Enterprise Data Warehousing with MySQL

Table of Contents

1 Introduction

Modern enterprises that stay ahead of the game know the critical importance of having the right information at their disposal to help make key decisions that impact the direction their business moves. This being the case, it is no surprise that a 2006 Gartner Group survey found that CIO's identified Business Intelligence (BI) as their number two technology priority, up from number ten just three years ago.[1] This emphasis was echoed by a 2007 survey performed by InformationWeek that found nearly half of IT executives plan to increase their spending on BI above 2006 levels.[2]

These trends clearly indicate that smart businesses recognize that their ability to compete greatly depends on their intelligent use of technology and information whether the aim is to help their customers communicate, socialize, share and locate information, entertain, or shop for goods and services in a more efficient manner.

However, the increasing thirst for more and better information to help make decisions has brought with it a number of interesting challenges. First the sheer volume of information being managed in corporate data warehouses is fast becoming a key issue. Commenting on a 2006 survey conducted by his company, TDWI senior manager of research Philip Russom says, "Data warehouses will experience on average at least 33 percent annual growth in data volumes, 50 percent or more when subject to aggressive collections of customer, supply chain, eCommerce or compliance data."[3] That same survey found that 36% of data warehouses were multi-terabyte in 2006 and that 48% would be so by the end of 2007.

Second, the increasing appetite for business intelligence information has spawned specialized data warehousing "workload types" (different patterns of data growth and usage) that cannot be reconciled or well managed in a single analytic data store. More will be said on this issue later.

Third and finally, the cost of implementing and managing complex business intelligence infrastructures throughout a growing and demanding organization is steadily moving upward. In the same InformationWeek study quoted above, 39% of those polled complained that expensive software licenses prohibited them from rolling out the data warehousing/business intelligence initiatives they would like.

To combat these issues and meet the goal of delivering scalable and fast-responding data warehousing systems, modern businesses are turning to open-source solutions to satisfy their needs. Open source software has proven itself in the online world and is moving steadily into enterprise software installations. For example, a Gartner Group study predicts that 70% of all IT organizations will use open source databases by the end of 2008.[4] This being the case, it was natural for open source technology to expand into the area of data warehousing and business intelligence.

Although the MySQL database server has been the proven leader in database management for online businesses, many have wondered if it has the capabilities to also lead in the area of data warehousing and business intelligence. This paper examines the MySQL strategy for data warehousing and showcases the unique abilities and advantages that come with choosing MySQL as a provider for data warehousing and business intelligence needs.

---

[1] http://www.intelligententerprise.com/channels/bi/showArticle.jhtml?articleID=198701576
[2] http://www.informationweek.com/showArticle.jhtml?articleID=198001258
[3] http://www.netezza.com/releases/2006/release061906.htm.
[4] Gartner Group, *Enterprise Databases in an Open Source World*, September 2006.

2 A Look at Data Warehouse Workloads

In their *Magic Quadrant for Data Warehouse Database Management Systems, 2006* study, Gartner Group identified several data warehousing workloads that typified the various use cases found at customer sites who were actively involved in business intelligence. These workloads were:

- Continuous (near-real-time) data loading — similar to an online transaction processing (OLTP) workload
- Large numbers of standard reports ranging in the thousands per day, requiring Structured Query Language (SQL) tuning and index creation
- An increasing number of true ad hoc query users with a random, unpredictable use of the data
- Analytics and BI-oriented functionality in OLTP applications

Translating these workloads into practical data warehousing installations yields the following use cases:

1. Small, semi real-time data marts
2. Continuous, real-time/query data warehousing
3. Traditional, standard reporting warehousing
4. Massive historical, with ad-hoc queries warehousing
5. BI, analytic in OLTP applications (an emerging trend)

Let's next examine each of these use cases in detail so the various critical distinctions can be understood.

2.1 Examining Data Warehouse Use Cases

The first standard data warehouse use case is the **data mart**. Data marts are normally characterized by two things: (1) their size (2) their focus. Data marts tend to be smaller in data volume than data warehouses and they tend to be more narrow in their scope, oftentimes only holding data for a particular area of a company or even a subset of an area in a company. A data mart's update frequency (how often its data is refreshed by daily/hourly transactional activity) depends on the need of the business area using its information to make decisions. If key decisions necessitate that the most up-to-date data as possible be present, real-time feeds into a data mart may be observed. Otherwise, daily or weekly refreshes will be normative.

The **real-time data warehouse** has seen increased popularity in recent years, mainly due to the increased desire to have the most current information as possible at the fingertips to navigate and outsmart the competition. The real-time data warehouse's attributes include constant resource contention between incoming data refreshes and queries being made against the same set of data objects, hourly and daily increases in storage, purge rituals of unneeded data, and an audience that can be either narrow or broad in focus depending on the subject areas that reside in the warehouse.

The **traditional data warehouse**, as its name implies, is the use case most think about when it comes to data warehousing. Usually sporting very large data volumes, having infrequent (defined as not hourly and sometimes not daily) refresh rates, and serving a wide and varied audience, the traditional data warehouse is what most businesses start with (after a data mart) when implementing a data store to use for business intelligence purposes.

The **historical data warehouse** is somewhat new in nature and has been born out of semi-recent mandates that require many businesses to keep large amounts of historical information at the ready for government or other business-compliance purposes. The historical data warehouse typically has data volumes that are multiples of traditional data warehouses, see semi-frequent data refreshes, but usually have less query traffic than data marts, real-time warehouses, or traditional data warehouses.

The **analytic OLTP warehouse** application is what some see as a dangerous return to the days when databases that were used for rapid transactional activity became the target for resource-intensive analytic queries also. The mix of these workloads usually spelled death for applications needing rapid response times to serve customers who demanded quick completions for their requests. The analytic OLTP warehouse is usually a database that back-ends a standard OLTP application, but also contains objects that are fed from the transactional objects and are designed to support business intelligence queries.

There may be other niche data warehouse installations, but the above represent the vast majority of what are found in IT organizations.

2.2 Upcoming Challenges for Data Warehousing

Each particular installation above possesses at least one of the workloads identified by Gartner and sometimes more than one. It is when a data warehouse installation contains a mixture of workloads, Gartner says, that issues begin to arise in terms of both management and performance. Gartner states:

> "The four workload types are creating issues for vendors, more than the actual size of the DW, even manifesting in database sizes less than 1TB. In addition to service-level expectations, the size and duration of "useful" data for each community often differs significantly, forcing every aspect of the DW environment — from input/output (I/O) channel balancing through disk management and into memory and processor allocation — to become involved. During the next three years, mixed workload performance will become the single most important performance issue in data warehousing."[5]

Addressing the mixture of workloads can result in financial cost issues as it oftentimes involves the splitting of data warehouse subject areas between different servers and databases. This is just one reason why IT professionals are now turning to open source solutions to solve their data warehousing needs. But cost is not the only reason for choosing a RDBMS like MySQL for data warehousing. A number of factors come into play that make MySQL an attractive choice in this arena including a strong and successful company history, a solid core feature set that lends itself to data warehousing, a unique architecture and design that solves the problem of mixing workloads in data warehouses, and a growing partner network of business intelligence software and tools.

3 What is MySQL?

MySQL is the leading open source database solution used today to power online enterprise, embedded, and business intelligence applications. For over twelve years, the MySQL database server has been the heart of database systems that serve a growing and intensely demanding customer base. The "M" in the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python), MySQL has been battle-tested by heavy transaction processing applications, terabyte-sized data warehouses, and high-traffic Web sites, and found to be the proven leader in open source database technology. Thousands of well-known companies such as Sabre, Google, Yahoo, NY Times, Cox Communications, The Associated Press, Symantec, Alcatel, Nokia, Nortel, Cisco, Zappos, and others rely on MySQL to manage their data-driven applications.

The same MySQL server that exceeds expectations in these environments is the same database that's also used to manage the information needs of small-medium applications that rely on a bundled database, as well as deeply embedded systems that demand a high-performing and reliable database with a very small footprint. Having proven itself in the bleeding-edge world of technology start-up's, Web 2.0 and other such forward-thinking companies, MySQL is now rapidly becoming the embedded database of choice for

[5] Gartner Group, *Magic Quadrant for Data Warehouse Database Management Systems*, 2006.

OEM's and ISV's who want to take advantage of the open source nature of MySQL, but use it in a commercial way. With its dual-licensing model, MySQL is able to satisfy this desire with a hassle-free solution that provides all the cost benefits and strengths of the open source software model along with the safety-net of services and support that ISVs and OEMs need for their commercial products.

No other open source database comes close to the popularity of the MySQL database, with over 11 million installations existing worldwide and more than 50,000 downloads occurring daily on the MySQL web site. Indeed, a Wall Street Journal study (December 2005) found the MySQL database behind only the Mozilla Firefox Browser in terms of overall open source software downloads (70 million total).

Such popularity is earned by MySQL through delivering on its promises of supplying the vast majority of features needed by database applications at a fraction of the cost. As just one example, Weather.com (the #1 news site on the web) switched from proprietary databases to MySQL and stated that the switch to open source and cheaper hardware resulted in "30 percent increased capacity and 50 percent decreased cost" according to a 2007 article in CIO magazine.[6]

4 MySQL's Data Warehousing Strategy

MySQL's data warehousing strategy can be simply stated as delivering on these three goals:

- Support the most common data warehousing use cases
- Partner with major business intelligence and data movement/integration vendors
- Offer a highly attractive total cost of ownership for data warehousing installations

Each of these objectives will now be explored in more detail in the sections that follow.

4.1 The Technical Blueprint behind MySQL and Data Warehousing

The technical attributes of MySQL's data warehousing strategy include four things. First, a commitment to an open source methodology that helps maximize innovation. Second, a solid RDBMS feature set that lends itself to data warehousing. Third, a unique architecture that underpins the MySQL database server and allows it to support each major data warehousing use case through one common interface layer. And fourth, a strategy for scaling data warehouses is supplied that eliminates the mixed workload challenge identified by Gartner.

4.1.1 Open Source equals Maximum Innovation

The superiority of the open source model vs. traditional software development approaches has ceased being a debate, with industry analysts and others recognizing the advantages that come with adopting an open source framework. Having a committed and loyal user base that uses, supports, and extends a piece of software allows much greater innovation to occur within a product, plus it helps the software evolve and improve much more quickly.

In the case of MySQL, the best of both worlds is found. MySQL is a formal company, owns the rights to the MySQL database server, and internally develops the product. But in addition, a huge community of MySQL users are constantly testing, submitting bug reports, and tendering enhancements (which undergo a rigorous review and approval process) for the server.

Finally, MySQL has a large and growing ISV partner network that also develops software and improvements for the MySQL server (something that will be demonstrated in greater detail later in this paper).

[6] http://www.cioinsight.com/article2/0,1540,2159186,00.asp.

With three strong avenues of development converging into one product, the end result is greater innovation, faster release cycles, and a product that is able to more quickly meet the exact needs of modern data warehousing applications.

4.1.2 MySQL's Core Feature Set for Data Warehousing

MySQL contains a solid core feature set that is suitable for all data warehousing use cases. The following are just some of the features in the MySQL database server that help enable data warehousing:

- Data/Index partitioning – available in MySQL 5.1 and higher; supports range, hash, key, list, and composite partitioning. Partition "pruning" is available, which involves MySQL only examining the partitions it needs to satisfy a particular query instead of an entire table or index. Partition management is also supported (`ADD PARTITIION, DROP PARTITION`, etc.)
- No practical storage limits – for example, 1 tablespace=110TB limit
- Automatic storage management – autogrowth data files, etc.
- ANSI-SQL support for all datatypes – including BLOB and XML
- Built-in Replication – simple and easy to configure
- Main memory tables – keeps all data in-resident in RAM; perfect for dimension tables
- Support for a variety of indexes – B-tree, fulltext, clustered, hash, GIS
- Multiple-configurable data/index caches
- Pre-loading of index data into index caches
- Unique query cache – caches result set + query, not just data and therefore provides near instantaneous response times for repetitive queries like those used in data warehousing
- Parallel data load – loads multiple files at the same time
- Multi-insert DML – allows array-style processing via normal INSERT commands
- Data compression – provides enormous storage savings
- Read-only tables – protects sensitive data
- Encryption – further protection for sensitive data
- Cost-based optimizer – eliminates need for rule-based query writing
- Wide platform support – no need for special hardware or operating systems

Support for other data warehousing features such as materialized views and other like objects will be provided in upcoming versions of the server.

4.1.3 A Visionary Architecture for Data Warehousing

A number of IT professionals choose MySQL as their data warehousing database because it offers a new and flexible paradigm of database management. One key technical differentiator between MySQL and other database platforms – whether they are proprietary or open source – is the pluggable storage engine architecture of MySQL.

The MySQL pluggable storage engine architecture allows a database professional to select a specialized storage engine for a particular application need while being completely shielded from the need to manage any specific application coding requirements. The pluggable storage engine architecture provides a standard set of management and support services that are common among all underlying storage engines. The storage engines themselves are the components of the database server that actually perform actions on the underlying data that is maintained at the physical server level.

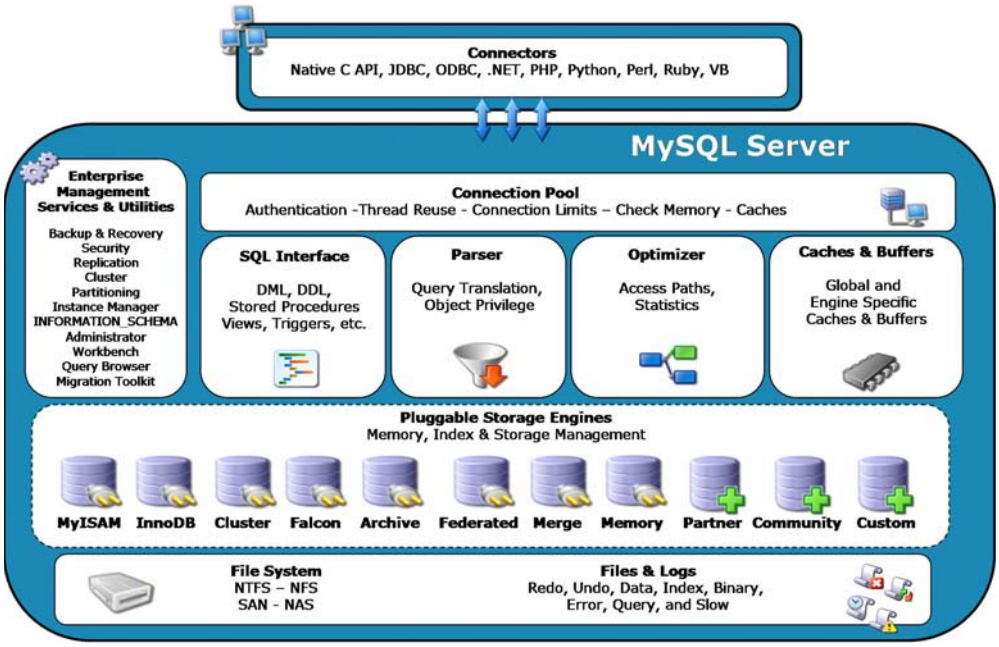


Figure 1 – The MySQL Architecture

This efficient and modular architecture provides large performance and manageability benefits for those wishing to specifically target a particular application need or data warehousing use case. The technical advantage experienced by the application provider is clearly evident in this respect as unnecessary overhead is avoided when only certain engines are chosen for application use.

For example, the MySQL server has built-in storage engines that are designed for:

- ACID transactional applications
- Non–transactional applications (that can therefore insert and read data faster)
- Main memory operations for very fast read times
- Clustered / High-availability database environments
- Compressing historical data down to a very small footprint
- Referencing Non-DBMS flat files as part of the MySQL database
- And several other uses

In addition, a single database or application can use different storage engines at the same time for maximum effect, with a single command being all that's needed to change from one engine to another. Finally, an innovative business can develop its own customized storage engine designed to exactly meet the needs of their particular application.

The storage engine power of MySQL results in the benefit of having many different databases in the same box to choose from, with true 'write once' ability in terms of building application code. No other database management system offers a visionary architecture that affords such power and flexibility in designing a data warehousing application like MySQL.

4.1.3.1 Do Storage Engines Make a Difference?

Some may wonder if the different storage engines really make a difference in terms of performance, storage use, etc. As just a simple example, the Archive storage engine of MySQL is designed to efficiently handle large volumes of inserts and compress data down to a small footprint. Below is an example of the

difference in INSERT performance that can be had by simply using the Archive storage engine over two other popular MySQL storage engines:

User Load	MyISAM Inserts Per Second	InnoDB Inserts Per Second	Archive Inserts Per Second
1	3,203.00	2,670.00	3,576.00
4	9,123.00	5,280.00	11,038.00
8	9,361.00	5,044.00	13,202.00
16	8,957.00	4,424.00	13,066.00
32	8,470.00	3,934.00	12,921.00
64	8,382.00	3,541.00	12,571.00

The Archive storage engine has 50% more INSERT throughput over MyISAM (MySQL's default storage engine) and 255% more than InnoDB (A third party transactional storage engine owned by Oracle).

In addition, an identically structured 11 million row table takes up 1GB of space if InnoDB is used, 795MB if MyISAM is used, but only 148MB if the Archive storage engine is used. As such a simple example shows, storage engines in MySQL can make a difference in data warehousing applications.

Next let's quickly profile internal MySQL storage engines (those developed internally by MySQL AB) as well as third-party storage engines that can be used for data warehousing.

4.1.4 Internal MySQL Storage Engines for Data Warehousing

Although any MySQL engine can be used for data warehousing, the ones that specifically lend themselves to the data warehousing arena include:

- MyISAM
- Archive
- Memory
- CSV
- Merge
- Federated

4.1.4.1 MyISAM

The **MyISAM** storage engine is the default engine for MySQL. It offers high-speed query/insert capability, is non-transactional, has table level locking (although it does have a concurrent insert feature that allows inserts to be performed without blocking queries), and has good support for indexes (B-tree, full-text, etc.) MyISAM is a good general engine for data marts and traditional data warehouses.

4.1.4.2 Archive

The **Archive** storage engine has already been covered in brief detail. It compresses data by up to 80% and therefore offers good storage savings. In addition, it supplies fast table scans for large tables (>1GB), and offers MVCC and row-level locking. Archive is also unique in that it only allows data to be inserted and read, but never selectively modified (i.e. no UPDATE or DELETE), which makes it good for auditing data or other sensitive information that should not be manipulated in any way.

4.1.4.3 Memory

The **Memory** engine is what one would expect – an engine that keeps all data in memory at all times. Main memory tables are especially useful in data warehouses for dimension tables that are the object of joins and single table scans, with MySQL memory tables offering very fast response times for both full table scans and index lookups (B-tree and hash indexes are supported).

4.1.4.4 CSV

CSV tables allow flat file data that exists in comma delimited form to be accessed via SQL from within the MySQL server. Data in CSV format can be instantly loaded into MySQL by simply creating a table object that mirrors the format of the CSV flat file, renaming the CSV flat file on the operating system to the name of the MySQL CSV table object, and all the data is immediately made available to the MySQL server. It doesn't matter if the file as one billion or 100 billion records, the data is instantaneously available for use. In addition, data in CSV tables can be manipulated via DML inside MySQL or edited outside of the server with file editors (and the proper file privileges).

4.1.4.5 Merge

The **Merge** engine is how basic data partitioning was accomplished in versions of MySQL below 5.1. A DBA can combine 1-n identical MyISAM tables together to form one table object that can inserted, updated, deleted, and queried. Each underlying MyISAM table can be placed on a separate physical disk drive than the others and can have its own indexes. In MySQL 5.1 and higher, the Merge engine is still available, however DBA's can also utilize normal data partitioning (horizontal partitioning of rows with range, hash, key, list, and composite being supported).

4.1.4.6 Federated

Finally, the **Federated** engine allows data warehouse designers to create one logical database out of many different physical database servers. Working in much the same was as Oracle database links or SQL Server Linked Servers do, the federated engine provides the ability to create distributed links to other physical database servers and reference their objects as if they existed on the initial source server.

Other internal storage engines are developed and maintained by MySQL (Falcon, Blackhole, etc.), but the above represent the engines that lend themselves best to a data warehousing installation.

4.1.5 External MySQL Storage Engines for Data Warehousing

There are three storage engines built and maintained by third-party storage providers that can be used to meet a few of the most common data warehousing use cases. These are InnoDB, NitroEDB, and BrightHouse.

4.1.5.1 InnoDB

First is **InnoDB**, which is a transactional storage engine maintained by Oracle. InnoDB obviously supports OLTP applications, but it can also serve analytic purposes too and therefore may be applicable to the OLTP/analytic hybrid use case identified by Gartner.

4.1.5.2 NitroEDB

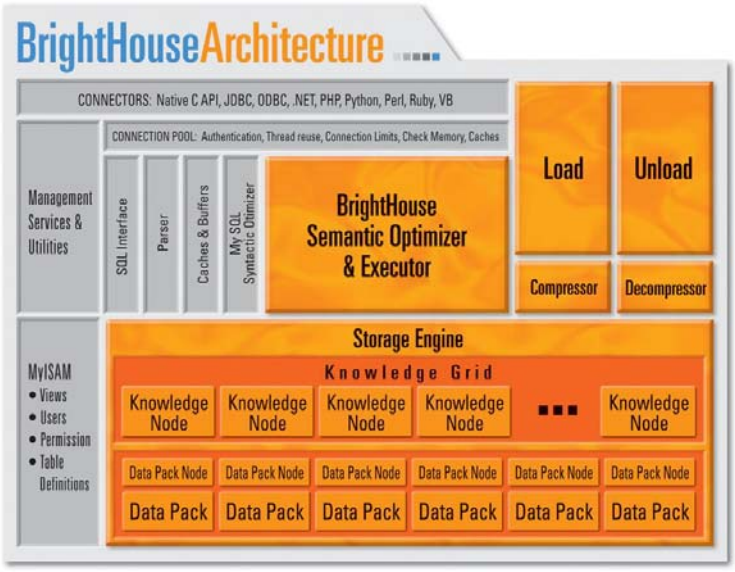
Nitrosecurity is a provider of security information management solutions, which are undergirded by a very powerful database management system. The Nitrosecurity database has been designed with a number of very unique characteristics so it can support the enormous amount of data volumes and concurrent queries that come as part of a security management software package.

Nitrosecurity (www.nitrosecurity.com) and MySQL have partnered to produce the **NitroEDB** storage engine, which has been created to handle the needs of real-time data warehousing customers. The NitroEDB engine has the ability to receive huge volumes of inserts while supporting concurrent queries against the same data at the same time, all with no performance penalty whatsoever. In addition, NitroEDB contains specialized indexes (N-tree, Microcluster) that are designed to deliver extremely fast response times for aggregate styled queries (SUM, AVG, etc.) due to the fact that the aggregate information is actually stored within the index itself.

4.1.5.3 Brighthouse

The last third party storage engine applicable for MySQL data warehousing is **Brighthouse** that is produced by a partnership between MySQL and Infobright (www.infobright.com). At it core, BrightHouse is a highly compressed column-oriented datastore that incorporates MySQL technology, although the engine uses its own load and unload utilities rather than MySQL's mainly because (1) compression and decompression are done on load and unload and (2) the BrightHouse Knowledge Grid is created on load. In addition, BrightHouse also uses its own optimizer instead of the MySQL optimizer because the BrightHouse optimizer knows how to use the information that is stored in its Knowledge Grid to optimize the execution of queries against the BrightHouse engine.

The Brighthouse engine consists of 4 key layers:



1. BrightHouse is a column-oriented data store. This means that instead of the data being stored row by row, it is stored column by column. There are many advantages to column-orientation, not the least of which is the ability to do more efficient data compression because since each column stored a single data type (as opposed to rows that typically contain several data types), compression can be optimized for each particular data type. The data itself within the columns is stored by 65K item groupings. We refer to each of these groupings as Data Packs. The use of Data Packs improves data compression and is also critical to how Infobright resolves complex queries.
2. Data Pack Nodes (DPNs) contain a set of statistics stored related to the data that is stored and compressed in each of the Data Packs. There is always a 1 to 1 relationship between Data Packs and DPNs.
3. Knowledge Nodes are a further set of metadata related to Data Packs, columns or table combinations. The set of these Knowledge Nodes taken together is called the Knowledge Grid.
4. The BrightHouse Optimizer uses the Knowledge Grid to determine the minimum set of Data Packs which need to be decompressed in order to satisfy a given query. In some cases, the information contained in the Knowledge Grid is sufficient to resolve the query, in which case nothing is decompressed.

The Brighthouse storage engine's design is perfect for the massive amounts of data and queries that are the standard fare of traditional or historical data warehousing.

4.1.6 MySQL Storage Engine Comparison Grid for Data Warehousing

A MySQL storage engine reference guide for data warehousing is provided below for a quick at-a-glance overview of the various internal and external engines offered by MySQL, which includes a brief description of their characteristics and the data warehouse use case they are best suited for.

Engine	Storage Cost	Locking	Indexes	Data Loads	Special Features	Best For
MyISAM	Low	Table w/ free inserts	B-Tree, R-tree, Fulltext	Fast	No Transactions, fast queries	Data Marts, Med DW
InnoDB	High	Row w/ MVCC	Clustered, B-Tree	Slow	Transactions	Data Marts, Med DW
Archive	Lower	Row w/ MVCC	B-Tree	Faster	Compressed Data, fast table scans	Historical DW, Auditing
Memory	N/A	Table	Hash, B-Tree	Fast	RAM-based tables	Dimension Tables
CSV	High	Table	None	File-level	Edit file or via SQL	Any needing flat files
NitroEDB	High	Dynamic (no real locking)	N-Tree, Hash, Microcluster	Faster	Concurrent load/query with no penalty	Real-Time DW
Infobright	Lowest	Table	Knowledge Grid	Fastest	Column oriented tables, data compression	Traditional or Historical DW

4.2 Data Warehouse Scale-Out with MySQL

One of the most successful technical strategies used by MySQL customers is “scale out”. The term refers to an architectural design where, rather than “scale up” a single server with additional CPU’s and memory to handle greater data management loads, a data-driven application scales out by separating and spreading a database’s workload across commodity hardware, thus achieving much better fault tolerance and performance. Many of the most heavily-trafficked Web sites use MySQL and a scale-out infrastructure to achieve their high levels of uptime and performance.

The scale-out blueprint can also be utilized for data warehousing/business intelligence, with the practice helping to overcome the mixed workload performance problem identified by Gartner as the biggest performance obstacle coming in data warehousing. Rather than trying to scale-up with expensive additions to monolithic hardware and/or utilize different database vendors to match each data warehousing use case, IT architects can standardize on MySQL, choose the right MySQL storage engine for the right data warehouse use case, and employ scale out to separate the various data warehouse workloads so that performance meets user’s expectations.

For example, the architects of a corporate business intelligence infrastructure can create ETL and data cleansing flows from the various OLTP/operational systems and feed the various “shards” (individual analytic MySQL data stores on commodity hardware) that target a particular BI subject area or meet a specific BI need. A small data mart that serves, for example, a financial analyst business unit may use MySQL with the MyISAM and memory storage engines that make up one shard. A marketing team may be served by another MySQL data warehousing shard that mines tons of historical data from the BrightHouse storage engine, while executive BI dashboards are being fed with real-time data by another MySQL shard using either the NitroEDB or MyISAM storage engines (see Figure 2).

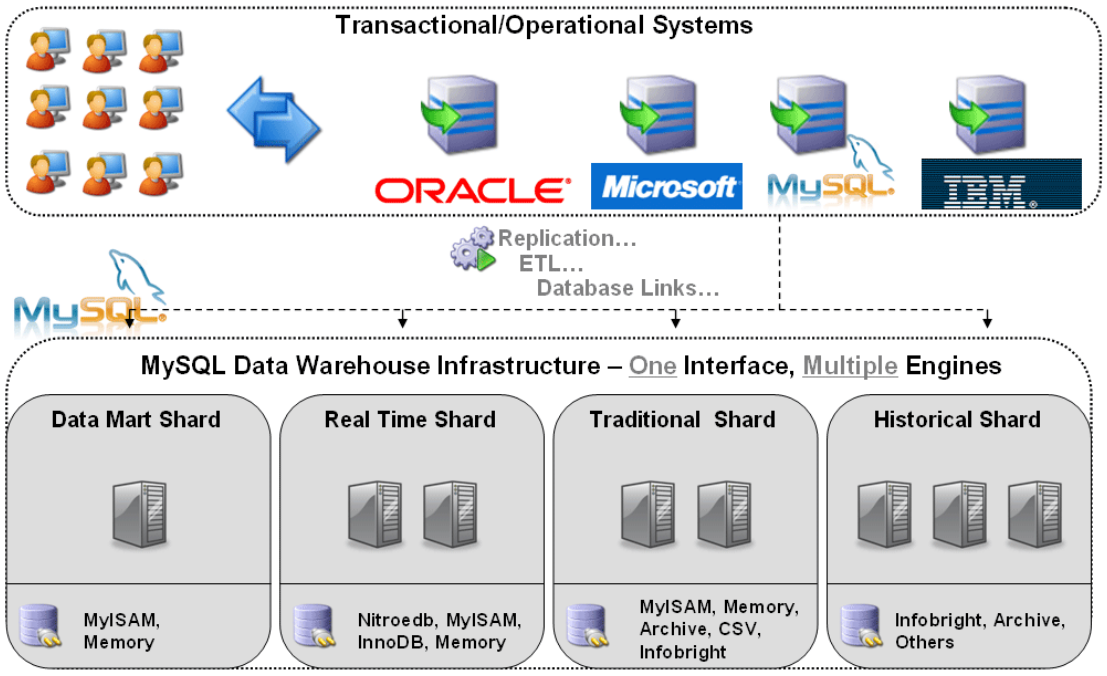


Figure 2 – Sample Scale-Out MySQL Data Warehouse Configuration

A scale-out MySQL data warehousing configuration serves as the heart of an overall modern business intelligence design, an example of which is shown in Figure 3.

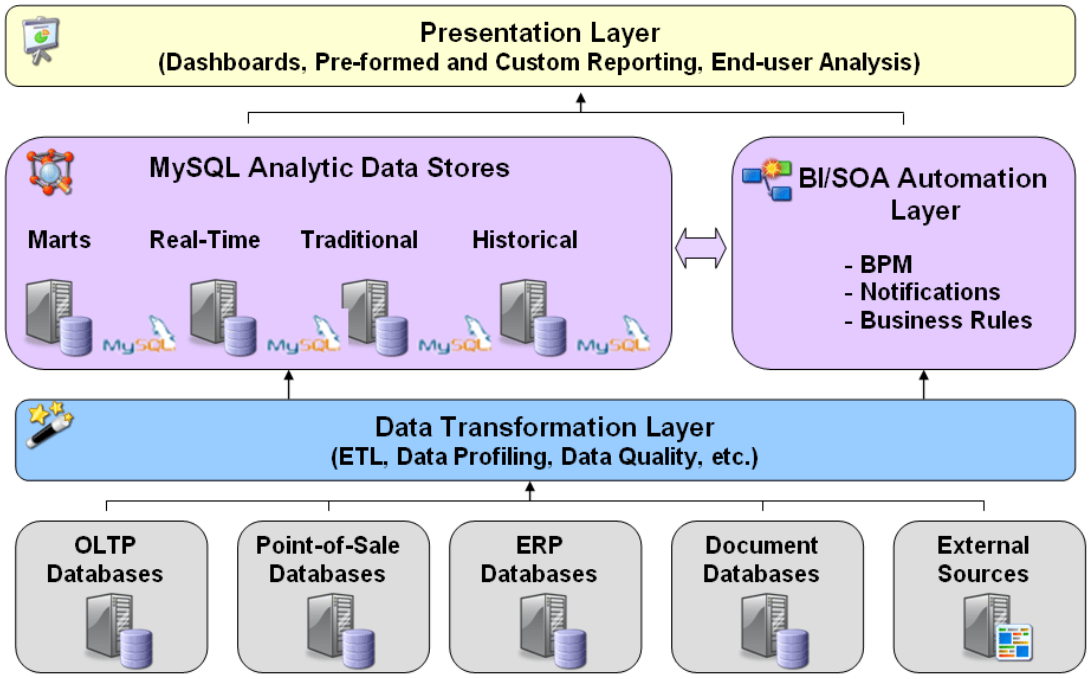


Figure 3 – Modern Business Intelligence Infrastructure

4.3 MySQL Business Intelligence Partner Network

In addition to having a strong database server that supports data warehousing, MySQL also has a vast network of partners who make both proprietary and open source business intelligence software. Whether it is in the area of reporting and dashboarding tools (e.g. Business Objects, SAS, Jaspersoft), ETL and data movement (e.g. Informatica, DataStage, Pentaho), or BI development and administration (e.g. Quest, Microstrategy, etc.), there are MySQL partners available who can assist in all phases of data warehouse construction and operation.

Figure 4 highlights some of the major BI partners that support MySQL in the solutions they offer.

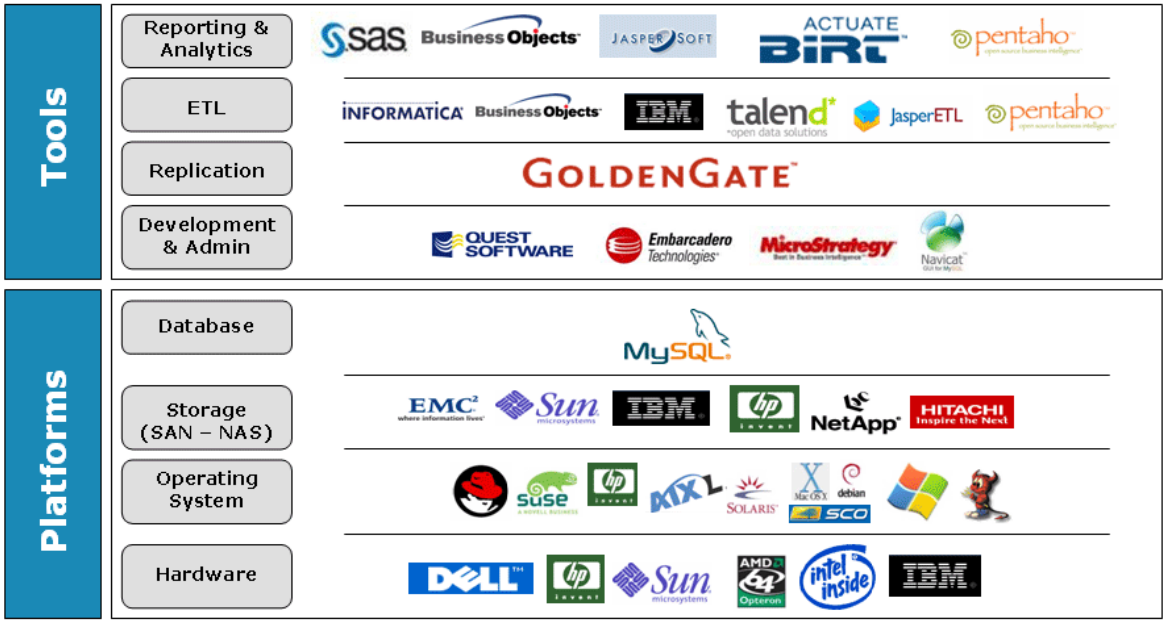


Figure 4 – Sample of MySQL Data warehousing/Business Intelligence Partners

4.4 Examination of MySQL Data Warehousing TCO

One of the driving factors that can be attributed to the popularity and adoption of open source database software in business today is the dramatic cost savings that accompanies owning such software. Open source is a much smarter way of producing and maintaining high quality software than traditional, proprietary methods, with the approach having a benefit of being able to keep costs low.

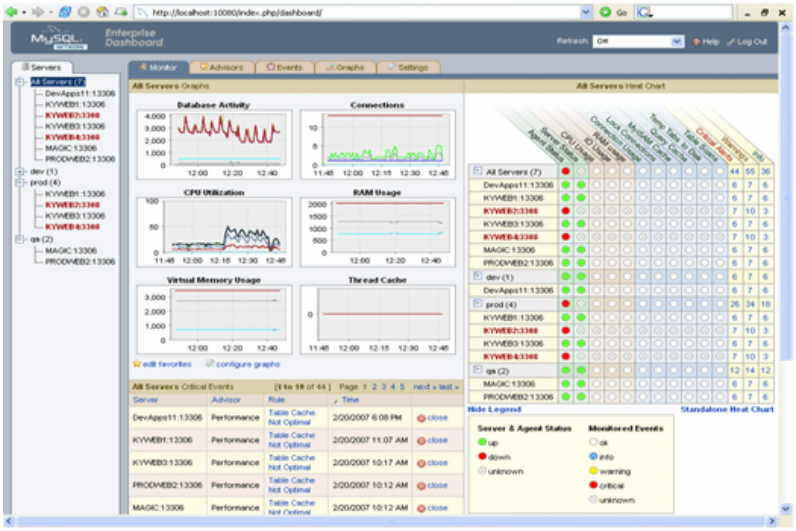
Customers who choose MySQL typically save 90% of the license, maintenance, and support costs over the expenses incurred from using proprietary databases. As the CEO of RightNow, a large CRM provider, has observed:

> "Using MySQL and other open source technologies, RightNow has built an enterprise-class CRM application hosting environment that supports over 3,000 deployments for some of the world's largest organizations. Our systems have facilitated over 1 billion customer interactions on behalf of our clients while maintaining reliability at or above 99.97 percent. Money spent on proprietary databases, when there is a viable open source alternative, is money wasted."
>
> - Greg Gianforte, CEO and founder of RightNow.

A sample of the cost comparisons between using MySQL and other proprietary platforms follows (using the default settings of the ROI calculator on the MySQL web site):

	MySQL	Microsoft	Sybase	IBM	Oracle
Product Line	MySQL Enterprise	Microsoft SQL Server 2005	Sybase ASE 15	IBM DB2 v9	Oracle 10g
Product Edition	Gold	Enterprise Edition	Enterprise Edition	Enterprise Edition	Enterprise Edition
Database Server Source Code	Open Source	Proprietary	Proprietary	Proprietary	Proprietary
Pricing Model	Per Server	Per CPU	Per CPU	Per CPU	Per CPU
Software License (Per Unit)	$0	$24,999	$24,995	$36,400	$40,000
Annual Subscription, Support & Maintenance (Per Unit)	$2,995	$5,000	$4,999	$7,280	$8,000
Costs					
Upfront Software License	$0	$299,988	$299,940	$436,800	$480,000
Subscription, Support & Maintenance (for 3 Years)	$53,910	$180,000	$179,964	$262,080	$288,000
Total Cost of Ownership					
TCO (for 3 Years)	$53,910	$479,988	$479,904	$698,880	$768,000
TCO Savings					
TCO Savings using MySQL (USD)		$426,078	$425,994	$644,970	$714,090
TCO Savings using MySQL (%)		88%	88%	92%	92%
Times more expensive than MySQL		8 x	8 x	12 x	14 x

With MySQL Enterprise – MySQL's solution for businesses that utilize the MySQL database server in critical production and business intelligence environments – customers get the highest quality database software, coupled with intelligent services that automatically assist in monitoring and tuning data warehouses, and global 24x7 production support to troubleshoot any issues and answer questions when they arise. All of this is delivered at a fraction of the cost of proprietary database licenses and support costs.



In particular the MySQL Enterprise Monitor (part of the MySQL Enterprise solution) supplies real value for data warehousing professionals who must monitor and maintain numerous analytic data stores. The automated nature of the MySQL Enterprise Monitor eliminates the need for manual monitoring of data warehouses because the service constantly scans each MySQL analytic data store and evaluates its uptime and performance (both database and server) against built-in rules that are recommended best practices from the experts at MySQL. Any deviations in availability or performance are immediately communicated to responsible parties, along with expert advice on how to go about fixing the issues that have been identified.

5 MySQL Data Warehousing Customer Profile

BlueLithium, a leading online marketing company, has implemented a business-critical data warehouse application built upon MySQL and the BrightHouse database engine from Infobright Inc. BlueLithium, one of the top five US online advertising networks, uses data from 145 million consumers worldwide, combined with sophisticated analytics and advanced targeting technologies to create value for both marketers and publishers.

"Our advanced analytics help us do better ad targeting which delivers increased value to our customers -- as well as a better user experience," said Jay Webster, General Manager, BlueLithium Performance Network. "BrightHouse allows us to do very complex analyses on over 30 terabytes of data while leveraging our skill set in MySQL."

6 Conclusion

Successful modern businesses have realized they need a smart data warehouse/business intelligence infrastructure to actively compete in today's marketplace, and are now beginning to recognize how they can use open source database technology to reach that goal. MySQL is capable of meeting each of the common data warehouse uses cases in play today and can also successfully navigate the issue of mixed workloads in a data warehouse via the server's multi storage engine architecture and scale out abilities.

In addition, customers can enjoy all the safety nets of proprietary software support and updates with MySQL Enterprise, but do so without breaking their budget The combination of the powerful and innovative MySQL Enterprise server, enterprise-ready services and production support, and a total cost of ownership that simply has no peer in the database industry make MySQL an obvious choice for modern businesses looking for top-notch data management capabilities in the area of data warehousing and business intelligence.