

**Map Middle East 2005**  
***Evaluating different approaches of spatial database management for  
moving objects***

*Authors:*

**Somayeh Dodge***(Presenting and Contact Person)*

M.Sc. Student, Dept. of GIS Eng.

Email: s2dodge@yahoo.com

**Ali A. Alesheikh**

Assistant Professor, Dept. of GIS Eng.

Email: alesheikh@kntu.ac.ir



**Ali A. Alesheikh** is assistant professor at the K.N.Toosi University of Technology, Tehran, Iran. Dr. Alesheikh got his Ph.D. in Geomatics Engineering from the University of Calgary, AB Canada in 1998, specializing in Geospatial Information Systems. In 1993, he got his Masters in Digital Photogrammetry from the University of New Brunswick, N.B. Canada. The areas of his interests are: Online GIS, Location Based Services, Uncertainty in GIS, and visualization.

Address: Faculty of Geodesy and Geomatics Eng., K.N. Toosi University of Technology  
Vali Asr St., Vanak Sq., Tehran, Iran, Post Box: 15875-15433

Fax: +98 21 878 6213

Tel: +98 21 877 9473-5

**Abstract:**

Currently, Moving Object Database (MOD) applications are being developed in an unordered fashion. Database Management System (DBMS) technology provides a potential foundation upon which to develop these applications, however, DBMS's are currently not tuned for this purpose. The reason is that there is a critical set of capabilities that are needed by moving objects database applications and are lacking in existing DBMS's. Existing DBMS's are not well equipped to handle continuously changing data.

This paper presents behavioral aspect of the spatio-temporal databases for managing and querying moving objects. Our objective is to evaluate and develop a framework for managing and querying moving object database. The results of the evaluation are demonstrated in this paper. A commercial software package has then been selected to design and develop the capabilities required for moving object's queries.

**Key words:** Moving object, MOD, GIS, DBMS, Spatio-temporal.

## 1. Introduction

Currently, Moving Object Database (MOD) applications are being developed in an unordered fashion. Database Management System (DBMS) technology provides a potential foundation upon which to develop these applications, however, DBMS's are currently not tuned for this purpose. The reason is that there is a critical set of capabilities that are needed by moving objects database applications and are lacking in existing DBMS's. Existing DBMS's are not well equipped to handle continuously changing data. The key problem is how to model the location of moving objects and enable DBMS to predict the future location of a moving object by using efficient index and query manners. Generally, a query in MOD applications involves spatial objects (e.g. points, lines, regions, polygons) and temporal constraints.

Nowadays, spatio-temporal decision-making in life support sciences often requires the use of datasets where, in addition to the measured values, the description of the physical geometry (in terms of spatial and temporal coordinates) is also part of these sets. Spatio-temporal database is a database that embodies spatial, temporal, and spatiotemporal database concepts, and captures spatial and temporal aspects of data.

The importance of incorporating time in modeling data is widely recognized by many researches, but most of the intentions mentioned are focused around the non-spatial domain.

Time has always been a fascinating dimension of human existence. It is also an aspect in the field of geographic information science that today is getting more attention than ever before. The integration time in existing desktop GIS packages is a great challenge today. TGIS is an interdisciplinary subject that can be considered in various application domains, including environmental monitoring and decision-making, health risk assessment, ecologic hypotheses testing, geocoding, geodesy, photogrammetry, computer-aided design, transportation planning, economic development, military operations, gender studies, retail planning, and resource management, to mention only a few.

This paper presents behavioral aspect of the spatio-temporal databases for managing and querying moving objects. Our objective is to evaluate and develop a framework for managing and querying moving object database. The results of the evaluation are demonstrated in this paper. A commercial software package has then been selected to design and develop the capabilities required for moving object's queries.

## 2. Spatio-temporal continuum

Nowadays, geographotemporal decision-making in life support sciences often requires the use of datasets where, in addition to the measured values, the description of the physical geometry (in terms of spatial and temporal coordinates) is also part of these sets.

The scientific analysis of geographotemporal situations requires the introduction of

- a. The notion of a spatio-temporal continuum equipped with a coordinate system and measure of space/time distances
- b. Models and techniques that establish linkages between spatio-temporally distributed data.

A spatio-temporal continuum (or domain)  $\mathcal{E}$  is a set of points associated with a continuous spatial arrangement of events combined with their temporal order. Spatio-temporal continuity implies an integration of space with time and is a fundamental property of the mathematical formalism of natural phenomena.

Within the spatio-temporal continuum  $\mathcal{E}$ , space represents the order of coexistence of events, and time represents the order of their successive existence.

For the continuum  $\mathcal{E}$  to be useful in real-world TGIS applications it must be equipped with a coordinate system identifying points in space/time and a metric that measures "distances" in space/time, i.e.,

$\mathcal{E}$ : (coordinates  $\mathbf{P}$ , metric  $|\mathbf{dP}|$ )

Generally speaking, a coordinate system is a systematic way of referring to places, times, things and events. A point in a spatio-temporal domain  $\mathcal{E}$  can be identified by means of two separate entities: the spatial coordinates  $\mathbf{s} = (s_1, \dots, s_n) \in S \subset \mathbb{R}^n$  and the temporal coordinate  $t$  along the temporal axis  $T \subset \mathbb{R}^1$ , so that the combined space/time coordinates are

$$\mathbf{P} = (\mathbf{s}, t) \in \mathcal{E} = S \times T$$

The above union of spatial coordinates  $\mathbf{s}$  and the time coordinate  $t$  is defined in terms of their Cartesian product  $S \times T$ . [1]

### 3. Integrating time in Geographic Information Systems

A spatio-temporal GIS extends a customary two or three dimensional spatial GIS on the time axis. A space-time path is defined by the changes of locations over time (figure 1). In a spatio-temporal framework space and time are equivalent features of an event or a process. Furthermore each entity which physically exists in the real world is defined in a spatio-temporal context which can be conceptualized using spatial, temporal and attribute information. [2]

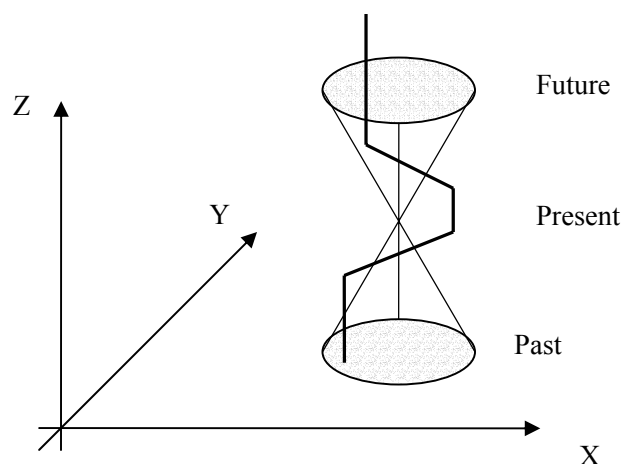


Fig. 1. Space and time path

Every spatial object used in a GIS has a temporal validity as well as one or many attribute values. The entity of a spatio-temporal process may change its spatial representation over time as well as its spatial relationship to other entities. In addition, the related attribute information may be subject to changes throughout time.

"the concept of time implies that changes occur throughout the present, the past and the future of the life span of a real-world phenomenon. A temporal GIS will aim to understand these changes and their effects over time rather than simply reproducing them by displaying a sequence of snapshots" (Wachowicz & Healey 1993).[2]

#### **4. Spatio-temporal databases**

Spatio-temporal database is a database that embodies spatial, temporal, and spatiotemporal database concepts, and captures spatial and temporal aspects of data. Spatio-Temporal Database is a special temporal database and it has all the features of temporal databases. If we look at any timestamp only, it is almost the same as the conventional spatial database. [3]

The importance of incorporating time in modeling data is widely recognized by many researchers, but most of the intentions mentioned are focused around the non-spatial domain. Particularly, temporality has been studied in banking, administrative and other commercial databases. Although the underlying concept of time handling is the same, the great difference between spatio-temporal and a spatial database is what they refer to. Since spatial data are spatially referenced, the spatial topology has to be maintained in the updating process. This maintenance of topology becomes more critical when different version of an object have to be accessible, for each given of time slices, and where correctness of topology is required.

Validity is one of the main concepts in spatial databases. Logically, no decision can be made when there is no valid data. If only the most recent data are kept, and out-dated ones are deleted, one can have the static database which is a snapshot of the real world (just like taking a photograph of a dynamic object). On the other hand, many spatial analysis applications do need to travel through the history of data and retain an impression about its old status. In this case, it can be seen that deleting data, even if it is non-valid, may be disastrous. In addition, many applications require estimation about the future. In this respect, one concludes that considering time as an independent dimension, which allows traveling along the time line, is necessary. Keeping an active time domain in a spatial database, gives rise to the term spatio-temporal databases.

The ideal spatio-temporal database mentioned, besides having the normal functions essential to every spatial database, also has the ability for keeping track of changed data. In such databases, the mechanism of renewing the most recent spatial and temporal topologies is implemented. In addition to the process of updating geographical objects, keeping the valid topological (either temporal or spatial) relationships are also operational. [3]

Spatio-temporal objects are limited to a given temporal validity. When new information is assigned, the database has to be changed, in order to add the new data. In such a case the database may have to rebuild completely for storing the new state, which causes a great amount of redundancy. Alternatively the database might be time stamped in order to indicate the temporal validity of its objects.

The ability of versioning a database is essential for all spatio-temporal applications, where knowledge of the real world is uncertain or changing through time.[2]

## **5. Spatio-Temporal database features**

There is a critical set of capabilities that are needed by moving objects database applications and Spatio-temporal database . Existing DBMS's are not well equipped to handle continuously changing data and they should be developed to be able to handle these data. We chose the following features for the database, that is use to manage spatio-temporal data:

- Elementary features (basic data types, SQL language features, declarative integrity constraints, programming abstractions, automatic generation of identifiers, national characters support).
- Transactions and multi-user access (transactions, locks, multi-user access).
- Programming in database (stored procedures and triggers).
- Elements of database administration (access control, backup copies, data migration).
- Portability and scalability
- Performance and VLDB (Very Large Database) issues (query optimization, structures supporting query optimization, support for analytical processing, allocation of disk space, data size limits, known VLDB implementations)
- Distributed databases (access to multiple databases, heterogeneous systems support)
- Special data types (large objects in database, post-relational extensions, support for special data types)
- Application development and interfaces (embedded SQL, standard interfaces, additional interfaces, interoperability with Web technology, XML, CASE)
- Reliability (failure recovery)
- Commercial issues (technical support available, market position)
- Spatial vision

## **6. Database Software**

With consider to above mentioned specifications, and based on the capabilities that are needed by moving objects database applications, we choose some existance DBMS's and assessed them to find best DBMS to extend it for design and development a moving object database.DBMS such as Oracle that here we mean Oracle 9i and DB2 universal dadabase that is short form of database 2 universal database and is an IBM production. SQL server version 2000 from Microsoft. And at last MySQL and PostgreSQL as Open source databases.

## **7. The comparison of different database software**

In this section we seek to compare products of a similar nature. Unfortunately this is not a simple task. It is not true that SQL Server 2000 is better than MySQL version 4.1 or Oracle 9i Database is better than DB2 Universal Database v8.1 or vice versa. All products can be used to build stable and efficient systems and the stability and

effectiveness of the applications and databases depends upon the experience of the database developers and database administrator rather than the database's provider. However, each database has some advantages in comparison with others and vice versa. Here are some of the differences and similarities between different databases:

### 7.1. Performance comparison

It is very difficult to make a performance comparison between databases. The performance of the databases depends upon the experience of the database developers and database administrator rather than the database's provider.

### 7.2. Platform comparison

All of Oracle 9i Database, DB2 Universal Database version 8.1, MySQL version 4.1 and PostgreSQL support all known platforms, including Windows-based platforms, AIX-Based Systems, HP-UX systems, Linux Intel, Sun Solaris and so on. SQL Server 2000 only works on Windows-based platforms, including Windows 9x, Windows NT, Windows 2000 and Windows CE.

### 7.3. Hardware requirements

To install software under the Windows-based platforms, we should have the following hardware:

Hardware	DB2 Universal Database v8.1	MySQL 4.1	SQL Server 2000	Oracle 9i
<b>Processor</b>	Pentium or Pentium compatible CPU	Pentium 166 MHz	Pentium 166 MHz or higher	Pentium 166 MHz or higher
<b>Memory</b>	RAM: 256 MB minimum, additional memory may be required.	near 32 Mb RAM	32 MB RAM (minimum for Desktop Engine), 64 MB RAM (minimum for all other editions), 128 MB RAM or more recommended	RAM: 128 MB (256 MB recommended) Virtual Memory: Initial Size 200 MB, Maximum Size 400 MB
<b>Hard disk space</b>	Typical installation: 350 Mb minimum Compact installation: 100 Mb minimum Custom installation: 100 Mb minimum. Additional disk space may be required on FAT drives with large cluster size.	near 60 Mb	270 MB (full installation), 250 MB (typical), 95 MB (minimum), Desktop Engine: 44 MB Analysis Services: 50 MB minimum and 130 MB typical English Query: 80 MB	140 MB on the System Drive plus 4.5 GB for the Oracle Home Drive (FAT) or 2.8 GB for the Oracle Home Drive (NTFS)

Table 1. Hardware comparison

#### 7.4. Features comparison

All of Oracle 9i Database, IBM DB2 Universal Database v8.1, MySQL version 4.1 and SQL Server 2000 support the ANSI SQL-92 entry level and do not support the ANSI SQL-92 intermediate level.

In general, Oracle offers all the features required for information integration in a single unified product. The inherent reliability and security features of the Oracle database are automatically inherited by information integration applications. Oracle is clearly the leader in the distributed space with its single, unified solution to satisfy a complete spectrum of information integration requirements. All editions of Oracle include; the PL/SQL engine to develop stored procedures, triggers, and functions; a Java compiler and Virtual Machine (JVM) to develop Java stored procedures and triggers; XML support; an Apache Web server; and object-relational capabilities. Oracle9i Database offers a rich set of development and administration tools, and recent releases have focused on security and scalability improvements. The implementation of various index types demonstrates how extensively Oracle9i Database focuses on fast data retrieval, especially in large databases such as data warehouses. In addition, the use of reverse key and function-based indexes, allows for a great degree in flexibility in quickly accessing data. In recognizing the high availability challenges every business faces, Oracle provides comprehensive, unique, powerful and simple to use capabilities to protect against most forms of unplanned downtime, including system faults, data corruption, human error and disaster. And, achieves this in an environment where the need for planned downtime is marginalized.

IBM on the other hand has a completely different model for information integration. They offer different products for different integration scenarios. This adds several layers of complexity, since each product has to be installed and configured separately.

Oracle and DB2 can scale to terabytes of data storage fairly easily. MySQL and PostgreSQL are known to run well into the hundreds of gigabytes. PostgreSQL and MySQL boast widespread use for relatively small databases (under 100GB, for example).

DB2 offers the very basic set of backup and recovery capabilities but lacks the completeness and depth of data protection required by most businesses today. In fact, DB2 is behind even Microsoft SQL Server when it comes to high availability.

The dialect of SQL supported by Oracle 9i Database is called PL/SQL. The dialect of SQL supported by IBM DB2 v8.1 is called DB2 SQL dialect. The dialect of SQL supported by Microsoft SQL Server 2000 is called Transact-SQL (T-SQL). The dialect of SQL supported by MySQL version 4.1 is called MySQL dialect. Transact-SQL dialect is a more powerful language than MySQL dialect. A brief comparison of PL/SQL, DB2 SQL, T-SQL and MySQL dialect follows: (Table 2)

Feature	PL/SQL	DB2 SQL dialect	T-SQL	MySQL dialect	PostgreSQL
<b>Indexes</b>	B-Tree indexes, Bitmap indexes, Partitioned	B-Tree indexes, Bitmap indexes, Partitioned	B-Tree indexes	single column, multi-column, unique, and	single column, multi-column, unique, and

	indexes, Function-based indexes, Domain indexes	indexes, block indexes, dimension block indexes		primary key indexes, B-tree indexes	primary key indexes
<b>Tables</b>	Relational tables, Object tables, Temporary tables, Partitioned tables, External tables, Index organized tables	Relational tables, Object tables, Temporary tables	Relational tables, Temporary tables	Relational tables, Temporary tables	Relational tables, Object-relational
<b>Views</b>	Supported	Supported	Supported	Not Supported	Supported
<b>Data types</b>	Complex data type, character, number, date, large object (LOB), raw, row identifier, user defined data types	Complex data types, character, number, date, large object (LOB)		string, numeric, date and time	string, numeric, date and time, large objects, user defined data types
<b>Triggers</b>	BEFORE triggers, AFTER triggers, INSTEAD OF triggers, Database Event triggers	BEFORE triggers, AFTER triggers, INSTEAD OF triggers	AFTER triggers, INSTEAD OF triggers	Not Supported	Supported
<b>User-defined functions</b>	External Scalar functions, External table functions, OLE DB External table functions, Sourced or	External Scalar functions, External table functions, OLE DB External table	Scalar functions, Inline table-valued functions, Multistatement table-	Not Supported	Not Supported

	Template functions, SQL Scalar, Table or Row functions	functions, Sourced or Template functions, SQL Scalar, Table or Row functions	valued functions		
<b>Procedures</b>	PL/SQL statements, Java methods, third-generation language (3GL) routines	DB2 SQL dialect statements, Java methods, third-generation language (3GL) routines	T-SQL statements	Not Supported	PL/PGSQL, PL/TCL, PL/Perl, SQL, C
<b>Multiple Triggers</b>	Supported	Supported	Not Supported	Not Supported	Not Supported
<b>Arrays</b>	Supported	Supported	Not Supported	Supported	Supported

Table 2. Feature comparison

### 7.5. Spatial Extender

In 1997, the Open GIS Consortium published the OpenGIS Simple Features Specifications for SQL, a document that proposes several conceptual ways for extending an SQL RDBMS to support spatial data.

DB2 Spatial Extender is used to generate and analyze spatial information about geographic features and to store and manage the data on which this information is based. The DB2 Spatial Extender allows the user to include spatial attributes (distances, times, and geographical information) into business analysis. It conforms to ISO and OpenGIS Consortium (OGC) standards. Additionally the latest version of the spatial extenders includes an Index Advisor for tuning spatial indexes, includes additional spatial functions, and the ability to export SDE transfer data and spatial data to geobrowsers.

MySQL implements spatial extensions following the specification of the Open GIS Consortium (OGC).

A major feature of Oracle 9i is Oracle Spatial. This is a spatial extender that provides storage, indexing and proximity queries for location-based information, which may include road networks, wireless service boundaries and geo-coded customer addresses, for example. Oracle Spatial makes it possible to combine the relational power of a database with spatial data. The ability to use indexes, various queries, and functions means complex spatial calculations may be pushed back onto large database servers. As mobile applications and technologies increase, so will the demands to store and analyze spatial data in a transactional setting. [4]

## 7.6. Security

Database security is a very important aspect of any relational database management system to protect access to the database operations and the data.

At first glance, Oracle and IBM appear to offer similar security solutions, but with closer inspection, it is plain to see that the two companies approach security differently and ship solutions at vastly different levels of maturity. Independent evaluations and feature-for-feature comparisons prove that the Oracle9i Database is more secure than IBM's DB2 Universal Database. Overwhelming evidence supporting this assertion, as established in this paper, proves that Oracle security is far superior to DB2 security.

It is difficult to make up for a lack of security built into the core DB2 product set, but IBM offers a variety of packaged service plans to do so. Oracle's security solutions are much less expensive than IBM's because customers do not have to pay for additional software and services. IBM's security solutions are less secure than Oracle's because they rely on external solution and services to implement security they've neglected to build into DB2, which does not provide equivalently robust, mature security features that Oracle has been shipping for years.

## 8. Summery

In summery, different DBMS package has different features and capabilities. The following tabale provide a brief review on the comparison different DBMS based on their featur and capabilities. (Table 3)

<b>Features</b>	<b>Oracle</b>	<b>DB2</b>	<b>MS SQL server</b>	<b>MySQL</b>	<b>PostgreSQL</b>
<b>Data types</b>	Very good	Good	Good	Good	Very good
<b>SQL language features</b>	Very good	Very good	Very good	Average	Good
<b>Declarative integrity constraints</b>	Very good	Very good	Very good	Average	Very good
<b>Primary key</b>	Yes	Yes	Yes	Yes	Yes
<b>Foreign key</b>	Yes	Yes	Yes	No	Yes
<b>Programming abstractions</b>	Very good	Good	Good	Poor	Average
<b>National characters support</b>	Very good	Very good	Good	Good	Good
<b>Transactions</b>	Very good	Very good	Very good	Poor	Good
<b>Locks</b>	Very good	Good	Good	Poor	Good
<b>Multiuser access</b>	Very good	Good	Average	Average	Average
<b>Stored procedures and triggers</b>	Very good	Very good	Good	Poor	Very good
<b>Access control</b>	Very good	Very good	Good	Very good	Good
<b>Backup copies</b>	Very good	Very good	Good	Average	Average
<b>Portability of</b>	Very good	Good	Poor	Good	Good

<b>DBMS</b>					
<b>Scalability</b>	Very good	Very Good	Good	Good	Average
<b>Query optimization</b>	Very good	Very good	Good	Good	Good
<b>Structures supporting query optimization</b>	Very good	Good	Good	Poor	Good
<b>Support for analytical processing</b>	Very good	Very good	Good	Poor	Poor
<b>Allocation of the disk space</b>	Very good	Good	Good	Average	Average
<b>Data size limits</b>	Very good	Very good	Very good	Good	Average
<b>VLDB (very large data base) implementations</b>	Very good	Very good	Good	Poor	Good
<b>Access to multiple databases</b>	Very good	Very good	poor	Average	Average
<b>Heterogeneous systems support</b>	Good	Good	Poor	Poor	Poor
<b>Large objects in database</b>	Very good	Very Good	Good	Good	Average
<b>User-defined data types</b>	Yes	Yes	No	No	Yes
<b>Object-relational extensions</b>	Yes	Yes	No	No	Yes
<b>Support for special data types</b>	Very good	Very good	Good	Poor	Average
<b>Complex Applications</b>	Very good	Very good	Average	Poor	Good
<b>Embedded SQL</b>	Very good	Very good	Good	Poor	Good
<b>Standard interfaces (ODBC, JDBC)</b>	Very good	Very good	Good	Good	Good
<b>Interoperability with Web technology</b>	Very good	Very good	Good	Good	Good
<b>XML</b>	Very good	Very good	Very good	Poor	Poor
<b>Recovery from failures</b>	Very good	Very good	Good	Average	Average
<b>Security</b>	Very good	Good	Good	Average	Average
<b>Ease of Use</b>	Average	Poor	Very good	Good	Good
<b>Mobile-commerce</b>	Very good	Very good	Good	Average	Poor
<b>Spatial extender</b>	Very good	Very good	Good	Average	Poor
<b>Prices</b>	High	Average	Low	Free	Free

<b>Technical support from manufacturer (Availability)</b>	Very Good	Good	Good	Average	Poor
<b>Position on the market</b>	Very good	Very good	Good	Poor	Poor

Table 3. Comparison of DBMS's at a glance

## 9. Conclusion

From the above presented comparison, the following conclusions can be made:

- SQL Server 2000 holds the top TPC-C performance and price/performance results.
- SQL Server 2000 is generally accepted as easier to install, use and manage.
- MySQL version 4.1 requires little hardware resources.
- You can use MySQL version 4.1 without any payment under the terms of the GNU General Public License.
- Oracle 9i Database, DB2 Universal Database v8.1 and MySQL version 4.1 supports all known platforms, not only the Windows-based platforms.
- PL/SQL is the most powerful language
- Oracle9i Database offers a rich set of development and administration tools
- Oracle and DB2 can scale to terabytes of data storage fairly easily
- When it comes to backups, open-source databases like MySQL and PostgreSQL may not completely fulfill your needs.
- Oracle Spatial makes it possible to combine the relational power of a database with spatial data

With consider to specifications of different commercial database management systems, that mentioned before and based on the capabilities that are needed by moving objects database applications, we recommend Oracle as the first choice for managing spatio-temporal data. DB2 universal database and SQL server are the next alternatives.

## 10. Future Activities

Our objective is to develop a framework for managing and querying moving object database. We will use Oracle software package to design and develop the capabilities required for moving object's queries.

## 11. References

- [1]. G.Christakos, P.Bogaert, M.Serre, , Temporal GIS: Advanced functions for field-based applications, Springer.2002.
- [2]. Time-integrative Geographic information system: Management and analysis of spatio-temporal data. Thomas Ott. Frank Swisaczny.2001.Springer

- [3]. Roshannezhad, Aliasghar, ”,(1996) The Management Of Spatio-Temporal Data in a National Geographic Information System”.
- [4]. Matt Bauer, Mapping Geometric Data with Oracle Spatial, 11/10/2003
- [5]. Xavier R. Lopez, Oracle Database 10g:A SPATIAL VLDB CASE STUDY, Oracle Corporation
- [6]. Alexander Chigrik. A Comparison of Oracle 9i with DB2 v8.1, September 10, 2003
- [7]. [www.databasejournal.com](http://www.databasejournal.com)
- [8]. Michael Otey . SQL Server 2000's Coolest Features, InstantDoc #9797, November 2000
- [9]. Alexander Chigrik. The comparison of SQL Server 2000 with MySQL v4.1, October 8, 2003
- [10]. Suita Gupta, Allan Tham, Raul Chong. Leveraging your Oracle 9i skills to learn DB2 UDB for Linux, UNIX and Windows V8, 09 Jan 2004
- [11]. Gregg Petri .A Comparison of Oracle and MySQL, 1st Qtr 2005
- [12]. Tim Conrad .PostgreSQL vs. MySQL vs. Commercial Databases: It's All About What You Need,2005
- [13]. Konrad Bohuszewicz, Maciej Czyżowicz, Micha Janik, Dawid Jarosz, Piotr Mazan, Marcin Mierzejewski, Miko aj Olszewski, Wiktor S. Peryt, Sylwester Radomski, Piotr Szarwas, Tomasz Traczyk, Dominik Tukendorf, Jacek Wojcieszuk .Comparison of Oracle, MySQL and Postgres DBMS.
- [14]. Technical Comparison of Oracle9i Database vs. IBM DB2 UDB: Focus on High Availability, An Oracle White Paper, February 2002
- [15]. Bloor Research .Databases An evaluation & comparison.