Inside MySQL 5.0

A DBA's Perspective

A MySQL® Business White Paper

July 2005

Table of Contents

Executive Summary

MySQL 5.0 represents a huge leap forward for the world's most popular open source database management system. While MySQL has been the database of choice for managing high-volume web sites and embedded database applications for years, version 5.0 provides exceptional new functionality that paves the way for larger adoption at the enterprise level. Advancements in the areas of application development, transactional processing, data integrity, and manageability put the MySQL database server on par with proprietary database vendors whose costs are many times more.

This paper first provides a technical overview of MySQL from the DBAs perspective, and then focuses on new features in 5.0 that demonstrate how the latest version of MySQL builds upon an already award-winning database engine to provide more industrial-strength capabilities that help to manage the needs of demanding transactional processing systems, large data warehouse applications, and high-traffic web sites.

Overview of the MySQL Database Server

Countless corporations, educational facilities, and government agencies rely on MySQL to manage their critical data assets and service anywhere from hundreds to billions of requests per day. With over six million active installations worldwide and 40,000 downloads occurring every day, the popularity and prominence of MySQL continue to rise.

Besides the distinction of being open source, what unique advantages does MySQL have over other proprietary database vendors? The following sections describe the architecture of the MySQL database server and highlight features and functions that help make MySQL the most popular open source database on the market.

Architecture Overview

The three priorities of the MySQL database server are:

- **Reliability** – Little or no intervention required for constant uptime to be achieved.
- **Performance** – High performance and fast response times for all database interactions.
- **Ease-of-Use** – Installation and configuration of MySQL should take no more than 15 minutes (most times is much less). Very little management necessary for ongoing maintenance or traditional administration tasks.

To accomplish these objectives, MySQL uses the unique Pluggable Storage Engine Architecture as shown in Figure 1.

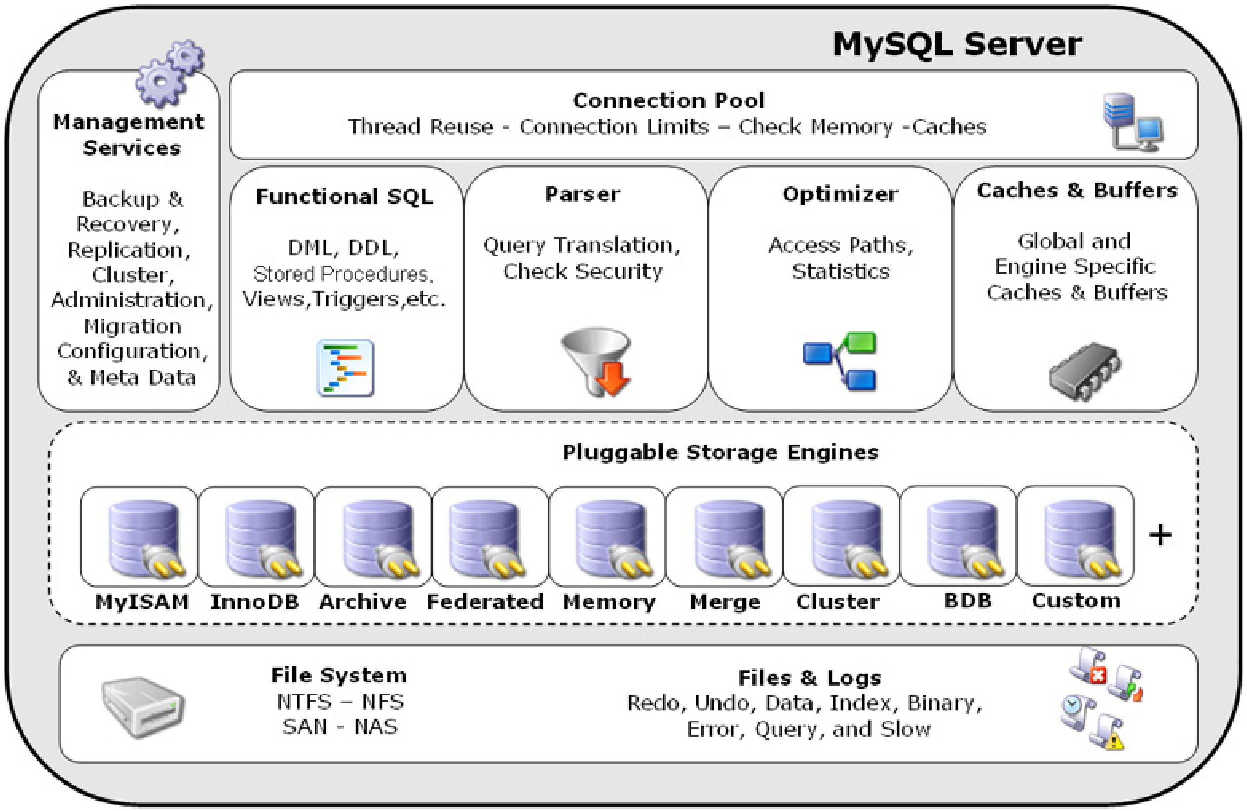

MySQL Server
Management Services
Backup & Recovery, Replication, Cluster, Administration, Migration Configuration, & Meta Data
Connection Pool
Thread Reuse - Connection Limits – Check Memory -Caches
Functional SQL
DML, DDL, Stored Procedures, Views,Triggers,etc.
Parser
Query Translation, Check Security
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
MyISAM
InnoDB
Archive
Federated
Memory
Merge
Cluster
BDB
Custom
+
File System
NTFS – NFS
SAN - NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query, and Slow

Figure 1 – MySQL's Multiple Storage Engine Architecture

Benefits of Scale-out with MySQL

Highlights regarding the MySQL database server architecture include:

- **Scalable Design** – MySQL is ideally suited to run on large 32 and 64 bit Intel, AMD, and PowerPC server environments that handle large OLTP and terabyte-sized data warehouse applications.

- **Simple Install** – Typical Linux and UNIX installations complete in less than ten minutes, and for the Windows platform, a graphical installation and configuration setup makes MySQL easier to install than SQL Server.

- **Unique Storage Engine Architecture** – MySQL offers a number of unique "storage engines" that are designed to meet specific application needs. For example, MySQL offers the InnoDB storage engine, which is tailor-made for heavy-duty transaction processing environments and the MyISAM storage engine, which contains unique features and capabilities to support high-traffic web sites as well as terabyte-sized data warehouses. Additional storage engines ship with MySQL that support other specialized functions. And because MySQL is open source, customers can create their own storage engines to handle their own unique requirements, such as querying legacy data from within the MySQL server.

- **Strong Security Model** – MySQL provides exceptional security features that include a robust authentication system capable of accepting/denying connections down to the client machine level if desired, a very granular user/object privilege system, support for views, and several methods of data encryption and decryption.

- **Always Available / Scale-Out Capability** – MySQL fully supports the cost-effective and highly available scale-out architectures that are the foundation of today's most high-volume web sites and application systems. Supported high availability models in MySQL include master-slave replication, main-memory Cluster configurations with instant failover, and third-party cluster software solutions. Whether implemented in a standalone server or in a scale-out architecture, MySQL is capable of servicing thousands of concurrent user requests though its robust multi-threaded database engine design.

- **High Performance Memory Caches** – MySQL is famous for providing rapid response times for even the most demanding application environments. A number of highly configurable memory caches help keep necessary data always available in RAM, with a unique memory cache only available in the MySQL database server – the Query Cache – keeping a combination of user queries and requested data together so that repetitive requests issued to MySQL receive near instant response times. In addition, many of the memory caches can be dynamically resized or removed and do not require the stopping/starting of the server.

As was mentioned above, the highly configurable nature of MySQL's storage engines allows database professionals to tailor the database server specifically for an intended application purpose. Of course, all storage engines may be used together in a database if desired, and it is very easy to mix-n-match engines depending on the particular application need. Three of the most common configurations are demanding online transaction processing applications, high-traffic Web sites, and large-scale data warehouses.

Robust Transactional Capabilities

MySQL offers one of the most powerful transactional database engines on the market today. The included InnoDB storage engine delivers the following features for handling high-volume OLTP work:

- **ACID Transactions** – MySQL supports ACID (Atomic, Consistent, Isolated, Durable) transaction model that ensures the integrity and validity of every entered transaction. Full COMMIT and ROLLBACK, with redo logs and undo (rollback) segment areas to ensure data consistency during crash recovery operations.

- **Server-Based Data Integrity** – Invalid data (bad dates, invalid numbers, etc.) can be automatically rejected at the server, with column-level rules enforcement being possible. In addition, full foreign-key support is provided so that complete data referential integrity is guaranteed.

- **Low Contention Issues** – Unlimited row-level locking is provided within InnoDB as is automatic deadlock detection, custom isolation levels (READ COMMITTED, etc.), to virtually eliminate locking issues. In addition, InnoDB supports multi-version concurrency control so that those reading data see a snapshot of the requested information as it existed at the moment their query was issued. This ability ensures that readers don't block writers and vice-versa, and leads to extremely high levels of data availability.

- **Automatic Storage Management** – MySQL's no-hassle storage management system automatically expands database storage as needed to accommodate incoming data loads so out-of-space errors become a thing of the past. In addition, the InnoDB storage engine automatically compacts database objects that have experienced heavy amounts of data removal, which (a) returns space back to the database for objects that need it and (b) eliminates the problem of scanning empty database object pages, which helps reduce table scan times. Finally, terabyte-sized storage levels are no problem as a single InnoDB storage structure can accommodate up to 64TB.

- **High-Performance Indexing** – MySQL's InnoDB transaction engine supports the standard B-Tree indexing as well as clustered indexes, which closely resemble that of an Oracle index-organized and SQL Server clustered table.

- **Online Backup and Point-in-Time Recovery** – All MySQL databases can be backed up with standard utilities and, through the use of binary logging, be restored to specific points in time. InnoDB offers a specialized hot backup capability that allows for online backups while normal data activity continues to occur.

- **Distributed Transactions (XA)** – MySQL fully supports XA transactions across multiple MySQL servers, so distributed transactional activity can easily be carried out.

Data Warehouse / Web Features

MySQL has been the de-facto database of choice for extremely high-traffic web sites such as Yahoo, Google, Friendster, and others. Others, like Los Alamos Lab, have used MySQL to handle terabyte-sized data warehouses. For both Web and business intelligence-based applications, the MySQL MyISAM storage engine is routinely used as its design perfectly matches the requirements for these types of systems. Some of the features that make MySQL so popular for such applications include:

- **Scale-Out Architecture** – Many customers use MySQL in a scale-out architecture where many MySQL databases are replicated across inexpensive, commodity hardware such as Dell or HP Linux servers. Such scale-out implementations provide near limitless flexibility to add horsepower as needed to high-traffic web sites or data warehouses, with cost savings that approach 80-90% of scale-up, proprietary architectures.

- **Large Database Support** – MySQL has no practical limit in terms of storage capabilities for large data warehouses, so corporations can build terabyte-sized databases with confidence. In addition, specialized applications such as geospatial are supported within the MySQL server.

- **High-Performance Framework** – MySQL provides a strong feature set for maximizing performance in Web or data warehousing environments. Because fast loading of data into such environments is important, MySQL offers a high-speed load utility, a unique multi-insert DML capability that dramatically reduces load time for multiple insert statements, and special memory caches that buffer incoming data for maximum performance. For reading and retrieving data, features like full text search, and customizable index caches enable MySQL to process enormous volumes of queries. As an example, Friendster handles over one billion queries per day with MySQL on their very popular Web site[1].

[1] http://www.mysql.com/why-mysql/case-studies/mysql-friendster-casestudy.pdf.

With MySQL, customers across all industries are finding they can easily handle nearly every type of database workload, with performance and scalability outpacing every other open source rival. As Los Alamos lab (who uses MySQL to manage their terabyte data warehouse) said, "We chose MySQL over PostgreSQL primarily because it scales better and has embedded replication."[2].

New Features in 5.0

MySQL offers a number of very exciting new features in version 5.0 that directly target the needs of many Enterprise customers. Enhancements in the following areas will be covered in detail in the sections that follow:

- Application Development and Management
- Transaction Processing
- Data Integrity and Protection
- High Availability
- Storage Management
- Manageability
- MySQL Network

Application Development and Management

One of the most exciting new features in MySQL 5.0 is support for stored procedures. Again, while there are some camps that prefer not to use stored procedures for business logic encapsulation in the database, most DBA's are very used to working with them and actually prefer their use for a number of reasons:

Stored Procedures

- **Code Reusability and Change Control** – Rather than embedding the same business logic in multiple applications or business intelligence functions, a stored procedure can be written once and accessed from many applications at once. DBA's can then properly manage such business logic via standard change management functions that are used to handle the versioning of other database assets like schema structures and security privileges.

- **Fast Performance** – DBA's like stored procedures because they can harness the set processing power of databases inside looping constructs and the like, which most times always beats application logic that works a row at a time. DBA's also prefer stored procedure executions to ad-hoc query requests because a DBA can tune a specific stored procedure query to run very efficiently and then turn it over to many users to run at will. This is oftentimes better than letting inexperience end users code complex SQL queries on their own.

- **Easier Security Administration** – For complex databases serving large user communities, security administration can be a heavy burden on the DBA. Granting the proper security privileges over lots of objects to many individual users can be a very time consuming task. Stored procedures ease the load on the DBA by allowing access control to be handled at the procedure level rather than at the object level. For example, if a user needed to run a

[2] "Open Source", IEEE Software, June/July 2005.

complex query that accessed ten different tables, a DBA would have to grant the user privileges to all ten tables. However, if that same query were instead placed inside a stored procedure, a DBA would only have to grant the user a single privilege to run the procedure.

- **Reduced Network Traffic** – Stored procedures help eliminate excessive network traffic because only a single call to the database server is required, rather than multiple calls.

Stored procedures in MySQL 5.0 are very easy to use and learn for DBAs and application developers alike because they adhere to the ANSI SQL 2003 specification, much like IBM DB2. And they offer certain ease-of-use capabilities over other stored procedure implementations from some proprietary database vendors. For example, MySQL stored procedures offer standard output ability for SELECT statements, which is different than some proprietary database vendors that require specialized package calls or ref cursors to handle SELECT statement output. MySQL is more like Sybase and Microsoft SQL Server in this respect. For example, to see the output from a query in a MySQL 5.0 stored procedure, you only need code the SELECT statement in the procedure and run it as in this example:

```
mysql> delimiter //
mysql> create procedure top_broker()
    -> select  a.broker_id,
    ->                 a.broker_first_name,
    ->                 a.broker_last_name,
    ->                 sum(broker_commission) total_commissions
    ->         from    broker a,
    ->                 client_transaction b
    ->         where   a.broker_id = b.broker_id
    ->         group by a.broker_id,
    ->                 a.broker_first_name,
    ->                 a.broker_last_name
    ->         order by 4 desc;
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call top_broker();
+-----------+-------------------+------------------+-------------------+
| broker_id | broker_first_name | broker_last_name | total_commissions |
+-----------+-------------------+------------------+-------------------+
|        20 | STEVE             | BOYCE            |        3864173.64 |
|         1 | JONATHAN          | MORTON           |        1584621.39 |
|        13 | JIM               | SANDERS          |        1369157.73 |
|         4 | DAVE              | TUCKER           |        1214111.75 |
|        14 | DENISE            | SCHWARTZ         |        1041040.98 |
|         7 | CHARLES           | MONSOUR          |         961663.56 |
|         9 | LAURA             | HIGHT            |         890148.78 |
|         3 | KARL              | PUGH             |         884260.64 |
+-----------+-------------------+------------------+-------------------+
8 rows in set (0.00 sec)
```

In addition to handling standard output for SELECT statements, MySQL 5.0 stored procedures support many of the common development constructs such as:

- In/out parameters
- Variable declaration
- Looping/WHILE statements with exits
- Conditional logic (IF, CASE, etc.)
- Condition handlers
- Procedures calling procedures
- COMMIT/ROLLBACK functionality for transactional tables
- DDL statements
- And more…

From a DBA standpoint, MySQL 5.0 stored procedures are managed as any other procedural code implementation in a database. The standard CREATE, ALTER, DROP syntax is used for addition, change, or removal, and GRANT is used to allow users to create or execute stored procedures.

To obtain metadata for a particular procedure, a DBA can either:

- Use the SHOW PROCEDURE STATUS function.
- Query the mysql.proc internal table.
- Use the new 5.0 INFORMATION_SCHEMA data dictionary (covered in more detail later in this paper).

The full create DDL for a procedure can be easily obtained by using the SHOW CREATE PROCEDURE <procedure name> function.

Functions

Functions differ from stored procedures in that a function's purpose is to return a value or information to a calling routine or query. Functions are now supported in MySQL 5.0, so DBA's and developers can create functions that perform certain tasks and return data back to a calling program.

Like stored procedures, functions allow business logic or processing to be embedded once in the database and reused afterwards. Many of the same benefits of stored procedures apply to functions. An example of a function creation and use would be:

```
mysql> delimiter //
mysql> create function broker_name (id int)
    -> returns varchar(50)
    -> deterministic
    -> begin
    -> declare broker_name varchar(50);
    -> select concat(broker_first_name,' ',broker_last_name)
    -> into broker_name
    -> from broker
    -> where broker_id = id;
    -> return broker_name;
    -> end
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select broker_name(1);
+-----------------+
| broker_name(1)  |
+-----------------+
| JONATHAN MORTON |
+-----------------+
1 row in set (0.00 sec)
```

Triggers

MySQL 5.0 triggers allow actions to occur before or after DML has been executed upon one or more rows in a table. There are many reasons a DBA or developer would code triggers on a table, with the most popular business use cases being data auditing and other security-related functions such as on-the-fly data encryption.

For example, a customer database may include customer social security numbers that the corporate security/auditing group mandates must be encrypted to disk. A DBA could create a trigger that automatically intercepts and encrypts data as it's inserted into the table as so:

```
mysql> delimiter //
mysql> create trigger t_customer_insert before insert on customer
    -> for each row
    -> begin
    ->  set NEW.customer_ssn = aes_encrypt(NEW.customer_ssn,'password');
    -> end;
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;

mysql> insert into customer values (1,'fred','smith','456097234');
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+-------------+---------------------+--------------------+------------------+
| customer_id | customer_first_name | customer_last_name | customer_ssn     |
+-------------+---------------------+--------------------+------------------+
|           1 | fred                | smith              | ☼Q▌─U¶¢ƒ╟▒4╨☺öm[ |
+-------------+---------------------+--------------------+------------------+
```

Support for cursors has been introduced in MySQL 5.0, with cursors being forward scrolling and non-updateable in their first release. Cursors may be used in any of MySQL 5.0's new code objects such as stored procedures and triggers, or they may be utilized in standalone procedural logic. An example of a simple cursor in a stored procedure might be:

```
mysql> delimiter //
mysql> CREATE PROCEDURE cursor_demo()
    -> BEGIN
    ->   DECLARE a, b CHAR(16);
    ->   DECLARE cur1 CURSOR FOR SELECT id,data FROM test.t1;
    ->   OPEN cur1;
    ->   REPEAT
    ->     FETCH cur1 INTO a, b;
    ->   UNTIL DONE END REPEAT;
    ->   CLOSE cur1;
    -> END
    -> //
```

MySQL Client Driver Enhancements

MySQL offers a number of front end connectors that can be used to develop applications against MySQL databases. Whether an application is targeting PHP, Java, .NET, Perl, or needing generic ODBC connection capabilities, MySQL has a driver that will fit the bill.

With the MySQL 5.0 release, new capabilities have been introduced in nearly every MySQL driver which helps increase performance and security. For example, Java applications can exploit new driver features that greatly improve performance through limiting the amount of data returned to a client application, and selectively requested more information as needed. On the security front, .NET applications can now make use of SSL when connecting to a MySQL database.

Distributed Transaction Processing

Oftentimes a single transaction needs to span multiple database servers. Such a situation can now be handled in MySQL 5.0 as full XA support has been implemented. Using MySQL's transactional storage engine – InnoDB – a DBA can now allow for distributed transactions that utilize a two-phase commit protocol to ensure transactional integrity.

Improved Data Integrity and Protection

MySQL 5.0 provides a number of new features that assist in securing and validating critical information stored in MySQL databases.

Views

Views, as their name implies, provide a window into a single table, or a join of multiple tables. Views have been widely used by DBAs and developers alike for two main reasons:

- **Security Administration** – Views help protect sensitive data through the hiding of one or more sensitive columns in a table, through limiting the actual rows returned in a table or table join, or through decrypting data that has been encrypted on disk.
- **Performance Tuning** – Views can be used by DBAs to help the performance of a database by pre-defining a tuned join condition of multiple tables or through limiting the amount of data returned in large tables

MySQL 5.0 provides support for views, with updates being possible through views in certain situations. An example of a view that decrypts sensitive data, such as a customer's social security number, that has been encrypted to disk (to thwart hackers and from backup files being accessed in an unauthorized manner) would be:

```
mysql> select * from customer;
+-------------+---------------------+--------------------+------------------+
| customer_id | customer_first_name | customer_last_name | customer_ssn     |
+-------------+---------------------+--------------------+------------------+
|           1 | fred                | smith              | ☼Q▌─U¶¢ƒ╟▒4╨☺öm[ |
+-------------+---------------------+--------------------+------------------+
1 row in set (0.03 sec)


mysql> create view v_customer as
    -> select customer_id, customer_first_name, customer_last_name,
    -> aes_decrypt(customer_ssn,'password') as customer_ssn
    -> from customer;
Query OK, 0 rows affected (0.03 sec)


mysql> select * from v_customer;
+-------------+---------------------+--------------------+--------------+
| customer_id | customer_first_name | customer_last_name | customer_ssn |
+-------------+---------------------+--------------------+--------------+
|           1 | fred                | smith              | 456097234    |
+-------------+---------------------+--------------------+--------------+
```

Server Enforced Data Integrity

Prior to MySQL 5.0, invalid data being sent to a MySQL database was denied at the server only through the use of column default values. In version 5.0 of MySQL, a new strict mode option is available that rejects invalid data outright and prevents any data misrepresentations from finding their way into the server. The sql_mode configuration parameter contains a variety of flexible options so the DBA can precisely control the way in which data integrity is handled by MySQL.

For example, a DBA may only want to have data rejected and rolled back in MySQL's transactional engine (InnoDB), in which case they might perform the following:

```
mysql> create table test (c1 int) engine=innodb;
Query OK, 0 rows affected (0.22 sec)

mysql> insert into test values ('test');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> show warnings;
+---------+------+------------------------------------------------------+
| Level   | Code | Message                                              |
+---------+------+------------------------------------------------------+
| Warning | 1264 | Out of range value adjusted for column 'c1' at row 1 |
+---------+------+------------------------------------------------------+
1 row in set (0.00 sec)

mysql> select * from test;
+------+
| c1   |
+------+
|    0 |
+------+
1 row in set (0.03 sec)

mysql> set sql_mode='strict_trans_tables';
Query OK, 0 rows affected (0.00 sec)

mysql> insert into test values ('test');
ERROR 1264 (22003): Out of range value adjusted for column 'c1' at row 1
```

Precision Math

MySQL 5.0 includes support for customers desiring very granular and exact precision for mathematical calculations, such as those used in financial modeling. Fifty-six digit precision is possible, with all calculations being performed in an extremely fast manner through static memory allocation.

High Availability

MySQL is used by countless customers to provide maximum uptime for critical business systems. In particular, MySQL Cluster is especially utilized by telecom customers to supply five nine's availability as well as rapid response times for quick SQL searches. In MySQL 5.0, a number of improvements have been implemented in MySQL Cluster that help extend it's already powerful feature set.

First, MySQL Cluster can now make use of the Query Cache, which is a specialized memory cache that stores repetitively issued SQL queries as well as their corresponding result set. This addition will greatly increase the performance of MySQL Cluster when identical queries are sent to the server as nearly all logical and physical I/O are eliminated.

Next, improvements in query processing at the cluster level greatly reduce the amount of data sent between the cluster nodes and the MySQL servers that act as SQL interfaces to the cluster engine. Queries that sample large data volumes will particularly benefit from this enhancement.

Storage Management Enhancements

MySQL 5.0 includes the following new features that help conserve and manage storage usage on database servers.

More Efficient Storage for InnoDB Transactional Engine

MySQL's InnoDB transactional storage engine has been enhanced in version 5.0 to better optimize space on a database server by utilizing an improved row storage format. Simply migrating a 3.x or 4.x InnoDB database to 5.0 can save up to 15-20% in total storage usage.

Archive Engine

Government compliance mandates such as Sarbanes Oxley, HIPAA, and others have forced many corporations to retain huge volumes of historical information on their database servers. This being the case, companies are looking for ways to reduce the amount of space this type of seldom-referenced data consumes.

To ease this burden, version 5.0 introduces a new Archive storage engine that automatically compresses data stored in MySQL tables. The MyISAM storage engine has had a compression option for years, however once data is compressed, it is read only. In addition, a DBA has to utilize a command line utility to compress/uncompress data.

The new Archive engine does not carry the compressed MyISAM table restrictions and allows both INSERT and SELECT on Archive tables. This is especially relevant for database security professionals who need to store audit data which should never be tampered with or selectively updated or deleted.

In addition, a table can be created as Archive, which alleviates having to use a command line utility to compress data. Finally, any MySQL table can be switched to an Archive table through the use of a simple ALTER TABLE command.

How much space can be saved by using the new Archive storage engine? As an example, an identically designed table was created as a standard MyISAM, a MyISAM compressed, InnoDB, and an Archive table, and then loaded with 1.6 million rows with the following result:

```
mysql> SELECT  table_name table_name,
    ->         engine,
    ->         ROUND(data_length/1024/1024,2) total_size_mb
    -> FROM    information_schema.tables
    -> WHERE   table_schema = 'gim' and
    ->         table_name like 't_%'
    -> ORDER BY 3;
+------------+---------+---------------+
| table_name | engine  | total_size_mb |
+------------+---------+---------------+
| t_archive  | ARCHIVE |         21.62 |
| t_comp     | MyISAM  |         28.00 |
| t_myisam   | MyISAM  |        112.84 |
| t_innodb   | InnoDB  |        148.69 |
+------------+---------+---------------+
```

The Archive table is 85% smaller than the InnoDB table and 81% smaller than a standard MyISAM table, which demonstrates a very powerful storage ROI for the Archive storage engine.

Another great benefit of the Archive engine is it is capable of very fast load times (even faster than MyISAM in many cases).

Manageability Advancements

MySQL 5.0 offers a wide array of new options to help busy DBAs manage their growing database server farm.

Instance Manager

Because MySQL has become so popular in nearly every area of data management, DBAs have to manage and control more MySQL databases than ever before. To more easily administer MySQL servers, MySQL 5.0 introduces Instance Manager. Instance Manager provides DBAs with a powerful utility to control and view many MySQL servers from one location.

With Instance Manager, a DBA can:

- View the up/down status of all MySQL instances on a server.
- Remotely start/stop any MySQL instance on a Linux or Unix server.
- Remotely edit any `my.cnf` file on any Unix/Linux server. You can add, change, and remove any `my.cnf` file entry and have it saved on the remote server.
- Read remote log files (error, query, etc.) on any MySQL Server. You can choose to view an entire log, only the last 50 lines, or whatever is desired.

With its current capabilities, Instance Manager can greatly ease the management burden of DBAs that are charged with managing an ever growing database farm by minimizing the time and effort involved in handling many of the traditional DBA tasks and troubleshooting exercises.

Federated Tables

Given the fact that data continues to be spread across the Enterprise, DBAs oftentimes need to reference remote database objects that exist on other database servers from a different source server. This technique makes several distinct physical databases appear as one logical database to the end user community and simplifies overall data access.

MySQL 5.0 brings federated table support to the table so DBAs can link together objects that exist on different MySQL database servers to create one or more logical databases. All storage engines can be the target of a federated table definition and complete support for SELECT, INSERT, UPDATE, and DELETE functions exists.

For example, supposed the following table exists on a Linux server:

```
mysql> desc client;
+-------------------------+-------------+------+-----+---------+-------+
| Field                   | Type        | Null | Key | Default | Extra |
+-------------------------+-------------+------+-----+---------+-------+
| client_id               | int(11)     | NO   | PRI |         |       |
| client_first_name       | varchar(20) | NO   |     |         |       |
| client_last_name        | varchar(40) | NO   |     |         |       |
| client_gender           | char(1)     | NO   |     |         |       |
| client_year_of_birth    | int(4)      | NO   |     |         |       |
| client_marital_status   | varchar(20) | YES  |     | NULL    |       |
| client_street_address   | varchar(40) | NO   |     |         |       |
| client_postal_code      | varchar(10) | NO   |     |         |       |
| client_city             | varchar(30) | NO   |     |         |       |
| client_state_province   | varchar(40) | NO   |     |         |       |
| client_phone_number     | varchar(25) | NO   |     |         |       |
| client_household_income | bigint(30)  | YES  |     | NULL    |       |
| client_country          | varchar(40) | YES  |     | NULL    |       |
| broker_id               | int(11)     | NO   | MUL |         |       |
+-------------------------+-------------+------+-----+---------+-------+
```

If a DBA wishes to reference the table from another MySQL server that runs on Windows, they would simply create a federate table on their Windows server:

```
mysql> CREATE TABLE client (
    ->   client_id int(11) NOT NULL,
    ->   client_first_name varchar(20) NOT NULL,
    ->   client_last_name varchar(40) NOT NULL,
    ->   client_gender char(1) NOT NULL,
    ->   client_year_of_birth int(4) NOT NULL,
    ->   client_marital_status varchar(20) default NULL,
    ->   client_street_address varchar(40) NOT NULL,
    ->   client_postal_code varchar(10) NOT NULL,
    ->   client_city varchar(30) NOT NULL,
    ->   client_state_province varchar(40) NOT NULL,
    ->   client_phone_number varchar(25) NOT NULL,
    ->   client_household_income bigint(30) default NULL,
    ->   client_country varchar(40) default NULL,
    ->   broker_id int(11) NOT NULL) ENGINE=FEDERATED
    -> COMMENT = 'mysql://myuser:password@192.164.3.2:3306/gim/client';
Query OK, 0 rows affected (0.20 sec)

mysql> select count(*) from client;
+----------+
| count(*) |
+----------+
|     5600 |
+----------+
1 row in set (0.50 sec)
```

Data Dictionary

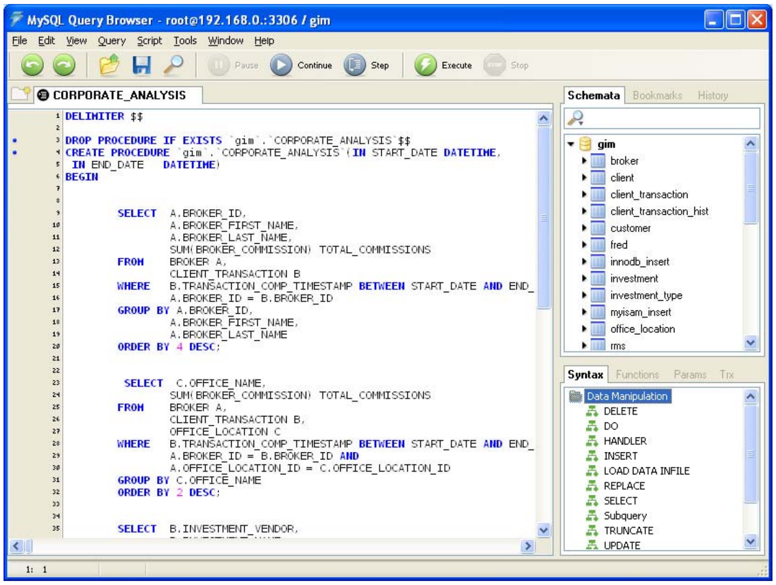
Because the use of metadata (data about data) has grown so popular, MySQL 5.0 has introduced a new database called `information_schema` that serves as a central data dictionary for all objects and other database-related items (such as security) defined on a MySQL server. DBAs and developers alike can use the `information_schema` database to extract metadata concerning various aspects of one or more databases that exist on a server.

For example, if a DBA wants to obtain a quick overview of the space situation that exists on a particular MySQL instance, she could issue the following query:

```
mysql> SELECT   a.schema_name db_name,
    ->  IFNULL(ROUND((SUM(b.data_length)+SUM(b.index_length))/1024/1024,2),0.00)
    ->  total_size_mb,
    ->  IFNULL(ROUND(((SUM(b.data_length)+SUM(b.index_length))-
    ->  SUM(b.data_free))/1024/1024,2),0.00) data_used_mb,
    ->  IFNULL(ROUND(SUM(data_free)/1024/1024,2),0.00) data_free_mb,
    ->  IFNULL(ROUND((((SUM(b.data_length)+SUM(b.index_length))-SUM(b.data_free))/
    ->  ((SUM(b.data_length)+SUM(b.index_length)))*100),2),0) pct_used,
    ->  COUNT(table_name) tables
    -> FROM     information_schema.schemata a
    ->  LEFT JOIN information_schema.tables b ON a.schema_name = b.table_schema
    -> WHERE    a.schema_name != 'information_schema'
    -> GROUP    BY a.schema_name
    -> ORDER    BY 1;
+---------+---------------+--------------+--------------+----------+--------+
| db_name | total_size_mb | data_used_mb | data_free_mb | pct_used | tables |
+---------+---------------+--------------+--------------+----------+--------+
| gim     |        432.67 |       432.67 |         0.00 |   100.00 |     16 |
| gim2    |          8.64 |         8.64 |         0.00 |   100.00 |      6 |
| mysql   |          0.33 |         0.33 |         0.00 |    99.69 |     18 |
| test    |          0.00 |         0.00 |         0.00 |     0.00 |      0 |
| tpcc    |        126.09 |       126.09 |         0.00 |   100.00 |      9 |
+---------+---------------+--------------+--------------+----------+--------+
```

Enhanced Graphical Management Tools

A database engine's ease of use is often greatly enhanced through a good set of database tools. Good DBA and development tools help reduce the amount of manual and error-prone processes that DBAs and developers use to handle database tasks. MySQL 5.0 offers an enhanced set of graphical management tools that run on Windows, Linux, and Mac. These tools can help jump-start those new to MySQL as well as others who have used MySQL for years. In addition, more third party software vendors have declared support for MySQL.



```
DELIMITER $$

DROP PROCEDURE IF EXISTS `gim`.`CORPORATE_ANALYSIS`$$
CREATE PROCEDURE `gim`.`CORPORATE_ANALYSIS`(IN START_DATE DATETIME,
 IN END_DATE   DATETIME)
BEGIN


        SELECT   A.BROKER_ID,
                 A.BROKER_FIRST_NAME,
                 A.BROKER_LAST_NAME,
                 SUM(BROKER_COMMISSION) TOTAL_COMMISSIONS
        FROM     BROKER A,
                 CLIENT_TRANSACTION B
        WHERE    B.TRANSACTION_COMP_TIMESTAMP BETWEEN START_DATE AND END_
                 A.BROKER_ID = B.BROKER_ID
        GROUP BY A.BROKER_ID,
                 A.BROKER_FIRST_NAME,
                 A.BROKER_LAST_NAME
        ORDER BY 4 DESC;


         SELECT  C.OFFICE_NAME,
                 SUM(BROKER_COMMISSION) TOTAL_COMMISSIONS
        FROM     BROKER A,
                 CLIENT_TRANSACTION B,
                 OFFICE_LOCATION C
        WHERE    B.TRANSACTION_COMP_TIMESTAMP BETWEEN START_DATE AND END_
                 A.BROKER_ID = B.BROKER_ID AND
                 A.OFFICE_LOCATION_ID = C.OFFICE_LOCATION_ID
        GROUP BY C.OFFICE_NAME
        ORDER BY 2 DESC;


        SELECT   B.INVESTMENT_VENDOR,
```

The MySQL Query Browser has been updated to include support for graphically building and editing stored procedures and views. In addition, new features and functions have been added to increase the speed at which queries and other database code can be developed.

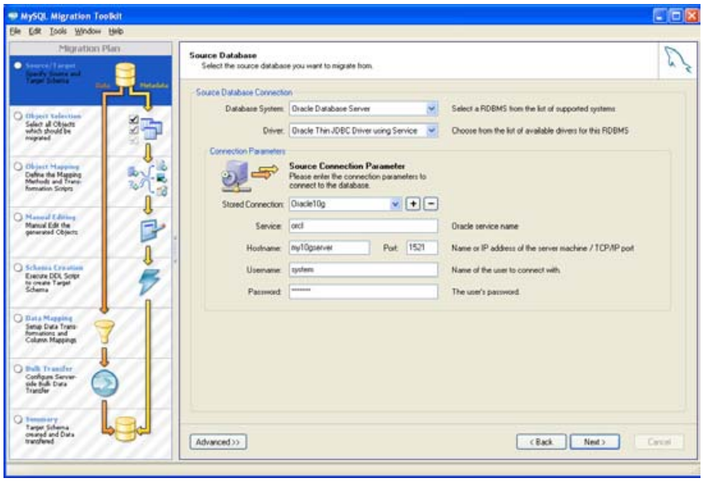
The MySQL Administrator tool has been enhanced with many new editors and functions that allow a DBA to easily manage any MySQL server. A DBA can create and edit tables of all different storage engine types, manage security, run maintenance tasks on all MySQL objects, and much more.

For Windows users, a new systray management utility has been added that allows a DBA to quickly start/stop any MySQL service on Windows from the systray area. In addition, the new systray utility allows the quick editing of database configuration files.

New Database Migration Tool

Migrating to MySQL has never been easier now that the new Migration Toolkit has been released. With the Migration Toolkit, DBAs and developers can easily migrate their existing Oracle, SQL Server, and Microsoft Access databases to MySQL. Any database engine that supports JDBC can also be migrated.

An intuitive graphical interface guides the DBA or developer in selecting what databases, schemas, and objects to migrate to MySQL. The Migration Toolkit converts the DDL syntax from any source database to MySQL with complete customization being permitted so that all datatypes and storage engines exactly match what is desired. Data can be ported over to MySQL as well with the movement of millions of rows of data being possible.


MySQL Migration Toolkit
File Edit Tools Window Help
Migration Plan
Source/Target
Specify Source and Target Schema
Data
Metadata
Object Selection
Select all Objects which should be migrated
Object Mapping
Define the Mapping Methods and Transformation Scripts
Manual Editing
Manual Edit the generated Objects
Schema Creation
Execute DDL Script to create Target Schema
Data Mapping
Setup Data Transformations and Column Mappings
Bulk Transfer
Configure Server-side Bulk Data Transfer
Summary
Target Schema created and Data transfered
Advanced >>
< Back
Next >
Cancel

Source Database
Select the source database you want to migrate from.

Source Database Connection

Database System: Oracle Database Server — Select a RDBMS from the list of supported systems
Driver: Oracle Thin JDBC Driver using Service — Choose from the list of available drivers for this RDBMS

Connection Parameters

Source Connection Parameter
Please enter the connection parameters to connect to the database.

Stored Connection: Oracle10g
Service: orcl — Oracle service name
Hostname: my10gserver Port: 1521 — Name or IP address of the server machine / TCP/IP port
Username: system — Name of the user to connect with.
Password: ******** — The user's password.

In addition to its graphical interface, the Migration Toolkit offers a command line scripting option that allows technical heavyweights to customize many aspects of their migration operations.

Third Party Software Support

The eco-system for MySQL continues to grow every day with new vendors being constantly added to the list of MySQL partners. Seeing that their customers can enjoy enormous database license cost savings, Dell, HP, and Novell now resell MySQL Network.

Other software vendors such as Quest and Embarcadero Technologies offer database management tools that help MySQL developers and DBAs in the areas of data modeling, administration, SQL development, job scheduling, benchmarking, and performance monitoring. And for enhanced replication, GoldenGate provides exceptional data replication abilities for MySQL that allow database tiering and other like architectures to be deployed.

MySQL Network

MySQL Network delivers the enterprise support and service corporate customers are used to getting with proprietary software. MySQL Network offers exceptional benefits and a fast jump-start for those wishing to broadly implement MySQL across their corporate IT environment.

The major benefits of MySQL Network include:

- Certified software that has undergone additional extensive testing and validation at MySQL testing facilities.
- Full 24 x 7 production support that can be tailored to meet the needs of the most demanding IT organization.
- A specialized Knowledge Base, which is indispensable for helping jump-start any MySQL effort or train an IT staff in MySQL details.
- Specialized Advisors, which can be customized to alert and inform an organization of changes that apply to their own special environment.
- Complete indemnification, which makes legal worries a thing of the past.

With MySQL Network, IT professionals can proceed with confidence in deploying MySQL throughout their environment. Enterprise customers get the same level of protection and support as they expect from proprietary database vendors.

MySQL Network reduces database TCO (Total Cost of Ownership) by 90% by reducing license fees, annual maintenance fees, and annual support costs. For more information about MySQL Network, please visit http://www.mysql.com/network/.

Conclusion

Whether it's handling over one billion queries a day for a hugely successful Web site or processing tens of thousands of transactions per minute in a high-volume retail store chain, MySQL is demonstrating every day that it is more than capable of serving as the database backbone for all types of corporate, government, and educational organizations.

About MySQL

MySQL AB develops and supports a family of high performance, affordable database products — including MySQL Network, a comprehensive set of certified software and premium support services. The company's flagship product is MySQL, the world's most popular open source database, with more than 6 million active installations. Many of the world's largest organizations, including Yahoo!, Sabre Holdings, The Associated Press, Suzuki and NASA are realizing significant cost savings by using MySQL to power high-volume Web sites, business-critical enterprise applications and packaged software.

With headquarters in Sweden and the United States — and operations around the world — MySQL AB supports both open source values and corporate customers' needs in a profitable, sustainable business. For more information about MySQL, please visit www.mysql.com.