Ross School of Business at the University of Michigan

<u>Independent Study Project Report</u>

TERM : **Spring 1996**

COURSE : **CIS 750**

PROFESSOR : **David Blair**

STUDENT : **Suresh Jayaraman**

TITLE : **A Comparative Analysis of Leading Relational Database Management Systems**

An Comparative Analysis of Leading Relational Database Management Systems

by

Suresh Jayaraman

A research paper submitted in fulfillment of the requirements for three credits,
GRADUATE INDEPENDENT RESEARCH PROJECT Spring Term 1996
Professor David Blair, Faculty Supervisor.

NOV 2 2 1996

Faculty Comments

This is an excellent report comparing the three major relational data base management systems: Sybase, Informix & Oracle. Because the industry is undergoing constant change and development it is very difficult to bring this kind of data together. The author is commended for doing such a thorough job.

D Blair

Signature of Faculty Supervisor

Assoc. Prof. CIS

Title

Table of Contents

1. EXECUTIVE SUMMARY....4

2. THE RELATIONAL DATABASE MANAGEMENT MARKET....5

2.1. MARKET DRIVERS....5
2.2. MARKET SIZE AND GROWTH....5

3. WHAT RDBMS FEATURES MATTER ?....7

4. RDBMS PERFORMANCE FEATURES....8

4.1. GENERAL PERFORMANCE FEATURES....8
4.2. ARCHITECTURE, SCALABILITY, AND SPEED....8
4.2.1. Hardware Parallelism....9
4.2.2. Software Parallelism....10
4.2.2.1. Inter-Query Parallelism....10
4.2.2.2. Intra-Query Parallelism....11
4.2.3. *Database Architectures for Parallel Processing*....12
4.3. OPTIMIZER CAPABILITIES....14
4.4. SUPPORT FOR COMPLEX DATA TYPES....14

5. INTEGRITY FEATURES....16

6. DATABASE ADMINISTRATION FEATURES....16

7. DATABASE CONNECTIVITY AND INTER-OPERABILITY....17

8. DISTRIBUTED DATABASE SUPPORT....18

9. SECURITY FEATURES....19

10. EVALUATING A DBMS AND ITS ENGINE....20

10.1. GENERAL EVALUATION ISSUES....20
10.2. EVALUATION CRITERIA....21

11. CONCLUSIONS....23

12. CONTACT INFORMATION....25

13. WORLD WIDE WEB PAGES....25

14. REFERENCES....26

Executive Summary

Information Systems (IS) organizations are constantly faced with implementation decisions for large, mission-critical systems involving On Line Transaction Processing (OLTP) and decision support . Because of their flexibility and productivity and the availability of robust application development tools, relational database management systems (RDBMSs) have established themselves as a durable technological option. The RDBMS vendors have invested in techniques such as parallelism to give their products scalability and availability features that make them worthy alternatives to legacy database management systems that reside on mainframes [1].

Software vendors such as SAP AG and PeopleSoft have responded by designing generic client/server applications for RDBMSs for business functions such as finance and human resources. As a result, RDBMSs have become an almost unavoidable choice for IS applications.

Although there is a variation by vendor, the leading RDBMSs on mainframe and UNIX systems are now well proven for small and medium scale OLTP systems. But organizations building large enterprise systems today are faced with a dilemma [1]. On one hand, they can build the new systems in "tried and tested" legacy DBMS technology, getting the benefit of reduced implementation risk in the short term, but potentially losing out in the long term by increasing the portfolio of inflexible legacy applications. On the other hand, they can "take the plunge" and go with the newer RDBMS technology, sometimes hoping that future product improvements will get them out of potential trouble. Most organizations would like to go with the new technology, but recognize that excessive risk is potentially injurious to the organization's and the IS executive's health. As a result, IS executives are wrestling with questions such as: Can we base applications of this size on an RDBMS ? Which RDBMS ? What are the limitations ? Who is already doing this ? How do the products compare ?

This study attempts to answer these questions by examining material from IS magazines, vendor literature, vendors' World Wide Web Sites, and industry analysts. To compare the product features, the three most popular UNIX RDBMSs Informix, Oracle, and Sybase are considered. The features that are important to OLTP and decision support applications are listed and described. Feature comparison matrices are shown under different functional categories. Since product evaluation and selection will be crucial to an IS manager, general and specific evaluation criteria that must be considered in the selection process are listed. Finally, how the RDBMS market may evolve, and how the current players will prosper in the next five years are examined. This report should help an IS manager, evaluate RDBMS vendors and their product offerings during the selection process of an RDBMS.

2. The Relational Database Management Market

Relational database software and the associated connectivity software, gateways, and development tools for the UNIX and other server operating environment such as NetWare, Windows NT, and OS/2 are the rapidly growing segments of the software market. Oracle, Sybase, and Informix continue to be the market leaders, estimated to hold a combined share of more than 70% in 1995 [13].

2.1. Market Drivers

A recent IDC report [2] lists the following key drivers behind the market trends:

- **Client/Server Trends**: The adoption of client/server application architectures continues to drive the sales of the software infrastructure for distributed applications. Database engines are a vital part of distributed applications.

- **Staffing Costs**: Companies are consolidating from single-function servers to multifunction servers to reduce overall staffing costs. The operating systems that support multifunction servers, UNIX and Windows NT, are the platforms often selected when companies undertake this consolidation.

- **Resource Costs**: Companies are often seeking ways to do more with fewer resources. RDBMSs are often considered as a way to allow developers to concentrate on business rules processing and not on how or where data is stored. The application tools market has fueled this trend by allowing developers to concentrate more on what needs to be accomplished and less on how the system actually accomplishes it.

2.2. Market Size and Growth

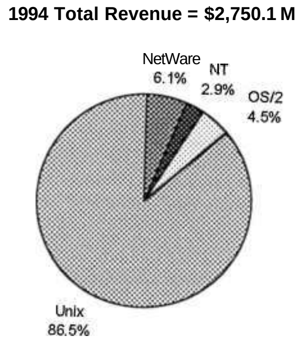
The UNIX and other non-UNIX open systems market for RDBMS appears to have a strong growth from 1994 to 1995. Figure 1 shows this trend. UNIX increased its grip on the position as the number one platform for RDBMS in this market. by increasing its share by 0.7% to 87.2% share.

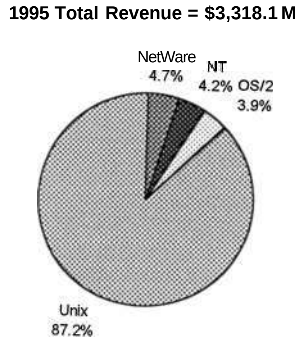
Netware dropped 1.4% to hold 4.7% share of the combined market while maintaining its number two position. Windows NT has shown the strongest growth in this market. OS/2 continues to slide as a platform for RDBMS software.

The combined worldwide UNIX RDBMS and tools software market is now estimated at $2.89 billion. This includes licenses and maintenance on database engines and development tools as shown in Table 1. Oracle Corporation dominates the UNIX RDBMS market, with a 41.3% share. Informix Software (16.1%) has moved up on Sybase (17.9%) but is still in the number three spot.

Figure 1

UNIX and Open Systems RDBMS and Tools Revenues by Platform





Source: International Data Corporation, 1995

IDC [1] (International Data Corp.) believes this can be attributed to a growing recognition of Informix Software's strong Dynamic Scalable Architecture (DSA) technology. Sybase appears to have been hurt badly by increasing concerns over the scalability of their flagship product SQL Server 10.

Table 1

Worldwide UNIX RDBMS and Tools License Revenue, 1995 (Estimate)

Company	Revenue($M)	Share(%)
Oracle Systems Corp.	1,194.30	41.3
Sybase Inc.	517.6	17.9
Informix Software	464.1	16.1
Computer Associates Int'l. Inc. (Ingres)	235.8	8.2
Progress Software	70.3	2.4
IBM	50	1.7
VMARK Software	29.5	1.0
InterSystems	29.4	1.0
Hewlet-Packard Company	24.9	0.9
Other	275.3	9.5
Total	2,891.20	100.0

Source: International Data Corporation, 1995

Most database vendors supply database engines and application development tools. Company reports often do not show their revenues from database server alone for fear of competitive reactions. A case in point is Sybase Inc. which acquired PowerSoft, the maker of PowerBuilder, a wildly popular application development tool, in 1994. The following year, 1995, however, turned out to be a bad year for their database server sales. Sybase does not publish their database server revenues separately since this information is likely to impact their position in the server and tools market.

A smaller market exists for RDBMS in non-UNIX open systems. These include NetWare, Windows NT, and OS/2 operating environment. IDC estimates revenues from these markets to be $157 million, $141 million and $124 million respectively. Among the three, the Windows NT RDBMS sales grew dramatically, capturing the first place in the non-UNIX opens systems environment. When Windows NT RDBMS market is compared with the UNIX RDBMS market, however, it appears small and volatile. Table 2 shows the worldwide Windows NT RDBMS and Tools license revenues.

Table 2

Worldwide Windows NT RDBMS and Tools License Revenue, 1995 (Estimate)

Company	Revenue($M)	Share(%)
Microsoft Corporation	56.0	39.8
Computer Associates Int'l. Inc. (Ingres)	17.5	12.4
Sybase Inc.	17.2	12.2
Oracle Systems Corporation	16.1	11.4
Informix Software	10.0	7.1
Progress Software	1.7	1.2
IBM	1.6	1.1
InterSystems	0.2	0.1
Other	20.5	14.6
Total	140.8	100.0

Source: International Data Corporation, 1995

3. What RDBMS Features Matter ?

Most RDBMS vendors offer the core features necessary for supporting today's business applications. The important question to ask is "How will a vendor enhance and support the RDBMS to keep pace with rapidly changing hardware, software, and networking technologies ?." Vendors tend to push many features that are not needed or used by the development organization.[3] Vendors are also continuously leapfrogging one another with respect to functionality. The following features are considered essential to any serious RDBMS vendor offering [3, 8]:

1. Performance features
2. Integrity features
3. Database administration features
4. Database connectivity and inter-operability
5. Distributed database support
6. Security features

4. RDBMS Performance Features

Under this category we examine the following sub-categories:

1. General Performance Features
2. Architecture, Scalability and speed
3. Optimizer capability
4. Support for complex data types

4.1. General Performance Features

A high degree of parity has been achieved across vendor offerings regarding support for performance features. Features such as locking granularity, isolation levels, and stored procedures are included in the comparison shown in Table 4.

4.2. Architecture, Scalability, and Speed [7, 16, 19]

As organizations collect more and more business data, they must deal with larger and larger database environments. Every major increment presents new challenges and users are constantly searching for better database performance in both speed and scalability. *Speed-up* means that the same request takes less time on the same amount of data. *Scale-up* means that the user gets comparable performance on a request as the size of the database increases. The overall goal is to increase speed and scalability in a linear fashion, i.e., doubling the number of processors cuts the response time in half or provides the same performance across twice as much data.

Imagine you work for a marketing research organization. You are tasked with searching for unusual buying patterns among customers. To do this, you will need details about the millions of customers and about their purchases for five years. Such Decision Support Systems (DSS) are not uncommon in today's business world. The total size of the database could be over 1 terabyte (TB), or 1,000 gigabytes (Gbs). The plain vanilla RDBMS, at a disk scan speed of 10 MB per second, will take more than *26 hours* to sequentially read *once* through this volume of data.

Such demands are also possible for on-line transaction processing (OLTP) systems. Video-on-demand, where a million subscribers may all want a movie delivered at the same time on a Friday night, is a good example.

As databases sizes grow it becomes essential that an RDBMS be able to 1) break up the management problem into smaller pieces; 2) speed up data-intensive operations, such as data loading, backup and recovery, by means of parallelism or reduction of RDBMS internal overhead. The quest for better performance and

scalability has given rise to *parallel processing.* Parallel processing can be examined from a hardware and software perspective.

4.2.1. Hardware Parallelism

Software parallelism is a natural follow-on to hardware parallelism. The computer hardware industry trends have fueled the evolution of database server technology. In the eighties, hardware vendors were eager to sell their next "bigger and better" model of computer when the installed platform could not keep up with the increasing demand from software applications and user base. The hardware vendors addressed this concern by giving trade-ins. As processor prices tumbled, the value of a trade-in became insignificant. The customer did not want to forego the initial investment to trade up to a new model. The key driver was cost. In response, the hardware vendors began to offer multiprocessor platforms in the early nineties. These platforms allowed the customer to expand the capability of the computer by simply adding more processors and memory. This was the beginning of hardware parallelism.

Examples of hardware parallelism are multiple processors (CPUs) within a single computer. An example of this is the Sparc Center 2000 from SUN offering up to 20 processors. The parallel architectures enable many smaller components to work together instead of depending on one big component - computer, CPU, disk - that can quickly become a bottleneck. Spreading the processing across multiple components improves both performance and availability. Three major multiprocessor systems architectures have emerged. Figure 3 shows the components of the three architectures.

SMP (Symmetric Multi Processing) Architecture

Every component of an SMP system is controlled by a single executing copy of an operating system (OS) managing a shared global memory. Because memory in an SMP system is shared among the CPUs, SMP systems have a single address space and run a single copy of the OS and the application. All processes are fully symmetric in the sense that any process can execute on any processor at any time. As system loads and configurations change, tasks or processes are automatically distributed among the CPUs -- providing a benefit known as dynamic load balancing.

Clustered SMP Architecture

As speedup and scaleup demands of data-intensive applications have outstripped a single SMP system's capabilities, a logical, evolutionary approach has been taken by SMP and RDBMS vendors. Scaleable hardware and software technologies are allowing shared-everything systems to be clustered (connected) together for greater scaleup, speedup and availability. Such systems share disk storage, data bus and multiple high-bandwidth I/O channels for high throughput levels. '

MPP (Massively Parallel Processing) Architecture

MPP, or shared-nothing, systems are composed of many loosely coupled processor/memory modules (P/M units or nodes) connected to one another by a high-speed communications mechanism optimized for fast message passing. Each node of a shared-nothing system is composed of its own P/M unit, which includes a CPU, associated memory to run programs, disks, one copy of the OS, and one copy of the RDBMS Independent instances of system and application programs use the common high speed interconnect to pass messages between cooperating nodes. Application connectivity in MPP systems is provided through message passing (as opposed to shared memory and pointers in SMP systems).

4.2.2. Software Parallelism

The main task of a database engine is to process the raw SQL (Structured Query Language) requests from application software. Database vendors began to take advantage of parallel hardware environments with the implementation of multiserver, multithreaded architectures to efficiently handle large numbers of client requests.

4.2.2.1. Inter-Query Parallelism

All serious RDBMSs already support one form of parallelism, called inter-query parallelism, where different server processes or threads handle multiple requests at the same time. This has been implemented in response to the growing popularity of SMP systems where multiple (parallel) processors share both memory and disk.

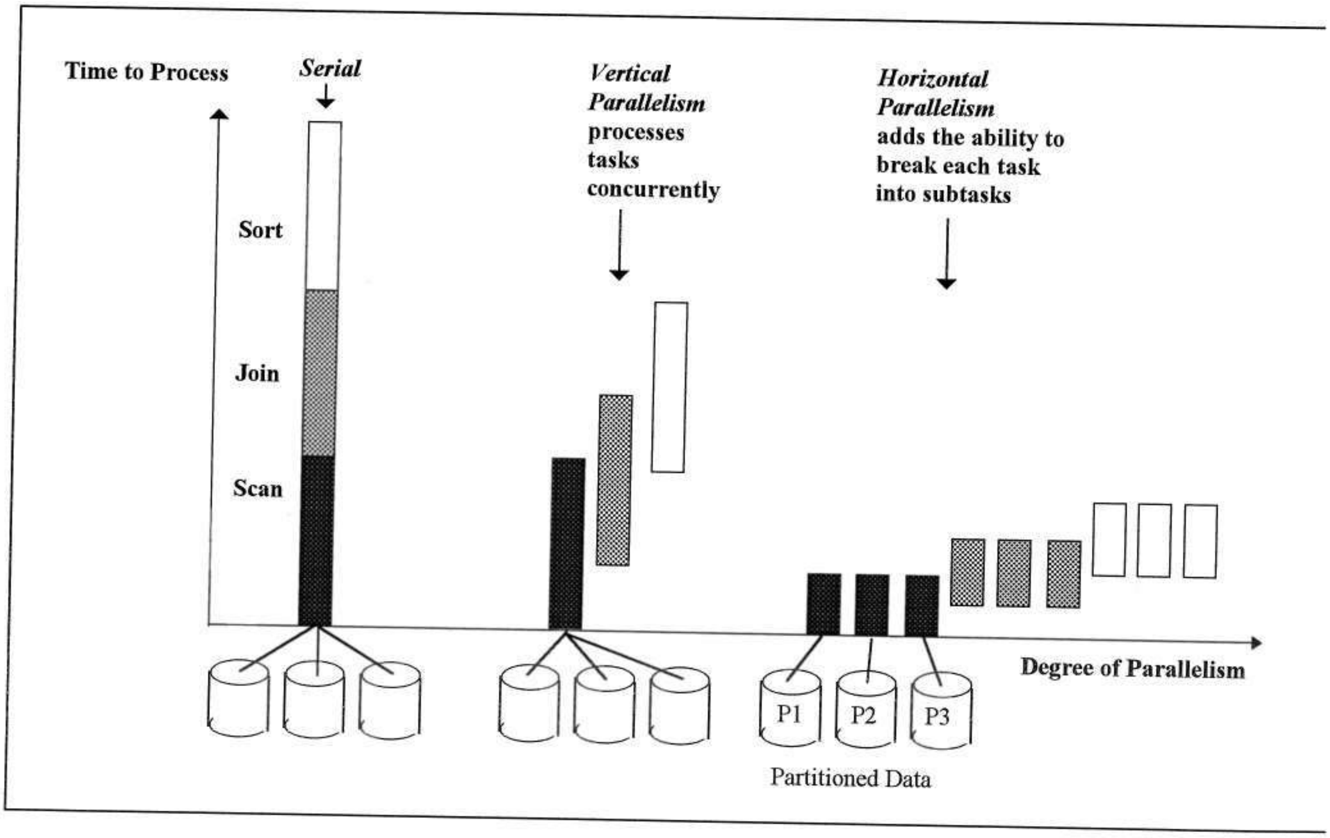
Inter-query parallelism increases throughput so that the DBMS can support more concurrent users. Multiple queries are processed simultaneously, but each request is still processed serially. Each task in the request is executed sequentiaUy and by a single process or thread. For example, to execute a request, first tables or indexes are scanned, then rows of two tables are joined then the resulting rows are sorted, and so on. Each operation must finish before the next one begins. This is illustrated in Figure 2. In a serial environment, many complex, long-running queries can eat up resources and result in poor performance for all users.

4.2.2.2. Intra-Query Parallelism

The next form of parallelism breaks a SQL statement into multiple tasks that are executed in parallel across multiple processors. In other words, *intra-query parallelism* processes multiple tasks in a single SQL statement simultaneously. Because the DBMS can devote additional resources to an individual user request, it takes less time to complete the request. ParaUel processing of tasks within a SQL statement can be done two ways - horizontally and vertically.

Figure 2

Horizontal and Vertical Parallelism [19]



Vertical Intra-Query Parallelism

The first method is called vertical parallelism, or pipelined parallelism. This is parallel processing among different tasks, for example, concurrent scanning, joining, and sorting. Here the query is divided into its different tasks (scan, join, sort), and the output from one task serves as the input to another task, all of which are executing in parallel. (See Figure 2.)

Horizontal Intra-Query Parallelism

In this method, known as horizontal parallelism, or partition parallelism, parallel processing occurs *within* specific task - for example, multiple concurrent table-scan operations. Each scan operation executes in parallel against a different set of data, or partition, stored on a different disk. Data partitioning is required for horizontal parallelism. (See Figure 2)

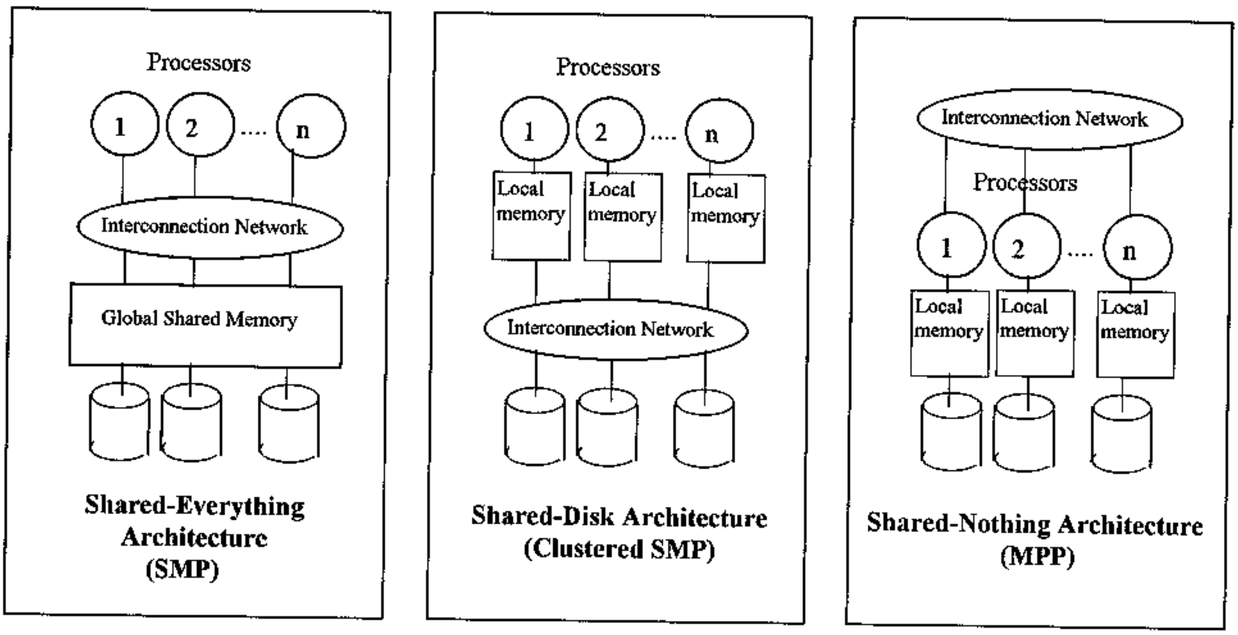
A state-of-the-art parallel database product should enable both types of parallel execution.

4.2.3. Database Architectures for Parallel Processing [16]

As the RDBMS vendors introduced their parallel software architectures, they borrowed the terms *shared everything, shared disk,* and *shared nothing,* originally used to describe parallel *hardware configurations.* The database architecture determines the ultimate scalability of the solution.

Figure 3

Database Architectures for Parallel Processing



Shared-Everything Architecture

In a shared-memory parallel database architecture, the DBMS assumes that multiple database components working on a single SQL statement use shared memory for communicating messages and data. All processors have access to all data, which are partitioned across local disks. This type of architecture is optimized for an SMP platform - a shared-everything, tightly coupled hardware platform where all CPUs share memory and disk storage. Informix Online 7.2, Oracle 7.1 and Sybase System 10 can take advantage of this architecture.

Shared-Disk Architecture

In a shared-disk system, the DBMS assumes that all processors have direct access to all disks (all data), and they do not share memory. The key point about this architecture is that ownership of the entire database is shared between RDBMS servers running on each node. The multiple RDBMS servers must cooperate to maintain locking consistency across multiple nodes. Oracle's parallel query option (PQO) takes advantage of this architecture.

Table 3

Relative Benefits of Parallel Architectures

Architecture	Advantages	Drawbacks
Shared-Everything (SMP) Examples *Informix OnLine 7.2* *Oracle 7.1* *Sybase System 10*	• Easy to develop since everything is local and shared • Using shared-memory for interprocess communication is usually faster than exchanging messages over an interconnect	• There are limits to scalability of shared memory
Shared-disk (Clustered SMP) Example *Oracle with Parallel Query Option*	• Eliminates memory access bottleneck of large shared-memory systems	• Don't scale well for OLTP applications because of the overhead of sharing information across nodes at a low level
Shared-nothing (MPP) Examples *Informix OnLine 8.0 XPS* IBM DB2 PE *Sybase Navigation Server*	• Near linear (or better) scalability across a large number of processors An effective way to apply the power of hundreds or thousands of processors to a single user for terabytes of data	• Significant shipping of data among nodes can be expensive

Shared-Nothing Architecture

In a shared-nothing environment, the database parallelizes a SQL statement across multiple nodes in the network. Each node has its own memory and disk storage and communicates with other nodes. The database is partitioned and servers on different nodes have exclusive ownership of those partitions. A shared-nothing architecture is optimized specifically for MPP and clustered platforms. The Sybase Navigation Server, IBM's DB2 PE, and Informix OnLine 8.0 take advantage of this architecture.

All three architectures are feasible mechanisms for harnessing the processing power of parallel hardware architectures. Their relative advantages and drawbacks are shown in Table 3.

4.3. Optimizer Capabilities

As databases grow and queries get complex, the ability of the database optimizer to understand and utilize metrics regarding the operating environment become significant. When optimizing a query, the database server takes into account many variables, such as the number and type of indexes, minimum and maximum index values, the number of records to be retrieved! All these factors are used to determine the cost of accessing the data requested by the SQL statement. The access method with the lowest cost is usually chosen by the optimizer. All vendors provide this feature.

4.4. Support for Complex Data Types

There is a growing demand for complex datatypes such as multimedia data (text, image, audio, video) and multidimensional data. The increasing popularity of the World-Wide-Web has created this demand. Content Management is the terminology used by leading RDBMS vendors to describe these needs. Oracle and Informix have announced their respective Universal Server products to address these needs. Sybase is lagging in this area, although simple user defined data types are supported today. Refer to Table 4 for a comparison of features.

Table 4

Performance Features of Leading RDBMS Products

Performance Feature	Informix 7.2	Oracle 7.1	Sybase System 10
General features			
Row-level locking	Yes	Yes	No
Page-level locking	Yes	Yes	Yes
Table-level locking	Yes	Yes	Yes
Database-level locking	Yes	Yes	Yes
Stored procedures in database	Yes	Yes	Yes
Triggers	Yes	Yes (Requires Procedural Option)	Yes
Architecture, Scalability & Speed			
Multithreaded architecture	Yes	No (Multi-process)	Yes
Core Parallelism	Yes	No (Requires PQO)	No (Requires Navigation Server)
Support for SMP	Yes	Yes	Yes
Support for loosely coupled systems	No (Requires OnLine 8.0)	Yes	No (Requires Navigation Server)
Optimizer capability			
Cost Based Optimizer	Yes	Yes	Yes
Non-Standard Data Types			
Binary Large Object (BLOB)	Yes	Yes	Yes
User Defined Datatype (UDT)	No (Supported in Universal Server)	No	Yes
Multimedia (Content)	No (Requires Universal Server)	No (Requires Universal Server)	No

5. Integrity Features

Data integrity is at the heart of every information system. The vendors have recognized this and support the declarative integrity model where the integrity constraints are declared at the time of creation of the tables. There is little differentiation across features supported for database integrity. Some vendors such as Oracle rely on database triggers to implement data replication, a feature to propagate data from one server to another. This form of data replication requires the database administrator to set up the order of firing of triggers when multiple triggers are declared per table.

Database event alerter is a feature first introduced by Ingres. This powerful feature allows the creation of user defined events as database objects. Once created by the DBA, the event can be registered by any application connected to the database. Any connected user or application can raise an event that will cause the database server to notify all users who have registered to receive notification. Event alerters offer messaging capabilities to on-line sessions. Integrity features of the leading RDBMS products are shown in Table 5

Table 5

RDBMS Integrity Features

Integrity Feature	Informix 7.2	Oracle 7.1	Sybase System 10
Declarative Referential Integrity	Yes	Yes	Yes
Cascading updates	Yes	No	No
Cascading deletes	Yes	Yes	No
Database Triggers	Yes	Yes	Yes
Programmer can control firing order of triggers	No	Yes	Yes
Event Alerters	No (Requires OnLine 8.0)	Yes	Yes

6. Database Administration Features [8,22]

Database administration is a key consideration for any information system whose operation relies on a database engine. Database administration includes database maintenance, DBA utilities for performance monitoring and tuning, backup/recovery, and data reorg utilities. Database administration has recently become easier and more efficient. Every major RDBMS vendor provides utilities for routine maintenance, backup and recovery. Parallel backup/recovery has become important with the advent of parallel hardware and database architecture. The three leading vendors Informix, Oracle, and Sybase offer this capability.

Companies with aggressive client/server implementation plans will require the capability to remotely administer dispersed database servers. All three leading RDBMS vendors allow users to manage remote databases from a single central console. Table 6 shows the database administration features of the three leading RDBMS vendors.

Table 6

RDBMS Administration Features of Leading Vendors (Adapted from [8])

Feature	Informix 7.2	Oracle 7.1	Sybase System 10
Portability	Yes	Yes	Yes
Automatic Database Recovery	Yes	Yes	Yes
On-line Database Backup	Yes	Yes	Yes
Database Mirroring	Yes	No	Yes
Query Resource Manager	Yes	Yes	No
Raw Disk Access	Yes	Yes	Yes
DBA Utilities	Yes	Yes	Yes
Parallel Backup	Yes	Yes	Yes
Parallel Recovery	Yes	Yes	Yes
Hot Standby Server	Yes	Yes	Yes
Database Auditing	Yes	Yes	Yes
Remote Administration	Yes	Yes	Yes

7. Database Connectivity and Inter-Operability

The rise of client/server computing has demanded the need for database connectivity and inter-operability. Connectivity refers to the ability to connect from an application residing on one machine to a database server (or servers) residing on a second machine (or several machines). Every RDBMS vendor offers the capability to connect an application to the vendor database. Most vendors also provide the capability to connect to a database from another vendor (heterogeneous connectivity). This is generally provided by a vendor-supplied database gateway. The capability to connect to a DB2 database residing on a mainframe from an Oracle application using the IBM connectivity standard, DRDA, is an example.

Inter-operability refers to the capability to change the database server (the backend) without extensive modifications to the application (front-end). This capability can be provided in two ways. The first method is by enforcing the use of ANSI compliant SQL in all applications. ANSI SQL compliance refers to whether the database server adheres to standards set forth by the American National Standards Institute (ANSI) for SQL language. Compliance with ANSI SQL will prevent the use of vendor enhancements to the SQL syntax. This does not prevent the user from using vendor specific application development functions.

The second method for inter-operability is through the use of ODBC (Open Database Connectivity), a specification developed by Microsoft and SQL Access Group. It is designed to eliminate the need for proprietary gateways for heterogeneous client/server database connectivity. An ODBC solution consists of two parts; the application tool, and the interface to the database server. The application tool must be ODBC enabled, and the interface to the database server must be able to accept ODBC calls. All major RDBMS have the capability to accept ODBC calls. Table 7 shows the database connectivity and inter-operability features of the three leading RDBMS vendors.

Table 7

Connectivity and Inter-Operability Features of Leading Vendors (Adapted from [8])

Feature	Informix 7.2	Oracle 7.1	Sybase System 10
Client/Server Connectivity	Yes	Yes (Requires the purchase of SQL Net)	Yes
DB2/MVS connection through DRDA	Yes	Yes	No (Proprietary)
ODBC	Yes	Yes	Yes
ANSI SQL Compliance	SQL 92 Entry Level	SQL 92 Entry Level	SQL 92 Entry Level

8. Distributed Database Support[8]

Organizations are increasingly becoming decentralized and are allowing IS functions across geographically distributed data centers. The two-phase commit protocol offered by Oracle and Informix coordinates work performed at multiple database servers on behalf of a single transaction. The X/Open XA compliant transaction managers (Tuxedo, TopEnd, Encina etc.) coordinate two-phase commits between heterogeneous, distributed databases. This allows a single transaction to span multiple XA-compliant databases.

Distributed DBMS is now coming of age because of the more common availability of data replication capabilities from leading vendors and the recent availability of more highly scalable RDBMS engines. Data replication is perceived as a key enabling technology that has helped to rationalize distributed DBMS because of its support for increased flexibility in RDBMS configurations.

Data replication is now available from virtually every leading vendor. Sybase, the pioneering vendor in data replication, has established a considerable following for its transaction (log) based replication architecture. Informix has also chosen this replication architecture. The differences lie in the level of skill required to set up and monitor the replication. The Sybase approach requires skilled administrators and developers. The Informix approach is less complex to use and administer. The Informix Online 7.1 server offers built-in replication features as part of their core product. No additional products need to be purchased for High-availability Data Replication (HDR) which provides a powerful hot stand-by function to a remote database server. Oracle and Sybase approaches require the purchase of additional products. Oracle has followed the Ingres approach of trigger-based replication architecture. These vendors tout the availability of update-anywhere and conflict resolution as unique features of their replication architecture. Table 6 shows the distributed database features of the three leading RDBMS vendors.

Table 8

Distributed Database Features

Feature	Informix 7.2	Oracle 7.1 (Requires Distributed Option)	Sybase System 10
Distributed Database	Yes	Yes	Yes (Requires OmniSQL product)
Two-Phase Commit	Yes	Yes	No
Distributed Query Optimizer	Yes	Yes	Yes (Requires OmniSQL product)
X/Open XA	Yes	Yes	Yes
Data Replication	Yes	Yes	Yes (Requires Replication Server)
Update-anywhere replication	No	Yes	No

9, Security Features

Most vendors now provide a "secure" version of their RDBMS server. The US government's National Computer Security Center (NCSC) has classified information security into seven classes of evaluation ratings, ranging from Al (most secure) through B3, B2, Bl, and CI to D (minimal security). A product must meet the functional requirements of the target evaluation class, and the implementation of security features to earn the appropriate rating. Table 9 provides a view of common security features.

Table 9

RDBMS Security Features [8]

Feature	Informix 7.2	Oracle 7.1 (Requires Distributed Option)	Sybase System 10
Operating System security integration	Yes	Yes	Yes
User group privileges/roles	Yes	Yes	Yes
NCSC Evaluation Ongoing	Yes	Yes	No
C2 Level Certification	Yes	Yes	Yes
B1 Level Certification	Yes	Yes	Yes

10. Evaluating a DBMS and its Engine [9]

The previous section described the features considered important from a technical perspective. An IS manager should familiarize himself or herself with the DBMS technology, the industry and the criteria for evaluating a DBMS engine. Gartner Group believes that 80 percent of IS shops will evaluate one or more DBMS engines and vendors over the next several years. They suggest a set of criteria for comparisons and a structure for evaluation. They examine several general evaluation issues and suggest specific criteria that should be used in an evaluation.

10.1. General Evaluation Issues

1. **Evaluation Criteria** - The first step is to begin gathering the application requirements such as database size, number of users, desired transaction throughput in transactions per second or minute and architectural requirements. In conjunction with the information gathering process, an RFI (Request for Information) could be sent out to the leading vendors to gain familiarity with the vendor and their offerings. Once this information is gathered internally, a request for proposal (RFP) could be sent to the DBMS vendors.

2. **DBMS Functionality Charts -** Gartner Group advises their clients against a function comparison chart for several reasons. Vendors tend to push these charts with features that are not needed or used by the development organization. Vendors also continuously leap each other in terms of functionality. Unless a feature is specifically required today, you can count on the vendor offering the feature in the reasonable future.

3. **Ranking or Weighting Systems -** Proper care must be taken with this approach since the important features may not carry enough weight to win over a large number of lesser values. If a numeric weighting system must be used, then it should be used in conjunction with subjective analysis.

4. **Sources of Information -** Do not rely on answers from vendor alone. Use third-party financial reports on the vendor and its products and written summaries from five to ten references. Many industry analysts such as Gartner Group also publish this kind of information.

5. **Makeup the Evaluation Team -** Include end users in addition to the normal group consisting of operations, technical support, DBAs and application development.

6. **Biased Questions -** Stay away from vendor suggested questions. Gartner Group notes that such questions are answerable by only the vendor that suggested it.

7. **Never Use the Word "MUST" -** Vendors may decline to answer an RFP if certain features are mandatory. Unless absolutely required (e.g., RDBMS must run on the HP/9000 hardware currently onsite), the preferred language is "desirable,", "should contain" or "important to contain."

8. **Functional Questions and Lists -** Great care must be taken when creating the list of required functionality so as not to preclude viable vendors from replying. Consider categorizing the functionality list by "required," "nice to have," and "optional."

9. **Software Status -** Never allow beta or nonproduction software to be referenced in any RFP or evaluation. If the evaluation is looking at mature DBMS engines only, then specify production or general availability for a minimum of 12 months. Also always ask for a list of release numbers, original release dates for beta and production and an estimate of the number of production licenses in use.

10.2. Evaluation Criteria

There are several specific criteria for the evaluation of the database engines and the vendors. Under most circumstances, no single DBMS evaluation will need to contain all of the items listed below. The actual RFP or evaluation should customize this list to those specific items that "fit" within the environment and architecture.

1. Company, Mission and Vision - Name, address and contact information. Request the mission and vision in depth. This is also the place for financial position and results, including references to annual reports and financial statements. Also, information such as fiscal year, total number of employees, G &A (general and Administrative) expense is appropriate for this section.

2. **Base Feature/Functional Capability** - This list should be a comprehensive list of features. The vendor should be encouraged not to answer this section with simple "yes" and "no" answers.

3. **Extended Functionality and Features -** These are the features that can help distinguish one engine from another. Examples of these features include Parallelism, Replication, Stored Procedures/Triggers, Object extensions, and Query Optimization.

4. **Reliability, Availability and Support** - This section should include information on the new release schedules, how well the vendor has met previous delivery dates, the number of employees dedicated to development technical support.

5. **Training and Consulting** - This section should contain the vendor's education offerings, education centers, cost of education, number of instructors and consultants (with worldwide breakdown) and sample listing of types of specific consulting engagements performed over the past year.

6. **Performance and Scalability** - TPC (Transaction Processing Council) benchmarks can be included here for reference. It should be noted, however, that TPC benchmarks are fabricated and not real-world. They should be used only as a guide. The only method for gaining finite information on performance is to perform a benchmark using actual production applications. Scalability is a very important concept, as it not only describes the growth potential of the database but also can be very revealing of design flaws in the engine.

7. **Database Connectivity, Interoperability and Platform Support** - This section should cover issues such as legacy DBMS support, ODBC implementation, support for IBM's DRDA connectivity standard. It should also include a complete list of hardware, operating system and resource requirements for each platform.

11. Conclusions [1,4,5,13]

During the past two years, the RDBMS have narrowed the gap in engine functionality. They have all delivered a growing, common set of functionality that fosters the notion of commoditization. With DBMS engines approaching functional equivalence, vendors are now striving for differentiation in other areas. In the mid-1980s, OLTP RDBMSs had to handle data measured in megabytes. Today, these databases must manage tens of gigabytes - and in the future they will have to cope with hundreds of gigabytes. Multimedia rich data and decision support systems of the future will require terabytes of data.

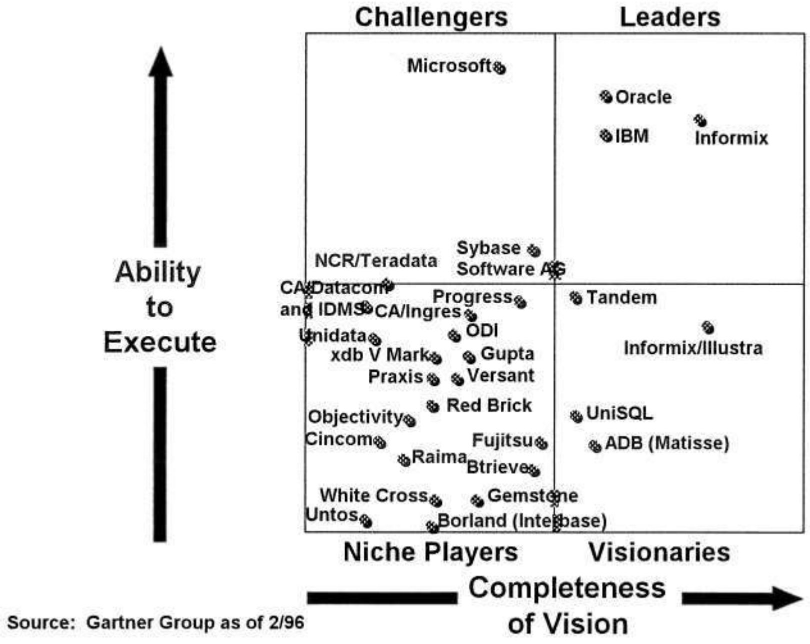
The leading RDBMS vendors are aggressively seeking to diversify their products as a means to ensure product differentiation and a potential competitive advantage. Some of key areas that enterprises should look for include:

RDBMS scalability
Parallel Processing
Support for Complex Datatypes
Data Replication
Query Optimization
Automated Operations

The Gartner Group publishes the DBMS "Magic Quadrant" which compares the players in the industry against two coordinates, the ability to execute and completeness of vision.

Figure 4

DBMS "Magic Quadrant"



Their February 1996 rating (Figure 4) shows how the DBMS market will evolve in the next five years. Oracle, Informix and IBM are in the Leader quadrant. Microsoft and Sybase are shown in the Challenger quadrant. The players in this industry will continue to diversify to a certain degree.

This emphasis on diversification is indicative of the desire on the part of the vendors to seek out new markets and is an important element in dispelling the notion that RDBMS engines should be perceived as commodities. For example:

- Sybase prides itself on having the best gateway support.

- Informix has made the exploitation of parallelism in both batch and on-line mode its trademark.

- Oracle strives to be the full-service provider, offering everything from CASE (Computer Aided Software Engineering) templates to services, as well as a reasonably competitive set of vertically integrated applications.

For enterprises, these new features are complicating the decision-making process. Far from being a tactical decision of choosing a commodity product, selecting an RDBMS is more of a strategic decision than ever. This report has presented the technology trends in the DBMS industry and the basics of evaluating a DBMS vendor and its engine. This is only a starting point. IS managers should put their own environment into each evaluation. Many features described here may have no pertinence for some IS environments. Finally RDBMS technology will continue to evolve adding new functions every 12 months. IS managers should expand this list to meet their needs and to make the process an ongoing project to reevaluate vendors and DBMS software on a regular basis.

12. Contact Information

Informix Software Inc.	4100 Bohannon Drive Menlo Park, CA 94025 Phone: (415) 926-6300
Oracle Systems Corporation	500 Oracle Parkway Redwood Shores, CA 94065 Phone: (415) 506-7200
Sybase Incorporated	6475 Christie Avenue Emeryville, CA 94608 Phone: (510) 658-9441

13. World Wide Web Pages

Informix Software Inc.	http://www.informix.com	Database vendor
Oracle Systems Corporation	http://www.oracle.com	Database vendor
Sybase, Inc.	http://www.sybase.com	Database vendor
Aberdeen Group, Inc.	http://www.aberdeen.com	Industry analyst group
GartnerGroup, Inc.	http://www.gartner.com	Industry analyst group
DBMS Magazine	http://www.dbmsmag.com	Industry magazine

14. References

1. Enslow *Tracking the Market Leaders,* Inside Gartner Group This Week, November 29, 1995
2. Kusnetzky *Relational Database Software - 1995 in Review,* IDC Reports #10861, December 1995
3. Feinberg; Strange *Evaluating a DBMS Vendor and its Engine,* GartnerGroup Research Note TV-000-049, June 30, 1995
4. Strange *The DBMS Market: A New Ranking Has Emerged, Part 1,* GartnerGroup Research Note M-550-116, February 27, 1996
5. Strange *The DBMS Market: A New Ranking Has Emerged, Part 2,* GartnerGroup Research Note M-550-116, February 27, 1996
6. *1996 Database Buyer's Guide and Client/Server Sourcebook,* DBMS Volume 9, No. 6
7. Radcliffe *RDBMS Scalability: Clear Differences Among Platforms,* GartnerGroup Research Note SPA-603-103, December 28, 1995
8. Hendrik *1995 RDBMS Server Feature/Function Audit,* IDC Report #9946, Vol. 1, August 1995
9. Radcliffe *Evaluating RDBMS Scalability in OLTP Systems - Selection Criteria,* GartnerGroup Research Note SPA-630-103, December 28, 1995
10. Percy *The Relational DBMS Market - On Main Street,* GartnerGroup Research Note TV-000-115, February 1996
11. *Sybase SQL Server 10 Whitepaper,* Sybase Inc., http://www.sybase.com/Offerings/Whitepapers/server10_wpaper.html
12. *The Distributed, Open Relational Database Management Systems Buying Guide,* Aberdeen Group, Inc., http://www.aberdeen.com/dors.htm
13. Kusnetzky *Relational Database Software - 1995 in Review,* IDC Report #10861, December 1995
14. Radcliffe *Ignoring Concurrency Issues Can Be Costly,* GartnerGroup Research Note TU-630-086, October 27, 1995
15. Radcliffe *RDBMS Concurrency Models Are Not Commodities,* GartnerGroup Research Note T-630-087, October 27, 1995
16. Radcliffe *Parallel RDBMS Architectures Explained and Assessed,* GartnerGroup Research Note TU-150-067, August 28, 1995
17. *Enterprise OLTP And Decision Support,* Oracle Corp. http://www.oracle.com/products/oracle7/server/whitepapers/
18. *INFORMIX-OnLine Dynamic Server, Version 7.2 for the UNIX® Operating System,* Technical Brief, http://www.informix.com/informix/products/techbrfs/
19. Davis, Judith. *Parallel Execution of Database Operations,* Open Information Systems, Vol. 10, No. 4, April 1995

20. Monash, C. *Software Obsolescence*, Monash Software Letter, Volume III, Issue 6, June 9, 1994

21. *SMP, Clustered SMP, and MPP, White paper*, Sequent Computers, 1995, http://www.sequent.com/public/news/papers/smp_mpp/smp.html

22. Cameron *Remote Database Management Promises - and some products*, Forrester Brief, Volume Five, Number Thirteen, September 30, 1994