Vol. II, No. I "Make a Difference by Sharing the Knowledge" August 2003

TUSC Client Chronicle

Atlanta
Chicago
Denver
Detroit
Milwaukee

Industry News from "The Oracle Experts" at TUSC

Comparing the Oracle9*i* Database and MySQL Database Server

Oracle9*i* Database and MySQL Database Server are relational database management systems (RDBMS) that efficiently manage data within an enterprise. Both are available for many hardware platforms and operating systems.

Oracle9*i* Database and MySQL, as with all relational database management systems, use structured query language (SQL) to interact with the database.

Oracle Corp. developed the first commercial RDBMS in 1979. Today, Oracle9*i* is one of the leading commercial databases available and organizations of all sizes rely on Oracle to power mission-critical and other applications. However, there's a perception out there that Oracle9*i* is expensive to license.

MySQL Database Server was developed by MySQL AB. It's one of the most popular open source SQL databases around. Because MySQL offers free licensing for its product, many companies are seriously considering it when looking for a database management system that's right for them.

This white paper compares the features, performance, and administration capabilities of Oracle9*i* and MySQL and provides a guide for firms that are deciding which system works best for their needs.

NOTE: *Many other relational database management systems, both commercial and open source, are available but are not discussed here.*

Continued on page 3

TUSC to host security panel at OracleWorld

Security and TUSC. They figure to be two of the biggest attractions during OracleWorld Sept. 7-11 at the Moscone Convention in San Francisco. In one particular instance, they'll be paired at the same time.

The "Ask the Security Expert" panel discussion on Sept. 9 at 11 a.m. is one of six presentations TUSC will be delivering at OracleWorld. TUSC Chief Executive Officer **Richard J. Niemiec** will host the forum, which includes Mary Ann Davidson, chief security officer, Oracle Corp.; John Pescatore, vice president and research director, Gartner Group; Jeffrey Jonas, founder, Systems Research and Development; and Aaron Newman, chief technology officer of Application Security Inc. The one-hour session promises to be a lively discussion on a broad range of security topics and shouldn't be missed.

TUSC's speakers will be quite visible during this annual showcase of anything and everything Oracle! "The Oracle Experts" will be well-presented inside the exhibition hall (booth 2102), as well, where attendees will have an opportunity to learn more about how TUSC can help create solutions for all sorts of business problems. They'll also have an opportunity to enter a drawing to win great prizes!

TUSC also will also be delivering presentations and have a booth presence Sept. 14-17 at Oracle Applications Users Group Connection Point 2003 at the San Diego Convention Center.

For a complete schedule of TUSC's presentations during OracleWorld, please turn to page 2.

6 presentations scheduled

Join TUSC at OracleWorld in San Francisco Sept. 7-11

Monday, Sept. 8

Title: **"Oracle RAC Tuning: a High-Speed Tour"**
Paper No.: **36880**
Time: **11 a.m. (1 hour)**
Presenters: **Richard J. Niemiec, chief executive officer and Steve Rubinow, chief technology officer, Archipelago**
Product area: **Database**
Solutions areas: **High Availability and Management**
Target audience: **Database Administrators and Architects**
Level: **Intermediate**
How does tuning change when you expand it to the RAC environment? A lot! This session explores how to tune the RAC environment using statspack. You'll look at the key things to investigate and when they're in or out of tolerance ranges. The session will be fast-paced and cover a lot!

Title: **"Introduction to SOAP and Oracle Web Services"**
Paper No.: **35781**
Time: **4 p.m. (1 hour)**
Presenter: **Bradley D. Brown, chairman of the board and chief architect**
Product area: **Application Server**
Solution areas: **Integration and Web Services**
Target audience: **General Audience**
Level: **Introductory**
What's the next big thing in information technology? Will it be Web Services? When Brown wanted to learn about Web Services and, specifically, SOAP, he purchased and read four books. He combines all of that information in this presentation and shares with you the things you'll really need to know about Web Services. Will it be the next big thing? Are you prepared? Attend this one-hour presentation and find out!

Tuesday, Sept. 9

Title: **"Experience the Oracle9*i* PL/SQL New Features"**
Paper No.: **36954**
Time: **8:30 a.m. (1 hour)**
Presenter: **Joseph C. Trezzo, president and chief operating officer**
Product area: **Application Development Tools**
Solution areas: **Integration and Web Services**
Target audience: **Database Administrators and Developers**
Level: **Intermediate**
This presentation covers the new PL/SQL-related features specific to Oracle9*i*, including release 2. New features addressed are: PL/SQL commands (CASE statement), Oracle-supplied packages (DBMS_REDEFINITION, DBMS_METADATA), datatypes, functions, SQL commands and more. Each includes sample code to illustrate their capabilities and usefulness.

Title: **"Ask the Security Expert"**
Paper No.: **40086**
Time: **11 a.m. (1 hour)**
Presenters: **Rich Niemiec, chief executive officer, TUSC; Mary Ann Davidson, chief security officer, Oracle Corp.; John Pescatore, vice president and research director, Gartner Group; Jeffrey Jonas, founder, Systems Research and Development; and Aaron Newman, chief technology officer of Application Security Inc.**
Product area: **Database**
Solution areas: **Security and Identity Management**
Target audience: **General Audience**
Level: **Intermediate**
Hear from security experts in an open panel discussion forum. This session promises to be a lively discussion on a broad range of security topics and shouldn't be missed.

Title: **"Integrating Forms, Reports and Discoverer into Portal"**
Paper No.: **36846**
Time: **3:30 p.m. (1 hour)**
Presenter: **Christopher Ostrowski, technical management consultant**
Product area: **Application Server**
Solution areas: **Portals and Wireless; Integration and Web Services**
Target audience: **Database Administrators and Developers**
Level: **Intermediate**
Oracle Portal provides a robust development environment that allows you to develop Web pages extremely quickly. But what if you've been using Oracle Forms, Oracle Reports and Oracle Discoverer and don't want to recode to move your existing applications to Oracle Portal? This presentation shows the mechanics of integrating existing Forms, Reports and Discoverer worksheets and workbooks into Portal. It addresses issues such as security; maintenance of the application environment; use of keymap files for Forms and Reports; development options available in Discoverer; and the maintenance and monitoring of *i*AS' OC4J processes facilitating the Forms, Reports and Discoverer environments via the Enterprise Manager Web site and use of the dcmctl command-line utility. Additional material covers scalability features of *i*AS, including use of multiple application servers and multiple forms and reports servers within a single application server.

Wednesday, Sept. 10

Title: **"Essential 9*i* Features for Developers"**
Paper No.: **36633**
Time: **8:30 a.m. (1 hour)**
Presenter: **Anthony Catalano, vice president — operations**
Product area: **Database**
Solution areas: **High Availability**
Target audience: **Information Technology Managers, Database Administrators, Developers and General Audience**
Level: **Intermediate**
As new database versions roll out, so do some exciting new features. This presentation discusses the new Oracle9*i* and Oracle8*i* features that enhance developers' ability to create mission-critical Web-based applications. See how you can add reliability, stability, speed and flexibility to any application. Just a few of the topics addressed are case expression and case statement, native compilation of PL/SQL code, autonomous transactions, native dynamic SQL, bulk binds with collections, merge statement and new datatypes. Attendees gain an understanding of the features, find out when to use them and learn the design considerations associated with each.

Oracle9*i* Database vs. MySQL Database Server

Continued from page 1

Oracle overview

The current version of Oracle9*i* Database is Oracle9*i* Release 2 (9.2.0.2). Oracle9*i* Database is available in Personal, Standard and Enterprise Editions. Oracle Universal Installer installs Oracle9*i* Database software and calls configuration assistants to set up 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

Designed for single-user development and deployment and is fully compatible with the 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, multimaster 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 it as a stand-alone database engine. MySQL is available as binaries or, because it's open source, developers can download, modify and compile the source code on the target server.

MySQL Control Center (MySQLCC) is a separate open source graphical application that manages MySQL databases.

MySQL supports transactional and non-transactional tables and one-way replication. 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's fully multithreaded using kernel threads.

Database design comparison

The design and structure of Oracle9*i* Database is significantly different from MySQL.

Oracle design

Oracle9*i* Database installs as one or more database instances on a server. Within an instance, users create objects in their schemas. With the proper permissions, users can access data in tables in any schema within the instance, or in other Oracle instances both on the same server and on separate servers. The metadata defining the structure of all database objects, such as table and column names, is stored in internal database tables.

MySQL design

MySQL installs as one or more databases within a server. Within a server, users create objects in any of the databases in which they have the required permission. With the proper permissions, users can access data in any table in any database on the server. MySQL creates a directory for each database and stores the metadata describing table layouts in files within the database directory.

Features comparison

Oracle9*i* Database offers a rich set of development and administration tools. Recent releases have focused on security and scalability improvements. MySQL, meanwhile, doesn't offer nearly as many development features. However, many of the main deficiencies are addressed in future releases.

To say that MySQL doesn't support a certain feature that Oracle9*i* Database does most likely will invoke a response of agreement, but with a caveat that the feature will be added in an upcoming release.

Depending on how important the missing feature is to an organization, not having it in the current release may not be acceptable.

Continued on page 4

Oracle9*i* Database vs. MySQL Database Server

Continued from page 3

Waiting for a specific feature is risky as MySQL doesn't guarantee the dates of future releases and when it finally does release a new feature, it's likely the new functionality will contains bugs.

The following sections compare how well Oracle9*i* Database and MySQL support major features in a relational database management system.

Oracle9*i* Database datatypes

Every column in a database table is assigned a datatype. It may be native to the database management system or a custom datatype.

Each Oracle9*i* Database native datatype belongs to one of six categories: character, number, date, large object (LOB), raw and row identifier.

Character datatypes

Store strings, with byte values corresponding to the character set established during database creation.

- **CHAR —** fixed length character strings between 1 and 2,000 bytes.
- **VARCHAR2** and **VARCHAR —** variable length strings between 1 and 4,000 bytes. VARCHAR is deprecated and is a synonym for VARCHAR2 and, consequently, should not be used.
- **NCHAR** and **NVARCHAR2 —** unicode datatypes used for databases created with a unicode character set.
- **LONG —** variable length strings up to 2 gigabytes. The LONG datatype has many restrictions; therefore, Oracle deprecated it.

Number datatypes

Stores integer and floating point numbers, up to 38 significant digits. Number columns with a scale of zero are integers.

Date datatypes

Store point-in-time data.

- **DATE —** stores dates and times using four-digit years between Jan. 1, 4712 BCE, and Dec. 31, 4712 CE. DATE values only store precision to a second and don't correspond to a time zone.
- **TIMESTAMP**, **TIMESTAMP WITH TIME ZONE** and **TIMESTAMP WITH LOCAL TIME ZONE —** TIMESTAMP columns store dates and times with subsecond precision and include a time zone component. Oracle9*i* Database automatically converts TIMESTAMP values between the database time zone and time zone for the current database session.

Large object datatypes

Are up to 4 gigabytes.

- **BLOB**, **CLOB**, **NCLOB** and **BFILE —** the four large object datatypes. BLOB columns store binary data, CLOB and NCLOB columns store character data and BFILE columns store unstructured data in operating system files outside of the database.

Raw datatypes

Store data that applications will not interpret, including data whose value will not change when converting the data between operating systems.

- **RAW —** RAW columns store variable length data.
- **LONG RAW —** LONG RAW is deprecated and shouldn't be used.

Row identifier

Uniquely identifies every row in the database.

- **ROWID —** provides information on where the row is stored in the database.
- **UROWID —** universal row identifiers that, similar to ROWID values, also provide information on where the row is stored in the database. However, UROWID provides unique row identifiers when selecting from a linked table in a non-Oracle database.

MySQL Database Server datatypes

MySQL categorizes datatypes into string, numeric and date and time types.

String datatypes

Store character data. Unlike Oracle9*i* Database, by default MySQL compares and sorts most values without regard to case, unless the table creation specified the BINARY attribute.

- **CHAR —** fixed length character strings between 0 and 255 bytes.
- **VARCHAR —** variable length strings between 1 and 255 bytes.
- **BLOB —** the four binary large object types that store variable data with their maximum size in bytes are TINYBLOB (2^8-1), BLOB (2^{16}-1), MEDIUMBLOB (2^{24}-1) and LARGEBLOB (2^{32}-1). MySQL compares and sorts BLOB data in case-sensitive fashion.
- **TEXT —** the four binary large object types that store variable data with their maximum size in bytes are TINYTEXT (2^8-1), TEXT (2^{16}-1), MEDIUMTEXT (2^{24}-1) and LARGETEXT (2^{32}-1). Unlike BLOB data, MySQL compares and sorts TEXT data in case-insensitive fashion.
- **ENUM —** enumerations of up to 65,535 elements define that valid set of values allowable for a column. Either a name or an index position references the enumerations values. Check constraints on a column uses an enumeration.

Continued on page 5

Oracle9*i* Database vs. MySQL Database Server

Continued from page 4

- **SET —** a SET contains up to 64 string members whose column values are stored in a binary representation (first value is 2^0, second value is 2^1, etc.) MySQL compares the returned values to the query value using a bitwise AND; for example, comparing a set column to nine returns the first (2^0=1) and fourth (2^3=8) values.

Number datatypes

Store integer and floating point numbers.

- **DECIMAL —** floating point numbers stored as a string to preserve all precision.
- **INT —** the five integer datatypes are TINYINT (1 byte), SMALLINT (2 bytes), MEDIUMINT (3 bytes), INT (4 bytes) and BIGINT (5 bytes). Values that exceed the storage space are "clipped" to the maximum value; ensure that the application properly handles this unusual behavior.
- **FLOAT —** approximate floating point numbers; defaults to 4 bytes.
- **DOUBLE PRECISION** and **REAL — b**oth are 8-byte float point numbers.

Date datatypes

MySQL stores both dates and times in several datatypes. Unlike Oracle, MySQL can store invalid dates since it only validates that the month is between 0 and 12 and that the day is between 0 and 31. Note that zero is allowed in both month and day values. Ensure that applications utilizing date values don't store an invalid date. MySQL doesn't have a datatype that stores time in fractions of seconds. In addition, MySQL doesn't store time zone information with date and time values and consequently doesn't translate date and time values between time zones.

- **DATETIME —** contains a date and time value between Jan. 1, 1000 00:00:00, and Dec. 31, 9999 23:59:59.
- **DATE —** DATE columns contain a date value between Jan. 1, 1000, and Dec. 31, 9999.
- **TIMESTAMP —** automatically sets the column to the current date and time upon an INSERT or UPDATE of the data.
- **TIME —** stores a time value between –838:59:59 and 838:59:59 with precision up to a second.
- **YEAR —** 1-byte datatype for representing years between 1901 and 2155.

Refer to table 1 for a summarized comparison of Oracle9*i* Database and MySQL datatypes.

Table 1 — Oracle vs. MySQL datatypes

General Datatypes	Oracle9*i* 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 4 gigabytes	Up to 4 gigabytes. Limited to 16 megabytes with some tables	Oracle will not limit the size of large text objects.
Large binary	Up to 4 gigabytes	Up to 4 gigabytes. Limited to 16 megabytes with some tables	Oracle will not limit the size of large binary objects.
Integer	Up to 38 digits of precision	Up to 2^{64} digits (approximately 20 digits) of precision	Oracle supports larger integers.
Floating point	Up to 38 digits of precision	-1.8^{308} to -2.2^{-308}, 2.2^{-308} to 1.8^{308}	MySQL supports larger floating point and double precision numbers.
Date	Date and time with second precision. Jan. 1, 4712 BCE, to Dec. 31, 4712 CE	Date and time with second precision. Jan. 1, 1000, to Dec. 31, 9999	Oracle ensures that date values are valid dates and can store them prior to 1000 CE. MySQL can store dates after 4712 CE.
Time	None	Second precision. -838:59:59 to 838:59:59	Oracle doesn't have a time-only datatype.
Timestamp	Subsecond precision with time zones	Second precision with no time zones	Oracle supports subsecond precision and automatic time zone conversions.
Year	None	1901 to 2155	Oracle doesn't have a year-only datatype.
Row identifier	Local and universal	None	MySQL doesn't support row identifiers; row identifier values are useful for quick updates.
Enumeration	None	Up to 65,535 distinct values	Oracle does have an enumeration datatype; however, it supports multiple value check constraints on columns.
Set	None	Up to 64 members	Oracle doesn't have a set datatype.
User defined	Developers define custom datatypes	None	Oracle supports complex custom datatypes.

Continued on page 6

Oracle9*i* Database vs. MySQL Database Server

Continued from page 5

Table 1 shows the different approaches Oracle and MySQL used when establishing datatypes. Oracle9*i* Database datatypes provide applications with rich functionality — like subsecond time values and time zone conversions — while ensuring data integrity, such as by validating date values.

By allowing developers to create custom datatypes, Oracle9*i* Database ensures maximum flexibility. MySQL designed datatypes to store data as quickly as possible. Consequently, it doesn't validate the data contents and truncates values that exceed the column precision. Applications storing data in MySQL must handle invalid values to ensure the database stores the expected values.

Tables

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

Oracle9*i* Database uses tablespaces consisting of one or more datafiles 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. Oracle9*i* Database supports the use of temporary tables for an individual session or global to all users. In addition, Oracle9*i* 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 don't support transactions (MyISAM, MERGE, ISAM and HEAP) and two of which do (InnoDB and BDB). All tables have a file in the database directory with a "FRM" extension that contains the table and column layout. Administrators define the type of table upon table creation and can later change it using an ALTER TABLE statement.

MyISAM tables
The default table type that uses a B-tree index. Data is stored in a file with a "MYD" extension and the index is stored in a file with a "MYI" extension. MyISAM tables are fast since there is no transaction overhead.

MERGE tables
A logical combination of identical MyISAM tables that MySQL treats as one table in SELECT, DELETE and UPDATE statements. The merge definition is stored in a file with a "MRG" extension.

ISAM tables
MySQL has deprecated ISAM tables and recommends not using them. MyISAM tables provide the same functionality.

HEAP tables
Use hashed indexes, but data will be lost if MySQL crashes. Therefore, HEAP tables should only be used for temporary tables.

InnoDB tables
Provide transactional tables with row-level locking and non-locking consistent reads, similar to Oracle9*i* Database tables. InnoDB tables have been included in MySQL 3.23 and later in response to users requiring transactional capabilities within the database. Unlike MyISAM tables, InnoDB tables support foreign key constraints.

BDB tables
BerkeleyDB (BDB) tables provide transactional capabilities similar to InnoDB tables; however, MySQL doesn't support them on all operating systems.

At first glance, it appears that because MySQL offers six table types compared to Oracle's one, MySQL table capabilities are more powerful than Oracle's. 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 don't support referential integrity, transactions and hot backups, no mission-critical application development could utilize these tables. Given that Oracle9*i* Database and other RDBMS vendors support transactional tables, MySQL included InnoDB and BDB in later releases to allow it to compete.

Serious application development would never consider using tables that don't support transactions or referential integrity, regardless of how fast the tables are accessible. With Oracle's focus on robust data integrity and quick recovery, non-transactional tables are of no value.

Continued on page 7

Oracle9*i* Database vs. MySQL Database Server

Continued from page 6

Any application development where data integrity is a priority and applications require transactional capabilities — including all corporate applications — should only consider using transactional tables.

Both Oracle9*i* Database and MySQL support temporary tables, although all Oracle tables are recoverable, so no data is lost in the event of a database failure. Because Oracle9*i* Database supports views, Oracle can utilize them and table rows appended by UNION statements to provide the functionality of MySQL MERGE tables.

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.

Oracle9*i* Database supports eight types of indexes.

MySQL only supports one index, but implements different data access paths using the various table types.

Oracle indexes

The eight types of Oracle indexes are:

- **B-tree —** the primary indexes used by Oracle9*i* Database and are similar to the MyISAM tables in MySQL.
- **B-tree clusters —** are a way to store data in tables that are frequently joined together, such that the joined column values are stored in the same data blocks.
- **Hash clusters —** are a method to store common data together that is useful when a table is frequently queried using equality comparison. Hash functions calculate index values, such that equal values are stored together. Hash clusters are similar to hash tables in MySQL, except that data isn't lost if the database crashes.
- **Reverse key —** are stored in reverse order within the index. Reverse key indexes are useful when data isn't evenly distributed using the first part of the values, but is evenly distributed using the last part of the values.
- **Bitmap —** each row identifier in the index is mapped to a bit value, improving efficiency for queries with many conditions in a WHERE clause on the same table.
- **Bitmap join —** star schemas use bitmap join indexes where the index is on the fact table with indexed columns corresponding to each dimension.
- **Function-based —** typically, indexes aren't usable if a function is applied to a column in the WHERE clause. Function-based indexes are B-tree or bitmap indexes where a function result is stored in the index. If the function defined in the index is applied in the WHERE clause, the function-based index can quickly return the associated row identifiers.
- **Domain —** extend over an application domain to manage complex datatypes, such as binary files and spatial data.

The implementation of various index types demonstrates how extensively Oracle9*i* 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 aren't available in MySQL, allows for a great degree in flexibility in quickly accessing data where a B-tree index wouldn't work.

Types

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

Oracle9*i* 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 doesn't support user-defined data types.

Complex datatypes provide a powerful mechanism for defining datatypes corresponding to objects in one place and are a key component in Oracle's object-relational capabilities. Columns in tables can specify the complex datatype, as well as variables in stored procedures, triggers and functions. If an object definition changes, administrators only need to make the change in one place. If an application using MySQL Database Server requires complex datatypes, a database engine such as Oracle9*i* Database is required alongside MySQL to provide this functionality.

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.

Continued on page 8

Oracle9*i* Database vs. MySQL Database Server

Continued from page7
Views should be updateable when it's possible to determine which underlying table the UPDATE statement affects.

Oracle9*i* Database supports the creation and update capabilities of views, including views built on multiple table joins. Oracle views are updateable.

MySQL doesn't support views. However, the release 5.1 schedule includes them.

Views are commonly used database objects that provide result sets of common queries so that applications don't 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.

Until release 5.1, MySQL didn't provide views based on the philosophy that they are unnecessary since MySQL provides column-level security. However, views provide many more capabilities beyond column-level security and are an important component in application development. Without views, a database cannot provide row-level security and applications must code common queries in multiple locations.

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 aren't separate database objects. Rather, inline views are defined within a FROM clause. Oftentimes, queries include inline views to aggregate data within a single query.

Oracle9*i* Database fully supports the use of inline views.

MySQL doesn't support inline views. However, the release 4.1 schedule includes them.

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. Without inline views, an application would need to load a temporary table with the interim result sets and join the main query to the temporary table, resulting in additional coding and database overhead.

Because of the capabilities provided by inline views, the absence of inline view support in a database is a strong factor in not using the database 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 doesn't exist. There are three types of outer joins: left (the result set contains rows in the lefthandside table even if the join cannot complete), right (the result set contains rows in the righthandside table even if the join cannot complete) and full (the result set contains rows from both tables even if the join cannot complete).

Oracle9*i* 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 them to allow users to specify objects without including the qualifier for which user owns the object or in which database instance the object resides. Synonyms don't link to database objects; dropping the actual object that a synonym aliases doesn't drop the corresponding synonym.

Oracle9*i* Database supports the creation of synonyms for individual users, as well as public synonyms that are available to all database users.

Continued on page 9

Oracle9*i* Database vs. MySQL Database Server

Continued from page 8

MySQL doesn't 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. If there are many columns that uniquely identify a row, often a sequence number value serves as the primary key to improve query performance with a unique index on the data columns that uniquely identify a row.

Oracle9*i* Database uses SEQUENCE objects to generate sequential numbers. Sequences aren't bound to an individual column, as sequence numbers are useful for purposes other than creating key values. In addition, a single sequence can create values for more than one column. 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 inserts 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 flexibility of not having to bind sequence numbers to specific table columns is useful in developing applications that utilize sequence numbers for purposes other than populating a primary key. Frequently, an application needs a sequential number, but must manipulate the next value before storing the value in the database. The manner in which Oracle9*i* 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 number values at the initial value within a group is a useful feature provided by MySQL that's not available in Oracle9*i* Database.

Selects

SELECT statements query data in the database. Within a SELECT statement, FROM, WHERE, GROUP BY, HAVING and ORDER BY clauses may be included to define, aggregate and sort the result set. Because SQL is a standardized language, the syntax of SELECT statements between Oracle9*i* Database and MySQL are virtually identical.

At a minimum, Oracle9*i* Database requires the SELECT and FROM clauses to be present in a complete query. If no database table is appropriate for the query, the single-row, single-column system table named DUAL can be used in the FROM clause. All values passed to functions and used in WHERE clause comparisons are case sensitive.

MySQL allows queries that only contain the SELECT clause; the FROM clause is optional. All values used in WHERE clause comparisons are case insensitive unless the SQL for creating or querying the table specifies the BINARY attribute.

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.

Oracle9*i* Database supports all three multiple result set operations.

Continued on page 10

Oracle9*i* Database vs. MySQL Database Server

Continued from page 9

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 a custom application. Developing a custom comparison program will result in a complex and slow query process.

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.

Oracle9*i* 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 doesn't support hierarchical queries. Because of that, queries requiring this functionality must call programs using recursive function calls to process all of the rows in the result set. Developing a custom program to show the data hierarchy will result in a complex and slow query process.

Subqueries, also known as subselects, 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.

Oracle9*i* Database supports both non-correlated and correlated subqueries.

MySQL doesn't 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, Oracle9*i* 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 Oracle9*i* 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 Oracle9*i* 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.

Oracle9*i* 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 doesn't 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.

Continued on page 11

Oracle9*i* Database vs. MySQL Database Server

Continued from page 10

Although the syntax for UPDATE statements is slightly different between Oracle9*i* 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 isn't available in Oracle9*i* Database.

Deletes

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

Oracle9*i* 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 Oracle9*i* 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 isn't available in Oracle9*i* Database.

TRUNCATE TABLE is an Oracle SQL extension that deletes all rows in a table. Unlike a DELETE statement, the RDBMS doesn't 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 it with a new value. Merge statements provide this functionality.

Oracle9*i* 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. Oracle implemented merge functionality for use in loading data into a data warehouse, thus MERGE statements aren't as useful in individual queries.

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 Oracle9*i* 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 precompiled and pre-optimized SQL statements incorporated into a procedural language. They're 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. They also reduce network traffic as the client only needs to send the stored procedure name and parameters to the database engine.

Oracle9*i* 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 — although non-PL/SQL programs require a PL/SQL wrapper program.

Continued on page 12

Oracle9*i* Database vs. MySQL Database Server

Continued from page 11

In addition to stored procedures, packages present 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 doesn't include a procedural language extension; therefore, it doesn't 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. Frequently, a user requires procedural constructs to perform a complex database operation or conditionally process one or more query result sets. Without a procedural language, developers can't store application logic in the database and must rely on an application server or external programs to process data. 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; those firing before the operation can modify the values specified in the insert or update statement. Also, triggers can fire once for every row affected or only one time to process all rows affected by the insert, update or delete statement.

Oracle9*i* Database fires triggers for both database- 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 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 doesn't support stored procedures, it also doesn't 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 denormalized columns.

With table-level triggers, the database engine guarantees it will automatically execute the trigger for every insert, update and delete statement for a table. If a transaction attempts to insert or update data that doesn't meet the business requirements, the trigger can abort the transaction to ensure that the table data remains clean.

In short, triggers are another way that Oracle9*i* Database maintains data integrity. Without the functionality of triggers, there's no way to ensure that all queries manipulating database data are meeting the business requirements and no way to update views built on multiple table joins.

Functions

Functions are similar to stored procedures, except they must return a value and can be included in SELECT and WHERE clause statements. Both Oracle9*i* 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 Oracle9*i* 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.

Continued on page 13

Oracle9*i* Database vs. MySQL Database Server

Continued from page 12

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 isn't 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 isn't sufficient. The inability of developers to code and test user-defined functions within MySQL without restarting the database is another major shortcoming.

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, Oracle9*i* 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 doesn't 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. It's a tag-based language similar to hypertext markup language (HTML) that stores data in a hierarchical format often extracted from one database, transformed and loaded into another database. A validating parser can verify that an XML document passes the structure requirements before processing the data. XML is a key component in Internet-based computing to transfer data between organizations.

Oracle9*i* Database provides native support for XML with a custom XML datatype and tools to query the data. Built-in packages provide additional functionality to receive, process and submit XML documents.

MySQL doesn't provide native XML support. Datatypes are available to store large strings; however, there is no built-in functionality to receive, parse, query and submit the XML documents. Until MySQL supports functions within the database, it can't accommodate advanced requirements, such as native XML support.

Both Oracle SQL*Plus and the MySQL command line interface can convert result set output to XML or HTML format.

Transactions

Transactional capabilities provide the ability to treat multiple SQL statements as a single unit. Data inserts, updates and deletes are committed to the database only if all of the statements within the transaction are successful.

Continued on page 14

Oracle9*i* Database vs. MySQL Database Server

Continued from page 13

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 Oracle9*i* 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, Oracle9*i* Database provides for autonomous transactions — a transaction within a transaction that's useful for logging or debugging purposes. Inserts, updates and deletes within an autonomous transaction are independent of any open transaction. Committing or performing a rollback of the autonomous transaction doesn't affect the disposition of pending changes within another open transaction.

MySQL views transactional tables as a significant performance hit and only recommends using transactional 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 doesn't 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. To create a transaction, either disable the auto-commit feature or explicitly start a transaction.

If a query or set of queries combine transactional and non-transactional tables, MySQL saves changes to inserts, updates and deletes even if the user issues a ROLLBACK statement. MySQL only supports the COMMIT and ROLLBACK statements; it doesn't support the use of save points.

Transactions are a major component of application development; frequently, an application can only save data changes from a single INSERT, UPDATE or DELETE statement if other statements within the transaction succeed. 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 Oracle9*i* Database tables and InnoDB or BDB tables in MySQL ensure that the query processes data as it existed when the query started.

The fact that MySQL doesn't support save points is another shortcoming. Applications utilize save points when processing large sets of data, where updates to one row have failed should not roll back the changes to the accepted rows, yet the application is not yet ready to commit the changes to the accepted rows.

Distributed databases

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

Oracle9*i* 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. Each table only exists in one Oracle instance, but is accessible by linked instances. 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 datafiles.

MySQL doesn't support data partitions and requires a server for each set of datafiles. 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 it resides.

Continued on page 15

Oracle9*i* Database vs. MySQL Database Server

Continued from page 14

Replication

Replicating data is the process of copying data — synchronously or asynchronously — from one database into another. Replicas are useful when users need to query a database, but the queries shouldn't affect the performance of inserting, updating and deleting data in the master database.

Oracle9*i* 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, Oracle9*i* 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. Also, be careful of potential issues when replicating transactional and non-transaction tables within the same transaction.

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 Oracle9*i* Database and MySQL support one-way replication; however, Oracle's support for two-way replication better positions Oracle9*i* Database as the preferred 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. The GRANT and REVOKE SQL statements maintain database and object privileges.

Oracle9*i* Database implements security for both users and roles. Administrators grant and revoke database- 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. An administrator creates users with a CREATE USER statement; new users don't have any privileges, including one to login, until the administrator uses GRANT statements to bestow the database and object privileges.

In addition to database privileges, the use of views and stored procedures is an additional method of implementing a security scheme. A user granted select permission on a view, but not on the base tables referenced by the view, can only query the rows and columns that satisfy the view query. Likewise, a user granted execute permission on a stored procedure, but not given access to the base tables referenced by the procedure code, can only execute the procedure and can't select, insert, update or delete the data in the base tables. Since MySQL doesn't currently support views or stored procedures, the privilege tables are the only mechanism available to secure the data.

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 doesn't 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. Granting the necessary permissions on new objects to a group of users, especially in the absence of database-level triggers, requires a significant effort by the database administrator. Without the ability to group users into roles, the DBA will have to rely on other methods to track which users should have which permissions to each database object.

Continued on page 16

Oracle9*i* Database vs. MySQL Database Server

Continued from page15

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.

Oracle9*i* 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 doesn't provide auditing capabilities.

In order to maintain a record of which users modified database objects or modified data especially in sensitive tables, companies should implement audit facilities to track all database and data changes. In the event of database problems stemming from malicious or accidental queries, the database administrator can determine the person responsible for the action.

Administration Comparison

Administration tools

Database administration tools provide a simpler means of administering and monitoring one or more databases instead of issuing multiple SQL statements.

Oracle9*i* Database includes Oracle Enterprise Manager (OEM), a Java application that allows the administrator to easily configure and monitor Oracle instances and user sessions. Using OEM, the administrator can create new users and objects, manipulate storage and manage all aspects of the database. OEM also provides advice on how to size memory and storage.

MySQL includes the command line MYSQLADMIN utility and a Windows-based WINMYSQLADMIN tool to administer the database.

For both Oracle9*i* Database and MySQL, graphical administration tools ease the tasks of maintaining database objects and users.

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 when it's down. In the event of a failure, administrators restore databases to a specified point in time using backup files and transactional logs since the last backup.

Oracle9*i* 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 can't 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.

MySQL's Web site contains links to graphs of benchmarks performed by *eWeek* magazine showing that Oracle9*i* Database and MySQL perform at similar levels in tests of response times and throughput. However, it's 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 doesn't, a multiple server comparison isn't possible.

Continued on page 17

Oracle9*i* Database vs. MySQL Database Server

Continued from page 16

Frequently Asked Questions

Question
What is MySQL?

Answer
MySQL is an open source relational database management system. According to MySQL documentation, "The MySQL software delivers a very fast, multithreaded, multi-user and robust SQL (structured query language) database server. MySQL Server is intended for mission-critical, heavy-load production systems, as well as for embedding into mass-deployed software." MySQL source code and compiled binaries are available at www.mysql.com.

* * * * *

Question
Who wrote MySQL?

Answer
The three men in Sweden who originally wrote MySQL are David Axmark, Allan Larsson and Michael "Monty" Widenius (the principal author). Today, MySQL AB employs a group of developers to continue development and support of MySQL.

* * * * *

Question
Do these guys make money?

Answer
MySQL AB makes money on support, services, commercial licenses and royalties.

* * * * *

Question
Who would use MySQL?

Answer
Companies that don't want to pay a licensing fee and don't need the additional functionality available in a commercial RDBMS, such as Oracle, DB/2 or Microsoft SQLServer. MySQL also is useful for developing prototype applications or proofs of concepts.

* * * * *

Question
Why would we use MySQL over Oracle?

Answer
Companies should consider MySQL if they don't want to pay a licensing fee to Oracle, don't need Oracle's functionality that isn't available in MySQL and don't need the advanced security and scalability capabilities of Oracle. Keep in mind that the free version doesn't include technical support or hot backup functionality and applications that rely on the open source license of MySQL become open source.

* * * * *

Question
When would we use Oracle over MySQL?

Answer
Companies use Oracle for application development requiring the capabilities that MySQL doesn't offer. For example, Oracle, 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.

* * * * *

Question
What key feature differences are there between Oracle and MySQL?

Answer

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 it's 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 18

Oracle9*i* Database vs. MySQL Database Server

Continued from page 17

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

* * * * *

Question
Is there a JDBC driver for MySQL?

Answer
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 percent in Java).

* * * * *

Question
Is there an ODBC driver for MySQL?

Answer
Yes there is! The MyODBC open source driver is available from MySQL.

* * * * *

Question
What other application program interfaces (APIs) does MySQL have?

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

* * * * *

Question
Is MySQL a Microsoft Access replacement?

Answer
No, unlike Microsoft Access, MySQL doesn't have an integrated development environment, nor does it provide the ability to create and process forms. MySQL is simply a database engine.

* * * * *

Question
Can I embed MySQL into a product without making my application open source?

Answer
Yes, MySQL AB offers a commercial license of MySQL that allows you to embed the MySQL database into your product. If you don't purchase a commercial license, your software is open source.

* * * * *

Question
Does MySQL use a system global area (SGA) buffer cache to cache SQL queries?

Answer
As of version 4.0.1, MySQL contains a feature called "Query Cache" that's similar to Oracle's buffer cache in the SGA. However, any DML or DDL (INSERT, UPDATE, DELETE, TRUNCATE, ALTER or DROP) statements to the table clear the cache for that table. Queries that contain functions (CONNECTION_ID, USER, etc.) aren't cacheable. In other words, the MySQL cache is new to MySQL and isn't as sophisticated as Oracle's SGA buffer cache.

* * * * *

Question
Why are there six types of tables for MySQL?

Answer
MySQL provides six table types as opposed to Oracle's single table type. Four of the table types (HEAP, ISAM, MERGE and MyISAM) aren't transaction-safe, while two types (i.e., InnoDB and BDB) support transactions. The main reason for the six different table types is various access types (non-transactional) and the need for transaction capabilities.

Developers require transaction support to build mission-critical applications; therefore, MySQL included two transaction-safe table types in version 3.23 to compete with databases such as Oracle.

Continued on page 19

Oracle9*i* Database vs. MySQL Database Server

Continued from page 18

All of Oracle's table types are transaction-safe. In other words, if Oracle9*i* Database (or MySQL for transaction-safe tables) or your server crashes, you can get your data back, either by automatic recovery or from a backup, plus the transaction logs. Non-transaction-safe tables don't support commit and rollback or two-phase commits. All transactions are immediately committed, similar to DDL statements. The advantage of non-transaction-safe tables is that they are faster since there is no transaction overhead and they use less disk space and memory.

Oracle9*i* Database provides different index types for different access types and temporary tables similar to HEAP tables in MySQL.

* * * * *

Question
How do I monitor the MySQL Database?

Answer
Create a table to monitor the database:

```
CREATE TABLE innodb_monitor(a INT) TYPE = INNODB;
```

To stop the log, drop the table:

```
DROP TABLE innodb_monitor;
```

* * * * *

Question
How do I get started with MySQL?

Answer
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.

Summary

Both Oracle9*i* Database and MySQL are powerful relational database management systems that effectively manage large amounts of data.

Oracle9*i* Database is a full-featured database engine that has successfully passed stringent security tests and 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 Oracle9*i* 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 Oracle9*i* Database and other relational databases, such as row-level locking and read consistency. As MySQL AB releases newer versions, it will continue to grow as a viable alternative to commercial databases. Keep in mind, however, MySQL support isn't free, the hot backup utility isn't included in the free distribution and applications that utilize the free license become part of the public domain.

Oracle9*i* Database is a proven leader for applications that require the strongest security measures available; performance and security benefits of a procedural language; support for views, subqueries and other advanced SQL features; benefits of a recovery manager to facilitate backups; and the ability to cluster across multiple servers.

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 can consider MySQL as an 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 Oracle9*i* Database offers.

Bibliography

Oracle9i Database Concepts Release 2 (9.2), March 2002

MySQL Reference Manual.

Acknowledgments

Oracle, Oracle9*i* Database and PL/SQL are registered trademarks of Oracle Corp.

MySQL is a registered trademark of MySQL AB.

Microsoft Access and Microsoft SQLServer are registered trademarks of Microsoft Corp.

DB/2 is a registered trademark of International Business Machines (IBM) Corp.

Training schedule for remainder of 2003

Equipping students with the tools and knowledge to meet the challenges of today's complex information environment is the primary goal of TUSC Training.

Through personalized hands-on learning, limited classroom seating and comfortable settings, our students receive an in-depth instruction of the topics covered in a cost-effective manner.

Whether you're a first-time user or seasoned professional, TUSC Training provides you with the building blocks to successfully implement current technology.

TUSC operates state-of-the-art training facilities — including large-screen projectors strategically placed in the classroom, plus MS Meeting on each workstation for students' viewing — at its offices in Chicago.

Class size is designed to accommodate 12 students. Students will have personal workstations to maximize their investment in learning.

Many clients have curriculum needs that are specific to their environments. To accommodate those customers, TUSC offers courses in a private and/or customized setting. In addition, TUSC offers mentoring packages to assist customers' personal needs at their own site.

Following is the TUSC Training schedule for October, November and December.

For more information on any of the classes, including course outlines and tuition, please call TUSC at 800/755-8872, email training@tusc.com or visit TUSC's Web site at www.tusc.com.

Chicago Office

	OCTOBER	NOV.	DEC.
Introduction to Oracle		14	
SQL & SQL*Plus	13-14	17-18	8-9
PL/SQL	15-17	19-21	10-12
Advanced PL/SQL		10-11	18-19
Oracle Forms Workshop		3-7	
Oracle Reports Workshop		12-13	
Application Tuning for Developers	14-16		15-17
Oracle Database Administration			
Oracle Backup & Recovery			
Performance Tuning	6-8		
Data Warehouse Bootcamp	20-22		
Introduction to Java			1-3
Integrating Oracle & Java			4-5
Introduction to Oracle9*i*AS Portal	27-29	***	***
Internet to Oracle Web Development	***	***	***
Oracle Application Server Development			
Internet Application Server (9*i*AS) Configuration and Management	***	***	***
Introduction to Oracle Portal	***	***	***

Underlined dates denote classes held at University of Illinois (see below).
*** Denotes classes scheduled upon demand.

TUSC/U of I training partnership

The University of Illinois at Urbana-Champaign and TUSC, the recognized leader in corporate Oracle training, invite you to improve your Oracle skills. Come to our new, fully equipped computer labs in downtown Chicago for best-in-class training. TUSC-certified instructors following the world-renowned curriculum developed by TUSC lead these non-credit workshops. The workshops are designed to give students the opportunity to learn the skills necessary to become an Oracle Certified Professional. All courses will be taught at the Illini Center, located in the Chicago Loop, across the street from the Sears Tower.