NAME

mysql_upgrade − check and upgrade MySQL tables

SYNOPSIS

mysql_upgrade [*options***]**

DESCRIPTION

Note

As of MySQL 8.0.16, the MySQL server performs the upgrade tasks previously handled by **mysql_upgrade** (for details, see Section 2.11.3, “What the MySQL Upgrade Process Upgrades”). Consequently, **mysql_upgrade** is unneeded and is deprecated as of that version, and will be removed in a future MySQL version. Because **mysql_upgrade** no longer performs upgrade tasks, it exits with status 0 unconditionally.

Each time you upgrade MySQL, you should execute **mysql_upgrade**, which looks for incompatibilities with the upgraded MySQL server:

- It upgrades the system tables in the mysql schema so that you can take advantage of new privileges or capabilities that might have been added.
- It upgrades the Performance Schema, INFORMATION_SCHEMA, and sys schema.
- It examines user schemas.

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 2.11.13, “Rebuilding or Repairing Tables or Indexes” for manual table repair strategies.

mysql_upgrade communicates directly with the MySQL server, sending it the SQL statements required to perform an upgrade.

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* upgrading your MySQL installation and running **mysql_upgrade**. See Section 2.11, “Upgrading MySQL”, for instructions on determining whether any such incompatibilities apply to your installation and how to handle them.

Use **mysql_upgrade** like this:

1. Ensure that the server is running.
2. Invoke **mysql_upgrade** to upgrade the system tables in the mysql schema and check and repair tables in other schemas:

   shell> **mysql_upgrade [***options***]**
3. Stop the server and restart it so that any system table changes take effect.

If you have multiple MySQL server instances to upgrade, invoke **mysql_upgrade** with connection parameters appropriate for connecting to each of the desired servers. For example, with servers running on the local host on parts 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.

By default, **mysql_upgrade** runs as the MySQL root user. If the root password is expired when you run **mysql_upgrade**, you will see a message that your password is expired and that **mysql_upgrade** failed as a result. To correct this, reset the root password to unexpire it and run **mysql_upgrade** again. First, connect

to the server as root:

shell> **mysql −u root −p**
Enter password: **** <− enter root password here

Reset the password using ALTER USER:

mysql> **ALTER USER USER() IDENTIFIED BY '***root−password***';**

Then exit **mysql** and run **mysql_upgrade** again:

shell> **mysql_upgrade [***options***]**

> **Note**
> If you run the server with the disabled_storage_engines system variable set to disable certain storage engines (for example, MyISAM), **mysql_upgrade** might fail with an error like this:
>
> mysql_upgrade: [ERROR] 3161: Storage engine MyISAM is disabled
> (Table creation is disallowed).
>
> To handle this, restart the server with disabled_storage_engines disabled. Then you should be able to run **mysql_upgrade** successfully. After that, restart the server with disabled_storage_engines set to its original value.

Unless invoked with the **−−upgrade−system−tables** option, **mysql_upgrade** processes all tables in all user schemas as necessary. Table checking 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. Table checking uses the FOR UPGRADE option of the CHECK TABLE statement. For details about what this option entails, see Section 13.7.3.2, “CHECK TABLE Syntax”.

mysql_upgrade marks all checked and repaired tables with the current MySQL version number. This ensures that the next time you run **mysql_upgrade** with the same version of the server, it can be determined whether there is any need to check or repair a given table again.

mysql_upgrade 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.

> **Note**
> The mysql_upgrade_info file is deprecated and will be removed in a future MySQL version.

mysql_upgrade checks mysql.user system table rows and, for any row with an empty plugin column, sets that column to 'mysql_native_password' if the credentials use a hash format compatible with that plugin. Rows with a pre−4.1 password hash must be upgraded manually.

mysql_upgrade does not upgrade the contents of the time zone tables or help tables. For upgrade instructions, see Section 5.1.13, “MySQL Server Time Zone Support”, and Section 5.1.14, “Server-Side Help Support”.

Unless invoked with the **−−skip−sys−schema** option, **mysql_upgrade** installs the sys schema if it is not installed, and upgrades it to the current version otherwise. An error occurs if a sys schema exists but has no version view, on the assumption that its absence indicates a user−created schema:

A sys schema exists with no sys.version view. If
you have a user created sys schema, this must be renamed for the
upgrade to succeed.

To upgrade in this case, remove or rename the existing sys schema first.

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. For information about option files used by MySQL programs, see Section 4.2.2.2, “Using Option Files”.

- **--help**

  Display a short help message and exit.

- **--bind-address=***ip_address*

  On a computer having multiple network interfaces, use this option to select which interface to use for connecting to the MySQL server.

- **--character-sets-dir=***dir_name*

  The directory where character sets are installed. See Section 10.15, “Character Set Configuration”.

- **--compress**, **-C**

  Compress all information sent between the client and the server if possible. See Section 4.2.6, “Connection Compression Control”.

  As of MySQL 8.0.18, this option is deprecated. It will be removed in a future MySQL version. See the section called “Legacy Connection Compression Configuration”.

- **--compression-algorithms=***value* The permitted compression algorithms for connections to the server. The available algorithms are the same as for the protocol_compression_algorithms system variable. The default value is uncompressed.

  For more information, see Section 4.2.6, “Connection Compression Control”.

  This option was added in MySQL 8.0.18.

- **--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 which client-side authentication plugin to use. See Section 6.2.17, “Pluggable Authentication”.

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

  Use *charset_name* as the default character set. See Section 10.15, “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. *file_name* is interpreted relative to the current directory if given as a relative path name rather than a full path name.

For additional information about this and other option−file options, see Section 4.2.2.3, “Command-Line Options that Affect Option-File Handling”.

- **−−defaults−file=***file_name*

  Use only the given option file. If the file does not exist or is otherwise inaccessible, an error occurs. *file_name* is interpreted relative to the current directory if given as a relative path name rather than a full path name.

  For additional information about this and other option−file options, see Section 4.2.2.3, “Command-Line Options that Affect Option-File Handling”.

- **−−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.

  For additional information about this and other option−file options, see Section 4.2.2.3, “Command-Line Options that Affect Option-File Handling”.

- **−−force**

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

- **−−get−server−public−key**

  Request from the server the public key required for RSA key pair−based password exchange. This option applies to clients that authenticate with the caching_sha2_password authentication plugin. For that plugin, the server does not send the public key unless requested. This option is ignored for accounts that do not authenticate with that plugin. It is also ignored if RSA−based password exchange is not used, as is the case when the client connects to the server using a secure connection.

  If **−−server−public−key−path=***file_name* is given and specifies a valid public key file, it takes precedence over **−−get−server−public−key**.

  For information about the caching_sha2_password plugin, see Section 6.4.1.3, “Caching SHA-2 Pluggable Authentication”.

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

  Connect to the MySQL server on the given host.

- **−−login−path=***name*

  Read options from the named login path in the .mylogin.cnf login path file. A “login path” is an option group containing options that specify which MySQL server to connect to and which account to authenticate as. To create or modify a login path file, use the **mysql_config_editor** utility. See **mysql_config_editor**(1).

  For additional information about this and other option−file options, see Section 4.2.2.3, “Command-Line Options that Affect Option-File Handling”.

- **−−max−allowed−packet=***value*

  The maximum size of the buffer for client/server communication. The default value is 24MB. The

minimum and maximum values are 4KB and 2GB.

- **--net-buffer-length=***value*

  The initial size of the buffer for client/server communication. The default value is 1MB − 1KB. The minimum and maximum values are 4KB and 16MB.

- **--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.

  The exception is that the .mylogin.cnf file, if it exists, is read in all cases. This permits passwords to be specified in a safer way than on the command line even when **--no-defaults** is used. (.mylogin.cnf is created by the **mysql_config_editor** utility. See **mysql_config_editor**(1).)

  For additional information about this and other option-file options, see Section 4.2.2.3, "Command-Line Options that Affect Option-File Handling".

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

  The password of the MySQL account used for connecting to the server. The password value is optional. If not given, **mysql_upgrade** prompts for one. If given, there must be *no space* between **--password=** or **-p** and the password following it. If no password option is specified, the default is to send no password.

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

  To explicitly specify that there is no password and that **mysql_upgrade** should not prompt for one, use the **--skip-password** option.

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

  On Windows, connect to the server using a named pipe. This option applies only if the server was started with the named_pipe system variable enabled to support named-pipe connections. In addition, the user making the connection must be a member of the Windows group specified by the named_pipe_full_access_group system variable.

- **--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 6.2.17, "Pluggable Authentication".

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

  For TCP/IP connections, the port number to use.

- **--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 result in use of a protocol other than the one you want. For details on the permissible values, see Section 4.2.4, "Connecting to the MySQL Server Using Command

Options”.

- **--server-public-key-path=***file_name*

  The path name to a file containing a client-side copy of the public key required by the server for RSA key pair-based password exchange. The file must be in PEM format. This option applies to clients that authenticate with the sha256_password or caching_sha2_password authentication plugin. This option is ignored for accounts that do not authenticate with one of those plugins. It is also ignored if RSA-based password exchange is not used, as is the case when the client connects to the server using a secure connection.

  If **--server-public-key-path=***file_name* is given and specifies a valid public key file, it takes precedence over **--get-server-public-key**.

  For sha256_password, this option applies only if MySQL was built using OpenSSL.

  For information about the sha256_password and caching_sha2_password plugins, see Section 6.4.1.2, “SHA-256 Pluggable Authentication”, and Section 6.4.1.3, “Caching SHA-2 Pluggable Authentication”.

- **--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.

  This option applies only if the server was started with the shared_memory system variable enabled to support shared-memory connections.

- **--skip-sys-schema**

  By default, **mysql_upgrade** installs the sys schema if it is not installed, and upgrades it to the current version otherwise. The **--skip-sys-schema** option suppresses this behavior.

- **--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.

  On Windows, this option applies only if the server was started with the named_pipe system variable enabled to support named-pipe connections. In addition, the user making the connection must be a member of the Windows group specified by the named_pipe_full_access_group system variable.

- **--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 the section called “Command Options for Encrypted Connections”.

- **--ssl-fips-mode={OFF|ON|STRICT}** Controls whether to enable FIPS mode on the client side. The **--ssl-fips-mode** option differs from other **--ssl-***xxx* options in that it is not used to establish encrypted connections, but rather to affect which cryptographic operations are permitted. See Section 6.5, “FIPS Support”.

  These **--ssl-fips-mode** values are permitted:

  - OFF: Disable FIPS mode.
  - ON: Enable FIPS mode.

- STRICT: Enable “strict” FIPS mode.

Note

If the OpenSSL FIPS Object Module is not available, the only permitted value for **--ssl-fips-mode** is OFF. In this case, setting **--ssl-fips-mode** to ON or STRICT causes the client to produce a warning at startup and to operate in non-FIPS mode.

- **--tls-ciphersuites=***ciphersuite_list*

  The permissible ciphersuites for encrypted connections that use TLSv1.3. The value is a list of one or more colon-separated ciphersuite names. The ciphersuites that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 6.3.2, “Encrypted Connection TLS Protocols and Ciphers”.

  This option was added in MySQL 8.0.16.

- **--tls-version=***protocol_list*

  The permissible TLS protocols for encrypted connections. The value is a list of one or more comma-separated protocol names. The protocols that can be named for this option depend on the SSL library used to compile MySQL. For details, see Section 6.3.2, “Encrypted Connection TLS Protocols and Ciphers”.

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

  Upgrade only the system tables in the mysql schema, do not upgrade user schemas.

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

  The user name of the MySQL account to use for 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**.

- **--write-binlog**

  By default, binary logging by **mysql_upgrade** is disabled. Invoke the program with **--write-binlog** if you want its actions to be written to the binary log.

  When the server is running with global transaction identifiers (GTIDs) enabled (gtid_mode=ON), do not enable binary logging by **mysql_upgrade**.

- **--zstd-compression-level=***level* The compression level to use for connections to the server that use the zstd compression algorithm. The permitted levels are from 1 to 22, with larger values indicating increasing levels of compression. The default zstd compression level is 3. The compression level setting has no effect on connections that do not use zstd compression.

  For more information, see Section 4.2.6, “Connection Compression Control”.

  This option was added in MySQL 8.0.18.

COPYRIGHT

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/).