

# Database Management Systems Comparative Study: Performances of Microsoft SQL Server Versus Oracle

**Cătălin Tudose\***, **Carmen Odubășteanu\*\***

\* - ITC Networks, Bucharest, Romania, e-mail: catalin\_tudose@yahoo.com

\*\* - Department of Computer Science, "POLITEHNICA" University of Bucharest, Spl. Independentei 313, 77206 Bucharest, Romania, e-mail: carmen\_od@yahoo.com

*Abstract: Modern software applications use databases on a large scale. Ones of the most used management systems are Microsoft SQL Server, with its different versions (6.5, 7.0 and 2000 being the most recent) and Oracle. This study is intended to evaluate the performances of Microsoft SQL Server versus Oracle, both running on a Windows platform. We used tests with scientific fundament and different queries. The obtained results show a series of recommendations concerning the way a database is organized and designed, but also some aspects regarding the access time to a database for some types of queries.*

*Keyword: Databases, queries, Microsoft SQL Server 2000, Oracle, index, non-cluster index*

## 1 Introduction

The use of databases for modern software applications is more and more extended. One of the problems that this kind of applications must face is the speed of query execution.

Database Management Systems include modules that make their own optimization for the access. Programmers have to take care of a good database design and to choose the best software platforms combination.

The document will evaluate the performances of the Microsoft SQL Server database management system versus Oracle, running on Windows platform.

The using database contains three tables and the important things are the fields type and number of records, not the significance of fields. For each query, their meaning is commented and the execution times on the two platforms are shown.

## 2 How the Test was Made

Computer configuration:

Intel Pentium IV 2.0 GHz

RAM memory: 512 MB

Operating system: Windows 2000 Professional

Tests were made using various queries, including selections, insertions, updates and deletions. The queries sequence was executed *the same order, using the same database, on the following platforms:*

- Windows 2000 Professional / Microsoft SQL Server 2000
- Windows 2000 Professional / Oracle 9i

The queries were executed using 3 tables: *Table1, Table2 and Table3.*

Initially, Table1 contains 100 000 records and Table2 contains 500 000 records. Table3 is empty.

*All 3 tables* have the same structure:

Field name	Field type	Length	Notes
INT1	integer		Primary key
INT2	integer		
INT3	integer		
NUM1	float		
NUM2	float		
NUM3	float		
ALPHA1	varchar	10	
ALPHA2	varchar	10	
ALPHA3	varchar	10	

Results:

	Query	Selected records	Windows 2000 Professional Oracle 9i	Windows 2000 Professional MS-SQL 2000	Notes
1	SELECT COUNT(*) FROM TABLE1;	100.000	0:00:02	0:00:01	Counts records from TABLE1
2	SELECT COUNT(*) FROM TABLE2;	500.000	0:00:02	0:00:01	Counts records from TABLE2
3	SELECT COUNT(*) FROM TABLE1 WHERE ALPHA2 LIKE 'A%';	3.846	0:00:02	0:00:01	Selection comparing on a string field

	Query	Selected records	Windows 2000 Professional Oracle 9i	Windows 2000 Professional MS-SQL 2000	Notes
4	SELECT COUNT(*) FROM TABLE2 WHERE ALPHA2 LIKE 'A%';	19.230	0:00:02	0:00:01	Selection comparing on a string field
5	SELECT COUNT(*) FROM TABLE1 WHERE NUM2 > 0.5;	33.311	0:00:02	0:00:01	Selection comparing on a float field
6	SELECT COUNT(*) FROM TABLE1 WHERE NUM2 > NUM3;	3.396	0:00:02	0:00:01	Selection comparing on float fields
7	SELECT COUNT(*) FROM TABLE1, TABLE2 WHERE TABLE1.INT3 = TABLE2.INT3;	500.000.000	0:00:32	0:00:01	Selection comparing on integer fields
8	SELECT COUNT(*) FROM TABLE1, TABLE2 WHERE TABLE1.NUM1 = TABLE2.NUM1;	100.000	0:00:06	0:00:01	Selection comparing on float fields
9	SELECT COUNT(*) FROM TABLE1, TABLE2 WHERE TABLE1.ALPHA2 = TABLE2.ALPHA2;	1.923.076.924	00:12:15	0:00:01	Selection comparing on string fields
10	INSERT INTO TABLE3 (INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3) (SELECT INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3 FROM TABLE1)	100.000	0:00:07	0:00:01	Insertion based on a selection
11	DELETE FROM TABLE3	100.000	0:00:21	0:00:01	Deletion from a table with no condition
12	INSERT INTO TABLE3 (INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3) (SELECT INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3 FROM TABLE1 WHERE INT2 > INT3)	38.016	0:00:05	0:00:01	Insertion based on a selection having a comparison on integer fields
13	DELETE FROM TABLE3	38.016	0:00:07	0:00:01	Deletion from a table with no condition

	Query	Selected records	Windows 2000 Professional Oracle 9i	Windows 2000 Professional MS-SQL 2000	Notes
14	INSERT INTO TABLE3 (INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3) (SELECT INT1, INT2, INT3, NUM1, NUM2, NUM3, ALPHA1, ALPHA2, ALPHA3 FROM TABLE1 WHERE NUM2 > NUM3)	3,396	0:00:02	0:00:01	Insertion based on a selection having a comparison on float fields
15	SELECT COUNT(*) FROM TABLE3 WHERE ALPHA2 IN (SELECT ALPHA2 FROM TABLE2 WHERE ALPHA2 IN (SELECT ALPHA2 FROM TABLE1 WHERE ALPHA2 LIKE 'A%'))	136	0:00:43	0:00:01	Cascaded sub-queries making joins on string fields and having a condition on string fields
16	SELECT COUNT(*) FROM TABLE3 WHERE ALPHA2 IN (SELECT ALPHA2 FROM TABLE2 WHERE ALPHA2 IN (SELECT ALPHA2 FROM TABLE1 WHERE NUM2 > NUM3))	3,396	0:00:43	0:00:01	Cascaded sub-queries making joins on a string fields and having a condition on float fields
17	DELETE FROM TABLE3	3,396	0:00:04	0:00:01	Deletion from a table with no condition
18	UPDATE TABLE1 SET INT2 = INT2+1 WHERE ALPHA2 LIKE 'A%'	3,846	0:00:02	0:00:01	Update having a condition on a string field
19	UPDATE TABLE2 SET ALPHA1 = 'ABC' WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE ALPHA2 LIKE 'T%')	3,846	0:00:13	0:00:01	Update making join on a string field and having a condition on a string field
20	UPDATE TABLE2 SET ALPHA1 = 'ABC' WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE NUM3 > 0.5)	90,385	0:00:13	0:00:02	Update making join on a string field and having a condition on a float field
21	UPDATE TABLE2 SET ALPHA1 = 'ABC' WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE INT3 > 50)	0	0:00:07	0:00:01	Update making join on a string field and having a condition on an integer field
22	DELETE FROM TABLE1 WHERE ALPHA2 LIKE 'A%'	3,846	0:00:07	0:00:01	Deletion from a table with a condition on a string field

	Query	Selected records	Windows 2000 Professional Oracle 9i	Windows 2000 Professional MS-SQL 2000	Notes
23	DELETE FROM TABLE2 WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE ALPHA2 LIKE 'T%')	0	0:00:07	0:04:35	Deletion from a table with a condition on a string field and a join on a string field
24	DELETE FROM TABLE2 WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE NUM3 > 0.5)	0	0:00:07	0:00:01	Deletion from a table with a condition on a float field and a join on a string field
25	DELETE FROM TABLE2 WHERE ALPHA1 IN (SELECT ALPHA1 FROM TABLE1 WHERE INT3 > 50)	0	0:00:07	0:00:01	Deletion from a table with a condition on an integer field and a join on a string field

### Conclusions

- Comparing the speed of the queries execution, the combination Windows 2000 Professional/MS-SQL 2000 gives much better results than the combination Windows 2000 Professional/Oracle 9i. Windows 2000 Professional/MS-SQL 2000 gives excellent results for conditional selections, joins, sub-queries, updates and deletions.
- The interaction between two products (Windows – operating system and MS-SQL – database management system) of the same company (Microsoft) works excellent for the speed and efficiency of database access.
- A surprising characteristic of Windows 2000 Professional/MS-SQL 2000 is that it consumes too much time for queries with a condition on a string field and a join on a string field.
- Conditional queries (having the clause WHERE) which include joins between tables execute much faster, as the records are first filtered according to that clause and the cross product will contain less records.
- For DELETE queries that do not remove any rows, the time is consumed in order to filter the records according to the given condition.
- On Oracle, the queries involving comparisons on integer fields are faster than those using comparisons on float fields, which are faster than those using comparisons on string fields. So, the recommendation is to try to use, as much as possible, integer primary and foreign keys.
- On MS-SQL, the differences are not so visible, as the queries execute with comparable speed. Our previous research indicate that for this RDBMS

comparisons on integer fields are faster than those using comparisons on string fields, which are faster than those using comparisons on float fields. The recommendation about using integer primary and foreign keys is still available.

What can be done to improve this experiment?

- Extend the tests using other operating systems and other database management systems.
- Introduce tests that will measure the time consumed for rows filtering according to the conditions and the time consumed for display/update/delete.
- A complex database will include triggers, which are special kind of stored procedures, executed when an insert/update/delete operation occurs in a database.
- We can use clustering and non-clustering indexes, to notice their performances. It is expected from them to increase the selection speed, but to decrease the insert/update/delete operations.
- To better understand what is happening “behind the closed doors” we have to analyze the characteristics of the database management system implementation. Generally, an RDBMS uses three kinds of algorithms for joins (from the simpler to the most complex): nested-loops, merge-join, and hash-join. “Nested-loops” is better for small tables, but for tables containing many records merge-join and hash-join would be more appropriated. The programmer can choose the algorithm to be applied. Some RDBMSs provide some analysis tools that can be used to compare the performances. It will also be interesting to find out what kind of data structures are used for indexing (B-trees, hash-tables). It is supposed that the multiway merge-sort algorithm is used when a cross product is calculated for a join and one of the relations (or even both) do not fit into the RAM memory.
- Tests can be repeated with increasing the RAM memory, in order to find out how this influences the execution time. Of course, it is expected that the execution will be faster having more memory, but it is important to find out a report between the amount of memory and the amount of speed.

### References

- [1] Inside MS-SQL Server (2000), Microsoft Press ACM
- [2] Lannes L. Morris-Murphy: Oracle 9i: SQL with an Introduction to PL/SQL, Thomson Course Technology 2003
- [3] J. D. Ulmann : Principles of database systems and knowledge, Rockville, Computer Science Press, 1988
- [4] Cătălin Tudose, Carmen Odubășteanu: Comparative study about the performances of the Microsoft SQL Server database management system, in Proceedings of Control Systems and Computer Science (CSCS 14), Bucharest, July 2003