

NAME

mysql_upgrade - check and upgrade MySQL tables

SYNOPSIS

mysql_upgrade [*options*]

DESCRIPTION

mysql_upgrade examines all tables in all databases for incompatibilities with the current version of MySQL Server. **mysql_upgrade** also upgrades the system tables so that you can take advantage of new privileges or capabilities that might have been added.

If **mysql_upgrade** finds that a table has a possible incompatibility, it performs a table check and, if problems are found, attempts a table repair. If the table cannot be repaired, see Section 2.11.3, “Rebuilding or Repairing Tables or Indexes” for manual table repair strategies.

You should execute **mysql_upgrade** each time you upgrade MySQL.

If you install MySQL from RPM packages on Linux, you must install the server and client RPMs. **mysql_upgrade** is included in the server RPM but requires the client RPM because the latter includes **mysqlcheck**. (See Section 2.5.1, “Installing MySQL on Linux Using RPM Packages”.)

Note

On Windows, you must run **mysql_upgrade** with administrator privileges. You can do this by running a Command Prompt as Administrator and running the command. Failure to do so may result in the upgrade failing to execute correctly.

Caution

You should always back up your current MySQL installation *before* performing an upgrade. See Section 7.2, “Database Backup Methods”.

Some upgrade incompatibilities may require special handling before you upgrade your MySQL installation and run **mysql_upgrade**. See Section 2.11.1, “Upgrading MySQL”, for instructions on determining whether any such incompatibilities apply to your installation and how to handle them.

To use **mysql_upgrade**, make sure that the server is running. Then invoke it like this:

```
shell> mysql_upgrade [options]
```

After running **mysql_upgrade**, stop the server and restart it so that any changes made to the system tables take effect.

If you have multiple MySQL server instances running, invoke **mysql_upgrade** with connection parameters appropriate for connecting to the desired server. For example, with servers running on the local host on ports 3306 through 3308, upgrade each of them by connecting to the appropriate port:

```
shell> mysql_upgrade --protocol=tcp -P 3306 [other_options]  
shell> mysql_upgrade --protocol=tcp -P 3307 [other_options]  
shell> mysql_upgrade --protocol=tcp -P 3308 [other_options]
```

For local host connections on Unix, the **--protocol=tcp** option forces a connection using TCP/IP rather than the Unix socket file.

mysql_upgrade executes the following commands to check and repair tables and to upgrade the system tables:

```
mysqlcheck --no-defaults --all-databases  
--fix-db-names --fix-table-names  
mysqlcheck --no-defaults --check-upgrade --all-databases  
--auto-repair  
mysql < fix_priv_tables
```

Notes about the preceding commands:

- **mysql_upgrade** also adds **--write-binlog** or **--skip-write-binlog** to the **mysqlcheck** commands, depending on whether the **--write-binlog** option was specified on the **mysql_upgrade** command.
- Because **mysql_upgrade** invokes **mysqlcheck** with the **--all-databases** option, it processes all tables in all databases, which might take a long time to complete. Each table is locked and therefore unavailable to other sessions while it is being processed. Check and repair operations can be time-consuming, particularly for large tables.
- For details about what checks the **--check-upgrade** option entails, see the description of the FOR UPGRADE option of the CHECK TABLE statement (see Section 13.7.2.2, “CHECK TABLE Syntax”).
- *fix_priv_tables* represents a script generated internally by **mysql_upgrade** that contains SQL statements to upgrade the tables in the mysql database.

All checked and repaired tables are marked with the current MySQL version number. This ensures that next time you run **mysql_upgrade** with the same version of the server, it can tell whether there is any need to check or repair the table again.

mysql_upgrade also saves the MySQL version number in a file named `mysql_upgrade_info` in the data directory. This is used to quickly check whether all tables have been checked for this release so that table-checking can be skipped. To ignore this file and perform the check regardless, use the **--force** option.

mysql_upgrade does not upgrade the contents of the help tables. For upgrade instructions, see Section 5.1.13, “Server-Side Help”.

mysql_upgrade supports the following options, which can be specified on the command line or in the `[mysql_upgrade]` and `[client]` groups of an option file. Unrecognized options are passed to **mysqlcheck**. For information about option files, see Section 4.2.6, “Using Option Files”.

- **--help**
Display a short help message and exit.
- **--basedir=*dir_name***
The path to the MySQL installation directory. This option is accepted for backward compatibility but ignored. It is removed in MySQL 5.7.
- **--character-sets-dir=*dir_name***
The directory where character sets are installed. See Section 10.14, “Character Set Configuration”.
- **--compress**
Compress all information sent between the client and the server if both support compression.
- **--datadir=*dir_name***
The path to the data directory. This option is accepted for backward compatibility but ignored. It is removed in MySQL 5.7.
- **--debug[=*debug_options*], -# [*debug_options*]**
Write a debugging log. A typical *debug_options* string is `d:t:o,file_name`. The default is `d:t:O,/tmp/mysql_upgrade.trace`.
- **--debug-check**
Print some debugging information when the program exits.
- **--debug-info, -T**
Print debugging information and memory and CPU usage statistics when the program exits.
- **--default-auth=*plugin***
A hint about the client-side authentication plugin to use. See Section 6.3.6, “Pluggable Authentication”.

This option was added in MySQL 5.5.10.

- **--default-character-set**=*charset_name*

Use *charset_name* as the default character set. See Section 10.14, “Character Set Configuration”.

- **--defaults-extra-file**=*file_name*

Read this option file after the global option file but (on Unix) before the user option file. If the file does not exist or is otherwise inaccessible, an error occurs. Before MySQL 5.5.8, *file_name* must be the full path name to the file. As of MySQL 5.5.8, the name is interpreted relative to the current directory if given as a relative path name.

- **--defaults-file**=*file_name*

Use only the given option file. If the file does not exist or is otherwise inaccessible, an error occurs. Before MySQL 5.5.8, *file_name* must be the full path name to the file. As of MySQL 5.5.8, the name is interpreted relative to the current directory if given as a relative path name.

- **--defaults-group-suffix**=*str*

Read not only the usual option groups, but also groups with the usual names and a suffix of *str*. For example, **mysql_upgrade** normally reads the [client] and [mysql_upgrade] groups. If the **--defaults-group-suffix**=*_other* option is given, **mysql_upgrade** also reads the [client_*other*] and [mysql_upgrade_*other*] groups.

- **--force**

Ignore the `mysql_upgrade_info` file and force execution even if **mysql_upgrade** has already been executed for the current version of MySQL.

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

Connect to the MySQL server on the given host.

- **--no-defaults**

Do not read any option files. If program startup fails due to reading unknown options from an option file, **--no-defaults** can be used to prevent them from being read.

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

The password to use when connecting to the server. If you use the short option form (**-p**), you *cannot* have a space between the option and the password. If you omit the *password* value following the **--password** or **-p** option on the command line, **mysql_upgrade** prompts for one.

Specifying a password on the command line should be considered insecure. See Section 6.1.2.1, “End-User Guidelines for Password Security”. You can use an option file to avoid giving the password on the command line.

- **--pipe**, **-W**

On Windows, connect to the server using a named pipe. This option applies only if the server supports named-pipe connections.

- **--plugin-dir**=*dir_name*

The directory in which to look for plugins. Specify this option if the **--default-auth** option is used to specify an authentication plugin but **mysql_upgrade** does not find it. See Section 6.3.6, “Pluggable Authentication”.

This option was added in MySQL 5.5.10.

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

The TCP/IP port number to use for the connection.

- **--print-defaults**

Print the program name and all options that it gets from option files.

- **--protocol**={TCP|SOCKET|PIPE|MEMORY}

The connection protocol to use for connecting to the server. It is useful when the other connection parameters normally would cause a protocol to be used other than the one you want. For details on the permissible values, see Section 4.2.2, “Connecting to the MySQL Server”.

- **--shared-memory-base-name=*name***

On Windows, the shared-memory name to use, for connections made using shared memory to a local server. The default value is `MYSQL`. The shared-memory name is case-sensitive.

The server must be started with the **--shared-memory** option to enable shared-memory connections.

- **--socket=*path*, -S *path***

For connections to localhost, the Unix socket file to use, or, on Windows, the name of the named pipe to use.

- **--ssl***

Options that begin with **--ssl** specify whether to connect to the server using SSL and indicate where to find SSL keys and certificates. See Section 6.4.2, “Command Options for Encrypted Connections”.

- **--tmpdir=*dir_name*, -t *dir_name***

The path name of the directory to use for creating temporary files.

- **--upgrade-system-tables, -s**

Upgrade only the system tables, do not upgrade data.

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

The MySQL user name to use when connecting to the server. The default user name is `root`.

- **--verbose**

Verbose mode. Print more information about what the program does.

- **--version-check, -k**

Check the version of the server to which **mysql_upgrade** is connecting to verify that it is the same as the version for which **mysql_upgrade** was built. If not, **mysql_upgrade** exits. This option is enabled by default; to disable the check, use **--skip-version-check**. This option was added in MySQL 5.5.32.

- **--write-binlog**

Cause binary logging to be enabled while **mysql_upgrade** runs. This is the default behavior; to disable binary logging during the upgrade, use the inverse of this option (that is, start the program with **--skip-write-binlog**).

COPYRIGHT

Copyright 1997, 2018, Oracle and/or its affiliates. All rights reserved.

This documentation is free software; you can redistribute it and/or modify it only under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with the program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA or see <http://www.gnu.org/licenses/>.

SEE ALSO

For more information, please refer to the MySQL Reference Manual, which may already be installed locally and which is also available online at <http://dev.mysql.com/doc/>.

AUTHOR

Oracle Corporation (<http://dev.mysql.com/>).