Oracle8™ Enterprise Edition Release 8.0

Oracle8 Enterprise Edition provides efficient, reliable, secure data management for applications ranging from high volume on-line transaction systems to query-intensive data warehouse applications. Oracle8's object-relational design provides new capabilities for managing data complexity. Oracle8 also provides the tools to manage the systems, the flexibility to distribute the data to users effectively and efficiently, and the scalability to achieve optimal performance from all available computing resources.

IT CHALLENGES

In today's demanding business world, operational systems are relied upon more than ever. User communities are growing in number and they are demanding higher levels of application performance, reliability, and availability. Business environments are also more dynamic than ever before, with continually decreasing time lags for their applications to respond to changing business requirements.

Also, strategic decisions must be made quickly, so businesses can react and adapt to reorganizations and process changes, regulation revisions, and competitors' directions. These decisions need to be based on accurate data and thorough analysis. The ability to analyze and exploit operational data in a data warehouse becomes a key competitive weapon.

To meet these challenges, IT organizations need an enterprise software strategy for managing any data, in any application, at any scale. Oracle8 is the only open systems solution that meets the demands of high-transaction on-line transaction processing (OLTP) systems, query processing in large-scale data warehouses, and manageability requirements of businesses that are distributing data throughout the corporate enterprise and the Internet.

Oracle8 also makes a major leap in data management technology with the introduction of an object-relational paradigm for complex applications. This improved way of defining data structures allows developers to directly define their business objects, such as purchase orders, inventory items, and data warehouse information, within Oracle8. This allows developers of mainstream commercial applications to better manage their business objects.

Oracle8 leverages your investment in development, deployment, and maintenance of applications by providing a single product to deliver the maximum benefits of open, multimedia systems across your enterprise, while minimizing the risks, complexity, and costs of moving to both centralized and distributed application environments.

NCA FOR NETWORK-CENTRIC COMPUTING

Oracle's open, standards-based Network Computing Architecture™ (NCA) allows IT organizations to spend less time struggling with interoperability and more time focusing on deploying solutions. Oracle8, a major component of NCA, is designed to meet the demands of network-centric computing and object-oriented development methods. NCA provides maximum extensibility and is based on open industry-standards such as CORBA and IIOP. Whether for traditional enterprise applications or electronic commerce on the Web, Oracle8 and NCA provide the power, robustness, network integration, and flexibility to support the most demanding applications.

ENTERPRISE-CLASS TRANSACTION PROCESSING

Oracle8's scalable, reliable architecture delivers unmatched scalability, availability, and performance needed for mission-critical OLTP systems. Integrated, dynamic facilities ensure that Oracle8 and the Oracle8 Parallel Server make efficient use of all system resources on hardware ranging from uniprocessors to symmetric multiprocessors (SMP), to clusters, to massively parallel processors (MPP).

Superior Scalability for Transaction Processing

The Oracle8 architecture provides OLTP applications with scalability to support large numbers of users and high-volume transaction workloads. Oracle8 provides exceptional scalability on SMP, clustered, and MPP machines.

OLTP applications take advantage of Oracle8's parallel architecture by distributing tasks across multiple processors or machines, such as in a clustered environment, which improves individual transaction response times and overall system throughput. An automatic, dynamic self-tuning capability balances processing workload evenly across allocated hardware and operating system resources.

Additional processors and/or nodes can be added to expand your configuration incrementally as both your organization and data volumes grow—with minimal disruption to your existing environment—resulting in dramatic performance improvements and breakthrough price/performance.

High Availability for Transaction Processing

Mission-critical, OLTP systems require high availability so your business can continue to operate when a hardware failure occurs. The Oracle8 Parallel Server uniquely extends the reliability of open systems applications by transparently harnessing the power of clustered computers in a single logical processing "complex" that can tolerate individual machine and/or node failures without loss of data availability. The Oracle8 Parallel Server also supports hybrid configurations, combining elements of clustered and MPP architectures.

Should a node in the parallel server fail, transparent application failover migrates the users' connections and automatically re-establishes their sessions on another node. The users' applications continue to run, and the users themselves may be unaware of the failure. This provides continuous availability in the event of scheduled and unscheduled outages.

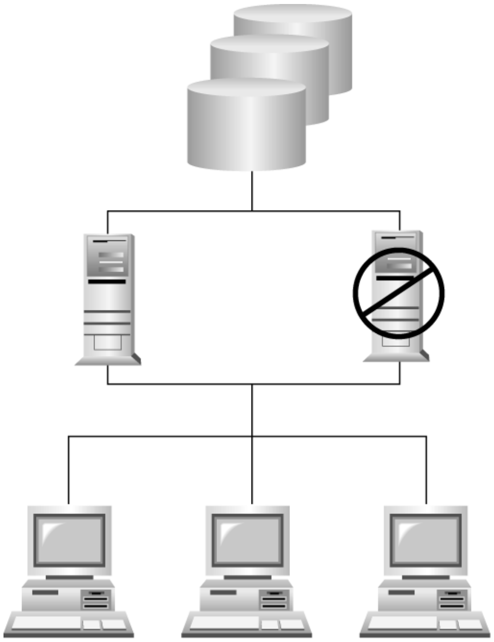

Figure 1: Oracle8 with Parallel Server provides superior scalability to meet increasing transaction processing requirements while assuring high availability in case of hardware failure or site disaster.

High Performance for Transaction Processing

For high-performance transaction processing, the Oracle8 multithreaded, multiserver architecture coordinates thousands of simultaneous user requests. Individual requests are queued and serviced by a minimum of server processes.

Sophisticated caching of database blocks, SQL execution plans, and executable stored procedures takes maximum advantage of available server memory. Available system resources can be precisely allocated with a high degree of control, optimizing performance to the capabilities of the system and to the system workload on a dynamic basis. I/O operations are similarly specialized, utilizing different techniques such as asynchronous and multiblock reads and writes to improve response times and overall system throughput for all users and all requests. Stored procedures and database triggers are stored in compiled form, allowing them to be executed directly without recompilation or parsing, resulting in optimum run-time performance.

Large User Population Support

Oracle8 and Net8™ efficiently utilize operating system and networking resources, allowing tens of thousands of concurrent users to connect over multiple network protocols. Connection pooling temporarily drops the physical connection for idle users (and transparently re-establishes the connection when needed), increasing the number of users that can be supported. The Connection Manager multiplexes several user sessions over one network connection, reducing resource requirements especially for multitier application architectures.

Transactional Data Access

Oracle8 offers a number of different access paths for locating transactional data quickly and efficiently including fast, full-table scans, B-tree single column and concatenated-index scans, clustered (pre-joined) tables, hash clusters (utilizing a single column or an application-specific SQL hash function), and unique row identifiers. The Oracle8 cost-based optimizer dynamically selects the fastest available access path, and satisfies query requests directly from indexes, whenever possible.

High-Performance Concurrency Control

With other database systems, applications can become “contention bound” with performance limited by transactions that lock data at the page level or escalate lower-level locks, regardless of available CPU power or I/O bandwidth. Oracle8 employs full, unrestricted row-level locking for data and indexes, and never escalates locks to ensure maximum concurrent access to data. Oracle8’s high-performance, scalable sequence number generator eliminates application contention to obtain unique numeric key values, a common requirement in transaction processing applications. Reverse-key indexes reverse the bytes in index entries, spreading the inserts of consecutive keys among different blocks to eliminate insertion hot spots.

Reliable Query Results

Oracle8 effectively supports mixed workload environments characterized by simultaneous query and update activity. Many databases force users to choose between good performance and guaranteed data consistency—one coming at the expense of the other. Oracle8’s non-blocking, multiversion read consistency always provides users with consistent query results, while never imposing a performance penalty on concurrent update activity.

Advanced Queuing

Oracle8 Advanced Queuing provides direct support in the database for high performance queuing operations. This capability provides asynchrony and eliminates the dependency on external systems for applications requiring high scalability. The enque and dequeue operations may also be used to shift processing from within a transaction to a background process, improving the transaction response time.

TP Monitor Support

Dynamic XA support provides high performance for multitier applications with industry-standard XA-compliant Transaction Processing (TP) monitors.

ENTERPRISE-CLASS DATA WAREHOUSING

Data warehouse applications require different processing techniques than OLTP applications, due to the complex, ad hoc queries running against large amounts of data. To address these special requirements, Oracle8 offers a rich variety of query processing techniques, sophisticated query optimization to choose the most efficient data access path, and a scalable architecture that takes full advantage of all parallel hardware configurations.

Rich Query Processing Techniques

Successful data warehouse applications rely on superior performance when accessing the enormous amounts of stored data. Oracle8 provides a rich variety of integrated indexing schemes and join methods to deliver answers quickly to data warehouse users.

Bitmapped indexes deliver dramatic performance benefits to data warehouse applications. Bitmapped indexes are fully integrated into the Oracle8 Server, and coexist with and complement other available indexing schemes including standard B-tree indexes, clustered tables, and hash clusters. While B-tree indexes may be the most efficient index to retrieve data using a unique identifier, bitmapped indexes are most efficient when retrieving data based on much wider criteria, such as "How many red cars were sold last month?" In data warehouse applications, end-users often query data based on these wider criteria. Oracle8 enables efficient storage of bitmap indexes through the use of advanced data compression technology. Parallel index creation and support for automatic index maintenance with update operations facilitate easy maintenance.

Hash joins deliver higher performance over other join methods in many complex queries, especially for those queries where existing indexes cannot be leveraged in join processing, a common occurrence in ad hoc query environments. Oracle8's hash joins eliminate the need to perform sorts by using an in-memory hash table constructed at run-time. They are also ideally suited for scalable, parallel execution.

In addition to enabling administration of large databases, partitioning also improves query performance. Partition elimination reduces the amount of data which must be considered for queries, and thus, improves performance.

Sophisticated SQL Optimizer

Oracle8's numerous, powerful query processing techniques are completely transparent to the end user. The Oracle8 cost-based optimizer dynamically determines the most efficient access paths and join methods for every query. The Oracle8 optimizer incorporates powerful query transformation technology that automatically 'rewrites' queries generated by end-user tools for efficient query execution.

To choose the most efficient query execution strategy, the Oracle8 cost-based optimizer takes into account statistics, such as the size of each table and the selectivity of each query condition. Histograms provide the cost-based optimizer with more detailed statistics for skewed, non-uniform data distribution.

The cost-based optimizer considers many other constraints when choosing an execution strategy. The user or application can specify whether it is more desirable to return the first row of a query quickly, or to complete the entire query before returning the data. The cost-based optimizer is also "parallel aware," which means it considers the availability of parallel resources when choosing the most efficient execution strategy.

Oracle8 includes high-performance star-join processing using either B-tree indexes or bitmap indexes. A patent-pending approach to star joins incorporates intelligent query transformations and an efficient algorithm for joining multiple tables in a single step, without the need to generate a Cartesian product of the dimension tables. This method makes innovative use of bitmap indexes and avoids the need for one or more multicolumn B-tree indexes on the fact table. It provides superior scalability to handle large or unconstrained dimensions and large number of dimension tables, and efficiently deals with sparse fact tables.

Single-column bitmap indexes are dynamically combined based on query criteria, eliminating the need to create and maintain multiple concatenated indexes. Further, the method is flexible enough to deal with complex schemas involving multiple fact tables and advanced models such as snowflake schemas, providing additional choices in modeling without any trade-off in access performance.

Oracle8 query processing not only includes a comprehensive set of specialized techniques in all areas—optimization, access and join methods, and query execution—but they are all seamlessly integrated and work together to deliver the full power of the query processing engine.

VERY LARGE DATABASE SUPPORT

Oracle8 addresses the largest and most demanding OLTP and data warehousing applications with databases into the terabytes and beyond.

Partitioned Tables and Indexes

Partitioned tables and indexes divide large tables and indexes into pieces which can be separately managed, instead of managing each table as one large, monolithic object. Partitioning is a ‘divide and conquer’ technique that provides scalable performance with a large amount of data. Partitioning decreases the time required to perform administrative operations by applying the operations to smaller units of storage, improves performance through increased parallelism, and improves availability by containing the impact of failures.

Administrators may specify storage attributes for each partition and the placement of the partition within the host file system, increasing the granularity of control for very large databases. Partitions may be individually taken off-line or brought on-line, backed up, recovered, exported and imported, and loaded; thereby limiting the time required for management operations. An individual index partition can be built for one table partition, bounding the time required for index maintenance operations. Various local and global index strategies are provided. Partition operations may be performed in parallel. Partitions increase availability by containing media and application failures—applications not requiring data from a down or off-line partition continue to run without impact.

Partitioning is transparent to applications and standard DML statements run against partitioned tables. The Oracle8 optimizer is partition-aware, and partitions which do not contain any data for a query are eliminated from the search.

Scalable Parallel SQL Architecture

Large data warehouse and transactional servers must also provide scalable performance for processing large amounts of data. Oracle8’s superior, integrated parallel SQL architecture provides excellent scalability on SMP, MPP, and hybrid hardware platforms.

Oracle8’s parallel SQL architecture increases the performance of database queries and updates by dynamically subdividing these operations into distinct tasks, and distributing the workload across all multiple processors. Oracle8 parallelizes more operations than any other database product. A partial list includes:

- queries
- insert, update, delete
- sorts (ORDER BY)
- aggregation (GROUP BY)
- table creation (CREATE TABLE…AS SELECT)
- tablespace creation
- data loading
- index creation
- recovery

Oracle’s parallel SQL architecture performs on all varieties of parallel hardware systems. Oracle’s parallel SQL is adaptive; it intelligently chooses the most appropriate parallel processing method for its hardware environment.

Oracle8 combines intelligent use of data partitioning in the form of partition elimination and parallelism across partitions, with Oracle's unique, dynamic, intra-partition parallel query execution. This architecture transparently delivers scalable query performance—leveraging, but not constrained by data partitioning. Since Oracle8 parallel execution is *not* completely based on data partitioning, administrators are not forced to choose between parallel execution performance and manageability. All available processing resources are utilized, even in the presence of real-world data skew.

Oracle8 performs bulk insert, update, and delete operations in parallel. These parallel data manipulation operations deliver high performance scalability and efficient utilization of all hardware resources and enable the completion of data management tasks within ever-shrinking batch windows. Parallel insert, update, and delete capabilities are very useful for bulk operations such as the creation of summary tables, purging historical data, and batch updates in support of “what-if” analysis or re-statement operations.

On an SMP platform, Oracle8 symmetrically parallelizes a database operation across all available processors. “Data shipping” is employed for internode communication, the most efficient method for these platforms.

On an MPP or cluster platform, Oracle8 transparently exploits data locality. Each processor is assigned to work on its local data partitions. Moreover, Oracle8 always utilizes all available processors, independent of how the data is partitioned. If one processor is overloaded, Oracle8 dynamically assigns remote processors to the task. Oracle8 makes extensive use of “function shipping” on these platforms to minimize data transfer across the interconnect.

Oracle8’s adaptive parallel architecture combines the best elements of “shared disk” and “shared nothing” approaches and provides excellent load balancing and dynamic load distribution. Oracle8 is a proven solution that enables customers to realize the maximum potential of parallel performance.

Extended Backup/Recovery Subsystem

Oracle8 server-managed backup and recovery provides a high level of backup and recovery functionality from within the Oracle Server. Oracle8 maintains detailed information on when backups are performed, exactly which parts of the database are backed up, and where the files are stored. Should a recovery be necessary, Oracle8 analyzes the state of the database and determines the operations necessary to repair the database. Oracle8 then automatically performs those operations, greatly simplifying the recovery for the administrator and reducing the possibility of human error. A simple graphical user interface (GUI) interface within the Oracle® Enterprise Manager controls backup and recovery. An application programming interface (API) is also available for third parties who may wish to provide an alternative interface. A media management layer interfaces with popular third-party tape management products.

Multilevel incremental backups greatly reduce the size of the backups, since only the changed blocks are backed up. This also reduces the time required to backup a datafile. Tablespace point-in-time recovery allows one or more tablespaces to be recovered to an earlier time, while the remainder of the database is up and running. This allows many types of user errors to be easily corrected. For example, if a user runs a batch job that incorrectly updates many records in a table, the table can be restored to a time previous to the batch job. Also, if a table is accidentally dropped or truncated, it can be restored to a time before this operation.

HIGH-AVAILABILITY SUPPORT

High-Availability Operations

Oracle8's on-line backup facility allows administrators to perform backup activities while the database is running and without interrupting transaction processing—even during periods of heavy OLTP usage. Should a device containing user data fail, lost datafiles can be recovered on another device, while Oracle8 continues processing requests directed at the remainder of the database. Oracle8 can use multiple processes to recover the database in parallel, speeding the on-line recovery.

Oracle8 incorporates a deferred transaction recovery mechanism to facilitate faster database startup following a system crash. Rollback operations related to uncommitted transactions are performed in parallel after startup, providing earlier system availability and improved performance.

Oracle8 also supports read-only tablespaces, saving time by eliminating backup and recovery of static data. Oracle8 includes a number of features that promote high availability in the presence of media failures or errors. Mirrored, multisegment logs ensure that critical log data remains available if a log device fails. Optional check-sum computation, verification for database blocks, and redo log blocks facilitate earlier detection and improve diagnosis of media errors.

High-Availability Applications

In addition to providing scalable performance for OLTP and data warehouse applications, the Oracle8 Server ensures high availability of your applications by providing a variety of implementation options for disaster recovery, allowing you to choose which is the most appropriate for your application. The Oracle® Parallel Server, the Oracle8 standby database feature, and Oracle8 Advanced Replication are methods for providing high availability for applications. Each is an integrated, supported function of the Oracle8 Server.

The Oracle8 Parallel Server ensures that your data is accessible in the event of a node failure when operating in a clustered computer environment. If any node in the system should fail, affected users can simply log in to another operating node and resume processing. Transactions committed on the failed node, but not written back to the database files, are automatically recovered by one of the remaining nodes, while transactions that were "in-flight" (not committed) are automatically rolled back.

The Oracle8 standby database feature provides a reliable and supported mechanism for implementing a standby database system to facilitate rapid disaster recovery. The scheme uses a secondary system on duplicate hardware, maintained in a constant state of media recovery through application of log files archived at the primary site. In the event of a primary system failure, the standby database can be activated with minimal recovery, providing immediate system availability.

Oracle8 Advanced Replication can also be used to provide high data availability. You can replicate data from your primary system to one or more alternative sites. Each alternative site is fully accessible and can be used for query access, and in some cases, for updates as well. In the event of a system failure, your application can failover to any of the alternative sites, providing continuous data availability.

OBJECT-RELATIONAL DATABASE

The Oracle8 object-relational paradigm allows developers to directly define their business objects such as purchase orders, inventory items, and data warehouse information within Oracle8. This allows developers of mainstream commercial applications to better manage their business objects. The object application development tool allows you to develop object-oriented applications using CORBA, OLE, and C++ objects with Oracle8 data.

Extensibility and User-Defined Datatypes

Oracle8 allows users to define custom object types. An object type is typically defined to correspond to some business object, such as a purchase order. The object type may contain multiple fields or attributes, and it may be nested within other object types. More complex objects that include a variable number of items are supported through variable length arrays and nested tables. This allows, for example, a purchase order object type, which can accommodate a variable number of line items.

An Oracle8 method is program code associated with an object type. Methods perform typical operations on the object type, for example, to calculate the total of a purchase order. Methods may be written in PL/SQL™ and stored within the database, or written in an external 3GL procedure and called through a safe external callout mechanism. User 3GL code is safely and dynamically linked and loaded, and a user cannot disrupt others using the same database. User-defined mapping and ordering methods provide database extensibility and allow users to define how comparisons and ordering should be performed on object types.

Oracle development products, including Oracle Call Interface™ and Pro*C™, support objects. Client-side caching, complex object retrieval, and navigational access provide client-side processing and minimize the number of network round-trips between the client and the server. Oracle Call Interface is modeled after CLI, adding in support for objects. The object type translator utility generates C header files for use in Pro*C and Oracle Call Interface applications.

Evolutionary, Open, Secure

Oracle8 eases the evolution from relational to object-oriented functionality. The Oracle8 object-relational capabilities are built on the same solid foundation as the relational functionality.

Like relational views, Oracle8 object views provide the appearance of a complex object, including support for object methods, but the data is stored in underlying object—or relational—structures. An object view allows, for example, a purchase order object view to be defined on existing relational purchase order data. This design then allows the existing relational applications, which still read and write rows and columns, to coexist with new object-oriented applications, which read and write purchase order objects. INSTEAD OF-triggers allow users to perform inserts, updates, and deletes against even the most complex object (and relational) views.

Objects are completely integrated throughout Oracle8, in all levels of the server, and are supported in both SQL and PL/SQL. The industrial-strength properties of the Oracle Server are provided with objects, including Oracle8's sophisticated concurrency model, industry-leading performance, scalability, reliability, manageability, and availability.

The ANSI and ISO SQL database committees are currently finalizing the definition for object extensions to SQL. This is informally known as SQL3. Oracle is very active in this standards process, and Oracle8 follows the draft SQL3 standard.

A key product which complements Oracle8 is Oracle's object-oriented application development tool. Other Oracle products including Developer/2000™, Designer/2000™, Oracle Media Objects®, Oracle Power Objects®, and Express Objects™ also support Oracle8 object-relational capabilities.

Java Support

Java code may be executed in the client or in a middle tier, and may access the database through JDBC or JSQL™.

Web Integration

Oracle8 integrates with Oracle Web Application Server, which can be used to fully integrate your existing Oracle8 business applications with Web technology and safely deploy them inside or outside your corporate firewall.

Oracle Web Application Server enables stored procedures to be invoked by clients using Web browsers to generate dynamic Web documents, so that Web pages are no longer limited to displaying information from static files.

ENTERPRISE APPLICATIONS DEVELOPMENT

Oracle8 includes the powerful, flexible, scalable application development features necessary to construct a new generation of sophisticated applications at low cost. Declarative facilities ensure scalable, reliable enforcement of data integrity while minimizing development, maintenance, and administration costs. PL/SQL, an advanced procedural 4GL language tightly integrated with the Oracle8 Server, provides the power to easily express complex business rules as stored, procedural code. The Oracle8 programmatic interfaces enable 3GL programs to access and manipulate Oracle8 data and schemas. Oracle8's productive development features improve application performance, scalability, and security to support hundreds of applications and thousands of users.

Powerful, Flexible SQL Language

Oracle8's 100-percent ANSI/ISO SQL92 entry-level, compliant, SQL implementation ensures a fully-open application development environment. In addition, Oracle8 offers a number of robust SQL extensions that allow complex operations to be expressed in SQL, improving developer productivity by reducing the need for procedural code. Application performance and scalability are enhanced by performing complex data manipulation operations within the Oracle8 SQL engine. INSTEAD OF triggers allow any view to be updated.

Server-Based Business Rules

The Oracle8 Server allows you to enforce constraints, either for database integrity purposes or business-related rules, at the database level, providing you with the greatest amount of security and business rule enforcement. Through the use of declarative integrity constraints, database procedures, and database triggers, Oracle8 enforces your application business rules easily and securely. Deferred constraint checking optionally shifts integrity constraint checking from the end of statement execution to the end of transaction execution. This simplifies the coding of certain operations involving integrity constraints. Also, many kinds of constraints can be enabled without stopping update activity on the table.

With PL/SQL-stored procedures, functions, packages, and database triggers, you can enforce complex business rules at the server level, improving application performance, scalability and security, and reduce development costs. Procedures and functions can accept arguments from calling client applications, and return one or more result values. Packages group together definitions of related procedures, functions, variables, cursors, and other database objects to improve development productivity. Database triggers are executed automatically, either once per row or once per statement, when rows are inserted, updated, or deleted from Oracle8 tables, and can be used to enforce complex integrity rules within the server. Oracle8's robust trigger implementation is modeled closely on the draft ANSI/ISO SQL3 specification.

Productive Application Development

Oracle8 stored procedures and triggers improve application development scalability and productivity by allowing common procedures to be developed once and maintained in a central location, instead of in every application. They improve application performance and scalability by allowing application logic to be invoked with a single call, minimizing usage of slow networks, and by isolating application processing on the server. Oracle8's stored procedure implementation also supports automatic dependency tracking for scalable application development. Oracle8 stored procedures can dynamically define and execute SQL statements, permitting especially powerful and flexible procedures. User-defined functions referenced from SQL statements improve productivity by enhancing the power of SQL, and enhance performance by executing application-specific logic within the server. PL/SQL cursor variables provide a handle to an SQL query, allowing stored procedures to encapsulate statically- or dynamically-defined queries, and return one or more multirow result sets to calling applications.

Oracle8 also supports calls to external 3GL procedures for high-performance, computationally-intensive processing, and to interface Oracle8 applications with external systems. User 3GL code is dynamically linked with Oracle8, and runs in a separate process to prevent any disruption of or risk to the database.

Productive 3GL Programming Interfaces

The Oracle Precompilers provide a high-level, highly productive, embedded SQL interface to the Oracle8 Server that is 100-percent compliant with ANSI/ISO standards. Oracle Call Interface provides a low-level, native procedure/function call interface to the server that allows absolute control over SQL statement processing. SQL*Module® supports the ANSI/ISO SQL module language and promotes developer specialization by separating application development and database programming tasks. Oracle Programming Interfaces can be used individually or in combination with different development styles to obtain the exact features desired for a given application.

Multimedia Data Support

Multimedia data may be stored in Oracle8 databases in character and binary large object (LOB) datatypes. LOBs allow for excellent control over storage because they are separate from the rest of the table's data. LOBs may be stored outside the database, and are very flexible for a wide variety of application requirements—a table may contain several LOB columns; pieces of LOBs may be accessed, updated, and appended, substantially reducing the memory requirements and improving performance; logging and caching of LOBs may be controlled; and LOBs may be replicated.

Comprehensive National Language Support

Oracle8 National Language Support (NLS) ensures that error messages, sort order, date format, and other conventions automatically adapt to the native language. Parameter settings at the Oracle8 and operating system levels determine the behavior of individual conventions. Oracle8 supports deployment of heterogeneous client/server and distributed database configurations by automatically and transparently performing any necessary character-set conversions. The NCHAR datatype allows dual character sets in one database.
This provides high performance and storage predictability for some Asian language multibyte character set databases.

Separate national calendars, including Japanese Imperial, ROC Official, Thai Buddha, Persian, and Arabic Hijrah, are supported. Arabic and Hebrew display character set support is also available.
Oracle8 provides extensive 8- and 16-bit character set support, including the Unicode variable-width UTF-2 encoding.

ENTERPRISE-WIDE DISTRIBUTED SOLUTIONS

The benefits of open, relational technology cannot be achieved without transparent integration of new and existing systems. Oracle8 provides flexible, integrated, manageable distributed database facilities that make the integration of enterprise data practical. Data can be replicated for direct and highly available local access. Data can be accessed remotely using both SQL and procedure calls in a fully transparent manner as if the data were local. Also, data can be in both Oracle and non-Oracle severs.

Data Replication

Oracle8 delivers the industry's most comprehensive replication capabilities—from basic to advanced.
For information dissemination throughout your enterprise, Oracle8 basic replication supports a simple primary site model of replication where one replica is updatable and all others are read-only. For more sophisticated distributed operational systems, including fail-over configurations, and mass deployment applications, such as sales force automation, Oracle8 advanced replication supports bi-directional replication and sophisticated conflict detection / resolution and management tool capabilities that you require.

Basic replication is easily configured using SQL and Oracle® Enterprise Manager. One replica, the master, is updatable and all other replicas, or snapshots, are read-only. Snapshots can contain all of the data in a master table or just a subset of rows or columns. Subsets can be defined using subqueries that reference data in other tables. Incremental row changes are propagated on demand or at time-based intervals using a fast refresh mechanism. Groups of related snapshots are refreshed to the same point in time to ensure transactional integrity and preserve referential parent/child relationships.

Advanced replication extends basic replication allowing snapshots to be updatable and supporting replication between master replicas. Snapshots (both updatable and read-only) are designed to support large numbers of remote sites which may typically operate in a disconnected manner, e.g., requiring only occasional dial-up connections from fixed or mobile sites. The subquery subsetting capability makes it practical for each site to maintain its own unique subset of the database. For example, each salesperson's PC can hold information about his or her customers and no others.

Replication between masters is designed to support high volume replication flows over high speed networks at "near real-time" or scheduled intervals. Replicated transactions are propagated in parallel from each source to target master site. Dependencies between transactions are automatically detected and tracked by the Oracle8 Server to ensure proper ordering when necessary.

Advanced replication environments are configured and maintained using Oracle® Replication Manager, a specialized component of Oracle Enterprise Manager. Wizards walk you through the configuration and deployment process. Groups of table replicas, and their supporting objects such as views, triggers, packages, and indexes, are automatically deployed to new sites through simple drag and drop operations. Conflict detection/ resolution rules, both predefined and user defined, are also selected and deployed automatically.

Advanced replication also supports specialized modes of replication including procedural replication to improve performance of large, batch-oriented operations against replicated data, and synchronous replication when absolute consistency between replicas is required.

Both basic and advanced replication are fully integrated into Oracle8. Updates to replicas are captured using internal triggers that run within the server executable for maximum performance. No external servers are required which can complicate management procedures such as recovery.

Transparent Distributed Query

With Oracle8, a single SQL statement can query data from multiple databases and even perform complex joins of data physically stored in different servers. Distributed optimization techniques select efficient execution plans and minimize data transmission over slow networks. Location transparency allows applications to be developed without knowledge of the location of data, and ensures that applications never need recoding if data moves from one node to another. Network transparency leverages existing networks and protocols to efficiently transmit data between servers and return results to clients. Distributed query functionality is tightly integrated into the core Oracle8 architecture, allowing sites to operate autonomously and preserve compatibility with all server functionality.

Transparent Distributed Transactions

Oracle8 employs a robust, transparent two-phase commit mechanism to ensure the integrity of distributed update transactions. Updates may be performed by remote or distributed SQL statements, or through remote procedure calls (RPCs) to distributed Oracle8 servers. Commit transparency ensures that all sites automatically commit or roll-back in response to the single standard SQL COMMIT statement—no complex procedural or 3GL coding is required. Oracle8 automatically detects failure conditions, and if necessary, automatically resolves them without manual intervention.

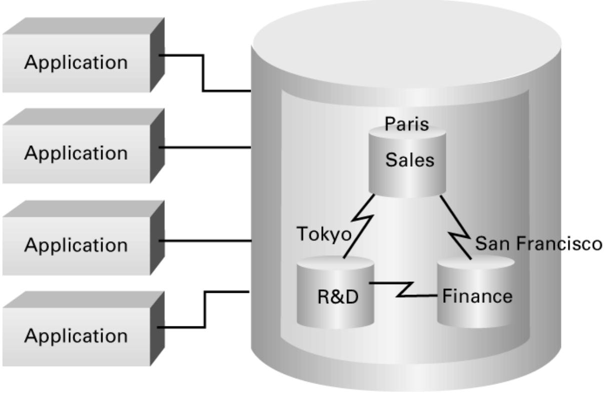

Figure 2: SQL statements are performed on the local or remote database. A robust transparent two-phase commit mechanism ensures the integrity of distributed update transactions.

Integrating Non-Oracle Systems

Oracle8's open architecture integrates Oracle8 and non-Oracle data sources and the industry's most comprehensive collection of tools, applications, and third-party software products into an industry-standard environment. Oracle Open Gateways™ product family adds data source transparency to Oracle8's transparent distributed query and distributed transaction facilities. Oracle Transparent Gateway® provides transparent SQL access to non-Oracle data sources, while Oracle Procedural Gateway® provides transparent procedural (RPC) access to virtually any non-Oracle system. Distributed queries and distributed transactions can include one or more non-Oracle systems, coordinated by Oracle8. Oracle8 also provides an XA-compliant interface that allows external transaction managers, such as TP monitors, to coordinate distributed transactions that span Oracle and non-Oracle systems.

ENTERPRISE DATA SECURITY

Oracle8, together with Net8, provide comprehensive, flexible, and reliable facilities to ensure proper user authentication, guarantee the privacy and integrity of data, manage the assignment of database privileges, and monitor database operations across the enterprise, including today's vulnerable intranet and Internet environments.

User Authentication

Oracle8 normally performs user authentication internally by verifying the password provided at login time. Optionally, authentication may be performed externally by the operating system or a security package; or with the Oracle8 Advanced Networking Option™ by an external authentication service ,such as a network operating system, network security service (e.g., Kerberos, SESAME, DCE); or authentication device (e.g., smart cards). This allows you to centrally manage Oracle security for the entire database or network, reducing administration costs considerably by specifying security policies once and enforcing them network-wide. Oracle8 also provides data signature services callable from Oracle Call Interface and PL/SQL

Database Privileges

Database privileges authorize users to perform certain SQL operations, such as insert, update, or delete, on selected database objects. Oracle8's fine-grain privileges allow you to precisely enforce database security policies, ensuring users have only the privileges they require.

Hierarchical Security Roles

Roles are used to group together privileges on tables and other database objects and granting them to individual users or groups of users as a whole. In this way, security administrators can authorize users to run entire applications with a single GRANT statement, thus significantly reducing the burden and cost of security management. Unlike other RDBMS' that have a single, all-powerful DBA, Oracle8 roles allow organizations to have multiple DBAs and precisely control the special privileges each DBA is given. Oracle8 users and roles can be defined within a single database, or globally across several related databases with centralized management. Oracle8's specification of roles was accepted by the ISO and ANSI standard SQL committees as the basis for future SQL security standards. When combined with the Advanced Networking Option, privileged management can be externalized to an enterprise security server, such as the DCE Security Service or SESAME.

Enforcement of Password Policies

Oracle8 can enforce your security policy. The security policy is encoded in a stored procedure and checks may be performed for example, for minimum length and complexity, or against a history of previously used passwords. Passwords may be authorized for a set time period, after which they would have to be changed.

Auditing

Oracle8 provides integrated, flexible, and reliable auditing capabilities that ensure all database operations of interest are recorded at the appropriate level of granularity. The AUDIT command declaratively enables the auditing of successful and/or unsuccessful user actions during a session, and user attempts to access database objects. Audit trail data are securely recorded in the Oracle8 data dictionary and/or operating system files. Procedural auditing through stored PL/SQL procedures and database triggers can be used to perform application-specific or context-sensitive auditing.

Network Security

Oracle8 always encrypts password information at sign-on time as it travels over your client/server network. With the Advanced Networking Option, full client/server, server/server, and server/gateway datastreams can be encrypted, guaranteeing the privacy and integrity of network communications. Attempts to modify, intercept, or replay messages terminate the current operation and are logged. The Oracle8 Advanced Networking Option and Net8 work over all popular network protocols, and network security is completely transparent to all existing applications. Oracle8 also provides support for privileged database links, which make it unnecessary to embed a password in a database link.

ENTERPRISE SYSTEMS MANAGEMENT

Oracle Enterprise Manager is an open and comprehensive family of systems management products that contain open interfaces for third-party and customer applications, including integration with all the major network management platforms. The product family consists of next-generation systems management tools designed to efficiently manage the complete Oracle environment, including systems, databases, networks, and applications, reducing the number of hours administrators need to spend managing complex production environments.

Oracle Enterprise Manager's client/server architecture, which is both scalable and lightweight, consists of a centralized console, common services, and intelligent agents running on the managed nodes. The console provides a central point of control for the Oracle environment, with common services that provide a job scheduling system and an event management system. The job scheduling system allows administrators to schedule repetitive jobs on remote sites, such as backing up the database or running reports, making a "lights out" systems management operation possible. The event management system monitors specific events at the central or remote sites and alerts the administrator, or takes an automatic corrective action, when a problem is detected.

Integrated database administrator applications running on top of the console include tools for handling user and group administration, storage management, backup/recovery, import/export, and data loading. Also included is Oracle Software Manager which distributes and installs both Oracle and non-Oracle software, and performs license and asset management.

Performance Management and Tuning

Oracle provides an advanced package of tools for performance tuning and diagnostics known as the Oracle® Enterprise Manager Performance Pack. The Enterprise Manager Performance Pack consists of products that help the administrator with performance management, tuning, and diagnostics. Oracle® TRACE and Oracle Expert™ enable application performance tracing and expert database tuning. Other functionality includes a graphical performance monitor, drill-down session analysis, object views of tablespaces, and advanced event management.

Open Server Monitoring

So it can easily integrate and function within your existing system management framework, Oracle8 includes full support for SNMP, a de facto standard that supports the monitoring and administration of components in a distributed computing environment. SNMP Management Information Bases (MIBs) define a series of objects (variables) which can be polled to monitor the operation and performance of a managed component. Oracle8 implements the IETF standard public RDBMS MIB, and an Oracle8-specific private database MIB. Oracle8 SNMP support easily integrates with third-party SNMP-based network and system management environments.

INFORMATION MANAGEMENT WITHOUT LIMITS

Oracle8 provides a single database management system that can satisfy all of your data requirements now and in the future. Oracle8 ConText® Cartridge enables Oracle8 to manage text with the same security, scalability, integrity, and intelligence as it manages structured data. Oracle8 Spatial Cartridge allows you to efficiently store, access, and manipulate your spatial data in the same way as your structured data. And Oracle Video Cartridge stores, manages, and delivers high-resolution, full-screen video and high-fidelity audio from a server to a client over a corporate network.

HARDWARE AND SOFTWARE REQUIREMENTS

Oracle8 is available on more than eighty platforms including 40+ UNIX, Windows NT, Alpha VMS, OS/390, and Novell Netware.

TRANSACTION PROCESSING AND DECISION SUPPORT

Transaction Processing

- Multithreaded server architecture
- Scalable SMP performance
- Shared database buffer cache
- Shared SQL cache (SQL statements, PL/SQL procedures, functions, packages, and triggers)
- Serially reusable SQL statements and PL/SQL procedures
- Shared dictionary cache
- Unlimited transaction length
- Fast and group commits
- Logging and archiving
- Deferred writes
- Serializable transactions
- Queuing
- Reverse key indexes to eliminate insertion hotspots
- Net8 Connection Manager and Connection Pooling
- XA support

Oracle8 Parallel Server

- Clustered and MPP parallel systems support
- Multiple nodes share access to a single Oracle8 database
- Provides single database image
- Easy incremental expansion of processing resources
- Linear scalable increases in performance
- High availability applications
- Consolidated system administration
- Parallel disk affinity
- Transparent application failover
- Fine-grained locking

SQL Optimizer

- Cost-based, syntax-independent optimization
- ANALYZE table statistics generation
- Nested-loop, hash, sort-merge, semi-, star-, anti-, and outer-joins
- Histograms
- Star join support using B-tree indexes or bitmap indexes
- Query transformations
- Parallel-aware optimizer
- Partition-aware optimizer
- EXPLAIN PLAN

Data Access

- B-tree indexes up to 32 columns
- Clustered tables and hash-clusters with application-specific hash functions
- ROWIDs
- Query results directly from index lookup
- Integrated bitmapped indexes
- Index organized tables

VLDB Support

- Unlimited database size
- Unlimited extent sizes
- Table partitioning
- Index partitioning
- 64-bit file system support

Concurrency Control and Reliable Results

- Unrestricted row-level locking
- Minimal contention between users
- No lock escalation
- Contention-free queries
- Non-blocking, multiversion read-consistent query results
- Unique sequence number generation
- Serializable transactions

Parallel SQL

- Parallel execution of SELECT, INSERT, UPDATE, and DELETE, with full application transparency
- Parallel execution of queries based on full table scans, index scans, and partition scans
- Both intra-partition and inter-partition parallelism
- Parallel sorts, joins, and aggregates
- Parallelization of PL/SQL functions

Parallel Data Management

- Parallel CREATE INDEX and CREATE TABLE AS SELECT
- Parallel, direct-path data loading
- Parallel constraint enabling, statistic collection, and summary creation
- Parallel direct database reads and direct database writes

High Availability

- On-line backup by file, tablespace, or database
- Server-managed backup/recovery
- On-line recovery
- Parallel recovery
- Parallel backup/restore utility
- Point-in-time database, tablespace recovery
- Database mirror resilvering
- Incremental multilevel backups
- Import/export utility
- Read-only tablespaces
- Mirrored multisegment log files
- Check-sums on database and redo log file blocks
- Dynamic resizing of database files
- Oracle8 Parallel Server for highly available applications
- Standby database

DISTRIBUTED SYSTEMS

Distributed Queries and Transactions

- Transparent remote and distributed query
- Distributed, optimized joins
- Location transparency, network transparency
- Integrated distributed query architecture
- Transparent, multisite distributed transactions
- Distributed SQL updates and remote procedure calls (RPCs)
- Commit transparency
- Automatic failure detection and resolution

Open Gateways

- Data source transparency
- Transparent SQL gateways
- Transparent procedural gateways
- Distributed queries and update transactions
- Compliant with XA standard for TP-monitor coordinated two-phase commit

Data Replication

- Multiple, read-only snapshots (basic primary site replication)
- Full transactional consistency and data integrity
- Full and subset table replication
- Incremental refresh of snapshot copied
- Event and demand based refresh
- Primary key snapshots

Advanced Replication

- Updatable snapshots (both master and snapshot tables updatable)
- Multiple master configurations (full table replication between master sites)
- Hybrid configurations (combine snapshot and multiple master configurations)
- Failover configuration support
- Automatic conflict detection and resolution
- Distributed schema management Synchronous and Asynchronous Replication support
- Subquery snapshots

OBJECT-RELATIONAL DATABASE

Object Types

- Complex object types
- VARRAY and Nested Table collection types
- Object references
- User-defined member methods, constructor methods, and ordering methods
- Follows SQL3

Views

- Relational and object views over relational and object data
- INSTEAD OF triggers to update views

Client-Side Object Support

- Client-side cache
- Object support in Pro*C, Oracle Call Interface
- Object Type Translator utility to generate C header files

LOBs

- Binary, character LOBs with random, piece-wise reads and writes; stored out of line
- LOB storage inside or outside the database

APPLICATION DEVELOPMENT

SQL Implementation

- 100-percent ANSI/ISO SQL 92 Entry Level compliant—NIST tested
- ANSI/ISO standard precompilers applications interface
- Robust SQL extensions including UNION, INTERSECT, MINUS, outer join, and tree-structured queries (CONNECT BY)
- SQL3 inline views (query in the FROM clause of another query)
- Updatable join views

Declarative Integrity Constraints

- 100-percent ANSI/ISO standard declarative entity and referential integrity constraints
- CHECK, DEFAULT, not NULL constraints
- PRIMARY, FOREIGN, UNIQUE keys
- Optional DELETE CASCADE
- Constraint checking at end of statement or end of transaction

Stored Procedures

- PL/SQL procedural extension to ANSI/ISO standard SQL. Strongly typed variable declarations (SQL datatypes). Block structure, flow control including FOR and WHILE loops, IF…THEN …ELSE SQL cursor support. Static and dynamic SQL support robust exception handling.
- Subprogram types: procedures, functions, and packages
- Subprograms stored in shared, compiled form
- Called from Oracle and third-party tools‡, Oracle Precompilers‡, Oracle Call Interface, SQL*Module‡ other stored procedures, database triggers
- Remote procedure calls (RPCs) protected by transparent two-phase commit
- User-defined PL/SQL functions in SQL
- Cursor variables for easy retrieval of multirow result sets
- Wrapper utility hiding PL/SQL application code in binary source format
- External procedure callouts
- Supplied packages for pipes, job queues, alerts, application info, HTML, and file I/O

Database Triggers

- Procedural code executed automatically on INSERT, UPDATE, or DELETE
- Triggers execute either BEFORE or AFTER operations
- Triggers fire once per statement or once per row
- Modeled after ANSI/ISO SQL3 specification

Programming Interfaces

- Oracle Precompilers‡
- Embedded SQL and PL/SQL application development 100-percent entry-level ANSI/ISO X3.135- 1992 compliant FIPS flagger meets FIPS 127-2 requirements
- Oracle Call Interface: procedure/function call interface
- SQL*Module‡
- SQL*Module language application development 100-percent entry-level ANSI/ISO X3.135- 1992 compliant FIPS flagger meets FIPS 127-2 requirements
- JDBC and JSQL for Java applications
- Multithreaded client application support
- Array operations
- Bundled calls

NATIONAL LANGUAGE SUPPORT

- Full 8- and 16-bit NLS support for European and Asian languages
- Unicode UTF-2 variable width encoding
- Per-session control of language preference with system defaults
- Character set conversion for heterogeneous client/server and distributed database environments
- National calendar support
- Dual database character sets

DATA SECURITY

- Choice of internal or external user authentication
- External choices include: operating system, OS security package, network operating system, security service, authentication device
- Encrypted passwords
- Password policy enforcement
- Global users and roles
- Full datastream encryption through DES and RSA RC4 encryption algorithms
- Complete protocol support and application transparency
- Fine-grain database privileges

- Hierarchical role-based security for group-level access control
- Site-customized DBA roles
- Roles are basis for ANSI/ISO SQL3 security standard
- Submitted for evaluation at US TCSEC C2, European ITSEC E3
- Automatic auditing on per-session or per-object basis
- Application-specific or context-sensitive auditing through PL/SQL stored procedures and database triggers

SYSTEMS MANAGEMENT

Database Reconfiguration

- Add files, tables, indexes, users on-line
- Dynamically and automatically grow and shrink files
- Truncate table, partition
- Add column to table on-line
- Dynamically modifiable system parameters
- Dependency tracking and automatic recompilation of views, stored procedures which access altered tables, views, procedures

Diagnostic and Monitoring

- Alert file
- SQL TRACE
- V$ monitoring views, GV$ global views
- DB VERIFY
- SGA monitoring API
- Application identification facility
- Remote SQL tracing

Oracle Enterprise Manager

- Systems Manager—easy to use, GUI, menu-driven DBA utility for Windows
- Secure remote database administration
- Data dictionary and dynamic performance tables
- Application identification facility
- PROFILEs limit user resource consumption
- Job Queues for automatic scheduling of PL/SQL stored procedure execution
- Remote SQL tracing
- SNMP Support: public RDBMS MIB, Oracle8 private MIB, Replication MIB
- Table TRUNCATE
- Enable/disable constraints and triggers
- Advanced space management features
- Oracle Expert and Oracle Trace

‡ Requires Oracle or third-party tool, Oracle Precompiler, or SQL*Module

ORACLE®

Enabling the Information Age™

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.415.506.7000
Fax +1.415.506.7200
http://www.oracle.com/

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.