

SQL Servers for OS/2

A Report from NSTL

In this report:

Program Evaluations -203

Rating Summaries..... -205

Performance Results..... -209

Vendors -212

Characteristics..... -213

Glossary of Terms..... -232

Synopsis

Focus
SQL Servers for OS/2 present a new environment for many business users. The SQL database offers a structured approach to data access in a client/server architecture. This report evaluates four SQL Servers for the OS/2 environment.

Program Recommendations

Microsoft SQL Server
Oracle Server for OS/2

Source

These tests were designed and conducted by National Software Testing Laboratories (NSTL), Inc., Plymouth Meeting, PA 19462. Telephone (800) 223-7093.

Programs Tested

IBM OS/2 Extended Edition
IBM Corp.

Microsoft SQL Server
Microsoft Corp.

Oracle Server for OS/2
Oracle Corp.

SQLBase Server
Gupta Technologies, Inc.

Ratings Key (On a scale of 0 to 10)	
Ratings	
●	7.0 - 10.0
○	5.0 - 6.9
◐	under 5.0
★	Recommended

Overall Rating	Program Name	Version Tested	Performance	Versatility	Error Handling	Ease of Learning	Ease of Use	Price
7.2	Microsoft SQL Server	1.0	●	○	○	●	○	\$2,495★
6.4	Oracle Server for OS/2	1.0	◐	○	○	○	○	\$2,499★
	IBM OS/2 Extended Edition							
5.0	Database Manager	1.2	◐	○	○	○	○	\$830
5.0	Gupta SQLBase Server	4.0.1	○	○	◐	◐	○	\$2,495

Overview

As their name implies, SQL servers use the powerful Structured Query Language command language. SQL can be used within applications to interface with databases. Database servers are central to the database management concept known as client/server architecture, in which applications run by users (client programs) interact with a separate database management system (server) that may reside on a system separate from the client programs.

With this evaluation, NSTL presents what it considers to be the definitive comparison of SQL database servers. Nearly five months of intensive research and hands-on testing preceded this report. In view of the quantity of existing test results (by vendors and reviewers) for these products, NSTL's evaluation goes beyond performance. Each program's features were evaluated against a 500-item checklist, and program usability and error handling were evaluated as well.

Although ease of learning and use may be irrelevant for products used to develop other applications, SQL servers must be accessible to programmers and database administrators, whose ability to master quickly the product's capabilities impacts productivity. Error handling is particularly important, combining analyses of safety and recovery features with tests of system failures and error conditions.

Evaluation Criteria

This report compares four SQL servers that run under the OS/2 environment: IBM OS/2 Extended Edition, Microsoft SQL Server, Oracle Server, and Gupta's SQLBase Server. DOS servers tend to be less sophisticated than OS/2 servers, in part because OS/2 is a more powerful operating system.

Oracle Server for OS/2 is essentially a mainframe/minicomputer product ported to OS/2 with only minor code modifications. A single-user Oracle product has been available to DOS users for several years, and the OS/2 server brings PC users

multiuser capabilities previously available only in larger environments. Oracle emphasizes its portability—applications can be moved from one environment to another with virtually no changes to the source code. Oracle Server for OS/2 is part of a total database management solution that includes a front-end interface and development tools called Professional Oracle, which is available for both DOS and OS/2.

Gupta Technologies' SQLBase Server was developed specifically for PCs. SQLBase under DOS became the first database server for PC LANs, and the OS/2 counterpart is relatively new. Gupta also offers front-end SQLWindows, which features a graphical interface. The Gupta product line emphasizes connectivity between databases, with tools that enable SQLBase databases to interact with other programs' databases.

Microsoft SQL Server is adapted to OS/2 from the Sybase server for minicomputer environments. Rather than offering a total database management solution with server and front end, Microsoft offers the server and programming tools, while encouraging third-party vendors to develop front ends. SQL Server incorporates important features not found in the other servers, but its usefulness has been hindered by the lack of front-end products, which are only beginning to appear.

IBM OS/2 Extended Edition is an extension of the OS/2 operating system. Unlike the other products that work with a separate base operating system and LAN operating system, IBM promotes an all-IBM solution encompassing the base operating system, database management, and communication between systems. The database management portion of Extended Edition cannot be used with a different operating system or LAN, and IBM supports the package only with IBM hardware. The included Query Manager front end provides an intuitive, menu-driven interface for data retrieval and for all other aspects of database administration.

Structured Query Language

The four products use variations of the SQL language to interface with the database. SQL's three basic components are the Data Definition Language (DDL), Data Control Language (DCL), and Data Manipulation Language (DML). DDL defines

data structures such as tables and views. DCL controls user access to data. DML adds, changes, and retrieves data.

Standard (ANSI) SQL establishes minimal language requirements, but each product has its own SQL "dialect" incorporating additional features and capabilities. Oracle's robust SQL implementation offers special functions and flexibility in statement syntax. Microsoft SQL Server matches Oracle's statement syntax, and provides control-of-flow language tools to compensate for any features it lacks. Add support for defaults, rules, and triggers, and SQL Server provides unparalleled power and flexibility. Gupta's SQLBase Server lacks some useful syntax options such as subqueries for creating and updating tables, but offers an excellent selection of special functions, including financial functions. IBM Extended Edition combines some outstanding features with severe limitations. Unique in its support of referential integrity constraints as part of table definition and its nonsupport for synonyms, it also lacks the same syntax options SQLBase Server lacks. Its almost complete neglect of nonaggregate special functions greatly weakens its overall SQL language implementation.

Network Capability

Once the network operating system has been installed, workstation and server software installation and connections are made easily in all the programs except IBM's. IBM's own Communications Manager handles communication among OS/2 stations, and its own Remote Data Services provide workstations with access to databases at remote locations. IBM's LAN Server and LAN Requester components must also be used if DOS workstations will access remote databases. Each workstation maintains a catalog of local databases; the workstation must catalog any remote workstations it will access and also catalog databases on those remote workstations.

As more users are added to the system and contention for resources grows, proper network configuration becomes crucial. IBM provides detailed documentation for configuring the Communications Manager for use with the Database Manager. SQL Server supplies a sample LAN Manager configuration file with notes on specific settings. Oracle and SQLBase provide no documentation regarding network settings. Oracle

provides detailed documentation on configuring its own parameters, but nothing about configuring the network. Users of both SQLBase and Oracle will find no clear information as to what database operations use what network and operating system resources.

Program Evaluations

IBM OS/2 Extended Edition

Strengths

- Embedded SQL
- Optimization during precompilation
- Excellent OS/2 interface
- Referential integrity constraints in table definition
- Execute server programs from workstation

Weaknesses

- Difficult to set up workstations
- No interface for DOS workstations
- Does not support non-IBM hardware
- Does not support non-IBM networks
- Very weak SQL language implementation
- No snapshot read
- Confusing documentation

Microsoft SQL Server

Strengths

- Stored procedures
- Procedural language as part of SQL
- Back-end triggers
- Column value constraints
- User-defined data types
- System procedures to access database information
- Two-phased Commit support
- Kill as selected user process
- Delayed and immediate shutdown

Weaknesses

- No snapshot read
- No exclusive lock on Select
- Programming interface difficult to use
- Supports only Named Pipes network protocol

Oracle Server for OS/2**Strengths**

- Embedded SQL
- Programming array support
- Robust SQL language implementation
- Can take an individual tablespace off-line
- Portability to other environments
- Snapshot read
- Sequence generator
- Delayed and immediate shutdown

Weaknesses

- Large number of configuration settings
- Poor handling of abnormal disconnections
- Difficult to set up Rollback segments
- Difficult to change physical file allocations
- No storing of preoptimized commands

Gupta SQLBase Server**Strengths**

- Good selection of SQL functions
- Windows interface for DOS workstations
- DOS server available
- Snapshot read
- Chained SQL commands

Weaknesses

- Very poor documentation
- Programming interface difficult to use
- Very slow at creating indexes
- Severe performance degradation on large inserts
- Unacceptable number and severity of bugs
- No exclusive lock on Select

Program Recommendations**Microsoft SQL Server**

SQL Server ranks highest in all the ratings categories. It is easiest to learn and use and the fastest overall. Back-end triggered procedures not only enforce referential integrity, but also any relationships between tables the user chooses to define. Control-of-flow language tools make its SQL language implementation more powerful than would merely adding special functions. Stored database procedures greatly enhance performance in that entire procedures with computations and variable manipulation can be stored in the database. Data definition is enhanced with user-specified default values, rules, and user-defined data types.

Oracle Server for OS/2

For users who cannot use SQL Server (e.g., network incompatibility; need to maintain the same system in several environments; need forms and report generation facilities), Oracle is a solid product. Oracle offers excellent performance, an outstanding SQL implementation including versatile syntax options and a wealth of special functions, a variety of configuration and backup/recovery options, and good programming features including array variable support in SQL statements. Oracle's complexity presents overwhelming amounts of information to be mastered, but all information is clearly presented and well organized.

Rating Summaries

Figure 1.
IBM/OS2 Extended Edition Ratings

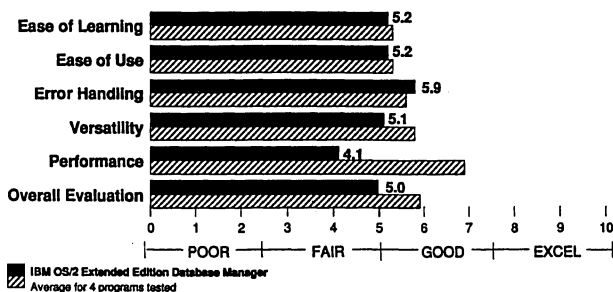


Figure 2.
Microsoft SQL Server

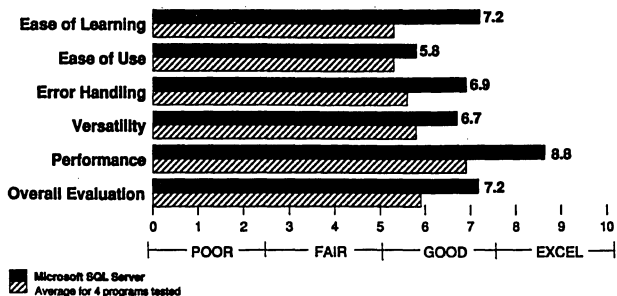


Figure 3.
Oracle Server for OS/2 Ratings

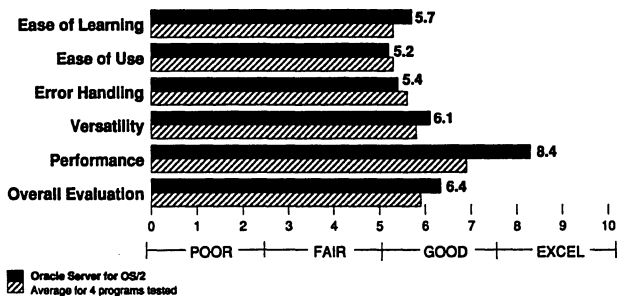
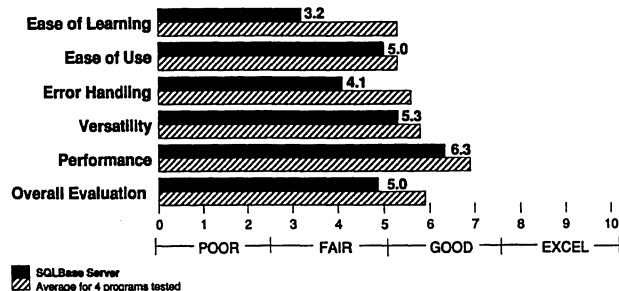


Figure 4.
SQLBase Server Ratings



Ease of Learning

Learning SQL servers is largely a function of the manuals and on-line help. SQL Server's concise package of four manuals gears the administrator's guide, programming guide, language reference guide, and language learning guide to specific audiences. Each provides all the information on its subject with virtually no need to cross-reference the other manuals.

The volume of information to be mastered with Oracle can be intimidating, but the process is made bearable by attractive and well-organized manuals. The OS/2 server package comes with 11 manuals including two different installation guides, three different programming guides, and a manual devoted to performance tuning. Oracle makes no claims to fast learning or mastery, but with time and concentration users can find all information, thoroughly and clearly documented with tables of contents and indexes. No cumulative index covers the entire package, but most information is intuitively found where expected. Cross-references, although frequent, are clear and specific.

Like Oracle, IBM Extended Edition presents an overwhelming volume of information to be mastered, but the manuals are much less friendly. Tasks require multiple references throughout the documentation; for example, the Database Manager is integrated with the operating system, requiring reference to different parts of the Database Manager documentation and to documentation covering the Communications Manager, LAN Requester, and base operating system, which individually pose major learning challenges. Once communications is established between workstations and remote databases, learning routine administration is greatly simplified by the Query

Manager on-line interface, which guides the user through literally everything with a series of menus and prompts. Query Manager works only under OS/2, and no comparable on-line interface exists for DOS workstations.

SQLBase comes with a main manual from Version 3.4, a supplement for Version 3.6.26, and another supplement for the current Version 4.0.1. Finding information often involves three references. No cumulative index covers all three, nor are the supplements individually indexed. (A single, combined manual is planned for the next release.) Even when the user finds the desired information, it is often unclear or incomplete. The instructions for creating a new database do not mention that a line must be added to a configuration file; this information is found elsewhere with a description of the configuration file. Instructions for setting a transaction to read-only do not mention a required global parameter setting. The programming documentation is the least clear of the sections, making it possible to study the manual's coverage of certain programming functions and still be unsure what syntax to use.

Methodology

Technicians evaluate the adequacy of program manuals as learning tools, and using the manuals and any on-line help, evaluate how easily users can learn to install the program, create and configure any necessary files, perform system administration tasks, manage data, and use the applications programming interface.

Ease of Learning Score = $(3 \times \text{Manuals Score}) + (3 \times \text{Installation and File Management Score}) + (3 \times \text{System Administration Score}) + (\text{Data Management Score}) + (2 \times \text{Programming Score}) \div 12$.

Ease of Use

Automated features contribute to making an SQL server easy to use. IBM Extended Edition and SQLBase Server create and allocate physical files automatically as needed by the database; the user does not have to explicitly allocate files as space is needed. Oracle Server and Microsoft SQL Server use specific commands to incorporate files into the database, whether for data storage or the transaction log.

Installation refers to the initial system setup and to the addition of workstations or databases over time. Once learned, SQLBase's installation is quite easy, and databases are added with a few quick commands. Microsoft SQL Server uses a menu-driven installation procedure. Adding and changing databases involve several steps that could probably be combined, but the procedures are clearly documented. Installing Oracle with the default settings is easy, but dozens of configuration parameters must be balanced in order to customize the system size, number of workstations, and type of application. Configuring IBM Extended Edition for multiple stations requires interrelated Database Manager, Communications Manager, and LAN Requester settings, and a complex system of cataloging is needed to make workstations recognize databases at remote locations.

Administration may be simplified using on-line interfaces and utilities. With IBM's Query Manager interface (available for OS/2 workstations only), virtually all tasks can be executed through a series of menus and prompts. Oracle's Server Manager, also for OS/2 workstations, provides similar capability, but many tasks including database monitoring require other utilities. SQLBase Server uses a Microsoft Windows interface for DOS, but most monitoring must be done at the server. SQL Server offers an interface for DOS and OS/2 with menu options for most administration tasks, and system procedures that allow workstation users to access system information.

Programming tends to be easiest in systems that offer embedded SQL with a precompiler. The precompiler translates the SQL commands into host language calls, rather than the programmer having to code the interface with the database directly. Oracle and IBM include embedded SQL. SQLBase and SQL Server do not offer embedded SQL, but in SQL Server, host language programs can execute stored procedures, which are universally easy to write.

Methodology

Technicians evaluate the adequacy of program manuals as reference guides, as well as the ease of installing the program, of creating and configuring system files, and of system administration tasks such as start-up/shutdown, backup/recovery, monitoring of users, setting performance parameters, managing data, and using the application programming interface.

Ease of Use Score = (Manuals Score) + (3 × Installation and File Management Score) + (3 × System Administration Score) + (Data Management Score) + (2 × Programming Score) ÷ 10.

Error Handling

Program documentation ranges in thoroughness with respect to error conditions. Oracle Server devotes a separate manual to error messages and instructions for recovery. The IBM appendix on error handling is not as easy to handle as a separate book, but it is nearly as thorough; in some cases, attempts to document several conditions in a single explanation become confusing. Microsoft SQL Server's error documentation is sketchy at best, and some on-line messages may be vague, such as "syntax error." SQLBase Server offers the user no explanation beyond the on-screen message.

Microsoft SQL Server offers the best selection of other error handling features, including direct referential integrity, and backup and recovery options. Backup files can be assigned time constraints, which prevent reuse before the specified time. SQL Server also has the ability to kill a specific client process, which avoids having to bring down the entire system when a single user has an unresolvable problem. IBM Extended Edition also offers direct referential integrity support. Oracle offers flexible backup and recovery options; different types of backups include the ability to take an individual tablespace off-line and back it up while the database is in use.

SQL Server is the only program in which NSTL's technicians did not encounter serious bugs. Oracle has difficulty handling abnormal workstation disconnects such as occur with power losses or system failure. If too many of these happen it becomes necessary to shut down and restart the server.

With both IBM Extended Edition and SQLBase Server, NSTL was able to crash the system in such a way that the database could not be recovered. Such a crash occurred only once with IBM, but with troubling frequency in SQLBase. Enabling the log file preallocate option caused an unrecoverable crash on the first rollback, and on another occasion, changing the log file drive designation without shutting down the server rendered the database unrecoverable.

Methodology

Error handling scores combine evaluations of safety and integrity features with test results for recovery procedures. NSTL evaluates the adequacy of on-line error messages and their documentation, transaction handling, backup and recovery options, recovery from data definition language errors, and referential integrity support. The programs are tested for recovery from power losses at the server and workstation and for their handling of filled storage capacity. Scores are also assigned for the number and severity of bugs encountered during testing. Bug refers to anything that is not as it should be, from a misspelled screen message to an unrecoverable database crash.

Error Handling Score = (2 × Messages Score) + (Transaction Handling Score) + (3 × Backup/Recovery Score) + (3 × Power Loss/Disconnect Score) + (DDL Errors Score) + (2 × Referential Integrity Score) + (2 × Capacity Filled Score) + (3 × Bugs Score) ÷ 17.

Performance

Microsoft SQL Server and Oracle Server earn very close performance scores with almost mutually exclusive capabilities. Oracle is especially fast with operations that make use of its program arrays as SQL statement variables, such as for multirow inserts and updates. SQL Server is significantly faster than the other programs for all types of operations except inserts requiring index updates. For this type of insert, SQL Server is among the slowest.

SQLBase Server performs roughly on par with Oracle for selects and single-row updates, but is not as fast for operations where Oracle can use array support. SQLBase is faster than SQL Server for inserts that update indexes, but not as fast for selects.

IBM Extended Edition slows considerably for selects using the SQL LIKE operator. In other tests with large numbers of users (in particular, those tests where SQL Server is slowed by updates to indexes), IBM exhibits speed comparable to SQL Server's.

Methodology

Each program's performance score is based on its average speed of execution for four different types of transactions, each involving different types of

database operations. Each of the four tests is run using 1, 8, 16, and 26 concurrent users.

$$\text{Individual Performance Score} = (\text{Single-User Score}) + (2 \times \text{8-User Score}) + (3 \times \text{16-User Score}) + (2 \times \text{26-User Score}) \div 8.$$

$$\text{Performance Score} = (3 \times \text{ISBN Order Score}) + (\text{Author Order Score}) + (2 \times \text{Payment Score}) \div 6.$$

Versatility

Microsoft SQL Server offers important unique features, including triggered procedures, defaults and rules in table definitions, user-defined data types, and two-phase commit support. Its Transact-SQL language adds procedural logic to standard SQL, greatly enhancing the flexibility of a single SQL procedure. The capability to store and execute such procedures simplifies programming and enhances performance. SQL Server scores first or second in every major features category except Data Integrity, where it is weakened by the lack of explicit locking and exclusive locks on select statements.

Although it does not provide as many unique features as SQL Server, Oracle Server is a solid program with a good selection of features in all areas. Except for stored procedures, it is difficult to identify a category in which Oracle alone is lacking. Only Oracle supports the use of program arrays as SQL variables, and it has a variety of options for file configuration, indexing and clustering, fine-tuning performance, backup, and database administration. Compatible servers are available in a number of operating environments, and it supports virtually any LAN configuration. Its database link feature makes it the only product to support joined selects from databases in different locations.

Again IBM Extended Edition exhibits areas of strength and weakness. Only Extended Edition includes referential integrity constraints as part of table definition, but otherwise its SQL language is very limited, with few special functions and no ability to update a column to a subquery. Using the remote execute feature, a workstation program can execute a program on the server. IBM provides the most options for concurrency control, offers embedded SQL for programming, and provides the most powerful interactive SQL interface (available only for OS/2 workstations).

No outstanding unique features nor glaring omissions characterize SQLBase Server; it supplies most of the features common among its competitors. Good SQL language implementation includes a variety of special functions, including financial functions not offered by the other products. The SQLTalk for Windows interface permits menu access to most administrative functions on DOS workstations. Stored commands can be retrieved and executed in a batch. Like SQL Server, SQLBase lacks explicit lock capability and the capability to acquire an exclusive lock on a select.

Methodology

The versatility score is a weighted average of the individual criteria scores.

$$\text{Versatility Score} = (3 \times \text{Data Definition Features Score}) + (4 \times \text{Security/Database Administration Score}) + (3 \times \text{Data Manipulation Score}) + (2 \times \text{Data Integrity Score}) + (4 \times \text{Programming and Other Tools Score}) + (\text{Portability and Network Capability Score}) \div 17.$$

Overall Evaluation

Microsoft SQL Server combines several unique features and excellent speed. Stored procedures and procedural language extensions not only enhance its power and speed, but simplify programming tasks. Its menu-driven system administration interface and predefined procedures for accessing database information simplify routine administration. Documentation is concise and thorough.

Oracle Server provides nearly the same levels of versatility and performance, but learning and administration are complex. All facets of administration are thoroughly and clearly documented.

IBM Extended Edition exhibits variable strengths and weaknesses. Only IBM offers referential integrity support as part of table definition, a remote interface whereby a workstation application can instruct the server to execute a program, and a Presentation Manager interface with menus and prompts for literally every function. Its SQL language implementation is very limited. Extended Edition is very difficult to install on multiple workstations. Its documentation is thorough, but difficult to understand.

Gupta's SQLBase is not a complex product, but usability is hindered by poor documentation and features that do not work as expected. Its

breadth of features is good; none is particularly unique, but no serious gaps are apparent.

Methodology

Overall Evaluation Score = $(4 \times \text{Ease of Learning Score}) + (5 \times \text{Ease of Use Score}) + (2 \times \text{Error Handling Score}) + (6 \times \text{Versatility Score}) + (6 \times \text{Performance Score}) + (\text{Testers' General Evaluation Score}) \div 24$.

Performance Results

Test Files

To test the performance of SQL database servers, NSTL developed a database for book order entry. A Book table contains one record for each book in inventory. Each record contains a unique ISBN number, title, price, and inventory information, with a unique index on the ISBN number and a nonunique index on the title. An Author table contains one record for each person who has authored or coauthored a book in inventory. Each record contains a unique ID number and the author's name with a unique index on the ID and a nonunique index on the name. A Link table links each author to the books he or she has written. Each link record contains an author ID and an ISBN number, each with a nonunique index. An Order table contains one record for each order placed, each record having a unique order number and customer information with a unique index on order number. An Entry table contains one or more records per order, each record corresponding to a book that has been ordered with a nonunique index on order number.

For this test, data was generated for 50,000 books and 10,000 authors; 100,000 link records paired each author with 10 books, and each book had two coauthors. Testing began with 20,000 orders, and the number of orders increased during the course of testing. The tests use the somewhat artificial ratio of five entries for every order, or

100,000 entries at the outset. For all the programs except Oracle, an additional table of one row and one column contained the most recently entered order number. Oracle used its own sequence generator in place of this table.

System Configuration and Database Creation

A 25MHz IBM PS/2 Model 80 with 16 megabytes of memory and two 300-megabyte disk drives acted as both the LAN server and the database server. Workstations were Everex 386 systems with 4 megabytes of memory and IBM Token-Ring adapters. For all the programs except IBM OS/2 Extended Edition, the server used the IBM OS/2 1.1 operating system, the workstations used the Compaq OS/2 1.1 operating system, and the network used the 3Com 3+Open 1.1 LAN operating system. IBM Extended Edition used its integrated base operating system, Communications Manager, and Database Manager. All the programs were set up to use one hard disk for the database and the other for log files. The application software resided on the same disk with the database in a separate partition. Oracle's rollback segments (at the vendor's recommendation) were located on the disk with the data tables.

IBM and SQLBase supply utilities for creating databases, with space allocated automatically during processing. SQL Server generates a 50-megabyte database file on the data drive and a 30-megabyte file allocated as the log device on the log drive in addition to the program's 10-megabyte start-up database. Oracle generated a 10-megabyte start-up database allocated to the system tablespace, a 50-megabyte tablespace for the test databases and indexes, a 30-megabyte tablespace for rollback segments, and two 15-megabyte log files on the log drive.

Test programs were written in C using each program's applications programming interface (API). Oracle offers a precompiler and a direct call interface (APIs), and on the vendor's recommendation, NSTL used the direct call interface. In SQL Server, SQL commands were executed as stored procedures in all but one case. SQLBase used stored command retrieval in all but that same case. IBM's precompiler optimizes SQL statements during precompilation and stores the resulting action plan in the database; this method is used for all commands with the same exception noted for SQL Server. The exception is a statement that compares

database columns with a wild card string using the SQL LIKE operator. Using the LIKE operator with the wild card string resulted in unacceptable performance (several minutes to execute a query), so for this query, dynamic SQL was used and the wild card string was patched directly into the command text prior to parsing.

Tests were executed with 1, 8, 16, and 26 concurrent users. At each user level, each test ran for five minutes, and the program computed the average time to process a single transaction (i.e., a single order or payment, depending on the test). No "think time" is inserted between transactions.

ISBN Order

An ISBN number was generated at random. The corresponding book was selected, and a three-table join on the Book, Link, and Author tables identified the book's authors. The selected book was retained or discarded based on a predetermined percentage. When five books had been selected and retained, an order was processed. Order processing consisted of incrementing the current order number, inserting an order record, inserting five entry records, and updating the inventory information for the five books ordered.

Single User

The order processing was most significant in the single-user test because the other tests did considerably more selecting. Oracle was by far the fastest at this portion of the test. Oracle's use of program arrays as variables in SQL statements enabled a single SQL statement to insert five entries and update five book records instead of using five statements. SQL Server used five statements, but performed selects (which in this test were done as stored procedures) over 50 percent faster than Oracle. IBM executed inserts and updates with speed comparable to SQL Server, and its selects were slower than Oracle's. SQLBase was not as fast as Oracle for inserts and updates, but its superior select speed gave it a better overall score.

Eight Users

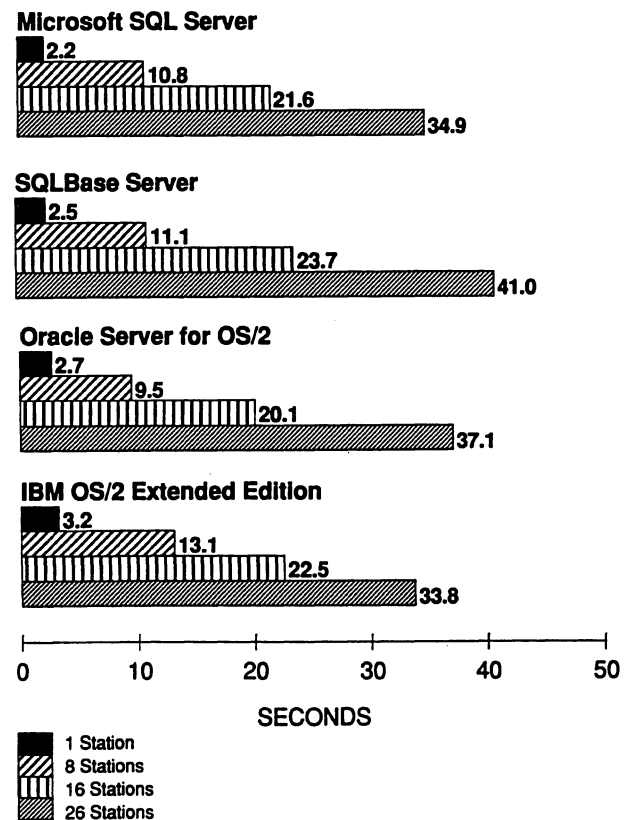
Oracle exhibited less performance degradation than SQL Server as a result of adding users, but Oracle's data cache had to be reduced to compensate for the demand on other resources. With only

eight times the number of users contending for resources, SQL Server took 11 times longer to process an order, resulting in less contention on the selects. IBM Extended Edition was actually faster than SQL Server when processing orders at this workload. SQLBase was still faster than Oracle at selects, but its order processing speed degraded substantially.

Sixteen Users

Oracle remained the fastest with 16 users despite further reductions in cache size because the array inserts and updates took only three times longer with 16 users than with one user. Oracle was actually the slowest at selects at this workload, but its order processing speed more than compensated. SQL Server's order processing speed degraded from 8 users to 16 users, and at this workload its superior select speed was barely sufficient to put it

Figure 5.
ISBN Order



ahead of IBM Extended Edition. SQLBase continued to do well with selects, but its order processing speed was over three times slower than with eight users.

Twenty-Six Users

IBM Extended Edition showed the least performance degradation as users were added, and as a result was the fastest with 26 users. The relationships between the other three programs were unchanged.

Author Order

This test simulated the process whereby a customer does not know an author's name, but identifies the correct name from a list, looks up books written by that author, and selects the correct book. The program selects Author names corresponding to a partial name key (with wild card character) using the SQL LIKE operator. One author and his/her books are selected by means of a three-table join on the Author, Link, and Book tables. When the desired title is found, additional information is selected from that book's record. The entire process is repeated until five books have been selected, and an order is processed.

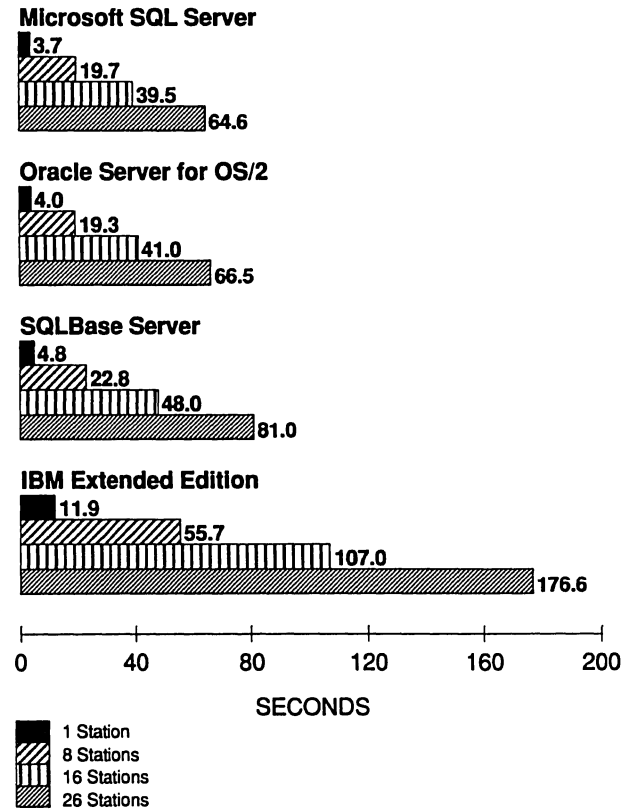
Single User

This test processes orders the same as in the first test, but executes a greater number of selects before processing each order. SQL Server was by far the fastest at selects, even though many of the selects were executed dynamically rather than through stored procedures. IBM Extended Edition exhibited particular difficulty with selects using the LIKE operator, which were executed multiple times for each of the five books. IBM's order processing time was faster than SQL Server's. Unlike in Benchmark A, Oracle performed slightly faster than SQLBase for selects; SQLBase appeared to have a problem similar to IBM's, but of far less magnitude.

Eight Users

SQL Server and Oracle remained very close in overall performance, with SQL Server being faster at selects and Oracle being faster at updates and inserts. IBM Extended Edition continued to struggle with the LIKE selects.

Figure 6.
Author Order



Sixteen Users

The programs exhibited the same basic relationships with 8 and 16 users. All except IBM took more than twice as long to execute a transaction than with eight users; in other words, with these three programs, 8 users could get more work done in a given amount of time than 16 users could.

Twenty-Six Users

SQL Server and Oracle were still very close overall, with SQLBase showing a somewhat greater performance degradation than the others.

Payment

An order number is generated at random. The Order record and its corresponding entries are selected, and the Order record updated.

Single-User

The two selects and update in this test involve single rows. Because Oracle could not take advantage of its array capabilities in this test, SQL Server performed faster at both selects and updates. All the SQL statements were somewhat long and complex in syntax, but relatively quick to execute, and SQL Server needed only to pass to the server the command to execute a stored procedure rather than use an entire SQL command. With SQLBase's stored commands, the entire command must be retrieved from the server and then sent back to the server.

Eight Users

Perhaps because of the relative simplicity of this test, the programs maintained the relative performance established with a single user.

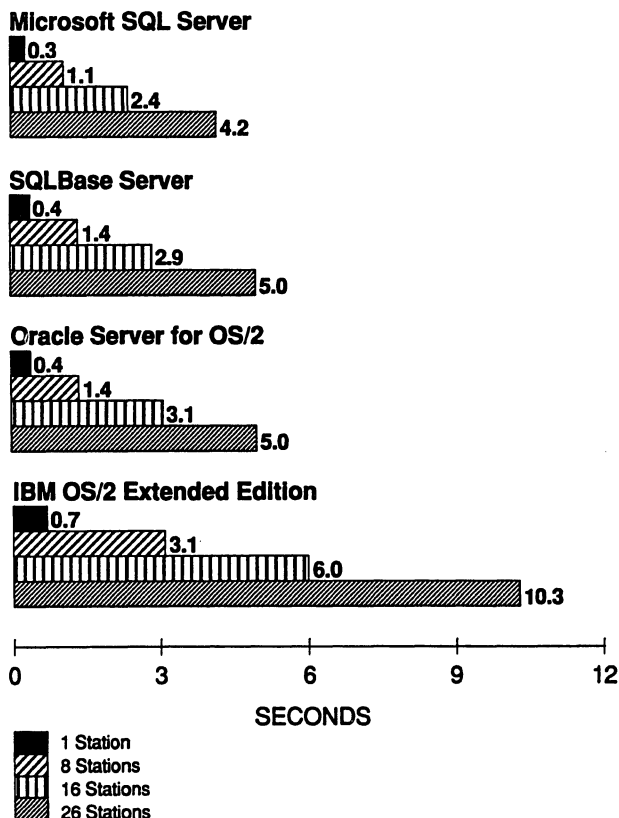
Sixteen Users

Although relative performance stayed the same with a single user and with eight users, IBM Extended Edition exhibited significantly less performance degradation than the other programs as users increased. Degradation from 8 to 16 users was less than 30 percent, and IBM actually performed selects at this workload slightly faster than Oracle. While IBM and SQL Server required much the same time to process an order (for inserts and updates), SQL Server was over three times faster with a simple update. SQL Server's single weakness in performance seemed to be with inserts, particularly if the table was indexed.

Twenty-Six Users

No significant changes distinguished the programs' relative performance from 16 to 26 users.

Figure 7.
Payment



Vendors

Gupta Technologies, Inc.

1040 Marsh Road
Menlo Park, CA 94025 (415) 321-9500

International Business Machines Corp. (IBM)

Old Orchard Road
Armonk, NY 10504
Contact your local IBM representative.

Microsoft Corp.

One Microsoft Way
Redmond, WA 98052 (206) 882-8080

Oracle Corp.

20 Davis Drive
Belmont, CA 94002 (415) 595-0630

Characteristics

Table 1. SQL Servers for OS/2 Characteristics

	Requirements	Supplier Support
IBM OS/2 Extended Edition 1.2 (Released March 1990)	<ul style="list-style-type: none">• Drives: Hard disk of sufficient size for database• Server memory: 4.5M bytes• Workstation memory: 3.5M bytes plus 1.6MB for Query Manager• Operating system: IBM LAN Support 1.1, DOS 3.3, OS/2 LAN Server 1.2• Network operating systems: IBM LAN Server	IBM Personal Systems Developer Publication; training programs available
Microsoft SQL Server 1.0	<ul style="list-style-type: none">• Drives: 40M-byte hard disk (otherwise determined by size of database)• Server memory: 8M bytes plus 42K bytes per connection; 10M bytes recommended• Workstation memory: 640K bytes; 3M bytes recommended• Network operating systems: All networks supporting Named Pipes	Telephone support; bulletin board; Microsoft Systems Journal; Microsoft University training program available
Oracle Server for OS/2 1.0	(Released October 1989) <ul style="list-style-type: none">• Drives: Hard disk of sufficient size for database• Server memory: 8M bytes plus 512K bytes per connection• Workstation memory: 640K bytes (DOS); 2MB bytes (OS/2)• Network operating systems: Banyan Vines, IBM NETBIOS, Novell SPX/IPX and NETBIOS, 3Com NETBIOS, Named Pipes	Journal; maintenance customers receive toll-free, 24-hour telephone support; automatic upgrades; training available
Gupta SQLBase Server 4.0.1 (Released November 1989)	<ul style="list-style-type: none">• Drives: 20M-byte hard disk (otherwise determined by size of database)• Server memory: 4M bytes plus 3K bytes per connection• Workstation memory: 512K bytes (DOS); 2M bytes (OS/2)• Network operating systems: All NET-BIOS-compatible networks	Telephone support and maintenance releases free for 90 days; Extended Support (\$300, DOS version; \$375, OS/2) includes telephone support, bulletin board, technical newsletter, maintenance releases, preferred access to training

Table 2. Data Definition Features

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Program Parameters	0				
Maximum records per table	0	U	U	U	U
Maximum columns per record	0	255	255	254	254
Maximum characters per record	0	4,005	A	130K	C
Maximum characters per alpha- numeric field	0	4K	A	64K	D
Maximum tables per database	0	U	U	U	U
Memory required (server)	0	4.5MB	8MB	8MB	4MB
Memory recommended (server)	0	NA	10MB	16MB	4MB
Memory required (OS/2 workstation)	0	3.4MB	3MB	4MB	2MB
Memory required (DOS workstation)	0	NA	640K	640K	512K
Maximum independent data- bases per server	0	U	B	1	255
Maximum users per database	0	U	250	40	U
SQL DDL Keywords	2				
CREATE TABLE	1	▲	▲	▲	▲
CREATE TABLE using query	—	E	▲	▲	
DROP TABLE	1	▲	▲	▲	▲
ALTER TABLE	1	▲	▲	▲	▲
ALTER TABLE ADD COLUMN	1	▲	▲	▲	▲
ALTER TABLE MODIFY COLUMN	1	—	—	▲	▲
ALTER TABLE DROP COLUMN	1	—	—	—	▲
CREATE INDEX	1	▲	▲	▲	▲
DROP INDEX	1	▲	▲	▲	▲
CREATE VIEW	1	▲	▲	▲	▲
DROP VIEW	1	▲	▲	▲	▲
CREATE SYNONYM	1	—	—	▲	▲
CREATE PUBLIC SYNONYM	1	—	—	▲	▲
DROP SYNONYM	1	—	—	▲	▲
Define object as temporary	2	—	▲	—	—
Rename column	2	—	▲	—	▲
Rename a table	2	—	▲	▲	▲
Domains	2				
Boolean	1	—	▲	—	—
Integer	2	▲	▲	F	▲
Short Integer	1	▲	▲	F	▲
Floating point	1	▲	▲	F	▲
Fixed point or money	1	▲	▲	F	▲
Fixed length character	2	▲	▲	—	—
Variable length character	2	▲	▲	▲	▲
Name	1	—	—	—	—

See Footnotes on page 217

Table 2. Data Definition Features (Continued)

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Date	2	▲	G	G	▲
Time	2	▲	G	G	▲
Fraction of second	2	H	G	—	▲
Date and time in same variable	2	▲	▲	▲	▲
Long text	2	▲	▲	▲	▲
Long binary	1	▲	▲	—	—
User-defined domains	2	—	▲	—	—
User-defined composite domains	1	—	—	—	—
Specify order of new domains	1	—	—	—	—
Specify conversion rules	1	—	—	—	—
Null value distinct from domain values	2	▲	▲	▲	▲
Domain Integrity	1				
Range check	1	—	▲	—	—
Set check	1	—	▲	—	—
Pattern check	1	—	▲	—	—
Formula check	1	—	▲	—	—
Procedure check	1	—	—	—	—
Date checking	2	▲	▲	▲	▲
Time checking	2	▲	▲	▲	▲
Define default value	1	—	▲	—	—
Structural Integrity	1				
Primary key	2	▲	▲		
Foreign key	2	▲	▲	—	—
Alternate key	—	1			
Foreign key may reference alternate key	1	—	—	—	—
Multiattribute primary key	2	▲	▲		
Multiattribute foreign keys	2	▲	▲	—	—
Enforce uniqueness of primary key	2	▲			
Enforce nonnull in primary key	1	▲	—	—	—
Referential Integrity	3				
Referential integrity enforced by data dictionary	2	▲	—	—	—
Referential integrity enforced by triggers	1	—	▲	—	—
Cascaded deletions	1	▲	J	—	—
Cascaded nulls upon deletion	1	▲	J	—	—
Forbid deletion of referenced record	1	▲	▲	—	—
Cascaded updates (referential)	1	—	J	—	—
Cascaded nulls upon update	1	—	J	—	—
Forbid update (referential)	1	▲	▲	—	—

J—Cannot cascade triggers; each trigger must address all referential levels for that table.

See Footnotes on page 217

Table 2. Data Definition Features (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Other Integrity Rules	2				
Not null	3	▲	▲	▲	▲
Unique	3	O	K	K	K
Range check	2	—	▲	—	—
Predicate check	1	—	L	—	—
Immediate record transition constraints	1	—	M	—	—
Deferred status constraints	1	—	—	—	—
Set validation	2	—	▲	—	—
Formula validation	1	—	L	—	—
Pattern validation	1	—	▲	—	—
Define default value	3	—	▲	—	N
System Catalog	2				
Query the system catalog	2	▲	▲	▲	▲
System catalog maintained automatically	2	▲	▲	▲	▲
Database-wide information	2	▲	▲	▲	▲
User-specific information	1	—	—	▲	—
Tables	3	▲	P	▲	▲
Columns	3	▲	▲	▲	▲
Indexes	2	▲	▲	▲	▲
View definitions	1	▲	▲	▲	▲
View dependencies	1	▲	▲	—	—
Procedure definitions	1	Q	▲	—	—
Procedure dependencies	1	Q	▲	—	—
Space allocation by object	1	▲	▲	▲	—
Space allocation for database	1	—	▲	▲	—
User information	1	—	▲	▲	▲
User authorizations	1	▲	▲	▲	▲
User authorizations by object	1	▲	▲	▲	▲
Physical device assignments	1	—	▲	▲	—
Locks	1	—	▲	—	—
Views	2				
Restriction views	2	▲	▲	▲	▲
Projection views	2	▲	▲	▲	▲
Join views	2	▲	▲	▲	▲
Views with calculated fields	2	▲	▲	▲	▲
Grouped views	2	▲	▲	▲	▲
Views with check option	3	▲	—	▲	▲
Delete from restriction view	1	▲	▲	▲	▲
Insert into restriction view	1	▲	▲	▲	▲
Update restriction view	1	▲	▲	▲	▲
Insert into projection view	1	▲	▲	▲	▲
Delete from projection view	1	▲	▲	▲	▲

See Footnotes on page 217

Table 2. Data Definition Features (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Update projection view	1	▲	▲	▲	▲
Update joined view	1	—	R	—	—
Insert into view defined with subquery	1	▲	—	▲	▲
Delete from view defined with subquery	1	▲	—	▲	▲
Update view defined with subquery	1	▲	—	▲	▲
Query grouped view using WHERE	1	▲	▲	▲	▲
Query grouped view using aggregate functions	1	▲	▲	▲	▲

A—1,920 for standard columns; unlimited for long text or long binary.

B—Master database must be present and is thus only independent database; 32767 total.

C—1924 for standard columns; unlimited for long text.

D—No limit other than max number of columns times max column size.

E—Result achieved by a different syntax.

F—Single numeric domain for which size and precision can be defined.

G—Single domain containing date and time.

H—Timestamp (date and time) variable includes microsecond; time variable does not.

I—Enforced via a unique index on each key.

J—Cannot cascade triggers; each trigger must address all referential levels for that table.

K—Enforced via a unique index.

L—Formula cannot reference another column.

M—Define trigger to compare old and new values.

N—Zero for numbers, current date/time for date/time.

NA—Vendor does not publish this information.

O—Enforced via unique index or designation as key field.

P—Sysobjects table also contains rules, defaults, and procedures.

Q—Maintains information about access plans for precompiled applications.

R—If all columns to be updated are from same table.

U—Unlimited (limited only by space and workload considerations).

Table 3. Security/Database Administration

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Security and Data Control Language	2				
Incorporates network security	1	▲	—	—	—
Change administrator password	2	▲	▲	▲	▲
Users change own passwords	1	▲	▲	—	▲
Grant privileges	2	▲	▲	▲	▲
Grant with grant option	3	V	—	▲	▲
Grant without grant option	3	▲	▲	▲	▲
Connect privilege	2	▲	▲	▲	▲
Create database privilege	1	W	▲	▲	—
System administrator privilege	2	▲	—	▲	S
Create table privilege	2	▲	▲	▲	▲

See Footnotes on page 220

Table 3. Security/Database Administration (Continued)

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Privilege to alter/drop existing table	1	▲	—	—	T
Privilege to create index on existing table	2	▲	—	▲	▲
Privilege to drop existing index	1	▲	—	—	—
Privilege to create trigger on existing table	1	—	—	—	—
References privilege	2	▲	—	—	—
Privilege to create stored procedures	1	X	▲	—	—
Privilege to execute existing procedure	1	X	▲	—	—
Privilege to alter/drop existing procedure	1	X	—	—	—
Select privilege	2	▲	▲	▲	▲
Update privilege on table	2	▲	▲	▲	▲
Update privilege on specific columns	2	—	▲	▲	▲
Insert privilege	2	▲	▲	▲	▲
Delete privilege	2	▲	▲	▲	▲
Grant all on table	2	▲	▲	▲	▲
Grant to public	2	▲	▲	▲	▲
Grant to user group	1	▲	▲	—	—
Grant select on view	2	▲	▲	▲	▲
Grant update on view	2	▲	▲	▲	▲
Grant insert on view	2	▲	▲	▲	▲
Grant delete on view	2	▲	▲	▲	▲
Privilege on view of withheld table	2	▲	▲	▲	▲
Backup/restore privilege	1	—	▲	—	—
Auditing	1				
Table audit	1	—	—	▲	—
User audit	1	—	—	▲	—
Access audit	1	—	—	▲	—
Session audit	1	—	—	▲	—
Audit specified commands	1	—	—	▲	—
Audit successful/unsuccessful access	1	—	—	▲	—
Physical Options	3				
Maximum B-tree (VSAM) indexes	2	U	250	U	U
Maximum multikey B-tree (VSAM) indexes	2	U	250	U	U
Maximum hash indexes	1	0	0	0	1
Maximum multikey hash indexes	1	0	0	0	1

See Footnotes on page 220

Table 3. Security/Database Administration (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Maximum indexes per table	2	U	250	U	U
Maximum columns per index key	2	16	16	Y	10
Unique indexes	2	▲	▲	▲	▲
Nonunique indexes	2	▲	▲	▲	▲
Index on expression	1	—	—	—	▲
Clusters	2	A2	▲	▲	▲
Multitable clusters	1	—	—	▲	—
Database on multiple disk drives	2	—	▲	▲	—
Transaction log on separate drive	3	▲	▲	▲	▲
Assign objects to specific files	2	—	Z	▲	—
Define minimum contiguous storage	1	—	—	▲	—
Define minimum contiguous storage for table	1	—	+	▲	—
Automatic space reclamation	2	—	▲	—	—
Set fill factor	1	—	A1	▲	A1
Database Administration	3				
Start-up/shutdown from remote station	1	A3	▲	A3	—
Delayed shutdown	1	—	▲	▲	—
Immediate shutdown	1	—	▲	▲	▲
Automatic file creation	2	▲	—	▲	▲
Automatic file expansion	2	▲	—	—	▲
Backup single table	1	▲	—	▲	▲
Backup selected tables	1	—	—	▲	▲
Backup user schema	2	—	A5	▲	▲
Backup entire database	2	▲	▲	A4	▲
Backup without shutdown	3	—	▲	▲	▲
Partial shutdown/backup/restore	2	—	—	▲	▲
Restore selected portion of backup	1	—	—	▲	—
Incremental backup	1	▲	—	—	—
Backup to/restore from workstation	1	A7	—	▲	▲
Initiate backup/restore from server at workstation	1	—	▲	—	▲
Monitor activity by process	1	▲	▲	▲	A6
Monitor locks	1	—	▲	▲	▲
Monitor statistics	1	▲	▲	▲	▲
Monitor from workstation	1	A7	▲	A3	—
Stop specified process	2	—	▲	—	—

See Footnotes on page 220

Table 3. Security/Database Administration (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Performance Tuning Features	2				
Syntax-independent optimization	2	—	▲	—	▲
Show optimization plan	1	—	▲	▲	▲
Trace feature	1	—	▲	▲	—
Set checkpoint interval	1	▲	▲	▲	▲
Disable archiving	2	—	▲	▲	A9
Set total memory size	3	—	▲	▲	▲
Set memory for programs and data	2	▲	▲	▲	—
Assign memory for parsed statements	1	▲	—	▲	—
Assign memory for data dictionary	1	—	—	▲	—
Single process with multiple threads	2	—	▲	—	—

S—Only the original system administrator may add new users and change users' passwords.
T—Alter only.
U—Unlimited (limited only by space).
V—Complete control of an object can be granted with grant option; not individual privileges.
W—Maintained as part of operating system security.
X—Privilege applies to access plans for precompiled applications.
Y—Maximum total characters 255.
Z—Comparable result can be achieved by assigning objects to different databases.
A1—For index only.
A2—Reorganization utility physically rearranges rows based on index.
A3—OS/2 workstation only.
A4—Single operation with Export; with operating system backup must backup each file.
A5—Can assign each user a database.
A6—Monitor users currently connected and corresponding databases.
A7—DOS workstation must incorporate function into an application program.
A9—Disables transaction rollback capability as well.

Table 4. Data Manipulation

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
SQL DML Keywords	2				
SELECT	1	▲	▲	▲	▲
INSERT	1	▲	▲	▲	▲
UPDATE	1	▲	▲	▲	▲
DELETE	1	▲	▲	▲	▲
FROM	1	▲	▲	▲	▲
WHERE	1	▲	▲	▲	▲
ORDER BY	1	▲	▲	▲	▲
GROUP BY	1	▲	▲	▲	▲
HAVING	1	▲	▲	▲	▲
INTO	1	▲	B2	▲	▲
UNION	1	▲	—	▲	▲

See Footnotes on page 225

Table 4. Data Manipulation (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Miscellaneous SQL Capabilities/Extensions	2				
Maximum levels of nested queries	0	15	255	255	255
Corellated subqueries	1	B3	▲	▲	▲
Subquery can return multiple values	2	—	—	▲	—
Update column to subquery result	2	—	▲	▲	—
Update using join	1	—	▲	—	—
Delete using join	1	—	▲	—	—
Insert using query	2	▲	▲	▲	▲
Insert using values	1	▲	▲	▲	▲
Insert with subquery in WHERE clause	1	▲	▲	▲	▲
Update with subquery in WHERE clause	1	▲	▲	▲	▲
Delete with subquery in WHERE clause	1	▲	▲	▲	▲
Tree-structured data (parts explosion)	1	—	—	▲	—
Table aliases	2	▲	▲	▲	▲
Column aliases	1	—	▲	▲	—
Group by expressions	2	—	▲	▲	B4
Order by expressions	2	B4	▲	▲	▲
Sort ascending and descending	1	▲	▲	▲	▲
Sequence generator	2	—	—	▲	—
Join Operations	2				
Maximum tables per operation	0	15	16	255	20
Maximum tables per outer join	1	0	16	1	1
Theta-join	3	▲	▲	▲	▲
Equijoin	3	▲	▲	▲	▲
Full outer theta-join	1	—	—	—	—
Left/right outer theta-join	1	—	▲	▲	▲
Full outer equijoin	1	—	—	—	—
Left/right outer equijoin	1	—	▲	▲	▲
Join table to itself	2	▲	▲	▲	▲
Join on nonindexed fields	2	▲	▲	▲	▲
Set Functions	2				
Distinct	2	▲	▲	▲	▲
Count	2	▲	▲	▲	▲
Minimum	2	▲	▲	▲	▲
Maximum	2	▲	▲	▲	▲
Sum	2	▲	▲	▲	▲
Average	2	▲	▲	▲	▲

See Footnotes on page 225

Table 4. Data Manipulation (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Standard deviation	1	—	—	▲	▲
Variance	1	—	—	▲	—
Boolean Functions	2				
BETWEEN	3	▲	▲	▲	▲
EXISTS	3	▲	▲	▲	▲
IN	3	▲	▲	▲	▲
IS NULL	3	▲	▲	▲	▲
IS NOT NULL	3	▲	▲	▲	▲
Soundex match	1	—	▲	▲	—
Wildcard match (single character)	1	▲	▲	—	▲
Wildcard match (variable length)	1	▲	▲	▲	▲
Wildcard match (variable length including 0)	1	▲	▲	▲	▲
AND/OR/NOT	2	▲	▲	▲	▲
Arithmetic Operators and Functions	1				
Modulo	2	—	▲	▲	▲
Exponentiation	2	—	▲	▲	▲
Square root	2	—	▲	▲	▲
Absolute value	2	—	▲	▲	▲
Ceiling	2	—	▲	▲	▲
Floor	2	—	▲	▲	▲
Rounding	2	—	▲	▲	▲
Sign	2	—	▲	▲	—
Truncate	1	—	▲	▲	—
Round/truncate to negative number of decimals	2	—	B5	▲	B5
Factorial	1	—	—	—	▲
Pi	1	—	▲	—	▲
Natural log	1	—	▲	—	▲
Log base 10	1	—	▲	—	▲
Trigonometric functions	1	—	▲	—	▲
Financial Functions	1				
Payments for amortization	1	—	—	—	▲
Payment amount for amortization	1	—	—	—	▲
Interest amount for payment period	1	—	—	—	—
Compounded balance of one-time investment	1	—	—	—	—
Compounded term of onetime investment	1	—	—	—	▲
Future value of annuity	1	—	—	—	▲
Compounded term with payments	1	—	—	—	—

See Footnotes on page 225

Table 4. Data Manipulation (Continued)

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Annual interest rate	1	—	—	—	▲
Internal rate of return	1	—	—	—	—
Modified internal rate of return	1	—	—	—	—
Present value	1	—	—	—	▲
Net present value	1	—	—	—	—
Double-declining depreciation	1	—	—	—	—
Straight-line depreciation	1	—	—	—	▲
Sum-of-year's-digits depreciation	1	—	—	—	▲
String Operators and Functions 1					
Concatenation	2	—	▲	▲	▲
Repeat string	1	—	▲	B6	▲
Substring	3	▲	▲	▲	▲
Find substring	2	—	▲	▲	▲
String length	2	▲	▲	▲	▲
Convert to uppercase	1	—	▲	▲	▲
Convert to lowercase	1	—	▲	▲	▲
Convert to lowercase with initial caps	1	—	—	▲	▲
Trim leading blanks	2	—	▲	▲	B7
Trim trailing blanks	2	—	▲	▲	B7
Date and Time Functions 1					
System time	3	▲	▲	▲	▲
System date	3	▲	▲	▲	▲
Time zone conversion	1	—	—	▲	▲
Day interval	2	—	▲	▲	▲
Month interval	2	—	▲	▲	—
Time interval	2	—	▲	—	—
Add/subtract days	2	—	▲	▲	▲
Add/subtract months	2	—	▲	▲	—
First of month	1	—	▲	▲	▲
First of quarter	1	—	▲	—	▲
First of year	1	—	▲	▲	▲
Extract day/month/year from date	2	—	▲	▲	▲
Extract year including century	1	—	▲	▲	▲
Extract year without century	1	—	▲	▲	▲
Extract day of week from date	1	—	▲	▲	▲
Extract month name from date	1	—	▲	▲	—
Extract two date parts in single function	1	—	—	▲	▲
Extract hour/minute/second from time	2	—	▲	▲	▲
Extract two time parts in single function	1	—	—	▲	▲

See Footnotes on page 225

Table 4. Data Manipulation (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Miscellaneous Functions	1				
IF/NULL/THEN function	2	—	▲	▲	▲
IF/BOOLEAN/THEN value ELSE other value	4	—	—	B8	C1
Convert string to number	2	—	▲	▲	▲
Convert number to string	2	—	▲	▲	▲
Convert number to string in user-specified format	1	—	—	▲	C2
Convert character to integer ASCII value	1	—	▲	▲	—
Convert integer ASCII value to character	1	—	▲	▲	▲
Convert date to string using user-specified format	1	B9	B9	▲	▲
Convert string in user-specified format to date	1	B9	B9	▲	B9
Function nesting	1	▲	▲	▲	▲
Triggered Procedures	3				
Triggered integrity procedures	1	—	▲	—	—
General triggered procedures	2	—	▲	—	—
Trigger SQL statement (DML)	2	—	▲	—	—
Trigger SQL statement (DDL/DCL)	1	—	C3	—	—
Trigger multiple SQL statements	3	—	▲	—	—
Trigger may include procedural logic	2	—	▲	—	—
Trigger separate stored procedure	2	—	▲	—	—
Trigger external program	1	—	—	—	—
After-insert trigger	1	—	▲	—	—
After-delete trigger	1	—	▲	—	—
After-update trigger	1	—	▲	—	—
After-update based on specific column	1	—	▲	—	—
Commit trigger	1	—	—	—	—
Multiple triggers on single table	2	—	▲	—	—
Multiple update triggers on single table	2	—	C4	—	—
Trigger recognizes modified rows	2	—	▲	—	—
Cascading triggers	2	—	—	—	—
Iterative trigger	1	—	—	—	—
Procedural Language Extensions	3				
IF/THEN	1	—	▲	—	—
IF/THEN/ELSE	2	—	▲	—	—
WHILE loops	2	—	▲	—	—

See Footnotes on page 225

Table 4. Data Manipulation (Continued)

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
REPEAT/UNTIL or DO/WHILE loops	1	—	—	—	—
FOR/NEXT loops	1	—	—	—	—
Case statements	1	—	—	—	—
Branching (GOTO)	1	—	▲	—	—
Integer variables	1	—	▲	—	—
String variables	1	—	▲	—	—
Floating point variables	1	—	▲	—	—
Date variables	1	—	▲	—	—
Time variables	1	—	▲	—	—
Arrays	2	—	—	—	—
Select into temporary variable	2	—	▲	—	—
SQL statements reference variables	2	—	▲	—	—
Modify variables using procedural logic	2	—	▲	—	—
IF/WHILE with subquery	2	—	▲	—	—

B2—Result achieved through a different syntax; SELECT INTO performs a different function.

B3—In WHERE and HAVING clauses only.

B4—Can refer to a column number in a select list, which can be an expression.

B5—Round only.

B6—Can pad either end of a string with a repeating substring.

B7—Single trim function trims both leading and trailing blanks.

B8—Boolean IF only; multiple conditions in single expression.

B9—Limited to a selection of vendor-defined patterns.

C1—Can evaluate only whether a number is zero or nonzero.

C2—Can specify number of decimals.

C3—Subject to stringent limitations.

C4—Single triggered procedure can incorporate functionality of multiple triggers.

Table 5. Data Integrity

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Concurrency Control	2				
Row locking	2	▲	—	▲	—
Page locking	1	—	▲	—	▲
Table locking	1	▲	▲	▲	▲
Shared row/page lock	1	▲	▲	—	▲
Exclusive row/page lock	1	▲	▲	▲	▲
Shared table lock	1	▲	▲	▲	▲
Exclusive table lock	2	▲	▲	▲	▲
Explicit exclusive table lock	1	▲	—	▲	
Explicit shared table lock	1	▲	—	▲	—
Implicit shared row/page lock	2	▲	▲	—	▲
Implicit exclusive row/page lock	2	▲	▲	▲	▲
Implicit exclusive lock on Select	4	C8	—	▲	—

See Footnotes on page 226

Table 5. Data Integrity (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Locks shared across cursors	3	▲	—	▲	▲
Repeatable read	2	▲	▲	▲	▲
Cursor stability	2	▲	▲	—	▲
Uncommitted read	1	▲	—	▲	—
Snapshot read	2	—	C6	▲	▲
Prevent DDL lock	1	▲	—	▲	—
Simultaneous table lock limit	0	U	50K	U	32,767
Simultaneous row/page lock limit	0	U	50K	U	32,767
List locks and owners	1	—	▲	▲	—
Lock list maintained in database table	1	—	▲	—	—
Deadlock detection	1	▲	▲	▲	▲
Specify deadlock detect interval	1	▲	—	—	—
Livelock detection	2	▲	▲	—	—
Optional wait/no wait for locks	1	—	—	—	—
Specify waiting period	1	—	C7	—	—
Automatic update of ROWID	1	—	C5	—	▲
Recovery	1				
Autocommit	2	—	▲	▲	▲
Autocommit on/off	1	—	—	▲	▲
Transactions (BEGIN, COMMIT, ROLLBACK)	5	▲	▲	▲	▲
Statement-level rollback	2	▲	▲	▲	—
Transaction log on/off	1	—	—	—	▲
Bypass log on delete	2	—	▲	—	—
Bypass log on insert	1	—	▲	—	—
Roll forward from archive log	2	—	▲	▲	▲
Archiving on/off	1	—	▲	▲	D1
Checkpointing on/off	1	▲	—	—	▲
Checkpointing on/archiving off	2	▲	▲	▲	—
Archive log separate from database	1	—	▲	▲	▲
Backup database and log together	1	D2	▲	—	▲
Backup database and log separately	1	D2	▲	C9	▲
Clear log upon backup	1	D2	▲	C9	—
Commit without closing cursors	2	▲	▲	▲	▲
Cursor stability across Commits	1	▲	—	—	▲
DDL/DCL in multistatement transaction	1	▲	—	—	▲

C5—Automatic update of timestamp column, which must be explicitly defined in table.

C6—Single select statement only.

C7—Library functions available for application programs.

U—Unlimited.

C9—Archive log backup and reuse is automatic when archiving is on.

D1—Disables transaction rollback capability as well.

D2—Not applicable; no need to back up log.

Table 6. Programming and Other Tools

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Interactive SQL	2				
Store/retrieve command in OS file	2	—	▲	D3	D5
Store/retrieve command in database table	2	D4	—	—	D6
Edit retrieved command before executing	1	D4	▲	D3	D5
Execute batch of commands from file/table	1	D4,D8	▲	▲	—
Enter/edit/execute batch of commands online	1	—	▲	—	D5
Command batch can include procedural logic	1	D4,D8	▲	—	—
Execute stored procedures	1	D4,D8	▲	—	D7
Enter/edit/save stored procedure online	1	D4,D7	▲	—	—
Store commands with runtime variables	1	D4	—	▲	▲
Runtime variable, repeated use	1	D4	—	▲	—
Modify/manipulate variables during processing	1	D4,D8	▲	—	—
List table columns and attributes	2	D4	▲	D3	—
Spool/save result to ASCII file	2	D4	▲	▲	▲
Print/spool query result	2	D4	▲	▲	▲
Specify column header for query results	1	D4	▲	D3	▲
Specify column width for query results	1	D4	▲	D3	▲
Specify numeric format for result column	1	D4	—	D3	▲
Specify date format for result column	1	D4	▲	D3	▲
Time operation	1	—	▲	—	▲
Scroll query results	2	D4	▲	—	—
Scroll left and right	2	D4	▲	—	—
Execute operating system command	2	—	▲	▲	
Set-level insert	2	D4	▲	▲	▲
Set-level delete	2	D4	▲	▲	▲
Set-level update	2	D4	▲	▲	▲
Prompted select/insert/update/delete	2	D4	—	—	—
Prompted table definition	2	D4	—	D4	—
Prompted user/security maintenance	2	D4	▲	D4	D5
Prompted backup/restore	2	D4	▲	D4	D5
Prompted startup/shutdown	2	D4	▲	D4	D5

See Footnotes on page 230

Table 6. Programming and Other Tools (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Load/Unload Utility	1				
Load appending to existing records	2	E4	▲	▲	▲
Load replacing existing records	1	E4	—	▲	—
Load updating existing records	1	—	—	—	—
Load fixed-length ASCII file	2	E4	▲	▲	—
Unload fixed-length ASCII file	2	—	▲	D9	D9
Load delimited ASCII file	2	E4	▲	▲	▲
Unload delimited ASCII file	2	E4	▲	D9	▲
Unload several tables to single file	1	—	—	E1	—
Load selected tables from file	1	—	—	E1	—
Load/unload in native format	1	E4	▲	▲	—
Load/unload in native format with Create Table	2	E4	—	▲	—
Load/unload in native format with Create Index	1	E4	—	▲	—
Load/unload ASCII file with Create Table	2	—	—	—	▲
Load/unload ASCII file with Create Index	1	—	—	—	—
Input column order different from table	2	E4	▲	▲	—
Output column order different from table	2	—	▲	▲	—
Input date field in user-specified format	1	E4	E3	▲	E3
Output date field in user-specified format	1	E4,E5	—	D9	D9
Output numeric field with user-specified precision	1	—	—	D9	D9
Input numeric field with implied decimal	1	—	—	—	—
Output numeric field with implied decimal	1	—	—	E2	E2
Load column based on expression	1	—	—	—	—
Save load file format to OS file	1	—	▲	▲	—
Save load file format to database table	1	E4	—	—	—
Save unload file format to OS file	1	—	▲	D9	D9
Save unload file format to database table	1	E4	—	—	—
Prompt user for file format	1	—	▲	—	—
Save responses to prompts in format file/table	1	—	▲	—	—
Load and edit format file/table	1	—	—	—	—

See Footnotes on page 230

Table 6. Programming and Other Tools (Continued)

	Weight	IBM OS/2 Extend- ed Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Stored Procedures	4				
Store as database object	2	▲	▲	—	▲
Store executable file on server	2	▲	—	—	—
Retrieve and edit procedure definition	1	▲	▲	—	—
Execute procedure without retrieving	4	▲	▲	—	E8
Explicit execution (nontriggered)	1	▲	▲	—	E8
Maximum levels of nested procedures	2	U	255	0	0
Values passed from calling pro- gram/procedure	2	▲	▲	—	▲
Values returned to calling pro- gram/procedure	2	▲	E6	—	▲
Procedures use local variables	1	▲	▲	—	—
Select into local variable	1	▲	▲	—	—
Arrays	2	—	—	—	—
Procedural language extensions	2	▲	▲	—	—
Multiple SQL commands	5	▲	▲	—	E9
SQL DML commands	2	▲	▲	—	E9
SQL DDL/DCL commands	1	▲	E7	—	—
SQL cursors	1	▲	—	—	—
Fetch next	1	▲	—	—	—
Fetch previous	1	—	—	—	—
Fetch ahead n rows	1	—	—	—	—
Host Language Interface	7				
C	2	▲	▲	▲	▲
Cobol	2	▲	—	▲	—
Fortran	1	▲	—	▲	—
PL/1	1	—	—	F1	—
BASIC	1	—	▲	—	—
Pascal	1	▲	—	F1	—
Embedded SQL	2	▲	—	▲	—
Precompiled interface	2	G2	—	▲	—
Direct call interface	1	—	▲	▲	▲
Bind prior to runtime	1	G2	F3	—	F8
Dynamic recompilation	1	—	F3	—	—
Preserve context	1	▲	—	▲	▲
Interpreted interface	1	▲	—	—	—
Dynamic SQL	1	▲	▲	▲	▲
SQL cursors	2	▲	F4	▲	▲
Fetch next	2	▲	▲	▲	▲
Fetch previous	2	—	G1	—	F9
Fetch ahead n rows	1	—	G1	—	G1
Select into array	2	—	F5	▲	—

See Footnotes on page 230

Table 6. Programming and Other Tools (Continued)

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Fetch cursor into array	2	—	—	▲	—
Insert using array	2	—	—	▲	—
Update using array	2	—	—	▲	—
Update where current	2	▲	F6	▲	▲
Update joined cursor where current	1	—	—	▲	—
Delete where current	2	▲	F6	▲	▲
Automatic ROWID validation	1	—	F7	—	▲
Whenever error processing	1	▲	—	▲	—
Whenever Not Found processing	1	▲	—	▲	—
Bulk execute	2	—	▲	▲	▲
Bulk execute with separate processing	1	—	▲	—	▲
Background execution	1	—	▲	—	—
Back-end result sets	2	—	G3	—	▲
Front-end result sets	1	—	F5	F2	▲
SQL DDL/DCL commands	1	▲	▲	▲	▲
Miscellaneous Front-End Support	1				
Forms generator	2	G6	—	G4	—
Report generator	2	G6	—	G4	G4
Windows/Presentation Manager interface	2	G6	G5	G6,G9	G8
Windows-based application generator	2	—	G7	—	G4
dBase compiler/interpreter	1	—	—	G4	—
Lotus 1-2-3 interface	1	—	G4	G4	—
Microsoft Excel interface	1	—	G4	—	—
Other spreadsheet interface	1	—	—	G4	—

D3—Available with vendor-provided front-end tools sold separately.

D4—OS/2 workstation only.

D5—DOS workstation only.

D6—Can execute directly; cannot retrieve.

D7—Limited to single command.

D8—Can store single commands and store procedure to execute those commands.

D9—Obtained using interactive SQL and spooling results to a file.

E1—Native format only.

E2—Obtained by selecting an expression and formatting output as integer.

E3—Converts automatically in one of several date formats prescribed by vendor.

E4—Available for OS/2; DOS workstations must use an application program to call function.

E5—Must use default date format, which can be changed.

E6—Value can be returned to a host language program that uses DB-Library, but not to another stored procedure.

E7—Subject to stringent limitations.

E8—Only in interactive SQL; application program must retrieve and then execute.

E9—Application program may chain several stored commands on retrieval; Select may not be part of chain.

F1—Available in other server environments.

F2—Can select into arrays.

F3—Executes stored procedures, which are dynamically recompiled.

F4—Does not support standard SQL cursors, but offers alternate functionality to fetch query results.

F5—Can select into a row buffer.

F6—System can generate Where clause based on row's primary key values.

F7—Automatic time stamp validation of tables created with time stamp column.

F8—Single command can be stored in optimized form, retrieved at runtime.

F9—Single-table cursors only.

G1—Fetch row in explicit position in result set.

G2—DOS applications must be precompiled on an OS/2 system.

G3—Can select into a temporary table.

G4—Available from vendor as a separate purchase.

G5—Available from third-party vendor.

G6—OS/2 workstations only.

G7—Can develop applications to run under Windows.

G8—DOS workstations only.

G9—Interactive SQL not included.

U—Unlimited.

Table 7. Portability and Network Capability

	Weight	IBM OS/2 Extended Edition	Microsoft SQL Server	Oracle Server for OS/2	Gupta SQLBase Server
Server Environments Supporting DOS/OS/2 Front-Ends	1				
VAX	1	—	▲	▲	—
MVS	1	—	—	▲	—
VM	1	—	—	▲	—
MS/DOS	1	—	—	H1	▲
UNIX/Xenix	1	—	▲	▲	—
Network Support	1				
Novell	2	—	H2	▲	▲
Microsoft	1	—	▲	▲	▲
3Com	1	—	▲	▲	▲
IBM	1	▲	▲	▲	▲
Banyan VINES	1	—	—	▲	—
Token-Ring	2	▲	▲	▲	▲
Ethernet	2	H3	▲	▲	▲
NetBIOS	2	H4	—	▲	▲
Named Pipes	2	—	▲	▲	—
SPX/IPX	2	—	—	▲	—
TCP/IP	1	—	▲	▲	—
Database server and file server in same system	1	▲	▲	▲	▲
Dedicated database server	1	▲	▲	▲	▲
Distributed Processing	1				
Access databases on multiple stations	1	—	▲	▲	▲
Join tables from multiple stations, single query	2	—	—	▲	—
Insert/update on multiple stations, same transaction	2	—	▲	—	—
Two-phased Commit	2	—	▲	—	—
Link with database server under another OS	1	—	H5	▲	▲
Link with other vendors' OS/2 servers	1	—	—	—	▲
Link with DB2	1	—	H5	—	▲

H1—Single-user version available.
H2—Works with Novell Named Pipes.
H3—OS/2 workstations only.
H4—DOS workstations only.
H5—Available from third-party.

Software Prices

	Price (\$)
SQL Servers for OS/2	
IBM OS/2 Extended Edition	830
Microsoft SQL Server	2,495
Oracle Server for OS/2	2,499
Gupta SQLBase Server	2,495

Glossary of Terms

Data Definition Features

Program Parameters

Maximum Records per Table: The maximum number of records that can be entered into a single table (a collection of related records).

Maximum Columns per Record: The maximum number of columns (a storage location in a record consisting of a single data item and identified by a column name) that can be contained in a single record.

Maximum Characters per Record: The maximum number of characters (text or numeric) that can be contained in a single record.

Maximum Characters per Alphanumeric Field: The maximum number of characters that can be contained in a single alphanumeric field.

Maximum Tables per Database: The maximum number of tables that can be constructed and managed by the program.

Memory Required (server): The amount of memory required to run the database server software.

Memory Recommended (server): The amount of memory recommended by the vendor to achieve what it considers an acceptable performance level.

Memory Required (OS/2 workstation): The amount of memory required to interface with the database server using a workstation running OS/2.

Memory Required (DOS workstation): The amount of memory required to interface with the database server using a workstation running DOS.

Maximum Independent Databases per Server: The maximum number of independent databases a single server can manage concurrently. (An independent database is one that can be used independently of any other database.)

Maximum Users per Database: The maximum number of users who can be connected to a database concurrently.

SQL DDL Keywords

CREATE TABLE: The program supports the SQL CREATE TABLE command.

CREATE TABLE Using Query: The program allows the creation of a table based on the results of a query.

DROP TABLE: The program supports the SQL DROP TABLE command.

ALTER TABLE: The program supports the SQL ALTER TABLE command.

ALTER TABLE ADD COLUMN: The ALTER TABLE command includes the ability to add a column to a table.

ALTER TABLE MODIFY COLUMN: The ALTER TABLE command includes the ability to modify column attributes.

ALTER TABLE DROP COLUMN: The ALTER TABLE command includes the ability to drop a column from a table.

CREATE INDEX: The program supports the SQL CREATE INDEX command.

DROP INDEX: The program supports the SQL DROP INDEX command.

CREATE VIEW: The program supports the SQL CREATE VIEW command.

DROP VIEW: The program supports the SQL DROP VIEW command.

CREATE SYNONYM: The program supports the creation of synonyms for database objects (tables, views, etc.)

CREATE PUBLIC SYNONYM: The program supports the creation of synonyms for database objects, and these synonyms can be referenced by any user having rights to the object.

DROP SYNONYM: The program supports the dropping of synonyms previously created.

Define Object as Temporary: The program allows an object to be defined as temporary; temporary objects are automatically dropped at the end of the current session.

Rename Column: The program supports renaming a column in a table.

Rename Table: The program supports renaming a table.

DOMAINS

Boolean: The user can define a field that accepts only logical data values (e.g., True/False, 1/0, Yes/No).

Integer: The user can define a field that accepts only numeric data values that are integers (e.g., 1, 5, 100,000).

Short Integer: The user can define a field that accepts only relatively small numeric data values that are integers.

Floating Point: The user can enter values with a variable number of places to the right of the decimal, and the program will accept and maintain the decimal places. For example, 0.003 and 0.046123 would both be acceptable values.

Fixed Point or Money: The user can specify that a certain number of digits to the right of the decimal point be defined in a given numeric field. The program must be able to align entries vertically on the decimal point.

Fixed Length Character: The user can define a field that accepts alphanumeric character strings as data values and stores them in fixed length storage areas. For example, the values "Ed" and "King Edward the Sixth" will occupy the same amount of storage; therefore, it is best used when all values are of the same length, such as two-character state codes.

Variable Length Character: The user can define a field that accepts alphanumeric character strings as data values and stores them in the amount of space necessary to accommodate their length. For example, the value "Ed" will occupy less storage than the value "King Edward the Sixth."

Name: The user can define an alphanumeric field that treats the last group of characters as the "last name" (for sorting or selecting).

Date: The program can store data items containing calendar dates (i.e., mm/dd/yy) and can sort them by date in a date format.

Time: The program can store data items that contain times of the day (i.e., 11:20:59) and sort them by time in a time format.

Fraction of Second: The time domain is of sufficient precision to store fractions of seconds.

Date and Time in Same Variable: A date and a time can be stored in a single variable.

Long Text: A special domain is defined for storing text data that is too long for an ordinary character variable.

Long Binary: Similar to long text, except that binary data can be stored.

User-Defined Domains: New domains can be created and then used to define attributes. For example, a color domain might contain the values red, black, blue, or green.

User-Defined Composite Domains: New domains can be created from a composite of values on existing domains. For example, a date domain is a composite of three integer values: month, day, and year.

Specify Order of New Domains: The user can specify an ordering predicate indicating how new domains should be sorted. For example, a date domain should be sorted by year, then by month, then by day.

Specify Conversion Rules: The user can specify a procedure for conversion between domains. For example, a length value (in inches) can be converted to length in centimeters by multiplying the number of inches by 2.54.

Null Value Distinct from Domain Values: A uniform null value is common to all domains and distinct from the values of domains. A null value is not a zero value; a null value specifies that the field is empty.

Domain Integrity

Range Check: A domain can be defined as a range of values within some existing domain. For example, a month domain contains the integers x such that $1 [Lz] x [Lz] 12$.

Set Check: A domain can be defined as a set of values. For example, a color domain is the set {Red, Black, Blue, Green}.

Pattern Check: A domain can be defined as strings of text that match a pattern string. For example, a phone number domain is the set of strings that match $\# \# \# - \# \# \# - \# \# \# \#$ where # matches any number.

Formula Check: A domain can be defined as the set of all values that cause some predicate to be true. For example, an integer domain could be defined as the set of floating point values x such that $\text{TRUNCATE}(x) = x$.

Procedure Check: Like a formula check, but the predicate is not limited to a formula; it may instead be some arbitrary procedure. For example, to check that a domain's values are all prime numbers, a procedure would have to be written rather than a function or a formula.

Date Checking: The program maintains the consistency of values defined on a date domain.

Time Checking: The program maintains the consistency of values defined on a time domain.

Define Default Value: A default value can be specified that will be inserted into a column defined with this domain whenever a record has no value for that column.

Structural Integrity

Primary Key: The program maintains a designated primary key (or unique index) for each table as a unique identifier. The primary key is a designated column or set of columns within the table that is unique to each entry in the table, such as social security number, account number, or telephone number.

Foreign Key: The program maintains designated foreign keys as attributes of tables in the data dictionary. A foreign key is a column or set of columns in one table whose nonnull values must also exist in the primary key columns of another table.

Alternate Key: The program maintains designated alternate keys as attributes of tables in the data dictionary. When a table has more than one unique key, one is designated the primary key and the other an alternate key. For example, if both social security numbers and telephone numbers are unique (i.e., no two persons have the same social security or telephone number), one field is designated the primary key and the other field as the alternate key.

Foreign Key May Reference Alternate Key: A foreign key may reference either the primary key or the alternate key in another table; it does not have to reference the primary key.

Multiattribute Primary Key: A primary key may be a composite of more than one attribute or column. For example, in a shipment table, the primary key may consist of the shipment number and the part number because both numbers are required to properly identify a shipment.

Multiattribute Foreign Keys: A foreign key may be a composite of more than one attribute or column.

Enforce Uniqueness of Primary Key: The program maintains the uniqueness of primary keys by rejecting or compensating for illegal operations.

Enforce Nonnull in Primary Key: The program will not accept null values in primary key fields.

Referential Integrity

Referential Integrity Enforced by Data Dictionary: The program imposes restrictions on columns based on relationships defined in the data dictionary in order to maintain legitimate referential relationships. Foreign key

columns, if not null, must contain values that exist in the primary key columns of its target relation.

Referential Integrity Enforced by Triggers: The program enforces referential integrity by automatically executing a procedure triggered whenever a foreign key value is inserted or updated, or whenever a referenced primary key value is deleted or updated. Referential relationships are defined in the trigger rather than in the data dictionary (the user must write the trigger).

Cascaded Deletions: Deleting a referenced record in one table causes the immediate deletion of all records (in other tables) that reference it.

Cascaded Nulls upon Deletion: A delete operation in a key column inserts null values in the foreign key column of all records (in other tables) that reference it.

Forbid Deletion of Referenced Record: Deleting referenced records is not allowed.

Cascaded Updates (referential): Updating a referenced primary key will immediately cause a corresponding change in foreign key columns (in other tables) that reference it.

Cascaded Nulls upon Update: Updating a referenced primary key immediately causes nulls to be placed in the foreign key columns (in other tables) that reference it.

Forbid Updates (referential): Updating a referenced record is not allowed.

Other Integrity Rules

Not Null: Columns can be defined as "not null," such that they will not accept null values, and the program enforces this restriction.

Unique: Columns can be defined as "unique," such that no two values are the same, and the program enforces this restriction. (Note if accomplished by unique index.)

Range Check: The program can restrict allowable field values to a specified range.

Predicate Check: The program can restrict allowable field values to those that cause a predicate to be true.

Immediate Record Transition Constraints: Integrity constraints can be specified involving the transition of records from one value to another. For example, Old:status New:status prevents a decrease in status.

Deferred Status Constraints: Integrity constraints are not enforced until a transaction is committed. For example, a constraint can ensure that each transaction leaves an accounts table balanced; otherwise, the transaction is canceled.

Set Validation: A column can be defined as a specific set of values. For example, a color column is the set {Red, Black, Blue, Green}.

Formula Validation: A column can be defined as the set of all values that cause some predicate to be true. For example, an integer attribute could be defined as the set of floating point values x such that $\text{TRUNCATE}(x) = x$.

Pattern Validation: A column can be defined as character strings that match a pattern string. For example, a phone number column is the set of strings that match $\# \# \# - \# \# \# - \# \# \# \# \#$ where $\#$ matches any number.

Define Default Value: A default value can be specified for a column, and that value is inserted whenever a record has no value for that column.

System Catalog

Query System Catalog: Information in the data dictionary (names of tables, columns, etc.) can be accessed through normal SQL commands.

System Catalog Maintained Automatically: Changes to the database structure and to actual data automatically update information in the appropriate data dictionary tables.

Database-Wide Information: The data dictionary includes a database table containing information regarding the entire database, not just objects belonging to specific users.

User-Specific Information: An individual user can access information on objects belonging to him/herself without specifying that restriction in a WHERE clause.

Tables: A data dictionary table contains information about specific tables.

Columns: A data dictionary table contains information about columns in tables.

Indexes: A data dictionary table contains information about indexes.

View Definitions: A data dictionary table contains the text used to define views.

View Dependencies: A data dictionary table specifies the tables on which a view is based.

Procedure Definitions: A data dictionary table contains the text used to define stored procedures. This information is maintained automatically when stored procedures are created or updated.

Procedure Dependencies: A data dictionary table specifies the tables affected by a stored procedure and the way in which they are affected. This information is maintained automatically when stored procedures are created or updated.

Space Allocation by Object: A user can query a table to determine the amount of space allocated to a specific object.

Space Allocation for Database: A user can query a table to determine the amount of database space already allocated.

User Information: A data dictionary table contains information about users who have access to the database.

User Authorizations: A data dictionary table indicates each user's privileges. For example, it would indicate whether user Joe has the privilege to create tables.

User Authorizations by Object: A data dictionary table indicates each user's privileges on specific objects. For example, it would indicate the privileges user Joe has on table EMPLOYEES.

Physical Device Assignments: A data dictionary table indicates what physical files are assigned to a database and their purpose. For example, it may indicate that data for the database TEST is stored in C:DATATEST.DBS and that the transaction log for the database TEST is D:LOGSTEST.LOG.

Locks: A data dictionary table contains information on each lock currently in effect.

Views

Restriction Views: Views can be created as subsets of the records in another table or view. In SQL, a view is a table that does not exist, but is derived from one or more other tables.

Projection Views: Views can be created as subsets of the columns in another table or view. Projection views are also called column subset views.

Join Views: Views can be created as the join of two or more tables.

Views with Calculated Fields: Views can include calculated fields.

Grouped Views: Views can be created based on a query containing a GROUP BY clause. Any aggregate

functions defined in the GROUP BY clause appear in the view as numeric columns.

Views with Check Option: A view can prevent inserts or updates that do not meet the view's original selection criteria. For example, if a view was defined as containing only those employees with states equal to CA, OR, WA, or NV, an employee from state TX cannot be inserted.

Delete from Restriction View: The program allows deletions on restricted views. Deleting records from the view causes the deletion of corresponding records from the base table.

Insert into Restriction View: The program allows insertions on restricted views of records. Inserting records into a view causes the insertion of corresponding records into the base table.

Update Restriction View: The program allows modification of records in a restricted view. Updating records causes corresponding changes in records in the base table.

Insert into Projection View: Records can be inserted through a key-preserving view (i.e., one that includes the key from the base table). Columns in the base table that are not included in the view will be set to null.

Delete from Projection View: Records can be deleted through a key-preserving view. Columns in the base table that are not included in the view will also be set to null.

Update Projection View: Records can be updated through a key-preserving view. Columns in the base table that are not included in the view will be set to null.

Update Joined View: Records can be updated through a view that is the virtual join (on their primary keys) of two or more tables.

Insert into View Defined with Subquery: Records can be inserted through a view that has been defined with a subquery.

Delete from View Defined with Subquery: Records can be deleted through a view that has been defined with a subquery.

Update View Defined with Subquery: Records can be updated through a view that has been defined with a subquery.

Query Grouped View Using WHERE: A grouped view column defined using an aggregate function can be referenced in a WHERE clause. In reality, the query uses the WHERE clause against the view to evaluate a HAVING condition in the base table, but the program does the necessary translation.

Query Grouped View Using Aggregate Functions: A grouped view column defined using an aggregate function can in turn be used as the argument of an aggregate function in a query against the view. For example, a query can contain the expression AVG(MAXPAY), where MAXPAY was defined in the view as MAX(PAY) for each group.

Security/Database Administration

Security and Data Control Language

Incorporates Network Security: Network user IDs, passwords, and security levels are recognized by the database management program.

Change Administrator Password: The password for the default system administrator ID can be changed after installation.

Users Change Own Passwords: A user can change his own password, rather than the database administrator having to do so by means of a Grant.

Grant Privileges: The creator or owner of an object (e.g., a table) has the ability to grant specific rights to that object to other users.

NOTE: For this and subsequent items in this section, if a privilege can be granted, it is by default withheld from any user (except a database administrator) to whom it has not been explicitly granted.

Grant with Grant Option: A user can be granted privileges with the ability to grant them in turn to another user.

Grant without Grant Option: A user can be granted privileges without the ability to grant them to another user.

Connect Privilege: A database administrator can grant new users the ability to connect to the database, and that privilege can subsequently be revoked.

Create Database Privilege: A user can be granted the ability to create new databases.

System Administrator Privilege: A user can be granted all the privileges of the original (default) system administrator. These privileges generally include all rights to all tables belonging to all users, all rights to all system tables, and the ability to perform any function of which the system is capable.

Create Table Privilege: Users can be given the ability to create tables.

Privilege to Alter/Drop Existing Table: A user can be given the ability to alter or drop a table that he or she did not create.

Privilege to Create Index on Existing Table: A user can be given the ability to create an index on a table that he or she did not create.

Privilege to Drop Existing Index: A user can be given the ability to drop an index that he or she did not create.

Privilege to Create Trigger on Existing Table: A user can be given the ability to create a trigger on a table that he or she did not create.

Reference Privilege: A user can be given the ability to reference a specified table when setting up referential integrity constraints in another table.

Privilege to Create Stored Procedures: A user can be given the ability to create stored procedures.

Privilege to Execute Existing Procedure: A user can be given the ability to execute a procedure he or she did not create.

Privilege to Alter/Drop Existing Procedure: A user can be given the ability to modify or drop a procedure he or she did not create.

Select Privilege: A user can be granted the ability to select from a table.

Update Privilege on Table: A user can be granted (in a single grant) the ability to update all columns in a table.

Update Privilege on Specific Columns: A user can be granted the ability to update only specified columns in a table.

Insert Privilege: A user can be granted the ability to insert records into a table.

Delete Privilege: A user can be granted the ability to delete records from a table.

Grant All on Table: All rights to a table can be granted to a user in a single command.

Grant to Public: A privilege may be granted (by means of a single command) to all users in the system, including any users added subsequently.

Grant to User Group: Users may be assigned to groups, and privileges granted to a group. Group privileges apply to all users in the group, including those added subsequently. Users dropped from the group no longer possess the group's privileges.

Grant Select on View: A user can be granted the ability to select from a view.

Grant Update on View: A user can be granted the ability to update records through a view.

Grant Insert on View: A user can be granted the ability to insert records through a view.

Grant Delete on View: A user can be granted the ability to delete records through a view.

Privilege on View of Withheld Table: A user can be granted privileges on a view based on a table or tables to which he or she has no access. For example, a user may not have the ability to select from a table, but may have the ability to select from a view containing only certain columns from the table. Columns containing sensitive information remain concealed from that user.

Backup/Restore Privilege: A user can be given the ability to back up or restore the database.

Auditing

Table Audit: All use of all tables is recorded in an audit table.

User Audit: All user activity is recorded in an audit table.

Access Audit: Each access is recorded to the audit file.

Session Audit: One audit record is kept per session, and that audit record shows every access by every user on every table.

Audit Specified Commands: Only accesses using specified commands are recorded in the audit table.

Audit Successful/Unsuccessful Access: Recording an access attempt can be conditional on whether the access attempt was successful.

Physical Options

Maximum B-Tree (VSAM) Indexes: This is the maximum number of B-tree indexes that can be automatically maintained for a single table. A B-tree index is a multi-level or tree-structured index in which an index level provides sequential access to data records, and other levels provide direct access to the sequential index.

Maximum Multikey B-Tree Indexes: This is the maximum number of B-tree indexes using composite keys that can be automatically maintained per table.

Maximum Hash Indexes: This is the maximum number of hash-type indexes that can be automatically maintained per table. A hash index provides direct access to data records stored at computed "hash addresses."

Maximum Multikey Hash Indexes: This is the number of hash-type indexes (using composite keys) that can be automatically maintained per table.

Maximum Indexes per Table: This is the maximum indexes (of all types) allowed for a single table.

Maximum Columns per Index Key: This is the maximum columns that can make up a single composite index key.

Unique Indexes: An index will enforce uniqueness by preventing insertion of a duplicate key value or by preventing creation of the index if a duplicate key value already exists.

Nonunique Indexes: An index will allow duplicate key values.

Index on Expression: An index key may contain an expression based on a column, rather than an actual column. For example, the index key YEAR(HIREDATE) would use only the year portion of the date value.

Clusters: Records are physically grouped based on their key values.

Multitable Clusters: Records from several tables with the same key values are physically grouped. For example, if several tables contain data for specific employees, records from all those tables for a single employee will be physically grouped.

Database on Multiple Disk Drives: The operating system files used by a database can reside on different disk drives.

Assign Objects to Specific Files: In a database using multiple physical files, the user creating an object can specify (table, index, etc.) the file in which it is to be stored. Sometimes, users divide the database into partitions, associating each partition with a file, and assigning each object to a partition.

Transaction Log on Separate Drive: The file or files containing uncommitted transaction data can reside on a disk separate from the disk where the file or files containing permanent data are stored.

Define Minimum Contiguous Storage: The user can specify allocation of storage space to objects that need more space; a unit of that size is allocated exclusively to a single object. Keeping the data for a specific object in a contiguous area maximizes performance, but too large an area may waste space.

Define Minimum Contiguous Storage for Table: The user can specify different amounts of contiguous space for different tables. Thus small tables will not waste space, and large tables will be stored in large contiguous blocks.

Automatic Space Reclamation: When rows in a table are deleted, the space is automatically made available for other database objects; it is not necessary to drop and re-create the table or run a utility program.

Set Fill Factor: The user can specify what portion of a storage unit will be filled with data before more space is allocated. The remaining space is left free for subsequent updates. Thus, if a record is inserted with some columns left null, when data is added for those columns, it can be stored in physical proximity to the original data for the same record.

Database Administration

Start-Up/Shutdown from Remote Station: A user can enter commands at a workstation to start up and shut down a database.

Delayed Shutdown: The system can be set to prevent new logins and shut down as soon as the last user has logged out.

Immediate Shutdown: The system can be set to log out all users, roll back uncommitted transactions, and shut down the database.

Automatic File Creation: A database operation that requires creation of an operating system file, such as database expansion, will create the file automatically, rather than requiring separate commands to create the file and allocate it.

Automatic File Expansion: A file expands automatically as data is added to it, rather than having a full size allocated and gradually filling it.

Backup Single Table: A single table can be backed up. The backup includes the table definition, rows in the table, and any indexes associated with the table.

Backup Selected Tables: Several selected tables can be backed up in a single operation.

Backup User Schema: All database objects belonging to a specific user can be backed up in a single operation.

Backup Entire Database: The entire database can be backed up in a single operation.

Backup without Shutdown: The database can be backed up while work is in progress, but the backup will not include uncommitted changes.

Partial Shutdown/Backup/Restore: A portion of the database can be shut down and backed up or restored, while the rest of the database remains in use.

Restore Selected Portion of Backup: A selected portion of a previous backup can be restored (different from a full restore of a partial backup).

Incremental Backup: The system can update an earlier backup with changes made to the database since that backup. The incremental backup is done by comparing the backup and the current database or by using a log of database activity since the last backup.

Backup to/Restore from Workstation: The database can be backed up to, and restored from, files residing at the workstation.

Initiate Server Backup/Restore from Workstation: The command to back up the database to the server, or restore from the server, can be entered at the workstation.

Monitor Activity by Process: The administrator can monitor the activity of each process connected to the database.

Monitor Locks: The administrator can monitor the locks in use by each process.

Monitor Statistics: The administrator can monitor statistics pertaining to system use and performance.

Monitor from Workstation: The administrator can monitor database activity from a workstation.

Stop Specified Process: The administrator can enter a command to stop a specified process without interfering with any other processes. The stopped process is immediately logged off, and uncommitted work is rolled back.

Performance Tuning Features

Syntax-Independent Optimization: The system will attempt to find the most efficient procedure for accessing the data requested by a query, regardless of the syntax by which it is requested. For example, the query `SELECT COUNT(*) FROM A WHERE KEY IN (SELECT KEY FROM B)` requests the same data as the query `SELECT COUNT(*) FROM A, B WHERE A.KEY = B.KEY`, and the optimizer should exhibit comparable performance for either syntax.

Show Optimization Plan: The system will display the plan by which a command is to be optimized when executed.

Trace Feature: The system will record performance information about individual commands as they execute.

Set Checkpoint Interval: The user can set the amount of time between checkpoints. When checkpointing is in effect, rather than permanently writing changes to the

database every time there is a commit, changes are designated as permanent but held in the transaction log until a checkpoint. Checkpointing reduces the frequency of database writes.

Disable Archiving: Archiving records all transactions since the last database backup; when necessary, a backup can be restored and a roll forward applied of all subsequent transactions in the archive log. When archiving is disabled, transactions are deleted from the log once they have been permanently written to the database.

Set Total Memory Size: The user can specify the total amount of RAM to be used by the program. Any amount above that total is available for caching to reduce disk swapping.

Apportion Memory for Programs and Data: The user can specify the proportion of available memory to be used to maintain application programs, and the remainder available for data caching. Complex applications manipulating small amounts of data will need relatively large program caches, while simple applications manipulating large volumes of data need relatively large data caches.

Assign Memory for Parsed Statements: A specific amount of memory can be designated as necessary to store parsed SQL statements to prevent having to reparse them upon reexecution.

Assign Memory for Data Dictionary Information: A specific amount of memory can be designated to store data dictionary information to avoid having repeatedly to access this information.

Single Process with Multiple Threads: The server uses a single OS/2 process for workstation connections, with each connection using a thread in that process. The server may use a total of more than one process, but the number does not increase as workstations are added. Using a single process requires substantially less memory than having each connection use a separate OS/2 process, and memory is left free for caching and other purposes.

Data Manipulation

SQL DML Keywords

SELECT: The program supports the SQL SELECT command.

INSERT: The program supports the SQL INSERT command.

UPDATE: The program supports the SQL UPDATE command.

DELETE: The program supports the SQL DELETE command.

FROM: The program supports the SQL FROM keyword. The FROM clause lists the tables to be queried.

WHERE: The program supports the SQL WHERE keyword. The WHERE clause specifies the selection and join criteria for a statement.

ORDER BY: The program supports the SQL ORDER BY keywords.

GROUP BY: The program supports the SQL GROUP BY keywords.

HAVING: The program supports the SQL HAVING keyword. The HAVING clause specifies selection criteria for grouped items.

INTO: The program supports the SQL INTO keyword. SELECT INTO assigns a column value or expression to a variable.

UNION: The program supports the SQL UNION keyword. UNION combines the results of two queries.

Miscellaneous SQL Capabilities and Extensions

Maximum Levels of Nested Queries: The maximum depth (in levels) to which queries can be nested. A nested query, also called a subquery, is a query incorporated into another SQL statement. For example, the statement, SELECT NAME, ADDRESS FROM EMPLOYEES WHERE EMP—ID IN (SELECT EMP—ID FROM MANAGERS) employs a subquery. A subquery can in turn employ a subquery in its own WHERE clause.

Correlated Subqueries: A subquery's WHERE clause may reference information from the outer statement; for example, UPDATE SALESMEN SET TOTAL—SALES = (SELECT SUM (AMOUNT) FROM SALES WHERE SALES.SALESMAN—ID = SALESMEN.SALESMAN—ID).

Subquery Can Return Multiple Values: A single subquery may return more than one value; for example, SELECT NAME, ADDRESS FROM EMPLOYEES WHERE (LOCATION,PAY) IN (SELECT LOCATION, MAX(PAY) FROM EMPLOYEES GROUP BY LOCATION).

Update Column to Subquery Result: An update command may set a column to the result of a query. See example under Correlated Subqueries above.

Update Using Join: An update's SET clause may reference more than one table without resorting to a subquery; for example, UPDATE EMPLOYEES SET

TOTAL—PAY = SALARY * SHIFT—DIFF FROM EMPLOYEES,RATES WHERE EMPLOYEES.SHIFT = RATES.SHIFT.

Delete Using Join: Similar to Update using join, a statement uses a FROM clause that references more than one table to identify and delete records in a single table. For example, to delete all books by the author Jones: DELETE BOOKS FROM BOOKS,AUTHORS WHERE BOOKS.AUTHORID=AUTHORS.AUTHORID AND AUTHORS.NAME='JONES'.

Insert Using Query: An insert may employ a query; for example, INSERT INTO MANAGERS SELECT * FROM EMPLOYEES WHERE TITLE IN ('MANAGER','PRESIDENT').

Insert Using Values: An insert may employ a list of literal values; for example, INSERT INTO MANAGERS (ID, NAME, TITLE) VALUES (1, 'SMITH', 'PRESIDENT').

Insert with Subquery in WHERE Clause: An insert statement that incorporates a query may employ a subquery in its WHERE clause.

Update with Subquery in WHERE Clause: An update statement may employ a subquery in its WHERE clause.

Delete with Subquery in WHERE Clause: A delete statement may employ a subquery in its WHERE clause.

Tree-Structured Data (parts explosion): Program extensions allow queries of tree-structured data, such as a bill of materials database.

Table Aliases: Tables can be assigned "aliases" (or alternative names) by which they can be referred. Aliases are especially useful when joining a table with itself or in correlated subqueries. Aliases exist only for the duration of the query in which they are used.

Column Aliases: Column names can be assigned "aliases" by which they can be referred. Column name aliases are useful when referring to a complicated expression referred to elsewhere in the query.

GROUP BY Expressions: A GROUP BY clause can reference an expression based on a column rather than a column itself; for example, SELECT YEAR(HIREDATE), COUNT(*) FROM EMPLOYEES GROUP BY YEAR(HIREDATE).

ORDER BY Expressions: An ORDER BY clause can reference an expression based on a column rather than a column itself. For example, in ORDER BY YEAR(HIREDATE), LASTNAME will list records in alphabetical order within year of hire, without regard to month or day.

Sort Ascending and Descending: ORDER BY accepts ASC and DESC options; for example, ORDER BY department (ASC), name (DESC), age (ASC), and so on.

Sequence Generator: The system can maintain a current sequence number that it increments each time it is accessed. This function is useful for generating unique key values such as transaction numbers. Although sequence generation can be easily duplicated using an ordinary database table, the sequence generator does not incur the performance overhead of locking the table while the value is updated and selected. The first application to retrieve a value retrieves 1, the second 2, and so on.

Join Operations

Maximum Tables per Operation: The maximum number of tables that can be referenced in a single SQL query.

Maximum Tables per Outer Join: The maximum number of tables that can be referenced by outer join operators in a single SQL query. For purposes of this definition, a single left or right outer join operator references one table; a single full outer join operator references two tables.

Theta-Join: The program's query language supports the theta-join of relations. A theta-join produces the Cartesian product of two relations (on columns with the same domain) with the appropriate theta restriction (i.e., =, [Lz], [Lx], etc.) on the result.

Equijoin: The program's query language supports the equijoin of two relations. An equijoin is a theta-join with theta being equality; a theta-join retains identical attributes.

Full Outer Theta-Join: The program's query language supports full outer theta-joins of relations. The result of an outer join includes rows that do not satisfy the join condition, with null values inserted.

Left/Right Outer Theta-Join: The program's query language supports left and/or right outer theta-joins. A left outer join of tables A and B eliminates rows for which the A values are null. A right outer join eliminates those for which the B values are null.

Full Outer Equijoin: The program's query language supports full outer theta-joins in which theta is equality.

Left/Right Outer Equijoin: The program's query language supports left or right outer theta-joins in which theta is equality.

Join Table with Itself: The program's query language supports joining a table with itself. The program may permit the use of a table alias, in effect allowing the table to act as two tables. For example, an Employees table may include an employee's ID and the name of his or her manager. To list the name of an employee and the name of his or her manager, SELECT A.NAME,

B.NAME FROM EMPLOYEES A, EMPLOYEES B
WHERE A.MANAGER—ID=B.EMP—ID

Joins on Nonindexed Fields: The program's query language supports joins on fields that are not indexed in any way.

Set Functions

Distinct: Modifies a list of values to eliminate duplicates.

Count: Returns the number of values in a list.

Minimum: Returns the minimum number in a list of numbers.

Maximum: Returns the maximum number in a list of numbers.

Sum: Returns the sum of a list of numbers.

Average: Returns the average of a list of numbers, excluding nulls.

Standard Deviation: A program function returns the standard deviation of a list of numbers, excluding nulls.

Variance: Returns the variation of a list of numbers, excluding nulls.

Boolean Functions

BETWEEN: True if x falls between y and z in a sorted ordering of the domain.

EXISTS: Used with SELECT and FROM; returns only records that exist in the FROM table.

IN: True if x is contained in list.

IS NULL: True if x is null.

IS NOT NULL: True if x is not null.

Soundex Match: True if two strings sound alike.

Wild Card Match (single character): True if a string matches a pattern string in which a wild card character can correspond to a single character.

Wild Card Match (variable length): True if a string matches a pattern string in which a wild card character can correspond to a variable number of characters.

Wild Card Match (variable length including 0): True if a string matches a pattern string in which a wild card character can correspond to a variable number of characters or no characters.

AND/OR/NOT: The program supports the logical operators AND, OR, and NOT. AND tests for the condition "all concurrently true;" OR tests for the condition "at least one true;" NOT tests for the negation of a statement.

Arithmetic Operators and Functions

Modulo: A program function returns the remainder of a division operation. For example, the division of 5/3 would return a remainder of 2.

Exponentiation: The program can perform mathematical calculations involving numbers raised to a certain power.

Square Root: The program can compute the square root of a number.

Absolute Value: A program function returns a positive value by stripping the data value of its sign. For example, 3 is the absolute value of both -3 and +3.

Ceiling: A program function returns the smallest integer greater than or equal to a given value.

Floor: A program function returns the largest integer equal to or less than a given value.

Rounding: The program allows the user to specify the number of decimal places to which a value is carried. For example, 10.255 rounded to one decimal places equals 10.3.

Sign: A program function assigns a minus sign to values less than 0 and a plus sign for values greater than 0.

Truncate: A program function truncates values to a specified number of decimal places. For example, TRUNC ((a[,b])) truncates a to b decimal places.

Round/Truncate to Negative Decimals: The program can round or truncate a number to a negative number of decimal places. For example, 10,255 rounded to -2 decimal places equals 10,300. (The same number truncated to -2 places is 10,200.)

Factorial: A program function can compute the factorial of a number using the formula $n(n-1)(n-2)...(n-k)$ where $k = n - 1$. For example, 5 factorial is 120; 10 factorial is 3,628,800.

Pi: A program function returns the value of pi (approximately 3.141592 . . .).

Natural Log: A program function returns the natural logarithm of a number (i.e., the base of the universal number e; approximately 2.71828). For example, the natural logarithm of 1.258 is 0.2295236.

Log Base 10: A program function returns the logarithm of a number using a base of 10. For example, the log (base 10) of 1,000 is 3.

Trigonometric Functions: The program can calculate the following trigonometric functions: sine, cosine, tangent, arcsine, arccosine, arctangent, and arctangent of two values (x and y).

Financial Functions

Payments for Amortization: This function computes the number of periods needed to reduce a principal amount to zero given the periodic payment amount and the interest rate.

Payment Amount for Amortization: This function computes the amount needed for each payment to pay off a loan given the interest rate, amount borrowed, and number of payments.

Interest Amount for Payment Period: A program function exists that returns the amount of a loan payment for a specified period that is going toward the interest on the loan.

Compounded Balance of Onetime Investment: This function computes the final amount of an initial onetime investment over a specified number of periods where a given interest rate is compounded periodically.

Compounded Term of Onetime Investment: This function computes the number of periods required for an initial onetime investment to grow into a specified future value at a specified interest rate compounded periodically.

Future Value of Annuity: A program function can calculate the resulting future value of a series of payments invested at a specified interest rate for a specified period of time.

Compounded Term with Payments: This function computes the number of periods required for an initial investment (present value) to grow into a specified amount (future value) at a specific interest rate.

Annual Interest Rate: This function computes the periodic interest rate given the principal amount, number of payments, and payment amount.

Internal Rate of Return: The program can calculate the approximate interest rate at which the net present value of a sequence of cash flows equals zero.

Modified Internal Rate of Return: A program function returns the internal rate of return when funds are reinvested at a fixed rate.

Present Value: This function computes the present value of equal payments at a specified interest rate per period over a specified time.

Net Present Value: This function can determine the present value of a specified sequence of cash flows in the future, given a specified interest rate.

Double-Declining Depreciation: A program function can calculate the depreciation allowance of an asset using the double-declining balance method, given the purchase price, salvage value, useful life, and current age of the asset.

Straight-Line Depreciation: This function calculates the straight-line depreciation allowance of an asset, given the purchase price, predicted salvage value, and useful life of the asset.

Sum-of-Year's-Digits Depreciation: This function calculates the sum-of-the-year's-digits depreciation allowance of an asset, given the purchase price, salvage value, useful life, and current age of the asset.

String Operators and Functions

Concatenation: The program can combine two or more strings of alphanumeric variables into a single variable string. For example, the user might concatenate LAST NAME and FIRST NAME and use FULL NAME for selection or sorting purposes.

Repeat String: A program function can repeat a specified string, either for a specified number of iterations or until a column is filled. For example, `repeat("ABC",3)` = "ABCABCABC."

Substring: The program can extract the portion of a string at a specified position and of a specified length; for example, `substr("ABCDE," 3,2)` = "CD."

Find Substring: This function finds the position in a string where a substring begins; for example, `findstr("ABCDE," "CD")` = 3.

String Length: This function returns the length of a string; for example, `length("ABCDE")` = 5.

Convert to Uppercase: A function converts all alphabetic characters in a string to uppercase; for example: `upper("abCDe")` = "ABCDE").

Convert to Lowercase: A function converts all alphabetic characters in a string to lowercase; for example, `lower("abCDe")` = "abcde").

Convert to Lowercase with Initial Caps: A function converts all alphabetic characters to lowercase, except the

first character of each word, which is converted to uppercase. For example, "JOHN Q. smith" converts to "John Q. Smith."

Trim Leading Blanks: A function removes all leading blanks from a string; for example, `ltrim(" ABCDE ")` = "ABCDE."

Trim Trailing Blanks: A function removes all trailing blanks from a string; for example: `rtrim(" ABCDE ")` = "ABCDE."

Date and Time Functions

System Time: A program function returns the current time from the system clock.

System Date: A program function returns the current date from the system clock.

Time Zone Conversion: The system can convert a time based on one time zone to its value in another time zone.

Day Interval: A program function or operator can determine the interval in days between two dates.

Month Interval: A program function or operator can determine the interval in months between two dates. Date subtraction using days would indicate that an employee who started work on January 1, 1988 had attained two full years of service on December 31, 1989 (using days between/365) but that an employee who started work on January 1, 1989 had not attained a full year of service on January 1, 1990 (using days between/365.25). Months between/12 produces the correct result in both cases.

Time Interval: A program function or operator can determine the interval between two times.

Add/Subtract Days: A program function or operator can add or subtract a specified number of days to/from a date. For example, `12/31/89 + 1` is 1/1/90.

Add/Subtract Months: A program function or operator can add or subtract a specified number of months to/from a date. For example, `add—months(1/1/89, 12)` yields 1/1/90.

First of Month: A program function returns the first day of the month in which a specified date occurs. For example, an employee may be eligible for a benefit retroactive to the first of the month in which he or she was hired. The effective date is `monthbeg(hiredate)`.

First of Quarter: A program function returns the first day of the quarter in which a specified date occurs.

First of Year: A program function returns the first day of the year in which a specified date occurs.

Extract Day/Month/Year from Date: Given a date value, a program function(s) or operator(s) can extract the month, day, or year of that date.

Extract Year Including Century: The extract year function extracts a four-digit number including the century; for example, the current year would extract as 1990.

Extract Year without Century: The extract year function extracts a two-digit number which does not include the century; for example, the current year extracts as 90.

Extract Day of Week from Date: The program can extract the name of the day of the week from a date (e.g., "Friday" from the date value 3/2/90).

Extract Month Name from Date: The program can extract the name of the month from a date (e.g., "March" from the date value 3/2/90).

Extract Two Date Parts in Single Function: The program can extract two parts of a date in a single function (e.g., the month and day without the year).

Extract Hour/Minute/Second from Time: Given a time value, a program function(s) or operator(s) can extract the hour or minute or second of that time.

Extract Two Time Parts in Single Function: The program can extract two parts of a time in a single function (e.g., the hour and minute without the second).

Miscellaneous Functions

IF/NULL/THEN Value: A program function will substitute a given value for null values.

IF/BOOLEAN/THEN Value ELSE Other Value: A program function or operator returns one of two arguments based on the boolean evaluation of a test condition.

Convert String to Number: A program function converts ASCII string data to number data (e.g., converts "003" to 3).

Convert Number to String: A program function converts number data to an ASCII string (e.g., converts 123 to the string "123").

Convert Number to String in User-Specified Format: The user can specify the format in which a number will be converted to a string. For example, 2.4 could be converted to "2.4," or "002.4," or "2.400."

Convert Character to Integer ASCII: A program function converts character data to its integer ASCII equivalent. For example, character "T" converts to 84.

Convert Integer ASCII to Character: A program function converts an integer ASCII value to character data. For example, 65 converts to "A."

Convert Date to String in User-Specified Format: The user can specify the format in which a date value is converted to a string. For example, to—char(01/15/90,"Today is the ddth day of Month in the year of our Lord YYYY") is "Today is the 15th day of January in the year of our Lord 1990."

Convert String in User-Specified Format to Date: The program can read a string in a user-specified date format and convert it for storage as a date value.

Function Nesting: The program can nest a function within another function; for example, ltrim(rtrim(" ABCDE ")) = "ABCDE"; upper(substring ("abcde",2,2)) = "BC."

Triggered Procedures

Triggered Integrity Procedures: Triggered procedures are used to maintain database integrity. At the database level, a series of compensating procedures is triggered by the occurrence of some integrity-threatening condition.

General Triggered Procedures: Triggered procedures can be used for any general purpose, such as cascading a deletion into related tables. [For this evaluation, triggers are stored at the database level and are triggered by a particular action regardless of the application causing the action.]

Trigger SQL Statement (DML): Triggers can cause SQL DML statements (e.g., Select, Insert, Update, Delete) to be executed.

Trigger SQL Statement (DDL/DCL): Triggers can cause SQL DDL (e.g., Alter Table) or DCL (e.g., Grant or Revoke) statements to be executed.

Trigger Multiple SQL Statements: Triggers can cause multiple SQL statements to be executed as the result of a single triggering action.

Trigger May Include Procedural Logic: A triggered procedure may include procedural logic (e.g., If-Then-Else logic or While loops) in addition to standard SQL statements.

Trigger Separate Stored Procedure: A trigger can cause a database procedure other than the trigger itself to be executed.

Trigger External Program: A trigger can cause an external program (written in C or Assembler, for example) to be executed.

After-Insert Trigger: Triggered procedures can be specified to occur after inserting a record to a particular table.

After-Delete Trigger: Triggered procedures can be specified to occur after deleting a record from a particular table.

After-Update Trigger: Triggered procedures can be specified to occur after updating a record in a particular table.

After-Update, Specific Column: Triggered procedures may be specified to occur after updating a specific column in a table.

Commit Trigger: Triggered procedures can be specified to occur when transactions are committed.

Multiple Triggers on Single Table: A single table may have multiple triggers; for example, the same table may have an insert trigger, an update trigger, and a delete trigger.

Multiple Update Triggers on Single Table: A single table may have different actions triggered by updates to different columns.

Trigger Recognizes Modified Rows: If a trigger needs to perform some action against the table by which it was triggered, that trigger is able to determine which row or rows underwent the activating change. For example, an insert trigger may need to update the row that was just inserted.

Cascading Triggers: A trigger can activate another trigger, which in turn activates another, and so on. For example, an insert trigger defined for Table A may cause an update to Table B, which activates another update trigger defined for Table B.

Iterative Triggers: A trigger can produce the same action that activated it, effectively reactivating the trigger. For example, a row in a table may be updated, activating an update trigger that under certain conditions updates that row again. If this happens, the update trigger will be reactivated.

Procedural Language Extensions

IF/THEN: Part of the product's SQL language implementation includes the ability to execute specific instructions if a conditional test is satisfied.

IF/THEN/ELSE: Part of the product's SQL language implementation includes the ability to determine which set of instructions to execute based on the results of a conditional test.

WHILE Loops: Part of the product's SQL language implementation includes the ability to repeatedly execute a specified number of commands until a specific condition is met. The condition is tested (i.e., true or false) at the top of the loop.

REPEAT/UNTIL or DO/WHILE Loops: Part of the product's SQL language implementation includes the ability to repeatedly execute a specified number of commands until a specific condition changes. The condition is tested at the bottom of the loop.

FOR/NEXT Loops: Part of the product's SQL language implementation includes the ability to repeatedly execute a specified number of commands a specific number of times.

Case Statements: Part of the product's SQL language implementation includes the ability to determine which set of instructions to execute based on the results of a multiple conditional test. This differs from IF/THEN/ELSE statements, which execute a set of instructions if the condition is not met. A case statement tests for multiple results of a condition and executes a different set of instructions for each result.

Branching (GOTO): Part of the product's SQL language implementation includes the ability to cause the execution of a procedure to jump out of sequence to another command in the procedure.

Integer Variables: Variables can be of an integer type and must be capable of having numeric operations performed on them.

String Variables: Variables can be of alphanumeric type and must be capable of storing more than one character.

Floating Point Variables: Variables can be of a numeric type and must be capable of having numerical operations performed on them and of calculating decimal places.

Date Variables: Variables can be of a date type and must be capable of having arithmetic operations performed on them.

Time Variables: Variables can be of a time type and must be capable of having arithmetic operations performed on them.

Arrays: SQL procedures are capable of storing an array of data items in one variable with the use of subscripts. For example, the variable \$1 can store the array \$11, \$12, \$13, and so on.

Select into Temporary Variable: SQL Select statements can select column values or expressions into temporary variables defined for the procedure.

SQL Statements Reference Variables: SQL statements may reference temporary variables defined for the procedures (e.g., in a Where clause or an Insert Values clause).

Modify Variables Using Procedural Logic: Temporary variables may be modified by the procedural logic during the course of the procedure.

IF/WHILE with Subquery: An If/While or other conditional expression may include an SQL query and use the result of that query as part of the condition (e.g., IF (SELECT COUNT(*) FROM EMPLOYEES) 0 THEN . . .).

Data Integrity

Concurrency Control

Row Locking: The program can lock a single row of data needed by an SQL statement without restricting access to the rest of the table.

Page Locking: The program can lock the page (or storage unit) of data containing the row needed by an SQL statement. The page may include several rows, but not the entire table, and should contain data from only one table.

Table Locking: The program can lock an entire table.

Shared Row/Page Lock: A shared lock allows other users to acquire shared locks (but not exclusive locks) on the same row or page. Because of the way SQL statements normally acquire locks (see Implicit Locking, below), shared locks usually mean that other users can query the data, but not update it.

Exclusive Row/Page Lock: An exclusive lock means no other user may acquire a lock of any kind on the locked row or page. Exclusive locks prevent not only updates, but also queries requiring shared locks.

Shared Table Lock: The program can place a shared lock at the table level; other users may query the table, but no rows in the table may be updated.

Exclusive Table Lock: The program can place an exclusive lock at the table level; no other user may acquire locks of any kind on the locked table.

Explicit Exclusive Table Lock: A user may explicitly place an exclusive lock on a table. Explicit locking is often accomplished by an SQL command such as LOCK TABLE.

Explicit Shared Table Lock: A user may explicitly place a shared lock on a table. Explicit locking is often accomplished by an SQL command such as LOCK TABLE.

Implicit Shared Row/Page Lock: Without an explicit lock request by the user, a Select statement acquires a shared lock on the row or page referenced by the query.

Implicit Exclusive Row/Page Lock: Without an explicit lock request by the user, an Insert, Update, or Delete statement acquires an exclusive lock on the row or page referenced by the statement.

Implicit Exclusive Lock on Select: A Select statement can be written in such a way that it acquires an exclusive lock. The SELECT FOR UPDATE syntax is a common implementation of this feature.

Locks Shared across Cursors: Multiple cursors in the same application will not restrict the others' access to data. For example, if a select cursor acquires an exclusive lock on a fetched row, another cursor in the same application can update that row, even though other users are prevented from accessing it.

Repeatable Read: The program holds shared locks until the end of a transaction (i.e., until Commit or Rollback). The user is assured that if he or she reads the same data unit repeatedly, it will not have been changed by another user between reads.

Cursor Stability: A shared lock on a row of data is released when the cursor fetches the next row; a shared lock on a page is released when a fetch moves the cursor off the page. A user who does not intend to repeatedly read the data can thereby allow other users access to the data.

Uncommitted Read: The program supports a Select statement syntax that does not acquire an implicit shared lock; this statement can read data that is exclusively locked (i.e., data that has been changed but not yet committed). This statement is used when the user needs access to information even if some details may be inconsistent.

Snapshot Read: The program can execute a read-only transaction that contains consistent data regardless of how long it takes to execute the transaction. The program in effect takes a snapshot of the data at the time of execution. Other users can make changes to the data without waiting for the transaction to complete, but data integrity is ensured in the read-only transaction.

Prevent DDL Lock: When an Uncommitted Read Select (one that acquires no row or page locks) is executed, the lock placed on the table does not restrict exclusive or shared locks at the row/page or table levels, but prevents DDL operations (e.g., altering or dropping the table).

Simultaneous Table Lock Limit: This is the maximum number of simultaneous locks that can be placed on a table.

Simultaneous Record Lock Limit: This is the maximum number of simultaneous locks that can be placed on a record.

List Locks and Owners: A software facility provides the names of the users owing locks and the objects on which the locks exist.

Lock List Maintained in Database Table: A list of the users owning locks and the locked objects is maintained in a database table.

Deadlock Detection: Deadlocks are detected and broken by algorithm. A deadlock occurs when two users are each waiting for each other to release a lock (e.g., two users have shared locks on the same table and both wish to update it). The program uses an algorithm to select transactions to cancel and roll back.

Specify Deadlock Detect Interval A program facility lets the administrator specify the frequency with which the program checks for deadlocks. Too short an interval adds unnecessary overhead; too long an interval can result in prolonged waiting.

Livelock Detection: Livelock occurs when a user waiting for an exclusive lock on an object is blocked by a shared lock on the same object; a succession of overlapping shared locks continues to preempt the exclusive lock even after the original shared lock is released. The program detects livelock and gives the exclusive lock request priority over subsequent shared lock requests.

Optional Wait/No Wait for Locks: Users can specify whether a lock request should wait for a successful lock or return unsuccessful.

Specify Waiting Period: The user can specify a maximum waiting period for lock requests.

Automatic Update of ROWID: Each row has a ROWID that is automatically updated whenever the row is updated. A user may select the ROWID and use it in the Where clause of a subsequent update; the update will fail if the row (and ROWID) has been changed.

Recovery

Auto Commit: The program has an auto commit option that implicitly treats each SQL command as a transaction without using BEGIN or COMMIT instructions. Aborted queries are automatically rolled back.

Auto Commit On/Off: The auto commit option can be disabled such that changes are not be committed except by an explicit Commit command even if no explicit BEGIN is entered.

Transactions (BEGIN, COMMIT, ROLLBACK): The program uses a set of explicit commands to frame and abort transactions.

Statement-Level Rollback: If a statement during a transaction produces an error, the program automatically rolls back only that statement rather than the entire transaction. The user can explicitly roll back the entire transaction.

Transaction Log On/Off: The writing of changes to a separate log prior to Commit or Rollback can be disabled. Disabling the transaction log forfeits the ability to roll back on an error, but permits processing of very large single statements, such as deleting all records from a large table, without running out of log space.

Bypass Log on Delete: The program offers an alternative to the standard Delete command whereby no transaction log is written. The alternative may be permitted only when all records in the table are to be deleted; otherwise, standard queries would be needed to find records for deletion.

Bypass Log on Insert: The program offers an alternative to the standard Insert command whereby no transaction log is written. The alternative may be allowed only when no index exists on the table; otherwise, an index would have to be updated.

Roll Forward from Archive Log: The program can "replay" transactions from the archive log to recover lost data. Backing up the transaction log provides an incremental back up of the database.

Archiving On/Off: The writing of transactions to an archive log for later rollforward can be turned on or off.

Checkpointing On/Off: The user can specify whether the program uses checkpoints or writes directly to the database. With checkpointing, changes to the database are designated as permanent after a Commit, but held in the transaction log until a checkpoint, rather than being written immediately to the database. Checkpointing reduces the frequency of database writes.

Checkpointing On/Archiving Off: The user can specify checkpointing without rollforward archiving such that the log is cleared at every checkpoint.

Archive Log Separate from Database: The system maintains the archive log in a file physically separate from the database.

Backup Database and Log Together: The user can back up the database and the current log in a single operation. Any uncheckpointed transactions can be written to the database when it is restored.

Backup Database and Log Separately: The user can maintain separate backups of databases and logs.

Clear Log upon Backup: The system automatically clears the transaction log when it is backed up; the user need not explicitly reinitialize it.

Commit without Closing Cursors: After a Commit, a user can continue to execute fetches from a cursor opened prior to the Commit.

Cursor Stability across Commits: After a Commit, a user can continue to execute fetches from a cursor opened prior to the Commit, and cursor stability locks (row or page locks held only while a cursor is positioned on the row or page) continue to be acquired and released as fetches are executed.

DDL/DCL in Multistatement Transaction: A transaction consisting of multiple SQL statements may include DDL (e.g., Alter Table) or DCL (e.g., Grant) statements that are committed or rolled back when the transaction is committed or rolled back.

Programming and Other Tools

Interactive SQL

Store/Retrieve Command in OS File: SQL commands can be saved in an operating system file and later retrieved and reused.

Store/Retrieve Command in Database Table: SQL commands can be saved in a database table and later retrieved and reused.

Edit Retrieved Command before Execution: A command retrieved from a file or table can be edited before it is executed.

Execute Batched Commands from File/Table: A batch of two or more SQL commands can be read from a file or table and executed.

Enter/Edit/Execute Batched Commands On-Line: A batch of two or more SQL commands can be entered on-line and stored or retrieved and edited prior to execution.

Command Batch with Procedural Logic: A batch of SQL commands may include procedural logic (e.g., If-Then-Else logic or While loops) in addition to standard SQL statements.

Execute Stored Procedures: The command to execute a stored procedure may be entered interactively or as part of a batch of SQL commands.

Enter/Edit/Save Stored Procedure On-Line: A stored procedure may be entered, edited, and stored to the database in interactive mode.

Store Commands with Runtime Variables: SQL commands can be defined with missing values; when the query is run, the program prompts the user for these values.

Runtime Variable, Repeated Use: A command file can use a runtime variable repeatedly in successive commands; the program prompts for a value the first time, and that value is retained during successive references to it.

Modify/Manipulate Variables during Processing: Runtime variables can be modified or manipulated during processing, and the modified value is used for successive references to the variable.

List Table Columns and Attributes: The interactive SQL facility includes a command that lists columns in a specified table along with their attributes (e.g., data-type, null/not null, etc.). The user need not explicitly query a system table containing that information.

Spool/Save Query Result to ASCII File: The results of a query can be stored in a DOS or OS/2 ASCII file.

Print/Spool Query Result: The results of a query can be directed to a printer.

Specify Column Header for Query Results: A column header can be specified for displaying query results.

Specify Column Width for Query Results: A column width can be specified for displaying query results.

Specify Numeric Format for Result Column: A format can be specified for displaying a numeric column in query results (e.g., display floating point values as 999.99).

Specify Date Format for Result Column: A format can be specified for displaying a date column in query results (e.g., display dates as MM/DD/YY).

Time Operation: The program can display the time elapsed to execute a command or set of commands.

Scroll Query Results: The user may scroll through the results of a query after it has finished executing.

Scroll Left and Right: The user may scroll left or right through query results that are wider than a single screen.

Execute Operating System Command: DOS and OS/2 commands can be executed without exiting the interactive SQL module.

Set-Level Insert: A number of records can be inserted into a table from a single interactive query command. "Set-level" means the action takes place across the board (i.e., more than one for each command).

Set-Level Delete: A single command in query mode can be used to delete a number of records from a table.

Set-Level Update: A single command in query mode can be used to modify a number of records in a table.

Prompted Select/Insert/Update/Delete: The user is guided through the entry of a Select, Insert, Update, or Delete command by a series of prompts or menus.

Prompted Table Definition: The user is guided through the creation or alteration of a table by a series of prompts or menus.

Prompted User/Security Maintenance: The user is guided through the setting up of user IDs and passwords and the granting and revoking of privileges by a series of prompts or menus.

Prompted Backup/Restore: The user is guided through the backing up and restoring of databases and logs by a series of prompts or menus.

Prompted Start-Up/Shutdown: The user is guided through the starting up and shutting down of databases by a series of prompts or menus.

Load/Unload Utility

Load Appending to Existing Records: Records can be added to a table that already contains records.

Load Replacing Existing Records: The program can delete any records already in the table and add the new ones.

Load Updating Existing Records: The program can match input records to existing records based on key fields and update the appropriate row in the table based to the values in the input record.

Load Fixed-Length ASCII File: The program can load records from a file in fixed-length ASCII format.

Unload Fixed-Length ASCII File: The program can unload records into a file in fixed-length ASCII format.

Load Delimited ASCII File: The program can load records from a file in delimited ASCII format.

Unload Delimited ASCII File: The program can unload records into a file in delimited ASCII format.

Unload Several Tables to Single File: Several tables can be unloaded to a single output file.

Load Selected Tables from File: The user may select tables to load from a file to which several tables were output.

Load/Unload in Native Format: The program can unload a table to a file in its native format and later reload it. Native format, used primarily as a form of backup, can generally only be read by the program that wrote it.

Load/Unload in Native Format with Create Table: The program can unload a table to a file in its native format with Create Table instructions. When the file is reloaded, the table is created.

Load/Unload in Native Format with Create Index: The program can unload a table to a file in its native format with Create Index instructions for any existing indexes. When the file is reloaded, the indexes are created.

Load/Unload ASCII with Create Table: The program can unload a table to an ASCII file with Create Table instructions. When the file is reloaded, the table is created.

Load/Unload ASCII with Create Index: The program can unload a table to an ASCII file with Create Index instructions for any existing indexes. When the file is reloaded the indexes will be created.

Input Column Order Different from Table: The order of columns on an input record can differ from the column order in the table to which the records will be loaded.

Output Column Order Different from Table: The order of columns on an output record can differ from the column order in the table from which the records are unloaded.

Specify Input Date Field Format: A user can specify the format for input date values.

Specify Output Date Field Format: A user can specify the format for output date values.

Specify Output Numeric Field Precision: A user can specify the number of decimal places for numeric output.

Implied Decimal in Input Numeric Field: The program can read numeric values with an implied decimal (e.g., interpreting the format 999V99, the program converts input value 98765 to 987.65 and 00300 to 3).

Implied Decimal in Output Numeric Field: The program can output numeric values with an implied decimal (e.g., interpreting the format 999V99, it converts 987.65 to 98765 and 3 to 00300).

Load Column Based on Expression: The program can insert a column value based on expression rather than an explicit input value. For example, if the program does not support implied decimals, it might allow the user to specify that the value be inserted as (input value/100).

Save Load File Format to OS File: Load file format instructions (e.g., length of columns, date and numeric formats, etc.) can be saved to an operating system file that can be read when the input file is loaded.

Save Load File Format to Table: Load file format instructions (e.g., length of columns, date and numeric formats, etc.) can be saved to a database table that can be read when the input file is loaded.

Save Unload File Format to OS File: Unload file format instructions (e.g., length of columns, date and numeric formats, etc.) can be saved to an operating system file that can be read when the data is unloaded.

Save Unload File Format to Table: Unload file format instructions (e.g., length of columns, date and numeric formats, etc.) can be saved to a database table that can be read when the data is unloaded.

Prompt User for File Format: The programs help the user specify file format instructions (e.g., length of columns, date and numeric formats, etc.) using a series of prompts or menus.

Save Response to Prompts in Format File/Table: The user's responses to prompts for file format instructions are saved in an operating system file or database table that can be read when the data is loaded or unloaded.

Load and Edit Format File/Table: The file or table containing file format instructions can be retrieved and edited before the data is loaded or unloaded.

Stored Procedures

Store as Database Object: A command or set of commands can be stored in the database. Each procedure, like each table, index, or view, is a database object.

Store Executable File on Server: A program executable can be stored on the server and accessed by workstation applications.

Retrieve and Edit Procedure Definition: A procedure definition can be retrieved from the database, edited, and stored in modified form without dropping and re-creating the procedure.

Execute Procedure without Retrieving: A procedure can be executed directly by the database management software (back end), rather than having to be retrieved into the front end and executed there as a command batch.

Explicit Execution (nontriggered): A command can execute a procedure by name or some other identifying symbol, rather than it being triggered by some other action.

Maximum Levels of Nested Procedures: This is the maximum depth in levels for procedures calling other procedures.

Values Passed from Calling Program/Procedure: A procedure can have values passed to it by the program or a procedure that calls it.

Values Returned to Calling Program/Procedure: A procedure can return values to the program or procedure that called it.

Procedures Use Local Variables: A procedure can define local variables, and those variables can be referenced and manipulated during execution.

Select into Local Variable: A procedure can select a value into a local variable.

Arrays: A procedure can store an array of data items in one variable using subscripts. For example, the variable \$1 can store the array \$11, \$12, \$13, and so on.

Procedural Language Extensions: A procedure may include procedural logic (e.g., If-Then-Else logic or While loops) in addition to standard SQL statements.

Multiple SQL Commands: A single stored procedure may contain more than one SQL command.

SQL DML Commands: A procedure may include SQL DML statements (e.g., Select, Insert, Update, Delete).

SQL DDL/DCL Commands: A procedure may include SQL DDL (e.g., Alter Table) or DCL (e.g., Grant or Revoke) statements.

SQL Cursors: SQL queries can access database tables using programming cursors. A programming cursor acts as a pointer, allowing the procedure to scan and use the selected rows. After a cursor is defined using an SQL command, individual rows are returned using successive fetches, with the cursor then pointing to the row just fetched.

Fetch Next: A procedure can include a command that moves a cursor from the current row to the row immediately following it in the set of rows returned by the query; the command then returns that row.

Fetch Previous: A procedure can include a command that moves a cursor from the current row to the row immediately before it in the set of rows returned by the query; the command then returns that row.

Fetch ahead N Rows: A procedure can include a command that moves a cursor from the current row to the row n (any specified number) rows ahead of it in the set of rows returned by the query; the command returns

that row. *n* is any specified number. These operations are accomplished with a single fetch, not with a loop consisting of *n* fetches.

Host Language Interface

C: A host language interface is available for C language.

Cobol: A host language interface is available for Cobol.

Fortran: A host language interface is available for Fortran.

PL/1: A host language interface is available for PL/1.

Basic: A host language interface is available for Basic.

Pascal: A host language interface is available for Pascal.

Embedded SQL: Host languages are interfaced by embedding SQL commands.

Precompiled Interface: A host language interface is implemented as a precompiler. Precompiled SQL commands are converted to the host language, optimizing performance of the query.

Direct Call Interface: The program interacts with the database through direct host language calls. With a precompiled interface, the programmer embeds SQL commands in a host language, and the precompiler converts the embedded SQL to host language commands. With a direct call interface, the programmer explicitly codes host language calls to the database. Coding calls to the database can be more difficult because it often involves learning a large number of vendor-supplied host language library functions, but it gives the programmer more direct control of the program's interface with the database.

Bind Prior to Runtime: At a point before a program is run, (such as at precompilation), the system determines and stores an optimization plan for execution of SQL statements based on existing conditions in the database (such as what indexes exist on the tables to be accessed). Programs to be executed against different databases at different times must be bound to each database individually.

Dynamic Recompilation: Embedded queries are dynamically recompiled when deemed necessary or advantageous by the database management system. The user does not have to specify a recompilation. For example, if the database is modified such that an index is deleted, a precompiled procedure will automatically re-define and recompile the SQL command to accommodate the change to the database.

Preserve Context: SQL statements can be parsed or compiled once and executed repeatedly without reparsing, even if the values for variables referenced by the statements change.

Interpreted Interface: An interface can be implemented as an interpreter; SQL commands are parsed and processed at runtime.

Dynamic SQL: The exact syntax of SQL commands can be specified at runtime. While too much variation between runs can make management of the variables impractical, some aspects of commands, such as what columns to include in a WHERE clause, may quite reasonably vary from run to run.

SQL Cursors: SQL queries can access database tables using programming cursors. A programming cursor acts as a pointer, allowing the program to scan and use the selected rows. After a cursor is defined using an SQL command, individual rows are returned using successive fetches, with the cursor then pointing to the row just fetched.

Fetch Next: A program can include a command that moves a cursor from the current row to the row immediately following it in the set of rows returned by the query; the command then returns that row.

Fetch Previous: A program can include a command that moves a cursor from the current row to the row immediately before it in the set of rows returned by the query; the command then returns that row.

Fetch ahead N Rows: A program can include a command that moves a cursor from the current row to the row *n* rows ahead of it in the set of rows returned by the query; the command then returns that row. The fetch ahead is accomplished with a single fetch, not with a loop consisting of *n* fetches.

Select into Array: A query that returns more than one row can be selected directly into a program array (e.g., a query that returns 25 rows can select into a set of variables defined with 25 or more occurrences).

Fetch Cursor into Array: Multiple rows from a cursor can be retrieved into an array using a single fetch operation. The program automatically fetches as many rows as occurrences of the variables being selected into. For example, a command to fetch a cursor into a set of variables defined with 25 occurrences returns 25 rows for each fetch. Every variable fetched into must have the same number of occurrences.

Insert Using Array: A single insert statement can insert values from an array. The first occurrence of all the variables being inserted constitutes one row, the second occurrence another row, and so on.

Update Using Array: A single update statement can update multiple rows using values from an array. Both the SET clause and the WHERE clause of the update statement refer to array values. Rows where column values equal the first occurrence of the variables referenced in the WHERE clause are updated to the first occurrence of the variables referenced in the SET clause. Rows with values equal to the second occurrence are set to the second occurrence, and so on.

Update Where Current: The program can update the current row without a WHERE clause identifying the row based on key values.

Update Joined Cursor Where Current: The program can update the current row for a cursor defined as a join of multiple tables without a WHERE clause to identify the row based on key values.

Delete Where Current: The program can delete the cursor's current row without a WHERE clause to identify the row based on key values.

Automatic ROWID Validation A host language interface facility automatically checks whether the ROWID for a row being updated has been changed since the row was selected.

Whenever Error Processing: A program can contain a single command dictating the action to be taken whenever any error is found on any subsequent database command.

Whenever Not Found Processing: A program can contain a single command dictating the action to be taken whenever any subsequent database command returns no rows.

Bulk Execute: Multiple database commands can be buffered and sent to the database server in bulk.

Bulk Execute with Separate Processing: Multiple database commands can be buffered and sent to the database server in bulk, but the results of each command are processed one at a time.

Background Execution: The program can send a command to the server, perform other tasks while the command is being processed, and then process the results of the command.

Back-End Result Sets: All rows returned by a multirow query can be stored concurrently and browsed by the program. Back-end result sets are stored as temporary database objects.

Front-End Result Sets: All rows returned by a multirow query can be stored concurrently and browsed by the program. Front-end result sets are stored in a program buffer or in a temporary workstation storage area.

SQL DDL/DCL Commands: A program may include SQL DDL (e.g., Alter Table) or DCL (e.g., Grant or Revoke) statements.

Miscellaneous Front-End Support

Forms Generator: The database server package includes a forms generator.

Report Generator: The database server package includes a report generator.

Windows/Presentation Manager Interface: The database server package includes a database interface module based on Microsoft Windows or OS/2 Presentation Manager, or on a comparable graphical interface.

Windows-Based Application Generator: The database server package includes the necessary tools for developing database applications to run under Microsoft Windows.

dBase Compiler/Interpreter: The database server package includes a compiler or an interpreter that permits programs written in dBase command language to interface with the database.

Lotus 1-2-3 Interface: The database server package includes the necessary tools to allow Lotus 1-2-3 spreadsheets to interface with the database.

Microsoft Excel Interface: The database server package includes the necessary tools to allow Microsoft Excel spreadsheets to interface with the database.

Other Spreadsheet Interface: The database server package includes the necessary tools to allow other spreadsheets to interface with the database.

Portability and Network Capability

Server Environments Supporting DOS and OS/2 Front-Ends

NOTE: In this section, "compatible server" means that DOS or OS/2 applications written for the vendor's OS/2 server will interface with a server running in the specified environment with no modification to the application.

VAX: The vendor offers a compatible database server running in the VAX VMS environment and the necessary workstation software for it to be accessed from DOS or OS/2.

MVS: The vendor offers a compatible database server running in the MVS environment, and the necessary workstation software to allow it to be accessed from DOS or OS/2.

VM: The vendor offers a compatible database server running in the VM environment and the necessary workstation software for it to be accessed from DOS or OS/2.

MS-DOS: The vendor offers a compatible database server running in the MS-DOS environment.

UNIX/Xenix: The vendor offers a compatible database server running in the UNIX or Xenix environments and the necessary workstation software for it to be accessed from DOS or OS/2.

Network Support

Novell: This product or a compatible product from the same vendor will run on a Novell network.

Microsoft: This product or a compatible product from the same vendor will run on a Microsoft network.

3Com: This product or a compatible product from the same vendor will run on a 3Com network.

IBM: This product or a compatible product from the same vendor will run on an IBM network.

Banyan Vines: This product or a compatible product from the same vendor will run on a Banyan Vines network.

Token-Ring: This product or a compatible product from the same vendor will run on a network using token-ring hardware.

Ethernet: This product or a compatible product from the same vendor will run on a network using Ethernet hardware.

NETBIOS: This product or a compatible product from the same vendor will run on a network using the NETBIOS protocol.

Named Pipes: This product or a compatible product from the same vendor will run on a network using the Named Pipes protocol.

SPX/IPX: This product or a compatible product from the same vendor can be run on a network using the SPX/IPX protocol.

TCP/IP: This product or a compatible product from the same vendor will run on a network using the TCP/IP protocol.

Database Server and File Server in Same System: The database server software and network file server can be installed and run on the same PC. When the primary purpose of the network is to provide access to the database, the single-server configuration saves hardware costs.

Dedicated Database Server: The database server software can be installed and run on a PC with no other software except the operating system and any necessary network connection software. Separating the database server and file server reduces CPU contention, thereby optimizing performance.

Distributed Processing

Access Databases on Multiple Stations: An application can access databases residing on two or more network stations at the same time.

Join Tables from Multiple Stations, Single Query: An application can execute a single query that retrieves data from tables from databases residing on two or more network stations.

Insert/Update on Multiple Stations, Same Transaction: An application can insert records into tables or update tables in databases residing on two or more network stations using a single transaction.

Two-Phased Commit: When a transaction involves two databases, Commits are processed in two phases. The program verifies that both databases are ready to Commit (i.e., no errors warranting rollback) and that if executed the Commit will be completed successfully. When verification is completed, the data is actually committed. Two-phased Commit ensures data integrity across multiple databases.

Link with Database Server under Another OS: Vendor-supplied tools enable an application to access simultaneously its OS/2 database server and its database server running under another operating system.

Link with Other Vendors' OS/2 Servers: Vendor-supplied tools enable an application to access simultaneously its OS/2 database server and another vendor's OS/2 database server.

Link with DB2: Vendor-supplied tools enable an application to access simultaneously its OS/2 database server and an IBM DB2 database. ■

SQL Front Ends

A Report from NSTL

In this report:

Program Evaluations -311

Rating Summaries..... -312

Performance Results..... -316

Vendors -322

Characteristics..... -323

Synopsis

Focus

This report evaluates four SQL front-end products. NSTL highlighted their implementation, application development capability, performance, capability to work with different SQL servers, and transportability across platforms.

Programs Tested

- Paradox*
Borland International, Inc.
- DataEase/SQL*
DataEase International
- Oracle Tools and Database*
Oracle Corp.

Advanced Revelation

Revelation Technologies, Inc.

Program Recommendations

- Borland Paradox
- Oracle Tools and Database

Source

Based on data generated by tests designed and conducted by National Software Testing Laboratories, Inc. (NSTL), a division of Datapro Research Group, Inc., Plymouth Meeting, PA 19462. Telephone (800) 223-7093.

<p>Ratings Key (On a scale of 0 to 10)</p> <p>Ratings</p> <p>● 7.0 - 10.0</p> <p>○ 5.0 - 6.9</p> <p>⊗ under 5.0</p> <p>★ Recommended</p>
--

Overall Rating	Program Name	Version Tested	Performance	Versatility	Error Handling	Ease of Learning	Ease of Use	Price
7.5	Borland International Paradox	3.5	●	○	●	○	●	\$1,290 ★
6.7	Oracle Tools and Database	5.1C	●	●	○	○	○	\$1,299 ★
6.5	DataEase International DataEase/SQL	1.01	⊗	●	⊗	●	○	\$995
5.5	Revelation Technologies Advanced Revelation	2.03	○	●	○	⊗	⊗	\$1,690

Overview

The relatively recent introduction of SQL servers for the OS/2 environment (Oracle Server for OS/2, Microsoft SQL Server, and IBM OS/2 Extended Edition Database Manager) has redefined network database management. Prior to the introduction of these products, Oracle offered communications tools whereby applications could access remote databases on machines running VAX VMS or UNIX operating systems. The OS/2 servers prove more conducive to PC networks running Novell NetWare or Microsoft LAN Manager.

Database servers take advantage of a client/server architecture. Workstation applications communicate with a separate program, running at a remote location, which handles all data management. The client/server architecture reduces network traffic and streamlines data management. Earlier server-based database management schemes focused all database management at the workstation processor, with several processors concurrently accessing data at a common location.

Client/server architecture permits the division of database applications into two components—the tools and the interfaces used by the user (i.e., the front end) and the data management (i.e., the back end). The introduction of database servers (at the back end) created the need for front-end access tools. Microsoft SQL Server was well received as a database manager, but it provides virtually no means for accessing data except through programs written in C or Basic. IBM offered some user interface features for OS/2 workstations, but nothing other than programming tools for DOS workstations. Oracle's Server for OS/2 was immediately accessible using the tools in its standalone PC product, but Oracle also solicited front-end support from third-party vendors. The second half of 1990 saw the introduction of front-end products designed to work with other vendors' databases.

Evaluation Criteria

NSTL's evaluation of SQL front ends considers products offering full database management capability, including interactive query tools, forms and report generation, the ability to create tables and indexes, and either a procedural language or facilities for using a standard programming language such as C or Cobol. All the products run under DOS and support one or more major SQL servers for OS/2 (Oracle, Microsoft, or IBM). Three multi-user database products have been enhanced or equipped with add-on modules for accessing SQL server data. Oracle's front-end product can be purchased as a user interface with development tools (separate from the built-in database) for use with a remote database.

More limited, specialized front-end tools such as spreadsheet interfaces and standalone report writers are excluded. Products with graphical interfaces are also excluded to avoid having to debate the merits of character-based and graphical interfaces.

Oracle Tools works with any Oracle database on any platform (tested with Oracle Server for OS/2). DataEase/SQL and Advanced Revelation, at their current levels of development, support only the Microsoft/Sybase platform. To facilitate comparison with all three competitors, Paradox is tested with both Oracle Server and Microsoft SQL Server; it also supports the IBM OS/2 Extended Edition Database Manager and RDB on a mini-computer.

Paradox and Advanced Revelation access SQL servers through add-on modules. Paradox provides menu options for server access when its SQL Link is installed. Advanced Revelation's SQL Server Bond is one of several environmental bond products available for accessing data in non-native formats. DataEase/SQL incorporates all the functionality of the standard DataEase package plus SQL server access. Oracle Tools and Database includes a local single-user database, and applications can access remote Oracle databases using SQL*Net drivers available separately or as part of server packages. Oracle Tools without a local database is available at a lower cost.

The evaluations consider NSTL's five standard software evaluation categories with some modifications. Ease of Learning considers users who are new to the products as well as those who must adapt knowledge of the standalone version for use with SQL servers. Ease of Use considers end users and application developers. Error Handling evaluations assess specific program features and their handling of actual error conditions. Versatility ratings consider features for accessing servers and application development flexibility. Performance measures each product's efficiency communicating with the server and managing data locally; interactive queries measure their efficiency translating commands into SQL, passing commands to the server, and retrieving data to the user.

System Requirements

All the front-end products require at least an 80286 processor and some minimum hard disk space (5M bytes or less for Advanced Revelation and Paradox; 15M bytes for Oracle Tools with all its components). All feature broad network support, with some restrictions imposed by the server platform (e.g., Microsoft SQL Server requires a network with Named Pipes support).

All the products include standard database management software plus software for accessing remote database servers, so 640K bytes of RAM will not be sufficient. Advanced Revelation requires a minimum of 640K bytes, but recommends an additional megabyte of expanded memory. The program will load and connect to the server with only 640K bytes, but many operations abort and return an error message indicating insufficient memory. Even with expanded memory, NSTL encountered frequent crashes for no apparent reason, sometimes preventing reentry to the program. Reconfiguring Microsoft LAN Manager to run without expanded memory avoids possible conflict and should reduce the frequency of problems, but NSTL was unable to eliminate these problems during testing. NSTL could determine no obvious reason for Advanced Revelation's crashes, but crashes occur only when using expanded memory and only when the Server Bond is in use.

The other three products use extended rather than expanded memory. Some of Oracle's tools run within 640K bytes, but SQL*ReportWriter and SQL*Menu require 1.5M bytes. An additional megabyte is required to access a local database. Paradox requires 1M byte for most platforms, and 1.5M bytes with Oracle Server for OS/2. DataEase/SQL requires 2M bytes and aborts frequently with protected mode violations.

Front-End Implementation

DataEase/SQL and Advanced Revelation offer their full range of commands for accessing and manipulating server data and local data. DataEase/SQL must designate a database engine type for forms. DataEase/SQL users can use embedded SQL to supplement standard DataEase commands, but virtually any server data operation can be accomplished with the standard functions. Advanced Revelation also supports the use of embedded SQL, but many of Advanced Revelation's features go beyond the scope of SQL. Applications designed to manipulate Advanced Revelation's native data format with maximum efficiency may require substantial modification to access SQL server data.

Paradox implements only a subset of its functionality for working with server data. Commands that add new data (e.g., Data Entry, Copy, Add) work the same with server tables and Paradox tables. Most Query By Example features work with server data, with the exception of some advanced functions (e.g., using Changeto with multiple tables; calculated aggregate values in selection criteria). The View and Edit commands work only with native Paradox data, and PAL language functions that work only in view or edit modes are not available. The user queries the records into a local answer table for viewing; updating or deleting existing records requires either an embedded SQL statement or a Query By Example.

Oracle Tools and Database implements a series of independent modules accessed separately from DOS. SQL*Plus allows entry of SQL commands and produces simple columnar reports. SQL*ReportWriter generates complex reports. SQL*Forms is a very powerful forms designer; SQL*Menu creates simple menus for accessing applications developed in the other modules. Utilities include SQL*Loader for importing nonnative data

into Oracle databases, Pro*C for accessing Oracle databases from C language programs, and others. The Client Manager lets users select modules from a menu system rather than from the DOS prompt.

As mentioned above, Oracle Tools accesses only Oracle databases, so all data is in native format. Oracle databases are available for practically any hardware platform and operating system, and Oracle's SQL*Net links most platforms to PC-DOS front ends. Any front end capable of working with an Oracle database can use SQL*Net to access data on other platforms.

Field Formats

To provide transparent access to all types of server data, the front-end product must convert server data formats to native formats, handling the different field types and indexing schemes used by various components. All the front ends and servers support character and numeric data, but each may support different variations (e.g., fixed- and variable-length character, small and large integer, fixed and floating-point decimal).

When accessing existing server tables, the front end translates server field formats into its own formats. When creating new server tables, the front end uses the server field type most closely resembling its own. Problems arise when server and front-end data types have no compatible counterpart. Advanced Revelation provides field types for every available SQL server data type, but not all servers support all Advanced Revelation data types.

Long-Text Fields

All the servers support a long-text format (longer than an ordinary character field), but only Advanced Revelation can access long-text fields. As long as long-text fields on the server are not defined as mandatory, the front end can access tables as though those fields do not exist. DataEase/SQL provides a long-text field for native data but cannot access Microsoft SQL Server's long-text fields. Oracle's SQL*Forms and SQL*ReportWriter tools cannot access long-text fields even in Oracle databases.

Date/Time Fields

Oracle Server and Microsoft SQL Server support a single date/time field that combines a date and a time. Advanced Revelation provides a similar date/time field and individual date and time fields that access the appropriate part of the server's combined date/time field and disregard the rest. When a front end uses a date or time field to write data to a server date/time field, a default value is written to the unused portion of the field (usually 12:00 a.m. and 1/1/00).

The other front ends support separate date and time fields. Oracle's SQL*Forms date field displays the date portion of a database field. Oracle's time field is actually a character field displaying a time format; a character substring function is used to access the time in a database field. DataEase/SQL's date and time fields access the appropriate part of a server date/time field. Paradox provides only a date field.

DataEase/SQL encounters problems with date/time values written by non-DataEase applications and with any combined date/time values written using embedded SQL. When a user attempts to modify data via a DataEase form, DataEase/SQL compares the database values with those originally selected by the form to check for modifications by another user (a concurrency control feature). For the form's individual date and time field, the program expects to find a default value in the unused portion of the server's date/time field. If the server's date/time field contains both a date and a time, and the form displays only a portion of that field, DataEase/SQL cannot update the record. The problem persists even when the form's date or time field is defined for display only.

Multivalue Fields

Advanced Revelation supports multivalue fields containing multiple values (e.g., an employee record might include a multivalue field listing all dependents). Advanced Revelation applications often use multivalue fields to store the key values of related records in one-to-many relationships. Advanced Revelation can store multivalue field data on a server in a long-text or standard character field, but other programs accessing the data will not recognize delimiters separating the values. If more than one product will be using the data, both the data structure and the accessing applications will have to be altered.

Indexes

Oracle Tools uses SQL commands in its SQL*Plus interface to maintain indexes on the server. The other products use translations comparable to those used with different field data types.

Paradox and Advanced Revelation translate primary keys (designated when creating tables) into unique clustered indexes in Microsoft SQL Server and unique indexes in other servers. Translation facilities are limited to primary keys, but indexes can be created for other key values using SQL commands. When accessing existing server tables, Advanced Revelation uses the table's unique clustered index as the primary key. If no unique clustered index exists, the table cannot be accessed in operations that use primary keys (e.g., updating records with a form, referencing the table in a lookup formula). In Paradox, the presence or absence of key fields has no impact on functionality.

DataEase/SQL uses the first few characters in its unique and indexed fields to create a unique clustered index in a server table. Separate indexes are created for other form fields defined as indexed, unique, or nonunique. Like Advanced Revelation, DataEase/SQL uses a record's unique clustered index to identify specific records when browsing a table via a form. Browsing is possible when no unique clustered index exists, but performance can be inefficient on large tables.

Advanced Revelation supports indexes with no server counterparts, such as Btree and relational indexing. Btree indexes facilitate lookups based on values other than the record's primary key; applications use the Btree.Extract command to perform such lookups. SQL commands can be used to create server indexes on nonkey fields, but applications using Btree.Extract must be recoded.

Relational indexing in Advanced Revelation stores keys of related records in a one-to-many relationship in multivalued fields. For example, each record in an Orders table may include a multivalued field storing record key values for related line items. Assuming that each line item record contains a part number, quantity, and price, the Orders record will be defined with multivalued lookup fields referencing part numbers, quantities, and prices for line items with matching record keys. To a standard Advanced Revelation application, the Orders table appears to contain the full line item information. Such relationships between tables are

foreign to SQL, but defining one-to-many relationships using normal SQL data structures requires sophisticated programming referencing multiple tables based on join criteria.

Accessing Server Data

In theory (and in Oracle Tools), establishing access to an SQL database is as simple as logging in (user ID and password) and entering SQL commands to access tables. In practice, most products use a form or other interface tools that require a few steps to recognize database tables.

DataEase/SQL controls all table access through forms. Even noninteractive procedures written in DQL reference data by form name. In standard DataEase, forms and tables are virtually synonymous; in DataEase/SQL, a form can be defined with a new database table (the form is said to "own" the table) or it can be defined on an existing table. Except with embedded SQL, DataEase/SQL cannot access a table without first defining a form for it.

Advanced Revelation treats data in a specific directory as a volume, and all data in a volume uses a single format (native format or one recognized by the server bond). The user defines a server-specific volume with connection parameters for the server as the volume location. Once the volume is attached, the program automatically creates local dictionary files for database tables, which can then be accessed as Advanced Revelation data files. Tables are created and modified using the program's standard file definition tools.

Paradox accesses database tables through replicas. Replicas resemble DataEase forms without screen formatting. A replica acts as a local pointer to a specific database object (including individual column definitions) that the system sees as a local Paradox table. The user can create a replica and a database table together (using the Create command with the Remote option) or create a replica for an existing table. Instructions for accessing existing tables are not easily found in the manual, and the processes of creating and using replicas are comparatively time consuming and confusing. Except for embedded SQL, Paradox commands reference replica names rather than table names.

Data Interfaces

Once forms, replicas, or volumes have been defined, the product's standard interfaces are used to access data. DataEase/SQL forms directly browse and update records. Forms can step (forward or backward) through records in database order based on a unique clustered index or step through records matching selection criteria. Stepping backward in a large table can involve delays (to several seconds); stepping backward in a Microsoft SQL Server table can cause the operation to abort. DataEase/SQL supports table relationships "predefined" in standard DataEase to facilitate joined queries, lookup fields in forms, master-detail forms based on one-to-many relationships, and enforcement of referential integrity. Relationships can be defined between two server tables, between local and server tables, or between tables on different servers.

Advanced Revelation forms (called windows) generally access a single table, but virtual lookups based on one-to-one and many-to-one relationships are easily defined. As mentioned previously under Indexes, one-to-many relationships are very difficult to define. Nested lookups to almost any level make it possible for a form or report (based on a single table) to access data from a large number of tables. Lookups can be defined between server tables, between local and server tables in either direction, or between tables on different servers. Unfortunately, lookups can be based only on the target table's primary key. An SQL command window provides easy access to data in multiple tables based on join criteria not easily defined with Advanced Revelation tools.

Paradox users access server data primarily through the product's Query By Example interface. Query results are returned to a local answer table for manipulation using Paradox' full range of tools. New records are entered using the data entry facility. A user can interactively update selected records using the QBE Changeto command or enter SQL commands using the MiniScript key or from a special SQL command window.

Oracle Tools and Database provides two tools for viewing data. SQL*Plus permits direct entry of SQL commands but provides no facilities for manipulating query results. The powerful SQL*Forms can generate a default form for editing a single table or incorporate a complex application into a single form.

Data Transfer Tools

Database users often need to download a quantity of server data to local tables for updates and then upload it. Data transfer and local manipulation keep server connection times to a minimum.

Paradox essentially transfers data for all types of manipulation. Query results are automatically written to local tables, and new data is entered and inserted using the data entry facility. The Copy and Add commands (used in standard Paradox to copy the contents of one table to another) can be used to upload data to a server or to transfer data between database platforms. Paradox' Import and Export commands do not work with remote tables.

Advanced Revelation's Record Copy facility copies records between tables on different servers or in different data formats provided the table structures are the same. DataEase/SQL uses a similar data transfer option, and an import facility works with Microsoft SQL Server's Bulk Copy utility.

Oracle Server permits definition of database links whereby a user connected to one Oracle database can download (but not upload) data from a linked database. Oracle Server's database links can be used by any front-end product, but Paradox accomplishes such link-related tasks more efficiently with its own tools. Oracle's SQL*Plus Copy command transfers data between databases in either direction; because it is not an SQL command, the SQL*Plus Copy command is not available in any other Oracle tools.

SQL Language Support

SQL commands constitute the primary means of accessing SQL server data, in addition to some server-specific import/export or backup/recovery utilities. All front-end programs communicate with the server using SQL commands entered by the user, embedded in application code, or translated from its proprietary commands. Query translation is relatively straightforward except for some refinements to take advantage of server-specific performance enhancements and concurrency control features.

Proprietary Command Translation

Paradox' query facilities most closely match SQL's powerful query capabilities, and its Query By Example generally executes as efficiently as SQL commands entered directly.

DataEase's DQL query language relies heavily on the FOR clause, which specifies the target table and defines record selection criteria. The FOR clause translates into an SQL WHERE clause, and nested FOR clauses work well with Microsoft SQL Server's multithreaded architecture, which permits multiple active SQL statements. Nested FOR clauses in master-detail reports select multiple detail records for each row returned by the master query. Because a FOR clause can reference only a single table, DQL must use nested FOR clauses where a single SQL join query would be more appropriate. DataEase/SQL effectively determines when to translate nested FOR clauses into nested queries and when to use a single join query.

Except for Oracle Tools, which almost always uses SQL statements coded by a user or programmer, all the products offer facilities for viewing SQL command translations of proprietary commands. Paradox translations can be edited and incorporated into programs as embedded SQL. Using Paradox' QBE interface, the user can compose a query, generate an SQL translation, and enhance it with SQL features not available in QBE.

Browsing

Advanced Revelation, DataEase/SQL, and Oracle forms can be used to browse a table. Selection criteria are translated into SQL queries, and if no criteria are entered, the program queries the entire table. The user can browse forward and backward but cannot move beyond the scope of the original query without entering another explicit query. Advanced Revelation and Oracle always work within a local query result set.

As in the other products, the DataEase/SQL user can enter selection criteria and step through records meeting the criteria. The F3 key accesses the next record in database order (based on the table's unique clustered index), and Shift-F3 steps backward. Each time the user changes the browse sequence or direction, the program executes a new query. In large tables (and depending on the user's location in the table), the F3 and Shift-F3 key-strokes can result in a very large query. Such large queries slow movement between records and can

abort owing to insufficient space in the temporary database (at the default size). Products working with query result sets rarely encounter this problem.

DataEase/SQL and Advanced Revelation can use Microsoft SQL Server's multiple threading capability to execute an update (or delete) query while a select query (on which the browse is based) is still active. Advanced Revelation generates an update or delete statement with a WHERE clause based on a record's primary key. DataEase/SQL bases its WHERE clause on all field values originally returned by the query.

DataEase/SQL can use Microsoft SQL Server's multithreading facility to access multiple tables with a single form. Predefined relationships can define a lookup for a single value in another table or define one-to-many relationships in a master-detail form. DataEase/SQL queries the base table, then queries additional tables using the base table values specified in the relationship definitions. Because the query on the base table is still active, the user can step to the next record in the result set and the related queries are executed again.

Embedded SQL

Advanced Revelation, DataEase/SQL, and Paradox programmers can embed SQL commands in procedures. Oracle Tools' SQL*ReportWriter bases reports on SQL commands explicitly written by the programmer. SQL*Forms generates implicit SQL commands for each form block, selecting all displayed database fields from the block's base table (without using WHERE clauses, which narrow the scope of a query). The programmer can add WHERE and ORDER BY clauses and triggers using embedded SQL. Oracle Tools' Pro*C interface supports embedding SQL into C programs. Pro*C offers a full implementation of standard SQL cursors, including the ability to update or delete the current row.

All the products can embed updates, inserts, deletes, and data definition language (DDL) statements such as Create or Alter Table, and all embedded statements can include the contents of program variables. All except DataEase/SQL can embed SQL Select statements to retrieve data into program variables. Although the DQL FOR command is powerful enough to substitute for embedded Selects in most cases, users cannot explicitly specify Select statement parameters such as Hold-

lock or For Browse, explicitly query multiple tables in a single command, use complex selection or join criteria in a WHERE clause, or use SQL language functions that have no DQL equivalent. Grouped queries are programmable in DQL but are much more difficult than in SQL.

Paradox offers an equivalent to SQL cursors, by which multiple row select statements can return and process query results a row at a time (rather than returning the entire result set). Paradox can optionally fetch successive rows into a program array. Paradox permits only one active SQL statement, so the "SQL cursor" option cannot be used for nested SQL operations (e.g., updating each row fetched). Nested SQL operations require a standard Select operation, and the user must step through rows in the answer table.

DataEase/SQL's DQL FOR clause can simulate embedded SQL cursors. The FOR clause includes a set of commands, each including additional database operations, executed once for each record meeting the specified criteria. Advanced Revelation returns the results of SQL Selects into a single multidimensional program array. Even when it is not feasible to store data on the server in multivalued format, data can still be retrieved into a multivalued format for use in Advanced Revelation applications.

Application Development

Advanced Revelation and Oracle Tools

Advanced Revelation and Oracle Tools provide powerful application development tools capable of virtually any developer activity. Advanced Revelation provides more features and flexibility but is geared less toward storing and accessing data in an SQL database. Advanced Revelation's report generator has a more intuitive interface than Oracle's, but it is more difficult to design reports that access more than one table. Oracle does not offer a sophisticated procedural language like Advanced Revelation's R/Basic, but its tools package includes an interface for embedding SQL commands in C programs. For many businesses, the choice between the two SQL front ends may be decided by the choice of SQL server (i.e., Oracle Server or Microsoft SQL Server). Either front end provides sufficient capabilities for developing sophisticated applications.

Both Advanced Revelation and Oracle Tools provide powerful forms designers to supplement the basic forms functions. Triggers can be executed upon entering or exiting a form, upon moving to or from a field, upon specific keystrokes, upon committing changes, or on a variety of other events. Triggers can perform computations or database operations. Advanced Revelation forms can call subroutines written in its powerful R/Basic language, and Oracle forms can call subroutines in C.

Advanced Revelation supports lookup formulas that access data in tables other than the form's base table. Oracle accomplishes this type of lookup using triggers. In both programs a form can call another form. Oracle forms can contain multiple blocks displaying different tables, with triggers defining relationships, passing data, and controlling movement between blocks. Advanced Revelation can display several forms concurrently, with relationships based on key fields or lookup formulas, and function keys controlling movement between them. Advanced Revelation forms can call pop-up windows offering a list of choices for field values or activity options.

DataEase/SQL and Paradox

DataEase/SQL outscores Paradox in NSTL's evaluation of SQL front-end features, but its database features are not as strong. DataEase/SQL incorporates standard DataEase features for use with SQL server data, but standard DataEase itself is not as powerful a product as its competitors. In contrast, Paradox provides fewer features than DataEase/SQL for accessing server data, but once the data has been accessed, Paradox offers more powerful application development capability.

DataEase/SQL excels at simple form-based applications for viewing, entering, and updating data. It effectively handles lookup fields and master-detail displays, and menus are easy to create. For more complex applications, its lack of program array variables forces the extensive use of temporary tables, slowing performance. Programmers will have difficulty enabling user interaction during procedures, and the results often seem awkward in appearance and function. Compared to other products, the DataEase developer has relatively little control over cursor movement, colors, and what options can be made available to the user.

What Paradox lacks in built-in facilities for directly viewing and editing server data can easily be programmed using basic PAL programming techniques. Using the Wait Record command, a developer can allow users to browse forward and backward through a query result set. A Function key can be designated to update the database using an easily written query based on the contents of the record's key fields. With little additional programming, the application can first determine whether the record has been updated by another user and automatically refresh the screen using a single-record fetch. Master-detail forms are only slightly more complicated to implement; Next Record or Previous Record keys trigger a query of the detail table using the next or previous master record's key values. Executing queries so frequently may seem like a lot of data retrieval, but it is no different from what DataEase/SQL and Oracle SQL*Forms do with their built-in features.

Cross-Platform Transportability

Many businesses need to port existing applications between environments or platforms. Applications written for one environment can easily be ported to another environment if its features work the same with native data and SQL data. Furthermore, a developer familiar with a product's programming techniques can begin writing applications for use with an SQL server using the same techniques.

Oracle Tools works only with Oracle databases, so applications require no modification when moved from one environment to another. The Oracle developer can create an application that addresses concurrency issues and run it on a standalone PC or with a server; the standalone system simply ignores commands related to concurrency.

DataEase/SQL supports SQL databases in addition to its own native data, and any application written for standard DataEase can be moved unaltered to a server environment. Fine-tuning may be needed to improve performance, especially if the application is set up with permanent data on a server and temporary tables maintained locally. For example, a DQL FOR clause comparing values in a local and a remote table may need to be reworked using temporary variables. The DataEase Lock command does not work with Microsoft SQL Server (which does not support explicit locking),

and some procedures may need to be restructured for optional concurrency control. Performance optimization and concurrency control issues are addressed in the documentation's discussion of database engines.

Advanced Revelation's environmental bonds effectively provide transparent access to all data types and indexing schemes in different environments. The program assigns an Advanced Revelation data dictionary to each table in every environment through which all applications access the data, and applications perceive all data to be in Advanced Revelation format. Problems arise only when an Advanced Revelation feature cannot be duplicated in the other environment (discussed under Data Interfaces and Indexes). Advanced Revelation supports the server's SQL dialect and offers its own SQL interface. Users who learn Advanced Revelation's version of SQL can compose generic queries that can access data from any data platform and even join tables across platforms.

At the heart of the benchmark application is a form displaying an order record and listing the books selected as part of each order. In standard Advanced Revelation, the order record includes a multivalued field containing the key values for the selections, with lookup fields based on those key values to display additional selection information. To accomplish the same effect with server data, access to the order record must trigger an R/Basic procedure using embedded SQL to retrieve the corresponding selections; posting of changes must trigger another embedded SQL procedure to update the selection records.

Despite a lack of features that work with both Paradox and server data, Paradox applications developed for use with SQL data are easily moved between server environments with little or no modification. Paradox queries are efficiently translated into the appropriate server's SQL dialect for optimal performance and concurrency control. Borland's benchmark application was developed for use with Microsoft SQL Server, and NSTL adapted it for use with Oracle Server with minor syntax changes to two embedded SQL statements.

Like Advanced Revelation, Paradox requires substantial modification to master-detail forms when moving from native data to an SQL server. With native data, scrolling through master records displays the detail records with no perceptible pause. An identical form can be used with server

data, but the form must reference two local query result tables. The program queries the master table, and each time the user moves to a new record in the result table, the corresponding detail records are queried and displayed. Lookups cannot be used with server tables. A PAL program can be written to execute the main query and, for each result record, execute a query to access the lookup value. The report developed by Borland (as the test application) uses a single multitable join query accessing all tables referenced in the report (producing the fastest report benchmark time).

Summary

DataEase/SQL and Paradox allow application developers to come up to speed quickly and offer usable tools for the nondeveloper. Advanced Revelation and Oracle Tools and Database offer more powerful application development capability and require time and training expenses to make use of that capability.

DataEase/SQL versus Paradox

An initial examination of the programs shows that many of Paradox' familiar features cannot be used with server data directly, while DataEase's familiar tools have been adapted to access server data as transparently as native data. Time and experimentation reveal that DataEase/SQL often hinders users with inefficient procedures, features that do not work as expected, and inexplicable errors causing tasks to abort. Examples include the ease of inadvertently initiating a full table query while browsing, inordinate delays moving between records, and incompatibilities with Microsoft SQL Server's date/time field.

Paradox' limitations become less significant with familiarity. While the inability to use the View command to browse a table can be annoying at first, with very large tables, practicality calls for browsing a subset of records based on selection criteria. Similarly, Paradox may not permit the use of its Edit command while browsing, but updates are easily accomplished in a PAL application using a local query result table with a Wait Record command or with an embedded SQL update statement. Paradox' data entry function works the same with server data as with native data. Most aspects of its

interactive query interface (the industry standard) work with server data, and for queries beyond its scope, SQL commands can be substituted. For more sophisticated applications, Paradox' PAL language provides considerably more flexibility in application design than does DataEase's DQL.

Advanced Revelation versus Oracle Tools

While no match for either Paradox or DataEase/SQL in terms of usability, Oracle Tools is far superior to Advanced Revelation in its documentation and the intuitiveness of its commands and interface, and it provides comparable sophistication and flexibility. In both products, forms can use triggered procedures and external program calls at virtually any point, giving on-line applications virtually unlimited capability. When used with its server bond, Advanced Revelation is prone to memory management problems, occasional crashes, and even data dictionary corruption in the hands of less than expert users. Oracle Tools seems solid and reliable in comparison.

Perhaps the most significant difference between the products is their capability to access SQL data. Oracle Tools was designed to work with SQL databases, whereas Advanced Revelation was originally intended to work with an entirely different architecture. Advanced Revelation's environmental bond concept for transparent access to different data types does not access SQL data as smoothly as does Oracle.

Advanced Revelation's use of multivalued fields to define relations between tables is quite different from SQL's relational model. Users must either simulate multivalued fields in the SQL database (the data cannot be queried using standard SQL joins) or abandon the program's standard application design methods in favor of very complex workarounds in the R/Basic.

Conclusion

The ability to access different data formats transparently with a single set of tools and applications can bring incalculable benefit to many installations. That goal underlies the development efforts of both Advanced Revelation, with its environmental bonds for a number of data types in addition to SQL databases, and DataEase/SQL, which

adapts an entire set of tools for accessing and manipulating SQL data. Both products work, but neither has the marks of a mature product. With time, these products should improve server data access, adding support for more server platforms. Both have the potential to make a significant contribution to the marketplace. At present, Oracle provides an excellent set of tools for a single database engine type, and Paradox works extremely well with limitations on its capability to manipulate server data.

Program Evaluations

Revelation Technologies Advanced Revelation

Strengths

- Flexible programming language
- Forms can include complex program logic
- Embedded SQL in procedures
- Two-phased Commit
- Long-text data format
- Announced support for multiple server platforms

Limitations

- Very difficult to learn
- No error message documentation
- Difficult to design forms and reports
- Difficult to define one-to-many relationships
- Prone to crash without warning
- Supports only Microsoft SQL Server at present

DataEase International DataEase/SQL

Strengths

- Browse and edit remote tables
- Easy to create multitable forms
- Excellent application portability
- Easy to learn
- Easy to join tables across platforms
- Two-phased Commit
- Announced support for multiple server platforms

Limitations

- Limited features for developing complex applications
- Operations with server data prone to unexplained errors
- Poor performance on multitable queries
- No SQL command mode
- Cannot use embedded SQL to select program variables
- Supports only Microsoft SQL Server and Sybase SQL Server at present

Oracle Tools and Database

Strengths

- Powerful forms designer
- Forms can call external programs
- Good SQL language documentation
- Good use of server's lock mechanisms in forms
- Excellent application portability

Limitations

- Supports only Oracle servers
- Difficult to design forms and reports
- Difficult to move between tool modules
- No word wrap in forms or reports
- No menu interface or Query By Example for interactive queries

Borland International Paradox

Strengths

- Currently works with several different servers
- Embedded SQL with row-by-row fetch option
- Excellent Query By Example that works with server data
- User can save/edit SQL query translation
- Easy data transfer between platforms
- Easy to create custom applications
- Easy to enter new records
- Flexible, easy concurrency options

Limitations

- Cannot view server tables directly
- Cannot edit server data directly
- Difficult to update server records
- Difficult to initially access existing server tables

Program Recommendations

Borland International Paradox

With support for a variety of platforms including three different SQL servers for OS/2, Paradox allows users to combine its state-of-the-art Query By Example interface and powerful PAL development language with any database server. Although not all of Paradox' standard facilities are available for accessing server data, simple PAL programming easily duplicates the built-in features of other products. Paradox commands translate efficiently into SQL and effectively take advantage of the server's performance enhancement and concurrency control features. Paradox provides the fastest data retrieval among the test products. Embedded SQL with optional single-row cursor fetches add flexibility to application development.

Oracle Tools and Database

Oracle's front-end tools work only with Oracle databases, but Oracle databases are available for practically any architecture and operating system. Although not as easy to learn or use as Paradox, Oracle Tools provides powerful tools for developing sophisticated applications. Using triggered procedures, SQL*Forms can create a single form with functionality that in other products would require extensive procedural code and several forms. Oracle Tools includes modules for producing menus and reports and for direct entry of SQL commands.

Rating Summaries

Figure 1.
Revelation Technologies Advanced Revelation Ratings

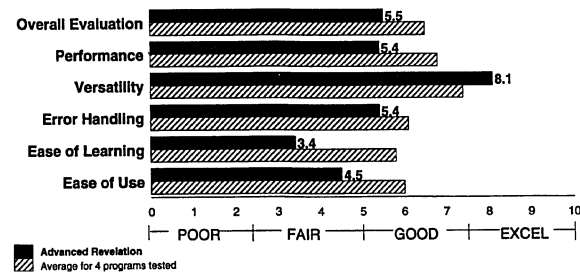


Figure 2.
DataEase International DataEase/SQL Ratings

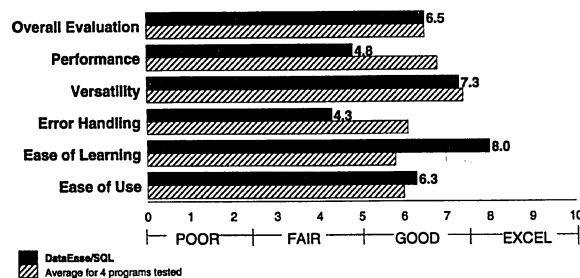


Figure 3.
Oracle Tools and Database Ratings

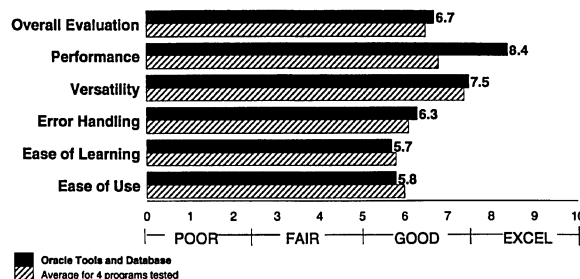


Figure 4.
Borland International Paradox Ratings



Overall Evaluation

Paradox ranks first in ease of use, error handling, and performance, but last in versatility. The program supports more server platforms than the other products, but many features available for native data cannot be used with server data. The inability to access certain features with server data requires some additional learning (even for established Paradox users) and programming workarounds. Testers rate Paradox' usability high because of its interactive queries and overall application development capabilities.

Oracle Tools' powerful application development tools work only with Oracle databases (the other front-end products work with multiple server types). New users will not find Oracle Tools particularly easy to learn, but it works the same in different database environments (e.g., local to server) and platforms (e.g., VAX to OS/2). Applications generated using SQL*Forms outperform all competitors; reports and interactive queries achieve average performance.

DataEase/SQL rates first in ease of learning, but last in performance and error handling. All standard DataEase/SQL features work with server data for excellent application portability, but advanced application development features are weak.

Advanced Revelation's many complex features make it the most difficult to learn and use. Excellent performance in custom applications is offset somewhat by mediocre performance in interactive queries and reports.

Methodology

The Overall Evaluation is a weighted average of scores for the individual criteria.

Overall Evaluation Score = (3 x Performance Score) + (6 x Versatility Score) + (2 x Error Handling Score) + (3 x Ease of Learning Score) + (6 x Ease of Use Score) + (Testers' General Evaluation Score) ÷ 21.

Performance

The user of an SQL front end experiences performance determined primarily by the database server's capability to manage database operations and the network's efficiency handling communication between the front end and server. Inherent differ-

ences between the front-end products have much less impact on performance than the server and network.

With NSTL's test configuration (optimized for relatively small numbers of users), front ends using Oracle Server for OS/2 perform faster than products using Microsoft SQL Server. The difference is especially noticeable when forms must retrieve and display multiple records; Oracle Tools' SQL*Forms and Paradox use Oracle Server's array fetch capability to accomplish these tasks. With large numbers of users, Oracle Server demands more server memory for caching than does Microsoft SQL Server.

With Oracle Server, custom applications in Oracle Tools' SQL*Forms tend to perform more efficiently than those in Paradox (although Paradox' performance is excellent). Custom applications perform less efficiently because of Paradox' need to download query results into local answer tables; in contrast, Oracle's SQL*Forms works only with Oracle (not with multiple servers). Paradox is more efficient with interactive queries and reports.

With Microsoft SQL Server, Advanced Revelation and Paradox custom applications perform comparably when Advanced Revelation can use R/Basic with embedded SQL to accomplish complex tasks. For interactive queries and reports, Paradox is clearly faster. DataEase/SQL performs competitively in relatively simple operations; more complex tasks are slowed by inefficient translation to SQL, execution of multilevel DQL FOR statements, and extensive use of temporary tables to manipulate data.

Methodology

Each product's performance score is based on its average speed executing seven functions using an on-line book order entry application and three interactive queries.

Order Entry Score = (1 x Select on ISBN Test) + (1 x Select on Author Name Test) + (1 x Select on Title Test) + (3 x Post Order Test) ÷ 6.

Entry Applications Score = (3 x Order Entry Test) + (2 x Payment Test) + (1 x Shipment Test) + (2 x Order Report Test) ÷ 8.

Interactive Queries Score = (2 x Single-Table Query Test) + (1 x Three-Table Join Query Test) + (1 x Grouped Query Test) ÷ 4.

Versatility

Despite its current support for only one server type, Advanced Revelation ranks near the top in virtually all the versatility categories with special strength in application development tools and support for server data types.

Close versatility scores belie important differences in the features provided by Oracle Tools for MS-DOS and DataEase/SQL. DataEase/SQL's focus on connectivity and server-specific features indicate plans to support a number of server types; tools for developing complex applications are lacking. Oracle Tools' application development tools are comparable to those found in Advanced Revelation, but Oracle Tools works only with Oracle databases.

Paradox supports the largest number of different server platforms and makes the best use of server concurrency control features. Its overall versatility is weakened by the large number of features that work with Paradox data, but not with server data.

Methodology

NSTL compared each product against a master list of SQL front-end features and verified support for each item. Weighted features are grouped into weighted versatility categories. Versatility Score is a weighted average of scores for the individual criteria.

$$\text{Versatility Score} = (\text{Data Types Score}) + (\text{Tables and Views Score}) + (2 \times \text{Data Validation Score}) + (2 \times \text{Interactive Query Score}) + (3 \times \text{Connectivity Score}) + (2 \times \text{Application Portability Score}) + (\text{Server-Specific Features Score}) + (3 \times \text{Concurrency Score}) + (\text{Query Language Functions Score}) + (2 \times \text{Procedures Score}) + (3 \times \text{Forms Score}) + (2 \times \text{Reports Score}) + (\text{Application Development Score}) \div 24.$$

Error Handling

Paradox' error handling capability includes good documentation of error messages, programmable recovery routines, a debugger, and good concurrency controls. Concurrency features include optional locks on queried records without using embedded SQL and Ctrl-Break cancellation of server operations, returning control to the user. When the user attempts to write to a full disk, Paradox exits to a DOS shell and instructs the user to

delete unwanted files; after the user deletes files, the program continues the write operation.

Oracle Tools features good, but not comprehensive, documentation of error messages, the capability to track and fix runtime errors (with some difficulty), and good concurrency controls. Error message documentation covers only some of the tool modules. Users can browse records during transaction updates, and Oracle locks records when changes are entered. Oracle provides clear update notification to users attempting to modify a record that has been changed by another user. The program does not specifically identify I/O errors, but the user can recover and continue working.

Advanced Revelation provides a program debugger and programmable error handling/recovery using existing and custom error codes. Error messages often go unexplained. During testing, Advanced Revelation experienced frequent system crashes, occasionally losing data or dictionary files and preventing program reentry. The Advanced Revelation user can set a lock time-out interval after which remote operations are canceled.

DataEase/SQL documents error messages in a manual appendix, but server operation errors are often explained only with a vague "transaction failed" message. The program supports programmable recovery routines with server error codes and server-specific deadlock messages. DataEase/SQL does not come with a debugger, but it flags syntax errors and invalid table or field names as procedures are written (minimizing runtime errors). DataEase returns error messages when records have been updated, but without displaying an explanation. I/O error handling can be problematic. Users must abort to DOS when printing to a printer that is "not ready," and memory errors commonly necessitate aborting to DOS. Browsing backward through a large table can take up to a minute, and the temporary server database can run out of space. A DataEase/SQL date or time variable can access a Microsoft SQL Server date/time field, but DataEase/SQL cannot update the record if the field contains both a date and a time.

Methodology

NSTL evaluated the products' on-line error messages, error message documentation, programmable error recovery capabilities, debugging features, concurrency control and transaction processing

options, and implementation of error handling/recovery features (i.e., bugs encountered). On-line tests assessed handling of disk and printer I/O errors and concurrent browsing and transaction processing. The Error Handling Score is a weighted average of scores for the individual criteria.

Error Handling Score = (3 x Error Messages Score) + (Debugger Score) + (4 x Concurrency Score) + (4 x Implementation Quality Score) + (3 x I/O Errors Score) ÷ 15.

Ease of Learning

DataEase/SQL offers the best combination of intuitive on-line functions and clear, well-organized documentation. Users can learn most tasks by navigating through the menu system and choosing among the options presented at virtually every point. Information in the manuals is accessible and detailed almost to the point of redundancy. Users familiar with the DataEase standalone version will have to learn only a few new options (e.g., for creating new tables, creating forms from existing tables, transferring data between local and server tables). A separate manual section covers concurrency control with respect to specific database engines.

Paradox offers an intuitive menu system and query interface, but the manual often lacks detailed descriptions of command functions and keywords. Unless the manual provides an example for a specific operation, users may have to resort to trial and error. Standalone Paradox users must learn to create "replicas" for accessing existing server tables. Some standard Paradox features do not work with remote tables, so users will have to develop new methods for obtaining comparable results. The DataEase/SQL manual for the SQL Link component is largely clear and well organized, but testers had trouble finding the section on accessing existing server data.

Oracle Tools provides separate manuals for its various components. The SQL*Plus manual is excellent, but the SQL*Forms and SQL*ReportWriter manuals are inadequate. Oracle Tools users must rely on trial and error to format reports and to get complex forms to function as desired. Oracle works almost identically in server and standalone environments.

Inadequate indexing and confusing organization characterize the Advanced Revelation manual. Users will have difficulty finding information, and concepts are not completely explained. The lengthy and detailed on-line tutorial focuses more on what keystrokes complete an exercise than on explaining the product's features. An addendum to the documentation deals specifically with the Server Bond component. Instructions for accessing the server are very clear, but instances when Revelation and server data must be handled differently lengthen the learning curve, and the new material is not indexed.

Methodology

Usability testers evaluated the adequacy of the programs' manuals as learning tools and attempted to use the manuals and on-line Help to learn administrative tasks, querying, report generation, forms design, and application development. Testers evaluated each product's overall ease of learning and the "transferability" of knowledge from the standalone version to a server environment. Scores for each of the following categories are weighted averages of General Learning (weight 3) and Transferability (weight 1).

Ease of Learning Score = (4 x Manuals/Learning Aids Score) + (Administration Score) + (3 x Querying Score) + (2 x Report Generation Score) + (2 x Forms Design Score) + (3 x Application Development Score) ÷ 15.

Ease of Use

Novices and experienced application developers will like Paradox' options for accessing SQL data. Ad hoc querying is especially convenient. Alternatives include Query By Example, direct entry of SQL commands using MiniScript, and the ability to save a Query By Example as an SQL command with editing capability.

Paradox' View and Edit facilities cannot be used to access and manipulate remote data, so users can work only with local query result tables. Posting changes to existing records requires extra programming in Paradox; the other products use a simple data entry form. For applications, the results of a query can be routed all at once to a local table or fetched and processed a row at a time. Once data has been queried into local tables, forms

can access and manipulate data with great flexibility using the PAL Wait command. Generic and server-specific error codes can be referenced in user-defined error handling routines. Although programming is required to update existing data on the server, new data can be entered using forms.

DataEase/SQL supports the full range of local data operations on server data as well. Predefined relationships between server tables permit forms with lookup fields and one-to-many relationships. Forms can be used to browse and edit server data. Function keys can be used to browse forward and backward, but moving backward in a large table can cause unreasonable delays and may cause the server's temporary database to run out of space (allocated during installation).

DataEase/SQL queries are actually quick reports, and users must format output in order to view results of even simple queries. Having grouped queries display only group totals and not individual record values is especially complicated. In more complex applications, it is difficult to work with a subset of records based on selection criteria without loading that data into temporary tables.

Ad hoc queries in Oracle Tools can be entered only with SQL commands, and SQL*Plus provides no facility for browsing query results. Forms and reports provide a friendlier interface for viewing data, but the developer must navigate a series of difficult interfaces to create them.

SQL*ReportWriter provides no WYSIWYG facility for laying out reports. SQL*Forms triggers provide almost unlimited flexibility, but complicated steps are required for anything beyond viewing and editing a single table. Oracle Tools consists of separate modules accessed independently from DOS; a Client Manager menu interface does not significantly improve the inconvenience of moving between different modules.

Advanced Revelation offers a menu interface for building simple forms and reports and an SQL command window similar to Oracle's. An extensive menu interface provides access to most functions, and experienced users can work directly from a command prompt. Like Oracle, Advanced Revelation allows almost unlimited flexibility, with considerable navigation through options, moving between components, and programming.

Methodology

Testers evaluated the ease of using each program for administrative tasks, querying, report generation, forms design, application development, and porting an application to another environment. The Ease of Use Score is a weighted average of scores for the individual criteria.

$$\text{Ease of Use Score} = (\text{Administration Score}) + (3 \times \text{Querying Score}) + (2 \times \text{Report Generation Score}) + (2 \times \text{Forms Design Score}) + (3 \times \text{Application Development Score}) + (2 \times \text{Application Portability Score}) \div 13.$$

Performance Results

Order Entry Database

NSTL tested SQL front-end programs with a book order entry database. A Book table contains one record for each book in inventory. Each record provides a unique ISBN number, book title, price, and inventory information.

For each author and coauthor of books in inventory, an Author table contains records listing a name and unique ID number. Link table records link each author to his or her books on the basis of author ID and ISBN number.

An Order table contains records for orders placed, each with a unique order number and customer information. An Entry table contains one or more records per order, each record corresponding to a book order.

The database contains information on 25,000 books and 5,000 authors; each book has 2 coauthors; 50,000 link records pair each author with 10 books. When testing begins, the Order table contains 10,000 orders and 50,000 entries (5 entries each).

Hardware Configuration

SQL front-end programs were tested on a Compaq Deskpro 386/33e database server with 16M bytes of RAM, connected to a Compaq Deskpro 386/25 workstation with 5M bytes of RAM. The workstation ran MS-DOS 3.31; the database server ran IBM OS/2 Standard Edition 1.2. The network used Microsoft LAN Manager 2.0 over IBM Token-Ring connections. Except for concurrency control, multiple workstations would impact server performance rather than the front-end performance. NSTL addresses concurrency control in its versatility and error handling evaluations.

Advanced Revelation and DataEase/SQL were tested with Microsoft SQL Server 1.1, and Oracle Tools for MS-DOS was tested with Oracle Server for OS/2 1.0. Paradox was tested with both SQL servers, and its performance score was averaged twice for each test. With both SQL servers, client/server communication uses the Named Pipes protocol.

DataEase/SQL, Paradox, and Oracle Tools were configured to use extended memory. When Paradox was tested with Oracle Server for OS/2, Oracle's Named Pipes driver ran in real memory to make all the extended memory available to Paradox. Advanced Revelation accessed expanded memory using Quarterdeck's QEMM expanded memory manager.

Benchmarks

The performance test suite included on-line book order entry applications and interactive queries. Overall, most observable performance differences are attributable to the SQL server rather than the front-end program. Performance factors at the front end include the product's efficiency at performing complex operations on selected data (measured by order entry applications) and its efficiency translating proprietary data manipulation functions into SQL for execution on the server (measured using interactive queries).

Vendors were given the opportunity to develop an application in which users select and post book orders, maintain and update order information, and report on orders and inventory. Test applications used each product's forms and reporting features and used combinations of embedded SQL and the proprietary query language at the vendor's discretion. The vendor-supplied application was

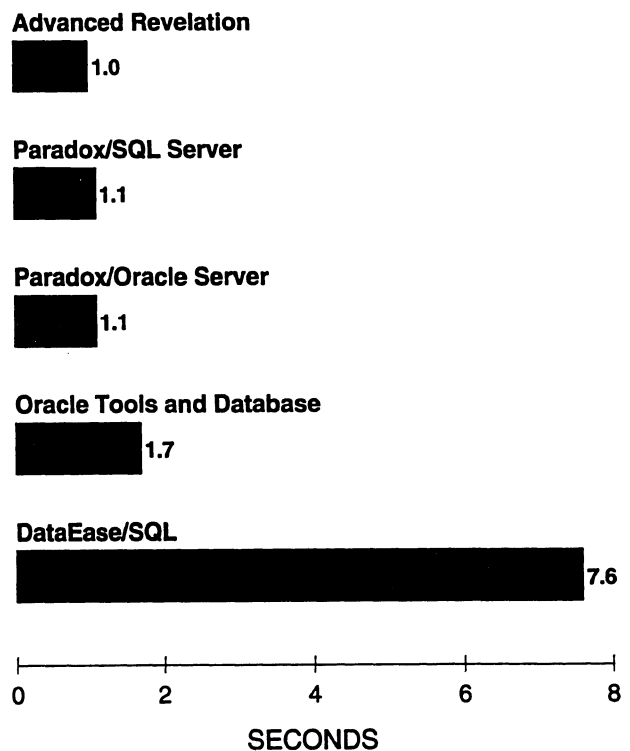
installed on the workstation, and tests were run based on the vendor's instructions. The Oracle Tools test application was developed by NSTL.

Interactive queries composed and executed by NSTL technicians used the products' generic query facilities (i.e., can be used with local or remote data and with any supported server platform): Paradox' Query By Example interface, DataEase/SQL's Quick Reports/QBE interface, Advanced Revelation's SQL command window, and Oracle Tools' SQL*Plus.

Select on ISBN Number

Each product finds and displays all Book record information for a selected ISBN number with its author/coauthor names (i.e., matching author ID and ISBN numbers in the Link table). The user enters the number of books to be ordered, and the record is saved, either in memory or in a temporary table, for later posting. The test time includes the times required to retrieve and display book and author information and save to a temporary table (if applicable).

Figure 5.
Select on ISBN Number



Analysis

The products execute two separate queries: one to retrieve the book record and one to match author names. Except for DataEase/SQL, the products have similar capability to retrieve and display the information. DataEase/SQL must insert the information into temporary tables before displaying it in a form, and selection information is stored in a temporary table (rather than in memory) for subsequent posting.

Select on Author Name

The program selects and displays the names of all authors corresponding to a partial author name entered by the user. The user selects one name, and the program lists all corresponding book titles (i.e., matching ISBN numbers and author IDs in the Link table). The user selects a title, and the program retrieves and displays all Book and Author information as in the first benchmark, Select on ISBN Number. The user enters the number of books to be ordered, and the record is saved, either in memory or in a temporary table, for later post-

Figure 6.

Select on Author Name

Oracle Tools and Database

2.4

Paradox/Oracle Server

3.2

Advanced Revelation

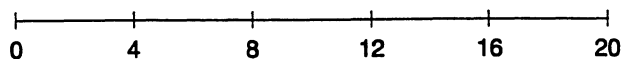
5.2

Paradox/SQL Server

7.3

DataEase/SQL

19.1



SECONDS

ing. The test time includes the times required to list authors, list titles, display information for a single book, and save to a temporary table (if applicable).

Analysis

The products using Oracle Server for OS/2 (Oracle Tools and Paradox) look up the author name and titles (for a single author) substantially faster than products using Microsoft SQL Server (including Paradox), probably because Oracle Tools' SQL*Forms and Paradox' Query By Example use array fetches to manage multirecord query results on the server. Paradox' complex process for placing query results into forms slows its performance slightly with both Oracle Server for OS/2 and Microsoft SQL Server. DataEase/SQL performs the author name lookup on a par with Advanced Revelation and Paradox, but it translates multilevel DQL FOR clauses into SQL and executes the title lookup much less efficiently than the other products' three-table joins. (DataEase/SQL cannot use embedded SQL to select database information into program variables.) DataEase/SQL is also slowed by its use of temporary tables as described in the analysis for the first benchmark, Select on ISBN Number.

Select on Title

The program selects and displays the titles of all books corresponding to a partial title entered by the user. The user selects a title, and the program retrieves and displays all Book and Author information as in the first two benchmarks, Select on ISBN Number and Select on Author Name. The user enters the number of books to be ordered, and the record is saved, either in memory or in a temporary table, for later posting. The test time includes the times required to list the titles, display information for a single book, and save to a temporary table (if applicable).

Analysis

The results demonstrate the relationships between the products as established in the second benchmark, Select on Author Name.

Figure 7.
Select on Title

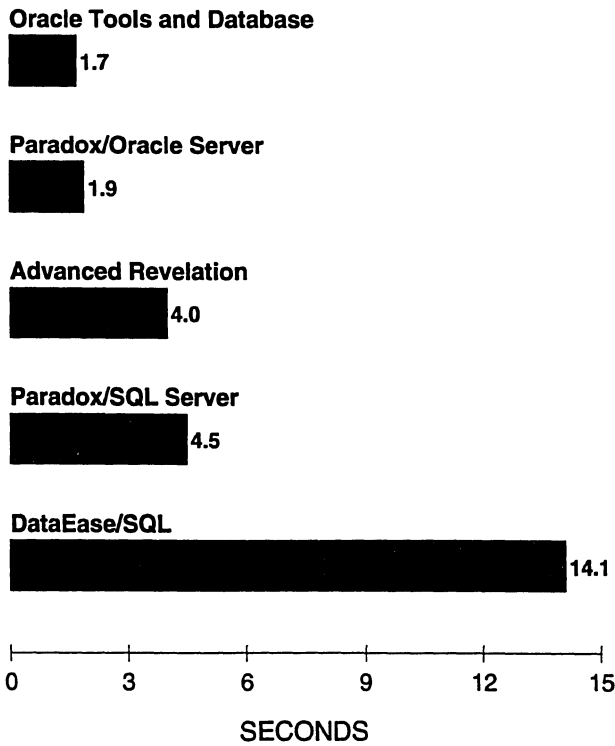
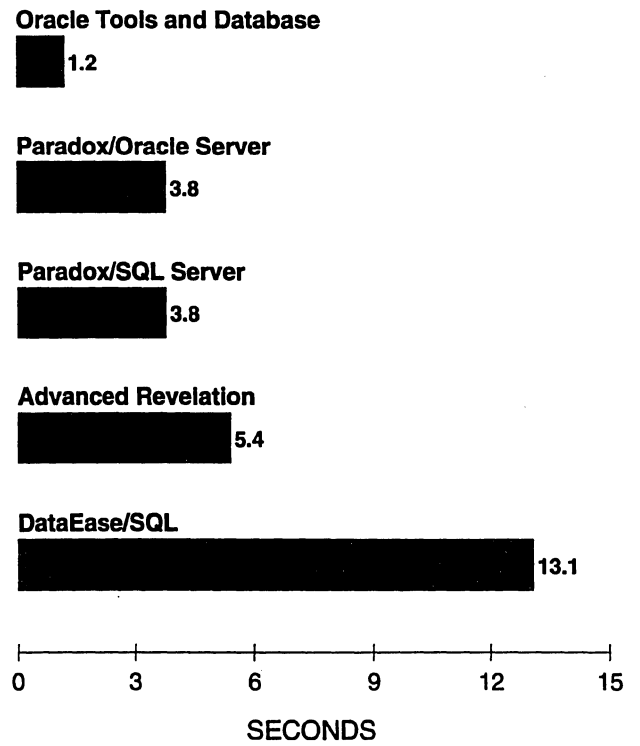


Figure 8.
Post Order



Post Order

The program displays the ISBN numbers, quantities, and prices of the three books selected in the first three benchmarks. The user enters customer and order information, and the program inserts one record into the Order table, inserts one record into the Entry table for each book selected, and updates the Book record for each book selected, increasing the value in the Sold field by the quantity ordered. The database increments the current order number, which it includes in the inserted Order and Entry records (when Order information is first displayed or at posting). The test time includes the times required to increment the order number, retrieve and display selected book information, and post the order.

Analysis

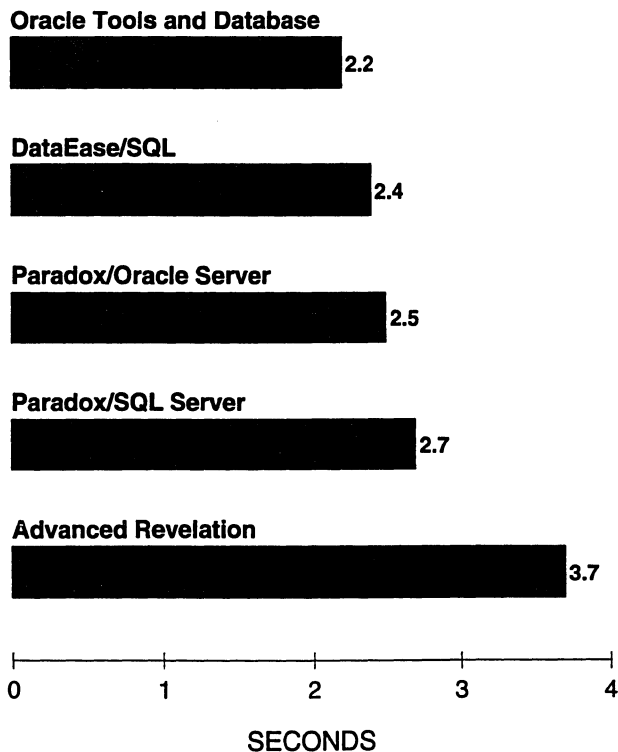
Unlike products designed to work with a variety of back-end platforms, Oracle Tools' SQL*Forms communicates with the database while data is being entered, resulting in extremely fast, permanent posting of data. Advanced Revelation posts new records slightly faster than Paradox but takes

slightly more time to initially load the form. DataEase/SQL loads book selection data from a temporary table and takes significantly longer to write the new records. Furthermore, DataEase/SQL cannot utilize all of Microsoft SQL Server's optimization features. For example, Advanced Revelation and Paradox can build an SQL statement dynamically; in this case, building all the inserts and updates into a single command string sent to the server in a single operation.

Payment

The program retrieves and displays the Order record and all corresponding Entry records for a selected order number. The user changes the payment status on-line, and the program updates the Order record. The test time includes the times required to retrieve and display the Order and Entry information and post the change to the Order record.

Figure 9.
Payment



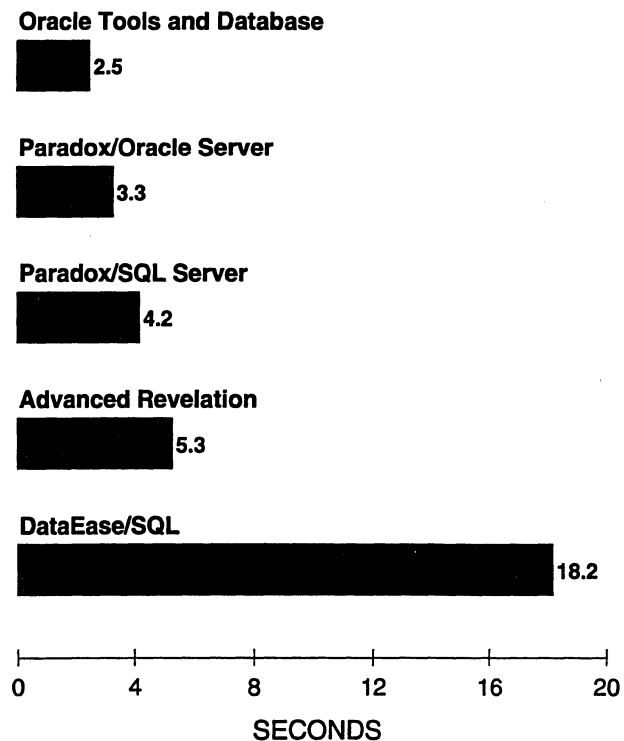
Analysis

DataEase/SQL is at its best with a simple operation of this sort, where it can browse server data directly rather than working with query result tables. It quickly retrieves and displays all the records in a master-detail form based on pre-defined relationships, and it updates the Order record faster than any of the products except Oracle Tools.

Shipment

The database maintains a current shipment number (the order number of the last order shipped). The program increments this number and retrieves and displays the Order record and all corresponding Entry records. The user changes the shipping status on-line, and the program updates the Order record and the Book record corresponding to each Entry record (i.e., matching ISBN numbers), subtracting the Entry quantity from the Books Sold and In Stock values. The test time includes the times required to increment the shipment number, retrieve and display the Order and Entry records, and post changes to the Order and Book records.

Figure 10.
Shipment



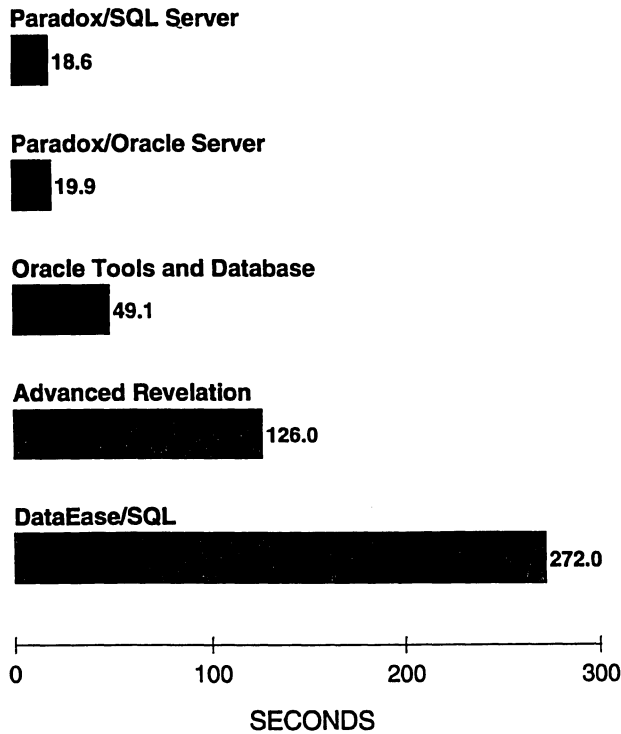
Analysis

Oracle Server's sequence generator maintains the current shipment number and enables faster performance for this type of operation (Microsoft SQL Server maintains the shipment number in a database table). Oracle Tools gains an edge over Paradox by browsing the tables directly without using query result tables. DataEase/SQL's use of temporary tables and its inability to use optimization features for multiple inserts and updates hamper its performance.

Order Report

All Order records with order numbers between 0 and 50 are processed in order by customer name with all corresponding Entry (matching order numbers) and Book records (matching ISBN numbers). The report also retrieves all Author records for each book by joining Link and Entry tables on ISBN number, and the Author and Link tables on author ID. Data for each order (one order per page) consists of the order number, customer name, and, for each entry, the ISBN, book title, and all authors. The report is written to an ASCII text file.

Figure 11.
Order Report



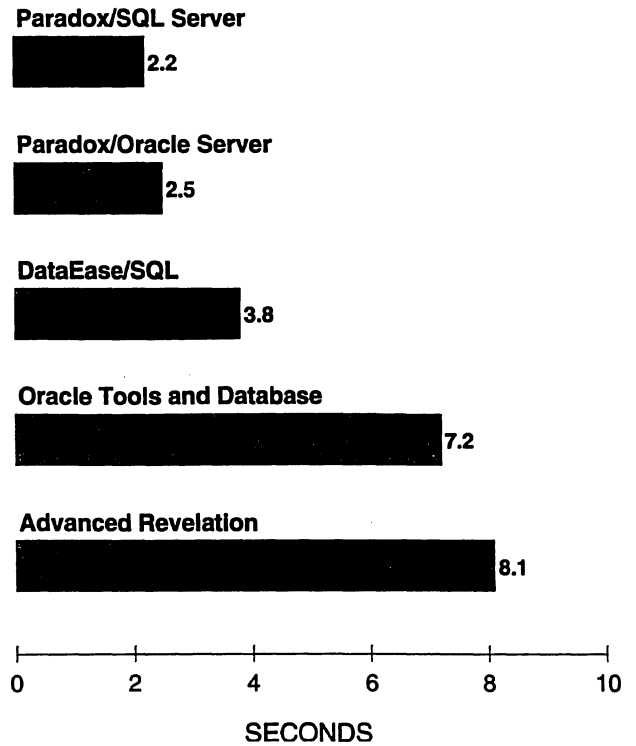
Analysis

Paradox' single five-table join query proves faster than Oracle Tools' SQL*ReportWriter, which requires three levels of nested queries to accommodate the three levels of grouping. Advanced Revelation uses essentially the same embedded SQL query as Paradox, so performance differences are isolated in multirow data fetching, formatting, and output file handling operations. DataEase/SQL's translation of five levels of nested DQL FOR clauses into SQL results in the least efficient server execution among the products.

Single-Table Query

A query selects the order number, customer name, payment status, and shipping status for all order numbers between 50 and 99.

Figure 12.
Single-Table Query



Analysis

Oracle Tools' SQL query is essentially the same as the SQL translation of Paradox' Query By Example, but Oracle Tools' SQL*Plus is slower writing output to the screen. DataEase/SQL's single-table query (a single DQL FOR statement) is processed much more efficiently than more complex operations in the other tests. Advanced Revelation is surprisingly slow processing interactive queries (considering that they are entered as SQL commands).

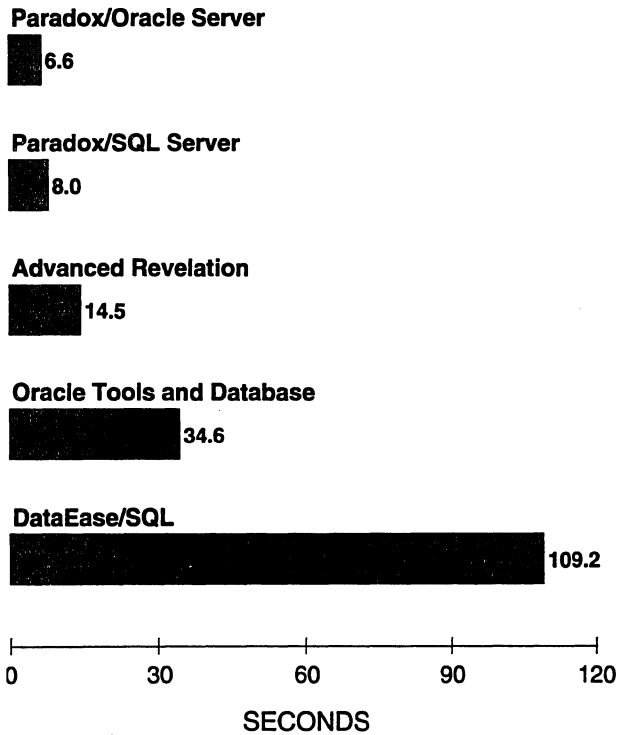
Three-Table Join Query

A query selects all author names and book titles corresponding to a partial author name string. The Author and Link tables are joined on author ID, and the Link and Book tables are joined on ISBN number. The query returns 200 records.

Analysis

Performance results are consistent with the single-table query, except that DataEase/SQL translates three levels of nested DQL FOR clauses much slower than it translates the single FOR statement in Benchmark H.

Figure 13.
Three-Table Join Query



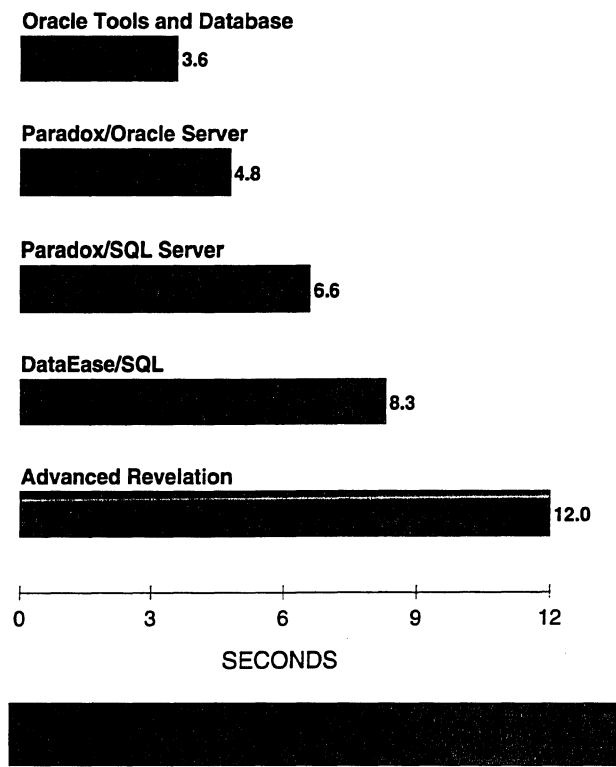
Grouped Query

The products list ISBN values, quantity in stock, quantity sold, number of matching Entry records, and the sum of prices in those matching records for all Book records with ISBN numbers below a specified number.

Analysis

Grouped queries are relatively complex in relation to the amount of data returned, so Oracle Tools' processing power is not offset by its slow screen writing speed. DataEase/SQL appears to be much more competitive with other products on two-table queries (even when grouped) than with three-table queries.

Figure 14.
Grouped Query



Vendors

Borland International, Inc.
1800 Green Hills Road
Scotts Valley, CA 95066 (408) 438-8400

DataEase International
7 Cambridge Drive
Trumbull, CT 06611 (203) 374-8000, (800) 243-5123

Oracle Corp.
500 Oracle Parkway
Redwood Shores, CA 94065 (415) 506-7000

Revelation Technologies, Inc.
Two Park Avenue
New York, NY 10016 (800) 262-4747

Characteristics

Table 1. SQL Front-End Characteristics

	Requirements	Operating Systems	Platforms	Supplier Support
Borland International Paradox	Drives: Hard disk (1M-byte space); Memory: 1M byte (OS/2 EE, MS SQL Server); 1.5M bytes (Oracle Server); System: 80286, 80386; Mouse: Not supported	IBM LAN Manager, LAN Server; Novell NetWare; 3Com 3+Open, and others	IBM OS/2 Data Manager; Oracle 6.0; Microsoft SQL Server	Progress Program for corporate customers; Borland SQL Partners for system integrators/VARs; Paradox Developer's Program available. Training available through Borland or third-party referrals
DataEase International DataEase/SQL	Drives: 10M-byte hard disk; Memory: 2M bytes for DOS; 4M bytes for OS/2; System: 80286, 80386; Mouse: Not supported	Microsoft LAN Manager; Novell NetWare 286, NetWare 386; DOS Extender; OS/2 Requestor Named Pipes	Microsoft SQL Server for OS/2; Sybase SQL Server; OS/2, UNIX, VAX	Telephone support; bulletin board service, quarterly newsletter; advanced technical support (45 days free; \$2,000 per year). Training available; call (800) 243-5123 for details
Oracle Tools and Database	Drives: Hard disk (15M-byte space); Memory: 640K bytes and 896K bytes EMM; 2.5M bytes EMM for SQL*Report-Writer; System: 80286, 80386; Mouse: Not supported	DOS 3.0 and higher, OS/2	Over 90 platforms	Package including telephone support, newsletter, and upgrades (35 percent of product list price); upgrades available (20 percent of product list price); CompuServe bulletin board. Training available; call ORACLE1 for details
Revelation Technologies Advanced Revelation	Drives: Hard disk (5M-byte space); Memory: 640K bytes; 2M bytes (1M byte EMS) recommended; System: 80286; 80386 recommended; Mouse: Supported; not required	IBM LAN Server; Novell NetWare; Banyan VINES; 3Com 3+Open; Microsoft LAN Manager and compatibles; MS-Net compatibles	Microsoft SQL Server; Sybase SQL Server	Telephone support (free for 90 days; \$400 per year), bulletin board service, magazine; Premium Support (\$3,500) includes publications, maintenance releases, and other services; technical bulletins (\$50-\$80 per year); maintenance release 1.02 (\$25 plus S&H); upgrade prices vary. Training available

Table 2. SQL Front-End Features

	Weight	Borland International Paradox	DataEase International DataEase/SQL	Oracle Tools and Database	Revelation Technologies Advanced Revelation
Field Data Types					
Fixed-Length Character	1	▲	▲	—	E
Variable-Length Character	1	▲	E	▲	▲
Integer	1	▲	▲	A	—
Small Integer	1	▲	▲	A	▲
Floating Point	1	▲	▲	A	▲
Fixed Point or Money	1	▲	▲	A	▲
Bit or Logical	1	▲	—	—	—
Long Text	2	▲	—	D	—
Long Binary	2	▲	—	D	—
Date	1	B	B	F	C
Time	1	B	B	F	—
Date/Time	1	▲	—	F	C
User Defined	1	E	—	—	E
Tables and Views					
Create Remote Table	1	▲	▲	▲	▲
Alter Remote Table	1	H	▲	▲	H
Copy Table Structure	1	▲	▲	▲	▲
Access Existing Table	1	▲	▲	▲	▲
Access Table with No Unique Key	1	▲	▲	▲	▲
Import Data into Remote Table	1	▲	▲	I	J
Create Index	1	▲	▲	▲	L
Use Existing Index	1	▲	▲	▲	▲
Multifield Indexes	1	▲	▲	▲	▲
Create View on Server	1	H	G	▲	H
Access Existing View	1	▲	▲	▲	▲
Create/Access Grouped Views	1	▲	▲	▲	▲
Update View on Server	1	▲	▲	▲	▲
Automatic Resynchronization	1	—	▲	—	—
Data Validation					
Null/Not Null	2	▲	▲	▲	▲
Range Validation	1	N	N	Q	R
Picture Validation	1	N	N	Q	R
Set Validation	1	N	N	Q	—
Default to Calculation	1	N	N	Q	R
Default to Table Lookup	1	N	N	Q	R
Lookup Validation (foreign key)	2	N	N	Q	R
Prevent Update/Delete (referential)	2	N	N	Q	—
Cascaded Update/Delete (referential)	2	N	N	Q	—

See Footnotes on page 329.

Table 2. SQL Front-End Features (Continued)

	Weight	Borland International Paradox	DataEase International DataEase/SQL	Oracle Tools and Database	Revelation Technologies Advanced Revelation
Interactive Query					
Query By Example	2	▲	▲	—	▲
Query Results to Local Table	2	▲	P	X	▲
Browse Remote Table	2	▲	▲	▲	—
Browse Query Results	2	▲	▲	T	▲
Update/Delete Remote Table in Browse	2	▲	▲	▲	—
Set Level Update/Delete	2	▲	P	▲	▲
Copy Records, Local to Remote	2	▲	▲	X	▲
Join Remote Tables	2	▲	▲	▲	▲
Join Local and Remote Tables	2	▲	▲	X	—
Grouped Query with Aggregates	1	▲	▲	▲	▲
Aggregates in Selection Criteria	1	▲	P	▲	V
Outer/Inner Joins	1	▲	▲	▲	▲
Allow/Supress Duplicates	2	▲	▲	▲	▲
Null/Not Null	2	▲	▲	▲	▲
Join on Nonequality	1	▲	—	▲	▲
Store/Reuse Query with Variables	1	▲	▲	▲	▲
Store/Reuse Query with User Input	1	▲	P	▲	P
View/Modify SQL Translation	1	▲	W	S	▲
SQL Command Mode	2	▲	—	▲	▲
Connectivity					
Gupta SQLBase Server	2	—	—	—	—
IBM OS/2 Extended Edition	2	Z	Z	—	▲
Oracle SQL*Net	3	Z	Z	▲	▲
Microsoft SQL Server	2	▲	▲	—	▲
IBM DB2	2	Z	—	▲	Z
RDB	1	Z	—	—	▲
Other Mini/Mainframe Databases	1	—	B1	—	C1
Switch Between Remote Databases	2	▲	▲	▲	▲
Switch Between Remote Platforms	2	A1	▲	—	▲
Access Multiple Remote Databases	1	▲	▲	DL	—
Access Multiple Platforms	1	A1	▲	DL	—
Two-Phased Commit	2	▲	▲	—	—
Simultaneous Local/Remote Access	2	▲	▲	DL	▲
Local Sessions	2	▲	▲	▲	▲

See Footnotes on page 329.

Table 2. SQL Front-End Features (Continued)

	Weight	Borland International Paradox	DataEase International DataEase/SQL	Oracle Tools and Database	Revelation Technologies Advanced Revelation
Application Portability					
Generic Forms	2	D1	▲	▲	E1
Generic Reports	2	D1	▲	▲	E1
Generic Interactive Query	2	▲	▲	▲	▲
Generic Procedural Language	2	▲	▲	▲	F1
Generic Error Recovery	2	▲	—	▲	▲
Macintosh Version	1	—	—	▲	—
OS/2 Version	1	—	▲	▲	—
Server-Specific Features					
Oracle Array Variables	2	—	—	G1	G1
SQL Server Stored Procedures	1	▲	K1	—	▲
SQLBase Chained Commands	1	—	—	—	—
OS/2 ARI	2	—	—	—	G1
Oracle OTEX	1	—	—	—	—
OS/2 Bind Prior to Runtime	0	—	—	—	—
OS/2 Record Blocking	0	—	—	—	—
SQL Server Browse Mode	1	H1	G1	—	—
SQLBase Result Sets	0	—	—	—	—
OS/2 Foreign Keys	0	—	—	—	—
SQL Server Rules/Defaults	1	J1	J1	—	—
SQL Server System Procedures	1	J1	G1	—	I1
SQLBase Context Preservation	0	—	—	—	—
SQL Server Select Info/Temp Tables	1	H1	G1	—	H1
Oracle Sequence Generator	1	—	—	H1	—
Multiple Threads	1	▲	▲	—	—
Concurrency					
Transactions	1	▲	▲	▲	▲
Auto Commit on/off	1	▲	▲	▲	▲
Interactive Transaction Support	1	▲	—	▲	▲
Concurrent Data Entry	1	▲	▲	▲	▲
Concurrent Browse/Edit	1	▲	▲	▲	P
Concurrent Browse/Transaction Processing	1	▲	▲	M1	P
Concurrent Query/Transaction Processing	1	▲	▲	▲	▲
Concurrent Report/Transaction Processing	1	▲	▲	▲	▲
Snapshot Read	1	—	—	▲	▲
Update Notification	1	—	L1	▲	—
Screen Refresh	1	—	▲	—	—
Read with No Lock	1	▲	▲	▲	▲
Hold Lock	1	▲	▲	▲	▲
Deadlock Management	1	▲	▲	▲	▲
Break Lock	1	N1	—	—	▲

See Footnotes on page 329.

Table 2. SQL Front-End Features (Continued)

	Weight	Borland International Paradox	DataEase International DataEase/SQL	Oracle Tools and Database	Revelation Technologies Advanced Revelation
Query Language Functions					
Aggregate Functions	1	▲	▲	▲	▲
Date Comparisons	1	▲	▲	▲	▲
Date Arithmetic	1	▲	▲	▲	P1
Extract Date Part	4	Q1	▲	▲	P1
Time Functions	4	Q1	▲	▲	—
Substring	1	▲	▲	▲	P1
Substring Position	1	▲	▲	▲	P1
String Length	1	▲	▲	▲	P1
Concatenation	1	▲	▲	▲	▲
Arithmetic Operators	1	▲	▲	▲	▲
If/Null/Then	1	▲	▲	▲	P1
If/Boolean/Then/Else	1	▲	▲	▲	P1
Function Nesting	1	▲	▲	▲	P1
Procedures					
Embedded SQL	2	▲	R1	T1	▲
Dynamic SQL	2	▲	R1	U1	▲
SQL Cursors	2	▲	S1	U1	▲
Update/Delete WHERE CURRENT	1	—	S1	U1	—
Nested Selects	2	▲	▲	U1	—
Preserve Context Across Commands	1	▲	▲	U1	—
Array Variables	2	▲	—	U1	▲
Multirow Array Operations	1	▲	—	U1	—
Procedural Logic	3	▲	▲	U1	▲
Screen Painting in Procedures	2	▲	—	—	▲
Call Forms from Procedure	2	▲	▲	U1	▲
Print from Procedure	2	▲	▲	▲	▲
External Program Calls	2	▲	—	U1	—
Program Debugger	3	▲	—	—	▲
Forms					
Define Form and Table	1	▲	▲	—	—
Query by Form	2	▲	▲	▲	—
Step through Query Results	2	▲	▲	▲	Y1
Browse/edit Forms	2	▲	▲	▲	—
Data Entry Forms	2	▲	▲	▲	▲
Multirecord Display	2	▲	▲	▲	▲
Multiline, Multirecord Display	2	—	▲	▲	▲
Edit Multirecord Display	1	▲	▲	▲	—
Scroll Multirecord Display	1	▲	▲	▲	▲
Select from Multirecord List	1	▲	▲	▲	▲
Linked Master-Detail Forms	2	▲	▲	▲	Z1
Remote Virtual Lookup	1	▲	▲	▲	—

See Footnotes on page 329.

Table 2. SQL Front-End Features (Continued)

	Weight	Borland International Paradox	DataEase International DataEase/SQL	Oracle Tools and Database	Revelation Technologies Advanced Revelation
Local-to-Remote Virtual Lookup	1	▲	▲	DL	—
Field-Level Triggers	1	▲	—	▲	▲
Key-Based Triggers	1	▲	W1	▲	▲
WYSIWYG Form Design	2	▲	▲	▲	▲
Multiple Forms per Table	1	▲	▲	▲	▲
Picture Format	2	▲	▲	▲	▲
Folded Field Values	2	X1	▲	—	▲
Horizontal Field Scroll	1	▲	—	▲	▲
Form-Level Validation	2	▲	▲	▲	▲
Display-Only Fields	2	▲	▲	▲	▲
Conditional Edit	1	▲	A2	▲	P
Virtual Calculated Fields	2	▲	▲	▲	▲
Default to Calculation/lookup	2	▲	▲	▲	▲
Field-Level Color Options	1	▲	▲	▲	▲
External Program Calls	2	▲	—	▲	—
Reports					
Master-Detail Reports	2	▲	▲	▲	C2
Lookup Fields	2	▲	▲	▲	C2
Local-To-Remote Lookup	1	▲	▲	DL	—
Joined Queries	2	▲	▲	▲	C2
Selection Criteria at Runtime	1	▲	P	▲	P
WYSIWYG Report Design	2	—	▲	—	▲
WYSIWYG Report Preview	2	▲	—	▲	▲
Word-Wrap Fields	2	▲	▲	—	▲
Picture Format	2	▲	▲	▲	▲
Group Nesting	2	▲	▲	▲	▲
Group Totals and Aggregates	2	▲	▲	▲	▲
Calculated Fields	2	▲	▲	▲	▲
Mailing Labels	1	▲	▲	▲	▲
Text Enhancements	1	▲	▲	▲	B2
Conditional Printing	1	▲	▲	▲	—
Output to File	2	▲	▲	▲	▲
Pause between Pages	1	—	▲	▲	▲
Application Development					
Menu Generator	3	▲	▲	▲	▲
Multiple Menu Formats	2	▲	—	—	—
Graphical Interface	0	—	—	—	—
Multiple Window Display	1	▲	—	—	—
Supports Windows DDE	0	—	—	—	—
Other Spreadsheet Link	1	D2	—	G2	H2
Menu-Level Help	1	▲	—	—	▲
Form-Level Help	1	▲	▲	▲	▲
Field-Level Help	1	▲	▲	▲	▲
Graphs	2	D2	—	—	▲
User Profiles	1	▲	▲	—	▲
Local User IDs	1	▲	E2	F2	▲

See Footnotes on page 329.

Table 2. SQL Front-End Features (Continued)

- ▲—Yes, has feature.
- A—Single-numeric format with user-defined precision.
- B—Date and time front-end variables can reference date/time database variables.
- C—Paradox date uses database date or date/time.
- D—Can be defined; not supported in forms or reports.
- E—User cannot create format; program can translate user-defined fields on existing tables.
- F—SQL*Forms reads data portion of date/time only; time portion is character field in time format.
- G—Using a DQL procedure.
- H—Only in SQL command mode.
- I—Available from vendor as separate purchase.
- J—Two-step process: import locally and copy to remote table.
- L—Only while creating table.
- N—Enforced locally only; not incorporated into database structure.
- P—Programmable.
- Q—Enforced locally in forms only.
- R—Enforced locally in data entry only, not in updates by Query By Example.
- S—Oracle's procedure language is SQL; no translation necessary.
- T—In SQL*Forms; not in SQL*Plus.
- V—Can enter SQL query directly using MiniScript.
- W—View only.
- X—Available with Oracle Tools and Database, not with Tools alone.
- Z—Announced; not yet available.
- A1—To be implemented with support for multiple platforms.
- A2—Can blank out a field based on condition.
- B1—Supports Sybase SQL Server on UNIX and VAX platforms.
- B2—Setup strings can be included.
- C1—Announced support for Sybase and Tandem.
- C2—Must base report on local query table results.
- D1—Applications using multivalued fields cannot be used with nonrevelation data when other front ends will be accessing data.
- D2—Available in third-party product.
- DL—Achieved with a database link available in Oracle Server.
- E1—Single table only.
- E2—Will prompt for server user ID if local ID not found on server.
- F1—Commands and functions for direct database manipulation cannot be used with remote tables.
- F2—Available with Oracle Tools and Database, not with Tools only.
- G1—Used by the product internally; not directly available to user.
- G2—Lotus 1-2-3 link available.
- H1—Available in embedded or interactive SQL.
- H2—Includes hot link to Quattro Pro.
- I1—Procedures returning more than one result set (e.g., sp-help) display only the last set.
- J1—Must use Transact/SQL; documentation provides instructions.
- K1—Subject to product's limits on the use of embedded SQL.
- L1—Message not explicit.
- M1—Any on-screen change locks out transaction processes.
- N1—User-defined lock time-out.
- P1—Must query full table column into local table, then reference in a PAL script; or enter SQL directly.
- Q1—Substring of formatted date or time.
- R1—Cannot select into program variables.
- S1—Can be simulated using DQL FOR clauses.
- T1—SQL is Oracle's procedure language.
- U1—Host language interface available with Oracle Tools and Database, not with Tools alone.
- W1—Limited options available.
- X1—Long-text fields only.
- Y1—Can display local query result table in form view.
- Z1—Must use local query result tables.

Software Prices

	Purchase Price (\$)
SQL Front Ends	
Borland International Paradox	795
Borland International Paradox SQL Link 1.0	495
DataEase International DataEase/SQL	995
Oracle Tools and Database	1,299
Revelation Technologies Advanced Revelation	995
Revelation Technologies SQL Server Bond 1.02	695

Glossary of Terms

Field Data Types

Assigned when creating tables on the server or recognized as being distinct from similar data types when accessing existing tables on the server.

Fixed-length character: Accepts alphanumeric character strings as data values; each value is stored in a fixed-length area.

Variable-length character: Accepts alphanumeric character strings as data values; each value is stored in the space required for the characters entered.

Integer: Accepts numeric data values that are integers.

Small integer: Accepts relatively small data values that are integers.

Floating point: Accepts numeric values with variable numbers of decimal places.

Fixed point or money: Accepts numeric values with fixed numbers of decimal places; values are aligned vertically on the decimal point.

Bit or logical: Accepts only two values, such as 0 and 1, or yes and no.

Long text: A special data field stores text data that is too long for an ordinary character variable.

Long binary: A special data field stores binary data that is too long for an ordinary character variable.

Date: Stores calendar date values, which can be sorted by date in date format (i.e., 01/01/91 is greater than 12/31/90).

Time: Stores data values containing times of day.

Date/time: A single field can contain a date and time, and values can be sorted by time within date.

User defined: Given a server that supports user-defined data types (e.g., Microsoft SQL Server), the SQL front end can recognize and utilize such data types.

Tables and Views

Create remote table: The program provides on-line facilities for creating a new table on the server.

Alter remote table: The program provides on-line facilities for altering the structure of an existing table on the server.

Copy table structure: The program provides on-line facilities for creating a new table on the server by copying the structure of another table.

Access existing table: The product can access tables on the server that it did not create.

Access table with no unique key: The product can access tables on the server that do not have unique keys.

Import data into remote table: The program provides on-line facilities for importing data into a remote table.

Create index: The program provides on-line facilities for creating an index on an existing server table.

Use existing index: The product can use indexes it did not create to enhance data retrieval speed.

Multifield indexes: The product can create or use an index that references more than one field. For unique indexes, uniqueness can be enforced against a combination of fields rather than a single field.

Create view on server: The program provides on-line facilities for creating a view on the server.

Access existing view: The product can be used to access an existing view on the server.

Create/access grouped views: The product can use grouped views on the server. Rather than displaying table data directly, a grouped view presents statistics across multiple records, such as the count of records for each department code, or the sum of sales amounts for each department code. Grouped views are normally created using an SQL GROUP BY query.

Update view on server: The product can update a view on the server, subject to only those restrictions imposed by the server.

Automatic resynchronization: With a single command or Function keystroke, the user can resynchronize an application to reflect structural changes to a server table.

Data Validation

Refers to validation of data entered into database server tables.

Null/not null: Table definitions can specify whether null values will be accepted in a particular field.

Range validation: Field values can be restricted to a specified range.

Picture validation: Field values can be made to conform to a specific pattern (e.g., a phone number field matches the pattern # # # - # # # - # # # #, where # can be any number).

Set validation: Field values can be restricted to a specific set of values.

Default to calculation: A field can be made to calculate a value based on a formula if the user does not explicitly enter a value.

Default to table lookup: A field can be made to look up a value in another table if the user does not explicitly enter a value.

Lookup validation (foreign key): Field values can be restricted to those found in a specified field in another table (often used to enforce referential integrity, when a field is defined as a foreign key and its value must match a primary key value in another table).

Prevent update/delete (referential): The product can prevent the user from deleting a record, or updating its primary key, if the record is referenced by a foreign key in another table.

Cascaded update/delete (referential): The product can automatically update a foreign key value when its referenced primary key value changes, or it can automatically delete a referencing record when the referenced record is deleted.

Interactive Query

Query By Example: A Query By Example interface is provided for querying remote tables.

Query results to local table: The results of a query to a remote table can be stored in a local table.

Browse remote table: A user can browse through records in a remote table (rather than first querying and then browsing through a local results table).

Browse query results: A user can browse through multiple records that constitute the result of a query, even if those records are not consecutive in the base table.

Update/delete remote table in browse: While browsing through records in a remote table, the user can update or delete the record where the cursor is positioned.

Set-level update/delete: A user can update or delete multiple records in a table with a single command using selection criteria.

Copy records, local to remote: The product offers interactive facilities for copying records from a local table to a remote table.

Join remote tables: A single interactive query can join two or more remote tables.

Join local and remote tables: A single interactive query can join a local table and a remote table.

Grouped query with aggregates: A query can return aggregate totals for groups of data, such as the count of records, the sum of sales amounts, and the average sales amount by distinct department code. The query interface translates the user's entry into an SQL GROUP BY statement and sends it to the server.

Aggregates in selection criteria: A grouped query can include aggregate values in the selection criteria (i.e., list all departments and departmental sales totals below a specified minimum). The query interface translates the user's entry into an SQL GROUP BY HAVING statement and sends it to the server.

Outer/inner joins: The product offers outer and inner join capability.

Allow/suppress duplicates: A user can specify whether to include or suppress duplicate rows in a query result. The query can be passed to the server as a standard SELECT or as a SELECT DISTINCT.

Null/not null: A query's selection criteria can include the condition that a field is null or not null. A null value means that the field is empty.

Join on nonequality: A query can join two tables based on a nonequal condition, such as where a field value in table 1 is greater than a field value in table 2.

Store/reuse query with variables: A query written to reference variables can be stored and reused with new values (if the values change).

Store/reuse query with user input: A query can be written (and stored) with a pause for the user to input variable values.

View/modify SQL translation: A user can view the SQL translation of a query entered interactively and modify the SQL statement before executing.

SQL command mode: The product includes an interface for interactively entering and executing SQL commands.

Connectivity

Gupta SQLBase Server: The product can connect to Gupta Technologies' SQLBase Server running under DOS or OS/2.

IBM OS/2 Extended Edition: The product can connect to the IBM OS/2 Extended Edition Database Manager Version 1.2 or higher.

Oracle SQL*Net: Using Oracle's SQL*Net, the product can connect to an Oracle Server database running in any environment that Oracle supports.

Microsoft SQL Server: The product can connect to Microsoft SQL Server.

IBM DB2: The product can connect to IBM's DB2 on a mainframe.

RDB: The product can connect to RDB on a minicomputer.

Other mini/mainframe databases: The product supports one or more minicomputer databases not listed above.

Switch between remote databases: The user can disconnect from one remote database and connect to another (or switch without disconnecting) without first exiting to the operating system. The databases can run on the same platform (e.g., two different Microsoft SQL Servers).

Switch between remote platforms: The user can disconnect from one remote database and connect to another running under a different platform (e.g., from Microsoft SQL Server to Oracle on a VAX) without first exiting to the operating system.

Access multiple remote databases: The user can simultaneously connect to databases on more than one server to access data.

Access multiple platforms: The user can simultaneously connect to databases under different platforms to access data.

Two-phased commit: When a transaction involves two databases, the program first verifies that both databases are ready to commit (i.e., no errors warranting rollback), then commits the data. Two-phased commit ensures data integrity across multiple databases.

Simultaneous local/remote access: A user can access local and remote tables in the same session.

Local sessions: The product can be used in a session with local data only, without connecting to a database server.

Application Portability

"Generic" refers to the capability to use the same application, unaltered, locally and on all database server platforms supported by the front end.

Generic forms: Forms created to work with one database platform work with any supported platform.

Generic reports: Reports created to work with one database platform work with any supported platform.

Generic interactive query: The standard on-line interface or interactive command syntax can be used to query any tables the product can access.

Generic procedural language: The standard command scripts, including commands for reading and writing data, can be used with any supported platform.

Generic error recovery: The product returns generic messages for common error conditions (e.g., query returns no rows, table does not exist, connection lost, etc.) rather than server-specific error code numbers. A programmer can write generic routines to detect these conditions rather than having to test for server-specific error code numbers.

Macintosh version: The product is available in a version that runs on Apple Macintosh computers. Applications developed for the DOS version run on the Macintosh version and vice versa.

OS/2 version: The product is available in a version that runs under OS/2. Applications developed for the DOS version run on the OS/2 version and vice versa.

Server-Specific Features

Oracle array variables: The product's command language supports array variables and can pass these variables to and from Oracle Server in SQL statements. For example, passing a group of five-occurrence array variables, which Oracle uses to insert five rows with a single SQL statement.

SQL Server stored procedures: The product supports the execution of Microsoft SQL Server stored procedures.

SQLBase chained commands: The product supports the retrieval and execution of SQLBase Server chained commands.

OS/2 ARI: The product supports execution of remote procedures using IBM OS/2 Extended Edition's Application Remote Interface.

Oracle OTEX: The product supports execution of several SQL commands in batches using Oracle Server's OTEX feature.

OS/2 bind prior to runtime: Optimization plans for database commands in applications used with IBM OS/2 Extended Edition are stored in the database prior to execution. These plans can be executed at runtime without being reprocessed.

OS/2 record blocking: When queries against an IBM OS/2 Extended Edition database return multiple rows, all the rows are returned to the workstation in a single operation. Individual rows can then be fetched locally rather than across the network.

SQL Server browse mode: When records are displayed for updating, the product makes use of Microsoft SQL Server's browse mode, which includes a time stamp field. Before updating a record, the program reads the time stamp to check for updates by other users since the record was selected. Altered records are reselected to reflect the current values.

SQLBase result sets: The product makes use of SQLBase Server's result sets for storing and accessing the results of queries that return multiple rows.

OS/2 foreign keys: When used with IBM OS/2 Extended Edition, the product's table creation interface can specify foreign keys for enforcing referential integrity. Foreign keys become part of the table definition stored in the database (i.e., not specific to the front end).

SQL Server rules: When used with Microsoft SQL Server, the product's table creation interface can specify data validation rules that become part of the table definition stored in the database (i.e., not specific to the front end).

SQL Server system procedures: The product provides a facility for executing Microsoft SQL Server system procedures (which provide database status or structure information) and displaying the results on-line without row-by-row fetches.

SQLBase context preservation: The product makes use of SQLBase Server's context preservation feature, which lets open cursors continue to access data after transactions have been committed.

SQL Server Select Into/temp tables: Applications using Microsoft SQL Server can store data in temporary tables and access data in those tables. Temporary tables are created using SQL Server's SELECT INTO syntax and deleted automatically at the end of a session.

Oracle sequence generator: The product uses Oracle Server's sequence generator to automatically sequence values in specified fields. Sequencing at the database level can be maintained even when multiple front ends insert records into the same table.

Multiple threads: The product can use multiple threads on servers that support the feature (e.g., multiple database processes with Microsoft SQL Server).

Concurrency: Concurrency is handled by the database server; these front-end features utilize server concurrency control features.

Transactions: The product supports the grouping of commands into transactions. Changes are permanently written (committed) only when the entire transaction is complete; if an error occurs, all operations included in the transaction can be rolled back.

Auto commit on/off: The product can be set up to automatically commit changes after each command or to wait for an explicit commit.

Interactive transaction processing: A user can group multiple interactive commands into transactions by selecting on-line options to begin, commit, or roll back.

Concurrent data entry: Multiple users can enter records into a table at the same time.

Concurrent browse/edit: Multiple users can browse the same records; one user can modify a record that another user is viewing. Data integrity mechanisms (e.g., update notification) do not prevent concurrent access.

Concurrent browse/transaction processing: A user can browse and edit records in a table while another user processes transactions that manipulate those same records. One user may have to wait while another commits changes to specific rows, but neither is prevented from proceeding. If the browsing user has not made uncommitted changes, transaction processing should proceed normally.

Concurrent query/transaction processing: A user can query records in a table while another user processes transactions that manipulate those same records. Data integrity mechanisms do not prevent either user from proceeding.

Concurrent report/transaction processing: A user can run a report while another user processes transactions

that manipulate data on which the report is based. Data integrity mechanisms do not prevent either user from proceeding.

Snapshot read: In certain read operations (usually reports), the program creates a "snapshot" of the data as it exists at that point and uses that data for the duration of the transaction. Subsequent modifications are not reflected in the first user's report.

Update notification: The user is notified if a record he or she is attempting to modify has been changed since it was last read.

Screen refresh: The application refreshes a displayed record to reflect changes made by another user since the record was originally read.

Read with no lock: The program can read records without locking them; other users are free to process changes to the table. If the server does not support no-lock Selects, the program automatically issues a Commit command after records are read to release the lock.

Hold lock: The user can optionally hold a lock on selected records, guaranteeing the ability to make and commit changes without intervention.

Deadlock management: Given a server that supports deadlock detection, the product can capture the deadlock message from the server and inform the user or return the appropriate information to an application program, so the user or programmer can decide how to respond.

Break lock: The product can break the operation causing a record lock rather than wait indefinitely for lock release.

Query Language Functions

Refers to functions that can be used in actual queries, either as data to be retrieved/updated or in selection criteria. It does not apply to data that can be computed by means of programming. The function must be supported in its proprietary query language or interface for use with remote databases; programs receive no credit for embedded SQL unless SQL is the product's principal query language.

Aggregate functions: A query can include any of the standard aggregate functions: Count, Sum, Average, Minimum, Maximum.

Date comparisons: A query can compare two database date values or compare a date with a constant.

Date arithmetic: A query can include date arithmetic, such as computing the number of days between two dates or adding a number of days to a date.

Extract date part: A query can extract a date part (i.e., month, day, or year) from a date value.

Time functions: The program can perform the same functions against time values as against date values (i.e., comparisons, arithmetic, extract time part).

Substring: A query can extract a portion of a string beginning at a specified position for a specified length. For example, substring("ABCDE",3,2) = "CD."

Substring position: A query can return the position in a string where a substring begins. For example, find("ABCDE", "CD") = 3.

String length: A query can return the length of a string. For example, length("ABCDE") = 5.

Concatenation: A query can combine two strings into one.

Arithmetic operators: A query can employ standard arithmetic operators: addition, subtraction, multiplication, division, modulo.

If/null/Then: A query can substitute a specified value for null values in a queried field.

If/Boolean/Then/Else: A query can return one of two arguments based on the Boolean evaluation of a test condition.

Function nesting: A query can nest functions. For example, if a field called Name contains the value "Smith,John," then substring(name, find(name, ",") + 1, length(name) - find(name, ",")) = "John."

Procedures

Embedded SQL: SQL commands can be embedded in procedures and sent to the server for execution when the procedure is run.

Dynamic SQL: Procedures can execute SQL statements built at run time based on information supplied by the user. Not only can an SQL statement reference program variables, the statement syntax itself can be variable.

SQL cursors: Procedures may contain SQL cursors, in which the results of a query can be fetched one row at a time and processed.

Update/delete WHERE CURRENT: The SQL WHERE CURRENT option can be used to update or delete the row last fetched from an open cursor.

Nested selects: Procedures may include nested select statements. Rows from one select statement can be

fetches one row at a time, and the processing for each row may include another select statement.

Preserve context across commits: Rows from a cursor can continue to be fetched even if changes have been committed since the cursor was opened.

Array variables: Procedures may reference variables that contain multiple values, each accessed using a subscript.

Multirow array operations: A command can use array variables to reference multiple rows in a table. For example, values from five rows could be selected into five occurrences for each of several array variables, or five rows could be inserted into a table using five occurrences for each of several array variables.

Procedural logic: A procedure can include procedural logic such as IF/THEN/ELSE and WHILE.

Screen painting in procedures: Procedures can include commands to display information at specified coordinates.

Call forms from procedure: Forms can be called from within a procedure.

Print from procedure: Procedures can include commands to print formatted output without using a separate report program produced by the report generator.

External program calls: Procedures can call, and pass data to and from, subroutines written in a host language such as C or Cobol.

Program debugger: The product includes a debugger for stepping through procedures and displaying variable contents for the purpose of identifying problems.

Forms

Refers to a form's capability to access database server tables, except where indicated. "Local" refers to data accessed directly without using a database server; "local" data may reside on a network.

Define form and table together: A form and a database table can be defined together as a single operation.

Query by form: The user can enter criteria into a form to view records that match those criteria.

Step through query results: When a form is used to view query results, the user can step through the records in the query result, even if those records are not consecutive in the database.

Browse/edit forms: A form can be used to browse records in a table and make changes to the displayed record.

Data entry forms: A form can be used to enter new records into a table.

Multirecord display: Several records from a table can be viewed simultaneously.

Multiline, multirecord display: Records in a multirecord display may span more than one line on the form.

Edit multirecord display: Records in a table can be changed on-line while displayed in multirecord format.

Scroll multirecord display: The user can scroll past the last record displayed to access additional records.

Select from multirecord list: A multirecord display can be treated as a menu; the user may position on and select the record whose values are to be used in subsequent processing.

Linked master-detail forms: A form may display a single record from one table and multiple records from a second table that match a specified key value (e.g., information about a customer and list orders placed by that customer).

Remote virtual lookup: A form displaying records from one table can also display a value looked up in another database server table. The lookup value does not correspond to a field in the base table.

Local-to-remote virtual lookup: A form displaying records from a local table can also display a value looked up in a database server table. The lookup value does not correspond to a field in the base table.

Field-level triggers: The developer may specify procedures triggered by specific actions related to a specific field (e.g., field value changes).

Key-based triggers: The developer may specify procedures triggered by a specific keystroke within the form.

WYSIWYG form design: A layout screen for form design permits direct placement of fields and constant text (versus specifying screen coordinates in a procedure).

Multiple forms per table: Multiple forms may be designed for a single table.

Picture format: The developer can design a form with "picture" formats representing field value types (e.g., 999,999.99 for a numeric field or MM/DD/YY for a date field).

Folded field values: Long field values can be displayed on multiple lines within a single field.

Horizontal field scroll: When a field value extends beyond its display area, the user can scroll horizontally to view the off-screen portion of the field.

Form-level validation: Field validation criteria can be specified as part of the form definition; the criteria are applied only when data is entered using the form.

Display-only fields: A form can include fields defined as "display only" that cannot be updated by the user.

Conditional edit: The developer can specify conditions under which a field may or may not be modified.

Virtual calculated fields: A form can display values calculated from other field values. The calculated value is for display only; it does not correspond to a field in the base table.

Default to calculation/lookup: The developer can specify a default field value that may be calculated or looked up in another table. The default value can be overridden by user entry.

Field-level color options: The developer can specify display colors for field components (e.g., foreground, background, labels).

External program calls: Forms can call and pass data to and from subroutines written in a host language such as C or Cobol.

Reports

Capabilities for accessing data from database server tables, except where indicated. "Local" refers to data accessed directly without using a database server; "local" data may reside on a network.

Master-detail reports: A report can include records from one table, each with multiple records from a second table matching a specified key value (e.g., print information for selected customers, each with a list of orders placed).

Lookup fields: A report listing records from one table can also show a value looked up in another server table.

Local-to-remote lookup: A form report listing records from a local table can also show a value looked up in a server table.

Joined queries: A report can be based on a query that joins records from two or more tables, treating the rows returned by the query as records in a single table.

Selection criteria at run time: Selection criteria for a report can be specified when a report is run.

WYSIWYG report design: A layout screen for report design permits direct placement of fields and constant text such as detail text for each record selected, group headers and footers, and page headers and footers.

WYSIWYG report preview: A report can be displayed as it will appear when printed.

Word-wrap fields: Columnar reports wrap long field values to multiple lines without exceeding column boundaries.

Picture format: A report definition can display "picture" formats representing field value types (e.g., 999,999.99 for a numeric field or MM/DD/YY for a date field).

Group nesting: Data may be printed in nested groups (e.g., all customers grouped by country, within country by state, within state by city).

Group totals and aggregates: A report can compute and print totals and other aggregate values for each group (e.g., total sales, average sales, etc.).

Calculated fields: A report can include values calculated from database fields.

Mailing labels: Output can be printed in mailing label format (i.e., several multiline records across the page).

Text enhancements: Reports can include text enhancements such as underlining and boldface.

Conditional printing: A report definition can include conditional instructions for printing a particular field; for example, IF balance 0 THEN print balance ELSE print "(" , balance, ")".

Output to file: Report output can be routed to a file rather than to a printer.

Pause between pages: A report can be set up to pause after each printed page (e.g., so that a new sheet of paper can be inserted).

Application Development

Menu generator: A program facility generates menus for selecting options in an on-line application.

Multiple menu formats: The menu generator offers more than one menu format (e.g., pop-up, pull-down, horizontal, vertical).

Graphical interface: A user-developed application can feature a graphical interface.

Multiple windows display: A user-developed application can display data in multiple windows that can be individually positioned and sized.

Supports Windows DDE: The product supports the use of Microsoft Windows Dynamic Data Exchange to pass data between database tables and other applications such as Microsoft Excel spreadsheets.

Other spreadsheet link: The product offers facilities for passing data between database tables and spreadsheets by some means other than DDE.

Menu-level Help: Developers can specify Help text for a specific menu

Form-level Help: Developers can specify Help text for a specific form.

Field-level Help: Developers can specify Help text for a specific field within a form.

Graphs: The product includes facilities for displaying database data graphically.

User profiles: Profiles can be set up for individual users specifying start-up menus, security levels, and other user specifics.

Local user IDs: User profiles can designate user IDs other than those used to connect to database servers. ■

LAN E-Mail Programs

A Report from NSTL

In this report:

Program Evaluations -408

Rating Summaries..... -411

Performance Results..... -414

Vendors -417

Characteristics..... -418

Synopsis

Focus
This report evaluates five electronic mail programs for use on LANs. NSTL explains and compares the mail systems' architectures, naming conventions, message creation procedures, and internetworking capabilities.

Editor's Note
Microsoft Corp. has acquired the assets of Consumers Software, Inc., publisher of Network Courier, which is evaluated in this report. Microsoft plans to release Microsoft Mail for PC Networks based on the current Network Courier product, preserving its architecture and feature set.

Programs Tested

PMX/StarMail
AT&T

cc:Mail
cc:Mail, Inc.
Network Courier
Consumers Software, Inc.
InBox Plus
Sitka Corp.
CompleE-Mail/MHS
Transend Corp.

Program Recommendations

- Consumers Software Network Courier
- cc:Mail

Source
Based on data generated by tests designed and conducted by National Software Testing Laboratories, Inc. (NSTL), a division of Datapro Research Group, Plymouth Meeting, PA 19462. Telephone (800) 223-7093.

Ratings Key
(On a scale of 0 to 10)

Ratings

- 7.0 - 10.0
- 5.0 - 6.9
- ⊙ under 5.0

★ Recommended

Overall Evaluation	Program Name	Version Tested	Performance	Versatility	Ease of Learning	Ease of Use	Price
8.6	Consumers Software Network Courier	2.1	●	●	●	●	\$1,090 (5 Users) ★
7.8	cc: Mail, Inc. cc: Mail	Pre-3.2	○	●	●	●	\$1,090 (25 Users) ★
5.8	Sitka InBox Plus	3.0	○	⊙	●	●	\$1,995 (100 Users)
5.1	AT&T PMX/Star Mail	2.2	⊙	⊙	○	○	\$1,295 (Unlimited Users)
4.4	Transend CompleE-Mail/MHS	1.04	⊙	⊙	○	○	\$295 (5 Users)

Overview

Messaging systems come in a range of styles and sizes—from global networks and common carrier utilities, to public access bulletin boards, to PBX voice messaging, and finally LAN mail systems. LAN E-Mail text messaging systems generally have the most user exposure, and many businesses consider E-Mail among the top reasons for installing a LAN.

Most mail systems provide an interface where users create and exchange text messages with other users on the LAN and provide facilities for attaching and transmitting files (e.g., graphics, executables). E-Mail vendors seeking to expand the scope of LAN mail packages are adding connectivity options, for accessing systems and users beyond the local network, and mail system interfaces with applications.

Evaluation Criteria

NSTL evaluated five LAN E-Mail software packages that support DOS workstations and run in text (character) mode; some offer Windows interface options. All the programs are compatible with Novell NetWare 386 Version 3.1 on a file server via 16M bps token-ring connections. Many E-Mail packages are file based and will run on any network that supports DOS file locking. Each mail system supports internetworking to LAN installations running the same package, and all support gateways to other messaging systems such as MCI Mail or AT&T Mail, UNIX Mail, VAX Mail, or PROFS.

Proprietary E-Mail systems offered by network operating system vendors are excluded. Banyan and 3Com integrate E-Mail into their operating system's naming scheme and take advantage of network wide area links. Version 2.0 of Davinci's E-Mail package was not released in time for testing.

NSTL rates E-Mail systems on the basis of usability, installation, features, and performance. The performance evaluation focuses on message transmission within a single LAN, monitoring for elapsed time and network usage. Consideration is given to the availability of more sophisticated internetworking capabilities using X.25 and other connections and gateways to large-scale messaging systems.

General Mail Architecture

LAN E-Mail packages typically build on a database application model with a specialized retrieval engine. The database or Post Office resides on a network server. The database stores messages awaiting delivery and overall configuration information such as remote Post Offices (on other LANS) and the local E-Mail administrator password. Post Office databases should be located on server volumes with plenty of available disk space for the accumulation of mail and other files. Programs within the E-Mail package act as a database engine and manipulate the mail database. These programs store and forward messages and provide a database interface for creating and retrieving messages.

A password-protected administrative program manages the mail system and controls the overall Post Office configuration. The mail administrator's tasks include adding new mail users and changing forgotten passwords. With all the mail systems except PMX/StarMail, only the administrator can run the administrative program. All PMX/StarMail users execute a subset of the administrative program to configure individual accounts, but only the administrator can access the full range of administrative functions and options.

A user application provides a mail system interface and acts as a database engine for the mail database. The user application sends and retrieves messages and provides a message editor for creating new messages. Mail system vendors generally offer multiple versions of the user application: a full-scale version, a memory-resident version with limited capability, and a notification memory-resident version that alerts the user to newly arrived mail.

Mail Directory Naming

User Mailboxes

Most of the E-Mail packages locate user mailbox files on the server. Mailboxes contain user information and messages or pointers to messages in the Post Office.

PMX/StarMail mailbox directories can be located on the user's local hard disk. Local installation makes it difficult for users to access the mail system from other workstations on the network. In describing the initial setup of individual user accounts, the PMX/StarMail documentation recommends using the local hard disk or an always accessible server directory. The documentation fails to mention that if the directories are not local, each user must use a separate server directory because casual use of the same path leaves the latest user owning all the mail.

Local User Names

When adding a mail user, the administrator assigns that user an E-Mail name, initial password, and mail address. The mail address defaults to the local Post Office, but remote users can be defined. cc:Mail and InBox Plus associate only the E-Mail name with the user; Network Courier allows a separate full name field. The mail administrator should use a consistent user naming convention (e.g., first name/last initial). Although users can select addressees from a name list, consistent naming facilitates addressing without searching the name list.

Remote Names/Addresses

Addressing users outside the local Post Office goes beyond simple naming. cc:Mail uses a two-level name/address scheme that concatenates the target user's mail name with an address (typically the name of the target's home Post Office). Network Courier uses a three-part designation including the user name, Post Office name, and network name. Three-part naming is more flexible because a single local network may have several Post Offices, and separate networks exchanging mail may have Post Offices with the same name. A special feature in cc:Mail is directory propagation, which automates distribution of user updates (i.e., new users are automatically added to all attached Post Offices).

Message Addressing

All the mail systems can send a message to multiple recipients. Primary recipients are entries in the message's To: field. Entries in the CC: field receive courtesy or carbon copies of the message. cc:Mail and PMX/StarMail offer a blind copy or BCC: field. A Subject: field holds a brief description of the message contents that are displayed in the recipient's mailbox. The mail systems implement a number of different methods for entering names in the To:, CC:, and BCC: fields.

The cc:Mail address menu forces selection of one or more names from a list of available mail users. The sender can type in the name(s) rather than select from the list. As the user types a name, the program speed searches for the first potential match in the user list. If the sender types a name not found in the list as an initial substring, the program responds with an error message saying that new names may not be entered. Names entered in the To: field are displayed across the screen, several names to a line. A two-key sequence exits addressing, interrupting the overall flow of the mail creation process.

Network Courier's Compose screen displays a To: field for message addressing. When the sender selects the To: field, the program displays a list of potential recipients. As in cc:Mail, keyed entries trigger a speed search of the list, but an invalid string causes the cursor to highlight the closest alternative. When selection is complete, the To: field expands to display a column of names down the top of the message. The recipient list can be a bit more difficult to read than the cc:Mail display because the list may be longer than the screen height.

PMX/StarMail's message creation screen starts the user in the Subject field rather than the To: field. A To: field is added by selecting F1 and typing in a name. The program does not automatically display a list of user names, but a list can be called and selections made from the list. To: field entries need not be local Post Office addresses (the user list includes only local users). Multiple destinations are displayed as a column of To: lines.

InBox Plus treats messages as memos. Message creation begins in the Subject: field (which can be skipped) then moves to the message text. Addressing is triggered by a Function key that activates an address list window. When addressing a message, the user is in To: or CC: or BCC: mode; switching modes requires tabbing (or clicking) to the mode selection area, which interrupts the flow of addressing. Multiple addresses are displayed in a column that does not overrun the screen. The address column is viewed through a scrollable, resizable window within the message.

From the top level of the CompletE-Mail desktop work space, a Function key initiates message creation beginning in a Subject: field followed by To:, CC:, BCC:, and Attachment fields before entering the message text. The user can manually enter recipients' names or invoke CompletE-Mail's address book for selection. Multiple selections display in a column within the message header, which may become too tall to fit within a screen.

Overall, addressing in cc:Mail and Network Courier is faster and easier than in the other products. Both feature speedy searching and natural flow. Drawbacks are cc:Mail's exit keystrokes and Network Courier's columnar display.

Foreign Addresses

Foreign addresses are destinations on systems outside the LAN E-Mail environment, such as MCI mailboxes or UNIX mail users. Because cc:Mail and Network Courier require address selections from a list, foreign addresses must be entered into the systems' address lists. PMX/StarMail allows direct entry of foreign addresses. InBox Plus does not support mail exchange with foreign destinations. CompletE-Mail relies on the MHS system for internetworking, so addresses must fit the MHS standard for username, possible application name, and workgroup or host name.

Distribution Lists

Groups of users, such as an entire department, may frequently receive the same messages. To obviate the tedium of repetitious entry or list selection, users can create distribution lists as aggregate destinations for messages. A name designating a distri-

bution list is expanded by the mail system into its set of users. In all the mail systems except CompletE-Mail, the administrator can create public distribution lists for global use. All the systems except InBox Plus permit users to create private distribution lists. InBox Plus supports nested distribution lists (lists with entries that designate lists).

Message Editor

The mail package's message editors generally create simple messages and offer limited features. cc:Mail and Network Courier offer the most editing versatility. All the programs can import DOS files into message text or copy part of an existing mailbox message into the current message.

cc:Mail supports very large messages to 20K bytes with word wrap, margins, reformatting, cut and paste, search and replace, ASCII file import and export, and highlighting. A snapshot utility captures screens in other applications and pastes them into messages. Reformatting is key triggered, an advantage only if formatting exceptions are desired.

Network Courier's editor also supports large messages with word wrap, cut and paste by line, keyed reformatting, and highlighting. The editor can import all or parts of ASCII files or existing messages into the current message and export the entire message as a file.

PMX/StarMail's editor supports messages to 200 lines (up to 78 bytes per line) with cut and paste by line and ASCII file import. Word wrapping and line flows are automatic within paragraphs. The editor can create electronic forms to accelerate repetitive messages such as phone orders or work sheets.

CompletE-Mail's editor permits messages to 64K bytes with word wrap, cut and paste, and text file import; and it can create electronic forms.

InBox Plus' simple editor imposes a 32-page limit per message with word wrap and cut and paste. While within the message text, the Tab key cannot be used to position text; Tab moves the cursor to the addressing option.

Message Folders

Mail systems often permit file management into folders similar in concept to those used in many graphical operating environments. cc:Mail, Network Courier, and PMX/StarMail support message management using folders for storage. Users can create folders and move or copy messages between folders. Network Courier permits nested folders for more hierarchical management. InBox Plus supports *storage boxes* for managing messages; the program comes with two default storage boxes, and the user can create and name additional storage boxes. CompletE-Mail's desktop model provides 8 default message *baskets*, and the desktop has room for 16 baskets named by the user.

Message Attachments and Graphics

Mail users can attach a DOS file (text, graphics, or binary) to a message for transmission without incorporating the file into the body of the message. All the mail systems support attachments or enclosures of arbitrary length and provide menus or utilities for searching directories and selecting files for attachment. Network Courier, InBox Plus, and CompletE-Mail allow unlimited numbers of attachments for any message; cc:Mail limits attachments to 20 and PMX/StarMail to 49 per message.

Network Courier and cc:Mail supply screen capture utilities for attaching text and graphics screen images to messages. A cc:Mail graphics editor incorporates free-form drawing, geometric images, and graphic lettering into messages.

Internetworking and Remote User Access

Internetworking and remote access capabilities expand the utility of an E-Mail system beyond the local network to enable mail exchange between local users and users on remote networks and dial-in connections. All the mail packages can exchange mail with more than one mail database or Post Office. A network workstation running the exchange software handles data communications between mail databases or Post Offices via a modem or other communications link. The exchange software generally runs as a dedicated application, rendering the workstation unavailable for other uses.

InBox Plus imposes the greatest limitations on nonlocal mail exchange, as it does not support internetworking. CompletE-Mail relies on MHS support for local and nonlocal mail distribution. cc:Mail sells cc:Mail Gateway for internetwork mail routing, a separate Remote program to originate remote calls, and a Dialin program to accept calls from remote users. PMX/StarMail includes a DOS-to-DOS Communication Protocol (DDCP) program for remote connectivity. Network Courier uses an External Mail exchange program and Modem Mail for remote dial-in access.

InBox Plus calls its inter-Post Office communications program Router (not to be confused with network-level packet routers). An InBox Plus Router requires a dedicated workstation when actually exchanging messages, but Sitka admits the possibility of running Router functions on a non-dedicated system. An InBox Plus Router will not perform true internetworking (e.g., over modems) unless the connection is established some other way by the LAN software itself or through a remote LAN bridge. Router is intended to move messages among InBox Plus message centers within a single LAN. Several Router stations can be set up with overlapping "routings," creating redundant Post Office-to-Post Office links. The redundancy is intended to improve both performance and fault tolerance; the expense is several workstations devoted to routing.

The cc:Mail Gateway handles message exchange between the local Post Office and other Post Offices on the same LAN, LANs accessible via bridges, and remote LANs connected by modems. Gateway requires a dedicated workstation, and a single Gateway workstation supports communications with combinations of local and remote Post Offices and remote users. The Dialin remote access program accepts calls and can prevent call origination at the LAN while enabling remote user access to a cc:Mail Post Office.

Network Courier's External Mail program exchanges messages between Post Offices on the same or separate LANs. A single copy of External Mail services Post Offices on one LAN or bridged LANs; otherwise, each Post Office requires an External Mail machine and modem. The Modem Mail program lets remote users dial into an External Mail program running with a modem.

Transend's CompletE-Mail requires and works with the Message Handling System (MHS) developed by Novell and Action Technologies. MHS provides the local Post Office directory structure. MHS requires a dedicated system to perform inter-Post Office routing. CompletE-Mail works exclusively in local mode even when no MHS router is running.

PMX/StarMail's DOS-to-DOS Communication Protocol (DDCP) program runs on a workstation and provides internetworking via a modem or an AT&T Datakit data switch.

Third-Party Connectivity

An important E-Mail feature is the capability to interface with public data networks and office automation systems running on IBM mainframes and Digital Equipment minicomputers. All the E-Mail packages provide some means of connecting to UNIX systems and IBM PROFS systems; PMX/StarMail requires routing via a UNIX server or AT&T Mail connection to PROFS. Network Courier and cc:Mail connect to IBM DISOSS and Digital Equipment ALL-IN-1 systems; CompletE-Mail connects to ALL-IN-1. CompletE-Mail and cc:Mail connect to VMSmail and Hewlett-Packard Desk.

The three largest domestic long-distance telephone carriers—AT&T, MCI, and Sprint—provide public mail systems. cc:Mail, CompletE-Mail, and Network Courier interface with MCI Mail and Sprint TeleMail and support X.400 connections. InBox Plus and PMX/StarMail interface with AT&T Mail. Access to public mail networks can be made with either a modem and conventional telephone line, or (for higher throughput) an X.400 connection can be established via X.25 leased lines (at noticeably greater expense). Businesses needing only occasional public network access typically use modems and regular phone lines to provide on-demand dial-out to a network mail system.

Administration and Reporting

Mail systems require administration for installation and setup, user configuration and management, and creation of any internetworking or gateway linkages with other systems. Small net-

work installations often designate the network administrator as the mail system administrator, but more sizable networks will be better served with a dedicated mail administrator.

CompletE-Mail largely relies on the administrative features included in the MHS program. The administrator adds users to the mail system with an initial password or no password (the user should immediately set his or her own password upon first use of the mail system).

InBox Plus administration is very simple except for intranetwork router configuration. Administrative functions are essentially limited to adding mail users.

Network Courier comes with a full-featured administration system that is fairly easy to use. Management options include the creation of send-only and receive-only mailboxes and the capability to delete classes of obsolete messages that accumulate.

cc:Mail provides good administrative programs with easy interfaces. The mail system's database model Post Offices rely on two main files for storing messages and configurations. Chkstat and Reclaim command line utilities report on and delete obsolete messages and recover allocated space.

PMX/StarMail administration is simple, but not designed for easy use. Unprompted menus access a limited range of administrative functions, and poorly organized functions strain the administrator's memory as lists (e.g., user or distribution lists) are displayed separately from their maintenance.

CompletE-Mail's reliance on MHS makes adding users cumbersome. MHS is cryptic and characterized by inefficient remote communications in the current revision. MHS does permit configuration of routing and limited management of message queues. User applications that use MHS would do well to supply an administrative interface, obviating use of MHS for local configuration tasks.

Security

E-Mail security has become an increasingly sensitive issue. Many LAN users willingly place confidential or nonpublic statements into mail messages. Recent court cases emphasize that interception of E-Mail messages by anyone other than

the addressees is generally prohibited, even when the interception is ostensibly a confidential review by a supervisor or administrator. The first line in E-Mail security is promulgation of a clear corporate policy concerning access rights and procedures.

None of the database-based mail systems can provide true programmatic security because users have some access to mail directories or files, providing the potential for damaging the mail system. The ultimate solution may be to adopt a true client/server model for mail engines (as is becoming well accepted in the database world). With an autonomous process at the server handling mail distribution, mail user access could be restricted to personal mail files and data. Given the restrictions of current E-Mail implementations, several areas of security can be addressed. General network security precautions apply to all mail systems.

User Access Rights

The network administrator defines directory rights to the Post Office database. cc:Mail and Network Courier document access rights assignments, but not in detail with respect to specific network operating system environments. PMX/StarMail installations set up common and user-specific mail directories. User directories not located on the user's local hard disk should be set up in areas where each user has exclusive access; however, the StarMail documentation is very unclear on this aspect of installation. InBox Plus installation, while simple, should give more attention to directory rights assignments. CompletE-Mail installs after MHS, and the appendix to the NetWare MHS installation and operation documentation lists the required NetWare file access assignments.

All the mail systems require a user login or mail ID, and some require a mail password. InBox Plus users can access another user's mailbox (given that user's password) without exiting the mail application. cc:Mail and Network Courier encrypt messages and attachments in the Post Office so that users with access to all network directories (i.e., network administrators) cannot read all the messages in the database. cc:Mail messages can be specifically encrypted so even users with password access to the mailbox cannot access the message without the encryption key.

Memory-Resident Versions

All the mail packages except InBox Plus come with Terminate and Stay Resident (TSR) versions of the user software. PMX/StarMail's Accessory (filename Smail.exe; about 77K bytes loaded into memory) provides message creation, transmission, and reception using a subset of the regular Access program's functions. cc:Mail's Messengr (about 70K bytes required) offers message reception and transmission. Network Courier offers a Monitor program (about 70K bytes, depending on options selected) and a separate Operator program for telephone operators taking messages. The CompletE-Mail program itself can be run as a TSR (in approximately 170K bytes; by command line selection) with limited functions.

Mail Notification

Except for CompletE-Mail, all the systems offer notification TSRs that alert users to the arrival of mail. The user must then enter the mail program to read messages. TSR versions of the main mail programs generally offer notification but do not coexist with the tiny notify programs; notification of this kind polls the mail database periodically to check for new messages.

Network Courier offers a NETBIOS notification option that circumvents the network delays and traffic possible with polling, but it adds NETBIOS driver overhead for networks without native NETBIOS protocols. cc:Mail, InBox Plus, and Network Courier include DOS command line programs for mailbox checking that can be useful in a login script.

Performance

Performance is less important in E-Mail systems than versatility and usability. Much of the user's time in an E-Mail application is spent composing text, choosing addressees, and reading messages. These activities depend on the user for pacing and on the efficiency of the interface (a usability criterion). Performance becomes an issue only if the mail system slows the user's work or excessively loads the network.

Network Courier provides excellent transmission speed with little network loading. Transmission to local mail users is quite brisk. cc:Mail transmits messages only slightly slower, and users will be satisfied with either of these products' performance. InBox Plus transmits messages somewhat slower than cc:Mail, but its efficient handling of attachments (versus message text) gives it nearly comparable performance. It is fast when sending large attachments to many users, but slower sending large messages. InBox Plus' performance is less a barrier to its use than its lack of internetworking and other features. CompletE-Mail and PMX/StarMail can be very slow; local transmissions can take up to several minutes, with the potential for even longer transmission of remote messages over slower media.

Differences in transmission speeds among the products are often the result of their design, procedures, and distribution methods.

Summary

LAN E-Mail systems offer a broad diversity of features, administrative functions, database designs, and performance characteristics. Choosing a good mail package is important because on many networks it is the most frequently used network application.

Network Courier offers good usability and ease of learning, excellent administrative functions and internetworking capabilities, and fast local performance. Excellent documentation and an easy user interface enhance its usability. Microsoft's recent acquisition of Network Courier may bring substantial changes to the product. Consumers Software had been strengthening the product's internetworking features (e.g., X.25 links) and providing APIs for applications that interact directly with the mail system.

cc:Mail is fairly easy to use and administer; its menus quickly become familiar. Message creation tools are powerful and flexible, the documentation is good, and internetworking and remote communication are easy and well designed. Security features are comparable with the other E-Mail packages, with the advantage of explicit password message encryption. Macintosh systems are supported. cc:Mail is slower than Network Courier for local messaging, but not noticeably slower.

InBox Plus may be useful in smaller networked installations that do not require internetworking and broad features. It is easy to learn and use, and its performance is reasonably good. Multiple message centers can be set up with intranet-work routing, and Macintosh systems can coexist with PCs on a DOS mail system.

PMX/StarMail and CompletE-Mail offer sparse features and slow local network performance. CompletE-Mail's reliance on MHS will likely be viewed as an asset only where other MHS applications are in use or where MHS has been chosen as a vehicle for custom application development. StarMail supports AT&T Mail, with some special features.

Program Evaluations

AT&T PMX/StarMail

Strengths

- Public and private distribution lists
- Connects to AT&T Mail
- Forward or reply to mail; send certified, receipt requested
- Password encryption in Post Office
- Electronic forms creation
- Imports/exports messages to DOS
- Memory-resident mode

Limitations

- No screen capture utility
- Only one Post Office per LAN
- No remote users
- No bulletin board
- No Macintosh or OS/2 protected mode versions

cc:Mail

Strengths

- Memory-resident mode
- Screen capture utility
- Private and public distribution lists
- Internetworking system supports multiple Post Offices

- Connects to VMS Mail, MCI Mail, Sprint TeleMail
- Supports X.400 connection
- Imports/exports messages from Post Office
- Macintosh, Windows, OS/2 versions
- Supports remote users
- Bulletin board feature
- Forward or reply to mail; send certified, receipt requested
- Passwords and messages encrypted in Post Office

Limitations

- Limited two-part naming scheme
- Network security setup poorly documented
- Cannot work with NetWare bindery for user installation
- No electronic forms
- No nested distribution lists
- Poor error handling in administrative programs

Consumers Software Network Courier

Strengths

- Three-part naming scheme
- Very good user and Post Office administration
- Screen capture utility
- Public and private distribution lists
- Internetwork system supports multiple Post Offices
- Internetwork message hop count limit, circular route detection
- Tracks toll calls
- TSR versions for users and operators
- OS/2 protected mode version
- User can switch mailboxes with password
- Bulletin board feature
- Forward or reply to mail; send certified, receipt requested

- Connections to AT&T Mail, MCI Mail, Sprint TeleMail
- Supports remote users
- Passwords and messages encrypted in Post Office
- Supports X.400 connection

Limitations

- Does not work with NetWare bindery for user installation
- No electronic forms

Sitka InBox Plus

Strengths

- Forward or reply to mail; send certified, receipt requested
- Supports Macintosh workstations
- Public and nested distribution lists
- Multiple Post Offices and intranetwork routers on one LAN
- Sorts messages by subject, sender, status, date
- Uses NetWare bindery for user installation
- Supports remote users
- Connects to AT&T Mail
- Bulletin board feature

Limitations

- No internetwork/gateway connections
- No remote users
- No connections to VMS Mail, MCI Mail, Sprint TeleMail
- No X.400 connections
- Minimal Post Office administration functions
- No screen capture utility
- No OS/2 protected mode version
- No password or message encryption in Post Office
- No electronic forms

Transend Complete-Mail/MHS**Strengths**

- Interfaces with other MHS applications
- Private distribution lists
- Electronic forms creation
- Connects to VMS Mail, MCI Mail, Sprint Tele-Mail
- Supports X.400 connection
- Imports/exports messages to DOS
- Supports remote users
- Memory-resident mode
- Forward or reply to mail; send certified, receipt requested

Limitations

- No screen capture utility
- No public distribution lists
- No Macintosh or OS/2 protected mode versions
- No bulletin board
- No password or message encryption in Post Office

Program Recommendations

cc:Mail and Network Courier are both very good LAN E-Mail products. Head-to-head competition between the two leading products has resulted in both offering generous feature sets and similar levels of power.

Network Courier

For the most powerful internetworking features, efficient server utilization, good ease of learning, and ease of use, Network Courier is an excellent E-Mail choice. Performance is good; transmission speeds are especially fast with large attachments sent to many local recipients. Good documentation and on-line Help promote easy installation and user and administrative programs that are easy to learn and use. (Network Courier has been purchased by Microsoft.)

cc:Mail

cc:Mail is versatile, easy to learn and use, and easy to administer. Internetworking setup is easy, and message creation features are powerful. Performance is good and server loading moderate. A directory propagation feature automates distribution of user updates (i.e., new users are automatically added to all Post Offices). The program comes with on-line Help and good documentation.

Rating Summaries

Figure 1.
AT&T PMX/StarMail Ratings

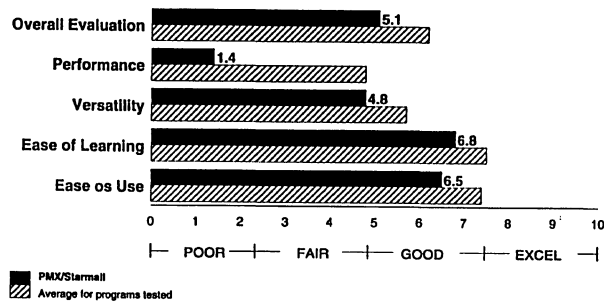


Figure 2.
cc:Mail Ratings

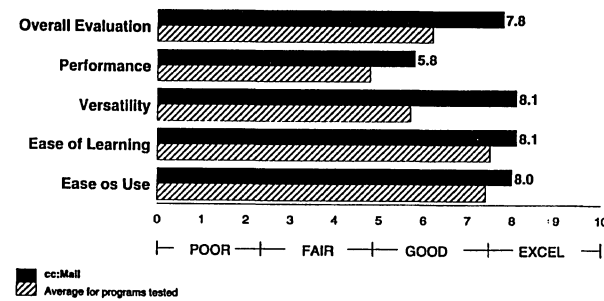


Figure 3.
Consumers Software Network Courier Ratings

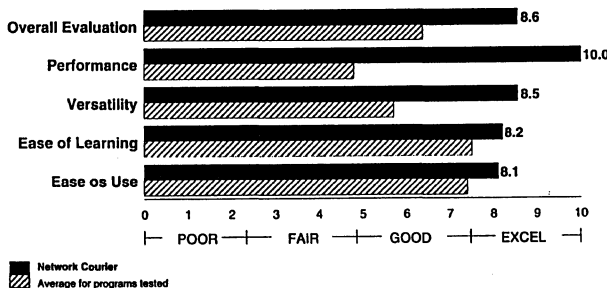


Figure 4.
Sitka InBox Plus Ratings



Figure 5.
Transend CompletE-Mail Ratings



Overall Evaluation

Network Courier and cc:Mail provide good performance, effective user interfaces, and many features including broad internetworking and gateway features. Both come with good documentation and on-line Help. Either program makes a fine LAN E-Mail system.

InBox Plus' features are somewhat limited. It lacks true internetworking, but it integrates mail from Macintosh and PC workstations on the same local network. Advanced features added to Version 3.0, along with easy administration, make InBox Plus useful for small networks.

CompletE-Mail and PMX/StarMail show relative performance and usability weaknesses. Both employ desktop models for message management. CompletE-Mail's MHS basis may be an advantage in environments running other MHS applications. PMX/StarMail can integrate AT&T Mail with signature registration and exchange mail with UNIX systems.

Methodology

The Overall Evaluation is a weighted average of scores for the individual criteria.

Overall Evaluation Score = (2 x Performance Score) + (8 x Versatility Score) + (3 x Ease of Learning Score) + (3 x Ease of Use Score) ÷ 16.

Performance

Network Courier provides outstanding LAN E-Mail message transmission speed. Messages and attachments are exchanged with a minimum of server performance resource utilization, and performance does not degrade sharply as attachment size and/or number of destinations increase.

cc:Mail transmits more slowly than Network Courier, most noticeably as the distribution list increases. cc:Mail goes through a presend step, copying the file for attachments and using file lock arbitration of its Post Office database, but most mail users will find cc:Mail's performance satisfactory.

InBox Plus copies a message for each recipient but sends only one copy of attachments. Performance degrades with messages sent to large distribution lists, but it levels as attachment size increases.

CompleatE-Mail and PMX/StarMail transmission times increase sharply as the number of recipients and message size increase. With respect to Network Courier, the difference is a few seconds for one addressee and a few minutes for large attachments sent to multiple recipients.

Methodology

The Performance Score is an evenly weighted average of scores for individual performance tests. Individual test scores are calculated by dividing the mail system's time into the fastest time for that test and multiplying by 10.

Versatility

Network Courier and cc:Mail provide full sets of message creation/editing features with accessories such as screen capture, and administrative features such as parameterized message deletion and reporting of Post Office message counts. Network Courier provides some features lacking in cc:Mail,

such as multiple system administrators and reporting of attachment counts per user. Either mail system will furnish a full-featured E-Mail environment.

PMX/StarMail offers several routing and connectivity features and is the only mail system that supports hard copy-only mailboxes for users without workstations. Messages can be sent via AT&T Mail with hard copy delivery, including preregistered signatures. Administrative and reporting features are few.

InBox Plus has roots in the Macintosh environment and supports varied configurations for intranetwork routing; it lacks internetwork routing features. Messages and attachments are not encrypted, and administrative features are few.

CompleatE-Mail stresses third-party connectivity options and interoperability with other MHS programs. Reliance on the MHS Post Office structure leaves users with few Post Office administration and reporting options. Messages and attachments are not encrypted.

Methodology

The Versatility rating is a weighted average of scores for individual E-Mail features. Features, grouped into categories, are presented with their methodology weights in Table 3.

Versatility Score = (Electronic Mail Environment Score) + (4 x Internetwork Routing/Gateway Services Score) + (2 x Third-Party Connectivity Score) + (2 x System Administration Score) + (2 x User-Administered Functions Score) + (2 x Post Office Administration Score) + (Mail Directory and Naming System Score) + (Administrative Reporting Score) + (Remote Users Score) + (2 x Message Sending Score) + (2 x Message Creation/Editing Score) + (2 x Message Receiving Score) + (3 x Security Score) + (Mail Notification Score) + (2 x Miscellaneous Features Score) ÷ 28.

Ease of Learning

Network Courier, cc:Mail, and InBox Plus come with excellent documentation that facilitates learning. Network Courier's reference materials include a User's Guide, an Administrator's Guide, a guide to script files, and guides to add-ons such as Modem Mail. cc:Mail provides an extended manual with good organization and clear examples. Separate documentation covers the Remote program.

InBox Plus' clearly written manuals sometimes adopt a condescending style. Reference materials include an Administrator's reference, a DOS user's guide, and a Macintosh user's guide. Small inserts provide important updates and NetWare installation hints.

PMX/StarMail provides user and administrator guides. Directions for setting up user mailbox directories do not clarify the necessary restrictions when using a shared (server) disk for the directories. Careless installation allows one user to overwrite another user's configuration.

CompleatE-Mail's installation requires prior installation of Action Technologies' MHS (not supplied), which ships with its own manual. The separate installation steps can be accomplished by following the manuals, but it is not sufficiently clear how steps in the installation of MHS and CompleatE-Mail interact. Care must be taken with user configuration, and the default MHS user installation conflicts with NetWare 386's password encryption option.

InBox Plus administration functions are easy to learn except for the configuration of InBox Plus mail routers. cc:Mail and Network Courier provide many more options that require commensurate effort in learning to administrate. CompleatE-Mail's administration is complicated as it consists primarily of MHS administration.

On-line Help in cc:Mail, Network Courier, and InBox Plus aids in learning product use. InBox Plus' simple approach to a hypertextual presentation adds a user-friendly touch.

Methodology

After study of the program documentation, evaluators install the mail systems and run administrative and user programs. The Ease of Learning Score is a weighted average of scores for the individual criteria.

Ease of Learning Score = (2 x Installation Score) + (2 x Documentation Score) + (4 x Administration Score) + (2 x Addressing Score) + (2 x Message Editing Score) + (4 x Message Receipt/Management Score) + (On-line Help as Learning Aid Score) + (2 x General Ease of Learning Score) ÷ 19.

Ease of Use

Network Courier combines facile message management and editing, good administration, instructive documentation, and convenient on-line Help.

Addressing in cc:Mail is interrupted by the required key sequence. Message management is good, and editing features are easy to implement considering their wide range. Clear program menus and on-line Help enhance usability.

InBox Plus is reasonably easy to use. Its interface mixes Windows and Macintosh presentation styles, requiring some adjustment. On-line Help makes a very good interactive aid. Administration functions are generally simple, but creating distribution lists involves a multipart screen that could be simplified.

Addressing messages in CompleatE-Mail using an Address Book folder as a model is less streamlined than the address list mechanisms in cc:Mail or Network Courier, which offer instant pop-up lists and speed searches. CompleatE-Mail documentation is terse.

PMX/StarMail's brutally simple (but not user friendly) approach to menuing in its administration program actually makes the program more difficult. Lists (users or distribution lists) are not conveniently displayed along with the corresponding management functions.

Methodology

Evaluators assess the usability of each mail system's creation of user and address lists and their procedures for changing user configurations. Evaluators run the user programs to create messages, add attachments, send and receive mail, and save messages to folders or disk files. The Ease of Use Score is a weighted average of scores for the individual criteria.

Ease of Use Score = (2 x Documentation Score) + (4 x Administration Score) + (2 x Addressing Score) + (2 x Message Editing Score) + (4 x Message Receipt/Management Score) + (On-line Help as Use Aid Score) + (2 x General Ease of Use Score) ÷ 17.

Performance Results

Test Configuration

E-Mail programs were tested on a 16M bps token-ring local area network running Novell NetWare 386 Version 3.1 and using IBM 16/4 Token-Ring interface adapters. The Compaq Deskpro 386/20 server was equipped with a Type 35, 130M-byte ESDI hard disk and 9M bytes of Compaq 32-bit memory. Workstations were a 16MHz AST Premium 386SX with 4M bytes of RAM and an AST Premium 386/25 with 3M bytes of RAM configured with DOS 4.0.

Each mail system's Post Office and executable files were installed in subdirectories on the file server's hard disk using supplied installation scripts or according to recommended configuration defaults. Workstations were mapped to the mail program's executable area on the server. All user mailboxes were located on the file server in order to ensure equal test conditions. Disk access from the file server was faster than access from the workstations in NSTL's test environment. All tests were performed with no other network traffic.

The AST 386/25 workstation ran the mail system's user and administrative programs. The 386SX automated the benchmarks and purged messages from the server disk after each test. To minimize the effects of disk fragmentation, mail was erased from the server after it was sent. Erasing or deleting accumulated mail is especially difficult in PMX/StarMail and Transend's CompleE-Mail, which require that files be explicitly deleted.

Test Procedures

All tests were performed with standard ASCII text messages. The 1,024-byte message was read into the current message buffer and sent from the workstation screen; larger messages were attachments sent from the 386/25 workstation's local hard disk. Each test measured the time required to send the message (with any attachments) from the 386/25 to the Post Office database and associated directory areas on the server. Times were measured from the

initiation of the Send request until return of program control at the workstation, even though some data may still be cached at the server and server disk activity may continue. NSTL ran each test multiple times and averaged the results to account for the dynamic retuning of network characteristics intrinsic to NetWare 386.

NSTL examined server and network utilization using a Network General Sniffer Token-Ring Analyzer to characterize network traffic generated by the mail systems.

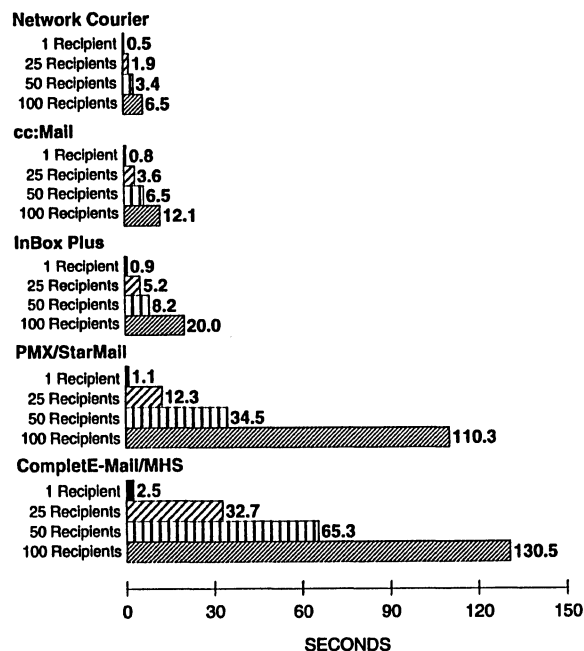
Send 1K-Byte Message

Each mail system sends a 1K-byte text message from a workstation to the local mail database. The test is timed from the Send command until return of control at the workstation. The message is sent first to a single recipient and then distributed to lists of 25, 50, and 100 recipients.

Analysis

Network Courier transmits the message fastest to all numbers of recipients because it moves fewer bytes across the network (about 25 percent fewer bytes than InBox Plus and about 50 percent fewer than cc:Mail for one recipient) and performs relatively few write requests. (For more information on

Figure 6.
Send 1K-Byte Message



NetWare request frame sizes, refer to the analysis for Server Traffic and Network Utilization.)

When sending to one recipient, none of these packages seem excessively slow; even 2.5 seconds is not a noticeable delay. As the distribution list grows, times can increase dramatically. Transend's CompletE-Mail takes much longer to send to larger lists because the program must adhere to the MHS directory and file structure and because the program repeatedly checks volume utilization on the Post Office (server) disk.

For most of the mail systems, transmission times grow in proportion to the number of recipients. Network Courier and InBox Plus show a bit more efficiency with the larger user numbers. PMX/StarMail takes more than twice as long each time the distribution list doubles because the program sends an individual copy of the message to each recipient; the other programs send one copy and create pointers for each recipient.

Send 10K-Byte Attachment

Each mail system sends an empty message with a single 10K-byte text attachment from a workstation to the local mail database. The test is timed from the Send command until return of control at the workstation. The message is sent first to a single recipient and then distributed to lists of 25, 50, and 100 recipients.

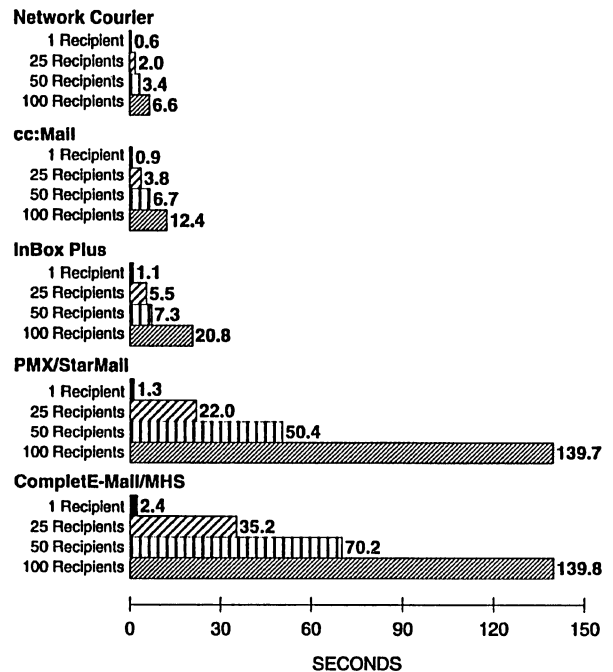
Analysis

Again, Network Courier transmits the message fastest to one and multiple recipients, moving the least amount of data and requiring relatively few file reads to interact with the Post Office database structure.

The larger distribution lists pose the greatest obstacle for PMX/StarMail. StarMail makes multiple copies of the message at the server and compounds the delay with server directory searches.

CompletE-Mail exhibits a nearly linear dependence on distribution list length. Its time per recipient includes the time required to copy the message to a server Post Office directory for each recipient. Processing overhead at the workstation is approximately one second per target as it creates unique message IDs (from the server clock) for interfacing with the MHS system.

Figure 7.
Send 10K-Byte Attachment



Compared with its time for the 1K-byte message, InBox Plus reveals a large fixed overhead per recipient for message distribution independent of the message size.

Send 100K-Byte Attachment

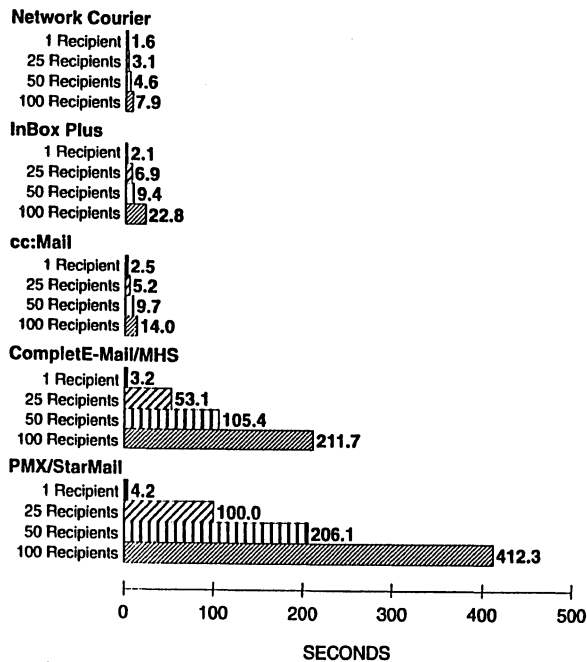
Each mail system sends an empty message with a single 100K-byte text attachment from a workstation to the local mail database. The test is timed from the Send command until return of control at the workstation. The message is sent first to a single recipient, and then distributed to lists of 25, 50, and 100 recipients.

Analysis

PMX/StarMail and CompletE-Mail show an increasing performance disadvantage with larger files and increasing numbers of recipients. StarMail makes a copy of the file for each recipient and CompletE-Mail copies the file to the Post Office directory for each recipient. InBox Plus continues to return times that are almost independent of attachment size.

cc:Mail and Network Courier take more time to send the larger attachment. cc:Mail takes just over 1 second to copy the 100K-byte file to the

Figure 8.
Send 100K-Byte Attachment



message immediately before the Send command (not included in its benchmark times). The copy time is more than reasonable for a large attachment, and it is independent of the length of the distribution list.

Send 500K-Byte Attachment

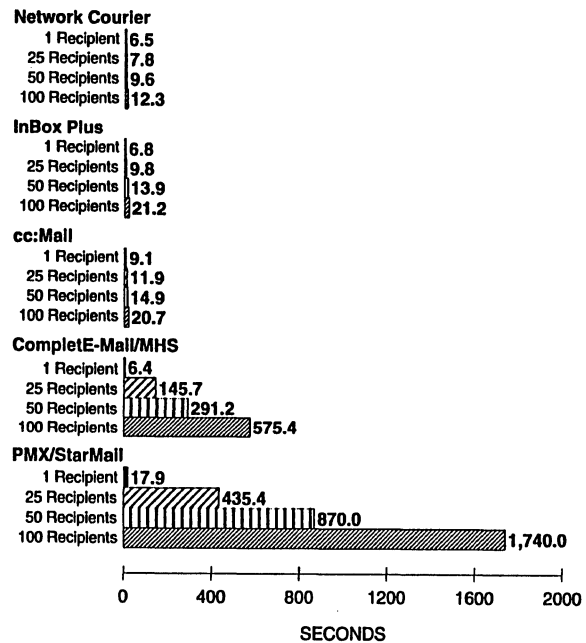
Each mail system sends an empty message with a single 500K-byte text attachment from a workstation to the local mail database. The test is timed from the Send command until return of control at the workstation. The message is sent first to a single recipient and then distributed to lists of 25, 50, and 100 recipients.

Analysis

CompletE-Mail uncharacteristically achieves the shortest transmission time to a single user. Per-user sequencing overhead and the time required to make multiple copies of the large file lower its ranking with transmission to more than one address.

Network Courier performs consistently well with one and multiple recipients. cc:Mail and InBox Plus perform comparably, and InBox Plus' times are barely longer than for smaller attachments.

Figure 9.
Send 500K-Byte Attachment



PMX/StarMail's directory and file overhead, plus the time it takes to make multiple copies of the message, make for comparatively slow performance.

Server Traffic and Network Utilization

NSTL examines server and network utilization with a Network General Sniffer Token-Ring Analyzer to characterize network traffic generated by the mail system. Measurements are taken while the system sends one 1K-byte message to 1 and 25 recipients (timed from the Send command until return of control at the workstation). Network loading to launch the application, verify passwords, or search distribution lists during message preparation is not included. A comparison of a mail system's results for the two scenarios shows where the system saves time by avoiding transmission of redundant information.

Each NetWare request frame is typically acknowledged by a response frame that may contain only an acknowledgement or response data. Frame counts include those sent by the workstation and by the server. Totalling the number of lock, open, read, and write requests generally does not produce the total number of request frames because it does not account for occasional time, directory searches, file size, and other queries.

Table 1a. 1K-Byte Message to One Recipient

	cc:Mail	CompleatE-Mail/MHS	InBox Plus	Network Courier	PMX/StarMail
Frames	64	436	64	72	206
Bytes	15,320	89,542	9,680	7,381	32,869
Locks	2	1	0	0	0
Opens	3	11	2	5	14
Reads	15	161	4	3	31
Writes	6	12	15	21	18

Table 1b. 1K-Byte Message to 25 Recipients

	cc:Mail	CompleatE-Mail/MHS	InBox Plus	Network Courier	PMX/StarMail
Frames	640	1,214	1,486	676	4,808
Bytes	149,936	186,304	239,914	83,971	592,538
Locks	26	25	0	0	0
Opens	27	61	54	79	114
Reads	159	163	131	84	399
Writes	54	108	277	70	1,513

Analysis

As the results show, sending a 1K-byte file generates much more than 1K byte of network usage as the mail system maintains its Post Office data structures by updating data file pointers or creating files in subdirectories.

Network Courier achieves its excellent overall performance partially by producing fewer NetWare frames containing the fewest total bytes. Sending to 25 recipients requires only about 12 times as many bytes, and no lock requests are generated. cc:Mail moves substantially more data across the network than Network Courier does, and it generates lock requests.

InBox Plus transmits with few bytes to one address, but larger distribution lists cause it to send the message multiple times. The problem does not arise when sending an attachment (as opposed to text within the message) to a distribution list because the program sends an attachment only once even to multiple recipients.

CompleatE-Mail generates many frames moving a large number of bytes. In addition to routine file operations, CompleatE-Mail generates server clock queries and checks server volume statistics. One-second delays for each clock query and server overhead compiling volume statistics significantly add to transmission times.

PMX/StarMail generates the largest numbers of frames and bytes. Directory searches, file renames, and erasures add overhead to the conventional file manipulations required to copy the message for each recipient.



Vendors

AT&T

1 Speedwell Avenue
Morristown, NJ 07960 (800) 247-1212

cc:Mail, Inc.

(A division of Lotus Development Corp.)
2141 Landings Drive
Mountain View, CA 94043 (415) 961-8800

Consumers Software, Inc.

73 Water Street
Vancouver, BC, Canada V6B 1A1 (604) 688-4548

Microsoft Corp.

One Microsoft Way
Redmond, WA 98052-6399 (806) 882-8080

Sitka Corp.

950 Marina Village Parkway
Alameda, CA 94501 (415) 769-9669

Transend Corp.

884 Portola Road
Portola Valley, CA 94028 (415) 851-3402

Characteristics

Table 2. LAN E-Mail Program Characteristics

	Requirements	Network Operating System Supported	Supplier Support
AT&T PMX/StarMail (Version 2.2)	<ul style="list-style-type: none"> Workstation Memory: 1MB; 2MB recommended Operating System: DOS 3.1 and higher 	Novell NetWare; UNIX	Telephone support (25 percent of product list price)
cc:Mail (Prerelease Version 3.2)	<ul style="list-style-type: none"> Workstation Memory: 320KB Operating System: DOS 3.1 and higher; OS/2 1.0 and higher 	Novell NetWare; IBM LAN Server; Microsoft LAN Manager; Banyan VINES	Call (800) 448-2500 for details
Consumers Software Network Courier (Version 2.1)	<ul style="list-style-type: none"> Workstation Memory: 640KB Operating System: DOS 3.1 and higher 	Novell NetWare; Banyan VINES; OS/2 LAN Manager; 3Com; DCA 10NET; MS-NET-compatible systems	Telephone support
Sitka InBox Plus (Version 3.0)	<ul style="list-style-type: none"> Workstation Memory: Information not supplied Operating System: DOS 3.1; 3.3 recommended 	Any network that supports file sharing	Telephone support
Transend CompletE-Mail/MHS (Version 1.04)	<ul style="list-style-type: none"> Workstation Memory: 512KB; 640KB recommended Operating System: DOS 3.1 and higher; MHS (available to NetWare installations from Novell, upon request) 	Novell NetWare 2.1 and higher; IBM PC LAN Program; Banyan VINES; MHS compatibles	Telephone support; E-Mail via CompuServe, MIC, Dialcom, AT&T Mail

Table 3. LAN E-Mail Program Features

	Weights	cc:Mail	Transend CompletE-Mail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Electronic Mail Environment	1					
Microsoft Windows version available	1	▲	—	—	▲	▲
OS/2 version available	1	▲	—	—	▲	—
Macintosh version available	1	▲	—	▲	▲	—
Mouse support	0	▲	—	▲	▲	▲
Runs with Novell NetWare	1	▲	▲	▲	▲	▲
Runs with Banyan VINES	1	▲	—	▲	▲	—
Runs with Microsoft LAN Manager	1	▲	—	▲	▲	—
Runs with IBM LAN Server	1	▲	—	▲	▲	—
TSR version included	3	▲	▲	—	▲	▲

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend CompletE-Mail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Post Office Parameters	0					
Maximum users (mailboxes) per Post Office	1	U	600	250	500	U
Maximum messages per mailbox	1	U	300	U	U	U
Maximum public mailing lists	1	200	MHS	U	U	U
Maximum entries per mailing list	1	200	MHS	U	U	U
Internetwork Routing and Gateway Services	4					
Routing by proprietary software	4	▲	—	▲	▲	▲
Routing by MHS	0	A	▲	—	A	—
Multiple local Post Offices	2	▲	—	▲	▲	—
Multiple Post Offices over bridged LANs	2	▲	MHS	▲	▲	▲
Addressing to foreign systems	1	▲	—	▲	▲	—
Connection service to remote user	1	▲	▲	▲	▲	—
Routing service in nondedicated station	0	▲	—	▲	▲	▲
Peer-to-peer connections	1	▲	MHS	▲	▲	▲
Master-slave connections	1	▲	MHS	▲	▲	—
Multiple-hop routing	2	▲	MHS	▲	▲	▲
Limit hop count	1	—	—	—	▲	—
Circular routing detection	1	—	—	▲	▲	—
Set call interval for remote connections	1	▲	▲	▲	▲	—
Schedule remote connections	1	▲	▲	▲	▲	▲
Call retry settings	1	▲	MHS	—	▲	—
Set retry interval	1	▲	—	—	▲	—
Remote connection password	1	▲	MHS	—	▲	—
Bypass call schedule	1	▲	—	—	B	▲

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Com- pletE-Mail/MHS	Sitka InBox Plus	Consumers Soft- ware Network Courier	AT&T PMX/StarMail
Minimum mes- sages queued before sending	1	▲	MHS	—	—	—
Maximum mes- sage size	1	▲	—	—	▲	—
Send-only connections	1	▲	MHS	—	▲	▲
Disable sending when polled	1	—	—	—	▲	▲
Reset modem to auto answer	1	▲	—	—	—	C
Toll call reporting	1	—	—	—	▲	—
Routing and de- livery statistics	1	▲	—	▲	▲	▲
Automatic file compression	1	▲	—	—	—	—
Encrypt messages	1	▲	—	—	▲	—
Disable mail receipts	1	—	—	—	▲	—
Supports Telebit Trailblazer	2	▲	—	—	▲	—
Supports Hayes V-Series 9600	2	▲	—	▲	▲	—
Dead letter management	1	▲	▲	▲	▲	—
Provides routing service APIs	2	▲	—	—	▲	▲
Third-Party Connectivity	2					
IBM PROFS	2	▲	▲	▲	▲	D
IBM DISOSS	2	▲	—	—	▲	—
Digital ALL-IN-1	1	▲	▲	—	▲	—
Digital VMS Mail	2	▲	▲	—	—	—
Hewlett-Packard Desk	0	▲	▲	—	—	—
MCI Mail	2	▲	▲	—	▲	—
MHS	1	▲	▲	—	▲	—
X.400	2	▲	▲	—	▲	D
SprintNet TeleMail	1	▲	▲	—	▲	—
3Com 3+ Mail	1	▲	—	—	—	—
UNIX Mail	1	▲	▲	▲	▲	▲
Fax connectivity	2	▲	▲	▲	▲	D
AT&T Mail	1	—	—	▲	—	▲
System Adminis- tration Functions	2					
Change user password	2	▲	MHS	▲	▲	▲
Enable/disable mailbox	1	—	—	—	—	—

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Complete-Mail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Receive-only mailbox	1	—	—	—	▲	—
Send-only mailbox	1	—	—	—	▲	—
Restrict remote transmission	1	—	—	—	▲	—
Limit mailbox size	1	—	—	—	—	—
Unlisted mailbox address in public directories	1	—	—	—	—	—
Hard copy-only mailbox	0	—	—	—	—	E
Public distribution lists	2	▲	—	▲	▲	▲
User directory templates	1	—	—	—	▲	—
Forward mail	1	▲	—	—	▲	—
Folder access rights	1	—	—	—	▲	—
Global defaults for adding users	0	—	—	—	▲	—
Folder creation rights	1	—	—	—	▲	▲
Public remote user aliases	1	▲	—	▲	▲	▲
User-Administered Functions						
Change password	2	▲	▲	▲	▲	▲
Define mailbox on local disk	1	—	—	▲	—	▲
Switch mailbox with password	2	—	—	▲	▲	—
Change color scheme	1	▲	▲	—	▲	▲
Enable/disable send notification	0	—	▲	—	—	—
Create private distribution lists	2	▲	▲	▲	▲	▲
Create personal alias for user/group	1	F	▲	F	▲	▲
Assign message life on remote transmissions	1	—	—	—	▲	—
Post Office Administration						
Multiple system administrators	1	—	—	▲	▲	—
Change Post Office administrator	0	▲	—	—	▲	—
Change administrator password	0	▲	MHS	▲	▲	▲

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Comple- tE-Mail/MHS	Sitka InBox Plus	Consumers Soft- ware Network Courier	AT&T PMX/StarMail
Change Post Of- fice name	0	▲	MHS	▲	▲	▲
Change Post Of- fice password	0	▲	MHS	▲	▲	▲
Determine mes- sage status (read/unread)	1	▲	—	—	▲	—
Delete messages per user	2	▲	—	—	▲	—
Delete messages sent within date range	1	—	—	—	▲	—
Delete messages sent before date	2	▲	—	—	▲	—
Delete messages sent before date, by user	2	▲	—	—	▲	—
Delete messages already read	2	▲	—	—	▲	—
Delete messages already read, by user	2	▲	—	—	▲	—
Compress indi- vidual messages	1	G	—	—	—	—
Compress indi- vidual messages in folders	1	G	—	—	—	—
Mail Directory and Naming System						
NOS naming scheme	3	—	MHS	—	—	—
Directory lists full username	1	▲	▲	▲	▲	▲
Directory lists mail login name	0	—	—	—	▲	▲
Directory in- cludes user description	1	▲	—	—	▲	▲
Local/remote designation	1	▲	—	—	▲	▲
Directory distri- bution to remote Post Offices	1	H	—	▲	I	—
Multipart hierar- chical naming system	1	—	▲	—	▲	▲
Maximum char- acters per username	1	126	8	32	30	20
Maximum char- acters per login name	0	NA	8	NA	10	8

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend CompletE-Mail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Administrative Reporting						
User data	1	▲	—	—	▲	—
Distribution list data	1		—	—	▲	▲
Message count in Post Office	2	▲	—	—	▲	—
Message count by user	1	▲	—	—	▲	—
Attachment count by user	1	—	—	—	▲	—
Oldest message in Post Office by user	1	—	—	—	▲	—
Unread messages by user	1	▲	—	▲	▲	—
Read messages by user	1	▲	—	▲	—	—
Messages sent before date	1	▲	—	—	▲	—
Messages sent before date, by user	1	▲	—	—	▲	—
Messages received within date range	1	—	—	—	▲	—
Post Office storage statistics	2	▲	—	—	▲	—
Mail items sent to remote Post Offices	1	▲	—	▲	▲	▲
Bytes sent to remote Post Offices	2	▲	—	▲	▲	—
Priority items sent to remote Post Offices	1	—	—	—	—	—
Calls made to remote Post Offices	2	—	—	—	▲	▲
Statistical/diagnostic reports to file	1	▲	—	▲	▲	J
Statistical/diagnostic reports to screen	1	▲	—	▲	▲	▲
Statistical/diagnostic reports to printer	1	▲	—	▲	▲	J
Remote connection information	1	▲	—	▲	▲	▲
Remote Users						
Remote dial-in	2	▲	▲	▲	▲	—
Automatic delivery	1	▲	—	▲	▲	—

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend CompletE-Mail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Background operation	1	—	▲	—	▲	—
Batch messages for transmission	1	▲	—	—	▲	▲
Point-to-point remote connections	1	▲	▲	—	—	—
Access local mailbox	1	▲	—	▲	▲	—
Message Sending	2					
Multiple recipients (To:)	1	▲	▲	▲	▲	▲
Multiple carbon copies (CC:)	1	▲	▲	▲	▲	▲
Multiple blind carbon copies (BCC:)	1	▲	▲	▲	—	▲
Longest Subject: line	1	60	66	30	40	21
Address message to folder	1	▲	—	—	▲	—
Address message to bulletin board	1	▲	—	—	▲	—
Send to public distribution list	2	▲	—	▲	▲	▲
Send to private distribution list	2	▲	▲	—	▲	▲
Registered mail (receipt requested)	2	▲	▲	▲	▲	▲
Set mail priority levels	1	▲	▲	▲	▲	K
User directory lookup	1	▲	▲	▲	▲	▲
Search directory	1	—	—	▲	▲	—
Message encryption	2	▲	▲	—	▲	L
Attachment encryption	2	▲	—	—	▲	▲
Message Creation and Editing	2					
Block operations	1	▲	▲	▲	▲	▲
Automatic word wrap	1	▲	▲	▲	▲	▲
Reformatting after deletions	1	M	▲	▲	▲	▲
Cut and paste within message	1	▲	▲	▲	▲	▲
Cut and paste to application	1	—	—	▲	—	—
Import DOS text file	1	▲	▲	▲	▲	▲

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Com- pletE-Mail/MHS	Sitka InBox Plus	Consumers Soft- ware Network Courier	AT&T PMX/StarMail
Import mail messages	1	▲	▲	▲	▲	▲
Maximum message length (char.)	1	20K	64K	32 pp.	32K	1,500
Add attachments	4	▲	▲	▲	▲	▲
Maximum attachments per message	2	20	U	U	U	49
Maximum attachment size	1	U	U	U	U	U
Attachment description field	1	▲	—	—	—	▲
Text screen capture	2	▲	—	—	▲	—
Graphic screen capture	2	▲	—	—	▲	—
Graphics editor	1	▲	—	—	—	—
Revise unsent messages	2	N	▲	N	N	▲
Revise sent messages	1	▲	—	N	N	▲
DOS directory lookup	2	▲	▲	▲	▲	▲
Indicate attachment size	1	▲	—	—	—	—
Electronic forms	0	—	▲	Q	—	▲
Create/edit multiple messages	0	—	—	▲	—	—
Message Receiving	2					
Flag unread messages in Inbox	1	▲	—	▲	▲	▲
Flag messages with attachments	1	▲	—	▲	▲	▲
Move message to folder	1	▲	▲	▲	▲	▲
Copy message to folder	1	▲	▲	▲	▲	▲
Save selected attachments to folder	1	—	—	▲	▲	—
Archive messages	1	▲	▲	▲	▲	—
Save message to disk	2	▲	▲	▲	▲	▲
Append message to DOS file	1	R	—	▲	▲	—
Print message	2	▲	▲	▲	▲	▲
Read next option	1	▲	▲	▲	▲	▲
Read previous option	1	—	▲	—	▲	▲

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Com- pletE-Mail/MHS	Sitka InBox Plus	Consumers Soft- ware Network Courier	AT&T PMX/StarMail
View attachments	1	▲	—	▲	▲	▲
Save attachments to disk	1	▲	▲	▲	▲	▲
Print attachments	1	▲	S	▲	—	▲
Reply to sender	2	▲	▲	▲	▲	▲
Automatic reply to sender/ recipients	1	▲	▲	▲	▲	▲
Reply to other recipients	1	▲	▲	▲	▲	▲
Reply attachments	1	▲	▲	▲	▲	▲
Copy original message to reply	1	▲	—	▲	▲	▲
Forward message	2	▲	▲	▲	▲	▲
Forward to multiple recipients	1	▲	▲	▲	▲	▲
Forward with annotations	1	▲	▲	▲	▲	▲
Forward with added attachments	1	▲	▲	▲	▲	▲
Forward with changes	0	▲	▲	▲	—	▲
Forward without attachments	1	▲	▲	▲	▲	▲
Security 3						
Post Office password	0	▲	MHS	▲	▲	▲
Administrator password	0	▲	—	—	▲	▲
User password	2	▲	▲	▲	▲	▲
Password encryption	4	▲	—	—	▲	▲
Message encryption	4	▲	—	—	▲	L
Encryption during transmission	2	▲	—	—	▲	—
Password encryption key	2	▲	—	—	—	—
Encrypt attachments	3	▲	—	—	▲	L
Message Search and Display 1						
Search from starting date	1	▲	—	T	▲	—
Search within date range	1	▲	—	T	▲	—
Search by sender	2	▲	—	T	▲	V

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend Com- pletE-Mail/MHS	Sitka InBox Plus	Consumers Soft- ware Network Courier	AT&T PMX/StarMail
Search by subject	2	▲	—	T	▲	V
Search for keyword	1	—	▲	▲	▲	▲
Display unread mail only	1	▲	—	—	—	—
Display mail with attachments only	1	—	—	—	—	—
Display CC: or BCC: mail only	0	—	—	—	—	—
Display priority mail only	1	▲	—	—	—	—
Display encrypted mail only	0	—	—	—	—	—
Folder Creation and Management	2					
Create folders	2	▲	X	▲	▲	▲
Delete folders	2	▲	—	▲	▲	▲
Rename folders	2	▲	▲	▲	▲	—
Create subfolders	1	—	—	—	▲	—
Move messages between folders	1	▲	▲	—	▲	▲
Copy messages between folders	1	▲	▲	—	▲	▲
Move/copy multiple messages between folders	1	▲	▲	—	▲	▲
Delete multiple messages	1	▲	▲	—	▲	▲
Maximum folders per user	1	200	16	U	32K	99
Maximum messages per folder	1	1,600	100	U	32K	99
Wastebasket folder	1	—	▲	W	—	▲
Log folder for outgoing mail	1	▲	▲	▲	▲	▲
Log folder for incoming mail	1	—	▲	—	▲	▲
Sort folder by date	1	▲	—	—	▲	—
Sort folder by sender	1	—	—	—	▲	—
Sort folder by priority	1	—	—	—	▲	—
Sort folder by subject	1	—	—	—	▲	—
Archive folder to DOS file	1	▲	—	—	▲	—
Retrieve archive folder	1	▲	—	—	▲	—

See footnotes on Page 428.

Table 3. LAN E-Mail Program Features (Continued)

	Weights	cc:Mail	Transend CompleMail/MHS	Sitka InBox Plus	Consumers Software Network Courier	AT&T PMX/StarMail
Copy message to DOS file	1	▲	▲	—	▲	—
Folder comment field	1	—	—	—	▲	—
Cut and paste from folder	1	—	—	—	▲	—
Mail Notification 1						
Background notification by polling	1	▲	—	▲	▲	▲
Background notification via NETBIOS	1	—	—	—	▲	—
DOS-level check for mail	1	▲	—	▲	▲	—
Miscellaneous Features 2						
Nested distribution lists	2	—	—	—	—	Z
Bulletin board	1	▲	—	▲	▲	—
Context-sensitive Help	0	▲	▲	▲	▲	—
Quick reference guide	0	▲	—	—	▲	▲
Tutorial guide	0	—	—	Y	▲	—
API libraries	4	▲	—	—	▲	▲
Demo disk	0	—	▲	—	▲	—
DOS shell	0	▲	▲	▲	▲	▲
Import/export Post Office files	3	▲	—	—	▲	—
Background notification via the NetWare Send command	1	—	▲	—	—	—

▲—Yes, has feature.

A—MHS can be used optionally.

B—Separate schedule can be entered for urgent messages.

C—User can create logoff script for this purpose.

D—Available with routing through UNIX server or AT&T Mail network.

E—Available through AT&T Mail network.

F—Must create a single-name mailing list.

G—Already compressed in Post Office.

H—Using separate DIRPROP utility.

I—Using Export utility.

J—Route to printer or file from DOS command line.

K—Mark for special attention.

L—Available through third-party product.

M—Alt-F8 reformats.

MHS—MHS function.

N—Save to and retrieve from a folder.

NA—Not applicable.

Q—Provides predefined form simulating telephone messages.

R—Using Archive function only.

S—Can print any file, including files saved to attachment directory.

T—Can change display sort order based on these criteria.

U—Unlimited.

V—Single option searches message header.

W—Deleted messages can be undeleted.

X—Comes with 16 predefined folders; no additional folders can be added.

Y—Some tutorial exercises provided in main manual.

Z—Available in UNIX version.

Glossary of Terms

Electronic Mail Environment

Microsoft Windows Version Available: A version of the program is written specifically to operate as a windowed, pull-down menu, mouse-oriented Microsoft Windows application.

OS/2 Version Available: A version of the program is available to run under OS/2, not just as a DOS application running in the Compatibility box.

Macintosh Version Available: A version of the program is available to run on Macintosh systems.

Mouse Support: The program supports the use of a mouse in conjunction with the keyboard for performing functions such as item selection, text manipulation, or graphics creation.

Runs with Novell NetWare: The program can operate on a Novell network.

Runs with Banyan VINES: The program can operate on a Banyan VINES network.

Runs with Microsoft LAN Manager: The program can operate in a Microsoft LAN Manager OS/2 environment.

Runs with IBM LAN Server: The program can operate on an IBM LAN Server network.

TSR Version Included: A reduced version of the program runs as a memory-resident TSR.

Post Office Parameters

Maximum Users (mailboxes) per Post Office: This is the maximum number of users that can send and receive mail via a single Post Office. If each user is assigned a single mailbox, this number is also the maximum mailboxes per Post Office.

Maximum Messages per Mailbox: This is the maximum number of mail messages that can be stored in a mailbox data area in a Post Office on a file server.

Maximum Public Mailing Lists: This is the maximum number of public mailing lists that can be created and

maintained by the system administrator for use by any mail system user. A public mailing list is a distribution list of mail users.

Maximum Entries per Mailing List: This is the maximum number of user mailbox names or user full names that can be stored in a public mailing list.

Internetwork Routing and Gateway Services

Routing by Proprietary Software: A vendor-developed communications system allows two Post Offices to exchange mail and user information. A proprietary solution typically provides better integration and performance than a third-party internetwork transport product such as MHS.

Routing by MHS: Action Technologies' Message Handling System (MHS) internetwork transport mechanism connects remote Post Offices, remote users, or third-party systems.

Multiple Local Post Offices: The internetwork routing system, which usually resides on a single network PC, can manage external communications for multiple Post Offices on the local network.

Multiple Post Offices over Bridged LANs: The internetwork routing system can manage communications among multiple Post Offices located locally and at remote sites connected by network bridges.

Addressing to Foreign Systems: The program can set up user addresses that do not correspond to its own addressing scheme. Foreign addressing routes mail to external systems with different naming conventions.

Connection Service to Remote User: Systems not physically connected to a LAN running E-Mail can access the LAN mail system via a dial-up connection. Remote systems must be running a remote version of the E-Mail software.

Routing Service in Nondedicated Station: The internetwork routing software can run in the background on a PC or as a process under a Windows or Desqview type of environment manager.

Peer-to-Peer Connections: The internetwork routing system can initiate connections to other external routing systems and perform all of the call-setup, send, receive, and call-terminate processes required to complete a transfer. Systems capable of peer-to-peer connections can set up multiple connections to multiple sites.

Master-Slave Connections: The program can define one internetwork routing system as the "master" that initiates calls to other "slave" routing systems and requests information as required. Slave systems usually do not initiate outbound calls or perform routing tasks.

Multiple-Hop Routing: The path required to transport mail from one location to another may be routed through intermediate locations; thus, messages "hop" from one location to another until they reach their destination. All intermediate routing locations understand how to route messages properly.

Limit Hop Count: The program can limit the number of hops in a mail route so that undelivered messages are returned to the sender or administrator for proper handling. Limiting the hop count counters circular routing (i.e., an intermediate Post Office routes mail through the sending Post Office) that may prevent mail from reaching its destination.

Circular Routing Detection: The internetwork system can detect circular message routing that prevents a message from reaching its destination. The system flags the circular route for the network administrator or reports the problem in a log file.

Set Call Interval for Remote Connections: The administrator can define the interval (e.g., hourly) between successive connections to a remote Post Office.

Schedule Remote Connections: The administrator can define a time of day for connection between two remote Post Office systems.

Call Retry Settings: The administrator can specify the number of times the internetwork routing system will reattempt connection to a remote Post Office after an unsuccessful attempt.

Set Retry Interval: The administrator can define the time interval between successive call retries.

Remote Connection Password: A system can verify that it should accept a call from another Post Office based on a predetermined password.

Bypass Call Schedule: The internetwork routing system can be forced to send an Urgent message immediately to a remote Post Office, bypassing any preset call schedule. The message must be flagged as being Urgent.

Minimum Messages Queued before Sending: The internetwork routing system can be set up to dial a remote Post Office only when a minimum number of messages are queued for sending.

Maximum Message Size: A maximum message size can be applied to transfers over remote connections to control the volume of data across a communications link, reducing costs, and provide room for more messages.

Send-Only Connections: The internetwork routing program can be set up to send mail to another routing program, but not retrieve messages (i.e., to control costs).

Disable Sending When Polled: The internetwork routing system's sending capabilities can be disabled such that the system receives mail from a polling system but does not send mail in return.

Reset Modem to Auto Answer: The internetwork routing program automatically resets the modem answer mode after completing a transmission sequence to the remote Post Office(s). Modem resetting ensures that the Post Office can receive mail from other Post Offices.

Toll Call Reporting: A program function tracks long-distance calls to remote routing systems for accounting and cost-tracking purposes.

Routing and Delivery Statistics: Each Post Office routing system can generate reports showing outbound and inbound internetwork traffic statistics that aid in traffic engineering, problem determination, and cost control.

Automatic File Compression: Mail bundles to be sent between two routing systems are automatically compressed (using compression algorithms) when sent.

Encrypt Messages: The program can encrypt messages during modem transfer to prevent disclosure of sensitive data via line monitoring.

Disable Mail Receipts: When certified or registered mail is sent over an internetwork routing link, a receipt is automatically returned when the recipient reads the message. The receipt option can be disabled, preventing the response from traveling back across the communications link.

Supports Telebit Trailblazer: The internetwork routing program works with the Telebit Trailblazer modem's 9600 bps mode and possibly its high-speed 19.2K bps mode.

Supports Hayes V-Series 9600: The internetwork routing program works with the Hayes 9600 V-Series modem's 9600 bps mode and possibly its high-speed 19.2K bps mode.

Dead Letter Management: Mail sent to a remote destination, but not delivered (i.e., circular routing, exceeding the hop count, or exceeding useful life), is either returned to the sender or delivered to a special area for handling by the administrator.

Routing Service APIs: The message transport mechanism provides a well-defined set of subroutine modules that can be called by application software.

Third-Party Connectivity

IBM PROFS: The mail system can exchange mail with IBM's Professional Office System.

IBM DISOSS: The mail system can exchange mail with IBM's Distributed Office Support System.

Digital ALL-IN-1: The mail system can exchange mail with Digital Equipment's ALL-IN-1 office automation system.

Digital VMS Mail: The mail system can exchange mail with Digital Equipment's VMS Mail system.

Hewlett-Packard Desk: The mail system can exchange data with Hewlett-Packard's Desk system.

MCI Mail: The mail system can exchange mail with the MCI Mail system.

MHS: The mail system can use Action Technologies' Message Handling System to transport mail documents to other MHS-compatible systems.

X.400: The mail system supports the CCITT X.400 international messaging format for exchanging mail with other X.400-compatible systems.

SprintNet TeleMail: The mail system can exchange mail with the SprintNet TeleMail system.

3Com 3+Mail: The mail system can exchange mail with 3Com local area network mail systems.

UNIX Mail: The mail system can exchange mail with the UNIX operating system's mail system.

Fax Connectivity: Mail system users can exchange fax messages with a fax device located on the local area network via the mail system.

AT&T Mail: The mail system can exchange mail with the AT&T Mail system.

System Administrator Functions

Change User Password: The system administrator can define and modify passwords required by users to sign on to the mail system.

Enable/Disable Mailbox: The system administrator can render a mailbox inactive when a user is unable to respond to messages for a period of time. Senders are notified that the recipient's mailbox is inactive.

Receive-Only Mailbox: The system administrator can restrict a user's mail rights to receiving mail only.

Send-Only Mailbox: The system administrator can restrict a user's mail rights to sending mail only.

Restrict Remote Transmission: The system administrator can selectively restrict users from sending mail to remote Post Offices as a cost control or security measure.

Limit Mailbox Size: The system administrator can limit the amount of Post Office storage space consumed by user mailboxes by imposing a maximum number of messages or bytes per mailbox. The limit does not apply to mailboxes located on workstation disk drives.

Unlisted Mailbox Address in Public Directories: The administrator can prevent a user's address from being displayed in public directories. An unlisted address (typically an executive) will not have to receive mail from all other mail users (typically subordinates).

Hard Copy-Only Mailbox: The administrator can designate a user mailbox as being "hard copy only," and all messages directed to the user are printed on the network printer and delivered by hand. With this feature, the distribution list for messages can include persons without computers or otherwise not connected to the mail system.

Public Distribution Lists: The system administrator can designate mailing lists that are accessible to all mail system users. Public distribution lists contain names of users who may belong to specific groups or departments. Mail system users can send mail to all users on a list without entering the names individually.

User Directory Templates: The administrator can create a template file with fields for storing user information in the Names Directory. The template file (not displayed by default) can be selected by users requiring more information.

Forward Mail: The administrator can define a mailbox forwarding address (i.e., another user mailbox) for handling mail when the addressee is unable to respond to mail.

Folder Access Rights: The administrator can define user access to shared system-wide and/or private folders.

Global Defaults for Adding Users: The administrator can define a set of common user characteristics (e.g., ability to send priority mail) to facilitate adding users to the mail system.

Folder Creation Rights: The administrator can designate whether a user can create shareable system-wide and/or private folders.

Public Remote User Aliases: The administrator can assign a shortened name (or alias) to a remote Post Office user (i.e., to replace a two- or three-part name such as Nick@Marketing@Chicago).

User-Administrated Functions

Change Password: A user can change his or her own mail login password.

Define Mailbox on Local Disk: A user can store mail messages in a mailbox residing on a local hard disk; storing messages locally enhances security.

Switch Mailbox with Password: A user can access multiple users' mail given the appropriate passwords; for example, an employee may be given access to mailboxes for superiors without having to exit the current mail session.

Change Color Scheme: Given a color monitor, the user can change the mail software's display color scheme.

Enable/Disable Send Notification: When sending mail, the user can control whether notification is transmitted to the recipient.

Create Private Distribution Lists: The user can create lists of mail system users from the names in the public mail system directory. Mail can be distributed to all users on these private lists.

Create Personal Alias for User/Group: The user can designate a shortened name (or alias) for a particular user or group address. The alias is substituted for a Post Office directory name.

Assign Message Life on Remote Transmissions: The user can place a time limit (or message life) on mail sent to remote Post Offices; the user is notified if the message is not delivered within this time frame.

Post Office Administration

Multiple System Administrators: More than one user can act as the administrator.

Change Post Office Administrator: The mail system administrator can assign another user as mail system administrator.

Change Administrator Password: The mail system administrator can change the Administrator password.

Change Post Office Name: The administrator can rename the Post Office.

Change Post Office Password: The administrator can change the Post Office password.

Determine Message Status (read/unread): The administrator can determine whether messages have been read in order to maintain message storage and monitor Post Office usage.

Delete Messages per User: The administrator can delete all messages stored in the Post Office mailroom and/or user mailbox for any mailbox located on the file server.

Delete Messages Sent within Date Range: The administrator can delete all messages stored in the Post Office mailroom and/or user mailboxes (on the file server) that were sent between specific dates.

Delete Messages Sent before Date: The administrator can delete all messages stored in the Post Office mailroom and/or user mailboxes (on the file server) that were sent before a specific date.

Delete Messages Sent before Date, by User: The administrator can delete all messages stored in the Post Office mailroom and/or user mailbox (on the file server) that were sent to a specific user before a specific date.

Delete Messages Already Read: The administrator can delete all messages stored in the Post Office mailroom and/or residing in user mailboxes on the file server if the messages have already been read by recipients.

Delete Messages Already Read, by User: The administrator can delete all messages stored in the Post Office mailroom and/or residing in a specific user mailbox on the file server if they have already been read.

Compress Individual Messages: A compression algorithm can be used to compress individual mail files, reducing their actual size while they reside in the Post Office mailroom and/or in user mailboxes on the file server.

Compress Individual Messages in Folders: A compression algorithm can be used to compress individual mail files stored in folders, reducing their actual size while they reside in the Post Office mailroom and/or in user mailboxes on the file server.

Mail Directory and Naming System

NOS Naming Scheme: The mail system supports the designated Network Operating System (NOS) user naming scheme. Using the NOS naming scheme accelerates setup in a large network.

Directory Lists Full Username: The Post Office mail directory includes each user's full name, which can be included in mail messages, displayed in distribution lists, or used for administrative tasks.

Directory Lists Mail Login Name: A shorter mail login name is included in the mail system directory; it is often used exclusively for user login purposes.

Directory Includes User Description: Each user entry in the mail system directory can include the user's title and function and other descriptive information.

Local/Remote Designation: The mail system directory indicates whether users exist locally on the network or in a remote location (e.g., a field indicating local or remote status; address field indicating a remote address).

Directory Distribution to Remote Post Offices: When new users are added to a mail system, the naming information is sent over the communications link to a remote system. This very powerful tool assists in administering mail systems in a multisite network; nearly all the systems require administrator-initiated distribution.

Multipart Hierarchical Naming System: A user's full address may be comprised of multiple components; for example, Location Name/Post Office Name/Username; or Network Name/Post Office Name/Username; or Location Name/Username.

Maximum Characters per Username: This is the maximum number of character spaces allotted in the mail directory for a local mail system user's full name.

Maximum Characters per Login Name: This is the maximum number of character spaces allotted in the mail directory for a local user's login name.

Administrative Reporting

User Data: The system can report user information such as name, description, password, and message status to the administrator.

Distribution List Data: The system can report all existing public distribution lists and their members.

Message Count in Post Office: The system can report the number of messages stored in the Post Office mailroom and/or user mailboxes on the file server.

Message Count by User: The system can report the total number of messages stored in the Post Office mailroom and/or user mailbox on the file server for a specified user.

Attachment Count by User: The system can report the total number of attachments stored with mail messages in the Post Office mailroom and/or user mailboxes.

Oldest Message in Post Office by User: For each user, the system can report the oldest message waiting for pickup (not read) in the Post Office mailroom.

Unread Messages by User: The system can report the number of unread messages in the Post Office mailroom and/or the user's mailbox on the file server.

Read Messages by User: The system can report the number of read messages in the user's mailbox on the file server.

Messages Sent before Date: The system can report the number of messages sent before a specific date that exist in the Post Office mailroom and/or user mailboxes on the file server.

Messages Sent before Date, by User: The system can report the number of messages sent to a specific user before a specific date that exist in the Post Office mailroom and/or user's mailbox on the file server.

Messages Received within Date Range: The system can report the number of messages received in the Post Office between specific dates that exist in the Post Office mailroom and/or user mailboxes on the file server.

Post Office Storage Statistics: The system can report statistics for the Post Office database, such as the amount of file server storage space used by mail files and database efficiency in terms of fragmentation.

Mail Items Sent to Remote Post Offices: The system can report the number of mail items sent to another Post Office within a specified time range.

Bytes Sent to Remote Post Offices: The system can report the total size of mail items transferred to another Post Office within a specified time range.

Priority Items Sent to Remote Post Offices: The system can report the number of urgent or priority messages sent to another Post Office within a specified time range.

Calls Made to Remote Post Offices: The system can report the number of calls made to a remote Post Office during a specified time range. This feature is useful in monitoring remote mail communications traffic patterns.

Statistical/Diagnostic Reports to File: Post Office statistics and diagnostic information can be directed to a disk file from within the mail administration system.

Statistical/Diagnostic Reports to Screen: Post Office statistics and diagnostic information can be directed to the screen from within the mail administration system.

Statistical/Diagnostic Reports to Printer: Post Office statistics and diagnostic information can be directed to a printer from within the mail administration system.

Remote Connection Information: The administrator can display and inspect communication and login parameters defined for accessing a remote Post Office.

Remote Users

Remote Dial-In: The mail package supports mail transmissions via modem to and from a remote workstation running a copy of the E-Mail software.

Automatic Delivery: A remote user notifies the local Post Office administrator when the remote user system is in receive mode, and at that time the local Post Office can send mail addressed to that remote user.

Background Operation: A remote user can transmit and receive messages in the background while other operations are performed.

Batch Messages for Transmission: Multiple messages for a destination can be bundled at a remote system and sent at once. The receiving Post Office sorts and forwards individual messages.

Point-to-Point Remote Connections: Remote user mail systems can establish connections and transfer messages without going through a central Post Office.

Access Local Mailbox: A remote user can exchange mail with a mailbox on the local area network (given access rights to that mailbox); typically, a remote user is assigned a local mailbox.

Message Sending

Multiple Recipients (To:): A message can be addressed to more than one user designated in one or more To: fields.

Multiple Carbon Copies (CC:): A message can be addressed to more than one user designated in one or more CC: ("courtesy copies") fields.

Multiple Blind Carbon Copies (BCC:): The message can be addressed to more than one BCC: recipient designated in one or more BCC: fields. A message goes to a BCC: recipient without other recipients knowing that the BCC: user is receiving the mail.

Longest Subject: Line: The maximum number of characters the sender can enter into the subject field of a message.

Address Message to Folder: A message can be addressed to one or more user mail folders where it is stored for archiving or retrieval.

Address Message to Bulletin Board: The user can send messages to a public bulletin board in the mail system to which all system users have access.

Send to Public Distribution List: The sender can enter the name of a public distribution list in any recipient field, and the message will be sent to all users in the list.

Send to Private Distribution List: The sender can enter the name of a personal distribution list in any recipient field, and the message will be sent to all users in the list.

Registered Mail (receipt requested): The sender can request notification that the recipient has read the message.

Set Mail Priority Levels: Mail messages can be assigned priority levels indicating levels of urgency or importance.

User Directory Lookup: The sender can display a directory list of mail system users and select recipients. Selected usernames are automatically inserted into the message recipient fields.

Search Directory: The sender can search the mail system directory for a particular name (mail address) to see whether the user is accessible by mail. The displayed address can be automatically entered in the message recipient field.

Message Encryption: The mail system encrypts user messages sent to the Post Office to foil Post Office intruders or hackers. LAN eavesdroppers cannot read encrypted messages even with sophisticated LAN monitoring equipment.

Attachment Encryption: The mail system encrypts message attachments to foil Post Office intruders or hackers.

Message Creation and Editing

Block Operations: Blocks of message text can be marked for copy, delete, and move operations.

Automatic Word Wrap: The message entry screen automatically maintains a right margin without manual carriage returns.

Reformatting after Deletions: The message entry screen automatically justifies and reformats text to adjust for deletions.

Cut and Paste within Message: Text blocks can be moved or copied from one location to another within a message.

Cut and Paste to Application: Message text can be moved directly into an application or from an application to a message (used with a TSR version of the mail package running over an application).

Import DOS Text File: The system can read a DOS file from disk into a message.

Import Mail Messages: All or part of another mail message can be integrated into the current message.

Maximum Message Length: This is the maximum number of characters or lines that can be entered as a single message.

Add Attachments: The system can append a DOS file (plain ASCII text file or a binary) to a mail message.

Maximum Attachments per Message: This is the maximum number of DOS file attachments that can be appended to a mail message.

Maximum Attachment Size: This is the maximum size attachment that can be appended to a mail message.

Attachment Description Field: A blank field associated with the attachment name can be used to describe its contents.

Text Screen Capture: A program facility can take a snapshot of on-screen text, convert it to a file, and attach the text image to a mail message. The recipient can display the attached screen image with the mail message. The screen capture capability is usually provided by a memory-resident utility.

Graphic Screen Capture: Given a CGA, EGA, VGA, or Hercules monitor, a program facility can take a snapshot of a displayed graphic, convert it to a file, and attach it to a mail message. The message recipient can display the attached screen image with the mail message. The screen grabbing capability is usually provided by a memory-resident utility.

Graphics Editor: A graphics editor included with the program can be used to create all-points-addressable graphics for attachment to messages. CGA, EGA, and Hercules Graphics adapters are typically supported.

Revise Unsent Messages: A message can be partially created, saved in the Inbox or in a folder, and retrieved for editing and completion.

Revise Sent Messages: A sender can retrieve a message from a folder or the Inbox and revise it for resending or storage.

DOS Directory Lookup: The sender can display a file directory and select DOS files for attachment. Files highlighted or flagged in the directory are automatically inserted into the message as attachments.

Indicate Attachment Size: The message editor can display the size of attachment files (bytes or kilobytes) added to a message.

Electronic Forms: The mail system supports the creation of forms characterized by protected and data entry fields. The user fills in the appropriate blanks and sends the form as a mail message.

Create/Edit Multiple Messages: Multiple mail messages can be edited or viewed simultaneously. The screens are usually windowed and active one at a time.

Message Receiving

Flag Unread Messages in Inbox: A user's Inbox indicates which messages have not been read. Unread messages are highlighted or displayed with a status indicator.

Flag Messages with Attachments: A user's Inbox indicates which messages have attachments by displaying a status field indicator next to the message title.

Move Message to Folder: A message may be moved from the Inbox to a folder (the message no longer exists in the Inbox).

Copy Message to Folder: A message may be copied from the Inbox to a folder (the message still exist in the Inbox).

Save Selected Attachments to Folder: The user can decide which of multiple attachment files should be saved to a folder.

Archive Messages: An Archive function copies specified messages from folders or from the mailbox to an archive file on a local disk drive. The Archive function may or may not compress archived messages, and archive files are usually encrypted. Archive files can be restored to folders within the mail system.

Save Message to Disk: The user can save a single message to disk.

Append Message to DOS File: A mail message can be written to the end of an existing DOS file.

Print Message: A mail message can be directed to a printer port.

Read Next Option: After reading a message, the user can move to the next message in the Inbox with a key-stroke.

Read Previous Option: After reading a message, the user can return to the previous message in the Inbox with a keystroke.

View Attachment: The user can view message attachments before saving them to disk.

Save Attachments to Disk: Message attachments can be saved to DOS files.

Print Attachments: Message attachments can be individually directed to a printer port.

Reply to Sender: A message recipient can reply to the sender by entering the mail editor and entering the sender's name in the To: field.

Automatic Reply to Sender/Recipients: A Reply option can automatically enter the sender's name in the reply To: field and other recipients' names in the reply CC: field.

Reply to Other Recipients: The user can issue a reply to selected recipients or to users other than the original recipients by entering their names in the CC: field.

Reply Attachments: A DOS file can be attached or appended to a message reply.

Copy Original Message to Reply: The entire text of the original message can be copied into the reply's text area. The respondent can correct or emend the original text without retyping or importing the original.

Forward Message: After reading a message, the recipient has the option of sending it to another user.

Forward to Multiple Recipients: After reading a message, the recipient has the option of sending it to multiple users.

Forward with Annotations: After reading a message, the recipient may add text and send the annotated message to other users. Comments are usually added outside the original message area.

Forward with Added Attachments: After reading a message, the recipient can add attachments and forward the message with all original and new attachments to other users.

Forward with Changes: After reading a message, the recipient can alter the text and forward the altered message to other users. The mail system should indicate to forwarding recipients that the original has been altered.

Forward Without Attachments: A message can be forwarded to other users without its original attachments.

Security

Post Office Password: Administrative tasks are performed within an administrator program; the administrator must enter a Post Office password to access Post Office data.

Administrator Password: An administrator password is required to access Post Office data when performing administrative functions.

User Password: Each user must log in to the mail system using a password. User passwords prevent unauthorized access to mail resources and user mailboxes.

Password Encryption: All user and administrator passwords are encrypted in the Post Office to prevent their use by intruders.

Message Encryption: All mail messages stored in the Post Office are encrypted to render them unreadable to intruders. Encryption is performed at the workstation prior to transmission. The Post Office file server manages data storage only.

Encryption during Transmission: Messages are encrypted during transmission between the workstation and server or between remote networks via internet-work routing.

Password Encryption Key: A sender and recipient can exchange a password encryption key for decoding messages.

Encrypt Attachments: Message attachments are encrypted at the workstation prior to transmission.

Message Search and Display

Search from Starting Date: The user can search the Inbox or a specific folder and display all mail received before a selected date.

Search within Date Range: The user can search the Inbox or a specific folder and display all mail received between selected start and end dates.

Search by Sender: The user can search the Inbox or a specific folder and display mail received from a specific sender.

Search by Subject: The user can search the Inbox or a specific folder and display messages pertaining to a selected subject.

Search for Keyword: The user can search the Inbox or a specific folder and display messages containing a specified keyword.

Display Unread Mail Only: Inbox messages can be sorted to display only unread messages.

Display Mail with Attachments Only: Inbox messages can be sorted to display only messages with attachments.

Display CC: or BCC: Mail Only: Inbox messages can be sorted to display only messages having CC: or BCC: recipients.

Display Priority Mail Only: Inbox messages can be sorted to display only urgent messages.

Display Encrypted Mail Only: Inbox messages can be sorted to display only messages requiring an encryption password (supported in very few mail systems).

Folder Creation and Management

Create Folders: Mail system users can create disk-based named storage areas for storing mail. A folder typically stores mail related on the basis of some naming convention. Folders often reside on the file server, but some systems allow local disk storage.

Delete Folders: A mail system user can delete a folder name; deletion revokes the folder's storage space and deletes its contents.

Rename Folders: Mail system users can change a folder's name, and messages within the folder are not altered.

Create Subfolders: Mail system users can create folders within folders (analogous to DOS subdirectories) for hierarchical mail classification and storage.

Move Messages between Folders: A message can be moved from one folder to another (removing the message from the source folder).

Copy Messages between Folders: A message can be copied from one folder to another; the message exists in both folders.

Move/Copy Multiple Messages between Folders: The user can highlight a number of messages to be moved or copied to another folder in a single operation.

Delete Multiple Messages: The user can highlight messages within a folder for deletion in a single operation.

Maximum Folders per User: This is the maximum number of folders a user can create.

Maximum Messages per Folder: This is the maximum number of messages a user can store in any one folder.

Wastebasket Folder: Deleted messages are temporarily stored in a Wastebasket folder that remains active for the duration of the mail session. Before exiting the session, the user is given the option of retrieving messages before they are irretrievably deleted.

Log Folder for Outgoing Mail: All outgoing mail messages can be automatically stored in a log folder (usually a selectable option).

Log Folder for Incoming Mail: All received mail can be automatically stored in a log folder separate from the Inbox (usually a selectable option).

Sort Folder by Date: Messages in a folder are sorted by date, usually with current messages on top.

Sort Folder by Sender: Messages in a folder can be sorted by sender name (usually alphabetically).

Sort Folder by Priority: Messages in a folder can be sorted by urgency or priority from most to least urgent.

Sort Folder by Subject: Messages in a folder can be sorted by subject (usually alphabetically).

Archive Folder to DOS File: All messages in a folder can be combined into a large archive file stored on a local drive (as a personal backup of folders maintained in the Post Office storage area).

Retrieve Archive Folder: A folder archived to a local drive can be restored to the mail system, and messages within the folder are accessible to the user.

Copy Message to DOS File: Individual messages stored in a folder can be written to a DOS file.

Folder Comment Field: The program assigns each folder a field where the user can describe its contents.

Cut and Paste from Folder: The user can extract part of a message stored in a folder and paste it into another message.

Mail Notification

Background Notification by Polling: The user can enable a TSR function that periodically polls the Post Office for new mail. Notification may take the form of a beep and/or small message box.

Background Notification via NETBIOS: The user can enable a TSR function that can service NETBIOS interrupts from a sending station indicating that mail has been sent to the user.

DOS-Level Check for Mail: A query function invoked from the DOS command line checks the Post Office for new mail; the user need not enter the mail system.

Miscellaneous Features

Nested Distribution Lists: A distribution list created by the user or administrator can contain references to other distribution lists. Address duplicates are discarded.

Bulletin Board: The mail system provides a common area where messages are accessible to all mail users. A bulletin board is usually defined by the administrator and organized by category or heading. Users can submit, but not delete, bulletin board messages; administrators maintain the bulletin board.

Context-Sensitive Help: The mail system offers help information specific to the current area of operation.

Quick Reference Guide: The mail package comes with a reference manual covering the system's command syntax.

Tutorial Guide: The mail package includes a printed tutorial on mail functions.

API Libraries: The mail package includes application interface features that let third-party developers access aspects of the mail system. APIs typically relate to communication functions.

Demo Disk: The mail package comes with a program that demonstrates mail system functions.

DOS Shell: The user can exit to a DOS shell from within the mail package, execute DOS operations, and return to the mail system without losing the mail system screen.

Import/Export Post Office Files: The program utility imports and/or exports mail files resident in the Post Office database (e.g., for batch updates or extractions of mail files).

Background Notification via the NetWare Send Command: Novell NetWare's Send facility notifies recipients of incoming messages. The originator invokes the Send command for each recipient.

Software Prices

	Price (\$)
LAN E-Mail Programs	
AT&T PMX/StarMail	
StarMail for Novell (unlimited users)	1,295
StarMail for AT&T 6386 UNIX Server	1,295
cc:Mail	
Platform: DOS, Macintosh, Windows, or OS/2 (plus user packs)	195
8 Users	395
25 Users	895
Consumers Software Network Courier	
Post Office (plus workstation packs)	695
5 Users	395
20 Users	1,349
100 Users	5,500
500 Users	22,500
Sitka InBox Plus	
100 Users	1,995
Transend CompletE-Mail/MHS	
5 Users	295