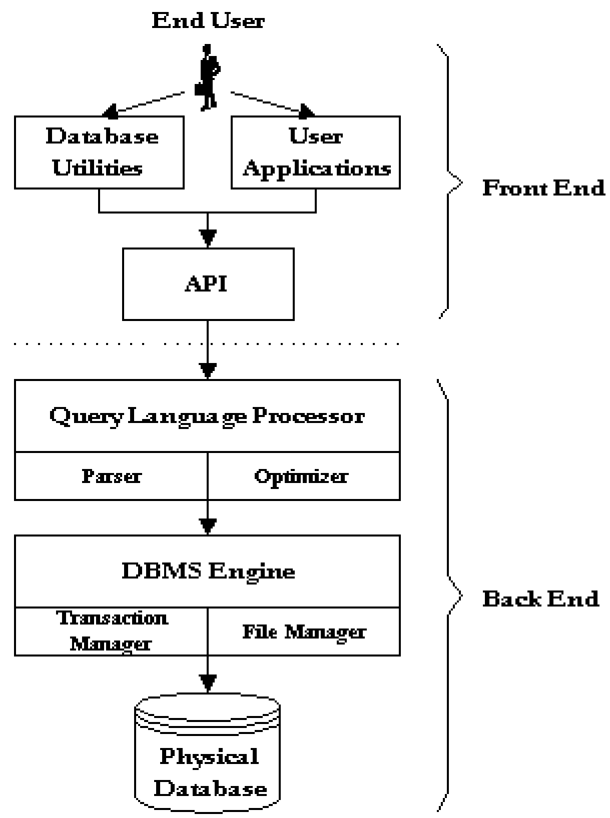


**Figure 2.4 Physical DBMS Architecture**

*Source: www.dbmaker.com.tw*

### 2.5 DBMS Functionality

Codd (1982) lists eight services that should be provided by any full-scale DBMS. The listed features of a DBMS provide several benefits over the older file-based computer storage systems they replaced.

#### 1. Data Storage, Retrieval, and Update

A DBMS must furnish users with the ability to store, retrieve, and update data in the database. This is the fundamental function of DBMS.

#### 2. A User Accessible Catalog

A DBMS must furnish a catalog in which description of data items are stored and which is accessible to users. A catalog is expected to be accessible to users as well as to the DBMS. A system catalog, or data dictionary, is a repository of information describing the data in the database. It is, the ‘data about the data’ or meta-data. The amount of information and the way of information is used vary with the DBMS. Typically the system catalog stores (Connolly, Begg, ch: 2):

- Names, types and sizes of data items.
- Names of relationships.
- Integrity constraints on the data.
- Names of authorized users who have access to data.

**3. Transaction Support**

A DBMS must furnish a mechanism, which will ensure that either all the updates corresponding to a given transaction are made or that none of them is made. A transaction is a series of actions, carried out by a single user or application program, which access or changes the contents of the database.

**4. Concurrency Control Services**

A DBMS must furnish a mechanism to ensure that the database is updated correctly when multiple users are updating the database concurrently. One major objective in using a DBMS is to enable many users to access shared data concurrently. Concurrent data is relatively easy if all users are only reading data, as there is no way that they can interfere with one another. However when two or more users are accessing the database simultaneously and at least one of them is updating data, there may be interference that can result in inconsistencies. To avoid inconsistencies, a DBMS must provide a mechanism for blocking access by others to the data that is being accessed by the transaction. Once a lock is set, it will remain in force until the transaction is committed or rolled back. The database must ensure that when multiple users are accessing the database, interference cannot occur (Connolly, Begg, ch: 2).

**5. Recovery Services**

A DMBS must furnish a mechanism for recovering the database in the event that the database is damaged in any way. The DBMS should provide a method for backing up and in the event of some type of system failure, restoring the data in the database.

Backups are important because no system is free from failures. It is necessary to restore and recover the data quickly to resume operations. The key to success in this situation is well defined backup and recovery strategy. If transaction fails then the database has to be returned to a consistent state. This may be result of system crash, media failure, hardware or software error causing the DBMS to stop, or it may be the result of the user's detecting an error during the transaction and aborting the transaction before it completes. In all these cases, the DBMS must provide a mechanism to recover the database to a consistent state (Connolly, Begg, ch: 2).

**6. Authorization Services**

A DBMS must furnish a mechanism to ensure that only authorized users can access the database. The DBMS should provide facilities for protecting the security and privacy of data from unauthorized users. Authorization control must guarantee that only authorized users perform operations they are allowed to perform on the database. Authorizations must be refined so that different users have different rights on the same objects.
(www.dbmaker.com.tw).

For example, organization may want only branch managers to see salary related information for staff and prevent all other users from seeing this data. The term security refers to the protection of database against unauthorized access, either intentional or accidental.

**7. Support for Data Communication**

A DBMS must be capable of integrating with communication software. Most users access the database from workstations. Sometimes these workstations are connected directly to the computer hosting the DBMS. In other cases, the workstations are at remote locations and communicate with the computer hosting the DBMS over a network. In either case, the DBMS receives requests as communication messages and responds in a similar way. A Data Communication Manager (DCM) handles all such transmissions. Although the DCM is not part of the DBMS, it is necessary for the DBMS to be capable of being integrated with a variety of DCMs if the system is to be commercially viable (Connolly, Begg, ch: 2).

**8. Integrity Services**

A DBMS must furnish a means to ensure that both the data in the database and changes to the data follow certain rules. Database integrity refers to the correctness and consistency of stored data. It can be considered as another type of database protection. The DBMS should guarantee that data in the database does not have invalid values, and that there is no inconsistency in related data. This prevents from accidentally entering invalid data or performing operations that can violate some dependency between data (www.dbmaker.com.tw).

## 2.6 Current Trends

### 2.6.1 Distributed DBMS

A Distributed Database Management System (DDBMS) consists of a single logical database that is split into a number of fragments. Each fragment stored on one or more computers under the control of a separate DBMS, with the computers connected by a communication network. Each site is capable of independently processing user requests that require access to local data (that is, each site has some degree of local autonomy) and is also capable of processing data stored on other computers in the network.

Users access the distributed database via applications. Applications are classified as those that do not require data from other sites (logical application) and those that do require data from other sites (global application). DBMS require having at least one global application. A DDBMS therefore has the following characteristics (Connolly, Begg, ch: 22).

- Remote database access.
- A collection of logically related shared data.
- The data is split into a number of fragments.
- Fragments are allocated to sites.
- A communications network links the sites.
- Multiple points of control.
- The data at each site is under the control of a DBMS.
- Some support of concurrency control and recovery of distributed transaction
- The DBMS at each site can handle local applications, autonomously.
- Each DBMS participates in at least one global application.
- Support for database administration and control. This feature includes tools for monitoring, gathering information about database utilization etc.

### 2.6.2 OODBMS

According to Maier (1990) an Object-Oriented DBMS must satisfy:

- It must provide database functionality
- It must support object identity
- It must provide encapsulation.
- It must support object with complex state.

The main advantage of OODBMS is that it allows the real world to be modeled more closely. OODDBMS allow new data types to be built from existing types. There have been a number of benchmarks that have suggested that OODBMS provide significant performance improvement over relational DBMS.

In comparison to relational DBMS the use of OODBMS is still relatively limited. There is no universally agreed data model and lack of standards for an OODBMS.

## 2.7 Emerging Trends

### 2.7.1 Web Technology and DBMSs

Some of the most important requirements for the integration of database applications with the web are ideals and not fully achievable at the present time. Not in any ranked order, the requirements are as follows (Connolly, Begg ch: 28):

- The ability to access valuable corporate data in a secure manner.
- Data and vendor independent connectivity to allow freedom of choice in the selection of the DBMS now and in the future.
- The ability to interface the database independent of any proprietary Web browser or Web server.
- A connectivity solution that takes advantage of all the futures of an organization's DBMS
- An open architecture approach to allow interoperability with a variety of systems and technologies
- Support for session and application based authentication.
- Acceptable performance
- Minimal administration overhead
- A set of high level productivity tools to allow applications to be developed, maintained, and deployed with relative ease and speed.

Some advantages and disadvantages of web DBMS approach are shown in table 2.

**Table 2.1 Web-DBMS approach**

| Advantages | Disadvantages |
| --- | --- |
| Simplicity | Reliability |
| Platform independence | Security |
| Graphical user interface | Cost |
| Standardization | Scalability |
| Cross platform support | Limited functionality of HTML |
| Transparent network access | Statelessness |
| Scalable deployment | Bandwidth |
| Innovation | Performance |

### 2.7.2 Data warehousing

The goal of data warehousing is to integrate enterprise wide corporate data into a single repository from which users can easily run queries, produce reports, and perform analysis.

The successful implementation of data warehouse can bring major benefits to an organization. However, a study by international data corporation (IDC) in 1996 reported that average three year returns in investment in data warehousing reached 401%, with over 90% of the companies surveyed achieving over 40% ROI, half the companies achieving over 160% ROI, and a quarter with more than 600% ROI (IDC, 1996). Companies gain competitive advantage by allowing decision makers access to data that can reveal previously unavailable, unknown and untapped information on, for example customers, trends and demands. Data warehouse integrates data from multiple incompatible systems into a form that provide one consistent view of the organization. By transforming data into meaningful information, a data warehouse allows business managers to perform more substantive, accurate and consistent analysis (Connolly, Begg ch: 30).

.

# 3. FREE AND OPEN SOURCE VS PROPRIETARY SOFTWARE

## 3.1 Objective and Structure

This chapter discusses the concept of Free and Open Source and proprietary software. Section 3.1 starts with objective and structure of chapter. Section 3.2 include a brief history its inherent strengths and looks in-depth advantages and disadvantages of OSS. Section 3.3 discusses a most popular OS DBMS. Section 3.4 explains proprietary software with advantages and disadvantages. Section 3.5 discusses a most popular proprietary DBMS

## 3.2 Overview of Free and Open Source Software

Free and Open Source Software (FOSS) programs are programs whose licenses permit users the freedom to run the program for any purpose, to study and modify the program, and freely redistribute copies of the original or modified program (Redners tryckeri, 2003).

In this research, I choose to treat both Free Software and Open Source Software in the same category, hence the abbreviation is OSS.

### 3.2.1 Free Software

The Free Software Foundation (FSF), established in 1985, is dedicated to promoting computer users rights to use, study, copy, modify, and redistribute computer programs.

The FSF also helps to spread awareness of the ethical and political issues surrounding freedom in the use of software. The definition of free software by FSF is:

"*Free software is a matter of the users' freedom to run, copy, distributes, study, change and improves the software. More precisely, it refers to four kinds of freedom, for the users of the software:*

- *The freedom to run the program, for any purpose (freedom 0).*

- *The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.*
- *The freedom to redistribute copies so you can help your neighbor (freedom 2).*
- *The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.*

*A program is free software if users have all of these freedoms. Thus, you should be free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission.*"(www.fsf.org)

Free Software is different from Freeware, Shareware, Adware, Spyware or Crippleware, which are all types of proprietary software made available at no price, providing various degrees of freedom of use, but in most cases not other freedoms as described by FSF.

### 3.2.2 Open Source Software

OSS is software where the source code (the language in which the program is written) is freely distributed with the right to modify the code, and on the condition that redistribution is not restricted, and indeed is obtainable for no more than the reasonable cost of production. Open Source Initiative (OSI) is a non-profit corporation dedicated to managing and promoting the Open Source definition for the good of the community.

"*The Open Source Initiative, a group of developers who disseminate information on the benefits of open source has posted on its web site a "meta-definition" of basic conditions which they feel should be included in an OSS license. These include:*

- *Allowing free redistribution of the software without royalties or other fees to the author.*
- *Requiring that source code be distributed with the software or otherwise made available for no more than the cost of distribution.*
- *Allowing anyone to modify the software or derive other software from it, and to redistribute the modified software under the same license terms*." (www.opensource.org)

Any software, which is distributed under a license, which conforms to these requirements, is open source software, according to the Open Source Initiative. The Open Source Initiative (OSI) tends to differ with FSF philosophy, and instead emphasizes the practicality and technical superiority of a method of software development (www.opensource.org).

According to open source organization the distribution terms of Open Source Software must comply with the following criteria (www.opensource.org):

**1. Free Redistribution**

There shouldn't be any restriction on any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require fee for such sale.

**2. Source Code**

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program.

**3. Derived Works**

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

**4. Integrity of the Author's Source Code**

The license may restrict source code from being distributed in modified form only if the license allows the distribution of patch files with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

**5. No Discrimination against Persons or Groups and Fields of Endeavor**

The license must not discriminate against any person or group of persons.

**6. No Discrimination against Fields of Endeavor**

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

**7. Distribution of License**

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

**8. License Must Not Be Specific to a Product**

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

**9. License Must Not Restrict Other Software**

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open source software.

**10. License Must Be Technology-Neutral**

No provision of the license may be predicated on any individual technology or style of interface.

### 3.2.3 Advantages of OSS

Motivations for using and developing OSS are mixed, ranging from philosophical and ethical reasons to pure practical issues. In this section, some practical advantages will be introduced.

- The availability of the source code and the right to modify it is very important. It enables the unlimited tuning and improvement of a software product. It also makes it possible to port the code to new hardware, to adapt it to changing conditions, and to reach a detailed understanding of how the system works. The right to modify and improvements to the code permits all the advantages due to the modifiability of the software to be shared by large communities (Redners tryckeri, 2003).

- The right to use the OSS in any way. A large number of users, which helps in turn to build up a market for support and customization of the software, which can only attract more and more developers to work in the project. As more developers look and work on the common source code, so there is higher quality and more rapid innovation.

- There is no one with the power to restrict in a unilateral way how the OSS is used, even in a retroactive way. For instance, when a proprietary software vendor decides not to upgrade some software product for some old platform. In this case, users can only stick to the old version of the software, or switch to another product. If OSS is used, users can also fund some development for the desired platform, or look for other vendors to provide the upgrades (of the very same product) (Jesus M. Gonzalez-Barahona 2004).

- There is no single entity on which the future of the OSS depends. If the group or company that originated the code decides to stop development, it is always possible to fund another software group to continue the maintenance and improvement, without legal nor practical limitations (Redners tryckeri, 2003).

- There are no black boxes possible in OSS. This point is so important that open source is now considered by many experts as one of the necessary conditions for dependable applications. There are several reasons for this importance. One of them is related to the dependability of the services provided by given software. Source code availability makes it possible to perform a thorough inspection and verify the correctness of the algorithm and the implementation scheme used. (Jesus M. Gonzalez-Barahona 2004).

- There is always the possibility of forking, or creating an alternative code base if the current one is in some way perceived as wrongly managed. A fork is a subdivision of the code base in two different parts, each managed by a different group. A fork happens for technical or license reasons, for example because a particular release is made under a non free license, the previous one is used as a base for subsequent free releases. Technical motivations are common, because there are sometimes many different ways to perform a task, and it is not possible to decide which is better. After a fork, both branches tend to compete for the user base with very similar products (Jesus M. Gonzalez-Barahona 2004).

- No per-copy fees can be asked for modified versions, and anyone can use the current code base to start new projects. Working knowledge can be gathered at a minimal cost.

- There is no single commercial entity of OSS pushing for precise delivery dates or features that must be supported. Usually OSS is delivered when the development team feels that its quality is good enough. This means that OSS usually does not need as many service packs, updates and reduce the maintenance cost. Of course this could be turned into disadvantage if a product is indefinitely delayed, or if some feature is missing one release after another. But in this case, the competition between projects may help. If a project starts failing to meet the expectations of its users, it often happens that a new project is forked, using the same code base, to fill this gap. This happens especially if a market exists for some new features, or for better quality versions of the application (Jesus M. Gonzalez-Barahona 2004).

- OSS provides a new forum for democratic action. As individuals and companies decide where to make improvements in the system, the collective desires of the community determine the overall direction of progress, and yet without compelling anyone. People with opinions about what direction is best can urge others to join in, request help, and in this way influence the overall direction of progress. A participating organization can reap the benefits of expertise not in its employ (Jesus M. Gonzalez-Barahona 2004).

- Establishment of a local software industry. It is difficult for foreigners to do localization as they do not normally have an intuitive feel for the local language and therefore the language is compromised in most cases. It also provides local control over software appearance and functionality.

- No need for local users to learn English first and then local programmers gain expertise and experience. Localization of applications can be prioritized according to the national needs.

### 3.2.4 Disadvantages of OSS

Of course, OSS development models lead also to the perception of some disadvantages. Some of them are:

- There is no guarantee that OSS development will happen. It is not possible to know if a project will ever reach a usable stage, and even if it reaches it, it may die later if there is not enough interest. Especially when a project is started without strong backing from one or more companies, there is a significant initial gap, when the source base is still immature and the development base is still being built (Jesus M. Gonzalez-Barahona 2004).

- There may be significant problems connected to intellectual property. This point is especially important, because some countries are accepting software and algorithm patents. It is very difficult to know if some particular method to solve a software problem is patented, and so the community can be considered guilty of intellectual property infringement. In other cases, developers consider source code not as an executable device, but a mere description of how a device (the computer) executes, and therefore uphold the idea that source code is not by itself (in absence of an executable program) covered by patent law even in countries where software patents are accepted. In any case, it still leaves problems for the users, who need the executable programs. Although the issue of software patents is a problem for the whole software industry, open source is one of the more clear cases where it can be shown how they harm the regular process of software development (Jesus M. Gonzalez-Barahona 2004).

- It is sometimes difficult to know if an OSS project exists, and its current status. There is not much advertising for OSS, especially for those projects not directly backed by a company willing to invest resources in marketing campaigns. There are only a few clearing houses for OSS and projects, and in many cases they are not really up to date.

- Documentation and user manuals can be hard to follow for non-techies.
- The OSS programs themselves may be extremely powerful once learned, but are not always as intuitive as they could be.
- Customization costs development time and money. There are limited financial incentives for improvements and innovations.
- Most businesses operate on proprietary programs, so sharing information or documents might be difficult.

## 3.3 Open Source DBMS Software

There are two primary open source DBMS competitors MySQL and PostgreSQL.

### 3.3.1 MySQL

MySQL, the most popular open source SQL DBMS, is developed, distributed, and supported by MySQL AB. MySQL AB is a Swedish company founded Swedish speaking Finns, David Axmark and Michael Widenius, that develops supports and markets the MySQL database server globally.

"*MySQL mission is to make superior data management available and affordable for all, and to contribute to building the mission-critical high-volume systems and products of tomorrow. MySQL AB also claims: Today MySQL is the most popular open source database server in the world with more than 4 million installations powering websites, datawarehouses, business applications, logging systems and more." MySQL AB's goal is to broaden MySQL's reach across the enterprise*." (www.MySQL.com)

MySQL AB is second generation open source company, with dual licensing that supports open source values and methodology in a profitable, sustainable business. MySQL runs on many operating systems including Linux, Unix, Mac OS X and Windows.

### 3.3.2 PostgreSQL

The object-relational database management system now known as PostgreSQL is derived from the POSTGRES package written at the University of California at Berkeley. PostgreSQL is an open source descendant of this original Berkeley code. It supports SQL92 and SQL99 and offers many modern features.

PostgreSQL is open source relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.

## 3.4 Proprietary Software

Proprietary software is software, which has been designed and coded by or for a specific person, organization or group of organizations, which hold ownership or intellectual property rights over the software. An individual or a company (usually the one that developed it) owns proprietary software. There are almost always major restrictions on its use, and its source code is almost always kept secret. Such software is usually built to specification by a company, used only by that company, and not available for sale to third parties or the general public. The user of the software has total control over the code and how the software is used. The proprietary software has its source code closed, so that one cannot view or study it, let alone copy, modify and redistribute it.

Some proprietary software comes with source code or provides offers to the source code. Users are free to use and even study and modify the software in these cases, but are restricted by either licenses or non-disclosure agreements to redistribute modifications. Licenses for proprietary software generally only give the user the right to use the software under certain conditions without any access to its source code (Redners tryckeri, 2003).

### 3.4.1 Advantages of Proprietary Software

- Proprietary software is successful business model. It has been very profitable, has attracted lots of investment, and has enabled software companies to create large numbers of well paying jobs. Proprietary software exists to generate revenue. When software generates profit, it enables companies to grow, attracts investment and enables those companies to grow into tremendous sources of innovation and local employment (David A. Wheeler January 5, 2005).

- The proprietary model provides the vendor a guaranteed income which can be used to better service their customers. A proprietary company usually needs to listen to the needs of their customers, and respond and develop accordingly.

- Support and speed of response to problems. If a problem arises with proprietary software, then the company can usually fix it more quickly with the help of consultant than waiting for developer's community.

- It guarantees structured innovation, which is innovation that is planned within a single responsible organization.

- Proprietary software licenses provides protection for intellectual property

### 3.4.2 Disadvantages of Proprietary Software

- It is expensive to license and maintain proprietary software.

- Dependent on proprietary or closed standards. Many technical specifications that are sometimes considered standards are proprietary rather than being open, and are only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright for the specification. In some cases certain customers are allowed to view the source code sometimes for an additional fee but even then they are not able to alter and redistribute the software (Jesus M. Gonzalez-Barahona 2004).

- Proprietary software has little or no local support. This is a very common concern with proprietary software. For instance that a company uses a software product, and relies on the software manufacturer for upgrades and continued development. If the software manufacturer decides to discontinue development of the product, no one has the right to take the program and continue development on it, effectively killing its usability in the market (David A. Wheeler January 5, 2005).

- The quality of the software depends entirely on the owner organization; this can be a problem if the owner organization does not place the same value on quality as the using organization, the owner organization has a different perspective on quality, or if quality resources are not allocated in ways that the using organization needs (Jesus M. Gonzalez-Barahona 2004).
- The high cost of the proprietary software leads to illegal copying of the software. This is a major barrier to access the proprietary software legally in developing countries.

- The local software industry is not developed because of major market share is owned by proprietary software.

- Proprietary software is dominated by the English language and controlled by foreign corporations.

### 3.5 Proprietary DBMS Software

The most common proprietary DBMS are oracle, DB2, MS SQL,

#### 3.5.1 Oracle

Oracle Corporation produces and markets the Oracle DBMS, which many database applications use extensively on many popular computing platforms. Oracle's database products are among the most popular in the world today.

Oracle is the first software company to develop and deploy 100 percent internet enabled enterprise software across its entire product line: database, business applications, and application development and decision support tools. Oracle is the world's leading supplier of software for information management, and the world's second largest independent software company (www.oracle.com)

#### 3.5.2 MSSQL

Microsoft SQL Server is a relational database management system produced by Microsoft. It supports Microsoft's version of Structured Query Language (SQL), the most common database language. It is commonly used by businesses for small- to medium-sized databases, and in the past five year large enterprise databases. Microsoft SQL server competes with other relational database products for this market segment.

#### 3.5.3 DB2

According to IMB DB2 is the database management system that delivers a flexible database platform to build robust on demand business applications. DB2 is developed, distributed, and supported by IBM. A number of open source programming languages like Perl, Python and PHP also support DB2.

# 4. IMPORTANT FACTORS IN SELECTION CRITERIA OF DBMS

## 4.1 Objective and Structure

Chapter 4 details important factors in selection criteria of DBMS software. This chapter looks at the relevance of important factors in DBMS decision. It also includes the framework for evaluating DBMS. This chapter focuses on one particular challenge facing many organizations – how to evaluate DBMS in general and then specifically OS DBMS against proprietary DBMS. Sections 4.2 to 4.13 are important factors for evaluating DBMS in general and specifically OS DBMS against proprietary DBMS.

## 4.2 Total Cost of Ownership

Total Cost of Ownership (TCO) is extremely sensitive to the set of assumptions organizations make. Many DBMS using companies are seeking simpler, easier to use products that provide enterprise functionality at low cost.

"*Computer Sciences Corporation' study "Open Source: Open for Business" identifies the TCO factors that it believes are most important for evaluating OSS with proprietary software: Hardware costs (including purchase price and hardware maintenance), direct software costs (including purchase price and support and maintenance), indirect software costs (especially administration of licenses), staffing costs, support costs, and downtime cost*" (David A. Wheeler February 8, 2005, why OSS).

Many companies often only look at the initial price but vendors often have hidden costs as well, so this complaint applies to both proprietary and OSS programs. Companies should include all costs in calculations, not just the cost of buying initial licenses. Other costs may include initial licensing cost, hardware expenditure, administration, engineering and support costs, system downtime and training cost.

In a study IDC (2002) shows that software running about 16% of TCO of database applications. A database must not only have lower upfront license costs, but also slash the running cost that make up the total cost of ownership. International Data Corporation found that software cost is only 15% of the total cost of deploying of an Oracle 8i Database application, hardware was 17%, staffing 21% and training 19%. A 28% of the total cost of Oracle database deployment is attributed to system downtime (MySQL Business White Paper Dec 2003).

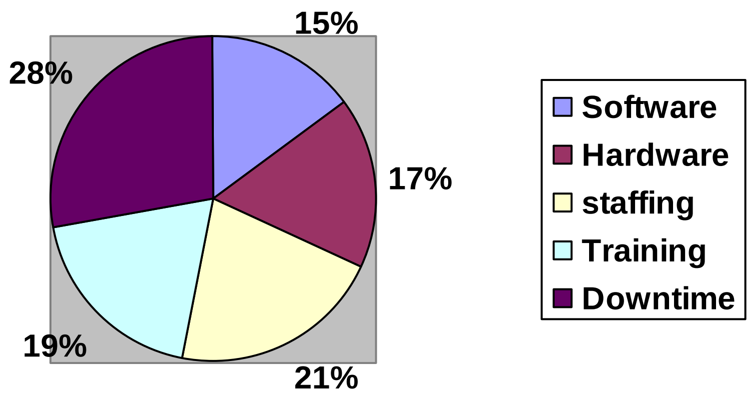


**Figure 4.1 TCO Breakdown of Database Software**

*Source: Forrester Inc. "Your Open Source Strategy". Schadler, Rustein, Lambert,Tseng, Whitely. September 2003.*

But this cost distribution applies mainly to developed countries, and needs to be calculated for developing countries for one simple reason: in developed countries the cost of labour (people working in the IT including Administration, trainers etc.) is very high while in countries like Pakistan it is a fraction of that cost, but the cost of the software is same. So the proportion of 16% of TCO of database applications is inaccurate for developing countries. Additionally developing countries are very much vulnerable to foreign exchange rate fluctuations. From the perspective of developing countries money spent on local providers of services like maintenance of the DBMS re-enters the local monetary system, while that paid to foreign vendors is a drag.

### 4.2.1 Licensing Cost

Software licenses are created for different business uses and for different types of relationships (such as vendor/customer or partner/partner). Most organizations are neither willing nor able to publish their intellectual property. They do not usually allow users to freely copy or modify the program. Proprietary licenses come in a variety of flavors, and do not generally allow access to the source code or modifications. In some cases, the cost of software depends only partially on the license fees. Table 4.1 shows license fee relative to GDP in Pakistan.

**Table 4.1 Licence fee relative to GDP/capita**

| **Country** | **GDP/cap** | **PCs (‘000s)** | **Piracy** | **WinXP Cost** | |
|---|---|---|---|---|---|
| | | | | Effective $ | GDP months |
| Pakistan | 415 | 585 | 83% | 47630 | 16.20 |

*Source: World Bank World Development Indicators Database, 2001; Piracy data from Business Software Alliance*

GDP/capita in US$, Windows + Office XP cost in effective US$ equivalent.

The restrictions on the use of proprietary software are usually enumerated in the end user license agreements (EULAs). For software provided by large companies, EULAs are generally long and complex contracts. Among the most common of the prohibitions for such programs are making unauthorized copies, using it on more than a certain number of computers and reverse engineering it (David A. Wheeler January 5, 2005).

Open Source licensing, on the other hand, does not embody beliefs about ownership aside from recognizing the fact that some organizations wish to own software and others don't. Low or no initial licensing cost of OS DBMS makes it possible to migrate off, if the existing system doesn’t meet the organization requirements (Redners tryckeri, 2003).

**Table 4.2 Comparison between different types of licenses (OSS)**

| Type of license | Free of charge | Further distribution permissible | No restrictions for use | Source code available without cost | Changes in source code permissible | Derivative work must also be open | Integration with proprietary software permissible |
|---|---|---|---|---|---|---|---|
| Public domain | X | X | X | X | X | | X |
| Shareware | (X)1 | X | | | | | |
| Freeware | X | X | X | | | | |
| GPL | X | X | X | X | X | X | |
| LGPL | X | X | X | X | X | X | X |
| MPL | X | X | X | X | X | X | X |
| BSD License | X | X | X | X | X | | X |
| 1) Shareware is only free for a limited period of time for evaluation of the software. | | | | | | | |

*Source: Redners Tyckeri, 2003*

### 4.2.2 Hardware Cost

Hardware cost is defined as all program cost, excluding software development. DBMS acquisition cost depends upon quantitative and qualitative parameters. Quantitative parameters include complexity, quantity, weight and size, and qualitative parameters such as environmental specification, type of packaging and level of integration. Hardware cost parameters brings speed, accuracy, and flexibility to the cost estimating process.

OS DBMS still cost money to deploy in the real world, because initial licensing costs are a minority of the costs in most software deployments.

### 4.2.3 Administration, Engineering, and Support Costs

This is a traditional cost for both OSS and proprietary software. In both cases vendors offer services, which are connected to the product. It is often these services, which generate the largest profits. Quality services are also an important criterion for many users when choosing DBMS. But there are also many smaller companies that only offer these kinds of services. They choose some of the most successful OSS/Proprietary products and acquire professional competence in these products (Redners tryckeri, 2003).

"*A combination of low complexity, high reliability and a wealth of support resources lower the cost of developing, maintaining and supporting databases. The elegant and uncomplicated architecture has numerous benefits.*

- *Engineers are less likely to make mistakes. Complex systems often harbor hard-to-find bugs that are not apparent until deployment. Uncomplicated systems help reduce downtime.*
- *Non-complex tasks can be allocated to less expensive developers enhancing the overall productivity and cost effectiveness of the team*" (MySQL Business White Paper Dec 2003).

### 4.2.4 System Downtime Cost

System downtime is often the most expensive cost of any application. As businesses have moved from batch processing to real-time enterprise systems running critical customer service or online sales systems, the impact of any outage directly impacts the bottom line.

Productivity of employees is dependent upon the system and downtime on transactional systems is counted in lost orders. In such cases, lost revenues could increase with time. Companies are increasingly encouraging customers to use online self service systems rather than costly call centers. This increases hard cost due to volume of transactions being routed to an increasingly overloaded call-center (MySQL Business White Paper Dec 2003).

### 4.2.5 Training Cost

In order to create a culture around selected DBMS organizations need to hire and train people to support it. Highly qualified skill staff is very important to increase operational efficiency by driving down the cost of ownership for new and existing applications of DMBS.

Products that perform similar functions will have similar training costs. Training costs are rarely a product differentiator, but occasionally a product is so much easier to use that it has a strikingly lower training cost. Transitioning data can be costly, but often people who understand how to use one system can quickly adapt to use another; as a result, the costs of retraining people can be easily overestimated. An organization pays additional for support of DBMS products, but if DBMS support competed or self-supported, in that case this is less expensive. Again, organizations will need to examine their specific circumstances (David A. Wheeler January 5, 2005).

### 4.3 Performance

Efficient database performance can enable additional savings. High speed DBMS are designed for complex decision support activity in a multi-user mixed workload environment. Organization needs to check on the maturity of the DBMS for supporting every type of query with good performance and determining the best execution plan based on changing data demographics. Many project websites include performance data. Some companies unsurprisingly, only present the most positive performance data near their front pages, so this may not present a full picture. The best way to measure performance is to try it on a real workload specific to organizational needs (David A. Wheeler February 8, 2005).

### 4.4 Reliability

Reliability measures how often the DBMS works and produces the appropriate answers. Reliability is difficult to measure, and strongly depends on how the program is used. Problem reports are not necessarily a sign of poor reliability. Companies often complain about highly reliable program, because their high reliability often leads both customers and engineers to extremely high expectations. Indeed, the best way to measure reliability is to try it on a real work load. The program web site itself is likely to try to describe the program’s reliability via different projects. Information is often available on program web site that may help to evaluate program’s reliability. In particular, a mature program is far more likely to be reliable. The program web site itself is likely to try to describe the program's maturity (David A. Wheeler January 5, 2005).

### 4.5 Scalability

There are three aspects of DBMS scalability:

1. The application should be able to scale out more servers as necessary. It should be easy to add new servers at any time.
2. The underlying database should be scalable as the amount of data accumulates quickly.

3. The communication approach should be scalable so that it is easy to disseminate the information through e.g. the web.

Scalability, in this context, suggests the size of data or problem the program can handle. If organization expects the program to be able to handle unusually large datasets, or be able to execute on massively parallel or distributed computers, there should be some evidence that the program has been used that way before.

### 4.5.1 Ability to Scale out More Servers as Necessary

Database should have capabilities to handle most application requirements with an architecture that is extremely fast, reliable and easy to use. DBMS platform hardware purchases typically have involved a large initial capital outlay, in order to accommodate resources capable of dealing with the projected workload for day one, up to some point in the future. As database loads grow, then new processors and CPU (Central Processing Unit) are installed, until eventually the server is fully configured. The next step is another large capital outlay on a new server. Whilst the previous system is either retired or moved onto new duties. This approach has significant cost and service implications for the end solution.

### 4.5.2 Independent Benchmarks

Several different types of organizations commonly benchmark Database Management Systems. Each of these organizations has a different set of goals and faces different constraints when conducting a DBMS benchmark. Some companies conduct their own DBMS benchmarks to try to profile the strengths and weaknesses of a DBMS. At first glance the process seems simple but there are some subtle issues that can cause serious problems (and occasionally complete failure). It is significant to note that a company will typically involve one or more database vendors and a computer hardware vendor in this process. These organizations will not encourage the customer to conduct more thorough and detailed tests, because such tests take longer and are more likely to uncover problems that might kill the sale. The customer will be encouraged to speed up the testing process and make the selection (Neal Nelson Database Benchmark).

The market needs an experienced database benchmarking organization that is not tied to any hardware or software vendor. This organization could run tests on computers selected by the customer, with DBMS products chosen by the customer, according to time schedules specified by the customer. The independent organization's goal would not be to sell a computer or a database package but rather to learn and report the truth to the customer.

The benchmark should measure many different aspects of DBMS performance at many different user levels and database sizes. The benchmark must report the results in detail so that both the strengths and weaknesses of the DBMS are uncovered. The benchmark should be run in its standard form for multiple database products on a single computer platform to show relative performance of the DBMS packages. The benchmark should be run in its standard form with the same DBMS package on multiple platforms to show relative performance of the computer equipment (Neal Nelson Database Benchmark).

### 4.6 Security

Evaluating a DBMS security is complicated, in part because different uses and different environments often impose different security requirements on the same type of product. Some common security functions are considered important in evaluating DBMS:

- Manage users, roles, logins, profiles, groups, and aliases.
- Migrate accounts between the same or different database platforms.
- Manage password security.
- Grant and revoke roles to users and to other roles, as well as system and object-level privileges.
- Set privileges for multiple users in a single operation.

"*Some proprietary products undergo "Common Criteria" evaluation for security however, the costs of this type of evaluation make it rare for open source products. If a Common Criteria evaluation is available, be sure to look at the product's "Security Target". The Security Target is publicly available document that specifies important information about the evaluation, such as exactly what configuration was tested, what assumptions were made, and what security requirements were tested. If the Security Target describes a configuration that's very different from what you will use, or doesn't include the security requirements that are important to you, the evaluation results may not be as applicable to you*"(David A. Wheeler January 2005).

At this time OSS users usually don't pay for a Common Criteria evaluation of product (due to the time and expense), but it's certainly possible. Of course software development experts will need to know how to develop secure system in the first place. If they don't already know how to do this in detail, make sure they learn how to do so first.

Another area where examining the DBMS code can be particularly valuable is when an organization wants to carefully evaluate a program's security. Companies can have software development experts look at the code to see if the DBMS appears to be trustworthy (e.g., if it follows good practices). For example, they could see if:

- Security (making sure only those with access privileges can access the data).
- It minimizes privileges (e.g., only small portions of the program have special privilege, or the program only has special privileges at certain times)
- It strives for simplicity (simpler designs are often more secure)
- It carefully checks inputs

### 4.7 Platforms

The companies look for advanced and standard compliant features DBMS. The product functionality should be especially familiar to database developers and DBAs which allowing IT departments to ease the integration and transition of their data, applications and skill sets. Interpretability of DBMS on wide range of platforms makes it accessible to work in different kinds of IT environments. Multiple language support enhances the product in foreign markets.

### 4.8 Standard

Inadequate interoperability between applications and systems is one of the main reasons why businesses and organizations hesitate to change over computers applications in their working environment (Redners tryckeri, 2003).

#### 4.8.1 The Mechanisms of Lock-in

Lock-in is process by which vendors try to coerce a customer into continued use of software by making it impossible or difficult to migrate to a different solution. Lock-in is serious problem in DBMS products.

##### 4.8.1.1 Changeover Costs

A change from one product or vendor to another always means higher costs, either for the companies or for the new supplier or vendor. According to (Redners tryckeri, 2003) changeover costs can be sorted into seven different categories. It is important to know about and to be able to identify these so that lock-in doesn't come as a surprise.

**1. Procurement of Products with a Long Actual Lifespan**

When changing from one product to another there are additional costs for replacing earlier investments in products with a long actual lifespan. These costs decrease in tune with the value of the original product, but by making available service agreements, upgrades, add-on products, etc., suppliers often attempt to prolong the lifespan of a product. The same tactics are often used to generate costs even for products with a short actual lifespan.

**2. Replace Knowledge**

A cost for learning to use a new product along with the decline in profits during the learning period is also a part of the costs for changeover. These costs increase over time, since knowledge of and familiarity with the old system increase.

**3. Conversion of Information and Data**

Because products often are incompatible with each other in a changeover situation, there are additional costs for transiting data from product to another. The longer a system is in use, the more information and data will be tied up in the system. This type of changeover cost increases, therefore over time.

**4. Specialized Vendors**

Dependency on an individual vendor who supplies highly specialized products can, over time, create a lock-in effect if competing vendors go out of business or lose their ability to offer compatible products.

**5. Costs of Finding an Alternative**

The cost of finding alternative suppliers, new procurement processes, signing agreements and risks that a new supplier does not fulfill his undertakings must also be considered lock-in effects.

**6. Customer Loyalty Programmes**

Corresponding programmes along with other benefits, especially those that accumulate over time, result in costs for changing from one vendor to another and are used frequently by vendors in order to create lock-in effects.

**7. Agreements**

Finally, agreements often bind a customer to a certain vendor and vice versa. Changeover costs in the form of damages to be paid if a buyer breaks his agreement with the vendor are a by-product of the mutual insurance that an agreement entails.

#### 4.8.2 How to Deal with Lock-in

Lock-in effects are almost impossible to completely avoid, except for products, which have completely comparable, exchangeable and with a large number of competing suppliers battling against each other in hard competition. But by being well aware of the different costs that are caused by lock-in, it is possible to assess different vendors and to decide which countermeasures are reasonable to carry out.

##### 4.8.2.1 Well-achieved Procurement

A well-achieved procurement is a successful means of minimizing the effects of lock-in. Shapiro and Varian provide the following practical advice for a procurement situation:

- "*Be aware of subtle lock-in. Even lock-in - which at the beginning can seem harmless - can grow to be a serious problem when in time one invests in add-on products and increasingly large amounts of data stored in proprietary formats and databases.*
- *Take it easy. Study the alternatives at hand and the consequences before making a decision on an individually selected choice of product or supplier.*
- *Always leave your door open for alternative possibilities. Prepare your procurement in an orderly manner*".

##### 4.8.2.2 Systems Architecture

In case where a company develops in-house or contracts for development, it can by preparing for interchangeability, minimize costs for change and therewith the degree of lock-in.

The higher costs for this kind of preparation can sometimes be hard to motivate if a changeover at the present time does not seem to be probable or desirable, but it is important to remember that lock-in gives room for the vendor to raise prices for service agreements, add-on products and other things in the future. Future negotiations can be made easier for the clients if wise decisions concerning systems architecture are made at an early stage (Redners tryckeri, 2003)

### 4.9 Management

If the system is difficult to manage, without the right tools, then mistakes will be made, leading to unplanned outages. Selecting the wrong solution can undermine the entire development effort and the credibility of the selection criteria. Once DBMS has been physically selected, the focus is then shifted to how that existing physical data is moved, migrated, and transformed into meaningful information (David A. Wheeler February 8, 2005).

Models are the de-facto standard ways to define logical data and their relationships, and models also tend to be the best way to define physical data movements between various source and target systems. The roles most active in DBMS management include data modelers, data architects, database administrators, and data extract transform-load (ETL) specialists.

### 4.10 Support

Support covers several areas including training users on how to use the product, installing the product, and answering users who have specific problems trying to use a working product. Further Support includes advanced support programs, 24x7x365 support, worldwide support, training, consulting and 3rd-party products and services. This includes product documentation and any other source of information.

One major difference between OS DBMS and proprietary DBMS is how support is handled. Fundamentally, OS DBMS users have few choices (1) they can choose a traditional commercial support model (2) they can choose to provide support in-house or (3) they can depend on the development and user community for support. These choices apply to each of the various tasks and the answers can even be different for the different tasks (David A. Wheeler January 5, 2005).

In OS DBMS case it can be risky to depend solely on community support without having in-house experience with the product. OS DBMS project page shows, which organizations are contributing in the project; this is usually easy to determine from the email addresses of contributors. In many cases there is a single company who offers primary support for a project in a manner similar to proprietary vendors, typically employing a key project developer; that company is usually the best choice for commercial support, because they will know the project direction and can quickly respond to client needs. Many organizations have ongoing relationships with suppliers and consultants who are qualified to do at least some of these tasks (David A. Wheeler January 5, 2005).

Proprietary DBMS support is usually only provided by the proprietary vendor and there may be several competing companies offering support. These should be evaluated in the usual way, looking at the company's reputations, consider the company's financial health, talking with their existing customers, and so on. But in addition, clients should favor companies that are clearly directly involved developing the DBMS projects.

Companies that include developers clearly demonstrate an understanding of the system, can potentially fix any problems that DBMS have with it, and can have the fix incorporated into the main project so that organization won’t have that problem again in the future.

Some organizations particularly large ones create their own support organization for the product. This is probably better considered after having used the product for a while; it can be risky to do self-support without having already had significant in-house experience with the product.

### 4.11 Maturity

Maturity of software product is a set of features that satisfy the needs of a particular market segment. Evaluating the maturity of DBMS product companies need to undertake a detailed set of steps and produce specific work products. For example, maturity of product, reliability of current version, frequency of updates, ease of installation, good documentation, best practices information available, original developers still involved and committed, strength of company behind the product, large community of users, certification available and user conferences. The output of these steps will help in the comparison of DBMS products in detail.

### 4.12 Expertise

The most sophisticated and useful software on the market can quickly become a burden if the vendor does not have the staff to adequately train users and respond to help requests. By contrast, the most helpful and customer-oriented vendor may also fail clients if they lack the computer science and engineering expertise to troubleshoot complex problems (www.sei.cmu.edu).

In both cases companies offer services, those are connected to the product itself. It is often these services, which generate the largest profits. Quality services are also an important criterion for many users when choosing DBMS. These companies choose some of the most successful DBMS products and acquire professional competence in these products. As DBMS becomes more and more diverse and widespread, so also will there be an increasingly expansive need for consultancy services for development, systems integration, special adaptations and training (Redners tryckeri, 2003)

### 4.13 References

The best way to check references is to find someone who has done this before. Questioning different clients about any vendor ties might help in decision making. The questions are best answered by experience of sites that have implemented the DBMS in the past. Insist on 10-12 references, preferably of own firm's size and in own industry. Use them as a guide throughout the process and as an outside check on the decisions being made. Taking these steps will greatly improve chances of making a wise selection (www.newportconsulting.com).

- Interview the consultants who will actually do the work and check their references.
- Define the deliverables from each step of the process, and the review and approval points.
- Agree on the level of formality of each deliverable. The more formal the process, the longer and more expensive the process

**Table 4.3 Evaluation Criteria overview**

| Evaluation Criteria | Criteria Categories |
|---|---|
| Total Cost of Ownership | Initial licensing cost<br>Hardware expenditure<br>Administration, engineering and support costs<br>System downtime cost<br>Training cost |
| Performance | Speed<br>Independent benchmarks on comparable hardware and load |
| Reliability | Server and application uptime<br>Test of code quality<br>Responsiveness to bug reports |
| Scalability | Ability to scale out more servers as necessary<br>Independent benchmarks |
| Security | No or low number of known vulnerabilities<br>Independent validation of security |
| Platforms | For which platforms is the database available<br>Multi-language support |
| Standards | Support industry standards<br>Interoperability with other software or systems<br>No vendor/platform lock-in |
| Management | Ease of management<br>Ability to manage more servers |
| Support | Standard support<br>Advanced support programs<br>24x7x365 support<br>Worldwide support<br>Training<br>Consulting<br>3rd-party products and services |
| Maturity | Maturity of product<br>Reliability of current version<br>Frequency of updates<br>Ease of installation<br>Good documentation<br>Best practices information available<br>Original developers still involved and committed<br>Strength of company behind the product<br>Large community of users<br>Certification available<br>User conferences |
| Expertise | Availability of experts<br>Books and reference material<br>Certified trainers<br>Certified consultants |
| Reference | References to talk to, preferably in your industry<br>Experience on performance, reliability, scalability<br>Best practices available<br>Good experience with support |

# 5. METHODOLOGY

## 5.1 Objective and Structure

This chapter contains the empirical part of thesis and includes hypothesis, research methodology, information regarding data and data collection methods and data analysis. Section 5.2 presents the hypothesis of the thesis. Section 5.3 introduces research methodology, data collection method, creation of online questionnaire, response rate and sample description. Section 5.4 gives the descriptive analysis of data including independent t-test results.

## 5.2 Hypotheses

This research aims to compare selection criteria of OS DBMS against the selection criteria of proprietary DBMS in Pakistan.

MySQL AB "Business White Paper 2004" provides a checklist for evaluating both proprietary and mature OSS. MySQL AB considered twelve important factors to evaluate software product. It includes Total Cost of Ownership, Performance, Reliability, Scalability, Security, Platforms, Standards, Management, Support, Maturity, Expertise and Reference.

David A. Wheeler (2005) designed a process to compare OSS program side by side with proprietary programs and other OSS programs and determines which one (if any) best meets organization needs. Important attributes to consider include functionality, cost, market share, support, maintenance, reliability, performance, scalability, usability, security, flexibility/customizability, interoperability, and legal/license issues. The study of David A. Wheeler (2005) proved that the checklist of MySQL AB is sufficient to evaluate OS DBMS against proprietary DBMS.

This forms a strong basis for subsequent benchmarking studies in this area. It provides first-hand information on what basis companies select OS DBMS or proprietary DBMS, which should be useful in the following aspects:

- Providing essential data for relevant government departments to prepare strategies for implementing open source applications in the IT infrastructure of country.
- Promoting awareness and the benefits of OS DBMS in Pakistan. Also encouraging companies to seriously consider in their business.
- Laying down a good foundation for relevant teaching and further research in the tertiary institutions in Pakistan.
- OSS is the only alternate for developing countries for sustainable economic growth.

The ultimate aim is to look at the relevance of Open Source Software for the developing countries like Pakistan

Thus, this leads us to the following hypotheses:

H1: The importance of selection criteria practices differ in the case of the OS DBMS against proprietary DBMS.

### 5.3 Research Methodology

The research methodology adopted in this project is an online questionnaire survey because it was considered the most appropriate data collection method in this case. The pilot work was undertaken in February 2005, by Pakistan Open Source organization. Based on the results and comments from the pilot tests, revisions were made to the questionnaire design. To make it more effective, it is focused toward particular groups. In addition, this approach allows having a large sample size and if the response rate stays too low, a follow-up reminder can be sent. Survey by post mail option was considered and then rejected. It would have high costs and more time consuming.

### 5.3.1 Data

The data needed for this study is taken from companies that have already implemented and are using DBMS in Pakistan.

### 5.3.2 Data collection method

The data collection methods for this study include questionnaire, e-mails and telephone calls. The type of survey was online questionnaire. The survey conducted during March 2005.

### 5.3.3 Questionnaire

The questionnaire used in this survey study is shown in appendix . I will describe the structure and source of questions that were presented in the questionnaire.

The objective of research considered important in designing the survey. First, the survey should provide accurate information about the decision criteria of DBMS in general as well as specifically, Open Source DBMS against the selection criteria of proprietary DBMS in Pakistan.

There were 71 questions. 4 were open ended questions, 66 were close ended questions and in one question response was optional. Companies were asked to answer the questions that were divided into four main sections. The first section was to get company background information. The questions in section two and three were based on MySQL AB checklist for evaluating both proprietary and mature OSS. Section four was optional. There was only one question in section four to get additional comments from company.

In section one; there were 8 questions to collect company background information and organizational characteristics. There were 4 open ended and five closed ended questions with multiple choice response. The company information includes company name, web address, name of person who is filling survey and that person's position in company. The organizational characteristics included company size, IT intensity, industry and whether company is using OS DBMS or Proprietary DBMS.

Indicator for size was the number of employees. Companies with less then 100 employees considered Small and Medium Enterprises. Companies with more then 100 employees considered big enterprises. The industry was divided in three sectors. The focus of survey was Banking & Finance, IT & Telecommunication and Consumer Goods industries. In addition, option for the rest was also included. IT intensity defined as importance of IT in developing product/services. Motivation for IT intensity was that industries with a high IT intensity and thus high IT expenditures might be more familiar with DBMS comparison therefore might show a different usage pattern from those with lower IT intensity

In section two, participants were asked to rank the 12 categories importance based on their decision in favor of open source or proprietary DBMS product. The measurement scale was based on the degree of importance (from 1= not important to 5 = very important).

Section three was derived from section two. In section three the 12 categories were further split into sub-categories to identify the importance of specific attributes in each category. The measurement scale was based on the degree of importance (from 1 = not important to 5 = very important).

Section four was optional. If any company had additional comment then they could add in that section.

After the entire questionnaire designed, Pakistan Open Source organization hosted the questionnaire online. Pilot test conducted to ensure that the questions and language were understandable and unambiguous. A test was launched on February 28th, 2005 by Pakistan Open Source organization and was done by four IT professionals. Usability analysis of online questionnaire also conducted. According to the feedback, some additions and changes in format suggested to Pakistan Open Source organization.

The respondents were informed in e-mail that the responses to the questions will be held in confidence, only statistical results will be revealed. The results will be used for my Master's thesis at Hanken, the Swedish School of Economics and Business Administration in Helsinki, Finland.

### 5.3.4 The Response Rate

The most appropriate option to get the response was online questionnaire. The online questionnaire was mailed to a total of 800 companies in Pakistan. The companies e-mail addresses were gathered from different sources that included yellow pages, stock exchanges and web sites. The target respondent group is Chief Information Officer (CIO)/ Chief Technology Officer (CTO)/ IT manager/ Database Administrator (DBA).

Altogether 29 companies responded, rated 3.625 %. 6 companies out of 29 were using OS DBMS, rated 0.75 % and 23 were using proprietary DBMS, rated 2.875 %. Due to low response rate, I have sent reminder twice but the response rate remained low. There were few reasons for such a low response rate.

- Managers in such high positions were too busy.
- IT personnel did not receive most of the e-mails; therefore, other respondents might have trouble understanding the questions, the most likely reason that they choose not to participate.
- Not all companies contacted had DBMS.
- Most likely most of the e-mails did not go through spam control software and deleted as spam.
- Questionnaire was a bit technical and lengthy.

Although response rate is rather low but I will run the tests and decide further on whether the data of the 6 vs 23 companies is enough to show any significant difference or research would need to be carried out with a different research method to bring better results. Descriptive and t-test statistics used to analyze the data.

### 5.3.5 Sample Description

The profiles of the companies who participated in the survey summarized in Table 5.1.

**Table 5.1 Sample Description**

| Organizational characteristics | OS DBMS respondents | Per cent | Proprietary DBMS respondents | Per cent |
|---|---|---|---|---|
| **DBMS:** | | | | |
| MySQL | 6 | .75 | | |
| Oracle | | | 10 | 1.25 |
| MSSQL | | | 13 | 1.625 |
| **Industry:** | | | | |
| IT &Telecommunication | 5 | 83.33 | 8 | 34.78 |
| Banking & Finance | 1 | 16.66 | 5 | 21.74 |
| Manufacturing | | | 3 | 13.04 |
| Consumer Goods | | | 2 | 8.69 |
| Others | | | 5 | 21.74 |
| **Intensity of IT in company:** | | | | |
| High | 6 | 100 | 18 | 78.26 |
| Low | | | 5 | 21.73 |
| **Total number of employees:** | | | | |
| 1-20 | 4 | 66.66 | 5 | 21.73 |
| 20-50 | 2 | 33.33 | 7 | 30.43 |
| 50-100 | | | 2 | 8.69 |
| more then 100 | | | 9 | 39.13 |

Of the 6 companies that are using OS DBMS (MySQL), 5 (83.33 per cent) are from IT & Telecommunication industry, 1 (16.66 per cent) from Banking & Finance industry. In terms of IT intensity, 6 companies (100 per cent) have high IT intensity. 4 companies have less then 20 employees and 2 have ranged from 20 to 50.

Of the 23 companies that are using proprietary DBMS, 13 (1.625 per cent) are using MSQL and 10 (1.25 per cent) are using Oracle. Of the 23 that participated in the study, 8 (34.78 per cent) are from IT & telecommunication industry, 5 (21.74 per cent) from Banking & Finance, 3 (13.04 per cent) from Manufacturing, 2 (8.69 per cent) from consumer goods and 5 (21.74 per cent) from others. In terms of IT intensity, 18 companies (78.26 per cent) have high IT intensity and 5 (21.73 per cent) companies have low IT intensity. 5 (21.73 per cent) companies have less then 20 employees, 7 (30.43 per cent) have employees ranged from 20 to 50. 2 (8.69 per cent) have employees ranged from 50 to 100. 9 (39.13 per cent) companies have more then 100 employees.

**5.4 Analysis of Data**

The following descriptive statistics taken from section 2 of the questionnaire (see appendix A):

Table 5.2 reports the descriptive statistics for selection criteria of OS DBMS. Categories ranked on the basis of importance. The most important category is Reliability (M=5). Second important category is TCO (M=4.833). Third important categories are Performance (M=4.667) and Platforms (M=4.667). Fifth important category is Standards (M=4.5). Sixth important category is Security (M=4.33). Seventh important category is Management (M=4.167). Eighth important category is Expertise (M=4). Ninth important categories are Support (M=3.833) and Maturity (M=3.833). Eleventh important category is scalability (M=3.5). Twelfth category is considered neutral References (M=3).

**Table 5.2 Importance of selection criteria by Mean Value for OS DBMS**

| Categories | OS DBMS | |
|---|---|---|
| | Mean | Std. Deviation |
| Reliability | 5,000 | 0,000 |
| Total Cost of Ownership | 4,833 | 0,408 |
| Performance | 4,667 | 0,516 |
| Platforms | 4,667 | 0,516 |
| Standards | 4,500 | 0,837 |
| Security | 4,333 | 0,516 |
| Management | 4,167 | 0,753 |
| Expertise | 4,000 | 0,632 |
| Support | 3,833 | 1,472 |
| Maturity | 3,833 | 0,753 |
| Scalability | 3,500 | 0,837 |
| Reference | 3,000 | 1,549 |

Number of respondents = 6

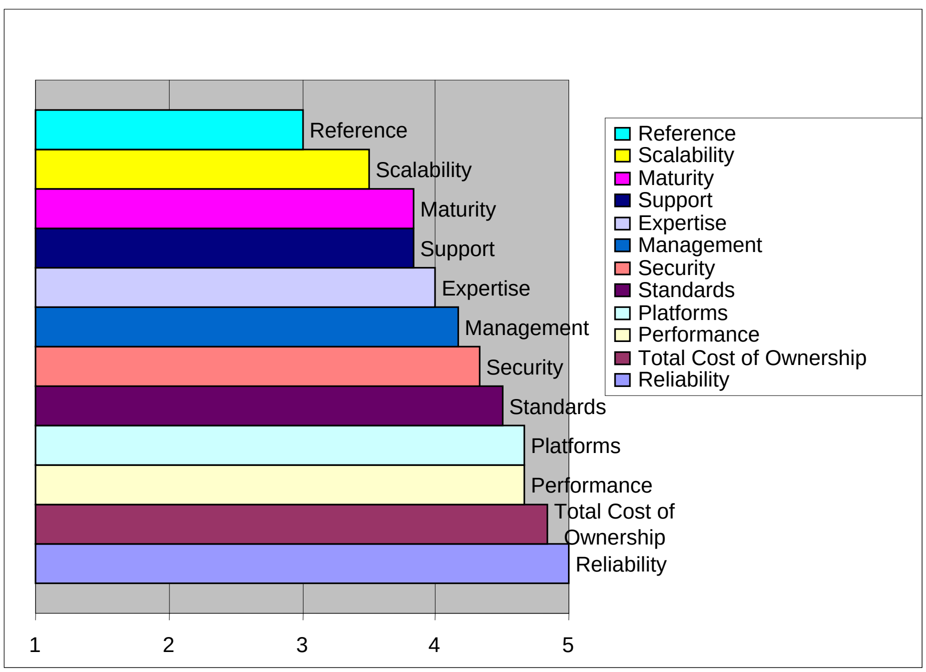


**Figure 5.1 Ranking of Selection criteria for OS DBMS**

Table 5.3 reports the descriptive statistics for selection criteria of proprietary DBMS. Categories ranked on the bases of importance. The most important categories are Reliability (M=4.286) and Performance (M=4.286). Third important category is Security (M=4.609). Fourth important category is Scalability (M=4.565). Fifth important category is Support (M=4.522). Sixth important category is Maturity (M=4.435). Seventh important category is Expertise (M=4.384). Eighth important category is Management (M=4.304). Ninth important category is Standards (M=4.130). Tenth important category is TCO (M=4.087). Eleventh important category is References (M=3.807). Twelfth important category is Platforms (M=3.826).

**Table 5.3 Importance of selection criteria by Mean Value for proprietary DBMS**

| Categories | Proprietary DBMS | |
|---|---|---|
| | Mean | Std. Deviation |
| Performance | 4,826 | 1,203 |
| Reliability | 4,826 | 0,388 |
| Security | 4,609 | 0,491 |
| Scalability | 4,565 | 0,662 |
| Support | 4,522 | 0,722 |
| Maturity | 4,435 | 1,072 |
| Expertise | 4,348 | 0,815 |
| Management | 4,304 | 0,822 |
| Standards | 4,130 | 0,730 |
| Total Cost of Ownership | 4,087 | 0,788 |
| Reference | 3,870 | 0,647 |
| Platforms | 3,826 | 0,968 |

Number of respondents = 23

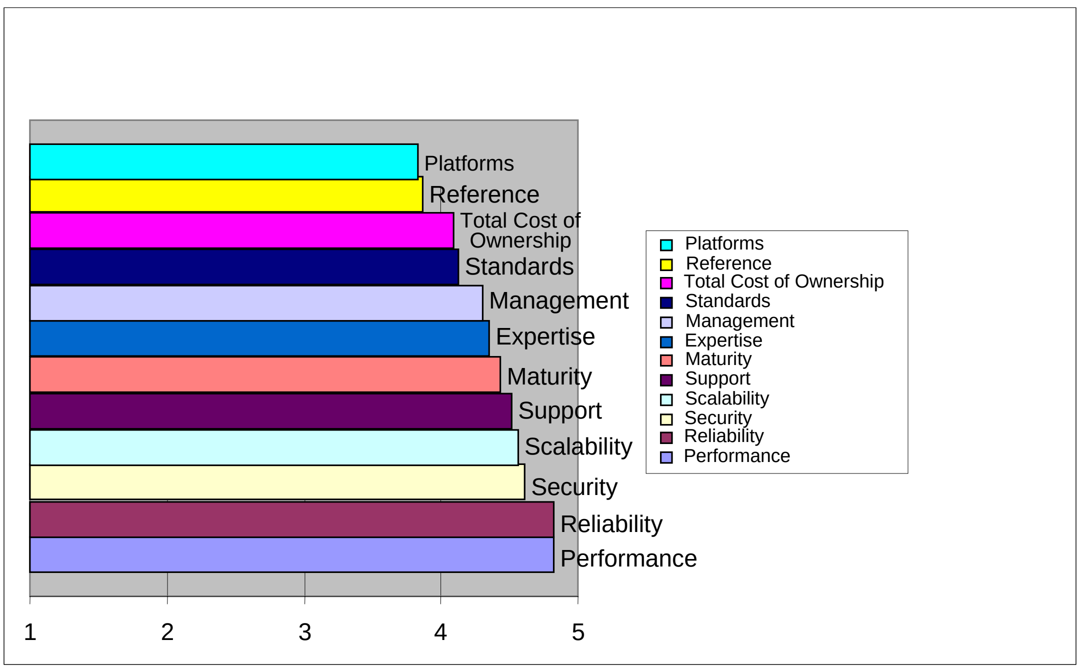


**Figure 5.2 Ranking of Selection criteria for proprietary DBMS**

The following descriptive statistics taken from section 3 of the questionnaire (see appendix):

**Table 5.4 Importance of sub-categories in DBMS selection criteria**

| Sub-Categories | Mean OSS DBMS | N | Std. Deviation | Mean Proprietary DBMS | N | Std. Deviation | Total Mean | N | Std. Deviation |
|---|---|---|---|---|---|---|---|---|---|
| **Total Cost of Ownership** | | | | | | | | | |
| Initial licensing cost | 4.667 | 6 | 0.516 | 3.739 | 23.000 | 1.176 | 3.931 | 29 | 1.132 |
| Hardware expenditure | 3.833 | 6 | 1.169 | 4.000 | 23.000 | 1.087 | 3.966 | 29 | 1.085 |
| Administration, engineering and support costs | 4.333 | 6 | 0.516 | 4.261 | 23.000 | 0.915 | 4.276 | 29 | 0.841 |
| System downtime cost | 4.333 | 6 | 1.211 | 4.435 | 23.000 | 0.788 | 4.414 | 29 | 0.867 |
| Training cost | 3.500 | 6 | 1.049 | 4.174 | 23.000 | 0.887 | 4.034 | 29 | 0.944 |
| **Performance** | | | | | | | | | |
| Speed | 4.500 | 6 | 0.548 | 4.565 | 23.000 | 0.590 | 4.552 | 29 | 0.572 |
| Independent benchmarks on comparable hardware and load | 3.500 | 6 | 0.837 | 3.913 | 23.000 | 1.083 | 3.828 | 29 | 1.037 |
| **Reliability** | | | | | | | | | |
| Server and application uptime | 4.833 | 6 | 0.408 | 4.652 | 23.000 | 0.647 | 4.690 | 29 | 0.604 |
| Test of code quality | 4.333 | 6 | 0.816 | 4.043 | 23.000 | 0.825 | 4.103 | 29 | 0.817 |
| Responsiveness to bug reports | 4.167 | 6 | 0.983 | 4.348 | 23.000 | 0.714 | 4.310 | 29 | 0.761 |
| **Scalability** | | | | | | | | | |
| Ability to scale out more servers as necessary | 3.833 | 6 | 0.983 | 4.043 | 23.000 | 0.706 | 4.000 | 29 | 0.756 |
| Independent benchmarks | 3.333 | 6 | 1.033 | 3.609 | 23.000 | 0.941 | 3.552 | 29 | 0.948 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Security** | | | | | | | | | |
| No or low number of known vulnerabilities | 3.833 | 6 | 0.983 | 4.261 | 23.000 | 0.810 | 4.172 | 29 | 0.848 |
| Independent validation of security | 3.667 | 6 | 0.816 | 4.000 | 23.000 | 1.044 | 3.931 | 29 | 0.998 |
| **Platforms** | | | | | | | | | |
| For which platforms is the database available | 4.667 | 6 | 0.516 | 4.174 | 23.000 | 0.650 | 4.276 | 29 | 0.649 |
| Multi-language support | 3.167 | 6 | 1.602 | 3.174 | 23.000 | 1.370 | 3.172 | 29 | 1.391 |
| **Standards** | | | | | | | | | |
| Support industry standards | 4.500 | 6 | 0.837 | 4.130 | 23.000 | 0.757 | 4.207 | 29 | 0.774 |
| Interoperability with other software or systems | 4.000 | 6 | 0.632 | 3.957 | 23.000 | 0.878 | 3.966 | 29 | 0.823 |
| No vendor/platform lock-in | 4.167 | 6 | 0.983 | 3.783 | 23.000 | 1.126 | 3.862 | 29 | 1.093 |
| **Management** | | | | | | | | | |
| Ease of management | 4.167 | 6 | 0.753 | 4.174 | 23.000 | 0.717 | 4.172 | 29 | 0.711 |
| Ability to manage more servers | 3.833 | 6 | 0.753 | 4.087 | 23.000 | 0.949 | 4.034 | 29 | 0.906 |
| **Support** | | | | | | | | | |
| Standard support | 3.500 | 6 | 0.837 | 4.261 | 23.000 | 0.752 | 4.103 | 29 | 0.817 |
| Advanced support programs | 3.333 | 6 | 0.516 | 4.174 | 23.000 | 0.650 | 4.000 | 29 | 0.707 |
| 24x7x365 support | 3.167 | 6 | 0.408 | 3.913 | 23.000 | 0.949 | 3.759 | 29 | 0.912 |
| Worldwide support | 3.333 | 6 | 0.516 | 3.783 | 23.000 | 0.998 | 3.690 | 29 | 0.930 |
| Training | 3.333 | 6 | 0.516 | 3.913 | 23.000 | 0.733 | 3.793 | 29 | 0.726 |
| Consulting | 3.167 | 6 | 0.408 | 4.000 | 23.000 | 0.798 | 3.828 | 29 | 0.805 |
| 3rd-party products and services | 3.500 | 6 | 0.837 | 3.783 | 23.000 | 1.085 | 3.724 | 29 | 1.032 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Maturity** | | | | | | | | | |
| Maturity of product | 4.167 | 6 | 0.753 | 4.391 | 23.000 | 0.656 | 4.345 | 29 | 0.670 |
| Reliability of current version | 4.667 | 6 | 0.516 | 4.478 | 23.000 | 0.593 | 4.517 | 29 | 0.574 |
| Frequency of updates | 3.833 | 6 | 0.753 | 4.174 | 23.000 | 0.576 | 4.103 | 29 | 0.618 |
| Ease of installation | 3.667 | 6 | 0.816 | 4.087 | 23.000 | 0.900 | 4.000 | 29 | 0.886 |
| Good documentation | 4.000 | 6 | 0.894 | 4.348 | 23.000 | 0.714 | 4.276 | 29 | 0.751 |
| Best practices information available | 3.333 | 6 | 0.516 | 4.130 | 23.000 | 0.920 | 3.966 | 29 | 0.906 |
| Original developers still involved and committed | 3.500 | 6 | 1.049 | 3.565 | 23.000 | 0.992 | 3.552 | 29 | 0.985 |
| Strength of company behind the product | 3.667 | 6 | 0.816 | 4.304 | 23.000 | 0.635 | 4.172 | 29 | 0.711 |
| Large community of users | 4.000 | 6 | 1.095 | 4.217 | 23.000 | 0.795 | 4.172 | 29 | 0.848 |
| Certification available | 3.000 | 6 | 1.095 | 3.739 | 23.000 | 0.964 | 3.586 | 29 | 1.018 |
| User conferences | 2.833 | 6 | 1.329 | 3.348 | 23.000 | 1.112 | 3.241 | 29 | 1.154 |
| **Expertise** | | | | | | | | | |
| Availability of experts | 3.333 | 6 | 1.033 | 4.522 | 23.000 | 0.511 | 4.276 | 29 | 0.797 |
| Books and reference material | 3.500 | 6 | 1.049 | 4.261 | 23.000 | 0.864 | 4.103 | 29 | 0.939 |
| Certified trainers | 3.000 | 6 | 0.632 | 3.739 | 23.000 | 0.864 | 3.586 | 29 | 0.867 |
| Certified consultants | 3.000 | 6 | 0.632 | 3.826 | 23.000 | 0.887 | 3.655 | 29 | 0.897 |
| **Reference** | | | | | | | | | |
| References to talk to, preferably in your industry | 2.833 | 6 | 1.329 | 3.870 | 23.000 | 0.920 | 3.655 | 29 | 1.078 |
| Experience on performance, reliability, scalability | 3.333 | 6 | 1.211 | 4.217 | 23.000 | 0.600 | 4.034 | 29 | 0.823 |
| Best practices available | 3.333 | 6 | 1.033 | 3.957 | 23.000 | 0.706 | 3.828 | 29 | 0.805 |
| Good experience with support | 3.333 | 6 | 1.033 | 4.174 | 23.000 | 0.576 | 4.000 | 29 | 0.756 |

Number of respondents for OS DBMS= 6, Number of respondents for Proprietary DBMS= 23, Total number of respondents= N

### 5.4.1 Independent Samples t-test

An independent sample t-test used to find a statistical difference in the mean scores for two groups (that is, do OS DBMS and proprietary DBMS differ significantly in terms of selection criteria). In statistical terms, I tested the probability that the two sets of scores (OS DBMS and proprietary DBMS) came from the same group. An independent –samples t-test conducted to compare the importance of selection criteria for OS DBMS against proprietary DBMS.

There was a significant difference in scores of Total Cost of Ownership for OS DBMS (M= 4.833, SD=0.408) and proprietary DBMS [M=4.087; SD=1.203; t (24.606) = 2.479; p=0.020].

**Table 5.5 Independent samples t-test for TCO**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Total Cost of Ownership ** | 2.479 | 24.606 | 0.02 | 0.746 |
| Initial licensing cost * | 1.865 | 27 | 0.073 | 0.928 |
| Hardware expenditure * | -0.33 | 27 | 0.744 | -0.167 |
| Administration, engineering and support costs * | 0.185 | 27 | 0.855 | 0.072 |
| System downtime cost * | -0.251 | 27 | 0.804 | -0.101 |
| Training cost * | -1.6 | 27 | 0.121 | -0.674 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

There was no significant difference in scores of sub-categories of Total Cost of Ownership. The most important sub-category of Total Cost of Ownership for OS DBMS was *Initial licensing cost* (M= 4.667), second important sub categories were *Administration, engineering and support costs* (M= 4.333) and *System downtime cost* (M= 4.333). Fourth important sub-category was *Hardware expenditure* (M= 3.833) and fifth was *Training cost* (M= 3.500)

The most important sub-category of Total Cost of Ownership for proprietary DBMS was *System downtime cost* (M= 4.435), second important was *Administration, engineering and support costs* (M= 4.261), third was *Training cost* (M= 4.174), fourth was Hardware *expenditure* (M= 4.000) and fifth was *Initial licensing cost* (M= 3.739).

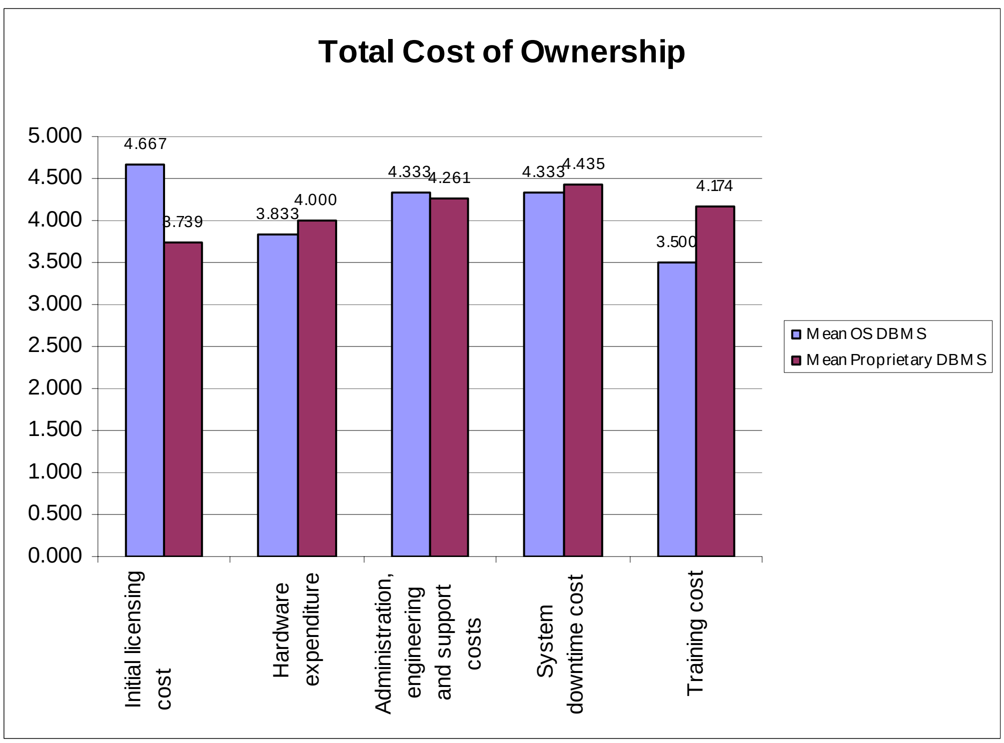


**Figure 5.3 Total Cost of ownership sub-categories comparison**

There was a significant difference in scores of Scalability for OS DBMS (M= 3.500; SD= 0.837and proprietary DBMS [M=4.565; SD= 0.662; t (27) = -3.329; p= 0.003].

**Table 5.6 Independent samples t-test for scalability**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Scalability * | -3.329 | 27 | 0.003 | -1.065 |
| Ability to scale out more servers as necessary * | -0.599 | 27 | 0.554 | -0.21 |
| Independent benchmarks * | -0.627 | 27 | 0.536 | -0.275 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

There was no significant difference in scores of sub-categories of Scalability. The most important sub-category of Scalability for OS DBMS was *Ability to scale out more servers as necessary* (M= 3.833), second important was *Independent benchmarks* (M= 3.333).

The most important sub-category of Scalability for proprietary DBMS was *Ability to scale out more servers as necessary* (M= 4.043), second *important was Independent benchmarks* (M= 3.609).

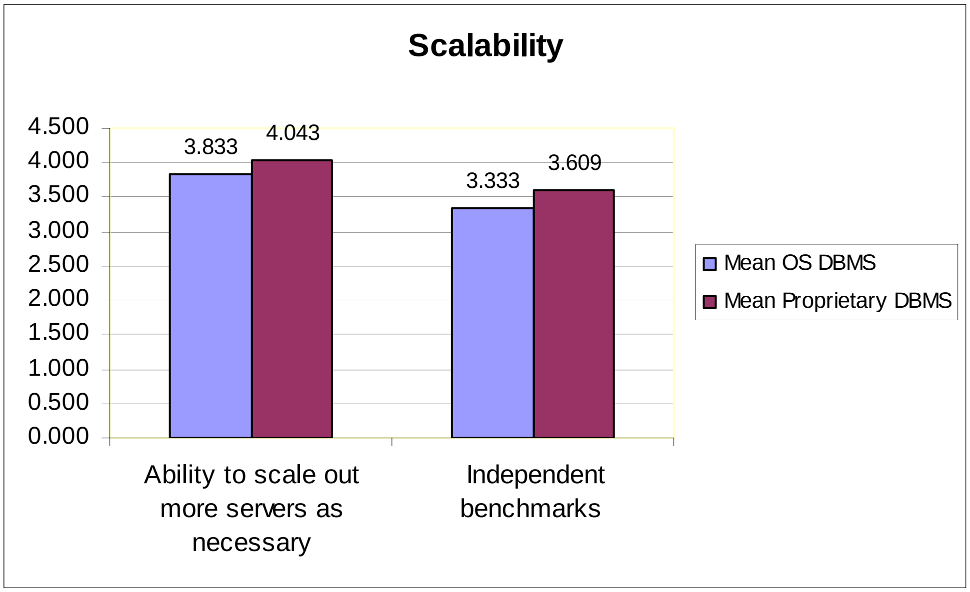


**Figure 5.4 Scalability sub-categories comparison**

There was no significant difference in scores of Performance.

**Table 5.7 Independent samples t-test for Performance**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Performance * | -0.839 | 27 | 0.409 | -0.159 |
| Speed * | -0.244 | 27 | 0.809 | -0.065 |
| Independent benchmarks on comparable hardware and load * | -0.865 | 27 | 0.395 | -0.413 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Performance for OS DBMS was *Speed* (M= 4.500), second important was *Independent benchmarks on comparable hardware and load* (M= 3.500).

The most important sub-category of Performance for proprietary DBMS was *Speed* (M= 4.565), second important was *Independent benchmarks on comparable hardware and load* (M= 3.913).

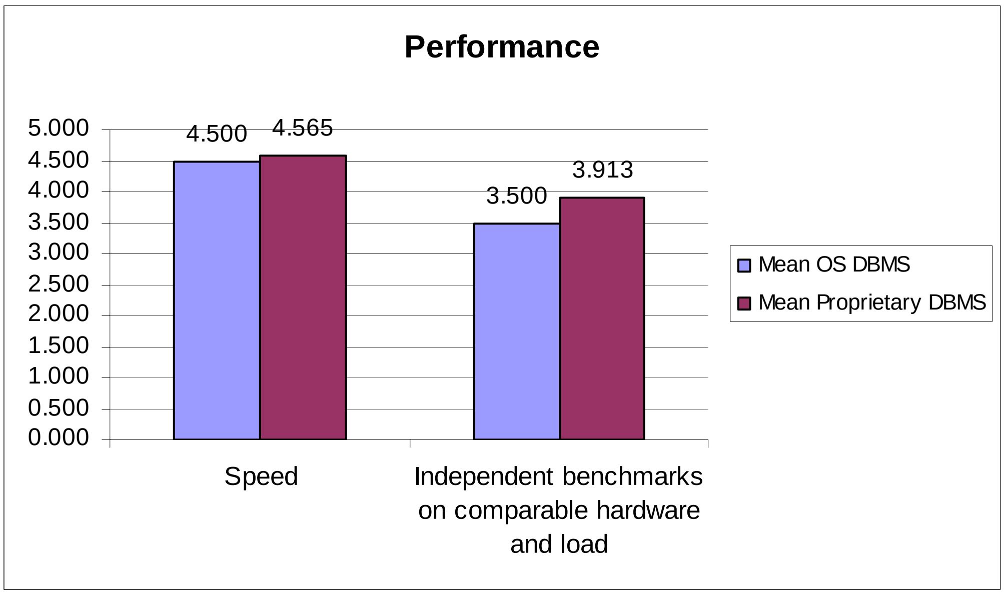


**Figure 5.5 Performance sub-categories comparison**

There was no significant difference in scores of Reliability.

**Table 5.8 Independent samples t-test for Reliability**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Reliability * | 0.856 | 27 | 0.4 | 0.174 |
| Server and application uptime * | 0.648 | 27 | 0.523 | 0.181 |
| Test of code quality * | 0.768 | 27 | 0.449 | 0.29 |
| Responsiveness to bug reports * | -0.513 | 27 | 0.612 | -0.181 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Reliability for OS DBMS was *Server and application uptime* (M= 4.833), second important was *Test of code quality* (M= 4.333). Third important sub category was *Responsiveness to bug reports* (M= 4.167).

The most important sub-category of Reliability for proprietary DBMS was *Server and application uptime* (M= 4.652), second important was *Responsiveness to bug reports* (M= 4.348). Third important sub category was *Test of code quality* (M= 4.043).

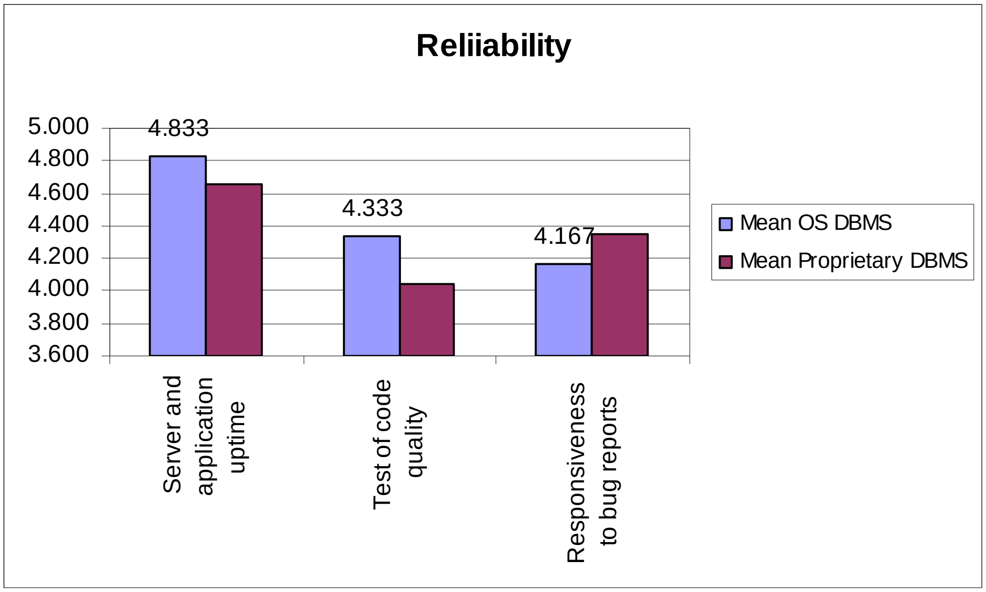


**Figure 5.6 Reliability sub-categories comparison**

There was no significant difference in scores of Security.

**Table 5.9 Independent samples t-test for security**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Security * | -0.872 | 27 | 0.391 | -0.275 |
| No or low number of known vulnerabilities * | -1.104 | 27 | 0.279 | -0.428 |
| Independent validation of security * | -0.723 | 27 | 0.476 | -0.333 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Security for OS DBMS was *No or low number of known vulnerabilities* (M= 3.833), second important was *Independent validation of security* (M= 3.667).

The most important sub-category of Security for proprietary DBMS was *No or low number of known vulnerabilities* (M= 4.261), second important was *Independent validation of security* (M= 4).

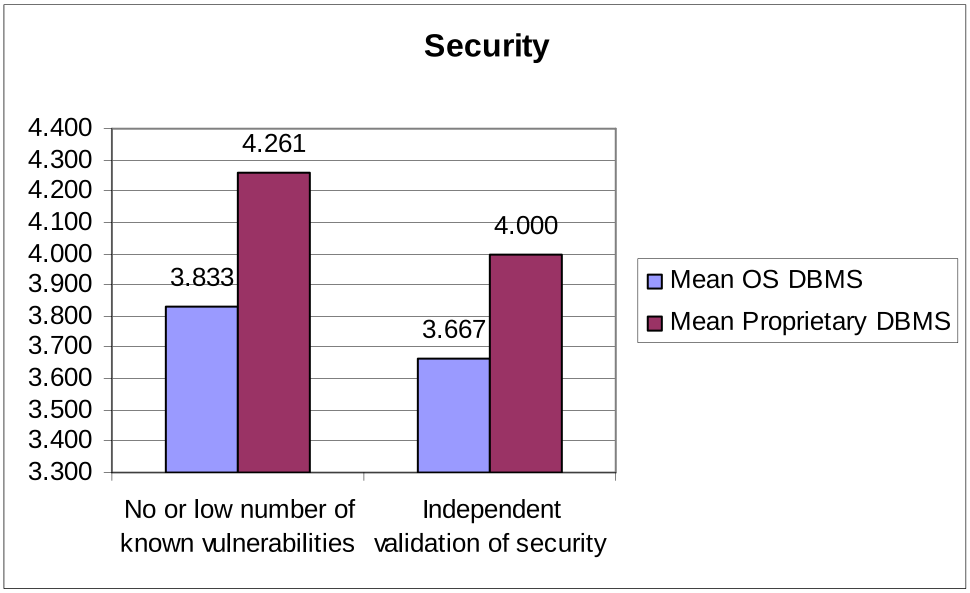


**Figure 5.7 Security sub-categories comparison**

There was no significant difference in scores of Platforms.

**Table 5.10 Independent samples t-test for Platforms**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Platforms * | 1.846 | 27 | 0.076 | 0.841 |
| For which platforms is the database available * | 1.712 | 27 | 0.098 | 0.493 |
| Multi-language support * | -0.011 | 27 | 0.991 | -0.007 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Platforms for OS DBMS was *for which platforms is the database available* (M= 4.667), second important was *Multi-language support* (M= 3.167).

The most important sub-category of Platforms for proprietary DBMS was *for which platforms is the database available* (M= 4.174), second important was *Multi-language support* (M= 3.174).

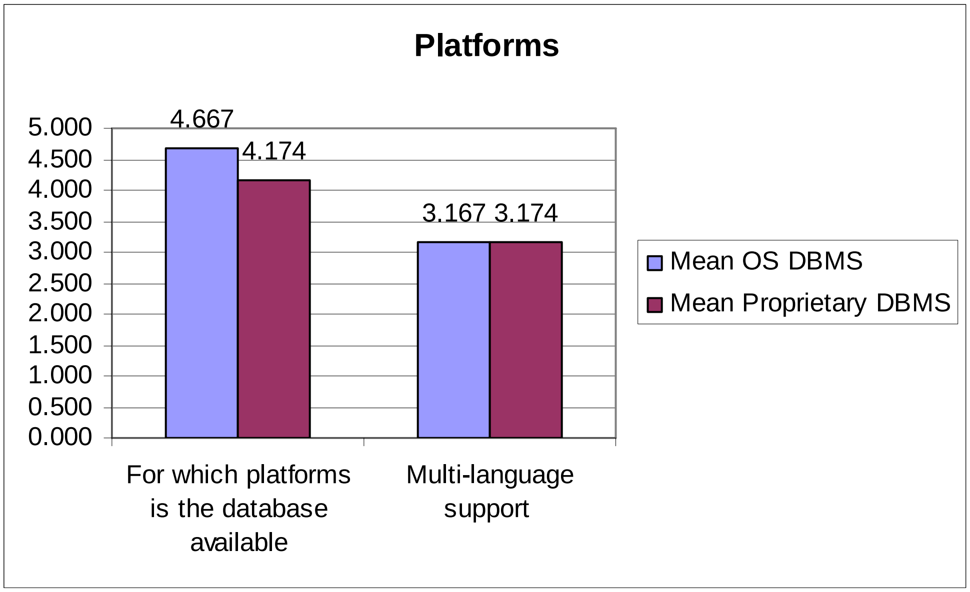


**Figure 5.8 Platforms sub-categories comparison**

There was no significant difference in scores of Standards.

**Table 5.11 Independent samples t-test for Standards**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Standards * | 0.984 | 27 | 0.334 | 0.37 |
| Support industry standards * | 1.044 | 27 | 0.306 | 0.37 |
| Interoperability with other software or systems * | 0.113 | 27 | 0.911 | 0.043 |
| No vendor/platform lock-in * | 0.761 | 27 | 0.453 | 0.384 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Standards for OS DBMS was *Support industry standards* (M= 4.5), second *important was No vendor/platform lock-in* (M= 4.167). Third important sub category was *Interoperability with other software or systems* (M= 4).

The most important sub-category of Standards for proprietary DBMS was *Support industry standards* (M= 4.130), second important was *Interoperability with other software or systems* (M= 3.957). Third important sub category was *No vendor/platform lock-in* (M= 3.783).

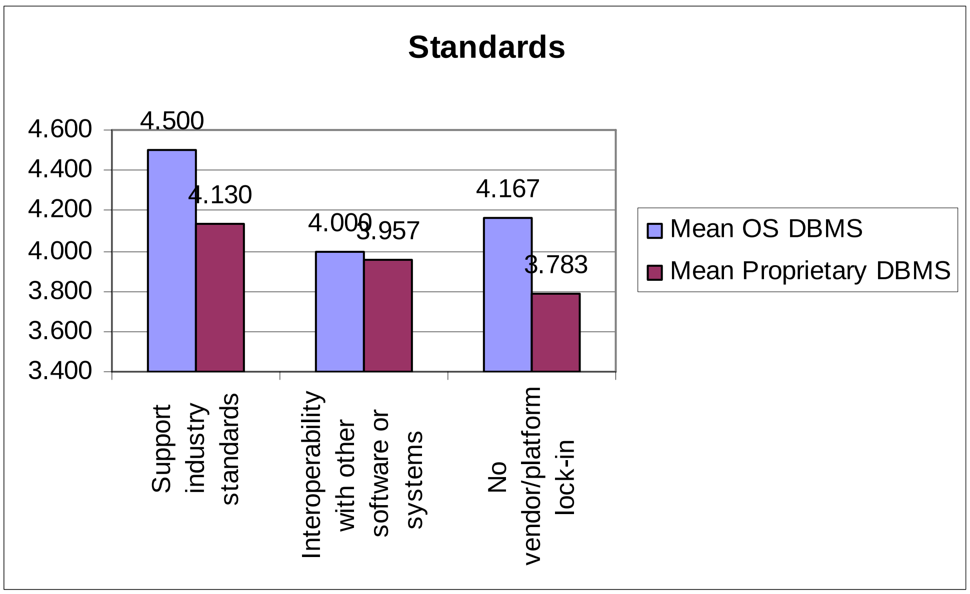


**Figure 5.9 Standards sub-categories comparison**

There was no significant difference in scores of Management.

**Table 5.12 Independent samples t-test for Management**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Management * | -0.371 | 27 | 0.714 | -0.138 |
| Ease of management * | -0.022 | 27 | 0.983 | -0.007 |
| Ability to manage more servers * | -0.604 | 27 | 0.551 | -0.254 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

The most important sub-category of Management for OS DBMS was *Ease of management* (M= 4.167), second important was *Ability to manage more servers* (M= 3.833).

The most important sub-category of Management for proprietary DBMS was *Ease of management* (M= 4.174), second important was *Ability to manage more servers* (M= 4.087).

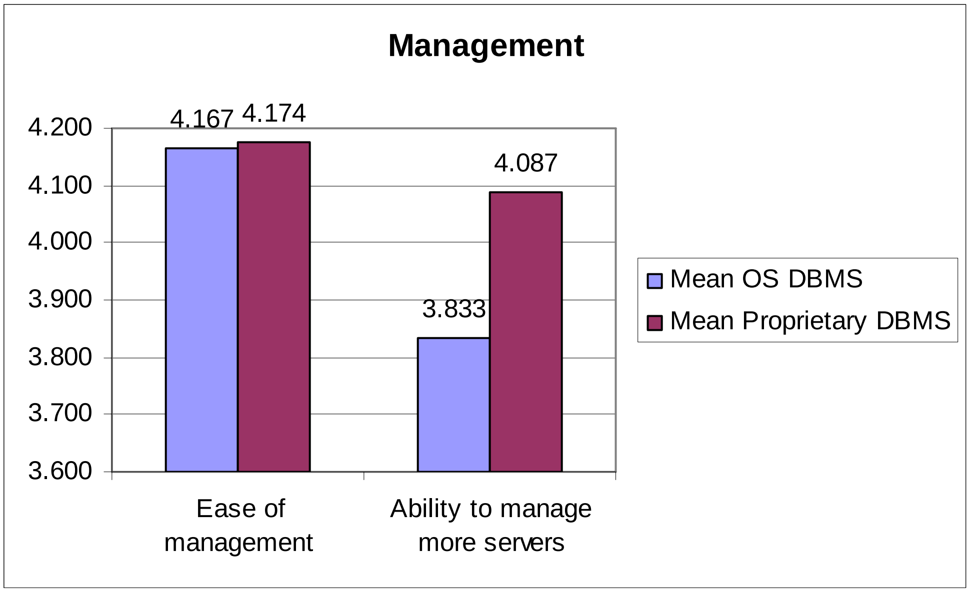


**Figure 5.10 Management sub-categories comparison**

There was a significant difference in scores of Support sub-categories. First was *Standard support* for OS DBMS (M=3.5 SD= 0.837) and proprietary DBMS [M= 4.261; SD= 0.752; t(27)= -2.161; p= 0.040]. Second significant difference was in *Advanced support programs* for OS DBMS (M= 3.333; SD= 0.516) and proprietary DBMS [M= 4.174 SD= 0.650; t (27) = -2.921; p= 0.007]. Third significant difference was in *Consulting* for OS DBMS (M= 3.167; SD= 0.408) and proprietary DBMS [M= 4 SD= 0.798; t (27) = -2.453; p= 0.021]. There was no significant difference in scores of *24x7x365 support*, *worldwide support, Training and 3rd-party products and services*.

**Table 5.13 Independent samples t-test for Support**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Support * | -1.642 | 27 | 0.112 | -0.688 |
| Standard support * | -2.161 | 27 | 0.04 | -0.761 |
| Advanced support programs * | -2.921 | 27 | 0.007 | -0.841 |
| 24x7x365 support * | -1.861 | 27 | 0.074 | -0.746 |
| Worldwide support * | -1.056 | 27 | 0.3 | -0.449 |
| Training * | -1.811 | 27 | 0.081 | -0.58 |
| Consulting * | -2.453 | 27 | 0.021 | -0.833 |
| 3rd-party products and services * | -0.591 | 27 | 0.56 | -0.283 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

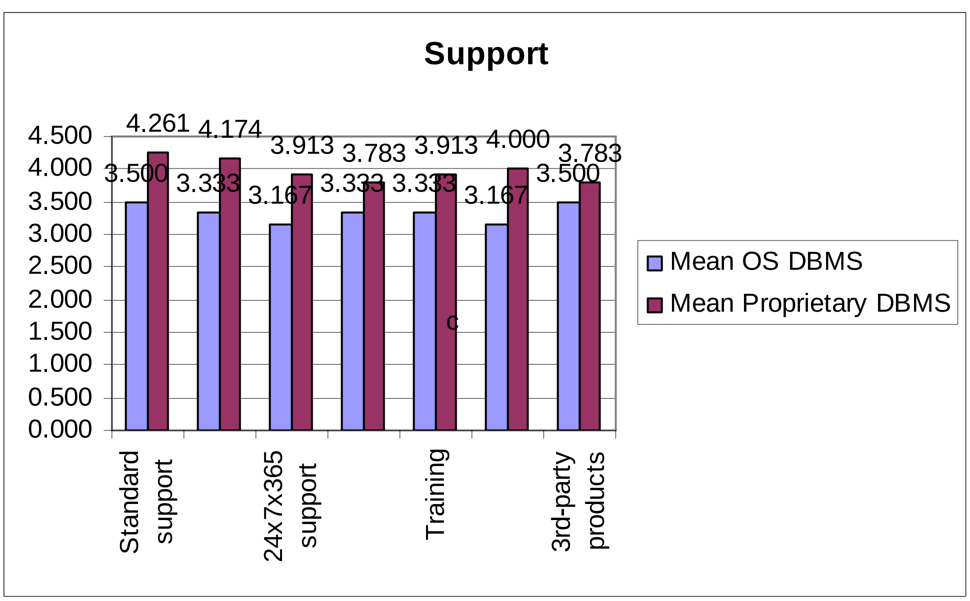


**Figure 5.11 Support sub-categories comparison**

There was a significant difference in scores of Maturity sub-categories. First was *Best practices information available* for OS DBMS (M= 3.333 SD= 0.516) and proprietary DBMS [M= 4.130; SD= 0.920; t(27) = -2.023; p= 0.053]. Second significant difference was in *Strength of company behind the product* for OS DBMS (M= 3.667; SD= 0.816) and proprietary DBMS [M= 4.304 SD= 0.635; t (27) = -2.069; p= 0.048]

There was no significant differences in scores of *Maturity of product, Reliability of current version, Frequency of updates, Ease of installation, Good documentation, Original developers still involved and committed, Large community of users, Certification available and User conferences.*

**Table 5.14 Independent samples t-test for Maturity**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Maturity * | -1.679 | 27 | 0.105 | -0.601 |
| Maturity of product * | -0.726 | 27 | 0.474 | -0.225 |
| Reliability of current version * | 0.709 | 27 | 0.484 | 0.188 |
| Frequency of updates * | -1.212 | 27 | 0.236 | -0.341 |
| Ease of installation * | -1.036 | 27 | 0.31 | -0.42 |
| Good documentation * | -1.011 | 27 | 0.321 | -0.348 |
| Best practices information available * | -2.023 | 27 | 0.053 | -0.797 |
| Original developers still involved and committed * | -0.142 | 27 | 0.888 | -0.065 |
| Strength of company behind the product * | -2.069 | 27 | 0.048 | -0.638 |
| Large community of users * | -0.552 | 27 | 0.585 | -0.217 |
| Certification available * | -1.63 | 27 | 0.115 | -0.739 |
| User conferences * | -0.971 | 27 | 0.34 | -0.514 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

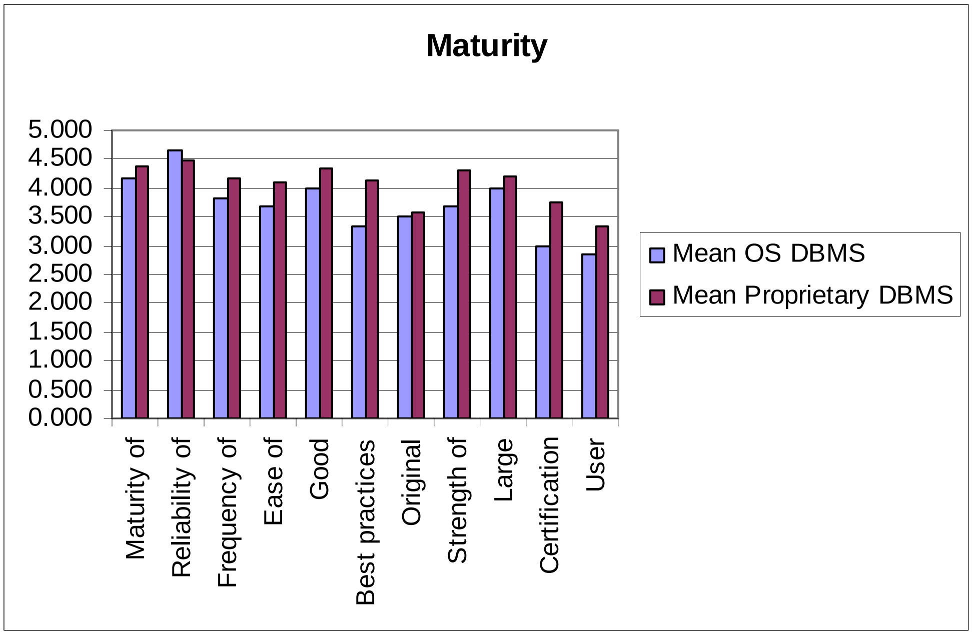


**Figure 5.12 Maturity sub-categories comparison**

There was a significant difference in scores of Expertise sub-categories. First was *Availability of experts* for OS DBMS (M= 3.333 SD= 1.033) and proprietary DBMS [M= 4.522; SD= 0.511; t(5.653)= -2.733; p= 0.036]. Second significant difference was in *Certified consultants* for OS DBMS (M= 3; SD= 0.632) and proprietary DBMS [M= 3.826 SD= 0.887; t (27) = -2.131; p= 0.042]

There was no significant difference in scores of *Books and reference material and Certified trainers.*

**Table 5.15 Independent samples t-test for Expertise**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Expertise * | -1.177 | 27 | 0.249 | -0.348 |
| Availability of experts ** | -2.733 | 5.653 | 0.036 | -1.188 |
| Books and reference material * | -1.841 | 27 | 0.077 | -0.761 |
| Certified trainers * | -1.951 | 27 | 0.061 | -0.739 |
| Certified consultants * | -2.131 | 27 | 0.042 | -0.826 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

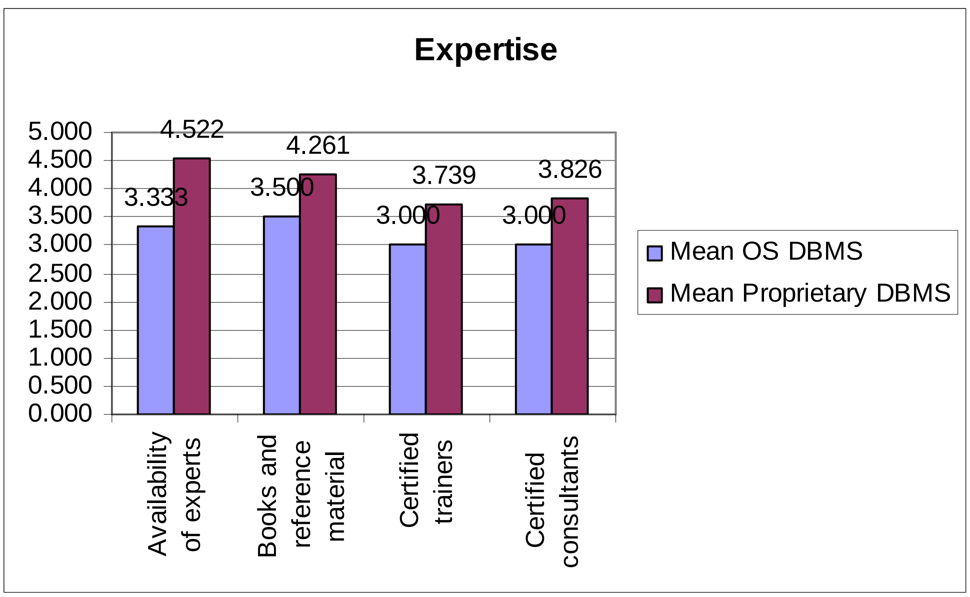


**Figure 5.13 Expertise sub-categories comparison**

There was a significant difference in scores of References sub-categories. First was *References to talk to, preferably in your industry* for OS DBMS (M= 2.833 SD= 1.329) and proprietary DBMS [M= 3.870; SD= 0.920; t(27)= -2.242; p= 0.033]. Second significant difference was in *Good experience with support* for OS DBMS (M= 3.333; SD= 1.033) and proprietary DBMS [M= 4.174 SD= 0.576; t (27) = -2.680; p= 0.012]

There was no significant difference is scores of *Experience on performance, reliability, scalability and Best practices available.*

**Table 5.16 Independent samples t-test for Reference**

| Selection Criteria for DBMS | t-test for Equality of Means | | | Mean Difference |
|---|---|---|---|---|
| | t | df | Sig. (2-tailed) | |
| Reference ** | -1.31 | 6.056 | 0.238 | -0.87 |
| References to talk to, preferably in your industry * | -2.242 | 27 | 0.033 | -1.036 |
| Experience on performance, reliability, scalability ** | -1.734 | 5.655 | 0.137 | -0.884 |
| Best practices available * | -1.75 | 27 | 0.091 | -0.623 |
| Good experience with support * | -2.68 | 27 | 0.012 | -0.841 |

.05 = Sig level
* = Equal variance assumed
** = Equal variance not assumed

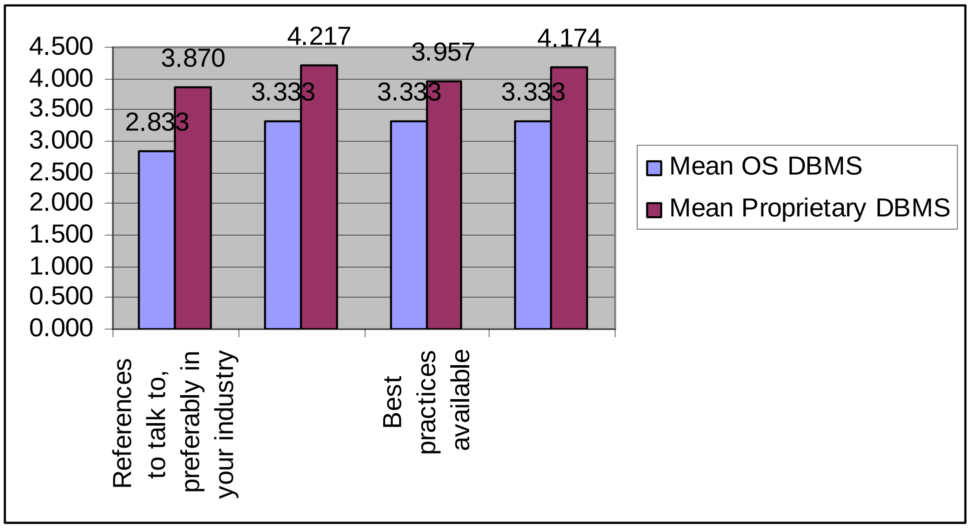


**Figure 5.14 References sub-categories comparison**

# 6. CONCLUSION AND RECOMMENDATION

## 6.1 Conclusion

The response rate remained low and analysis carried out on a limited sample, which still proved some statistical significance. The research results are limited to developing countries and may not give over all picture of software industry. The research results are viable for developing countries with low GDP, high software piracy rate and other similar economic conditions like Pakistan. The economic conditions in developed countries are different so the same research methodology in developed world may show different results.

The following conclusion is from section 2 of questionnaire.

**Table 6.1 Research conclusion**

| Hypothesis | Evaluation Criteria | Conclusion |
|---|---|---|
| H1: The importance of selection criteria practices differ in the case of the OS DBMS against proprietary DBMS | Total Cost of Ownership | Strongly Agree |
| | Scalability | Strongly Agree |

As difference in some of the major factors in DBMS software selection criteria practices could be assumed from theory and previous researches but this research identified these differences statistically. TCO (M=4.833) is ranked second most important category in OS DBMS and tenth in case of proprietary DBMS (M=4.087)

There are many companies in Pakistan who use proprietary DBMS might not be paying for it, creating the illusion that they have low costs and it could be guess from the overall piracy rate in the country. This is also one of the reasons the TCO does not figure high importance criteria in case of proprietary DBMS in Pakistan in Table 5.3.

Scalability (M=4.565) is fourth important category in proprietary DBMS and stands eleventh in case of OS DBMS (M=3.5). Scalability, in this context, suggests the size of data or problem the program can handle. Indicator for size is number of employees. Companies with less then 100 employees considered small and medium enterprises. Companies with more then 100 employees considered big enterprises.

OS respondents are small and medium size companies while 9 respondents of proprietary DBMS are large size companies. Large companies strive for some evidence that the program has been used that way before. Therefore proprietary DBMS users ranked Scalability importance high in Table 5.3.

The following conclusion is from section 3 of questionnaire.

**Table 6.2 Research conclusion**

| **Hypothesis** | **Evaluation Criteria** | **Conclusion** |
|---|---|---|
| H1: The importance of selection criteria practices differ in the case of the OS DBMS against proprietary DBMS | **Support**<br>• Standard support<br>• Advanced support programs<br>• Consulting | <br>Strongly Agree<br>Strongly Agree<br><br>Strongly Agree |
| | **Maturity**<br>• Best practices information available<br>• Strength of company behind the product | Strongly Agree<br><br>Strongly Agree |
| | **Expertise**<br>• Availability of experts<br>• Certified consultants | Strongly Agree<br><br>Strongly Agree |
| | **Reference**<br>• References to talk to, preferably in your industry<br>• Good experience with support | Strongly Agree<br><br>Strongly Agree |

There is a significant difference in scores of Support sub-categories. In OS DBMS case it can be risky to depend solely on community support without having in-house experience with the product. Proprietary DBMS support is usually only provided by the proprietary vendor and there are several competing companies offering support.

There is a significant difference in scores of Maturity sub-categories. Proprietary DBMS users ranked two of Maturity sub categories high against open source. Many technical specifications that are sometimes considered standards are proprietary rather than being open, and are only available under restrictive contract terms (if they can be obtained at all) from the organization that owns the copyright for the specification. If the software manufacturer decides to discontinue development of the product, no one has the right to take the program and continue development on it.

There is a significant difference in scores of Expertise sub-categories. Proprietary DBMS users ranked two of Expertise sub categories high against open source. The most sophisticated and useful software on the market can quickly become a burden if the vendor does not have the staff to adequately train users and respond to help requests. In both cases companies offer services, those are connected to the product itself. There is a lot of competition among proprietary vendors. This is very significant for Proprietary DBMS users to select a right vendor for right product.

There is a significant difference in scores of Reference sub-categories Proprietary DBMS users ranked two of Reference sub categories high against open source. There are lots of companies, who implemented proprietary DBMS in Pakistan. The best way to check references is to find someone who has done this before. Taking this step increased the importance of Reference sub-categories in proprietary DBMS decision.

## 6.2 Further Research Suggestions

This paper addresses the challenges, which management encounters when faced with selection of new software and in particular comparison of Open Source Software against proprietary software.

This research provides essential data for relevant government departments to prepare strategies for implementing open source applications in the IT infrastructure of country. Further it provides a good foundation for relevant teaching and further research in the tertiary institutions in Pakistan.

Although the research so far rests on a limited sample, which proved some statistical significance, further research with case studies may suggest different trends. Another area of research in developing countries can be which government agencies could benefit from a transition to OSS.

## REFERENCES

### Books & Journals

A Guide to Developing an Enterprise Open Source Strategy, the Rise of Open Source and the LAMP Stack, A MySQL Business White Paper July 2004.

A Guide to Lower Database TCO. How the Open Source Database MySQL Reduces Costs by as Much as 90%, A MySQL Business White Paper December 23, 2003

Bernard Marr and Andy Neely (2003), Automating the balanced scorecard–selection criteria to identify appropriate software applications.

David A. Wheeler Revised as of January 5, 2005, How to Evaluate Open Source Software / Free Software (OSS/FS) Programs.

David A. Wheeler, Revised as of February 8, 2005. Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!

Dr. Nic Peeling and Dr Julian Satchell (2001), Analysis of the Impact of Open Source Software, QinetiQ ltd.

Dr. Thorsten Wichmann (2002), Evidence from Germany, Sweden and UK. Use of Open Source Software in Firms and Public Institutions, BERLECON RESEARCH GmbH.

Gregory McFarland, Andres Rudmik, and David Lange (18 December 1997), Object-Oriented Database Management Systems Revisited, DoD Data & Analysis Center for Software (DACS)

Jean-Michel Dalle and Nicolas Jullien (2001), Open Source vs Proprietary Software.

Jesus M. Gonzalez-Barahona (2004) Free Software/Open Source: Information Society Opportunities for Europe?

Kevin Lomangino, Cara S. Kaufman, Alma J. Wills January 2002, Implementing Information Technology Systems, The Sheridan Press.

Larry Ellison (2003), Oracle RAC on Linux:Database Service Utopia? Linxcel Europe Ltd

Qiping Shen, Heng Li, Liyin Shen, Derek Drew and Jacky Chung, Benchmarking the use of information technology by the quantity surveying profession. Benchmarking: An International Journal Vol. 10 No. 6, 2003

Redners tryckeri (2003), Free and Open Source Software - A feasibility study, The Swedish Agency for Public Management Publication Service. ISBN: 91-7220-526-1

Sanjiva Weerawaraba and Jivaka Weeratunga (January 2004), Open Source in Developing Countries, Published by Sida 2004, Department for Infrastructure and Economic Cooperation.

Thomas M. Connolly, Carolyn E. Begg, Database systems a practical approach to design, implementation and management

Vicki Martin (October 2003), Why DB2 vs Open Source Databases Sales Guide, IBM Software Group

**Websites**

(Alexander Limi, Alan Runyan). (2000)
http://www.fsf.org/philosophy/free-sw.html
Last Accessed on January 13, 2006

Berloon Research GmbH
http://www.berlecon.de/studien/downloads/200207FLOSS_Use.pdf
Last Accessed on September 22, 2006

Bruce Perens (1997)
http://www.opensource.org/docs/definition_plain.html
Last Accessed on October 20, 2006

DBMS
www.dbmaker.com.tw
Last Accessed on June 10, 2005

Database benchmarking
http://www.eweek.com/slideshow_viewer/0,2393,l=&s=1590&a=23120&po=1,0
0.asp
Last Accessed on February 20, 2006

Database benchmarking
http://www.dba-oracle.com/art_db_benchmark.htm
Last Accessed on March 2, 2005

Database Knowledge Base
http://database.ittoolbox.com/nav/t.asp?t=376&p=376&h1=376
Last Accessed on November 7, 2005

Dmsoft Technologies
http://data-conversions.net/wp01.html
Last Accessed on September 22, 2006

Free Software Foundation
http://www.fsf.org
Last Accessed on September 20, 2006

Mitch Stoltz (1999)
http://www.netaction.org/opensrc/oss-whole.html
Last Accessed on September 22, 2006

MySQL AB
http://dev.mysql.com/doc/mysql/en/index.html
Last Accessed on November 7, 2005

Online Questionnaire Link
http://www.pakistanopensource.org/projects/survey
Last Accessed on October 22, 2006

Open Source Initiative
http://www.opensource.org
Last Accessed on September 7, 2006

Open Source Projects
http://greg.abstrakt.ch/docs/OSP_framework.pdf
Last Accessed on August 2, 2005

SPSS
www.spss.com
Last Accessed on June 7, 2005

Total Cost of Ownership
http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf
Last Accessed on June 7, 2006

Yellow Pages Pakistan
http://www.yellowpagespk.com/
Last Accessed on June 7, 2005

## **APPENDIX:** Online Questionnaire

**Welcome!**

This questionnaire is meant for a group of approached companies from Pakistan. It aims to compare selection criteria of open source database server software, specifically MySQL against the selection criteria of proprietary database server software in Pakistan.

The results will be used for my Master's thesis at Hanken, the Swedish School of Economics and Business Administration in Helsinki, Finland.

The ultimate aim is to look at the relevance of Free and Open Source Software FOSS for the developing countries like Pakistan.

If you have any queries, please email me at sohail dot khan (at) hanken dot fi.

**Section 1**

| | |
|---|---|
| **1** | **Please answer the following questions.** |
| **1.1** | Name of company: |
| **1.2** | Name of person: |
| **1.3** | Title of post: |
| **1.4** | Company URL: |
| **1.5** | Industry:<br>Options: **Banking & Finance** **IT & Communication** **Manufacturing**<br>**Consumer Goods** **Other** |
| **1.6** | Intensity of IT in company (Importance of IT in developing products, services)<br>Options: **Low** **High** |
| **1.7** | Number of employees:<br>Options: **1-20** **20-50** **50-100** **more than 100** |
| **1.8** | Please select which Database Software server is in use in your company.<br>**Mysql** **PostgreSQL** **Oracle**<br>**DB2** **Microsoft SQL** **Other** |

**Section 2**

| | |
|---|---|
| **2** | **Please inform how important each of the following criteria was on average for your decision in favor of Open source or proprietary database product:** |
| **2.1** | **Total Cost of Ownership**<br>very important important neutral less |

| | |
|---|---|
| | important<br>not important |
| **2.2** | **Performance**<br>very important important neutral less important<br>not important |
| **2.3** | **Reliability**<br>very important important neutral less important<br>not important |
| **2.4** | **Scalability**<br>very important important neutral less important<br>not important |
| **2.5** | **Security**<br>very important important neutral less important<br>not important |
| **2.6** | **Platforms**<br>very important important neutral less important<br>not important |
| **2.7** | **Standards**<br>very important important neutral less important<br>not important |
| **2.8** | **Management**<br>very important important neutral less important<br>not important |
| **2.9** | **Support**<br>very important important neutral less important<br>not important |
| **2.10** | **Maturity**<br>very important important neutral less important<br>not important |
| **2.11** | **Expertise**<br>very important important neutral less important<br>not important |
| **2.12** | **References**<br>very important important neutral less important<br>not important |

**Section 3**

| | |
|---|---|
| **3** | **Please inform how important each of the following criteria was on average for your decision in favor of Open source or proprietary database product**. |
| **3.1** | **Total Cost of Ownership** |
| **3.1a** | **Initial licensing cost**<br>very important important neutral less important<br>not important |
| **3.1b** | **Hardware expenditure** (Hardware cost is defined as cost of equipment)<br>very important important neutral less important<br>not important |
| **3.1c** | **Administration, engineering and support costs** (Cost of deploying, developing, maintaining and supporting databases)<br>very important important neutral less important<br>not important |
| **3.1d** | **System downtime cost** (Loss of productivity and revenue during the time server is down)<br>very important important neutral less important<br>not important |
| **3.1e** | **Training cost** (Cost of hiring and training people to support database)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.2** | **Performance** |
| **3.2a** | **Speed**<br>very important important neutral less important<br>not important |
| **3.2b** | **Independent benchmarks on comparable hardware and load**<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.3** | **Reliability** |

| | |
|---|---|
| **3.3a** | **Server and application uptime**<br>very important important neutral less important<br>not important |
| **3.3b** | **Test of code quality**<br>very important important neutral less important<br>not important |
| **3.3c** | **Responsiveness to bug reports** (How fast are bugs fixed when discovered and reported)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.4** | **Scalability** |
| **3.4a** | **Ability to scale out more servers as necessary** (Add & configure more servers, expand capacity)<br>very important important neutral less important<br>not important |
| **3.4b** | **Independent benchmarks** (By experienced database benchmarking organization that is not tied to any hardware or software vendor)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.5** | **Security** |
| **3.5a** | **No or low number of known vulnerabilities**<br>very important important neutral less important<br>not important |
| **3.5b** | **Independent validation of security** (Validation of security by third party)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.6** | **Platforms** |

| | |
|---|---|
| **3.6a** | **For which platforms is the database available** (Server operating system: Windows, Linux, Solaris etc)<br>very important important neutral less important<br>not important |
| **3.6b** | **Multi-language support** (English, Urdu etc)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.7** | **Standards** |
| **3.7a** | **Support industry standards**<br>very important important neutral less important<br>not important |
| **3.7b** | **Interoperability with other software or systems** (Integrated access to the web, internal networks, and corporate mainframes)<br>very important important neutral less important<br>not important |
| **3.7c** | **No vendor/platform lock-in** (Process by which vendors try to coerce a customer into continued use of a software by making it impossible or difficult to migrate to a different solution)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.8** | **Management** |
| **3.8a** | **Ease of management** (Availability of tools to efficiently manage by database administrator)<br>very important important neutral less important<br>not important |
| **3.8b** | **Ability to manage more servers** (Management of additional servers with minimum resources i.e hardware, staff, space etc)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.9** | **Support** |

| | |
|---|---|
| **3.9a** | **Standard support**<br>very important important neutral less important<br>not important |
| **3.9b** | **Advanced support programs**<br>very important important neutral less important<br>not important |
| **3.9c** | **24x7x365 support** (7x24x365 means that you can get assistance no matter what time it is)<br>very important important neutral less important<br>not important |
| **3.9d** | **Worldwide support**<br>very important important neutral less important<br>not important |
| **3.9e** | **Training**<br>very important important neutral less important<br>not important |
| **3.9f** | **Consulting**<br>very important important neutral less important<br>not important |
| **3.9g** | **3rd-party products and services**<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.10** | **Maturity** |
| **3.10a** | **Maturity of product** (The software has gone through a cycle of development resulting in wider use)<br>very important important neutral less important not important |
| **3.10b** | **Reliability of current version**<br>very important important neutral less important<br>not important |
| **3.10c** | **Frequency of updates** (How many upgrades have been made, and how often)<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.10d** | **Ease of installation**<br>very important important neutral less important<br>not important |
| **3.10e** | **Good documentation**<br>very important important neutral less important<br>not important |
| **3.10f** | **Best practices information available**<br>very important important neutral less important<br>not important |
| **3.10g** | **Original developers still involved and committed**<br>very important important neutral less important<br>not important |
| **3.10h** | **Strength of company behind the product**<br>very important important neutral less important<br>not important |
| **3.10i** | **Large community of users**<br>very important important neutral less important<br>not important |
| **3.10j** | **Certification available**<br>very important important neutral less important<br>not important |
| **3.10k** | **User conferences**<br>very important important neutral less important<br>not important |
| **3.11** | **Expertise** |
| **3.11a** | **Availability of experts** (Especially in the host country, Pakistan)<br>very important important neutral less important<br>not important |
| **3.11b** | **Books and reference material**<br>very important important neutral less important<br>not important |
| **3.11c** | **Certified trainers** (Does vendor have the certified staff to adequately train users)<br>very important important neutral less important |

| | |
|---|---|
| | not important t |
| **3.11d** | **Certified consultants**<br>very important important neutral less important<br>not important |

| | |
|---|---|
| **3.12** | **References** |
| **3.12a** | **References to talk to, preferably in your industry** (Recommended by others who have been using the software)<br>very important important neutral less important<br>not important |
| **3.12b** | **Experience on performance, reliability, scalability**<br>very important important neutral less important<br>not important |
| **3.12c** | **Best practices available**<br>very important important neutral less important<br>not important |
| **3.12d** | **Good experience with support**<br>very important important neutral less important<br>not important |

**Section 4**

| | |
|---|---|
| **4** | Additional Comments (optional) |