



A Comparison of Oracle and MySQL



By Gregg Petri

Oracle9i Database and MySQL Database Server are relational database management systems (RDBMS) that efficiently manage data within an enterprise. Both relational database management systems are available for many hardware platforms and operating systems. Oracle9i Database and MySQL, as with all relational database management systems, use Structured Query Language (SQL) to interact with the database.

This white paper compares the features, performance, and administration capabilities of Oracle9i and MySQL to assist companies in choosing between these two relational database management systems. Note that many other relational database management systems, both commercial and open source, are available but are not discussed here.



Oracle Corporation, which developed the first commercial RDBMS in 1979, develops the Oracle9i Database. Oracle9i is one of the leading commercial databases available, and organizations of all sizes rely on Oracle to power mission-critical as well as other applications. However, some individuals or organizations perceive Oracle 9i Database as being expensive to license.

MySQL AB develops the MySQL Database Server, which is one of the most popular Open Source SQL databases. Because MySQL offers a free license, many companies are looking closely at MySQL when choosing the database management system that is right for them.

Oracle Overview

The current version of Oracle9i Database is Oracle9i Release 2 (9.2.0.2). Oracle9i Database is available in Personal, Standard, and Enterprise Editions. The Oracle Universal Installer installs the Oracle9i Database software, and calls configuration assistants to setup the networking components and create databases. Oracle Enterprise Manager (OEM) is a graphical application included with all editions that administrators use to manage the Oracle instances.

All editions include; the PL/SQL engine to develop stored procedures, triggers, and functions; a Java compiler and Virtual Machine (JVM) to develop Java stored procedures and triggers; XML support; an Apache Web server; and object-relational capabilities.

Personal Edition is for single user development and deployment, and is fully compatible with Standard and Enterprise Editions.

Standard Edition may be licensed on machines containing up to four processors, and supports online backup and recovery; recovery manager; flashback queries; autonomous transactions; external tables; advanced queuing; basic replication; distributed queries and transactions; the Internet File System; database and view triggers; and various other features.

Enterprise Edition includes all of the components of Standard Edition, plus advanced features such as Real Application Clusters (RAC); partitioning; online analytical processing (OLAP); data mining; parallel queries; multi-master replication; and spatial data support.

MySQL Overview

The current version of MySQL Database Server is 4.0.13. Developers can embed MySQL as a library within an application, or use MySQL as a standalone database engine. MySQL is available as binaries, or because it is Open Source, developers can download, modify, and compile the source code on the target server.

MySQL supports transactional and non-transactional tables, and one-way replication. The MySQL developers built the database for high performance, especially with the use of non-transactional tables. In addition, MySQL takes advantage of multiple processors, as it is fully multi-threaded using kernel threads.

Features Comparison

Oracle9i Database offers a rich set of development and administration tools, and recent releases have focused on security and scalability improvements. MySQL does not offer nearly as many development features, however many of the main deficiencies are addressed in future releases. To say that MySQL does not support a certain feature that Oracle9i Database supports will most likely invoke a response that agrees, but with a caveat that the feature is in an upcoming release. Depending on how important the missing feature is to an organization, not having the feature in the current release may or may not be acceptable.

The following sections compare how well Oracle9i Database and MySQL support major features in a relational database management system.

Data Types

Every column in a database table is assigned a data type. The data type may be native to the database management system, or may be a custom data type.

Each Oracle9i Database native data type belongs to one of six categories: character, number, date, large object (LOB), raw, and row identifier. Oracle

continued on page 42

Table 1. Comparison of Oracle and MySQL Data Types

| General Data Types | Oracle9i Database | MySQL | Notes |
|------------------------|--|---|---|
| Fixed-Length String | Up to 2,000 bytes | Up to 255 bytes | Oracle supports larger fixed length strings within tables. |
| Variable-Length String | Up to 4,000 bytes | Up to 255 bytes | Oracle supports larger variable length strings within tables. |
| Large Text | Up to 4GB | Up to 4GB Limited to 16MB with some tables | Oracle will not limit the size of large text objects. |
| Large Binary | Up to 4GB | Up to 4GB Limited to 16MB with some tables | Oracle will not limit the size of large binary objects. |
| Integer | Up to 38 digits of precision | Up to 264 digits (approximately 20 digits) of precision | Oracle supports larger integers. |
| Floating Point | Up to 38 digits of precision | -1.8308 to -2.2-308 2.2-308 to 1.8308 | MySQL supports larger floating point and double precision numbers. |
| Date | Date and time with second precision January 1, 4712 BCE to December 31, 4712 CE | Date and time with second precision January 1, 1000 to December 31, 9999 | Oracle ensures that date values are valid dates, and can store dates prior to 1000 CE. MySQL can store dates after 4712 CE. |
| Time | None | Second precision -838:59:59 to 838:59:59 | Oracle does not have a time-only data type. |
| Timestamp | Sub-second precision with time zones | Second precision with no time zones | Oracle supports sub-second precision and automatic time zone conversions. |
| Year | None | 1901 to 2155 | Oracle does not have a year-only data type. |
| Row Identifier | Local and universal | None | MySQL does not support row identifiers. Row identifier values are useful for quick updates. |
| Enumeration | None | Up to 65,535 distinct values | Oracle does have an enumeration data type; however, Oracle supports multiple-value check constraints on columns. |
| Set | None | Up to 64 members | Oracle does not have a set data type. |
| User-Defined | Developers define custom data types | None | Oracle supports complex custom data types. |

date data types only store valid dates, support automatic time zone conversion, and store sub-second precision.

MySQL categorizes data types into string, numeric, and date and time types. Unlike Oracle, MySQL date data types can store invalid dates, do not support time zones, and the precision of time values is limited to one second.

Refer to Table 1 for a summarized comparison of the Oracle9i Database and MySQL data types.

Table 1 shows the different approaches Oracle and MySQL used when establishing data types. Oracle9i Database data types provide applications with rich functionality, such as sub-second time values and time zone conversions, while ensuring data integrity, such as by validating date values. MySQL designed data types to store data as quickly as possible. Consequently, MySQL does not validate the data contents, and truncates values that exceed the column precision.

Tables

Tables, also known as relations, are the objects used to store data in a relational database management system.

Oracle9i Database uses tablespaces consisting of one or more data files on disk in which to create tables. Each tablespace can use a different block size to optimize storage. All tables consist of the same structure within the database. Oracle9i Database supports the use of temporary tables for an individual session, or global to all users. In addition, Oracle9i Database supports external tables to allow users to query data in flat files as if the data was in a database table.

MySQL supports six different types of tables, four of which do not support transactions (MyISAM, MERGE, ISAM, and HEAP) and two of which support transactions (InnoDB and BDB). All tables have a file in the database directory with a 'FRM' extension that contains the table and column layout.

At first glance, it appears that because MySQL offers six table types compared to Oracle's one table type, MySQL table capabilities are more powerful than Oracle tables. However, the multiple table types are a result of the initial MySQL design where fast, non-transactional tables were the only option. Because non-transactional tables do not support referential integrity, transactions, and hot backups, no mission-critical application development could utilize these tables. Given that Oracle9i Database and other RDBMS vendors support transactional tables, MySQL included InnoDB and BDB in later releases to allow MySQL to compete with other RDBMS vendors.

Serious application development would never consider using tables that do not support transactions or referential integrity, regardless of how fast the tables are accessible. Any application development where data integrity is a priority and applications require transactional capabilities, which includes all corporate applications, should only consider using transactional tables.

Both Oracle9i Database and MySQL support temporary tables, although all Oracle tables are recoverable, so no data is lost in the event of a database failure.

Indexes

Indexes are optional objects that are the primary method of providing faster access to data. Unique indexes only store unique key values from one or more columns. Non-unique indexes allow duplicate key values.

Oracle9i Database supports eight types of indexes. MySQL only supports one index type, but implements different data access paths using the various table types. The eight types of Oracle indexes are B-tree, B-tree Cluster, Hash Cluster, Reverse Key, Bitmap, Bitmap Join, Function-Based, and Domain.

The implementation of various index types demonstrates how extensively Oracle9i Database focuses on fast data retrieval, especially in large databases such as data warehouses. In addition, the use of reverse key and function-based indexes, which are not available in MySQL, allows for a great degree in flexibility in quickly accessing data where a B-tree index would not work.

Types

Complex data types extend the native data types provided by the relational database management system. Complex data types can limit the size, precision, and scale of a native data type, or can combine several other data types into a structure.

Complex data types provide a powerful mechanism for defining data types corresponding to objects in one place, and are a key component in Oracle's object-relational capabilities. Oracle9i Database supports both object types and collection types. Object types are similar to a class in object-oriented languages, which consist of attributes and methods. Collection types are similar to ENUM text types in MySQL where each column contains an indefinite number of elements all of the same type. Object types can inherit from a single parent type.

MySQL does not support user-defined data types.

Views

Views are database objects that return a query result from tables or other views. The query result appears the same as selecting rows from a table, and can restrict and aggregate data. Views should be updateable when it is possible to determine which underlying table the UPDATE statement affects.

Oracle9i Database supports the creation and update capabilities of views, including views built on multiple table joins Oracle views are updateable.

MySQL does not support views. However, the release 5.1 schedule includes views.

Views are a commonly used database objects that provide result sets of common queries so that applications do not need to code the same query in more than one place. In addition, views provide a means to restrict and aggregate data columns and rows to both summarize and limit the data available to users. To improve security, an administrator can grant access to a view but not allow access to the underlying tables.

Inline Views

Inline views, also called derived tables, provide similar functionality to views where a query result behaves the same as a table, except that inline views are not separate database objects. Rather, inline views are defined within a FROM clause. Often queries include inline views to aggregate data within a single query.

Oracle9i Database fully supports the use of inline views.

MySQL does not support inline views. However, the release 4.1 schedule includes inline views.

Inline views are a powerful mechanism for including a result set within a FROM clause of a query, such as to show the percentage of the whole for each row in a result set, or combine the results of several queries into a single result set. Because of the capabilities provided by inline views, the absence of inline view support in a database is a strong factor in not using MySQL for serious application development.

Outer Joins

Outer joins allow for a join predicate in a WHERE clause to be satisfied even if the value of the joined column in one of the tables does not exist. There are three types of outer joins: left, right, and full.

Oracle9i Database supports all three types of outer joins using ANSI syntax, and supports left and right outer joins using proprietary syntax in older releases.

MySQL supports left and right outer joins, but not full outer joins.

Without the support of full outer joins, MySQL queries requiring this functionality must use a UNION statement to concatenate the result set of query using a left outer join with the same query using a right outer join. Coding the same query twice, changing only the outer join from left to right in the two queries, is inefficient and redundant.

Synonyms

Synonyms provide an alias for any object in the database. Administrators and users often create synonyms to allow users to specify objects without including the qualifier for which user owns the object, or in which database instance the object resides.

Oracle9i Database supports the creation of synonyms for individual users as well as public synonyms that are available to all database users.

MySQL does not support synonyms. However, MERGE tables can specify a single table to act as a table synonym or alias.

Without the benefit of synonyms, queries must qualify database objects by prefixing object names with the database and owner names, unless the current user owns the objects. Forcing queries to qualify database objects reduces application flexibility, because the developer must update the queries within an application if an object moves to another database or owner.

Sequences and Auto-Increment

Sequence number generators are useful for requirements such as inserting a unique number into a column, for example to create primary key values.

Oracle9i Database uses SEQUENCE objects to generate sequential numbers. Sequences are not bound to an individual column, as sequence numbers are useful for purposes other than creating key values. Sequences can return numbers in ascending or descending order by any increment. An application or query selects the next value from the sequence, and uses the value as necessary to populate a primary key column or perform other application logic.

Similar to Microsoft Access, MySQL uses an AUTO_INCREMENT attribute assigned to a column when creating a table. As a query insert rows into a table, the AUTO_INCREMENT attribute automatically assigns the next value to the column. In addition, an AUTO_INCREMENT value can set sequential values for individual groups of data when an ENUM column is used with the automatically incrementing column. In this case, the value of the incremented column within each group starts at one and is incremented in relation to other elements in the group only.

The manner in which Oracle9i Database allows the developer to retrieve the next value as needed provides a high degree of flexibility in application development.

On the other hand, the ability to start and increment sequence numbers values at the initial value within a group is a useful feature provided by MySQL that is not available in Oracle9i Database.

continued on page 44

SELECTS

SELECT statements query data in the database. Because SQL is a standardized language, the syntax of SELECT statements between Oracle9i Database and MySQL are virtually identical.

Since SQL is based on set theory, result sets can be joined together using a UNION operation, subtracted using a MINUS operation, and elements in common found using an INTERSECT operation. Oracle9i Database supports all three multiple result set operations. MySQL only supports UNION. Without the capabilities of MINUS and INTERSECT operations, users running MySQL queries requiring this functionality must process the separate result sets using by a custom application.

Although relational database managements systems store data in two-dimensional tables, data can represent a hierarchical relationship. Within a table, one set of attributes can store the parent values of the current row, with the parent values at the top of the hierarchy not having any value assigned. Oracle9i Database supports hierarchical queries using CONNECT BY and START WITH clauses, and provides a LEVEL pseudo-column showing the hierarchy level for each row in the result set. MySQL does not support hierarchical queries. Because MySQL does not support hierarchical queries, queries requiring this functionality must call programs using recursive function calls to process all of the rows in the result set.

Subqueries, also known as sub-selects, provide an interim result set used by the main query to create the result set. Subqueries are non-correlated and executed once, or correlate to the outer query where the subquery executes for every row in the result set. Oracle9i Database supports both non-correlated and correlated subqueries. MySQL does not support subqueries; however, version 4.1 introduces subquery support. Subqueries are an important part of developing complex queries, and often an individual query cannot create the desired result set without subquery capabilities. Queries needing to correlate the rows in an outer result set with a subquery must use a custom application program to process each row in the result set separately, resulting in poor query performance.

Inserts

INSERT statements add data into one or more database tables. To insert data, specify the column values within a VALUES clause, or return the new values with a result set from a SELECT statement.

If the INSERT statement uses the VALUES clause, Oracle9i Database only allows the statement to specify data for one row in one table. However, INSERT statements using a SELECT statement can insert multiple rows including inserting data into different tables depending on which conditional clauses the data satisfies.

Unlike Oracle9i Database, MySQL allows INSERT statements with a VALUES clause to specify multiple sets of data to insert. However, INSERT statements using a SELECT statement can only insert data into one table.

The conditional insert capabilities included in Oracle9i Database are a powerful way to implement application logic within an INSERT statement. This functionality is especially useful when loading data into multiple tables within a data warehouse. On the other hand, the capability within MySQL to insert multiple rows using a single VALUES clause replaces the need for multiple INSERT statements, resulting in smaller data loading scripts.

Updates

UPDATE statements change existing values in one or more database tables to new values.

Oracle9i Database only allows each UPDATE statement to specify one table.

To update rows based on the values of corresponding data in another table, create a correlated update statement using a subquery.

MySQL allows an UPDATE statement to specify multiple tables. However, since MySQL does not support subqueries, specify join predicates within the WHERE clause to correlate updates using data in multiple tables. MySQL also allows an UPDATE statement to limit the number of rows to update, including using an ORDER BY clause to ensure that the limit applies to the top or bottom specified number of rows.

Although the syntax for UPDATE statements is slightly different between Oracle9i Database and MySQL, both methods provide sufficient flexibility to handle all update requirements. The ability to update a limited number of sorted rows is a useful feature available in MySQL that is not available in Oracle9i Database.

Deletes

DELETE statements remove one or more rows from one or more database tables.

Oracle9i Database only allows each DELETE statement to specify one table. To delete rows based on the values of corresponding rows in another table, create a correlated delete statement using a subquery.

As with UPDATE statements, MySQL allows a DELETE statement to specify multiple tables. Specify join predicates within the WHERE clause to correlate deletes using data in multiple tables. In addition, MySQL provides the ability to limit the ordered set of rows affected by the DELETE.

As with UPDATE statements, both Oracle9i Database and MySQL provide sufficient flexibility to handle all delete requirements. The ability to delete a limited number of sorted rows is a useful feature available in MySQL that is not available in Oracle9i Database.

TRUNCATE TABLE is an Oracle SQL extension that deletes all rows in a table. Unlike a DELETE statement, the RDBMS does not log the truncate operation; therefore, there is no way to undo the table truncation. MySQL supports the TRUNCATE TABLE command by dropping and recreating the table.

Merges

A common operation generically called “upsert” is to attempt to insert data into a table, and if the data already existed, update the data with a new value. Merge statements provide this functionality.

Oracle9i Database provides the MERGE statement to insert data into a table, while updating existing values in the table. The MERGE statement requires a clause to specify the target table, and a clause to specify one or more tables containing the new values. WHERE clause predicates specify how to join the tables, and one or more WHEN clause predicates specify how to insert or update the target table.

MySQL includes a REPLACE statement to provide “upsert” functionality. If the value to insert violates a primary key or other unique constraint, MySQL deletes the existing row and inserts a row with the new values. In addition, release 4.1 adds an ON DUPLICATE KEY UPDATE clause to the INSERT statement to update existing data during an insert operation.

Both Oracle9i Database and MySQL provide sufficient functionality to implement a data merge. MySQL offers the added flexibility to merge data without having to specify two separate tables, making the MySQL merge capability useful in individual queries.

Stored Procedures

Stored procedures are programs saved in the database containing one or more pre-compiled and pre-optimized SQL statements incorporated into a

procedural language. Stored procedures are an efficient way to query and process data in the database because the SQL statements are already parsed and optimized. Stored procedures provide a way to conditionally process SQL statements, and support procedural constructs such as conditional statements and iterations. Stored procedures also reduce network traffic, as the client only needs to send the stored procedure name and parameters to the database engine.

Oracle9i Database includes PL/SQL, Oracle's procedural language for writing programs that directly interact with SQL. Because SQL processes sets of data, PL/SQL is useful for writing program blocks where procedural capabilities are required to process data in the database. Oracle supports stored procedures written in PL/SQL as well as other languages such as Java. In addition to stored procedures, packages provide a way to combine multiple stored programs in an object-oriented fashion to provide parameter overloading, encapsulate programming logic, and inherit attributes and programs from parent procedures.

MySQL does not include a procedural language extension; therefore, MySQL does not support stored procedures saved in the database. However, release 5.0 includes stored procedure support.

The lack of a procedural language and stored procedure support is a major shortcoming of MySQL. Without a procedural language, developers cannot store application logic in the database. Running programs within the database provides for high performance, as well as increased flexibility as users who can connect to the database can execute the procedures.

In addition to the performance and program accessibility benefits, stored procedures provide an additional security layer in the database. Administrators can grant execute permission on a stored procedure without granting privileges on the tables that the procedure inserts, updates, or deletes. This allows for a strong degree of control over the application logic that manipulates the data in the database.

Triggers

Triggers are stored procedures that automatically execute when a database event occurs, or during insert, update, or deleted operations on a table. Triggers associated with table and view inserts, updates, or deletes can fire before or after the operation; triggers firing before the operation can modify the values specified in the insert or update statement. In addition, triggers can fire once for every row affected, or only one time to process all rows affected by the insert, update, or delete statement.

Oracle9i Database fires triggers for both database-level and table-level events. Database events that fire triggers include system startup and shutdown, object creation and alteration, and user logins and logouts. For tables and views, triggers can fire before or after the insertion, update, or deletion of one or more rows. A trigger can also fire based on an update of a specific column, rather than fire for any update to the table. Views based on joining tables use INSTEAD OF triggers to update the necessary columns in the base tables.

Since MySQL does not support stored procedures, MySQL also does not support triggers. However, release 5.0 includes some level of trigger support.

Missing trigger capability is another major shortcoming of MySQL. Database administrators rely heavily on database-level triggers to monitor database events. Application developers utilize table-level triggers extensively to ensure that data meets the business requirements, and to maintain values for de-normalized columns.

Functions

Functions are similar to stored procedures, except that they must return a value, and can be included in SELECT and WHERE clause statements. Both Oracle9i Database and MySQL include several built-in functions to support mathematical, character, and date processing. Allowing the developer to create custom functions increases the power of the database to solve complex queries.

Functions in the Oracle9i Database are stored procedures with a RETURN clause. Database packages can include functions as well as stored procedures to provide all of the package benefits.

MySQL allows developers to create user-defined functions using C or C++ programs that contain a predefined set of functions and must be thread-safe. These languages are more difficult to learn than PL/SQL, as they require an understanding of pointers, memory allocation, and the C or C++ compiler. After compiling the shared object, place the shared object in the linkage editor's library path, and recompile the MySQL engine to use dynamic loading. Upon server startup, the operating system dynamically links the user-defined functions into the MySQL engine. Because the required predefined C or C++ functions include the function name, it is not possible to overload a user-defined function.

Because writing functions in MySQL requires linking to the database engine and restarting the database, database administrators will limit the use of user-defined functions. However, user-defined functions are an integral part of developing programs that reside in the database when the standard set of functions is not sufficient. The inability for developers to code and test user-defined functions within MySQL without restarting the database is another major shortcoming of MySQL.

Java

Java is a free, platform-independent, object-oriented programming language developed by Sun Microsystems, Inc. A Java compiler creates byte-code class files, and a Java Virtual Machine (JVM) converts the byte-code into the target's native language at runtime. Because Java is platform-independent, Java programs compiled on one platform are executable on another platform without any changes. This platform independence allows Java developers to write programs that execute on any application tier such as client, server, and database; there is no need to learn different languages for the various application tiers.

Because PL/SQL is a proprietary programming language specific to Oracle products, Oracle9i Database includes a Java compiler and JVM within the database engine. This allows developers to write stored procedures, triggers, and functions in the standard Java programming language instead of the proprietary PL/SQL programming language. Developers compile Java programs directly into the database, or load an existing class file using the Oracle-supplied utility LOADJAVA.

MySQL does not support storing or executing Java programs in the database. As pointed out in the Stored Procedures section, the lack of an embedded programming language makes MySQL an unsuitable alternative for applications that require programs within the database.

XML

Extensible Markup Language (XML) is a standard file format used to transfer data between systems.

Oracle9i Database provides native support for XML with a custom XML data type and tools to query the data. Built-in packages provide additional functionality to receive, process, and submit XML documents.

MySQL does not provide native XML support.

continued on page 46

Transactions

Transactional capabilities provide the ability to treat multiple SQL statements as a single unit. A COMMIT statement permanently saves database changes, and a ROLLBACK statement undoes uncommitted changes. In addition, SAVEPOINT statements allow the user to define locations within the set of queries in which to issue a partial rollback, instead of issuing a rollback for all of the uncommitted changes.

Because Oracle9i Database only supports one type of table within the database, all data inserts, updates, and deletes for Oracle tables are transactional. Oracle takes the approach that every insert, update, and delete statement is part of transaction by default, and supports all transactional capabilities including save points.

In addition, Oracle9i Database provides for autonomous transactions. Autonomous transactions are a transaction within a transaction, and are useful for logging or debugging purposes. Inserts, updates, and deletes within an autonomous transaction are independent of any open transaction.

MySQL views transactional tables as a significant performance hit, and only recommends using transaction tables if necessary. If an application requires transactional capabilities, a table type that supports transactions, either InnoDB or BDB, is required; the default MyISAM table type does not support transactions. Both non-transactional and transactional tables can exist in the same database, and a single query can select from or modify both types of tables. MySQL takes an alternate approach that nothing is part of a transaction by default, and the database engine automatically commits all inserts, updates, and deletes unless the auto-commit functionality is disabled. MySQL only supports the COMMIT and ROLLBACK statements; it does not support the use of save points, another major shortcoming of MySQL.

Applications that depend on data integrity should never consider using non-transactional tables. In addition, the benefits of row-level locking and read consistency associated with all Oracle9i Database tables and InnoDB or BDB tables in MySQL ensure that the query processes data as it existed when the query started.

Distributed Databases

Partitioning data across physical databases allows for increased scalability, and can improve performance by placing data physically closer to the applications.

Oracle9i Database supports both data partitioning within a database instance, and database links to access data in other databases as if the data was located in the same database. In addition, Oracle provides a strong scalability and fault tolerant solution with Real Application Clusters (RAC) where multiple instances access the same set of physical data files.

MySQL does not support data partitions, and requires a server for each set of data files. Scalability is limited to the size of a single server.

The limitations within MySQL of not supporting distributed databases is a serious roadblock in creating a high-performance, scalable database solution. Without the benefits of clustering, a database is only as scalable as the hardware on which the database resides.

Replication

Replicating data is the process of copying data, synchronously or asynchronously, from one database into another database.

Oracle9i Databases create materialized views as replicas of tables residing in the master database. Because the materialized views are updateable where

the changes propagate to the master database, Oracle9i Database supports two-way replication.

MySQL supports one-way replication from the master to the replica by applying transaction log files to the replica database. If the database replicates transactional tables, utilize the hot backup utility to transfer the log files containing the changes to apply to the replica, or shutdown the database before copying the log files. Note that hot backup utility is not part of the free MySQL distribution.

Replication is a powerful tool for creating copies of databases to minimize the load of the master database, where the replication facility updates the replica as needed. Both Oracle9i Database and MySQL support one-way replication, however Oracle's support for two-way replication better positions Oracle9i Database as the better solution in a replicated environment.

Security

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

Oracle9i Database implements security for both users and roles. Administrators grant and revoke database-level privileges and object-level privileges to users and roles, and grant and revoke users and role membership to roles. Roles provide a method of granting privileges to many users with a single GRANT statement, or revoking privileges from many users with a single REVOKE statement. Oracle supports default and non-default roles. Logging into the database enables privileges granted to default roles; users can enable privileges associated with non-default granted roles, often by supplying a role password, after successfully logging in. This functionality is useful in restricting a user to only having the privileges if the user is running a specific application.

MySQL uses the user name and host to lookup the user's privileges in the system tables. The user table stores database-level privileges by user, and two tables maintain object-level privileges to restrict access to tables and columns. An administrator creates users by issuing GRANT statements, or by inserting values directly into the user table. MySQL does not use roles or groups to grant and revoke privileges to multiple users in individual statements.

The absence of database roles is another major shortcoming of MySQL. Without the ability to group users into roles, the database administrator will have to rely on other methods to track which users should have which permissions to each database object.

Auditing

To provide an additional security measure to create an audit trail of database and object-level statements, auditing facilities track commands issued against the database and audited tables.

Oracle9i Database provides an auditing facility to track individual users, database statements, and object statements. In addition, triggers can capture the text of all SQL statements issued against audited tables and views.

MySQL does not provide auditing capabilities.

Administration Comparison

Backups. The ability to reliably backup and restore a database in a timely fashion is an important component of any database. Administrators perform hot backups while a database is operational and cold backups while a database is down. In the event of a database failure, administrators restore

databases to a specified point in time using the backup files and the transactional logs since the last backup.

Oracle9i Database provides for both hot and cold backups, and includes the Recovery Manager (RMAN) utility to facilitate the backup process.

Since MySQL stores data in operating system files, administrators perform cold backups by simply copying the files. MySQL supports hot backups of InnoDB tables using a utility purchased from MySQL; hot backups are not part of the free distribution.

The ability to perform hot backups is a key component for databases that cannot afford any downtime. Users should account for the cost of the hot backup utility when evaluating whether MySQL is a viable RDBMS.

Performance Comparison

Companies frequently run benchmark tests to compare performance statistics between databases. Benchmarks are important if two or more databases satisfy a company's criteria for a relational database management system, and performance is the deciding factor.

The MySQL Web site contains links to graphs of benchmarks performed by eWeek magazine showing that Oracle9i Database and MySQL perform at similar levels in tests of response times and throughput. However, it is not clear from these graphs whether the benchmark used MyISAM or InnoDB tables. Since all of Oracle's tables are transactional, an "apples to apples" benchmark test must use transactional tables in the MySQL database as well. In addition, Oracle provides database clustering to achieve improved scalability and throughput, and since MySQL does not provide clustering, a multiple server comparison is not possible.

Summary

Both Oracle9i Database and MySQL are powerful relational database management systems that effectively manage large amounts of data.

Oracle9i Database is a full featured database engine that has successfully passed stringent security tests and has excelled in performance benchmarks. With built-in support for PL/SQL and Java, developers can build complex stored procedures, functions, and triggers that are stored and executed within the database. Views, subqueries, bi-directional replication, clustering, role-based security, and native support for Internet-based computing with the included Apache server and XML tools are just a few examples of how Oracle9i Database is well suited for managing mission-critical data for any size organization.

MySQL, a free alternative to Oracle, provides some of the advanced features of Oracle9i Database and other relational databases, such as row-level locking and read consistency. As MySQL AB releases newer versions of MySQL, MySQL will continue to grow as a viable alternative to commercial databases. Keep in mind, however, the MySQL support is not free, the hot backup utility is not included in the free distribution, and applications that utilize the free license become part of the public domain.

For applications that require the strongest security measures available; the performance and security benefits of a procedural language; support for views, subqueries, and other advanced SQL features; the benefits of a recovery manager to facilitate backups; and the ability to cluster across multiple servers, Oracle9i Database is a proven leader. On the other hand, if an application primarily requires a high-performance data store, MySQL is a strong option. Developers who only require a data store should consider MySQL as a viable alternative to commercial relational database management systems. However, in general, experienced developers insist on strong transactional integrity and the ability to create database programs, and

require the types of development capabilities that Oracle9i Database offers.

Frequently Asked Questions

What is MySQL?

MySQL is an Open Source relational database management system.

The MySQL source code and compiled binaries are available at <http://www.mysql.com>.

Who would use MySQL?

Companies that do not want to pay a licensing fee, and do not need the additional functionality available in a commercial RDBMS such as Oracle, DB/2, or Microsoft SQLServer.

MySQL is also useful for developing prototype applications or proofs of concepts.

Why would we use MySQL over Oracle?

Companies should consider MySQL if they do not want to pay a licensing fee to Oracle, they do not need Oracle's functionality that is not available in MySQL, and they do not need the advanced security and scalability capabilities of Oracle. Keep in mind that the free version does not include technical support or hot backup functionality, and applications that rely on the Open Source license of MySQL become Open Source.

When would we use Oracle over MySQL?

Companies use Oracle for application development requiring the capabilities that MySQL does not offer. For example, Oracle, but not MySQL, includes a procedural language to develop stored procedures, triggers, and functions; views and inline views; subqueries; hierarchical queries, advanced replication; dynamic role-based security, bitmap and reverse key indexes; and native Internet-based computing support.

What key feature differences are there between Oracle and MySQL?

| Feature | Oracle | MySQL |
|-----------------------|---------|---|
| Hot backups | Yes | Only for InnoDB tables, not part of the free license. |
| Transactions | Yes | Only for InnoDB tables |
| Referential Integrity | Yes | Only for InnoDB tables |
| Row level locking | Yes | Only for InnoDB tables |
| Read consistency | Yes | Only for InnoDB tables |
| Role-based Security | Yes | No |
| Replication | Two-way | One-way |
| Clustering | Yes | No |

You can see from the following list of future MySQL functionality that MySQL is basic compared to Oracle's functionality. Many of the features that Oracle has provided for several releases are not available in the current version of MySQL.

continued on page 48

| Feature | Planned MySQL Version |
|------------------------|-------------------------|
| Subqueries | 4.1 |
| Inline views | 4.1 |
| R-trees | 4.1 (for MyISAM tables) |
| Stored procedures | 5.0 |
| Foreign keys | 5.1 (3.23 with InnoDB) |
| Cursors | 5.0 |
| Views | 5.1 |
| Triggers | 5.1 |
| Full outer join | 5.1 |
| Constraints | 5.1 |
| Hierarchical queries | Not planned |
| INTERSECT, MINUS | Not planned |
| Inherited tables | Not planned |
| Extensible type system | Not planned |

Is there a JDBC driver for MySQL?

Yes, it's part of the MySQL Connector/J package on the MySQL site. The MySQL JDBC driver is a type 4 driver (the best kind—i.e., written 100% in Java).

Is there an ODBC driver for mysql?

Yes there is! The MyODBC Open Source driver is available from MySQL.

What other APIs does MySQL have?

C, PHP, and Perl are also supported APIs for MySQL.

How do I get started with MySQL?

Download the current production version of the software from the MySQL Web site. MySQL is available as compiled binaries, or you can download and compile the source code yourself. To install compiled binaries, simply create the MySQL directory and run the installation program to uncompress the binaries in the install directory.



About the Author

Gregg Petri is a senior consultant for TUSC. He holds a BS in Applied Computer Science and an MS in Computer Science, and he is OCA certified. He can be reached at PETRIG@tusc.com.