

22

MySQL and Programs and Utilities

MySQL comes with a wealth of programs and utilities to make interacting with the database server easier. Some of these programs are used by the end user to read and write from the database, while others are meant for the database administrator to maintain and repair the database as a whole.

In this chapter, we'll provide detailed documentation for all the programs and utilities that are available. First, here's an inventory of them all with a brief description of each.

mysql

The MySQL SQL command shell. This is presented first because it is used so frequently..

mysqld

The MySQL server. In this section we cover all of the command line options for the mysql server. In addition, we cover `safe_mysqld`, the `mysqld` wrapper script, and `mysql.server`, a script for starting and stopping `mysqld` on System-V style Unix systems.

myisamchk/isamchk

Maintenance utilities for MyISAM/ISAM files. Among other things, these are used check tables for errors and repair them.

myisampack/pack_isam

Compressed, read-only table generators.

mysqlaccess

A script to check the access privileges for a host, user, and database combination.

mysqladmin

A utility for performing administrative operations.

mysqlbug

A utility for submitting MySQL bugs.

mysqlcheck

Maintenance utility for MyISAM files. This is similar to myisamchk but can be used while the server is running.

mysqldump

A utility to dump the contents a one or more databases as a set of SQL commands. Can be used to back up databases or create a copy of a database.

mysqlhotcopy

A perl script to make a binary copy of a database.

mysqlimport

A utility to load data into a database. This is essentially a command line version of the LOAD DATA INFILE SQL command.

mysqlshow

A utility to show information about databases, tables and columns.

Configuration Files

Many of the programs and utilities allow you to specify options in a configuration file as well as on the command line. The utilities that support this are:

- mysql
- mysqladmin
- mysqld
- safe_mysqld
- mysqldump
- mysqlimport,
- myisamchk
- myisampack.

Each of the programs that support configuration files support the following options:

--no-defaults

Don't read any option files.

--print-defaults

Print the program name and all options that it will get.

--defaults-file=full-path-to-default-file

Only use the given configuration file.

--defaults-extra-file=full-path-to-default-file

Read this configuration file after the global configuration file but before the user configuration file.

See Chapter 5 for an in-depth discussion of how to set up and use configuration files.

mysql, the MySQL Command Shell

mysql

mysql [options] [database]

Since mysql is the utility you'll likely be using the most, we'll cover that first.

mysql is a simple SQL command shell. It is a general purpose client which will allow you to execute arbitrary SQL statements against your database. This is the utility you use when you want to run ad-hoc queries against your database, create tables or indexes, etc.

It supports interactive and non-interactive (i.e. as a filter) use. In interactive mode, GNU readline capabilities are provided and data is displayed in an ASCII-table format. In non-interactive mode, the data is presented in a tab-separated format.

Using mysql interactively is simple. Simply type

```
| % mysql <dbname>
```

or

```
| % mysql --user=<username> --password=<password> <dbname>
```

where <dbname> is the name of the database you wish to connect to, and <username>/<password> are for the user you wish to connect as. If you don't specify --user, the \$USER environment variable will be used. If you don't supply a password, mysql will prompt for it.

Once started, mysql presents a prompt. Here you can type SQL commands. Commands can span multiple lines and must be terminated with ';' or '\g'. So, for example, typing

```
| mysql> SELECT *
| mysql> FROM FOOBAR ;
```

would execute the SELECT statement. That's all there is too it.

mysql has command line editing (like a bash shell), because it uses the same GNU readline library that bash uses. For example, you can complete a word by using the tab key, press Ctrl-a to jump to the start of the current line or Ctrl-e to jump to the end, press Ctrl-r to perform a reverse search, and use the up arrow to retrieve the previous command.

mysql also provides history. By hitting the up/down arrow, you can scroll through your history of SQL commands. This works similar to bash history.

To run, mysql non-interactively, you redirect your SQL commands into mysql. You can also redirect the output to a file. For example,

```
| % mysql --user=<user> --password=<password> <dbname> < script.sql > script.out
```

In this way, the mysql command can be combined in shell pipelines just like any other UNIX filter. This is tremendously useful for constructing scripts to access your database.

mysql has a number of built-in commands. Each command has a long format and a short format. These are listed below. The short format is listed in parentheses.

When using full word commands (go, print, etc.) the command must be entered on a line by itself. Escape character commands (\g, \p, etc.) can be used at the end of any line. In addition, a semicolon can be used to end an SQL statement just like \g.

help (\h)
Display the help for mysql.

? (\h)
Synonym for `help`.

clear (\c)
Clear command. Clear the query buffer.

connect (\r)
Reconnect to the server. Optional arguments are db and host.

edit (\e)
Edit command buffer with the text editor specified in the environment variable \$EDITOR.

ego (\G)
Send command to mysql server, display result vertically.

exit (\q)
Exit mysql. Same as quit.

go (\g)
Send command to mysql server.

nopager (\n)
Disable pager, print to stdout.

notee (\t)
Disable tee. Don't write into outfile.

pager (\P)
Print the query results via the command specified in the PAGER environment variable or in the --pager command line option..

print (\p)
Print current command.

quit (\q)
Quit mysql.

rehash (\#)
Rebuild completion hash.

source (\.)
Execute a SQL script file. Takes a file name as an argument.

status (\s)
 Get status information from the server.

tee (\T)
 Append all output into given outfile.

use (\u)
 Use another database. Takes database name as argument.

mysql supports the following command line options:

-, --help
 Display the help for mysql and exit.

-A, --no-auto-rehash
 Disable automatic rehashing. Normally, when mysql starts up, it reads the table and column names for a database to provide tab completion. When this is disabled, mysql will start up faster, but you will have to use the 'rehash' mysql command to get table and field completion.

-B, --batch
 Print results with a tab as separator, each row on a new line.

--character-sets-dir=...
 Directory where character sets are located.

-C, --compress
 Use compression in server/client protocol.

-, --debug[=...]
 Enable the debug log. Default is 'd:t:o,/tmp/mysql.trace'.

-D, --database=...
 Database to use. This is mainly useful in the my.cnf file to set your default database.

--default-character-set=...
 Set the default character set.

-e, --execute=...
 Execute command and quit. This option allows you to supply an SQL command on the mysql command line. It will execute the command and return the results as if it were in --batch mode.

-E, --vertical
 Print the output of a query (rows) vertically. Without this option you can also force this output by ending your statements with \G.

-f, --force
 Continue even if we get a SQL error.

-g, --no-named-commands
 Long format built-in commands are disabled. Use * form only, or use named commands only in the beginning of a line ending with a semicolon (;).

-G, --enable-named-commands
 Long format built-in commands are enabled, as well as shortened * commands.

-i, --ignore-space
 Ignore space after function names.

- h, --host=...
Connect to the MySQL server on the specified host.
- H, --html
Produce HTML output.
- L, --skip-line-numbers
Don't write line number for errors. This is useful when one wants to compare result files that includes error messages
- no-pager
Disable pager and print to stdout.
- no-tee
Disable outfile.
- n, --unbuffered
Flush buffer after each query.
- N, --skip-column-names
Don't write column names in results.
- O, --set-variable var=option
Set a variable. The mysql variables are described below. Use --help to list variables.
- o, --one-database
Only update the database specified on the command line. This is useful for playing back a set of updates from the update log and/or binary log. All updates to databases other than the database on the command line will be ignored.
- pager[=...]
Set the pager to use for displaying output in interactive mode. If this is unspecified, it defaults to the pager defined by your PAGER environment variable. Valid pagers are less, more, cat [> filename], etc. This option does not work in batch mode. --pager works only on UNIX.
- p[password], --password[=...]
Password to use when connecting to server. If a password is not given on the command line, you will be prompted for it. Note that if you use the short form -p you can't have a space between the option and the password.
- P --port=...
TCP/IP port number of the server you wish to connect to.
- q, --quick
Don't cache results, instead print them row-by-row. This may slow down the server if the output is suspended.
- r, --raw
Write column values without escape conversion. Typically used with --batch
- s, --silent
Be more silent.
- S --socket=...
Socket file of the server you wish to connect to.
- t --table
Output in ASCII table format. This is the default in interactive mode.

- T, --debug-info
Print some debug information at exit.
- tee=outfile
Append everything into outfile. Does not work in batch mode.
- u, --user=#
User for login if not current user.
- U, --safe-updates[=#], --i-am-a-dummy[=#]
Only allow UPDATE and DELETE statements that are constrained in the WHERE clause by a indexed column. This is useful for preventing accidental deletion of all rows from a table, for example. In addition, the select_limit and max_join_size variables are consulted. SELECT statements are limited to select_limit rows, and all queries with joins that need to examine more than max_join_size rows are aborted. You can reset this option if you have it in your my.cnf file by using --safe-updates=0.
- v, --verbose
Enables more verbose output (-v -v -v enables the table output format).
- V, --version
Output version information and exit.
- w, --wait
Wait and retry if connection is down instead of aborting.

mysql also provides a small set of variables that can be set with the -O or --set-variable command:

- connect_timeout
Number of seconds before connection is timed out. 0 indicates not timeout. The default is 0.
- max_allowed_packet
Maximum packet length to send/receive from server.
- net_buffer_length
Size of the buffer for TCP/IP and socket communication.
- select_limit
Row limit for SELECT with --safe-updates/--i-am-a-dummy enabled.
- max_join_size
Maximum number of rows to be examined to satisfy a join with --safe-updates/--i-am-a-dummy enabled.

mysqld, the MySQL Server

mysqld, mysqld-max

mysqld [*options*]

mysqld-max [*options*]

mysqld is the MySQL server. The recommended way to invoke mysqld is via the `safe_mysqld` script (described below). `mysqld-max` is a version of the MySQL server with support for BDB and InnoDB tables compiled in.

mysqld supports the following options:

`--ansi`

Use ANSI SQL syntax instead of MySQL syntax. This has the following effects:

- `||` is acts the string concatenation operator instead of `OR`.
- Any number of spaces are allowed between a function name and the ```. This forces all function names to be treated as reserved words.
- ``"` acts as an identifier quote character (like the MySQL ````` quote character) and not a string quote character.
- `REAL` is a synonym for `FLOAT` instead of a synonym of `DOUBLE`.
- The default transaction isolation level is `SERIALIZABLE`

`--ansi` is equivalent to `--sql=REAL_AS_FLOAT,PIPES_AS_CONCAT,ANSI_QUOTES,IGNORE_SPACE,SERIALIZE,ONLY_FULL_GROUP_BY`

`-b, --basedir=path`

Path to installation directory. All paths are usually resolved relative to this.

`--big-tables`

Allow big result sets by saving all temporary sets on the file system. It solves most 'table full' errors, but also slows down the queries where in-memory tables would suffice. Since Version 3.23.2, MySQL is able to handle this automatically by using memory for small temporary tables and switching to disk tables where necessary.

`--bind-address=IP`

IP address to bind to.

`--character-sets-dir=path`

Directory where character sets are.

`--chroot=path`

Chroot mysqld daemon during startup. Recommended security measure. It will somewhat limit `LOAD DATA INFILE` and `SELECT ... INTO OUTFILE` though.

`--core-file`

Write a core file if mysqld dies. For some systems you must also specify `--core-file-size` to `safe_mysqld`. See the following section about `safe_mysqld`.

`-h, --datadir=path`

Path to the database root.

`--default-character-set=charset`

Set the default character set.

- `--default-table-type=type`
Set the default table type for creating tables.
- `--debug[...]=`
If MySQL is configured with `--with-debug`, you can use this option to get a trace file of what mysqld is doing.
- `--delay-key-write-for-all-tables`
Don't flush key buffers between writes for any MyISAM table.
- `--enable-locking`
Enable system locking. Note that if you use this option on a system which a not fully working `lockd()` (as on Linux) mysqld will deadlock.
- `-T, --exit-info`
This is a bit mask of different flags one can use for debugging the mysqld server; One should not use this option if one doesn't know exactly what it does!
- `--flush`
Flush all changes to disk after each SQL command. Normally MySQL only does a write of all changes to disk after each SQL command and lets the operating system handle the syncing to disk.
- `-, --help`
Display short help and exit.
- `--init-file=file`
Read SQL commands from this file at startup.
- `-L, --language=...`
Language to use for client error messages. May be given as a full path.
- `-l, --log[=file]`
Enable the query log. See Chapter 5 for more details.
- `--log-isam[=file]`
Enable the ISAM/MyISAM log (only used when debugging ISAM/MyISAM).
- `--log-slow-queries[=file]`
Enable the slow query log. See Chapter 5 for more details.
- `--log-update[=file]`
Enable the update log. See Chapter 5 for more details.
- `--log-long-format`
Log some extra information to update log. If you are using `--log-slow-queries` then queries that are not using indexes are logged to the slow query log.
- `--low-priority-updates`
All table-modifying operations (INSERT/DELETE/UPDATE) will have lower priority than selects. It can also be done via `{INSERT | REPLACE | UPDATE | DELETE} LOW_PRIORITY ...` to lower the priority of only one query, or by `SET OPTION SQL_LOW_PRIORITY_UPDATES=1` to change the priority in one thread.
- `--memlock`
Lock the mysqld process in memory. This works only if your system supports the `mlockall()` system call (like Solaris). This may help if you have a problem where the operating system is causing the mysqld processs to swap.

`--myisam-recover [=option[,option...]]`

Set the MyISAM recovery options. option is any combination of DEFAULT, BACKUP, FORCE or QUICK. You can also set option explicitly to "" if you want to disable this option. If this option is used, mysqld examine each MyISAM file on open. If the table is marked as crashed or if the table wasn't closed properly, mysqld will run check on the table. If the table was corrupted, mysqld will attempt to repair it. The following options affects how the repair works.

- **DEFAULT** The same as not giving any option to `--myisam-recover`.
- **BACKUP** If the data table was changed during recover, save a backup of the ``table_name.MYD'` data file as ``table_name-datetime.BAK'`.
- **FORCE** Run recover even if more than one row will be lost from the `.MYD` file.
- **QUICK** Don't check the rows in the table if there aren't any delete blocks.

Before a table is automatically repaired, MySQL will add a note about this in the error log. If you want to be able to recover from most things without user intervention, you should use the options `BACKUP, FORCE`. This will force a repair of a table even if some rows would be deleted, but it will keep the old data file as a backup so that you can later examine what happened.

`--pid-file=path`

Path to pid file used by `safe_mysqld`.

`-P, --port=...`

Port number to listen for TCP/IP connections.

`-o, --old-protocol`

Use the version 3.20 protocol for compatibility with some very old clients.

`--one-thread`

Only use one thread (for debugging under Linux).

`-O, --set-variable var=option`

Give a variable a value. `--help` lists all variables. See Chapter 18 for more information about variables.

`--safe-mode`

Skip some optimize stages. Implies `--skip-delay-key-write`.

`--safe-show-database`

Don't show databases for which the user doesn't have any privileges.

`--safe-user-create`

If this is enabled, a user can't create new users with the `GRANT` command, if the user doesn't have `INSERT` privilege to the `mysql.user` table or any column in this table.

`--skip-concurrent-insert`

Turn off the ability to select and insert at the same time on MyISAM tables. This should only to be used if you think you have found a bug in this feature.

- `--skip-delay-key-write`
Ignore the `delay_key_write` option for all tables.
- `--skip-grant-tables`
This option causes the server not to use the privilege system at all. This gives everyone full access to all databases! You can tell a running server to start using the grant tables again by executing `mysqladmin flush-privileges` or `mysqladmin reload`.
- `--skip-host-cache`
Don't use host name cache for faster IP address resolution. This causes `mysqld` to query DNS server on every connect.
- `--skip-locking`
Disable system locking. When this is disabled, you must shut down the server to use `isamchk` or `myisamchk`.
- `--skip-name-resolve`
Hostnames are not resolved. All Host column values in the grant tables must be IP numbers or `localhost`.
- `--skip-networking`
Don't listen for TCP/IP connections at all. All interaction with `mysqld` must be made via Unix sockets. This option is highly recommended for systems where only local requests are allowed.
- `--skip-new`
Don't use new, possibly wrong ISAM routines. Implies `--skip-delay-key-write`. This will also set default table type to ISAM.
- `--skip-symlink`
Don't delete or rename any files that a symbolically linked file in the data directory points to.
- `--skip-safemalloc`
If MySQL is configured with `--with-debug=full`, all programs will check the memory for overruns for every memory allocation and memory freeing. As this checking is very slow, you can avoid this, when you don't need memory checking, by using this option.
- `--skip-show-database`
Don't allow 'SHOW DATABASE' commands, unless the user has process privilege.
- `--skip-stack-trace`
Don't write stack traces. This option is useful when you are running `mysqld` under a debugger.
- `--skip-thread-priority`
Disable using thread priorities for faster response time.
- `--socket=path`
Socket file to use for local connections instead of default `/tmp/mysql.sock`.
- `--sql-mode=option[,option[,option...]]`
Option can be any combination of: `REAL_AS_FLOAT`, `PIPES_AS_CONCAT`, `ANSI_QUOTES`, `IGNORE_SPACE`, `SERIALIZE`, `ONLY_FULL_GROUP_BY`. It can also be empty (""), if you want to reset this. By specifying all of the above options is same as using `--ansi`.

transaction-isolation=level

Sets the default transaction isolation level. Possible level values are READ-UNCOMMITTED, READ-COMMITTED, REPEATABLE-READ, or SERIALIZABLE.

-t, --tmpdir=path

Path for temporary files. Use this if your default /tmp directory resides on a partition too small to hold temporary tables.

-u, --user=user_name

Run mysqld daemon as user user_name. This option is mandatory when starting mysqld as root.

-V, --version

Output version information and exit.

-W, --warnings

Print out warnings like Aborted connection... to the .err file.

safe_mysqld

safe_mysqld [*options*]

safe_mysqld is a wrapper script for mysqld and is the recommended way to start a MySQL server. safe_mysqld will attempt to determine location of the installation, so it is able to start a server that was installed from a source or binary distribution. safe_mysqld will restart the server when an error occurs and it redirects run-time output to a log file.

When starting up, safe_mysqld first looks relative to the current working directory for the 'bin' and 'data' directories (from a binary distribution), or for 'libexec' and 'var' directories (from a source distribution). So if you start safe_mysqld from the your installation directory, it ought to be able to find all the relevant files.

If the server binaries and databases cannot be found relative to the current working directory, safe_mysqld looks in the standard locations. These locations are depend on the type of distribution you have installed. If you installed your distribution in a standard location, it should be able to find everything.

If the options --mysqld and --mysqld-version are not specified safe_mysqld will start the mysqld-max if it can be found. If mysqld-max is not found, mysqld will be started.

safe_mysqld supports a few options over and above those supported by mysqld. But all options supplied on the command line to safe_mysqld are passed directly to mysqld. So, if you want to specify any safe_mysqld options, they must be supplied in an option file. safe_mysqld will read all options from the [mysqld], [server] and [safe_mysqld] sections of option files. See Chapter 5 for more information on setting up option files for your server.

The options supported by safe_mysqld are:

`--basedir=path`
Same as for `mysqld`.

`--core-file-size=#`
Limit the size of the core file `mysqld` can create. Passed to `ulimit -c`.

`--datadir=path`
Same as for `mysqld`.

`--err-log=path`
Location of the error log.

`--ledir=path`
Path to `mysqld`

`--log=path`
Location of the query log.

`--mysqld=mysqld-version`
The full name of the `mysqld` binary version in the `ledir` directory to start.

`--mysqld-version=version`
Similar to `--mysqld=` but here you only give the suffix for `mysqld`. For example if you use `--mysqld-version=max`, `safe_mysqld` will start the `ledir/mysqld-max` version. If the argument to `--mysqld-version` is empty, `ledir/mysqld` will be used.

`--open-files-limit=#`
Number of files `mysqld` should be able to open. Passed to `ulimit -n`. Note that you need to start `safe_mysqld` as root for this to work properly!

`--pid-file=path`
Path to the pid file used by `safe_mysqld`.

`--port=#`
Same as for `mysqld`.

`--socket=path`
Same as for `mysqld`.

`--timezone=#`
Set the timezone (the TZ) variable to the value of this parameter.

`--user=#`
Same as for `mysqld`.

mysql.server

`mysql.server start`

`mysql.server stop`

`mysql.server` is a script that can, on a System-V like UNIX sytem, be used to automatically start and stop `mysqld` at system boot and shutdown. See chapter 5 for more information on automatically starting and stopping your server.

Other programs and utilities

myisamchk/isamchk

```
myisamchk [options] table_file [table_file...]
```

```
isamchk [options] table_file [table_file...]
```

myisamchk and isamchk are identical except that they operate on different file type. myisamchk is meant to work with MyISAM files -- they have an extension of .MYI. isamchk is meant to work with ISAM files -- those with an extension of .ISM. For the remainder of this discussion, we will refer only to myisamchk, but all of the concepts also apply to isamchk.

This utility is used to check and repair the files, as well as report information about them. You must provide the correct path to the ISAM file you wish to examine. For example,

```
| % myisamchk /usr/local/data/foobar/*.MYI
```

will execute against all MyISAM files in the database 'foobar'.

myisamchk/isamchk should only be used when the MySQL is not running. When the server is running, you can use the mysqlcheck command (see below).

-# or --debug=debug_options

Output debug log. The debug_options string often is 'd:t:o,filename'.

-? or --help

Display a help message and exit.

-O var=option, --set-variable var=option

Set the value of a variable. myisamchk --help will report all variables and values. Two important variables are:

key_buffer_size

key_buffer_size is used when you are checking the table with --extended-check or when the keys are repaired by inserting key row by row in to the table (like when doing normal inserts). Repairing through the key buffer is used in the following cases:

- If you use --safe-recover.
- If you are using a FULLTEXT index.
- If the temporary files needed to sort the keys would be more than twice as big as when creating the key file directly. This is often the case when you have big CHAR, VARCHAR or TEXT keys as the sort needs to store the whole keys during sorting. If you have lots of temporary space and you can force myisamchk to repair by sorting you can use the --sort-recover option.

Reparing through the key buffer takes much less disk space than using sorting, but is also much slower.

`sort_buffer_size`

`sort_buffer_size` is used when the keys are repaired by sorting keys, which is the normal case when you use `--recover`.

`-s` or `--silent`

Silent mode. Write output only when errors occur. You can use `-s` twice (`-ss`) to make `myisamchk` very silent.

`-v` or `--verbose`

Verbose mode. Print more information. This can be used with `-d` and `-e`. Use `-v` multiple times (`-vv`, `-vvv`) for more verbosity!

`-V` or `--version`

Print the `myisamchk` version and exit.

`-w` or `--wait`

Instead of giving an error if the table is locked, wait until the table is unlocked before continuing. Note that if you are running `mysqld` on the table with `--skip-locking`, the table can only be locked by another `myisamchk` command.

Check Options

`-c` or `--check`

Check table for errors. This is the default operation if you are not giving `myisamchk` any options that override this.

`-e` or `--extend-check`

Check the table VERY thoroughly (which is quite slow if you have many indexes). This option should only be used in extreme cases. `myisamchk` or `myisamchk --medium-check` should, in most cases, be able to discover any errors in the table. If you are using `--extended-check` and have much memory, you should increase the value of the `key_buffer_size` variable considerably.

`-F` or `--fast`

Check only tables that haven't been closed properly.

`-C` or `--check-only-changed`

Check only tables that have changed since the last check.

`-f` or `--force`

If `myisamchk` finds any errors in the table, restart `myisamchk` with `-r` (repair) on the table,.

`-i` or `--information`

Print informational statistics about the table that is checked.

`-m` or `--medium-check`

Faster than `extended-check`, but only finds 99.99% of all errors. Should, however, be good enough for most cases.

`-U` or `--update-state`

Store in the MyISAM file when the table was checked and if the table crashed. This should be used to get full benefit of the `--check-only-changed` option. Don't use this

option if the mysqld server is using the table and you are running mysqld with --skip-locking.

-T or --read-only

Don't mark table as checked. This is useful if you use myisamchk to check a table that is in use by some other application that doesn't use locking (like mysqld --skip-locking).

Repair Options

The following options are used if you start myisamchk with -r or -o:

-D # or --data-file-length=#

Max length of data file (when re-creating data file when it's 'full').

-e or --extend-check

Try to recover every possible row from the data file. Normally this will also find a lot of garbage rows. Don't use this option except as a last resort.

-f or --force

Overwrite old temporary files (table_name.TMD) instead of aborting.

-k # or keys-used=#

If you are using ISAM, tells the ISAM table handler to update only the first # indexes. If you are using MyISAM, tells which keys to use, where each binary bit stands for one key (first key is bit 0). This can be used to get faster inserts.

-l or --no-symlinks

Do not follow symbolic links. Normally myisamchk repairs the table a symlink points at.

-r or --recover

Can fix almost anything except unique keys that aren't unique (which is an extremely unlikely error with ISAM/MyISAM tables). If you want to recover a table, this is the option to try first. Only if myisamchk reports that the table can't be recovered by -r, you should then try -o. If you have lots of memory, you can increase the sort_buffer_size variable to make this run faster.

-o or --safe-recover

Uses an old recovery method (reads through all rows in order and updates all index trees based on the found rows); this is a magnitude slower than -r, but can handle a couple of very unlikely cases that -r cannot handle. This recovery method also uses much less disk space than -r. Normally one should always first repair with -r, and only if this fails use -o. If you have lots of memory, you can increase the size of the key_buffer_size variable to make this run faster.

-n or --sort-recover

Force myisamchk to use sorting to resolve the keys even if the temporary files should be very big. This will not have any effect if you have fulltext keys in the table.

--character-sets-dir=...

Directory where character sets are stored.

--set-character-set=name

Change the character set used by the index

.t or --tmpdir=path

Path for storing temporary files. If this is not set, myisamchk will use the environment variable TMPDIR for this.

-q or --quick

Faster repair by not modifying the data file. One can give a second -q to force myisamchk to modify the original datafile in case of duplicate keys

-u or --unpack

Unpack file packed with myisampack.

Other Options

-a or --analyze

Analyze the distribution of keys. This improves join performance by enabling the join optimizer to better choose in which order it should join the tables and which keys it should use.

-d or --description

Prints some information about table.

-A or --set-auto-increment[=value]

Force auto_increment to start at this or higher value. If no value is given, then sets the next auto_increment value to the highest used value for the auto key + 1.

-S or --sort-index

Sort the index tree blocks in high-low order. This will optimize seeks and will make table scanning by key faster.

-R or --sort-records=#

Sorts records according to an index. This makes your data much more localized and may speed up ranged SELECT and ORDER BY operations on this index. To find out a table's index numbers, use SHOW INDEX, which shows a table's indexes in the same order that myisamchk sees them. Indexes are numbered beginning with 1.

myisampack/pack_isam

myisampack [*options*] table_name

isam_pack [*options*] table_name

These utilities generate compresses, read-only MyISAM and ISAM files. myisampack is used to compress MyISAM tables, and pack_isam is used to compress ISAM tables. Table compression reduces datafile size from 40 to 70% while maintaining speedy access. myisampack works with all column types. pack_isam will not work with tables that have BLOB or TEXT columns.

myisampack and pack_isam only modify the specific datafiles. To update the indexes, run myisamchk -rq/isamchk -rq after running myisampack/pack_isam.

Options

-b, --backup

Make a backup of the table as tbl_name.OLD.

`-#`, `--debug=debug_options`

Output debug log. The `debug_options` string often is `'d:t:o,filename'`.

`-f`, `--force`

Force packing of the table even if it becomes bigger or if the temporary file exists. `mysampack` creates a temporary file named ``tbl_name.TMD'` while it compresses the table. If you kill `mysampack`, the ``.TMD'` file may not be deleted. Normally, `mysampack` exits with an error if it finds that ``tbl_name.TMD'` exists. With `--force`, `mysampack` packs the table anyway.

`-?`, `--help`

Display a help message and exit.

`-j big_tbl_name`, `--join=big_tbl_name`

Join all tables named on the command line into a single table `big_tbl_name`. All tables that are to be combined **MUST** be identical (same column names and types, same indexes, etc.).

`-p #`, `--packlength=#`

Specify the record length storage size, in bytes. The value should be 1, 2, or 3. `mysampack` stores all rows with length pointers of 1, 2, or 3 bytes. In most cases, `mysampack` can determine the right pack length value before it begins packing the file. Sometimes it may notice during the packing process that it could have used a shorter length. In this case, `mysampack` will print a note that the next time you pack the same file, you could use a shorter record length.

`-s`, `--silent`

Silent mode. Write output only when errors occur.

`-t`, `--test`

Don't actually pack table, just test packing it.

`-T dir_name`, `--tmp_dir=dir_name`

Use the named directory as the location in which to write the temporary table.

`-v`, `--verbose`

Verbose mode. Write information about progress and packing result.

`-V`, `--version`

Display version information and exit.

`-w`, `--wait`

Wait and retry if table is in use. If the `mysql` server was invoked with the `--skip-locking` option and the table has a chance of being updated during while pack is being performed, it is not a good idea to invoke `mysampack`.

mysqlaccess

`mysqlaccess [host [user [database]]] options`

This script is used to test the result of adding privileges to a database. It functions by creating temporary copies of the `user`, `db`, and `host` tables from the `mysql` database. IF the privileges work out, you can commit them back to the `mysql` database. This is very useful for testing a set of changes before applying them to the system.

Options

-b, --brief

Display results in an abbreviated, single-line format.

--commit

Copies the temporary grant tables into the `mysql` database. As always, run `mysqladmin flush-privileges` after performing the commit so the server reloads the privileges.

--copy

Loads the grant tables into the temporary tables.

-d database, --db=database

Specifies the database name.

--debug=n

Sets the debug level. `n` can be an integer from 0 to 3.

--howto

Displays some examples of how to use `mysqlaccess`.

--old_server

Specifies that the server is older than MySQL 3.21. `mysqlaccess` needs to use different queries in this case.

--plan

Displays a list of enhancements planned for future releases of `mysqlaccess`.

--preview

Display the privilege differences between the actual and temporary grant tables.

-H hostname, --rhost=hostname

The remote server to connect to.

--rollback

Undoes the changes made to the temporary tables.

-P password, -spassword=password

The password of the MySQL superuser, or any user with sufficient privileges to modify the grant tables.

-U username, --superuser=username

The user name of the MySQL superuser.

-t, --table

Display results in tabular format.

mysqlaccess

`mysqladmin` [*options*] *command* [*command-options*] ...

Commands

`create databasename`

Create a new database.

`drop databasename`

Delete a database and all its tables.

extended-status

Gives an extended status message from the server.

flush-hosts

Flush all cached hosts.

flush-logs

Flush all logs.

flush-tables

Flush all tables.

flush-privileges

Reload grant tables (same as reload).

kill id,id,...

Kill mysql threads. Use mysqladmin processlist or the SQL command SHOW PROCESSLIST to see the active server threads.

password

Set a new password. Change old password to new-password.

ping

Check if mysqld is alive.

processlist

Show list of active threads in server.

reload

Reload grant tables (same as flush-privileges).

refresh

Flush all tables and close and open logfiles.

shutdown

Shut the server down.

slave-start

Start slave replication thread.

slave-stop

Stop slave replication thread.

status

Gives a short status message from the server.

variables

Prints variables available. See chapter 18 for more information on server variables.

version

Get version info from server.

Options

-, --help

Display the help for mysql and exit.

-C, --compress

Use compression in server/client protocol.

- #, --debug[=...]
Enable the debug log. Default is 'd:t:o,/tmp/mysql.trace'.
- f, --force
If multiple commands are specified on the command line, mysqlaccess will normally halt when an error occurs. When --force is enabled, mysqlaccess will continue to the next command on error.

Also, --force will force a “drop database” operation without confirmation.
- h, --host=...
Connect to the MySQL server on the specified host.
- p[password], --password[=...]
Password to use when connecting to server. If a password is not given on the command line, you will be prompted for it. Note that if you use the short form -p you can't have a space between the option and the password.
- P --port=...
TCP/IP port number of the server you wish to connect to.
- relative
When used with the extended-status command with the --sleep option, this displays the difference between the current and previous values.
- s, --silent
Be more silent.
- i n, --sleep=n
Executes the command(s) on the command line every n seconds.
- S, --socket=...
Socket file of the server you wish to connect to.
- t n, --timeout=n
Timeout after n seconds when attempting to connect to the server.
- u, --user=#
User for login if not current user.
- V, --version
Output version information and exit.
- w [n], --wait[=n]
Retry n times when a attempts to connect to the server fail. By default, n is 1.
- ?, --help
Display usage information.
- b, --brief
Display results as a brief single line table.
- commit
Move changes from temporary table to the actual grant tables. You must run mysqladmin reload before the changes will take effect.
- copy
Renew the temporary table from the actual grant tables.

-d database, --db=database
The database to which to connect.

--debug=debuglevel
Set the debugging level (0 through 3).

-h host, --host=host
The host whose access rights are examined.

--howto
Usage examples for the program.

-H host, --rhost=host
Connect to a database server on a remote host.

--old-server
Connect to a pre-3.21 MySQL server.

-p password, --password=password
Check the password of the user being examined.

--plan
Display suggestions for future releases.

--preview
Show difference between temporary table and actual grant tables.

-P password, --spassword=password
Administrative password used to access the grant tables.

--relnotes
Display the release notes for the program.

--rollback
Undo the changes made to the temporary table.

-t, --table
Display results in full table format.

-u username, --user=username
User to be examined.

-U username, --superuser=username
Administrative username used to access the grant tables.

-v, --version
Display version information.

mysqlbug

mysqlbug

Report a bug in a MySQL program or utility. This program collects information about your MySQL installation and sends a detailed problem report to the MySQL team.

mysqlcheck

mysqlcheck [*options*] *database* [*tables*]

```
mysqlcheck [options] --databases dbname [dbname ...]
```

```
mysqlcheck [options] --all-databases
```

mysqlcheck provides a convenient way to execute the MySQL CHECK, REPAIR, ANALYZE and OPTIMIZE commands from the command line. This utility can be used while the server is running in contrast with the myisamcheck/isamcheck which should only be used when the server is down.

The mysqlcheck binary can be renamed to change it's default behavior. The possible names for the binary are

```
mysqlrepair
```

The default option is --repair.

```
mysqlanalyze
```

The default option is --analyze.

```
mysqloptimize
```

The default option is --optimize.

```
mysqlcheck
```

The default option is --check.

Options

-A, --all-databases

Check all the databases. This is same as --databases with all databases listed.

-l, --all-in-1

Instead of making one query for each table, execute one query separately for each database. Table names will be in a comma separated list.

-a, --analyze

Enable analyze mode. This is the default when the binary is named mysqlanalyze.

-, --debug=...

Output debug log. Often this is 'd:t:o,filename'

--character-sets-dir=...

Directory where character sets are

-c, --check

Enable check mode. This is the default when the binary is named mysqlcheck.

--compress

Use compression in server/client protocol.

-, --help

Display a help message and exit.

-B, --databases

To operate against several databases. All arguments are regarded as database names.

--default-character-set=...

Set the default character set

-f, --force

Continue even if an sql error is encountered.

- e, --extended
If you are using this option with “check” mode, it will ensure that the table is 100 percent consistent, but will take a long time. If you are using this option with “repair” mode, it will run an extended repair on the table, which may not only take a long time to execute, but may produce a lot of garbage rows also!
- h, --host=...
Connect to host.
- o, --optimize
Enable optimize mode. This is the default behavior with the binary is named `mysqloptimize`.
- p, --password[=...]
Password to use when connecting to server.
- P, --port=...
Port number to use for connection.
- q, --quick
If you are using this option with “check” mode, it prevents the check from scanning the rows to check for wrong links. This is the fastest check. If you are using this option with “repair” mode, it will try to repair only the index tree. This is the fastest repair method for a table.
- r, --repair
Enable repair mode. Can fix almost anything except unique keys that aren't unique.
- s, --silent
Print only error messages.
- S, --socket=...
Socket file to use for connection.
- tables
Specify a the set of tables to operate on. Overrides option `--databases (-B)`.
- u, --user=#
User to connect as.
- v, --verbose
Print info about the various stages.
- V, --version
Output version information and exit.

Check Options

- auto-repair
If a checked table is corrupted, automatically fix it. Repairing will be done after all tables have been checked, if corrupted ones were found.
- C, --check-only-changed
Check only tables that have changed since last check or haven't been closed properly.
- F, --fast
Check only tables that have not been closed properly

-m, --medium-check

Faster than extended-check, but only finds 99.99 percent of all errors. Should be good enough for most cases.

mysqldump

mysqldump [*options*] *database* [*tables*]

mysqldump [*options*] --databases *dbname* [*dbname* ...]

mysqldump [*options*] --all-databases

Outputs the contents of the given database (or table within a database) as a series of ANSI SQL commands. This can be used to back up a database, or copy the contents of a database to another SQL database, MySQL or otherwise. The dump files are ASCII files so they are portable across different machine architectures. This command is also handy for breaking up a database; use the -l and -opt options.

Options

--add-locks

Add lock table and unlock table commands for each table. This will enable you to get faster inserts into MySQL when the dump is read back in.

--add-drop-table

Add a drop table before each create statement.

-A, --all-databases

Dump all the databases. This will be same as --databases with all databases listed.

-a, --all

Include all MySQL-specific create options.

--allow-keywords

Allow creation of column names that are keywords. This is done by prefixing each column name with the table name.

-c, --complete-insert

Use complete insert statements.

-C, --compress

Compress all client/server communication.

-B, --databases

To dump several databases. All name arguments are regarded as database names. use db_name commands will be included in the output for each database.

--delayed

Insert rows with the insert delayed command.

-e, --extended-insert

Use the new multiline insert syntax.

-#, --debug[=option_string]
Trace usage of the program (for debugging).

--help
Display a help message and exit.

--fields-terminated-by=...
--fields-enclosed-by=...
--fields-optionally-enclosed-by=...
--fields-escaped-by=...
--lines-terminated-by=...
These options are used with the -T option and have the same meaning as the corresponding clauses for the LOAD DATA INFILE SQL command.

-F, --flush-logs
Flush log file in the MySQL server before starting the dump.

-f, --force,
Continue even if we get a SQL error during a table dump.

-h, --host=..
Dump data from the MySQL server on the named host.

-l, --lock-tables.
Lock all tables before starting the dump. The tables are locked with READ LOCAL to allow concurrent inserts in the case of MyISAM tables.

-n, --no-create-db
create database statements will not be put in the output file. If this is not specified, create database statements will be added if --databases or --all-databases option are given.

-t, --no-create-info
Don't write create table statements.

-d, --no-data
Don't write any row information for the table. This is very useful if you just want to get a dump of the structure for a database or a table.

--opt
Same as --quick --add-drop-table --add-locks --extended-insert --lock-tables. Should give you the fastest possible dump for reading into a MySQL server.

-ppassword, --password[=password]
The password to use when connecting to the server.

-P port_num, --port=port_num
The TCP/IP port number to use for connecting to a host.

-q, --quick
Dump queries directly to stdout without buffering.

-r, --result-file=...
Direct output to a given file. This option should be used in MSDOS, because it prevents new line '\n' from being converted to '\n\r' (new line + carriage return).

- S --socket=...
Socket file of the server you wish to connect to.
- tables
Specifies tables to copy. Overrides option --databases (-B).
- T, --tab=path-to-some-directory
Creates a table_name.sql file, that contains the SQL commands, and a table_name.txt file, that contains the data, for each give table. This only works if mysqldump is run on the same machine as the mysqld daemon. The format of the .txt file is made according to the --fields-xxx and --lines--xxx options.
- u, --user=#
User for login if not current user.
- O var=option, --set-variable var=option
Set the value of a variable. The variables are listed below.
 - net_buffer_length=#
When creating multi-row-insert statements (as with option --extended-insert or --opt), mysqldump will create rows up to net_buffer_length length. If you increase this variable, you should also ensure that the max_allowed_packet variable in the MySQL server is bigger than the net_buffer_length. net_buffer_length must be less than 16M.
- v, --verbose
Verbose mode. Print out more information on what the program does.
- V, --version
Print version information and exit.
- w, --where='where-condition'
Dump only selected records. The quotes are mandatory:

mysqlhotcopy

mysqlhotcopy *database* [*backup_dir*]

mysqlhotcopy *database* [*database*] *backup_dir*

mysqlhotcopy is a perl script that uses LOCK TABLES, FLUSH TABLES and cp or scp to quickly make a binary backup of a database. It's the fastest way to make a backup of the database. Backups make with mysqlhotcopy are not machine architecture independent.

Options

- , --help
Display a help message and exit
- u, --user=#
User for database login
- p, --password=#
Password to use when connecting to server

- P, --port=#
Port to use when connecting to local server
- S, --socket=#
Socket to use when connecting to local server
- allowold
Don't abort if target already exists (rename it _old)
- keepold
Don't delete previous (now renamed) target when done
- noindices
Don't include full index files in copy to make the backup smaller and faster The indexes can later be reconstructed with myisamchk -rq..
- method=#
Method for copy: cp or scp.
- q, --quiet
Be silent except for errors
- debug
Enable debug
- n, --dryrun
Report actions without doing them
- regexp=#
Copy all databases with names matching regexp
- suffix=#
Suffix for names of copied databases
- checkpoint=#
Insert checkpoint entry into specified db.table
- flushlog
Flush logs once all tables are locked.
- tmpdir=#
Directory to use for temporary files.

mysqlimport

mysqlimport [*options*] *database* [*file*]

Reads a file of data in a variety of common formats (such as comma delimited or fixed width) and inserts the data into a database. A table with the same name as the file must exist in the database with enough columns of the appropriate type to store the data. mysqlimport is essentially a command line version of the LOAD DATA INFILE SQL statement.

Options

- c, --columns=...
A comma-separated list of field names. The field list is used to create a proper LOAD DATA INFILE command, which is then passed to MySQL.

- C, --compress
Compress all client/server communication.
- , --debug[=option_string]
Trace usage of the program (for debugging).
- d, --delete
Empty the table before importing the text file.
- fields-terminated-by=...
- fields-enclosed-by=...
- fields-optionally-enclosed-by=...
- fields-escaped-by=...
- lines-terminated-by=...
- These options have the same meaning as the corresponding clauses for LOAD DATA INFILE.
- f, --force
Ignore errors. If a table for a text file doesn't exist, continue processing any remaining files. Without --force, mysqlimport exits if a table doesn't exist.
- help
Display a help message and exit.
- h host_name, --host=host_name
Import data to the MySQL server on the named host.
- i, --ignore
The --replace and --ignore options control handling of input records that duplicate existing records on unique key values. If you specify --replace, new rows replace existing rows that have the same unique key value. If you specify --ignore, input rows that duplicate an existing row on a unique key value are skipped. If you don't specify either option, an error occurs when a duplicate key value is found, and the rest of the text file is ignored.
- l, --lock-tables
Lock ALL tables for writing before processing any text files. This ensures that all tables are synchronized on the server.
- L, --local
Read input files from the client. By default, text files are assumed to be on the server if you connect to localhost
- p, --password[=your_pass]
The password to use when connecting to the server.
- P port_num, --port=port_num
The TCP/IP port number to use for connecting to a host.
- r, --replace
The --replace and --ignore options control handling of input records that duplicate existing records on unique key values. If you specify --replace, new rows replace existing rows that have the same unique key value. If you specify --ignore, input rows that duplicate an existing row on a unique key value are skipped. If you don't specify either option, an error occurs when a duplicate key value is found, and the rest of the text file is ignored.

- s, --silent
Silent mode. Write output only when errors occur.
 - S /path/to/socket, --socket=/path/to/socket
The socket file to use when connecting to localhost.
 - u user_name, --user=user_name
The MySQL user name to use when connecting to the server. The default value is your Unix login name.
 - v, --verbose
Verbose mode. Print out more information what the program does.
 - V, --version
Print version information and exit.
-

mysqlshow

mysqlshow [*options*] [*database*] [*table*] [*field*]

Displays the layout of the requested database, table or field. If no argument is given, a list of all of the databases is given. With one argument the layout of the given database is show. With two arguments, a table within the database is displayed. If all three arguments are present, the information about a specific field within a table is presented.

Options

- , --help
Display usage information.
- # debuglevel, --debug=debuglevel
Set the debugging level.
- h hostname, --host=hostname
Connect to a remote database server.
- k, --keys
Display the keys of a table.
- p [password], --password=password
Password used to connect to the database server. If no argument is given, the password is asked from the command line.
- P port, --port=port
Port used to connect to a remote database server.
- S file, --socket=file
The Unix socket used to connect to the local database server.
- u username, --user=username
Username used to connect to the database server.
- V, --version
Display version information.