mysql_fix_privilege_tables

`mysql_fix_privilege_tables`

At various points in time, the user security database `mysql` underwent some changes: the complexity of the passwords was changed, more privileges were added, etc. To make upgrading an existing database easier, you can use this utility to implement the changes between versions. Be sure to restart the MySQL server when you are finished running this utility for the changes to take effect. As of version 5.0.19 of MySQL, this utility has been replaced by `mysql_upgrade`. It performs the same functions and has other capabilities.

The only options for the program are `--password`, in which the *root* password is given, and `--verbose` to display more information when running the program.

This program is not available on Windows systems. However, there is an SQL file, *mysql_fix_privilege_tables.sql*, that may be run with the `mysql` client as *root* to perform the same tasks. The SQL file is located either in the *scripts* or the *share* directory where MySQL is installed.

mysql_setpermission

`mysql_setpermission options`

This utility is an interactive Perl program that allows an administrator to set user privileges. To run the program, you would typically give the `--user` option with the administrative username so you can set privileges. A text menu of options will be displayed for a variety of user administration tasks, including setting the password and privileges for an existing user and creating a new user. The program requires that Perl and the Perl DBI module be installed on the system where it's executed.

mysql_setpermission options

`--host`
> This option specifies the name or IP address of the server for connection.

`--help`
> This option displays help information about the utility.

`--password=password`
> This option provides the password of the administrative user with which the utility is to log into the server, not the user for which to change privileges.

`--port=port`
> This option specifies the port number to use for connecting to the server.

`--user=user`
> This option provides the administrative username for logging into the server, not the user for which to change privileges.

`--socket=filename`
> This option provides the name of the server's socket file.

mysql_tableinfo

mysql_tableinfo *options new_database* [*existing_database* [*existing_table*]]

This utility creates a table containing information about existing tables in a database. You have to specify the database that will contain the newly created metadata tables, and the utility will create the database if it does not exist. If given the name of an existing database, it will use its metadata as its basis. If also given a table name, it will refer to its metadata.

The utility will create four tables in the database: db, col, idx, and tbl. This last table may be named tbl_status instead. It uses the SHOW COLUMNS, SHOW DATABASES, SHOW INDEXES, SHOW TABLES, and SHOW TABLE STATUS statements to get metadata information. The user given must have the necessary privileges for these statements.

mysql_tableinfo options

--clear

This option drops all four tables to be created by the utility if they exist, before creating new ones and populating them.

--clear-only

This option drops all four tables to be created by the utility if they exist, but doesn't create new ones—the utility will exit when it's finished deleting the tables.

--col

This option puts column metadata into the col table.

--help, -?, -I

This option displays help information about the utility.

--host=*host*, -h *host*

This option specifies the host on which to obtain metadata information and to create tables.

--idx

This option puts index metadata into the idx table.

--password=*password*, -p *password*

This option provides the password of the user logging into the server.

--port=*port*, -P *port*

This option specifies the port on which to connect to the server. The default is 3306.

--prefix=*string*

This option adds a prefix to the names of the tables that the utility creates (e.g., metadata_db instead of db).

--quiet, -q

This option suppresses all messages except for error messages.

--socket=*filename*, -S *filename*

This option provides the name of the server's socket file.

--tbl-status

This option takes metadata from the SHOW TABLE STATUS statement instead of SHOW TABLES. The result is more metadata but a slower process.

--user=*user*, -u *user*

This option provides the username for logging into the server.

mysql_upgrade

mysql_upgrade *options*

Use this utility after upgrading to a new version of MySQL. It checks all tables for version incompatibilities or problems, and attempts to repair or correct tables if possible. It also updates tables in the mysql database for new privileges and other factors available in the newer version of MySQL. Tables that are checked are tagged for the new version so they won't be checked twice. The utility notes the version number in the *mysql_upgrade_info* file located in the data directory for MySQL. This utility replaces the mysql_fix_privilege_tables utility because it performs the same function and more.

mysql_upgrade options

--basedir=*path*
> This option specifies the base directory of the MySQL server.

--datadir=*path*
> This option specifies the data directory of the MySQL server.

--debug-check
> This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info, -T
> This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--force
> This option forces the utility to check tables despite the *mysql_upgrade_info* file indicating that the tables are marked the same as the version noted in that file.

--help, -?, -I
> This option displays help information about the utility.

--password=*password*, -p *password*
> This option provides the password of the user logging into the server.

--user=*user*, -u *user*
> This option provides the username for logging into the server. If the option is not given, the *root* user is assumed by default.

--verbose
> This option displays more information from the utility.

mysql_waitpid

mysql_waitpid *options PID wait_time*

This utility uses the Unix system kill command to terminate the process identified by a given process identification number, and to wait for termination for the time given in seconds. The process identified and the seconds given must be positive integers. It returns 0 on success or if the process didn't exist. It returns 1 after timeout.

The only options available for this utility are for help (--help, -?, -I), the version number (--version, -V), and verbosity (--verbose, -v).

mysql_zap

mysql_zap [*options*] *pattern*

Use this utility to kill processes based on a given pattern.

mysql_zap options

--help, -?, -I
   This option displays help information about the utility.

-f
   This option forces the utility to kill the processes without confirming the action with the user first.

-signal
   This option specifies the type of kill: TERM (or signal 15) or KILL (signal 9). You can give either the name or the number for the kill type. Notice that there is only one dash, not two with this option, and it must be entered before the other options.

-t
   This option tests the patterns given without killing the processes.

mysqlaccess

mysqlaccess [*host* [*user* [*database*]]] [*options*]

This utility checks the privileges that a user has for a specific host and database. One use is to run it as a preliminary tool to check for user permissions before proceeding with a customized program that uses one of the APIs.

If MySQL was not installed in the default location for the version you're using, you'll have to set the variable MYSQL in the mysqlaccess script. Change it with a plain text editor. Look for the following line (near the beginning) and change the file path to where the mysql client is located:

```
$MYSQL = '/usr/local/mysql/bin/mysql';  # path to mysql executable
```

With regard to the syntax, the hostname is the first argument and is optional. If not given, *localhost* is assumed. The username given in the second argument is the name of the user for which the utility is checking privileges. The third argument is the database against which to check privileges. The fourth argument involves several possible options, one of which could be the username by which the utility will access the server to gather information on the user named in the second argument. Here is an example of how you might use this utility:

```
mysqlaccess localhost marie workrequests -U russell -P
```

In this example, I give the utility the hostname, then the user I'm inquiring about, then the database name for which I want user privilege information. The -U option specifies the username with which to access the server to gather information. This user has full access to the mysql database. The -P instructs the utility to prompt me for a password.

Here are the results of the preceding inquiry:

```
Access-rights
for USER 'marie', from HOST 'localhost', to DB 'ANY_NEW_DB'
```

```
+------------------+---+ +------------------+---+
| Select_priv      | Y | | Shutdown_priv    | N |
| Insert_priv      | N | | Process_priv     | N |
| Update_priv      | N | | File_priv        | N |
| Delete_priv      | N | | Grant_priv       | N |
| Create_priv      | N | | References_priv  | N |
| Drop_priv        | N | | Index_priv       | N |
| Reload_priv      | N | | Alter_priv       | N |
+------------------+---+ +------------------+---+
NOTE:    A password is required for user 'reader' :-(
The following rules are used:
db    : 'No matching rule'
host  : 'Not processed: host-field is not empty in db-table.'
user:'localhost','marie','6ffa06534985249d','Y','N','N','N',
'N','N','N','N','N','N','N','N','N','N'
```

First, a table is presented that displays the privileges for the combination of the database named, the host given, and the user. This user has only SELECT privileges.

Additionally, the results are given in raw form for each component. This user's privileges are the same for all databases and hosts (i.e., there are no entries in the db or the host tables in the mysql database), so there aren't any results for those particular components. For the user component, the command displays details without labels, but they are presented in the order that they are found in the user table in the mysql database. The third field is the password in the encrypted format in which it is stored. The Ys and Ns are the settings for each user privilege.

Here is a list of options available for this utility in alphabetical order:

--brief, -b
: This option provides a brief display of results from an inquiry.

--commit
: This option copies grant rules from temporary tables to the grant tables.

--copy
: This option reloads temporary tables with original data from the grant tables so that privileges take effect.

--db=*database*, -d *database*
: This option explicitly specifies the database against which to query the user privileges.

--debug=*level*
: This option sets the debugging level. The choices are from 0 to 3.

--help, -?
: This option displays basic help information.

--host=*host*, -h *host*
: This option specifies the host on which to obtain privilege information. The localhost is the default.

--howto
: This option displays basic examples of usage with sample results.

--old-server

    This option stipulates that the server to which the utility is connecting is running an older version of MySQL (prior to 3.21), requiring a different method with regard to WHERE clauses in SQL statements.

--password=*password*, -p *password*

    This option provides the password of the user logging into the server, not the user on which to check for privileges.

--plan

    This option displays plans for further development of the utility by its developers.

--preview

    This option displays the differences in temporary grant tables before they are committed.

--relnotes

    This option displays notes on each release of the utility.

--rhost=*host*, -H *host*

    If the utility is not being run on the same server as the MySQL server that's being queried, use this option to specify the address of the MySQL server to query.

--rollback

    This option undoes the last change to user privileges.

--spassword=*password*, -P *password*

    This option provides the password when using a superuser.

--superuser=*user*, -U *user*

    This option provides a superuser's username.

--table, -t

    This option displays data in an ASCII table format.

--user=*user*, -u *user*

    This option provides the username for logging into the server, not the user on which to check for privileges.

--version, -v

    This option displays the version of the utility.

mysqladmin

`mysqladmin [options] command [command_options]`

This utility allows you to perform MySQL server administration tasks from the command line. You can use it to check the server's status and settings, flush tables, change passwords, shut down the server, and perform a few other administrative functions. This utility interacts with the MySQL server.

Here is an alphabetical list of options that you can give as the first argument to the utility:

--character-sets-dir=*path*

    This option specifies the directory that contains character sets.

--compress, -C

    This option compresses data passed between the utility and the server, if compression is supported.

--connect_timeout=*number*
> This option sets the number of seconds a connection may be idle before it will time out.

--count=*number*, -c *number*
> This option specifies the number of iterations of commands to perform in conjunction with the --sleep option.

--debug=*options filename*, -# *options, filename*
> This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list of options under the mysqldump utility for an explanation of these flags and others that may be used.

--debug-check
> This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info
> This option adds debugging information and CPU and memory usage information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--default-character-sets-dir=*path*
> This option specifies the directory that contains the default character sets.

--force, -f
> This option forces execution of the DROP DATABASE statement and others despite error messages.

--help, -?
> This option displays basic help information.

--host=*host*, -h *host*
> This option specifies the name or IP address of the server for connection.

--no-beep
> This option instructs the utility not to emit a warning sound for errors. It was added as of version 5.1.17 of MySQL.

--password[=*password*], -p[*password*]
> This option provides the password to give to the server. No spaces are allowed between the -p and the password. If a password is not given, the user will be prompted for one.

--port=*port*, -P *port*
> This option specifies the port on which to connect to the server. The default is 3306.

--relative, -r
> This option displays the differences between values with each iteration of commands issued with the --sleep option.

--shutdown_timeout=*number*
> This option sets the number of seconds the client should wait before shutting down.

--silent, -s
> This option tells the utility to exit without error messages if a connection to the server cannot be established.

--sleep=*seconds*, -i *seconds*
> This option specifies the number of seconds to wait between the repeated execution of commands. The number of iterations is set by the --count option.

`--socket=`*filename*`, -S `*filename*
> This option provides the name of the server's socket file.

`--ssl`
> This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled.

`--ssl-ca=`*pem_file*
> This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

`--ssl-capath=`*path*
> This option specifies the path to the trusted certificates file (i.e., the *pem* file).

`--ssl-cert=`*filename* ·
> This option specifies the name of the SSL certificate file to use for SSL connections.

`--ssl-cipher=`*ciphers*
> This option gives a list of ciphers that may be used for SSL encryption.

`--ssl-key=`*filename*
> This option specifies the SSL key file to use for secure connections.

`--ssl-verify-server-cert`
> This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

`--start-slave`
> This option is issued on a slave server to start replication.

`--stop-slave`
> This option is issued on a slave server to stop replication.

`--user=`*user*`, -u `*user*
> This option specifies a MySQL user other than the current filesystem user.

`--verbose, -v`
> This option displays more information.

`--version, -V`
> This option displays the version of the utility.

`--vertical, -E`
> This option displays output in a vertical format with a separate line for each column of data.

`--wait[=`*number*`], -w [`*number*`]`
> This option instructs the utility to wait until it can connect to the server. It will retry once unless the number of times it is to retry is given with this option.

mysqladmin commands

The main focus of `mysqladmin` is the commands that perform administrative tasks. Commands are given as the second argument. You can issue one or more commands on the same line. Here is an alphabetical list of commands (with options for some) and an explanation of each:

`create` *database*
> This command creates the new database specified.

❀ debug

> This command enables debugging of the utility. It writes debugging information to the error log.

❀ drop *database*

> This command deletes the database specified.

❀ extended-status

> This command displays the MySQL server's extended status information.

❀ flush-hosts

> This command flushes all cached hosts.

❀ flush-logs

> This command flushes all logs.

flush-privileges

> This command reloads the grant tables.

flush-status

> This command flushes status variables.

flush-tables

> This command has the utility flush all tables.

flush-threads

> This command flushes the thread cache.

kill *id*

> This command kills the server thread specified by an identifier. Additional threads may be given in a comma-separated list.

old-password *password*

> This command changes the password of the user currently connected to the server through the utility to the password given, but in the older encryption method prior to version 4.1 of MySQL.

password *password*

> This command changes the user's password to the given *password*. Only the password for the user connecting to the server can be changed.

ping

> This command determines whether the server is running.

processlist

> This command displays a list of active server threads. With the --verbose option, more information is provided on each thread.

refresh

> This command flushes all tables and reloads log files.

reload

> This command reloads the grant tables.

shutdown

> This command shuts down the MySQL server.

start-slave

> This command starts a replication slave server.

status

> This command displays the server's status.

stop-slave
> This command stops a replication slave server.

variables
> This command displays the variables and the values of the server.

version
> This command displays the version of the utility.

mysqlbinlog

mysqlbinlog [*options*] *filename*

This utility formats the display of the binary log for a MySQL server. Customized applications can also use it for monitoring server activities. The path to the log file to format is given as the second argument for the utility. Additional log files may be given either with filesystem wildcards or by listing them individually, separated by spaces.

Here is an alphabetical list of the options, along with a brief explanation of each:

--base64-output
> This option is used to write binary log entries using base-64 encoding. This is used for debugging and should not be used in production. It's available as of version 5.1.5 of MySQL.

--character-sets-dir=*path*
> This option specifies the directory containing character sets.

--database=*database*, -d *database*
> This option displays information regarding only the database given.

--debug[=*options*], -# [*options*]
> This option logs debugging information, along with various settings (e.g., 'd:t:o,logname').

--debug-check
> This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info
> This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--disable-log-bin, -D
> This option disables binary logging.

--force-read, -f
> This option forces the reading of unknown log information.

--hexdump, -H
> This option dumps the log in hexadecimal format.

--help, -?
> This option displays basic help information.

--host=*host*, -h *host*
> This option specifies the hostname or IP address of a remote server containing the log file to format.

Command-Line Utilities

--local-load=*path***, -l** *path*
> This option specifies the local directory in which temporary files are to be prepared for LOAD DATA INFILE statements.

--offset=*number***, -o** *number*
> This option skips *number* entries at the start of the log file before starting the display.

--open_files_limit
> This option sets the maximum number of open files allowed. The default is 64.

--password=*password***, -p** *password*
> This option provides the password to the remote server that is being accessed.

--port=*port***, -P** *port*
> This option specifies the port to use for connecting to a remote server.

--position=*number***, -j** *number*
> This option sets the number of bytes to skip at the beginning of the log file. It is deprecated; use --start-position instead.

--protocol=*protocol*
> This option specifies the protocol to use when connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--read-from-remote-server, -R
> This option reads the binary log from a remote server instead of the local machine. You will need to include the necessary options for connecting to a remote server: --host, --password, and --user. You might also need to include --port, --protocol, and --socket.

--result-file=*filename***, -r** *filename*
> This option redirects the results of the utility to a given file.

--server-id=*identifier*
> This option returns entries from the binary log that were generated by a connection matching the given process identifier number. This option is available as of version 5.1.4 of MySQL.

--set-charset=*character_set*
> This option adds a SET NAMES statement to the results to indicate the character set used. It is available as of version 5.1.12 of MySQL.

--skip-write-binlog
> This option disables the --write-binlog option, which is enabled by default. Otherwise, ANALYZE TABLE, OPTIMIZE TABLE, and REPAIR TABLE statements executed by the utility will be written to the binary log. It's available as of version 5.1.18 of MySQL.

--short-form, -s
> This option changes the output to a shorter format.

--socket=*filename***, -S** *filename*
> This option provides the name of the server's socket file for Unix systems, piped name for Windows systems.

--start-datetime=*datetime*
> This option begins reading the log from the first event recorded with a date and time equal to or greater than the one given. The time can be in DATETIME or TIMESTAMP format. Use the time zone of the server.

--start-position=*number*
> This option sets the position to start reading the log file.

--stop-datetime=*datetime*
> This option instructs the utility to stop reading the log at the first event recorded with a date and time equal to or greater than the one given. The time can be in DATETIME or TIMESTAMP format. Use the time zone of the server.

--stop-position=*number*
> This option sets the position to stop reading the log file.

--table=*table*, -t *table*
> This option obtains information on the table named.

--to-last-log, -t
> This option instructs the utility to continue on in sequence reading through all binary logs, starting with the one given until the last log file is processed.

--user=*user*, -u *user*
> This option specifies the username to use when connecting to a remote server.

--version, -V
> This option displays the version of the utility.

--write-binlog
> With this option, ANALYZE TABLE, OPTIMIZE TABLE, and REPAIR TABLE statements executed by the utility will be written to the binary log. It's available as of version 5.1.18 of MySQL and is enabled by default. To disable it, use --skip-write-binlog.

mysqlbug

mysqlbug

This is a script you can use to report bugs to MySQL AB developers. Executed at the command line of the server, this script gathers information on the version of MySQL and related libraries installed, the operating system, as well as how MySQL was compiled.

To run the utility, simply type the command without any options or arguments. After a few moments, a text editor (e.g., Emacs) will be started with a form for reporting the bug. Several of the details will be filled in with information gathered by the script. You can modify this information, and you are expected to answer questions about the bug discovered. This includes a description of how to reproduce the problem or what circumstances occurred that may have caused or contributed to the problem. If you discovered a workaround solution, report this as well. The report created (saved in the */tmp* directory on Unix systems) should be emailed to *dev-bugs@mysql.com.* Go to *http://bugs.mysql.com* to report bugs online.

mysqlcheck

mysqlcheck [*options*] *database* [*table*]

This utility checks, repairs, and optimizes MyISAM tables. It works in part on tables for other storage engines as well. It uses the ANALYZE TABLE, CHECK TABLE, OPTIMIZE TABLE, and REPAIR TABLE statements. Therefore, if the storage engine supports any of these statements, the operations that can be performed by supported statements can be done using this utility. For MyISAM tables, this utility is similar in use and purpose to

myisamchk. Instead of working with the table files directly as myisamchk does, though, this utility interacts with the MySQL server.

The name of the database containing the tables to check is given as the second argument to the utility. The table to check is given as the third argument. Additional tables may be given in a space-separated list.

Here is a list of options that you can give and a brief explanation of each:

--all-databases, -A
 This option checks all databases.

--all-in-1, -1
 This option executes all queries for all tables in each database in one statement rather than as separate queries for each table.

--analyze, -a
 This option analyzes tables.

--auto-repair
 This option automatically repairs any corrupted tables found.

--character-sets-dir=*path*
 This option specifies the directory containing character sets.

--check, -c
 This option checks tables for errors.

--check-only-changed, -C
 This option checks only tables that have changed since the last check, as well as tables that were not closed properly.

--compress
 This option compresses data passed between the utility and the server, if compression is supported.

--databases *databases*, -B *databases*
 This option specifies more than one database for checking. To specify tables along with databases with this option, use the --tables option.

--debug[=*options*], -# [*options*]
 This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list of options under the mysqldump utility for an explanation of these flags and others that may be used.

--debug-check
 This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info
 This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--default-character-set=*set*
 This option specifies the default character set. Enter SHOW CHARACTER SET; on the server for a list of character sets available.

--extended, -e

   This option ensures consistency of data when checking tables. When repairing tables with this option, the utility will attempt to recover all rows, including intentionally deleted ones.

--fast, -F

   This option checks only tables that were improperly closed.

--fix-db-names

   This option converts the names of databases that contain characters no longer permitted by MySQL as of version 5.1. It's available as of version 5.1.7 of MySQL.

--fix-table-names

   This option converts the names of tables that contain characters no longer permitted by MySQL as of version 5.1. It's available as of version 5.1.7.

--force, -f

   This option forces processing of tables regardless of SQL errors encountered.

--help, -?

   This option displays basic help information.

--host=host, -h host

   This option specifies the name or IP address of the server for connection.

--medium-check, -m

   This option is more thorough than --check but less thorough than --extended.

--optimize, -o

   This option optimizes tables.

--password[=password], -p[password]

   This option provides the password to pass to the server. A space is not permitted after -p if the password is given.

--port=port, -P port

   This option specifies the port to use for connecting to the server. The default is 3306.

--protocol=protocol

   This option specifies the protocol to use when connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--quick, -q

   This option checks tables faster by not scanning rows for incorrect links. When used to repair tables, it has the utility repair only the index tree. This option is the fastest method.

--repair, -r

   This option repairs tables. Note that it can't repair unique keys containing duplicates.

--silent, -s

   This option suppresses all messages except for error messages.

--socket=filename, -S filename

   This option provides the name of the server's socket file.

--ssl

   This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled. If this option is enabled on the utility by default, use --skip-ssl to disable it.

`--ssl-ca=pem_file`

This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

`--ssl-capath=path`

This option specifies the path to the trusted certificates file (i.e., the *pem* file).

`--ssl-cert=filename`

This option specifies the name of the SSL certificate file to use for SSL connections.

`--ssl-cipher=ciphers`

This option gives a list of ciphers that may be used for SSL encryption.

`--ssl-key=filename`

This option specifies the SSL key file to use for secure connections.

`--ssl-verify-server-cert`

This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

`--tables`

This option specifies table names when using the `--databases` option.

`--use-frm`

This option uses the table structure in the *.frm* file for repairing a corrupted index.

`--user=user, -u user`

This option specifies the username for connecting to the server.

`--verbose, -v`

This option displays more information.

`--version, -V`

This option displays the version of the utility.

mysqldump

```
mysqldump [options] --all-databases
mysqldump [options] --databases database [database ...]
mysqldump [options] database [table]
```

This utility exports MySQL data and table structures. Typically, you use it to make backups of databases or to copy databases from one server to another. You can run it on an active server. For consistency of data between tables, the tables should be locked (see the `--lock-tables` option) or the `mysqld` daemon should be shutdown.

There are three syntaxes for this utility. The first method shown makes a backup of all databases for the server. The second method backs up specific databases, named in a space-separated list, including all tables in each database. The third method backs up specific tables of a specific database.

Here is an example using the first method, backing up all databases on the server:

```
mysqldump --host=russell.dyerhouse.com --user=russell --password \
    --lock-tables --all-databases > /tmp/workrequests.sql
```

Because the backup is being run from a remote server (i.e., not the localhost), the `--host` option is given with a domain name address for the host. An IP address could be given instead. Making a backup remotely like this will work only if the host grants the

necessary privileges to user *russell* with the host from which mysqldump is running. The example redirects the results with a greater-than sign to a text file.

To make a backup of a specific database, use the second syntax for this utility. Enter something like the following from the command line:

```
mysqldump -u russell -p --lock-tables workrequests > /tmp/workrequests.sql
```

In this example, the username is given with the -u option. The -p option tells the utility to prompt the user for a password. These shorter options are interchangeable with their longer, more verbose ones, but the verbose ones are becoming the norm and should be used. The --lock-tables option has the server lock the tables, make the backup, and then unlock them when it's finished. Next, we specify the database to back up (workrequests). Finally, using the redirect (the greater-than sign), the output is saved to the filename given.

The --lock-tables option is generally not necessary because the --opt option is a default option and includes locking tables. In fact, if you're making a backup and you do not have the LOCK TABLES privilege, you will receive an error when running mysqldump because of --opt. In such a situation, you'll need to include the --skip-opt option to specifically disable --opt and thereby not attempt to lock the tables.

If you want to back up specific tables and not an entire database, you can use the third syntax shown at the start of this section for this utility. It's not a very verbose syntax: you simply give the name of the database followed by one or more tables. You don't identify them individually as a database versus tables; you just put them in the proper order without the --all-database option. Here's an example of this syntax:

```
mysqldump -u russell -p workrequests work_req clients >
    /tmp/workreq_clients_tables.sql
```

In this example, the database is workrequests and the tables to be backed up are work_req and clients. Their table structures and data will be copied into the text file *workreq_clients_tables.sql*.

The backup or dump file created by mysqldump will be in the text file format. It generally will contain a CREATE TABLE statement for each table in the database. If you want to eliminate the CREATE TABLE statements, add the --no-create-info. If they are not included in the dump file generated on your server, add the --create-options option and run mysqldump again. The dump files will also generally contain a separate INSERT statement for each row of data. To back up the data faster, you can add the --extended-insert option so that only one INSERT with multiple values will be generated for each table instead of separate INSERT statements for each row of data.

To restore the data from a dump file created by mysqldump, you can use the *mysql* client. To restore the file created by the preceding statement, you can enter the following from the command line:

```
mysql -u russell -p < /tmp/workrequests.sql
```

This example redirects the stdin by means of the less-than sign. This instructs the *mysql* client to take input from the file given. It will execute the series of SQL statements contained in the dump file. You won't be placed into monitor mode; you will remain at the command line until it's finished.

Command-Line Utilities

You can determine the contents of the dump file by the options you choose. Following is an alphabetical list of options, along with a brief explanation of each. For some options, there is a shorter, less verbose version (i.e., -u for --user). These shorter options are interchangeable with their longer, more verbose ones, but the verbose ones are becoming the norm and should be used.

mysqldump options

--add-drop-database
>This option adds a DROP DATABASE statement followed by a CREATE DATABASE statement to the export file for each database, thus replacing the existing database and data if restored.

--add-drop-table
>This option adds a DROP TABLE statement to the export file before each set of INSERT statements for each table.

--add-locks
>This option adds a LOCK statement before each set of INSERT statements and an UNLOCK after each set.

--all, -a
>This option includes all MySQL-specific statements in the export file. This option is deprecated as of version 4.1.2 of MySQL. It is replaced with the --create-options option.

--all-databases, -A
>This option exports all databases.

--all-tablespaces, -Y
>This option is used with MySQL Cluster so that the utility will include the necessary SQL statements related to the NDB storage engine. This option is available as of version 5.1.6 of MySQL.

--allow-keywords
>This option makes keywords allowable for column names by including the table name and a dot before such column names in the export file.

--character-sets-dir=*path*
>This option specifies the directory containing character sets.

--comments[=0|1], -i
>If this option is set to a value of 1 (the default), any comments from a table's schema will be included in the export file. If it is set to 0, they won't be included. To disable this option since it's the default, use the --skip-comments option.

--compact
>This option omits comments from the dump file to make the file more compact. It also calls the --skip-add-drop-table, --skip-add-locks, --skip-disable-keys, and --skip-set-charset options. Don't confuse this option with --compress. Before version 5.1.2 of MySQL, this option did not work with databases that contained views.

--compatible=*type*
>This option makes the export file's contents compatible with other database systems. The choices currently are: ansi, mysql323, msyql40, postgresql, oracle, mssql, db2, maxdb (or sapdb for older versions), no_key_options, no_table_options, and

no_field_options. More than one type may be given in a comma-separated list. This option is used with version 4.1.0 of MySQL or higher.

--complete-insert, -c
This option generates complete INSERT statements in the export file.

--compress, -C
This option compresses data passed between the utility and the server, if compression is supported.

--create-options
This option includes all MySQL-specific statements (e.g., CREATE TABLE) in the export file. It's synonymous with the --all option.

--databases, -B
This option names more than one database to export. Table names may not be given with this option unless using the --tables option.

--debug[=options], -#[options]
This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list of options for this utility for an explanation of these flags and others that may be used. Here is an example of how you might use this option:

```
mysqldump -u russell -p --debug='d:f:i:o,/tmp/mysql_debug.log'
    workrequests > /tmp/workrequests.sql
```

--debug-check
This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info
This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--default-character-set=set
This option specifies the default character set for the utility to use. Execute SHOW CHARACTER SET from MySQL on the server to get a list of possibilities. By default, recent versions of the utility use UTF-8. Previous versions used Latin 1.

--delayed-insert
This option adds the DELAYED keyword to INSERT statements in the export file. In older versions of mysqldump, this option was --delayed.

--delete-master-logs
This option instructs the utility to lock all tables on all servers and then to delete the binary logs of a master replication server after completing the export. Using this option also invokes the --master-data option.

--disable-keys, -K
For MyISAM tables, this option adds an ALTER TABLE...DISABLE KEYS statement to the export file before each set of INSERT statements, and an ALTER TABLE...ENABLE KEYS statement after each set to optimize later restoration.

--events, -E
This option includes events from the databases. It is available as of version 5.1.8 of MySQL.

--extended-insert, -e

    This option bundles INSERT statements together for each table in the export file to make the export faster. Otherwise, a separate INSERT statement for each row of each table will be placed in the dump file.

--fields-enclosed-by=*characters*

    Use this option with the --tab option to specify the characters that start and end fields in the data text file.

--fields-escaped-by=*character*

    Use this option with the --tab option to specify the character that escapes special characters in the data text file. A backslash is the default.

--fields-optionally-enclosed-by=*characters*

    Use this option with the --tab option to specify the characters that can be used when necessary to start and end fields in the data text file.

--fields-terminated-by=*character*

    Use this option with the --tab option to specify the characters that end fields in the data text file.

--first-slave

    This option locks all tables on all servers. It has been deprecated and replaced with --lock-all-tables.

--flush-logs, -F

    This option flushes all logs. It requires the user to have RELOAD privilege on the server.

--flush-privileges

    This option flushes all privileges. It was added as of version 5.1.12.

--force, -f

    This option instructs the utility to continue processing data despite errors. This is useful in completing dumps for irrelevant errors such as ones related to views that no longer exist.

--help, -?

    This option displays basic help information.

--hex-blob

    This option uses hexadecimal equivalents for BINARY, BIT, BLOB, and VARBINARY columns.

--host=*host*, -h *host*

    This option specifies the name or IP address of the server for connection. The localhost is the default. The user and host combination and related privileges will need to be set on the server.

--ignore-table=*database.table*

    This option instructs the utility not to export the given table of the given database. For more than one table, enter this option multiple times with one database and table combination in each.

--insert-ignore

    This option adds the IGNORE keyword to INSERT statements in the dump file.

--lines-terminated-by=*character*

    Use this option with the --tab option to specify the character that ends records in the data text file.

--lock-tables, -l

This option instructs the utility to get a READ LOCK on all tables of each database before exporting data, but not on all databases at the same time. It locks a database when it's dumping and releases the lock before locking and dumping the next database. This option is typically used with MyISAM tables. For transactional storage engines, use --single-transaction instead.

--lock-all-tables, -x

This option locks all tables on all servers. It replaces --first-slave, which has been deprecated.

--log-error=*logfile*

This option writes errors and warning messages to the file named. The file path may be included. This option is available as of version 5.1.18 of MySQL.

--master-data=*value*

This option is used with replication. It writes the name of the current binary log file and server's position in the log file to the dump file. It requires the RELOAD privilege. It will typically disable --lock-tables and --lock-all-tables.

--no-autocommit

This option adds SET AUTOCOMMIT=0: before each INSERT statement, and a COMMIT; statement after each INSERT statement.

--no-create-db, -n

This option instructs the utility not to add CREATE DATABASE statements to the export file when the --all-databases option or the --databases option is used.

--no-create-info, -t

This option instructs the utility not to add CREATE TABLE statements to the export file.

--no-data, -d

This option exports only database and table schema, not data.

--opt

This option is a combination of several commonly used options: --add-drop-table, --add-locks, --create-options (or --all before version 4.1.2), --disable-keys, --extended-insert, --lock-tables, --quick, and --set-charset. As of version 4.1 of MySQL, the --opt option is enabled by default. Use --skip-opt to disable it for users with limited access.

--order-by-primary

This option sorts rows of tables by their primary key or first index. It slows down the backup process, though.

--password[=*password*], -p[*password*]

This option provides the password to pass to the server. A space is not permitted after -p if the password is given. If the password is not given when using the -p option, the user will be prompted for one.

--port=*port*, -P *port*

This option specifies the port number to use for connecting to the server. A space is expected before the port number when using the -P form of the option.

--protocol=*protocol*

This option is used to specify the type of protocol to use for connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--quick, -q

This option instructs the utility not to buffer data into a complete results set before exporting. Instead, it exports data one row at a time directly to the export file.

--quote-names, -Q

This option places the names of databases, tables, and columns within backticks (`` ` ``). This is the default option. If the server is running in ANSI_QUOTES SQL mode, double quotes will be used instead. This option is enabled by default. Use --skip-quote-names to disable it.

--replace

This option puts REPLACE statements into the dump file instead of INSERT statements. It was added as of version 5.1.3 of MySQL.

--result-file=*filename*, -r *filename*, > filename

This option provides the path and the name of the file to which data should be exported. Use the --result-file option on Windows systems to prevent newline characters (\n) from being converted to carriage return and newline characters (\r\n).

--routines, -R

This option dumps stored procedures and functions. It was added as of version 5.1.2 of MySQL. It requires the SELECT privilege in the proc table of the mysql database. The statements written to the dump file related to these routines do not include timestamps, so the current time will be used when restoring instead.

--set-charset

This option adds the SET NAMES statement to the dump file. It's enabled by default. Use --skip-set-charset to disable it.

--single-transaction

This option executes a BEGIN statement before exporting to help achieve data consistency with the backup. It's effective only on transactional storage engines. It should not be used with MySQL Cluster.

--skip-comments

This option instructs the utility not to export any comments from a table's schema to the export file. It disables the --comments option.

--skip-opt

This option disables the --opt option.

--skip-quote-names

This option disables the --quote-names option.

--ssl

This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled. If this option is enabled on the utility by default, use --skip-ssl to disable it.

--ssl-ca=*pem_file*

This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

--ssl-capath=*path*

This option specifies the path to the trusted certificates file (i.e., the *pem* file).

--ssl-cert=*filename*

This option specifies the name of the SSL certificate file to use for SSL connections.

`--ssl-cipher=ciphers`
> This option gives a list of ciphers that may be used for SSL encryption.

`--ssl-key=filename`
> This option specifies the SSL key file to use for secure connections.

`--ssl-verify-server-cert`
> This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

`--socket=filename, -S filename`
> This option provides the name of the server's socket file on a Unix-type system or the named pipe on Windows systems.

`--tab=path, -T path`
> This option creates two separate export files: one for the table schema (e.g., *table.sql*) and another for the data (e.g., *table.txt*). The data text file will contain data in a tab-separated format. This option requires FILE privilege, and the MySQL server must have write permission for the directory it is to write the exported file.

`--tables`
> This option specifies tables to dump. All names after the --tables option are treated as table names and not as database names.

`--triggers`
> This option includes triggers in dump files. It is the default. Use --skip-triggers to disable it.

`--tz-utc`
> This option adds SET TIME_ZONE='+00:00'; to the dump file so that the dump files may be restored on a server in a different time zone and not cause inconsistencies with TIMESTAMP columns. This option is available as of version 5.1.2 of MySQL and is enabled by default. Use --skip-tz-utc to disable it.

`--user=user, -u user`
> This option specifies the username for connecting to the server. A space is expected after the -u option. If the -u version of this option is used and the username is not given, the current system user is assumed.

`--verbose, -v`
> This option displays more information.

`--version, -V`
> This option displays the version of the utility and exits.

`--where='condition', -w 'condition'`
> This option sets a WHERE condition for selecting rows from tables to be exported. For instance, suppose that we want to back up the clients table with only the clients who are located in New Orleans. We could run the utility like so:

```
mysqldump -u russell -p /
    --where="client_city='New Orleans'" workrequests clients > /tmp/
    workreq_clients_neworleans.sql
```

`--xml, -X`
> This option exports databases in XML format.

⌐ mysqldump --debug options

Table 16-1 lists the debugging, tracing, and profiling flags used with the --debug option for several MySQL-related utilities. The format is generally --debug='*flag:flag:flag*'. When a particular option needs more details, follow the flag with a comma and the details or extra settings in a comma-separated list: --debug='*flag:flag,setting,setting:flag*'. An alternative to the --debug='*flag:flag:flag*' syntax is --#*flag:flag:flag*. This syntax lacks the equals sign or quotes; the space afterward marks the end of the flags and settings.

Table 16-1. Debugging options

Flag	Description
d	Logs the DBUG macros. To log only certain macros, give the d flag followed by the specific macro keywords.
D	Used to specify a delay after each line in the debugging log. After the flag and a comma, give the number of tenths of a second to delay (e.g., D,10 for a 1 second delay).
f	Limits debugging, tracing, and profiling to particular functions. The f flag with no functions listed results in all functions being filtered out of the log.
F	Names the source filename for debugging and tracing output.
i	Specifies the process identifier (PID) or thread identifier for each line of debugging and tracing output that is logged.
g	Enables profiling. A file named *dbugmon.out* may be used to provide details for profiling. A list of functions to profile may be given after this flag. If none are specified, all functions will be included.
L	Includes the source file's line number in each line of the debugging and tracing log.
n	Logs the nesting depth of each function for debugging and tracing.
N	Includes a line number in each line of the log.
o	Redirects debugging information to a given file, rather than stderr. The filename is given after the flag, separated from it by a comma (e.g., o,/tmp/mysql_debug.log).
O	This is the same as the o flag, but the log file is flushed between each write, and possibly opened and closed each time.
p	Limits debugging to given processes. Each process has to be specified with the DBUG_PROCESS macro.
P	Writes the current process name for each line to the debugging and tracing logs.
r	Resets the previous state's function nesting level.
S	Used with safemalloc to locate memory leaks. Will run until nonzero is returned.
t	Enables call and exit trace logging. A numeric maximum trace level may be given after the flag, separated from it by a comma.

⌐ mysqldumpslow

```
mysqldumpslow [options] [filename]
```

⌐ Use this utility to display a summary of the slow query log. The name of the log file may be given in the second argument. Otherwise, the utility will look to the server's configuration file (i.e., *my.cnf* or *my.ini*, depending on your system) for this information. The following options can narrow the summary or change what is displayed.

mysqldumpslow options

-a
> This option instructs the utility not to combine queries with similar SQL statements.

--debug, -d
> This option enables debugging mode.

-g *expression*
> This option extracts information on queries that meet the given expression.

-h *host*
> This option specifies the host's name for which the utility is to scan. By default, log files are named with the server's hostname as the filename's prefix.

--help
> This option displays help information on the utility.

-i *host*
> This option specifies the hostname of the server.

-l
> With this option, the lock time is added to the execution time for the utility's summary.

-n *number*
> This option sets the minimum number of occurrences for reporting.

-r
> This option reverses the order of sorts for reporting.

-s *type*
> This option specifies the type of queries on which to report. The choices are al for average lock time, ar for average rows, at for average execution time, l for lock time, r for rows, and t for execution time.

-t
> This option sets the number of queries on which to display.

--verbose, -v
> This option displays more information.

mysqlhotcopy

mysqlhotcopy *database* [*path*]

Use this utility to make backup copies of databases while the server is active. It works only on MyISAM and ISAM tables. It makes a simple copy of each database directory and each table file. This results in a separate directory for each database and usually three files for each table: one for the schema, another for the data, and a third for the index. It places a read lock on all of the tables in the database while copying them. Here is an example of how you can copy a database with mysqlhotcopy:

```
mysqlhotcopy -u russell -p password workrequests /tmp/backup
```

Note that unlike other MySQL utilities, there is a space between the -p and the password. Next, specify the database (workrequests). Finally, give the path to write the backup directories. To restore databases or tables that were copied by mysqlhotcopy, just copy the table files to be restored to their original data directories.

mysqlhotcopy options

--addtodest

    This option instructs the utility not to abort the session or to rename the backup directory, but to add new files to the directory.

--allowold

    This option renames any existing backup directory with an _old suffix so that the copying may be completed. If the new copy is successful, the old directory is deleted. If it's unsuccessful, the old directory is restored.

--checkpoint=*database.table*

    This option saves logging information to the named database and table.

--chroot=*path*

    This option is used to specify the base directory of the chroot in which the *mysqld* daemon is located, which should have the same directory of the --chroot option.

--debug

    This option is used to enable debugging information.

--dryrun, -n

    This option has the utility test the backup process without actually making a copy.

--flushlog

    This option flushes logs after all tables are locked.

--help, -?

    This option displays basic help information.

--host=*host*, -h *host*

    This option specifies the name or IP address of the server for connection.

--keepold

    This option instructs the utility when using the --allowold option not to delete the old directory if the copying is successful.

--method=*method*

    This option sets the method used by the utility for copying files. The choices are cp or scp.

--noindices

    This option copies only the headers of index files. Indexes may be rebuilt when restoring copies.

--password=password, -ppassword

    This option provides the password to pass to the server. A space is permitted after the -p option, before the password.

--port=*port*, -P *port*

    This option specifies the port number to use for connecting to the server.

--quiet, -q

    This option suppresses all messages except for error messages.

--record_log_pos=*database.table*

    This option is used to specify the database and table to record the log position and status of the master and slave servers when using replication.

--regexp=*expression*
>   This option provides a regular expression for determining which databases to copy based on the name.

--resetmaster
>   This option executes a RESET MASTER statement after tables are locked.

--resetslave
>   This option executes a RESET SLAVE statement after tables are locked.

--socket=*filename*, -S *filename*
>   This option provides the name of the server's socket file.

--sufix=*string*
>   This option specifies the suffix for the copies of databases. The default is _copy.

--tmpdir=*path*
>   This option specifies the temporary directory to use. The default is */tmp*.

--user=*user*, -u *suser*
>   This option specifies the username for connecting to the server.

mysqlimport

mysqlimport [*options*] *database filename*[...]

Use this to import data and table structures from a text file given as the third argument into a database named in the second argument. This utility interacts with the server and uses the LOAD DATA INFILE statement. The root name of the text file being imported must be the same as the table name. Additional text files may be given in a space-separated list. Options may be given on the command line as the first argument of the utility, or they may be provided in the server's configuration file (e.g., *my.cnf*) under the heading [client] or [mysqlimport]. When included in the configuration file, options appear without the leading double dashes. Here is an alphabetical list of options you can give for the first argument, along with an explanation of each.

mysqlimport options

--character-sets-dir=*path*
>   This option specifies the directory containing character sets.

--columns=*columns*, -c *columns*
>   This option identifies the order of fields in the text file as they relate to the columns in the table. Columns are given in a comma-separated list.

--compress, -C
>   This option compresses data passed between the utility and the server, if compression is supported.

--debug[=*options*], -# [*options*]
>   This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list of the mysqldump utility options earlier in this chapter for an explanation of these flags and others that may be used.

--debug-check
>   This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info

This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--default-character-set=*set*

This option specifies the default character set.

--defaults-extra-file=*filename*

This option takes additional options from the text file named.

--defaults-file=*filename*

This option instructs the utility to accept options only from the text file named.

--delete, -d

This option deletes all data from each target table before importing data from the text file.

--fields-enclosed-by=*characters*

This option identifies the characters that indicate the start and end of fields in the text file being imported.

--fields-escaped-by=*character*

This option identifies the character that will escape special characters in the text file being imported. A backslash is the default.

--fields-optionally-enclosed-by=*characters*

This option identifies the characters that indicate the start and end of fields in the text file being imported.

--fields-terminated-by=*character*

This option identifies the character that indicates the end of fields in the text file being imported.

--force, -f

This option instructs the utility to continue importing data despite errors encountered.

--help, -?

This option displays basic help information.

--host=*host*, -h *host*

This option specifies the name or IP address of the server for connection.

--ignore, -i

This option instructs the utility to ignore error messages regarding rows containing duplicate keys and thereby not to replace such rows with imported data.

--ignore-lines=*number*

This option instructs the utility to ignore the first number of lines specified. It's useful in skipping headings in the text file being imported.

--lines-terminated-by=*character*

This option identifies the character that indicates the end of records in the text file being imported.

--local, -L

This option tells the utility that the text file to import is located locally on the client and not on the server, which is the default assumption.

--lock-tables, -l

This option locks all tables before importing data.

--low-priority
> This option has the utility use the LOW PRIORITY keyword when importing data.

--no-defaults
> This option tells the utility not to accept options from a configuration file.

--password[=*password*], -p[*password*]
> This option provides the password to pass to the server. A space is not permitted after the -p option if the password is given. If the password is not given, the user will be prompted for one.

--port=*port*, -P *port*
> This option specifies the port number to use for connecting to the server.

--print-defaults
> This option displays related options found in the server's configuration files.

--protocol=*protocol*
> This option is used to specify the protocol to use when connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--replace, -r
> This option replaces rows that contain duplicate keys with the imported data.

--silent, -s
> This option suppress all messages except for error messages.

--socket=*filename*, -S *filename*
> This option provides the name of the server's socket file.

--ssl
> This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled. If this option is enabled on the utility by default, use --skip-ssl to disable it.

--ssl-ca=*pem_file*
> This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

--ssl-capath=*path*
> This option specifies the path to the trusted certificates file (i.e., the *pem* file).

--ssl-cert=*filename*
> This option specifies the name of the SSL certificate file to use for SSL connections.

--ssl-cipher=*ciphers*
> This option gives a list of ciphers that may be used for SSL encryption.

--ssl-key=*filename*
> This option specifies the SSL key file to use for secure connections.

--ssl-verify-server-cert
> This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

--user=*user*, -u *user*
> This option specifies the username for connecting to the server.

--verbose, -v
> This option displays more information.

--version, -V
> This option displays the version of the utility.

Command-Line
Utilities

mysqlshow

```
mysqlshow [options] [database [table [column]]]
```

Use this utility to obtain a list of databases, tables, or descriptions of tables. It interacts with the server and uses the SHOW DATABASES, SHOW TABLES, and SHOW TABLE statements. If no database name is given for the second argument, all database names will be listed. If a database name is given along with a table name, the table named will be described. To limit information to specific columns, list the columns desired in the fourth argument:

```
mysqlshow --user=russell -ppassword workrequests work_req
```

The results of this command will be the same as entering the following SQL statement from the *mysql* client:

```
SHOW TABLE workrequests.work_req;
```

Here is an alphabetical list of options that you can give as part of the first argument of the utility, along with a brief explanation of each.

mysqlshow options

--character-sets-dir=*path*
: This option specifies the directory containing character sets.

--compress, -C
: This option compresses data passed between the utility and the server, if compression is supported.

--count
: This option returns the number of rows for the given table.

--debug[=*options*], -# [*options*]
: This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list the mysqldump utility options earlier in this chapter for an explanation of these flags and others that may be used.

--debug-check
: This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info
: This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--default-character-set=*set*
: This option specifies the default character set.

--help, -?
: This option displays basic help information.

--host=*host*, -h *host*
: This option specifies the name or IP address of the server for connection.

--keys, -k
: This option displays table indexes.

--password[=*password*], -p[*password*]
>   This option provides the password to pass to the server. A space is not permitted after the -p option if the password is given. If the password is not given, the user will be prompted for one.

--port=*port*, -P *port*
>   This option specifies the port number to use for connecting to the server.

--protocol=*protocol*
>   This option specifies the protocol to use when connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--show-table-type, -t
>   This option adds a column to the results to indicate the type of table: a base table or a view.

--socket=*filename*, -S *filename*
>   This option provides the name of the server's socket file.

--ssl
>   This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled. If this option is enabled on the utility by default, use --skip-ssl to disable it.

--ssl-ca=*pem_file*
>   This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

--ssl-capath=*path*
>   This option specifies the path to the trusted certificates file (i.e., the *pem* file).

--ssl-cert=*filename*
>   This option specifies the name of the SSL certificate file to use for SSL connections.

--ssl-cipher=*ciphers*
>   This option gives a list of ciphers that may be used for SSL encryption.

--ssl-key=*filename*
>   This option specifies the SSL key file to use for secure connections.

--ssl-verify-server-cert
>   This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

--status, -i
>   This option displays additional information regarding tables.

--user=*user*, -u *user*
>   This option specifies the username for connecting to the server.

--verbose, -v
>   This option displays more information.

--version, -V
>   This option displays the version of the utility.

Command-Line
Utilities

mysqlslap

mysqlslap [*options*] *database*

This utility is used to emulate a load of multiple clients on the server to check the timing of the system. It's available as of version 5.1.4 of MySQL.

mysqlslap options

--auto-generate-sql, -a
> If you do not want to use or have a file containing SQL statements for testing the server, nor do you want to manually supply SQL statements from the command line, you can use this option to instruct the utility to automatically generate SQL to emulate a client load.

--*compress*, -C
> This option compresses data passed between the utility and the server, if compression is supported.

--concurrency=*number*, -c *number*
> Use this option to specify the number of clients to simulate.

--create=*value*
> This option is used to specify a file or string to use for creating a table for use in testing.

--create-schema=*value*
> This option is used to specify a file or string containing a table schema to use for creating a table for use in testing.

--csv[=*filename*]
> This option returns data in a comma-separated value format. It will export the data to the standard output, unless a filename is given. Then it will save the information to that file.

--debug[=*options*], -# [*options*]
> This option logs debugging information. The set of options used by default is 'd:t:o,logname'. See Table 16-1 at the end of the list of the mysqldump utility options earlier in this chapter for an explanation of these flags and others that may be used.

--debug-check
> This option writes debugging information to the log when the utility ends. It's available as of version 5.1.21 of MySQL.

--debug-info, -T
> This option writes debugging information and CPU and memory usage information to the log after the utility ends.

--delimiter=*string*, -F *string*
> Use this option to specify the delimiter used in the SQL file given.

--engine=*engine*, -e *engine*
> Use this option to specify the storage engine to use for the test table.

--host=*host*, -h *host*
> This option specifies the host on which to connect to the server.

--help, -?
> This option displays help information about the utility.

--iterations=*number*, -i *number*

This option is used to specify the number of times to run the client load emulation tests.

--number-char-cols=*number*, -x *number*

When specifying --auto-generate-sql, use this option to specify the number of VARCHAR columns to use.

--number-int-cols=*number*, -y *number*

When specifying --auto-generate-sql, use this option to specify the number of INT columns to use.

--number-of-queries=*number*

This option is used to specify the number of queries for each client.

--only-print

This option instructs the utility not to run the tests on the server, but to display only what would have been done based on the options given.

--password=*password*, -p *password*

This option provides the password of the user logging into the server.

--port=*port*

This option specifies the port on which to connect to the server. The default is 3306.

--preserve-schema

This option preserves the schema used when the utility was run.

--protocol=*protocol*

This specifies the protocol to use when connecting to the server. The choices are TCP, SOCKET, PIPE, and MEMORY.

--query=*value*, -q *value*

This option is used to give the string or to specify the file to use that contains the SELECT statement for querying the server for testing.

--silent, -s

This option displays no messages.

--socket=*filename*, -S *filename*

This option provides the name of the server's socket file for Unix-type systems or the named pipe for Windows systems.

--ssl

This option specifies that secure SSL connections should be used. It requires the server to have SSL enabled. If this option is enabled on the utility by default, use --skip-ssl to disable it.

--ssl-ca=*pem_file*

This option specifies the name of the file (i.e., the *pem* file) containing a list of trusted SSL CAs.

--ssl-capath=*path*

This option specifies the path to the trusted certificates file (i.e., the *pem* file).

--ssl-cert=*filename*

This option specifies the name of the SSL certificate file to use for SSL connections.

--ssl-cipher=*ciphers*

This option gives a list of ciphers that may be used for SSL encryption

--ssl-key=*filename*
> This option specifies the SSL key file to use for secure connections.

--ssl-verify-server-cert
> This option verifies the client's certificate against the server's certificate for the client at startup. It is available as of version 5.1.11 of MySQL.

--use-threads
> On Unix-type systems, the mysqlap utility uses fork( ). This option will instruct the server to use pthread( ) instead. On Windows systems threads are used by default.

--user=*user*, -u *user*
> This option provides the username for logging into the server.

--verbose
> This option displays more information from the utility.

--version
> This option returns the version of the utility.

perror

perror [*options*] *code*

This utility displays descriptions of system error codes that MySQL receives. Multiple error codes may be given in a space-separated list as the second argument. The only options available are for help (--help), the version number (--version), and verbosity (--verbose). As of recent versions of MySQL, the --ndb option has been added to get MySQL Cluster error messages.

replace

replace *options filename*

This program searches and replaces text in a simple text file, such as a dump file. Give the text to be replaced followed by the replacement text. Multiple pairs of such text can be given in a space-separated list. A double-dash (--) is used to mark the end of text replacement pairs, after which you list the names of files on which to perform the replacement in a space-separated list.

The only options available for this utility are for help (-? or -I), silent mode (-s), the version number (-v), and verbosity (-V). You can also specify -# followed by a space and flags for debugging. See the explanation of --debug under mysqldump earlier in this chapter for options that may be given with this flag.

The strings for which the utility is to search may include a few regular expression parameters: \^ to indicate the start of a line; \$ for the end of a line; and \b for a space.

resolveip

resolveip [*options*] *host* ...

This is a simple network program that translates a hostname to its related IP address. If an IP address is given, it returns all domains associated with the address. It has nothing to do with MySQL per se, but it is included in the normal distribution package.

resolve_stack_dump

```
resolve_stack_dump options symbols_filename [numeric_dump_file]
```

This utility resolves addresses and other numeric data into a stack to symbol names. The symbols file given should be the output of executing the following at the command line:

```
nm --numeric-sort mysqld
```

The numeric file named should be the numeric stack from mysqld.

Instead of following the basic syntax, you can specify the symbols file with the --symbols-file option. You can also specify the numeric dump file with the --numeric-dump-file option. For both options, the option is followed by an equals sign and the filename.