



MySQL Workbench

Database Design. Development. Administration. Migration.

A MySQL® White Paper



Table of Contents

1	Introduction	3
2	Types of Data to Manage	3
3	Model-Driven Data Management.....	4
4	MySQL Workbench – Data Modeling/Design for MySQL	8
5	MySQL Workbench: SQL Development	13
6	MySQL Workbench: Administration.....	16
7	MySQL Workbench: Performance.....	23
8	MySQL Workbench: Tables and Data Migration	24
9	Conclusion.....	24
10	Additional Resources	25

1 Introduction

Data is the lifeblood of all successful businesses, no matter the size of the company or the industry they serve. A company's data takes many forms: it can be new transactional data in the form of orders from a Web site, business intelligence data that's gleaned from customer information which helps a company's management staff make smart strategic decisions, historical data trails needed for compliance officers, or metadata that describes the various data elements that make up data-driven systems and how each one of those elements is used. Should a company mishandle or lose key data, it is oftentimes a devastating experience and one that can end in great financial costs as well as loss of reputation.

This being the case, modern and successful companies leave nothing to chance when it comes to the definition, design, implementation, and administration of their data throughout its lifecycle. This comes from a professional and process-driven approach to the creation and management of data that will flow through its business systems – a process that is led by data management professionals who utilize a model-driven and lifecycle oriented approach to data and employing the right tools to ensure that the capture and administration of data is properly carried out. In many cases this falls within what many also call a DevOps process, where companies deploy methods and automation to provide a streamlined development to release process. As you'll see throughout this paper, MySQL modeling, development and administration tools provide many components that apply to a proper Data and DevOps managed lifecycles in an enterprise.

This paper looks at the various categories of data that businesses manage, examines the reasons why a model-driven and lifecycle oriented approach to data management is key, and outlines the benefits such an approach provides. It also highlights how the agile tools for data modeling, development, and administration in MySQL Workbench can be an indispensable aid in the hands of experienced data modelers, developers, and DBAs who are tasked with managing the complex data management infrastructure of a dynamic and growing business.

2 Types of Data to Manage

There are many different ways to classify and categorize data. Generally most data falls into one or more of seven categories:

1. Operational data – normally transactional processing data in the form of new/updated customer orders and other data that supports a company's products and services. This data is generally found in relational databases that support transactional data processing.

2. Business Intelligence data – exists in the form of current and past operational data that is being used to understand things like customer purchasing trends, the impact of marketing programs, and more. This data typically resides in databases known as data warehouses or analytic data stores, and is separated from the operational data to improve system response times for those systems.



3. **Historical data** – represents the historical activity of business systems or audit trails of data usage throughout an organization. It differs from business intelligence data in that it is seldom accessed and is primarily kept online to meet government or industry compliance regulations.

4. **Integration data** – used to manage the flow of data from operational systems to analytic or historical data stores. It most often defines the process of how transactional data is transformed into business intelligence data.

5. **Master data** – equates to “reference data”. Reference data does not depend on other data elements for its identity or meaning, and usually serves as consensus data that is shared consistently across systems.

6. **Metadata** – is “data about data” and serves as the definition point of data elements along with describing how they should be used.

7. **Unstructured data** – was traditionally handled in content management systems although with today’s large datasets much has moved directly into a combination of database and big data storage engines), to manage the life cycle of often digital and but also other information (video files, documents, event data, etc.)

Again, there may be more narrowly-defined classifications of data, but the above represents the bulk of what today’s modern enterprise tackles in the area of data management.

3 Model-Driven Data Management

The approach acknowledged by experienced data professionals as best for managing, capture, definition, and implementation of data throughout an organization is one that is model-driven. That is, it depends on modeling the use and relationships of data that exist in the categories of data discussed in the previous section. This applies to both existing systems as well as ones that are in the process of being constructed.

Models are the best means for representing and visualizing the definition of data elements that support the various data stores found throughout an enterprise. Most IT organizations utilize practices such as entity relationship diagramming (ERD) or other forms of modeling to capture and preserve their data structures.

3.1 Model-Driven Data Management Benefits

There are a number of benefits that model-driven data management brings to the table, with the most tangible being the following:

- **Metadata management** – ensures data consistency, enforces standards of data elements used throughout an organization, and assists in identifying and cataloging elements for data governance
- **Rapid application delivery** – reduces the time it takes to craft and implement a new physical data design and also the application that makes use of the underlying database
- **Change management** – helps to manage change between different iterations of data designs
- **Packaged application management** – removes the ‘black box’ feel of packaged applications by graphically rendering the heart of any application, which is the database.
- **Reporting and communication** – greatly simplifies the communication and reporting of new and modified data designs
- **Performance management** – helps to more quickly pinpoint design flaws in data designs that contribute to inefficient response times in actual data-driven systems

Each one of these areas is explored in more detail in the sections that follow.

3.2 Metadata Management

The importance of having well-defined and standardized data element definitions is understood by companies who have wrestled with the difficult task of tracing and fixing the use of data artifacts that mean and refer to the same thing, but are defined differently across various applications and systems. Ensuring, for example, that a data element named CUSTOMER_ID is defined consistently among all systems that use it (e.g. correct datatype, meaning, etc.) lessens the pain of using it in future applications or in analytic data stores that are fed from many operational systems.

In addition, government and industry compliance regulations have created a whole new set of reasons why proper metadata management is critical. The need to ensure the highest possible data quality (no invalid data; data matches its purpose and definition 100%), data privacy (internal only or external), data security (highly sensitive, not sensitive, etc.), what business area owns the data element (e.g. Finance, etc.), and what compliance is being ensured (Sarbanes Oxley, etc.) are all needs that modern enterprises now have.



A model-driven approach to metadata management lets data architects and others responsible for managing data throughout an organization capture and catalog all aspects of data elements. Models and the tools that facilitate metadata management also provide an easy way to reference, report on, search, and examine change control aspects of all data elements defined within them.

3.3 *Rapid Application Delivery*

The heart of the vast majority of modern applications is the underlying database. This being the case, it is critical that there be no delay in designing and physically implementing the data structures that will support critical business applications.

There is really no substitute for what a model-driven approach offers when it comes to quickly delivering good physical database designs. Many times a database design begins in a conceptual way, which equates to defining data entities, their relationships, and attributes in a way that does not target any one physical database vendor's platform. Once a logical data model is agreed upon, it is then transformed into a physical design that is directed at a specific RDBMS (e.g. MySQL, Oracle, etc.) The logical model is oftentimes amended to meet the requirements of a particular physical platform and for performance purposes (e.g. adding partitioning to tables, etc.)

In addition to modeling standard data-related objects (tables, indexes, etc.), code may also be included in certain physical models so that everything related to or that touches data is kept in one place. This means including objects such as stored procedures, triggers, and more in a model. In addition, the security aspects of a design can also be included so that user login details and individual permissions are recorded. Such abilities (all of which are included in MySQL Workbench) greatly enhance the capabilities of developers to move projects forward as everything they need to create and manage the data aspects of their applications resides in their model-driven tools.

All good modeling tools, like MySQL Workbench, support forward engineering of physical designs, which means that all the SQL code used to create a database and its dependent objects is automatically created and runs right the first time. This eliminates the both error-prone and time-consuming processes of a developer or DBA hand coding database creation syntax (e.g. autogeneration of SQL DDL).

3.4 *Change Management*

Data designs and their corresponding databases have version control and lifecycle management considerations just like applications do, so it is important that data architects, DBAs, and developers understand the need to perform proper change management for their databases and metadata designs. A model-driven approach that's linked into some form of repository is best, although manual methods of archiving versioned data models and metadata schemes can also work.

Perhaps one of the most difficult challenges facing data management professionals in the area of change management is successfully performing complex alterations to existing physical database designs. Performing the detailed impact analysis of proposed changes and preparing the database change scripts can be a long and error-ridden task, which is unfortunate as mistakes made during database alterations can be very costly.

Fortunately, most good modeling tools like MySQL Workbench alleviate such problems as they contain synchronization utilities that allow a DBA or developer to make changes to a physical data model and then synch those changes with an existing physical database. The tool does all the work of performing the impact analysis and generating the right database altering code, with all changes

being easily previewed before they are executed against the target database.

In most tools, the reverse can be done as well – a data model can be altered to reflect changes that have been made to an existing physical database. This aspect is important as oftentimes emergency changes must be performed to a database, but if models are being used for change management and revision purposes, they need to be updated to contain what is currently present in the IT infrastructure.

3.5 *Package Application Management*

Packaged/purchases applications are often black boxes for data professionals and application developers making them difficult to understand in terms of how an underlying database is used. Furthermore, when enhancements or other modifications are made to the packaged application, it's important to understand the impact of those changes and to ensure they are done right the first time.

A model-driven approach to packaged application management can greatly simplify this process. Models are the perfect way to visually understand the data relationships and data definitions of a complex database structure. Via a reverse-engineering utility found in nearly all good modeling tools, a DBA or developer can quickly visualize the inner workings of a packaged application's database and understand how parts of the database are used. Should modifications or additions to the database need to be made, a modeling tool can generate both an impact analysis of the changes and the code needed to make things happen.

3.6 *Reporting and Communication*

Modern businesses have great number of people who consumer the data that is captured and in a companies databases. Some may be technically savvy, but many are not, so the challenge becomes how best to educate and communicate the ways to access and understand the data that they need. Further, the same need is present in large development teams who are all working on the same set of data. They routinely need to understand how to code their part of a collectively-built application so that it properly accesses the targeted database.

Again, models can come to the rescue as they are a perfect way for both technical and non-technical people to grasp how data is defined and how it can be accessed. Good modeling tools like MySQL Workbench help on this front as they offer the ability to create visual views (pictures) of model designs, and provide reporting facilities that create Web-based and/or text-based reports that break down large models into report formats that are easily read and understand.

3.7 *Performance Management*

Fast databases are not a nice-to-have – they are a necessity. Databases that power globally used web sites must complete user transactions and present information at a rate fast enough that prevents impatient customers from clicking to a competitor's site. Further, businesses needing up-to-date internal information cannot wait for long drawn out processes that crunch numbers and detail competitive statistics. Instead, they need databases capable of quickly churning out the data necessary

to compete in today's economy.

In the pursuit for better overall database performance, many professionals too often ignore what is perhaps the number one factor leading to excellent RDBMS speed – the physical database design. If a database runs slow, oftentimes those implementing it will add additional hardware power, but throwing hardware at a bad design does not always work out for the best in the long run. In the short term, things may get better and if the database is relatively static in nature, things may remain that way. But if the database is dynamic and the data/user load continues to grow, the situation will oftenslide back to the point where it once was. Instead, its better to first assess the physical design of the database should for design related performance flaws and thus tuning opportunities. Basically , if the foundation is flawed, then the house needs to be put in order at that level prior to anything else being done.

One of the keys to understanding the discipline of performance monitoring is: ***When a DBA monitors a database for performance, they are really validating their physical design implementation.*** If the performance monitor they choose to use is blasting them with flashing lights, alarm bells, and pager alerts, it's probably because their physical design is failing. If all is quiet in the performance monitor, then their physical design is likely a current success. Some will try and argue that SQL code is the number one factor in database performance, but in reality, it is the physical design (the code follows from that).

Creating robust efficient physical designs can be difficult and intricate work. IT professionals need to arm themselves with some serious power tools that have the capability to slice through the difficulties involved in building and retrofitting complex physical database designs. Long gone are the days when a DBA or modeler could handle most of their work with a SQL query interface and a drawing tool. Today, relational databases are just too robust and contain too many complexities for such primitive methods.

At a minimum, those creating databases will need two things: a solid data modeling tool and a robust performance monitoring product. As has already been mentioned, performance monitoring is really the validation of a database's physical design. When foundational cracks are identified with the monitor, DBAs and developers will need a high-quality design tool to aid in rectifying the situation – one that makes quick work of sometimes complicated schema changes.

4 MySQL Workbench – Data Modeling/Design for MySQL

MySQL Workbench includes a powerful visual database modeling and design tool that helps developers and DBAs create new physical data models for MySQL databases and modify existing physical MySQL databases with reverse/forward engineering and change management functions. MySQL Workbench is designed to increase user productivity by providing the means to conceptualize, communicate, build, and maintain key business metadata, high-performance databases and data warehouses.

MySQL Workbench's graphical interface and automated processes ensures out-of-box success for a variety of audiences within modern enterprises including database administrators, application developers, data architects and IT management. The product runs on and provide native look and feel for Windows, Linux, and Mac so that users can design their databases from all the popular desktop operating systems. A quick tour of MySQL Workbench's feature set showcases how the tool supplies

the benefits previously discussed in the area of model-driven data management.

4.1 Database Design

Core to MySQL Workbench's is to helping developers and DBAs visually create physical database designs that eventually end up as MySQL databases. Models are far and away the best and most efficient means to understanding and creating valid and well-performing databases, and MySQL Workbench helps users accelerate their work through a variety of functions and utilities.

MySQL Workbench allows a developer or DBA to create one or many models within its interface and supplies various views of the objects (tables, views, stored procedures, etc.) being designed.

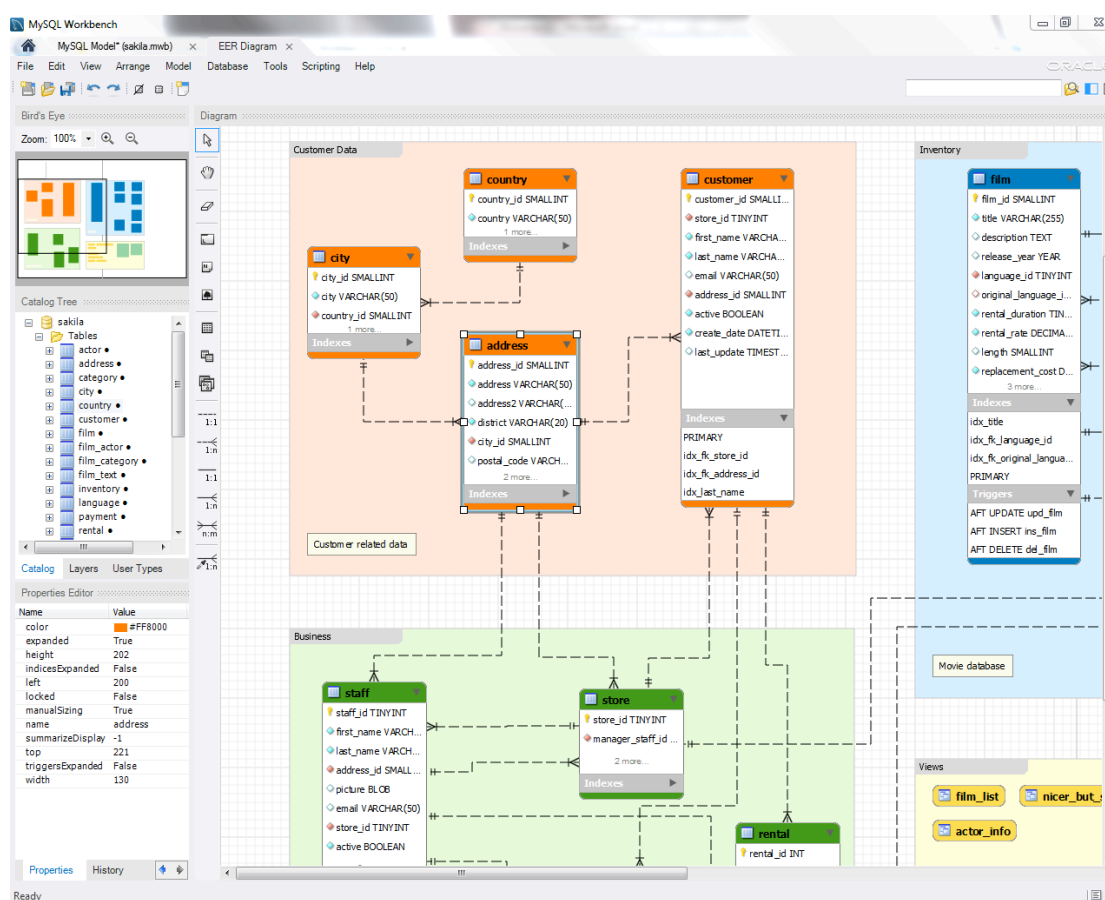


Figure 1 - MySQL Workbench's Visual Data Modeling Interface

Many different utilities and aids within MySQL Workbench help a data modeler quickly design and implement their physical data models. MySQL Workbench contains all the core data modeling elements one would expect in a database design tool: visual design of all objects that make up a database, including tables, relationships between tables (foreign keys), views, triggers, stored procedures, functions, object permissions, and more. Easy to use editors make adding or changing elements like

columns, indexes, foreign keys, privileges, etc., simple to do. For changes made that need to be undone, an Undo function returns things to the way they previously were before a change took place.

In addition, MySQL Workbench has a number of other aids that help DBAs and developers quickly create database designs right the first time. Its Model Validation utility checks a data model for any possible mistakes and reports all found issues back to the user. For large models that are difficult to navigate, the Zoom feature allows a data modeler to easily zoom in and out to get either a bird's eye view of an entire data model or just focus on one specific area. To locate various objects (tables, column names, etc.) in a large model, an advanced Find utility locates all occurrences of whatever search criteria the user supplies, with the results providing point-and-click navigation to whatever is selected in the output.

Finally, a variety of other helpful functions exist in the tool such as the allowance for different modeling notations, an Auto-layout function that automatically arranges tables on a diagram, and a scripting ability that lets advanced users extend the tool via Python

4.2 Forward and Reverse Engineering

One of the most used functions in good modeling tools involves the forward engineering of a physical database design, meaning that the tool generates all the SQL code necessary to create a physical database on a target server. MySQL Workbench provides just such a utility so that a visual data model created in the tool can easily be quickly translated into a physical database on a target MySQL Server, with a nice benefit being that all the SQL code runs right the first time, which eliminates the normal error-prone process of manually writing complex SQL code.

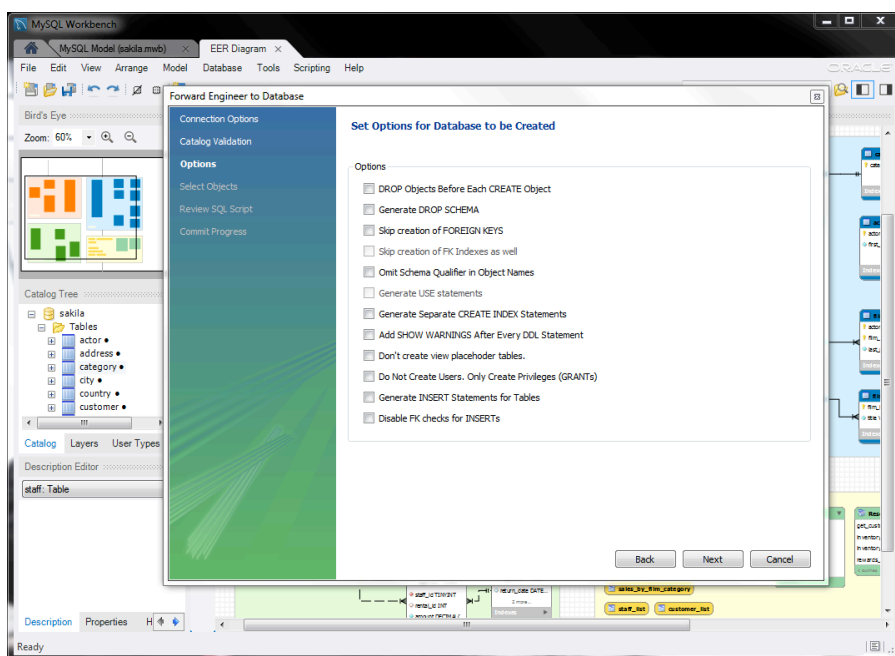


Figure 2 - Forward Engineering Operations can be easily customized in MySQL Workbench

In addition, MySQL Workbench also supplies reverse engineering, to help understand existing databases that may either be custom in nature or part of a packaged application. The reverse engineering utility connects to any existing MySQL Server and builds a new data model from either all or part of the source MySQL databases. Once in the tool, all objects and their relationships can be easily viewed, understood, managed or altered.

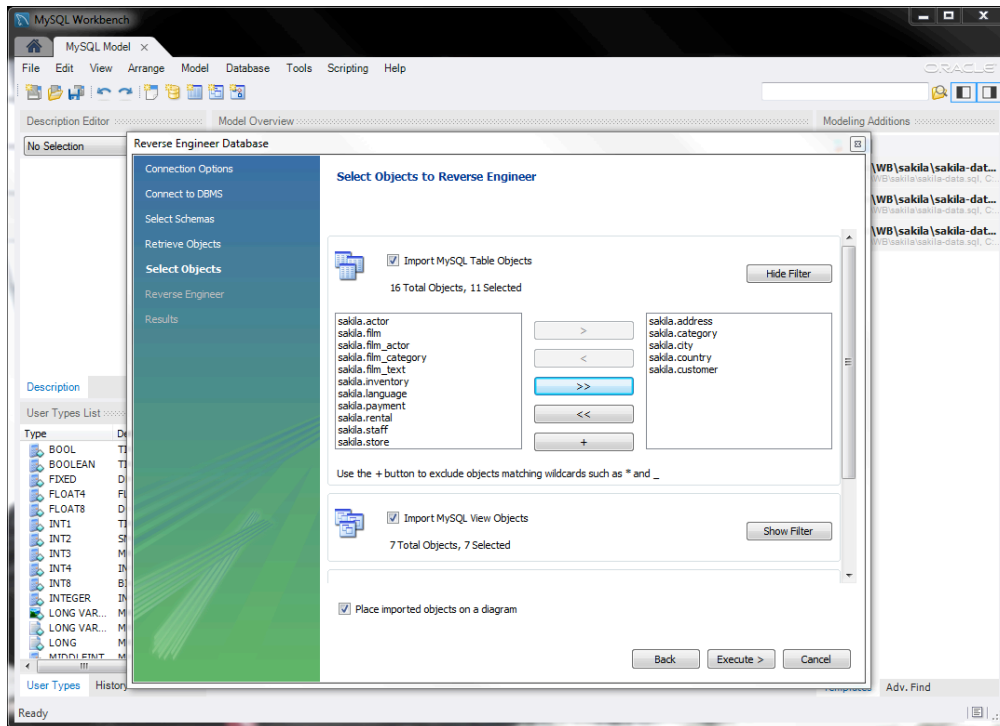


Figure 3 - Reverse Engineering utilities are provided to model/modify existing databases

4.3 Change Management

Database change management, which normally equates to maintaining different versions of database schemas and competently modifying existing databases that are complex in nature, is oftentimes a difficult and challenging task.

To help DBAs and developers with change management, MySQL Workbench offers a Synchronization utility that will compare a data model against a target MySQL server and can be used to perform synchronization between the two. Using MySQL Workbench, a DBA or developer first connects to a target MySQL server, then the tool compares all aspects of the current model with the actual physical MySQL database. The Synchronization utility then presents a graphical display of all the differences that were found and lets the developer or DBA decide what they want to do next.

MySQL Workbench provides three options for dealing with comparing the differences between a model and a physical database:

1. Ignore differences
2. Synchronize the model with the physical database server
3. Synchronize the physical database server with the model

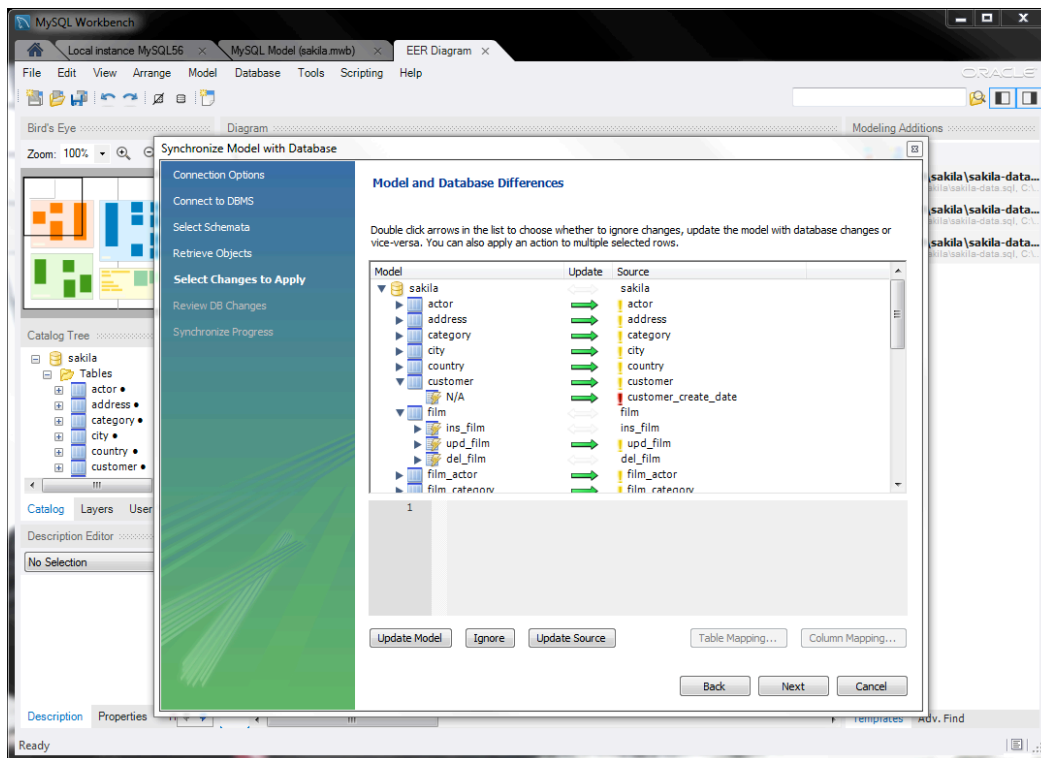


Figure 4 - Synchronization of database and model

MySQL Workbench also allows a user to make such decisions on either a global or per object basis, so everything can be handled in the exact way a DBA desires. Once the synchronization operation is complete, a DBA can save the MySQL Workbench model to preserve versions of how the model and database looked at a particular point in time.

4.4 Schema Validation

MySQL Workbench provides advanced schema validation modules that enable DBAs and developers to test their design models prior to implementation. Modules for validating general and MySQL specific schemas are designed to test for problems around empty content, tables lacking a primary key, duplicate object or column names, foreign keys that reference non-primary key columns and other problems that could lead to less than optimal application performance, scale or reliability.

4.5 Reporting and Documentation

Documenting database designs can be a time-consuming process, but fortunately MySQL Workbench provides DBDoc, a point-and-click reporting and documentation utility that makes such things very easy. A model can be documented in either HTML or plain text format, and includes all the objects and models in a current MySQL Workbench session.

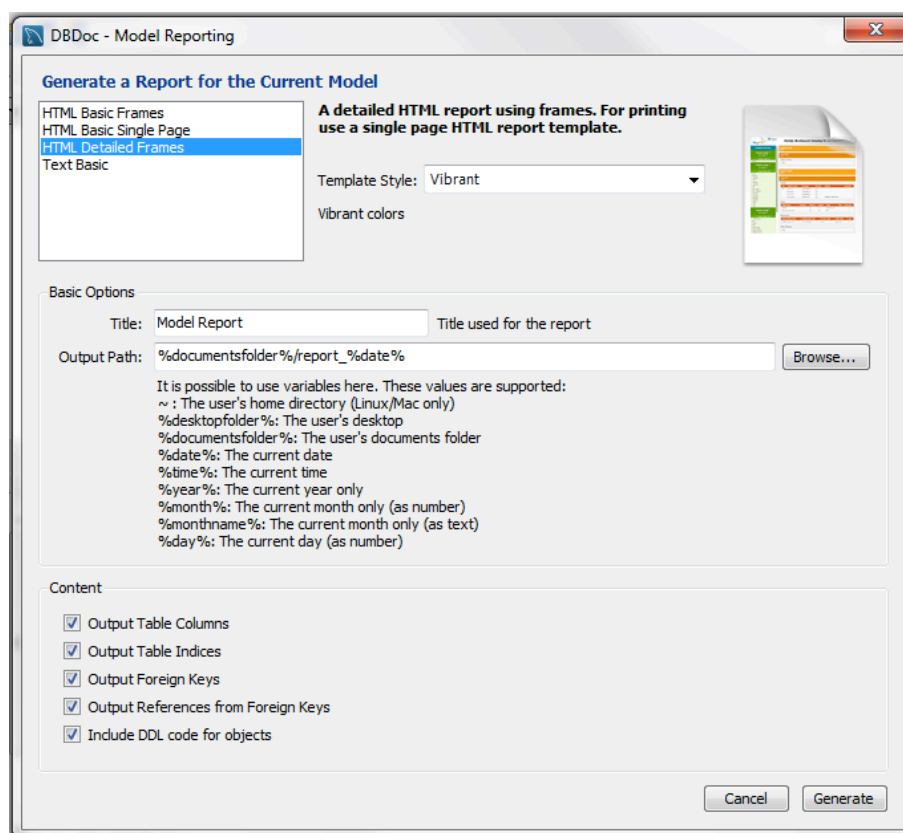


Figure 5 – DBDoc Reporting functions quickly document any existing model

5 MySQL Workbench: SQL Development

MySQL Workbench gives developers a complete set of visual tools to create, edit, format, debug, and manage SQL queries, database connections and objects, as well as simplifying managing the data within your databases. Context sensitive help is also available by just hovering over the SQL you need details on and using the auto-complete mode helps efficiency – both lowering your keystrokes as well as finding and ensuring correct SQL statements and object naming.

5.1 Visual SQL Editor

The visual SQL Editor lets developers build, edit and run queries, create and edit data, and view and export results. Color syntax highlighting helps writing and debugging SQL statements. And EXPLAIN plans can easily be gathered to help optimize queries.

With the SQL editor multiple queries can be executed simultaneously and results can be viewed on individual Tabs in the Results Window. The Table Editor makes it easy to edit data and commit changes using a simple grid format – and provides the developer the SQL DML used for these changes as well. The export facility allows developers to export results data to common formats.

The History Panel provides complete session history of queries and statements showing what queries were run and when. Developers can easily retrieve, review, re-run, append or modify previously executed SQL statements.

In addition, the SQL Snippet panel lets developers save and easily reuse common Selects, DML and DDL code. The most common queries can be placed in snippets so that they can quickly be retrieved and re-used later by simply double clicking in the snippet list.

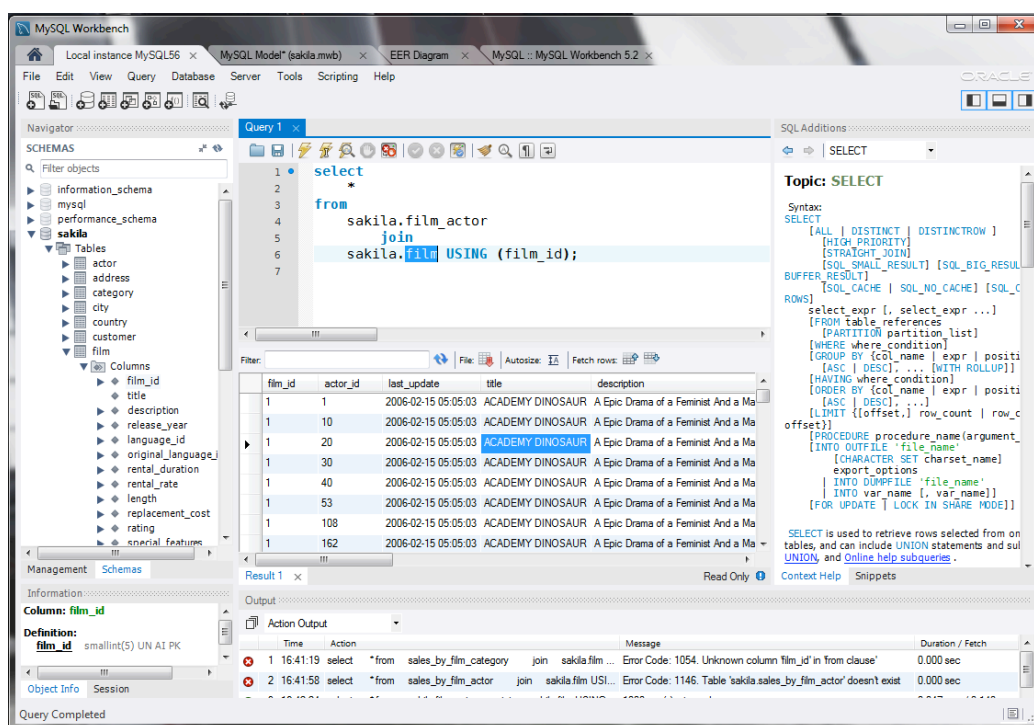


Figure 6 - MySQL Workbench tools for creating, executing, and optimizing SQL queries.

5.2 Connection Management

From MySQL Workbench users easily see all their databases and connections.. For adding and editing connections, the Database Connections Panel and Connections Wizard enables developers to create, organize, and manage database connections. Advanced users can use the Manage Connections dialog to enter connection parameters like IP address, port, username, and password. The connection manager lets users test newly created connections and supports various enhanced security and remote management options including support for ssh tunneling – quite handy for using MySQL Workbench to access and manage MySQL databases running in remote datacenters or in a cloud environment.

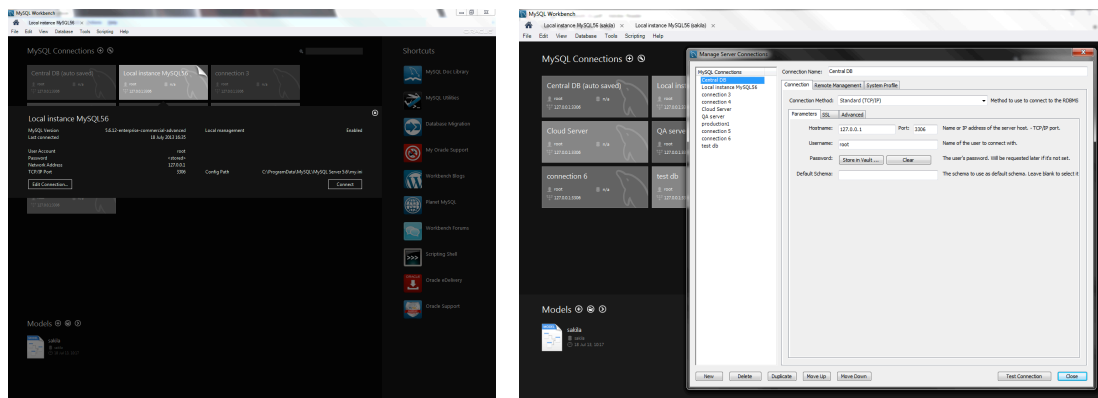


Figure 7 – Viewing, Creating, and Editing Connections in MySQL Workbench

5.3 Navigator

The MySQL Workbench Navigator is a tree based database Object Browser that enables developers to easily navigate database schemas, down to object details. Based on the object type the DBA can select from up to 10 actions on that object. For example, DBAs can visually select tables and fields to query, auto create SQL statements, edit tables, create new tables and databases, drop tables and databases, or perform other maintenance tasks.

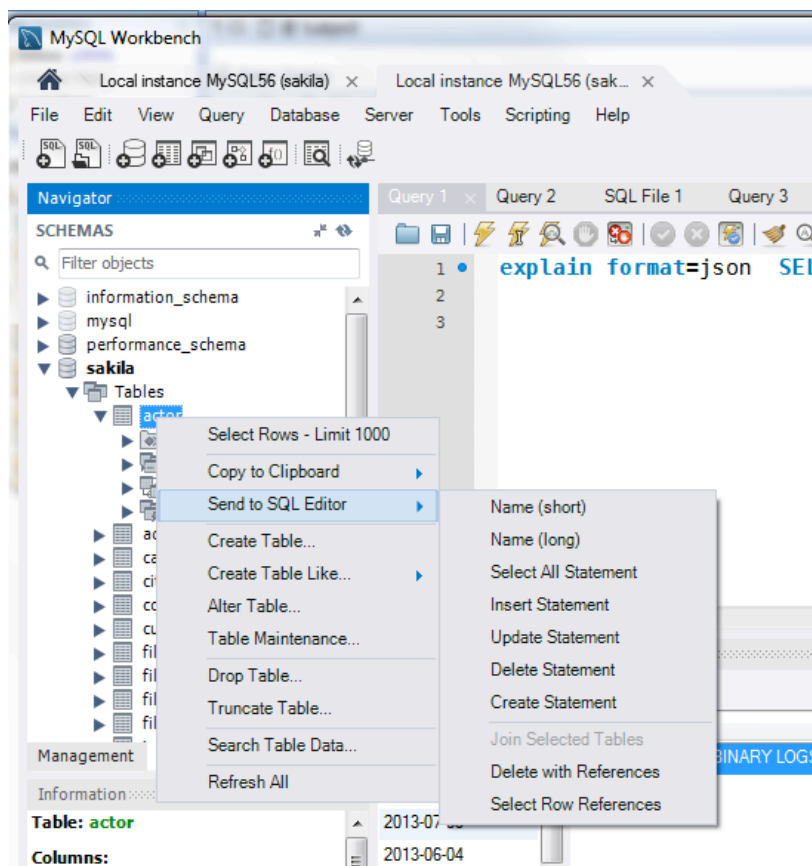


Figure 8 – Browse, Click, and Create SQL Statements and code snippets

6 MySQL Workbench: Administration

MySQL Workbench integrates database administration tools into a simple to use graphical user interface. The visual console enables DBAs to easily perform operations such as configuring servers, administering users, export and import, and viewing logs.

6.1 User Administration

Administering users, granting privileges and viewing privilege information has never been easier. Using MySQL Workbench DBAs can visually add users, assign passwords and setup user profiles. Assigning and revoking global and database privileges is as easy as adding and removing privilege items from an available list.

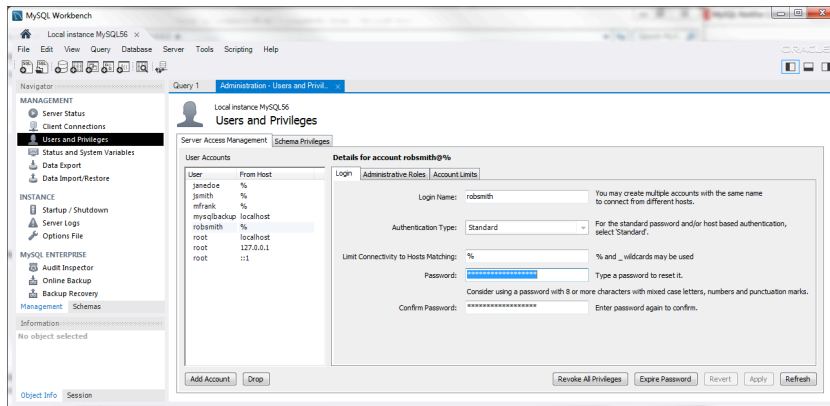


Figure 9 - MySQL Workbench visual console to easily administer user accounts and related security privileges.

6.2 Server Administration

6.2.1 Configuration

MySQL Workbench allows DBAs to easily view and edit general and advanced parameters to quickly fine tune MySQL servers.

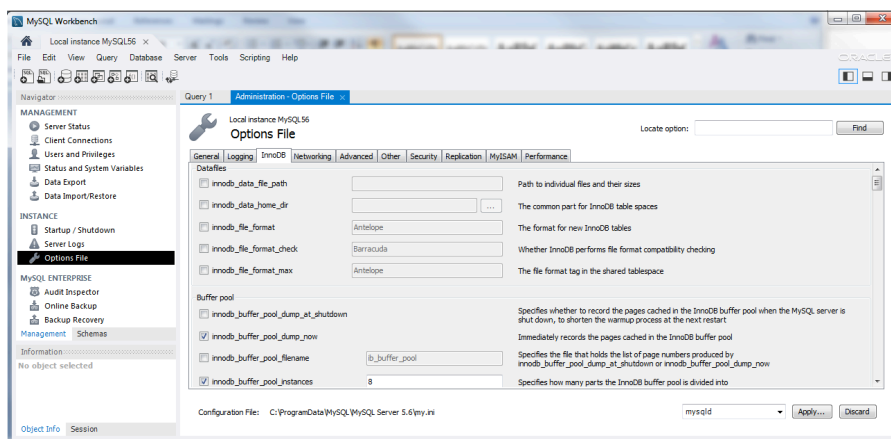


Figure 10 – MySQL Workbench provides visual tools for configuring MySQL Servers.

6.2.2 Schema Inspection and Maintenance

MySQL Workbench provides a suite of eight detailed views of all objects in the schemas plus access to table, index, and other maintenance operations such as ANALYZE and OPTIMIZE TABLE. Schema inspector put includes valuable statics for DBAs such as row counts, average, max sizes, and free space, but other details such as types, character sets, and much more in viewable and sortable grids. With Schema inspector finding your largest table, spotting datatype inconsistencies, permissions issues, and other tasks is straight forward.

In addition to inspecting, schema inspector provides quick maintenance tasks, so once you've spotted issues you can easily click and select from various maintenance or editing options to correct issues with least effort possible.

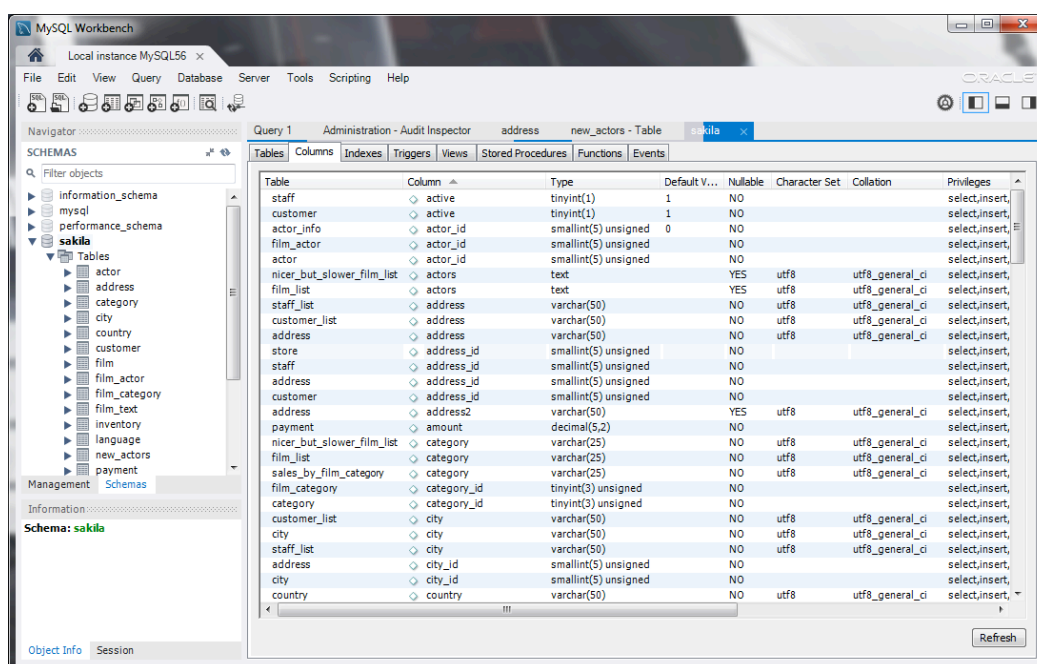


Figure 10 – MySQL Workbench Schema Inspector – shown column view – sorted by name.

6.2.3 Active Server Views

Improve server management with a comprehensive real-time view of all active client connections to the server. Visual tree based navigation provides detailed information about server and status variables, including number of threads, bytes sent and received by clients, buffer allocations size, and more.

6.2.3 Exploring Server Logs

View all the MySQL log files including error logs, binary logs, and InnoDB logs from a single,

centralized administration environment to diagnose server problems quicker and track database changes. '

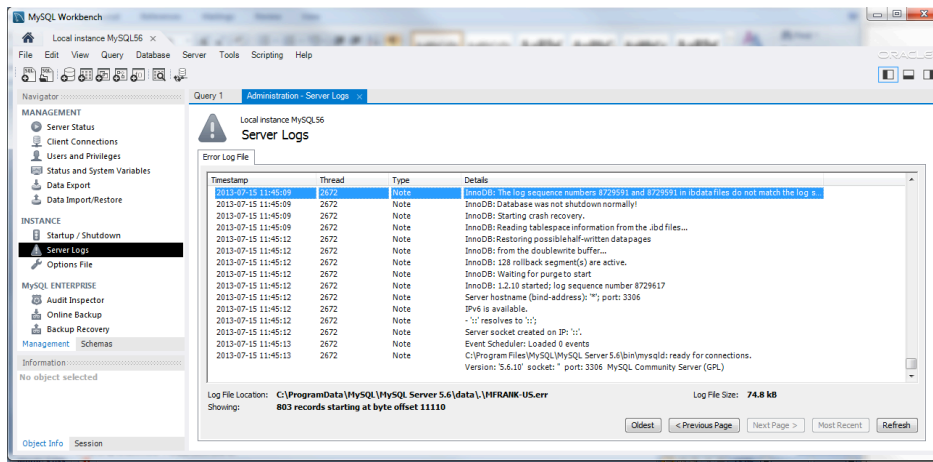


Figure 11 – Easily access and browse server logs in an easy to read grid.

6.3 Data Management and Protection

6.3.1 Backup/Recovery

MySQL Workbench also includes a visual wizard for MySQL Enterprise Backup. This makes creating, configuring, scheduling, monitoring, tracking, and restoring backups easy. Using MySQL Workbench DBAs can run full, incremental, and partial backups. Backups jobs for full and incremental backups can be run immediately or easily scheduled. Users can easily track backups in progress and select from previous backups to quickly restore and recover.

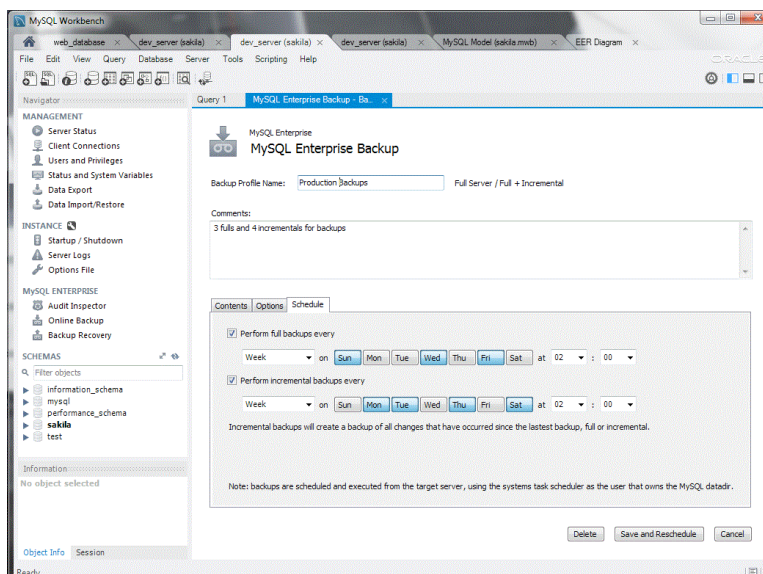


Figure 12 – Simple to use Backup Wizard

6.3.2 SSL Certificate Wizard

The SSL Certificate Wizard helps create proper SSL certificates for both MySQL clients and MySQL servers. Connections in MySQL Workbench are updated with the certificates by the wizard.

6.3.3 Export/Import

MySQL Workbench includes visual tools for mysqldump. Easily export or import data by selecting entire database schemas and/or specific tables and objects from the Object Browser. Advanced configurations are available for concurrency controls, backup type, and output formatting.

6.3.4 Table Data Import/Export Wizard for JSON and CSV

The Table Data Import/Export Wizard supports import and export of JSON and CSV files allowing for easy interoperability with MySQL tables. The new Table Data Import/Export Wizard also provides flexible configuration options including separators, column selection, encoding selection, and more.

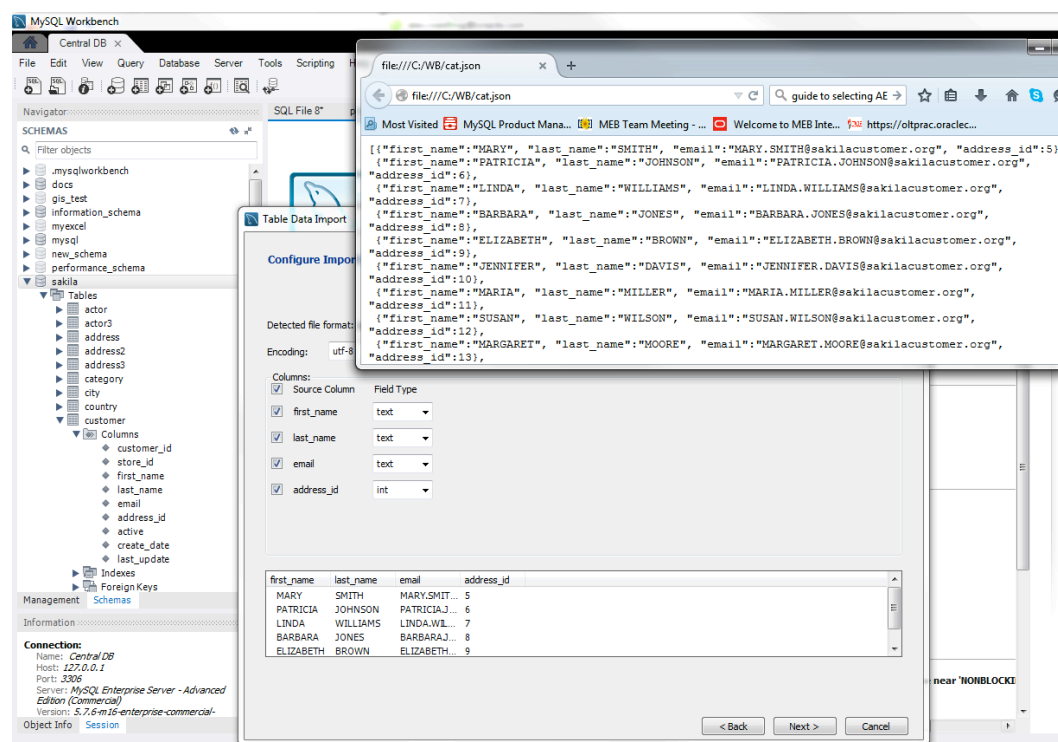


Figure 13 – Import and Export JSON files

6.3.5 Audit Data

The MySQL Audit Log Inspector that allows users to securely browse and analyze audit data generated via MySQL Enterprise Audit. MySQL Workbench will automate enabling auditing if MySQL Enterprise Audit is installed. From there users are can use powerful filtering and search capabilities to easily assess the audit log content generated via MySQL Enterprise Audit.

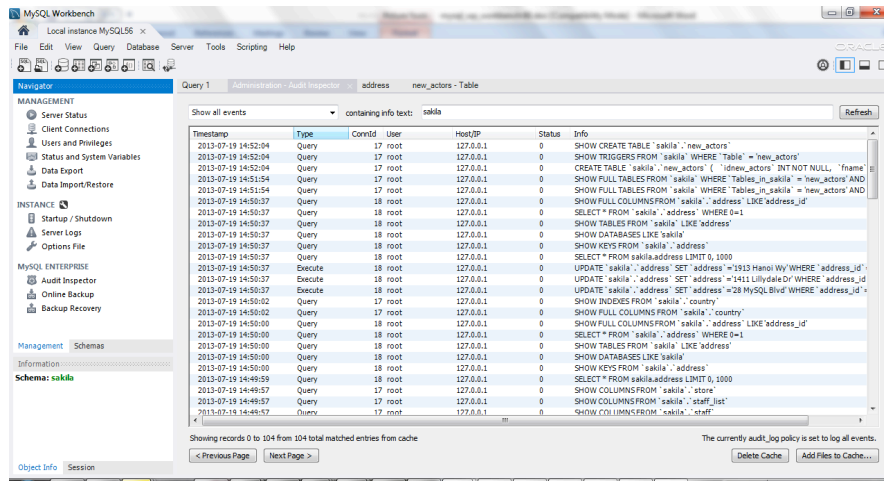


Figure 14 – Quickly view and analyze Audit Logs

6.4 System Status and Setup View

Get an immediate view into the basic health indicators and counters within a MySQL environment. Easily see key values for server configuration from available features, key server directories, replication, security settings for authentication, and ssl via basic quick view reports. Simple graphs enable DBAs to easily see memory usage, connection usage, hit rates, and more.

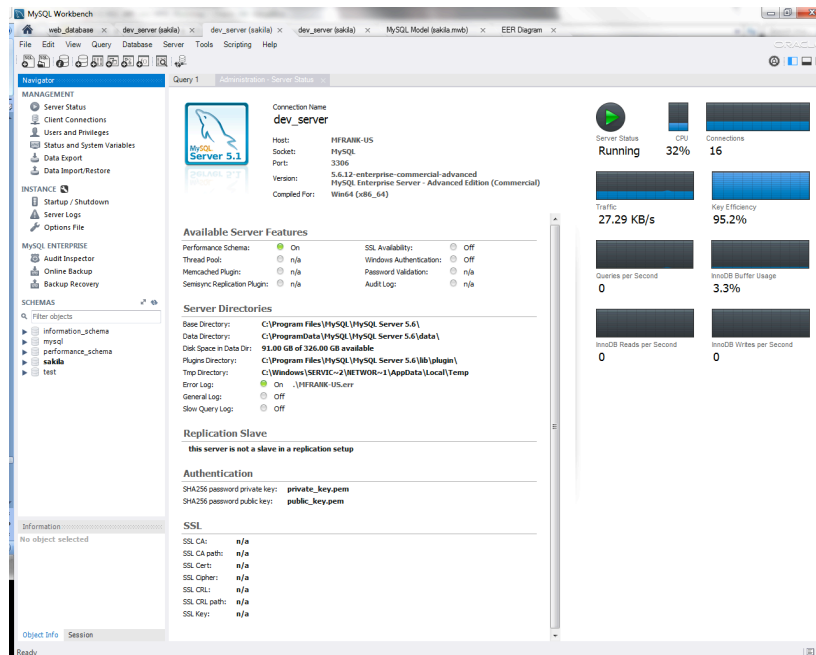


Figure 15 – In single view - quickly see Key Server values and Metrics

7 MySQL Workbench: Performance

MySQL Workbench includes various tools for both DBAs and Developers related to viewing and improving performance. The Performance Dashboard and reports allow DBAs to easily view overall server performance, and various reports provide views of IO hotspots, SQL statements, Network, Data Engine, and more. For developers, MySQL Workbench provides easy to understand views into optimizing queries and data access.

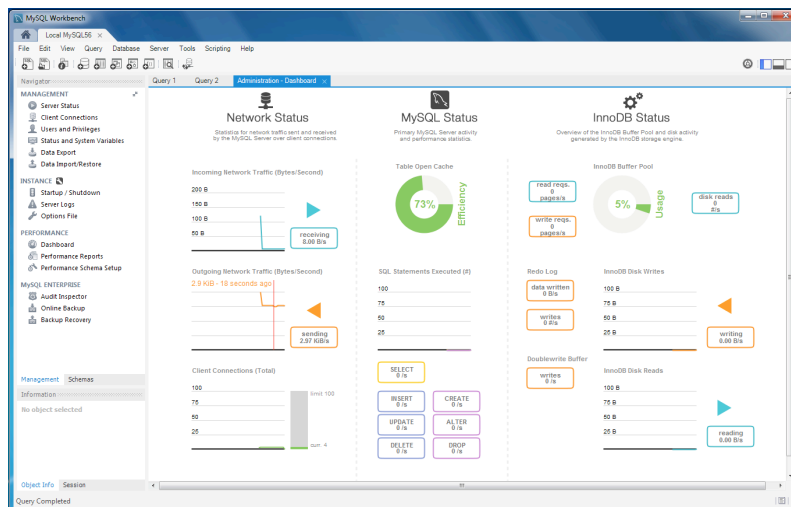


Figure 16 - MySQL Workbench Performance Dashboard provides a visual overview of MySQL server performance.

- **Performance Dashboard** - The Performance Dashboard provides quick "at a glance" views of MySQL performance on key server, network, and InnoDB metrics. Simply mouse over various graphs and visuals to get added details.
- **Performance Reports** - Over 20 reports help to analyze the performance of your MySQL databases. Targeted reports make analyzing IO hotspots, high cost SQL statements, Wait statistics, InnoDB engine metrics. MySQL Workbench leverages the SYS views on the Performance Schema.
- **Explain Plan** - The explain plan shows the operations MySQL performs when it runs SQL statements. This information can help optimize SQL performance. MySQLWorkbench Visualize Explain plans graphically show and highlight how SQL statements execute within MySQL. By showing developers costs and tuning hints, MySQL Workbench improves and simplifies SQL statement performance tuning.
- **Query Statistics** - Query Statistics provide instant statistics on SQL executed from the Workbench Editor, such as details about the fields in your result set and key performance statistics from your query, such as client timing, network latency, server execution timing, index usage, number of rows scanned, joins, use of temporary data storage, and more.

8 MySQL Workbench: Tables and Data Migration

MySQL Workbench provides a complete, easy to use solution for migrating Microsoft SQL Server and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL and any of the operating systems supported by MySQL.

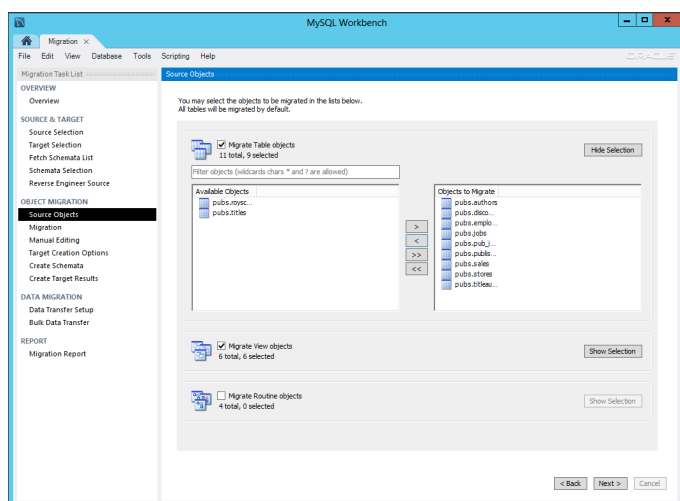


Figure 15 - MySQL Workbench Migration Tool provides a visual, point and click interface to easily migration other RDBMS schema, objects and data to MySQL.

The MySQL Workbench Migration Tool is designed to save DBA and developer time by providing visual, point and click ease of use around all phases of configuring and managing a complex migration process:

- **Migration project management** - allows migrations to be configured, copied, edited, executed and scheduled.
- **Source and Target selection** – allows users to define specific data sources and to analyze source data in advance of the migration.
- **Object migration** – allows users to select objects to migrate, assign source to target mappings where needed, edit migration scripts and create the target schema.
- **Data migration** – allows users to map source and target data and data types, set up data transfer and assign post data transfer events where needed.

With the MySQL Workbench Migration Tool users can convert an existing database to MySQL in minutes rather than hours or days that the same migration would require using traditional, manual methods.

9 Conclusion

Modern businesses know the value of using a model-driven approach to manage the definitions and



designs of data used in their key production systems. Models stand second-to-none when it comes to quickly understanding, organizing, and managing both custom built and packaged data-driven applications.

To help data professionals tackle the challenges that come with designing and understanding complex databases, Oracle offers MySQL Workbench, which helps DBAs and developers both quickly create new MySQL databases and manage the lifecycle of existing MySQL databases with change management and reporting functions. With MySQL Workbench, the productivity of a database professional is increased as the tool eliminates the complex and error-prone processes of manually performing the previously mentioned tasks, with the end result being a quicker and more efficient delivery of MySQL-based applications.

Additional information about the MySQL Workbench features is available at:
<http://www.mysql.com/products/workbench/>.

10 Additional Resources

White Papers

<http://www.mysql.com/why-mysql/white-papers/>

Case Studies

<http://www.mysql.com/customers/>

Press Releases, News and Events

<http://www.mysql.com/news-and-events/>

Live Webinars

<http://www.mysql.com/news-and-events/web-seminars/>

Webinars on Demand

<http://www.mysql.com/news-and-events/on-demand-webinars/>

MySQL Enterprise Edition Trial

<http://www.mysql.com/trials/>

MySQL Workbench and Migration Toolkit (download and demos)

<http://www.mysql.com/products/workbench/>

To contact an Oracle MySQL Representative

<http://www.mysql.com/about/contact/>