itcourseware™

TRAINING MATERIALS FOR IT PROFESSIONALS

MySQL Administration

MySQL Administration

John McAlister and Dave Shaw

Published by ITCourseware, LLC., 7245 South Havana Street, Suite 100, Centennial, CO 80112

Contributing Authors: Danielle Hopkins

Editor: Ginny Jaranowski

Special thanks to: Many instructors whose ideas and careful review have contributed to the quality of this workbook and the many students who have offered comments, suggestions, criticisms, and insights.

Contents

CHAPTER 1 - INTRODUCTION TO DATABASE CONCEPTS AND MYSQL

OBJECTIVES

- ❈ Describe the structure of a relational database.
- ❈ Describe the uses of the SQL programming language.
- ❈ Explain the different ways that users can access a database.
- ❈ Describe MySQL's benefits and features.
- ❈ Describe the history of MySQL.

Features of a Relational Database

- A *relational database* organizes data into tables, each table being much like a spreadsheet.
- Each *table* in a database normally describes a specific group of information.
  - Each table is given a specific name.
- A table is organized in columns, each with a specific name and each capable of storing a specific datatype.
  - Each column, also called a *field*, represents an attribute of the table.
- A *row* is a distinct set of values, one value for each column
  - Each row, also called a *record*, represents a single entity in the real world.
  - A column value might be empty, or null, for a particular row.
- A user accesses a subset of the data, the result set of a query, based on existing tables.
- The logical data structure of a database is determined by the relationships between the tables.

Consider the following as a table design for sports teams:

Team

Sport	**Team Name**	**Affiliated City**
Baseball	Rockies	Denver
Baseball	Astros	Houston
Baseball	Braves	Atlanta

Player

Player Name	**Player Number**	**Position**
Jeff Baker	10	Outfield
Matt Holiday	5	Outfield
Chris Ianetta	20	Catcher

Where Does SQL Fit In?

- The Structured Query Language (SQL) was first developed by IBM in the mid-1970s for their DB2 database.
  - It is typically pronounced as "sequel" or S...Q...L.
- SQL is the international standard language for relational database management systems.
  - SQL is considered a fourth-generation language, more abstract than Java, COBOL, etc.
    - SQL is a non-procedural language that emphasizes what to get, but not how to get it.
  - It is English-like and intuitive.
    - SQL statements look more like natural language than many other programming languages
  - SQL is robust enough to be used by:
    - Users with non-technical backgrounds.
    - Professional developers.
    - Database administrators.
- Each vendor has its own implementation of SQL as well as extensions for greater functionality.
  - MySQL implements a subset of SQL: 2008.

SQL statements can be placed in two main categories:

Data Manipulation Language (DML)		
Query	**Data Manipulation**	**Transaction Control**
SELECT	INSERT	COMMIT
	UPDATE	ROLLBACK
	DELETE	

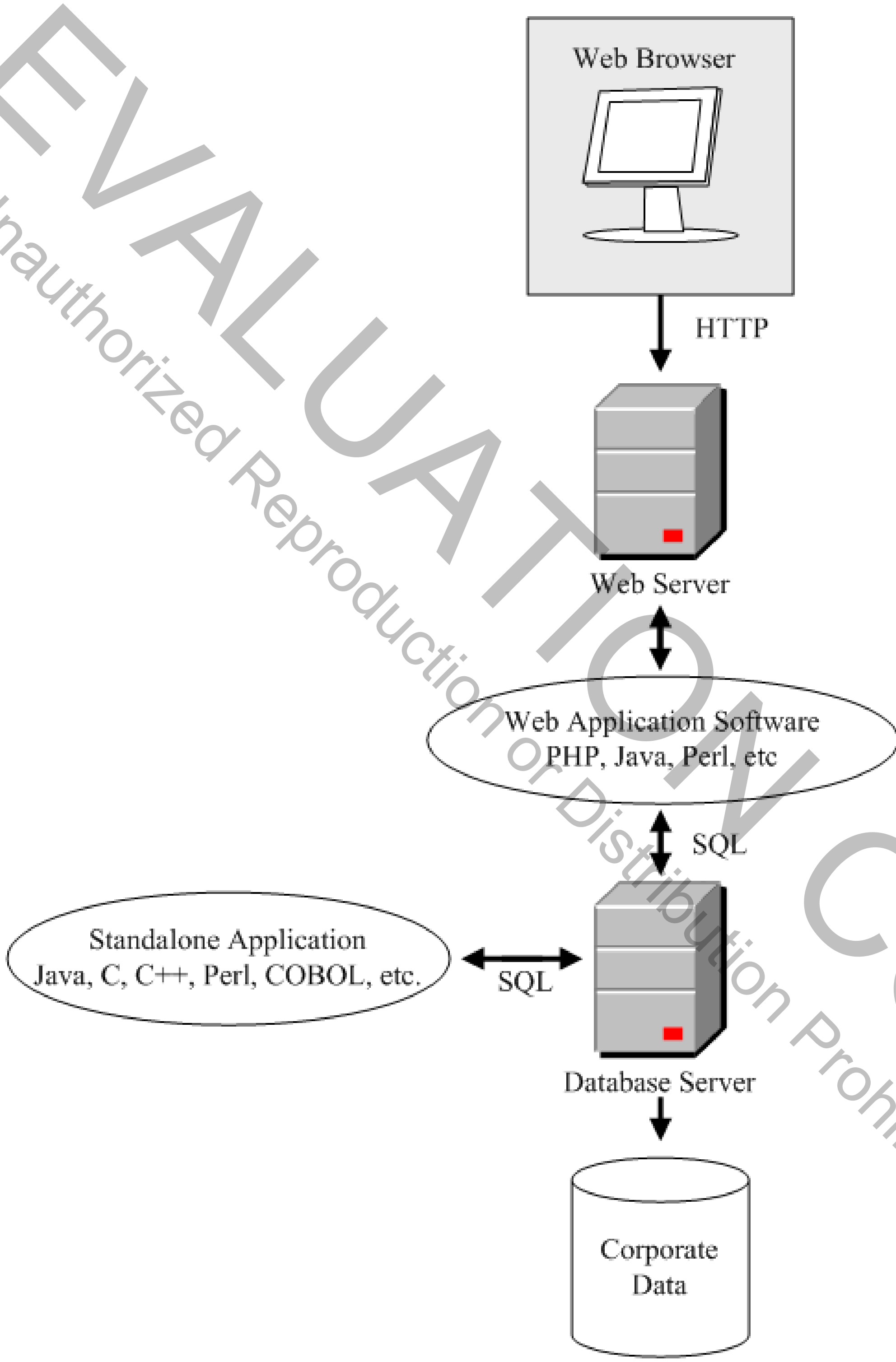
Data Definition Language (DDL)	
Data Definition	**Data Control**
CREATE	GRANT
ALTER	REVOKE
DROP	

Users and programmers spend most of their time working with only four of the keywords (**SELECT**, **INSERT**, **UPDATE**, and **DELETE**).

Standard SQL Versions		
Year	**Name**	**Alias**
1986	SQL-86	SQL-87
1989	SQL-89	FIPS 127-1
1992	SQL-92	SQL2, FIPS 127-2
1999	SQL:1999	SQL3
2003	SQL:2003	
2006	SQL:2006	
2008	SQL:2008	
2011	SQL:2011	

Database Access

- ❋ Before the World Wide Web, the only way to access a database was through a standalone software application.
  - ➢ Such applications may be provided by the database vendor, or can be written in C, C++, Perl, COBOL, Java, or other languages.
- ❋ With the World Wide Web, many databases are also accessed from internet-based applications.
  - ➢ The user is typically accessing the database through a web page loaded into a web browser.
  - ➢ Some of the same underlying languages might be in use on the server.
    - ▪ These languages have the ability to retrieve data from the database, and pass that data back to the web browser, through a web server.
- ❋ Access to the data is only available through the database server.
  - ➢ A valid username and password must be provided.


Web Browser
HTTP
Web Server
Web Application Software
PHP, Java, Perl, etc
SQL
Standalone Application
Java, C, C++, Perl, COBOL, etc.
SQL
Database Server
Corporate
Data

Why MySQL?

- MySQL is not a database, but a *Relational Database Management System* (RDBMS); it is software that enables you to create, maintain, and manage relational databases.

- MySQL is popular because of its:

  - Size — Very modest hardware requirements.

  - Speed — Some of the newer features can slow down processing, but MySQL's modular design allows you to pick and choose.

  - Ease of installation — Installation on some platforms is easy (e.g., Windows, or a Linux RPM), but others can be more complicated.

  - Attention to standards — Even though multiple standards exist for SQL, MySQL tries to stay current.

  - Community relations — MySQL is open-source software and encourages individual participation.

    - There is a large amount of user-contributed software available, and many local user groups.

  - Easy interface to other software — Software interfaces, or drivers, to MySQL exist for many programming languages.

The History of MySQL

- ❋ In 1994, Oracle, Sysbase, DB2, and Informix dominated the SQL-based RDBMS market with their powerful, very resource-intensive, and expensive systems.
  - ➢ Postgres was one of a few cheap database engines available, but wasn't SQL-driven.
- ❋ At about the same time, David Hughes developed a DBMS that used SQL to support his Ph.D. work, based on Postgres.
  - ➢ To enable SQL, he built a translator called mSQL that converted SQL to PostQUEL.
  - ➢ The Postgres engine was too ungainly for his needs, so he incorporated a DBMS into mSQL.
- ❋ In 1979, Michael Widenius, as the only database developer at TcX, developed a tool called UNIREG for managing databases.
  - ➢ In 1994, TcX began developing web-based applications, using UNIREG to support page generation, but UNIREG proved too ungainly.
  - ➢ Using the best features of UNIREG, and building on the mSQL API, MySQL 1.0 was released in May, 1995, by the Swedish company MySQL AB, founded by Widenius and David Axmark.
  - ➢ Shortly thereafter, MySQL was released on the Internet, under a flexible copyright (currently the GPL).
- ❋ In January, 2008, Sun bought MySQL AB.
- ❋ In the summer of 2009, Oracle bought Sun, acquiring the copyright for the MySQL product.

Some of MySQL's major features include the following:

- Internals and Portability:
  - Written in C and C++, and tested with a broad range of different compilers.
  - Works on many different platforms.
  - It can easily use multiple CPUs if they are available.
  - Uses very fast B-tree disk tables (MyISAM) with index compression.
  - Very fast joins using an optimized one-sweep multi-join.

- Statements and Functions (full support is provided for):
  - Operators and functions in **SELECT** list and **WHERE** clause of queries.
  - SQL **GROUP BY** and **ORDER BY** clauses.
  - **LEFT OUTER JOIN** and **RIGHT OUTER JOIN**.

- Security:
  - A privilege and password system that allows host-based verification.
  - All password traffic is encrypted when you connect to a server.

- Scalability and Limits:
  - Handles large databases (e.g., 60,000 tables and 2^{32} rows, depending on the storage engine.)
  - Up to 64 indexes per table are allowed, depending on the storage engine.

- Connectivity:
  - Clients can connect to the server using TCP/IP sockets, named pipes, shared-memory, or socket files, depending on the platform.
  - APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available.
  - Interfaces are available for ODBC (Connector/ODBC (MyODBC)), Java (Connector/J), .NET (Connector/NET), and others.

- Localization:
  - The server can provide error messages to clients in many languages.
  - All data is saved in the chosen character set.

- Clients and Tools:
  - Command-line tools for issuing SQL commands (mysql), dumping and importing table data (mysqldump and mysqlimport), table maintenance (mysqlcheck), etc.

  - A general purpose GUI tool (MySQL WorkBench) to aid in administration tasks, converting to/from alternative database technologies, and generating queries.

Chapter 2 - Installation, Configuration, and Upgrading

Objectives

- Describe the MySQL version numbering scheme and the different release types.
- Install MySQL.
- Configure the MySQL server.
- Upgrade a MySQL server.
- Move a MySQL database from one OS to another.

MySQL Software

- The MySQL platform is available on multiple systems including Windows and Linux.
- The MySQL version numbering scheme, typically written as *mysql-m.r.v-suffix*, conveys information about the content.
  - The *m* is the major version number.
    - All releases within a major version use the same disk file formats.
  - The *r* is the release level.
  - The combination of major version and release level is referred to as the *release series number*.
  - The *v* is the version number, incremented with each new release.
  - The *suffix* can be **alpha**, **beta**, **rc** (Release Candidate), or empty, indicating production quality code.
- Both binary and source versions of a given release are typically available.

The entry web pages into MySQL info are:

- ***http://www.oracle.com/us/products/mysql***
- ***http://www.mysql.com***

which provide starting points into:

- Advertising, documentation and white papers.
- Various products and services, partners and customers, news and industry reports.

The platform is supported on a variety of operating systems, but beware the list is rapidly being trimmed of older OSes. For details, see:

- ***http://www.mysql.com/support/supportedplatforms/database.html.***

A large variety of mailing lists is available at: ***http://lists.mysql.com***.

A wide variety of discussion forums is available at: ***http://forums.mysql.com***.

A bug database is available at: ***http://bugs.mysql.com***.

A splash page into contributed software, conferences, FAQs, etc. for the user-community is available at:

- ***https://wikis.oracle.com/display/mysql/Home***.

Note:

Navigating the MySQL web pages will often suggest you sign up for a free Oracle web account, which provides some interesting features, including among others:

1. Access to white papers and presentations
2. The ability to post in the MySQL Discussion Forums
3. Report and track bugs in the MySQL bug system (reading is available without an account).

MySQL Software Features

- ❋ Oracle provides MySQL software in several configurations, ranging from the free Community Server to the Cluster Carrier Grade Edition.
  - ➢ All support transaction-safe, ACID-compliant databases with full commit, rollback, crash recovery, and row-level locking capabilities.
- ❋ Oracle distinguishes between the use of MySQL as a technology embedded in a larger application/product and the use of MySQL as a stand-alone technology, and licenses the product accordingly.
- ❋ The MySQL Community Server is freely available under the open source GPL license.
- ❋ Support packages for all editions of MySQL can be purchased through Oracle.

The entry web pages into the free Community Edition Server can be found through:

- ***http://www.oracle.com/us/products/mysql/mysqlcommunityserver***
- ***http://www.mysql.com/products/community/***

The free Community Edition Server (and other packages) can be downloaded at:

- ***http://dev.mysql.com/download***

A list of editions and features suitable for an ISV, OEM, or VAR software product using an embedded database can be found at: ***http://www.mysql.com/oem***.

A list of editions and features suitable for stand-alone databases can be found at: ***http://www.mysql.com/products***.

A list of support options for all of Oracle's MySQL offerings is described in: ***http://www.mysql.com/support***. Various 3rd-party companies also offer MySQL support: SkySQL, Percona, etc.

Preparing to Install MySQL

- Installing is best done using a binary RPM package on a Linux machine.
  - RPM processing will take care of directory ownership, permissions, etc.
  - The Server and Client RPM packages are all that are typically required for a working system.
  - Non-RPM downloads are also available (e.g., *.tar.gz* files).
- After downloading your desired version, you should check that the download succeeded.
  - If your system has the **md5sum** (or possibly just **md5**) program available, run it and verify that the string of hex digits it prints matches that found on the download site.

```
md5sum package_name
```

    - Note that this should be done against the *.rpm*, *.tar.gz*, or *.zip* file, not on the unwrapped contents.
  - As an alternative to **md5sum**, GNU's Privacy Guard (gnuPG) can be used.
  - If downloading an RPM package, the integrity check built into RPM can be used.

```
rpm --checksig package_name.rpm
```

Installation packages for the MySQL server and other software are available from dev.mysql.com/downloads. As part of the download process you will select the target architecture and the version (if applicable).

When selecting a server version, note two things:

1. Oracle is aggressively trimming support for older versions on older machines. For example, they have already discontinued support for XP machines on the most up-to-date releases. In some cases, being "not supported" may mean the product won't install at all.

   For more details on supported architectures and versions, see: ***http://www.mysql.com/support/supportedplatforms/database.html***.

   You can also find the MySQL up-to-date end-of-life summary at: ***http://www.mysql.com/support/eol-notice.html.***

2. Multiple versions of the server are typically under development concurrently, implying an earlier version might be sufficient for your needs. A good illustration of what's going on can be found in the "Versions" section of: ***http://en.wikipedia.org/wiki/MySQL***.

Installation packages for a Windows system are either an .msi install file, or, for newer releases, a .zip file, which must be installed manually.

- Installation should be done using an "administrator" account.
- If tables larger than 4GB are required, install on an NTFS filesystem.

When installing on Windows, note there is a web page devoted to known limitations: ***http://dev.mysql.com/doc/refman/5.5/en/limits-windows.html***

Note:

A command line version of md5 is available for Windows from: www.fourmilab.ch/md5.
A GUI version is available from: www.nullriver.com/downloads/Install-winMd5Sum.exe.

To run gnuPG (obtained from: www.gnupg.org), you'll need a copy of the MySQL public GPG build key, available from: pgp.mit.edu. The key you want is named 'mysql-build@oss.oracle.com'. Save it in a local file. Import the key into your local environment using: gpg --import local_file_name. When downloading the mysql distribution, also download the signature file, which has a name similar to: package_name.asc. Verify your download with: gpg --verify package_name.asc. A more detailed explanation of this procedure can be found at: ***http://dev.mysql.com/doc/refman/5.5/en/checking-gpg-signature.html***.

Available Client Software

- A wide variety of client software is available as a MySQL download; here are some of the most commonly used.

- General Utilities

  - **mysql** — Command-line client for entering SQL statements and performing administrative operations.

  - **mysql-workbench** — A GUI tool to help build and work with databases. Notable features include a color-highlighted editor for working with SQL queries, and a console for visibility into servers, databases, and user accounts.

  - **mysqladmin** — a command-line tool for database administrative operations.

  - **mysqlshow** — Displays information about databases, tables, columns, and indexes.

  - **perror** — Displays the meaning of system or MySQL error codes.

- Input/Output Utilities

  - **mysqldump** and **mysqlimport** — Clients for dumping a database into a text file and importing such a file back into a database.

  - **mysqlhotcopy** — Quickly makes backups of MyISAM tables on Unix-y systems while the server is running.

- Table Maintenance Utilities

  - **mysqlcheck** and **myisamchk** — Table-maintenance clients that check, repair, analyze, and optimize tables.
  - **myisamlog** and **mysqlbinlog** — Utilities for reading and processing log files. Can be used to help recover from a crash.

The Importance of Backups

Utility programs such as **mysqlcheck**, **myisamchk**, **myisamlog**, and **mysqlbinlog** all make permanent changes to the disk. Being able to back out of such changes if something goes wrong is completely up to you.

Manipulating complicated tables, including those which govern access permissions, sometimes goes awry.

Operating systems have been known to crash, the power to fail, hardware to fail, etc.

Although Oracle would like releases to be compatible with each other, they are also open about the fact that that's not always the case. Table formats change, including that of system tables.

- Sometimes this is due to bug fixes.
- Sometimes this is the addition of new features.
- Sometimes this is due to what are referred to as incompatible changes.

Having a backup could be your only way to recover from any of these calamities.

Note:
MySQL provides tools to help simplify recovery of both transactional and non-transactional tables.

After the Download

- Standard RPM processing can be used to install on Linux.
  - For the server:

    ```
    rpm -i MySQL-server-version.i386.rpm
    ```

  - For client programs:

    ```
    rpm -i MySQL-client-version.i386.rpm
    ```

- Using RPM to install the server will run the following steps:
  - A **mysql** login account will be created (if one doesn't exist) that's used for running the MySQL server.
  - Changes will be made to */etc/init.d* to start the server at every boot.
  - The server will be started.

When installing on Windows using the installation wizard, there are several types of installs available.

The Windows installation wizard makes several changes to the system.

Performing a custom install on Windows allows you to choose to install from the following (depending on the package downloaded):

- MySQL Server
- Client Programs: MySQL Command Line Shell, MySQL Command Line Utilities, MySQL Server Instance Configuration tool
- MySQL Instance Manager
- Developer Components: C Include Files/Lib Files, Scripts, Examples
- You can also choose a target drive/directory where the software will be installed. The default is *C:\Program Files\MySQL\MySQL Server* version, for which all subsequent installs of MySQL software will be under *C:\Program Files\MySQL*.

Documentation is available for viewing online, and can be downloaded via a separate web page: ***http://dev.mysql.com/doc***. Additional documentation (e.g., for the sample databases) can be found at: ***http://dev.mysql.com/doc/index-other.html***.

Configuring the Server

- MySQL server execution is configured by a plain text file containing a variety of configuration values.

  my.cnf (UNIX filename)

```
[mysqld]
port=3306
basedir="C:/Program  Files/MySQL/MySQL  Server  5.5/"
datadir="E:/dshaw/ITC/mysql.data/Data/"
character-set-server=latin1
default-storage-engine=INNODB
max_connections=100
query_cache_size=0
...
```

  - The file is broken up into sections beginning with **[*group*]**, where ***group*** can be the name or the "group" of a MySQL utility program.
  - You can put any valid long options that can be given to a program at the command line into the configuration file.

- MySQL provides sample configuration files in INSTALLDIR for different sizes of databases. Release 5.5 will be the last release for this practice.

- A MySQL Server Instance Configuration Wizard is provided for all Windows versions.

Running the Configuration Wizard, at installation time or later, provides some choices.

- If a configuration file already exists, the wizard allows it to be:
  - Deleted, removing any existing service at the same time; the server **installation** and its data files are not removed.
  - Reconfigured, in which case the existing configuration file is renamed to a backup name.
- If a configuration file doesn't already exist, or an existing file is to be reconfigured, the wizard offers the following two types of configuration.
  - Standard Configuration — Okay for non-intensive use only.
  - Detailed Configuration — Allows you to make choices about the way MySQL will be used.
- Modify Security Settings.
  - It is strongly recommended to set a **root** password, for the local machine only.
  - Establishing an anonymous account is discouraged.

On Windows, the configuration file is named *my.ini*, and is located in the MySQL install directory.

In the server configuration file, "group" may also be the word "client", which will provide options for all *client* programs, thus excluding mysqld.

Each operating system platform has a different default installation directory:
Windows installs in *C:\Program Files\MySQL\MySQL Server version*.
Linux RPM installs in */*.
UNIX *tar.gz* is typically installed in */usr/local/mysql*, but can be placed in any directory.
UNIX source is typically installed in */usr/local*, but can be placed in any directory.

The more important installation items and their default installed sub-directories are:

Directory Contents	Windows	Linux	UNIX	Source
Client programs	*bin*	*/usr/bin*	*bin*	*bin*
The **mysqld** server	*bin*	*/usr/sbin*	*bin*	*libexec*
Log files, databases	*C:\Documents and Settings\All Users\Application Data\MySQL*	*/var/lib/mysql*	*data*	*var*
Utility scripts	*scripts*	*/usr/bin*	*scripts*	*bin*

These directories can be overridden.

Starting the Server

- On UNIX systems, including Linux, server startup is best accomplished via the startup script **mysqld_safe**, which invokes **mysqld**.

```
mysqld_safe --user=mysql  &
```

  - After starting the server, **mysqld_safe** monitors it and if it fails, tries to restart it.
  - Any errors generated during the run are written by default to:
    On Windows systems: “host_name.err” in the data directory.   On Unix-y systems: stderr

- A boot-level startup script named *mysql.server* is also available under Linux.
  - The Linux RPM will install this file in */etc/init.d* so it will be started during system boot.
    - On systems that support it, the server can be started using:

```
service mysql start
```

  - Depending on what Linux you're running, activate it to run at boot time using a command similar to:

```
chkconfig —level 345 mysql on
```

- To stop the server, use **mysqladmin**.

```
mysqladmin -u root -p shutdown
```

  - On systems that support it, use:

```
service mysql stop
```

On Windows, the server can be installed as a Windows service.

Before installing as a service, shut down any server currently running:

```
NET STOP MySQL
```

To establish the server as a Windows service (default servicename: **mysql**), issue the following command:

```
mysqld --install [servicename] [--defaults-file=filename]
```

The server must still be started manually within the current boot cycle, either from the Services utility, or by issuing the command:

```
NET START MySQL
```

In most cases, this will have been done by the installation wizard.

The Initial User Accounts

- The grant tables for the initial users are provided pre-configured when you install from an RPM.
    - Four accounts are created automatically, two for **root** and two for an anonymous account.
- When you run a client program, you can specify which user you are logging in as, and which host you are logging into.
    - Connecting to the host by hostname or IP address, even **127.0.0.1**, will use TCP/IP for communication.

On Windows systems, two accounts are created, **root** and an anonymous account. There is no distinction between connection types to the local system. However, accounts for connections from remote systems can optionally be created, one for **root**, and one for an anonymous account for the **test** database.

For all UNIX (non-Linux) installations, default initial permissions are installed by the server using the user ID **mysql**. If such an account doesn't exist, create one by changing directories to the top-level directory of the MySQL installation and issuing the command:

```
mysql_install_db --user=mysql
```

The server's data directory and the directories for the **mysql** and **test** databases are created. Note that **mysql_install_db** does not overwrite existing privilege tables. If you need to re-create the tables, stop the server, rename the data directory, and then run **mysql_install_db**. Restart the server. Verify the server is running and then shut it down by issuing the command:

```
bin/mysqladmin -u root shutdown
```

Start the server again.

Note:
If the **root** password is lost, it can be reset.

- On Windows, log in using an account with administrative privileges. Stop the MySQL server.
- On UNIX systems, log in as **root**. Issue a normal **kill** (not **kill -9**) to the **mysqld** process.

Create a text file containing the following command:

```
UPDATE mysql.user SET Password=PASSWORD('yournewpassword')
WHERE User='root';
FLUSH PRIVILEGES;
```

On Unix/Linux, start the server using:

```
  mysqld --init-file=mytextfile
```

On Windows, start the server using:

```
  mysqld_safe --init-file=mytextfile
```

After the server is running, delete the text file, then stop and restart the server normally.

Verifying Server Operation

- After a server is running, you should verify that it is working properly.
  - Determine the number of databases that have been created.

    ```
    mysqlshow
    ```

    - You should have **mysql** and **test** databases**.**
  - View the tables in one of the databases.

    ```
    mysqlshow mysql
    ```

    - There should be several tables including **host**, **db**, **user**, **tables_priv**, **columns_priv**, **procs_priv**, etc.
  - Simple queries agains the **mysql** database will print current values.

    ```
    $ mysql -e "SELECT host, db, user FROM db" mysql

    host |  db  | user
    -----+------+------
      %  | test%|
    ```

  - Print parameter values in use by the server.

    ```
    mysqladmin variables
    ```


There are several things you can try if the automated procedure doesn't run or fails to start the MySQL server.

- Inspect the error log; its location is specified in the configuration file.
- Verify that the server can access the data directory; filesystem permissions are what matter here.
- Try starting the server manually, using the command **mysqld --console**.
    - If the server still won't start, try running it as:
      mysqld --debug (on UNIX)
      or
      mysqld-debug (on Windows, if available)
- To enable logging on Windows, consider using the **--general_log/--general_log_file** options on mysqld (which will log for everyone), or, if you can communicate with the server, the **--tee** option on mysql, which will log for that run only.
- If you can communicate with the server, try running: **mysqladmin variables** to see what options the server is using.

Upgrading

- Always do a backup of all of your databases before upgrading, including the system databases.

- Special recommendations about an upgrade have been supplied by Oracle/ MySQL for every release series upgrade in the past.

  - Always consult the release notes and the news file, which provides details on the changes made.

    - Be careful about using configuration files from the previous release; they may need to be updated.

    - Beware of name collisions for functions — you may need to change their names.

- After upgrading, you should check (and repair, if necessary) your tables, with **mysql_upgrade**.

  - Options can be used to specify directory locations.

    ```
    mysql_upgrade --basedir=path --datadir=path
    ```

  - You will be prompted for the **root** password, if needed.

  - Internally, **mysql_upgrade** runs two commands: **mysqlcheck** and **mysql_fix_privilege_tables**.

    - These check all tables in all databases for incompatibilities, and if any problems are found, the table is repaired.

  - All tables are marked internally to indicate that they've been upgraded.

Upgrading on Windows requires several steps before installing the new version.

Stop the server.
If installed as a service, issue the command:

```
NET STOP MySQL
```

Otherwise issue the command:

```
mysqladmin -u root shutdown
```

If upgrading from a version installed without the Windows Installer:
Manually remove the previous installation.
If installed as a service, remove the Server Instance Manager:

```
mysqld --remove
```

If it is ever necessary to downgrade an installation, start by making a backup. The same data files should be usable if downgrading within the same release series. If downgrading to an earlier version, the recommended procedure is as follows:

1. Dump your tables using **mysqldump** in your starting version, eliminating those features not available in your ending version (e.g., triggers).
2. Reload the data using **mysqlimport** in your ending version.

Note:
Incompatible changes can occur within a release series, but are more prevalent between major version series. For example, moving from release series 4.1 to 5.0 introduced eight incompatible changes to the server, and fourteen incompatible SQL-related changes. Some examples of these changes include:

- For **BINARY** columns, the pad value and how it is handled was changed.
- User variables became case sensitive.
- Natural joins and joins with **USING** are processed according to the SQL:2003 standard.

For a general synopsis of what to watch out for when considering an upgrade, look for pages similar to the following: ***http://dev.mysql.com/doc/refman/5.5/en/mysql-nutshell.html***

There is an *extensive* discussion on installation and upgrading provided in Section 2.3 of the MySQL manual, addressing the *many* quirks of working with different OSes and different OS levels.

Copying a Database Between Architectures

- The **mysqldump** command creates a file of SQL statements to recreate a database or tables in a database.
  - Use options to specify a format other than SQL statements.
    - **--xml** will build an XML file.
    - **--tab=*path*** will produce a *.sql* file to build the table, and a tab-separated *.txt* file with the data.
  - Some non-MySQL database systems can import a custom formatted dump file, or read the SQL from a standard dump file (although it may contain MySQL-specific commands).
  - The **--compressed** option will compress the file, which can be uncompressed by the **mysql** client when it is read in.
    - The file can also be compressed and uncompressed using command-line tools outside of MySQL.
- The **mysql** command can read the SQL dump file piped into it to rebuild tables and data into an existing database.
- The **mysqladmin** command can create an empty database.
- All of these commands can take the **-h** option to specify that they will run on a remote host.
  - The **mysqld** server must be running on the remote host.
- Don't forget to copy the **mysql** (grant table) database.

MyISAM files can be copied directly if the same floating point format is used on both machines. MySQL will handle any byte-swapping issues.

Examples:

If both machines are running and accessible, the following commands can be used on the machine being copied from:

```
mysqladmin -h 'other_hostname' create db_name
mysqldump --opt db_name | mysql -h 'other_hostname' db_name
```

If both machines are running and accessible, the following commands can be used on the machine being copied to:

```
mysqladmin create db_name
mysqldump -h 'other_hostname' --opt --compress db_name |
mysql db_name
```

If both machines are not accessible at the same time, you can create an intermediate file and move it in a separate step:

```
mysqldump --quick db_name | gzip > db_name.gz
```

After transferring the file to the new machine, issue the commands:

```
mysqladmin create db_name
gunzip < db_name.gz | mysql db_name
```

Using **mysqlimport** and **mysqldump** is a faster way to move large tables.

```
mysqldump --tab=new_directory db_name
```

After transferring the resulting directory structure to the new machine, issue the commands:

```
mysqladmin create db_name
cat new_directory/*.sql | mysql db_name
mysqlimport db_name new_directory/*.txt
```

Environment Variables

- Environment variables can override some built-in values that affect runtime operations.
- Default values for many runtime options are compiled into **mysql** programs, but can be overridden at runtime.
  - Overrides come from the value of environment variables, option files, and command-line arguments.

Variable	Description
HOME	Typically used to find **mysql_history**.
TZ	This should be set to your local time zone for running **SELECT NOW** queries and for running **UNIX_TIMESTAMP()**.
UMASK_DIR	The user-directory creation mask when creating directories. Note that this is and-ed to your **UMASK** value.
MYSQL_HOST	The default hostname used by the **mysql** command-line client.
MYSQL_TCP_PORT	The default TCP/IP port number.
MYSQL_UNIX_PORT	The default socket filename; used for connections to localhost.
MYSQL_GROUP_SUFFIX	Option group suffix value (like specifying **--defaults-group-suffix**).
MYSQL_HISTFILE	The path to the **mysql** history file. Default value: *$HOME/.mysql_history*.
MYSQL_HOME	The directory that contains the server-specific *my.cnf* file.
TMPDIR	The directory where temporary files are created.
MYSQL_PS1	The command prompt to use in the **mysql** command-line client.

Windows supports the following environment variables:

- **MYSQL_HOST**
- **MYSQL_GROUP_SUFFIX**
- **TMPDIR**
- **MYSQL_PS1**
- **USER** — default username when connecting to **mysqld**.

Some values are overridden using command-line options rather than environment variables. These are:

- **TZ**
- **MYSQL_TCP_PORT**
- **MYSQL_UNIX_PORT**

Labs

❶ (Optional) Download the Community Edition of MySQL for your target platform.
Note: You'll need version 5.0 or later.
(Solution: *download.txt*)

❷ Verify the contents of the downloaded file(s).
(Solution: *verify.txt*)

❸ Install MySQL, including both the server and the client on Unix/Linux systems.
(Solution: *install.txt*)

❹ Verify that the server is running, and that you can stop and restart the server.
(Solution: *server.txt*)

itcourseware
TRAINING MATERIALS FOR IT PROFESSIONALS

7400 E. Orchard Road, Suite 1450 N
Greenwood Village, Colorado 80111
Ph: 303-302-5280
www.ITCourseware.com

9-38-00181-000-09-17-13