

## Performance Evaluation of a Database Management System (A Case Study of INTERBASE and MySQL)

A.O. Afolabi and A.O. Ajayi

Department of Computer Science and Engineering,  
Ladoke Akintola University of Technology, Ogbomoso, Nigeria

---

**Abstract:** A database management system (DBMS) is a system, usually automated and computerized, for the management of any collection of compatible and ideally normalized data. The focus of this research is to compare 2 database technologies for software developers to have the advantage of selecting the best technology in terms of Execution time, Resource consumption and Space and data capacity. In the course of this research work the following database management systems (DBMS) were considered, MySQL Server and Borland Interbase Server. Selecting from diverse existing software development tools, Microsoft Visual Basic 6.0 was used in designing the front-end engine for the entire database system analysis and evaluation. It was found out that for large quantities of data that needs to be tracked and also seek or make use of selection, update and delete transaction will request more of the interbase server, because of its high speed in performing those transaction, while for insertion transaction MySQL server performed better.

**Key words:** DBMS, performance evaluation, MySQL Server, transaction

---

### INTRODUCTION

The advent of data processing and information storage has brought about a dynamic change both in information technology industry and even the commercial world. Data provide the basis for the advancement and development in every field of study. With this background, it is therefore, necessary to ascertain the database management system of better performance for improvement of result.

A database management system (DBMS) is a program (or more typical, a suit of them) designed to manage a database (a large set structure data) and run operation on the data request by numerous clients. Typical example of DBMS use includes accounting, human resource and customer support system. Originally found only in large organization with the computer hardware needed to support large data sets, DBMS have more recently emerge as a fairly standard part of any company back office. A database management system (DBMS) is a system, usually automated and computerized, for the management of any collection of compatible and ideally normalized data. Sometimes DBMS's are built around a private multitasking kernel with built in networking support although nowadays these function are left to the operating system. Databases have been in use since the

earliest days of electronic computing, but the vast majority of these were custom program written to access custom database. Unlike modern system which can be applied to widely different database and needs these systems were tightly linked to the database in order to gain speed at the expense of flexibility (Karl, 2006). These research focus on 2 remote database technologies which are INTERBASE and MySQL.

### MATERIALS AND METHODS

The research is carried out using methodology that involves certain and a well defined step to achieve develops objectives these include:

- Detailed study of the two database, their features, service the offer, weakness and strength.
- Subject the two database management system into stress testing by generating random numbers to populate the database with:
  - 1000 Thousand records.
  - 10000 Thousand records.
  - 50000 Thousand records.
  - 100000 Thousand records.
  - 200000 Thousands records.

- Hardware specification: This will be general for the database.

|                  |   |                        |
|------------------|---|------------------------|
| RAM              | - | 128                    |
| Operating system | - | Windows XP             |
| Microprocessor   | - | Pentium III            |
| Platform         | - | Microsoft Visual Basic |

**Over view of database management system:** A database management system (DBMS) is a system, usually automated and computerized, for the management of any collection of compatible and ideally normalized, data. Databases have been in use since the earliest days of electronic computing, but the vast majority of these were custom programs written to access custom databases. Unlike modern systems which can be applied to widely different databases and needs, these systems were tightly linked to the database in order to gain speed at the price of flexibility.

**Object DBMS:** Multidimensional DBMS did have one lasting impact on the market; they led directly to the development of the object database systems. Based on the same general structure and concepts as the multidimensional systems (Afolabi, 2002). Object database management (ODBMSs) are designed to directly support the object data model, composite object and relation in the database. That is, the programming constructs being used in the object oriented (OO) programming world could be used directly in the database, instead of first being converted to some other format. This could happen because of the multidimensional system's concepts of ownership. In an Object Oriented program a particular object will typically contain others; for example, the object representing Bob may contain a reference to a separate object referring to Bob's home address. Adding support for various Object.

Oriented languages and polymorphism re-created the multidimensional systems as object databases, which continue to serve a niche today (Bancilhom, 2000).

## RESULTS

### Performance evaluation of dbms

**Choice of development tools:** In the course of implementing and executing this research work the following database management systems (DBMS) were considered, MySQL Server and Borland Interbase Server. Selecting from diverse existing software development tools, Microsoft Visual Basic 6.0 was used in designing the front-end engine for the entire database system analysis and evaluation.

**MySQL server database management system:** MySQL Server 4.1 is a Database Server and is known for its support for many data types, scalability, simplicity and efficiency. It is a remote database server with a heavy support for client-server and distributed computing. It allows easy access and migration from and to other database systems such as Oracle Server, Mimer, Visual FoxPro Ingress Server and Postures Server among others. It is an Open Source project of MySQL AB Community and MySQL Network. The source binary code is open to whosoever wishes to develop on it on the condition that he/she will make it open to others after the development. MySQL Network also provides a comprehensive knowledge base library that includes hundreds of technical articles resolving difficult problems on popular database topics such as performance, replication and migration.

The MySQL® database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used in more than 10 million installations ranging from large corporations to specialized embedded applications on every continent in the world (Khandare, 2004).

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control. Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful.

### Borland interbase server database management system:

InterBase is a high performance, relational database management system, providing SQL92 conformance, good transaction model, extensible user defined functions and powerful stored procedures (Bill, 2006). Connecting to and provide data for your application, web server or development system, the connectivity options available to potential user of InterBase are legion, in fact to a new recruit, the number and range of options may be somewhat daunting. Compared to many of the other relational databases that are available, InterBase appears to be one of the simplest to install and by using the available console and command line tools to connect and retrieve data via simple SQL statements. Interbase also has API function which allows it to connect conveniently to any data sources. API function used by Interbase to connected to the application program using (ODBC) is

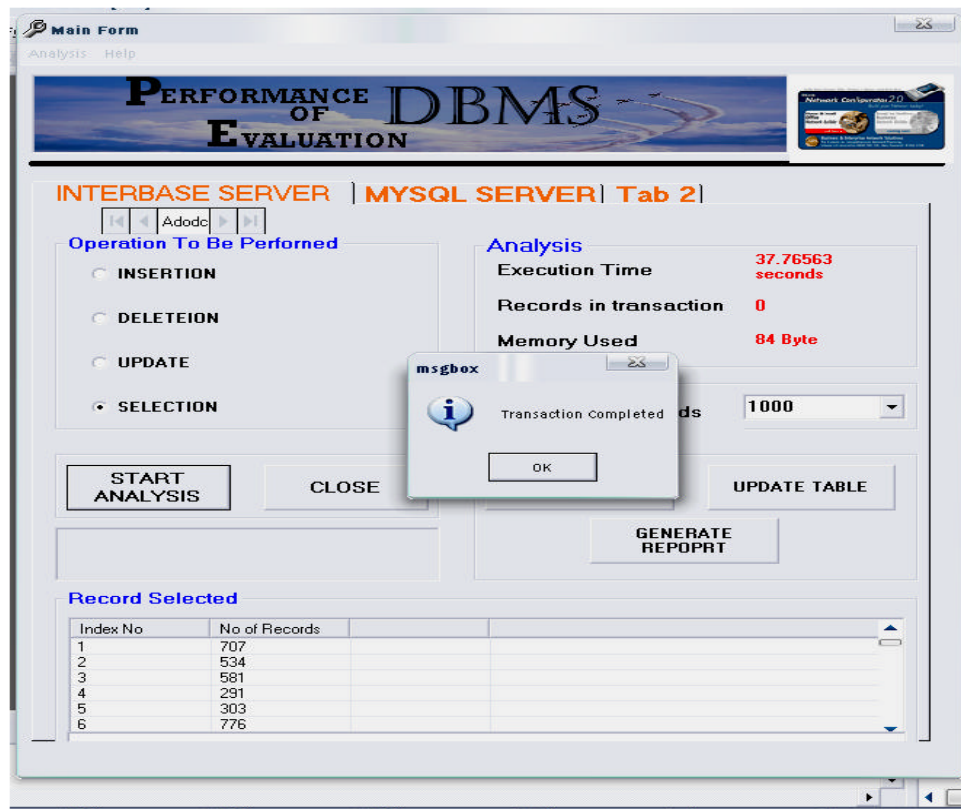


Fig. 1: Interface for INTERBASE server database

namely GDS32.DLL for windows or on UNIX platforms libgds so (or variations on the name, normally including so, gds and lib in some combination). This library irrespective of its name provides the basic InterBase API (Deitel *et al.*, 2002). This is the collection of function calls normally beginning with isc, which provides the functionality required for all (with one exception, I will mention later) the connection methods to InterBase. This talk is aimed at both the new and existing user of InterBase who may be in this position. Having installed InterBase and verified its running, then having created a few test tables and inserted some data using either IBConsole or isql. InterBase server runs on a variety of platforms, including Microsoft Windows. NET, Windows 2000 and windows 98/ME, Linux and several UNIX operating systems. The InterBase server software makes efficient use of system resources on the server node. The server process uses little more than 1.9 MB of memory. Typically, each client connection to the server adds approximately 115 KB of memory. This varies based on the nature of the client applications and the database design, so the figure is only a baseline for comparison. The minimal software installation requires disk space ranging from 9-12MB, depending on platform. During operation,

InterBase's sorting routine requires additional disk space as scratch space. The amount of space depends on the volume and type of data the server is requested to sort.

**Microsoft visual basic 6.0:** Visual Basic evolved from BASCI (Beginner's All-purpose Symbolic Instruction Code). The wide spread of BASIC with various types of computers led to many enhancements to the language. With the development of Microsoft. Window *graphical user interface*. A Windows based application of BASIC was developed.

Visual Basic is a Microsoft Windows programming language; it was created in an Integrated Development Environment (IDE). The IDE allow a programmer to create, run and debug Visual Basic programs conveniently. IDEs allows a programmer to create working program in a fraction of time that it would normally take to code programs without using Ides (Deitel *et al.*, 1999). Visual Basic 6.0 was used for the development of the front-end engine for the entire system. Visual Basic provide some other power features like graphical user interface, event handling, access to the WIN32 API, object-oriented features, error handling, structured programming and more.

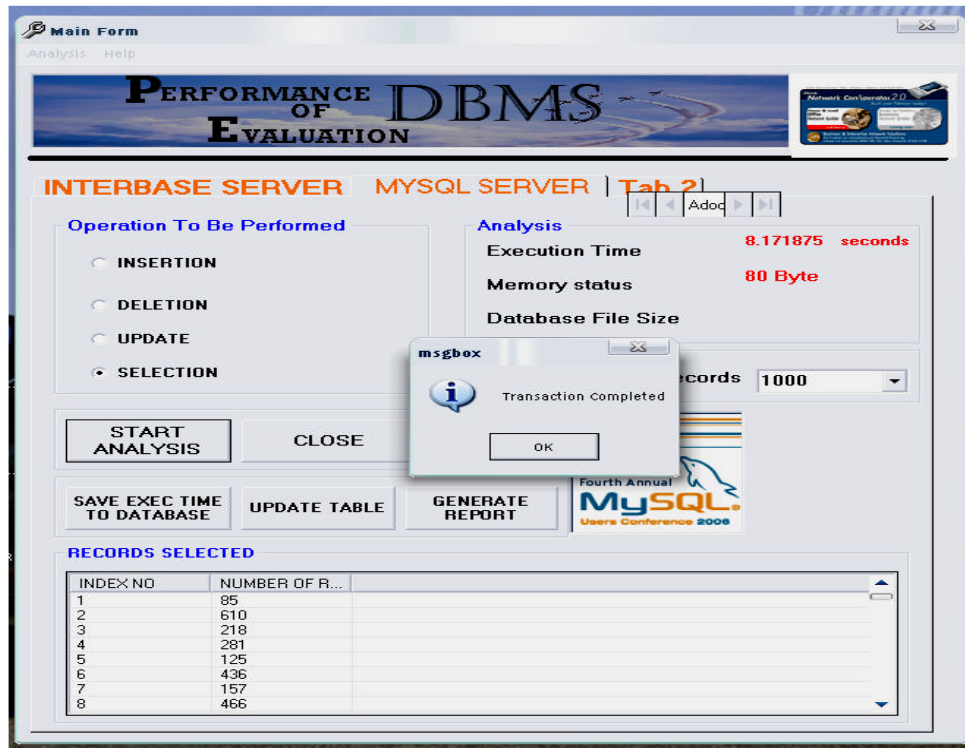


Fig. 2: Interface for MySQL server database

Figure 1 shows a select transaction in Borland interbase server Database which one thousand records was retrieved and displayed. The execution time and the memory used was captured and displayed.

**Implementation of the performance evaluation:** Figure 2 shows a select transaction in MySQL server Database which one thousand records was retrieved and displayed. The execution time and the memory used was captured and displayed.

### DISCUSSION

**Execution time for INTERBASE server and MySQL server:** Table 1 that depict the execution time of INTERBASE server, it was seen that Insertion Execution Time is slower than Select Execution Time ,Update Execution Time and the three are also slower then the Delete Execution Time. Therefore, in INTERBASE server database system the time required to execute Delete statement is faster than any other action statement.

Table 2 depict the execution time of MySQL server, it was seen that Insertion Execution Time is slower than Select Execution Time, Update Execution Time and the three are also slower then the Delete Execution Time. Therefore, in MySQL server database system the time

required to execute delete statement is faster than any other action statement.

**Comparison of interbase server with MySQL server:** Table 3 graph it is observed that the Insert execution time of MySQL server is faster than the Insertion Time in INERBASE server.

Table 4 shows the graph and table that depict the Update execution time of MySQL server is slower than the Update Time in INTERBASE server.

Table 5 graph that depict the Select execution time of MySQL server is slower than the Select execution Time in INERBASE server.

Table 6 it is observed that the Delete execution time of MySQL server is slower than the Delete execution Time in INERBASE server.

**Memory used:** Table 7 and 8, show the comparison of memory used in MySQL and INTERBASE.

The variations in the memory usage is as depicted by the graphs.

**Implementation approach:** The implementation, performance evaluation and analysis measure was achieved by benchmarking on the following database transaction processes:

**Table 1: Execution time for interbase server**

| No of records | Insert execution time (millsec) | Update execution time (millsec) | Select execution time (millsec) | Delete execution time (millsec) |
|---------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 1000          | 4195                            | 1328                            | 1742                            | 1023                            |
| 10000         | 29546                           | 1984                            | 7132                            | 1218                            |
| 50000         | 143132                          | 6445                            | 33646                           | 2984                            |
| 100000        | 283601                          | 16414                           | 70078                           | 6421                            |
| 200000        | 563710                          | 27725                           | 164046                          | 13203                           |

**Table 2: Execution time for MySQL server**

| No of records | Insert execution time (millsec) | Update execution time (millsec) | Select execution time (millsec) | Delete execution time (millsec) |
|---------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 1000          | 3554                            | 1703                            | 2375                            | 1258                            |
| 10000         | 28804                           | 3031                            | 9617                            | 2093                            |
| 50000         | 133469                          | 14304                           | 44835                           | 3835                            |
| 100000        | 268425                          | 26000                           | 92578                           | 7898                            |
| 200000        | 539796                          | 34960                           | 204304                          | 18266                           |

**Table 3: Comparison of INTERBASE server with MySQL server on insertion time**

| No of records | Insert execution time (millsec) MySQL | Insert execution time (millsec) INTERBASE |
|---------------|---------------------------------------|---|
| 1000          | 4195                                  | 3554                                      |
| 10000         | 29547                                 | 28804                                     |
| 50000         | 143133                                | 133469                                    |
| 100000        | 283606                                | 268425                                    |
| 200000        | 563711                                | 539796                                    |

**Table 6: Comparison table of INTERBASE and MySQL for delete time**

| No of records | Delete execution time (millsec) INTERBASE | Delete execution time (millsec) MySQL |
|---------------|---|---------------------------------------|
| 1000          | 1023                                      | 1258                                  |
| 10000         | 1218                                      | 2093                                  |
| 50000         | 2984                                      | 3835                                  |
| 100000        | 6421                                      | 7898                                  |
| 200000        | 13203                                     | 18266                                 |

**Table 4: Comparison of table INTERBASE and MySQL on update time**

| No of records | Update execution time (millsec) INTERBASE | Update execution time (millsec) MySQL |
|---------------|---|---------------------------------------|
| 1000          | 1326                                      | 1703                                  |
| 10000         | 1984                                      | 3032                                  |
| 50000         | 6445                                      | 14304                                 |
| 100000        | 16414                                     | 26000                                 |
| 200000        | 27726                                     | 34960                                 |

**Table 7: Memory usage (MySQL)**

| No of records | Memory usage of (Bytes) MySQL |
|---------------|-------------------------------|
| 1000          | 80                            |
| 10000         | 82                            |
| 50000         | 86                            |
| 100000        | 90                            |
| 200000        | 95                            |

**Table 5: Comparison table of INTERBASE and MySQL for selection time**

| No of records | Select execution time (millsec) INTERBASE | Select execution time (millsec) MySQL |
|---------------|---|---------------------------------------|
| 1000          | 1742                                      | 2375                                  |
| 10000         | 7132                                      | 9617                                  |
| 50000         | 33646                                     | 44835                                 |
| 100000        | 70078                                     | 92578                                 |
| 200000        | 164046                                    | 204304                                |

**Table 8: Memory usage of (INTERBASE)**

| No of records | Memory usage of (Bytes) INTERBASE |
|---------------|-----------------------------------|
| 1000          | 89                                |
| 10000         | 90                                |
| 50000         | 94                                |
| 100000        | 98                                |
| 200000        | 105                               |

- Insertion
- Update
- Retrieval
- Deletion
- Memory usage

This was done by a program to auto-generating a set Random numbers and inserting them into the master database file based on the record number chosen by the user at the pull down menu of analyzer interface engine for the insertion transaction process. The same program, using the various SQL statements like the delete and select to perform other transaction processes, but in the update transaction the set of random numbers generated

are picked multiplied by the number 2 and automatically updated back into the record of the master database file. Under each of the transaction processes the execution time was taken and tag to the record number of the process that has taken place. And for the memory usage the VBAPI (Application Programming Interface) viewer application is responsible for performing this task, it stores Win32 API function which is implemented in the DLL (Dynamic Link Library) called or referenced by programming language. The Win 32 API function that was picked is thekernel 32 and contains the internal library call the "GlobalMemoryStatus" that will be referenced by the application program to determine the size of memory used by the application running uses per time and display it.

## CONCLUSION

Interbase and MySQL are database management system technology that have brought a drastic change in information word in the information processing is enough but the format of storing of this information is of equivalent importance. It provides users with one of the simplest and most flexible DBMS solutions on the market today.

Organizations that are overwhelmed by the large quantities of data that needs to be tracked and also seek or make use of selection, update and delete transaction will request more of the interbase server because of it high speed in performing those transaction. While those that work with insertion transaction will request of MySQL SERVER. Or perhaps they are currently using a paper filing system, text documents or a spreadsheet to keep track of their of critical information. What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access,

interbase, Oracle or MySQL Server provides the software tools needed to organize that data in a flexible manner.

## REFERENCES

- Afolabi, A., 2002. Fundament of Software Engineering, Nigeria, Revealed Word Publ., pp: 38-39.
- Bill Todd, 2006. Borland Interbase White Paper on technical comparison, <http://www.nap.edu/readingroom/book/far/ch7.html>.
- Bancilhom, F., 2000. Object Oriented Database System, 7th ACM SIGART-SIGMOD-SIGART symposium on principle of database systems. Austin, Texas, pp: 34-56.
- Deitel, H.M., P.J. Deitel and T.R. Nieto, 2002. How to Program in Visual Basic Singapore. Pearson Education.
- Karl E. Peterson, 2006. Classic Visual Basic. <http://www.classicvb>.
- Khandare, S.S., 2004. Management of Oracle Database, Nagpur. Yeshwantrae Chavan College Enginerum, pp: 28-29.