

Getting Started with MySQL

Abstract

MySQL is the world's most popular open-source database. Despite its powerful features, MySQL is simple to set up and easy to use. Below are some instructions to help you get MySQL up and running in a few easy steps. We also explain how to perform some basic operations with MySQL using the `mysql` client.

Notes

- The following instructions are for MySQL Community Server 5.7 and do not necessarily apply to earlier or other versions of MySQL.
- These are instructions for a fresh installation only. If you are already using MySQL and want to upgrade to a newer version, see [Upgrading MySQL](#).

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#), where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the [MySQL Documentation Library](#).

Document generated on: 2017-09-18 (revision: 53987)

Table of Contents

..... 1

-
- [Installing and Starting MySQL](#)
 - [Connecting to the MySQL Server with the mysql Client](#)
 - [Some Basic Operations with MySQL](#)
 - [Other Important Tasks to Perform](#)
 - [Troubleshooting](#)
 - [Other Helpful Resources](#)

Installing and Starting MySQL

There are different ways to install MySQL. The following covers the easiest methods for installing and starting MySQL on different platforms.

- **Linux.** The easiest way to install MySQL is to use the [MySQL repositories](#):
 - For Yum-based Linux distributions like Oracle Linux, Red Hat Enterprise Linux, and Fedora, follow the instructions in [A Quick Guide to Using the MySQL Yum Repository](#). If your system cannot use the [MySQL Yum repository](#) for some reason, follow the instructions in [Installing MySQL on Linux Using RPM Packages from Oracle](#).
 - For APT-based distributions like Debian and Ubuntu, follow the instructions in [A Quick Guide to Using the MySQL APT Repository](#). If your system cannot use the [MySQL APT repository](#) for some reason, follow the instructions in [Installing MySQL on Linux Using Debian Packages from Oracle](#).
 - For SUSE Linux Enterprise, follow the instructions in [A Quick Guide to Using the MySQL SLES Repository](#). If your system cannot use the [MySQL SUSE repository](#) for some reason, follow the instructions in [Installing MySQL on Linux Using RPM Packages from Oracle](#).

For Linux distributions that do not support the [MySQL repositories](#) or the installation packages mentioned above, you can install MySQL using generic binaries:

- Download the [.tar](#) or [.tar.gz](#) archive for the generic binaries for Linux from the [Download MySQL Community Server](#) page.
- See [Installing MySQL on Unix/Linux Using Generic Binaries](#) for instructions on installing the binaries.
- After installing the binaries, following the instructions given in [Initializing the Data Directory](#). It is especially important to note the random `root` password generated for you during the initialization process; see [Initializing the Data Directory Manually Using mysql](#) for more detail.
- Next, follow the instructions given in [Starting the Server](#).

Detailed instructions, as well as other methods for installation, can be found in [Installing MySQL on Linux](#).

- **Microsoft Windows.** The recommended way to install MySQL on Microsoft Windows is to use the MySQL Installer; see [MySQL Installer Method](#) on how to download and run the MySQL Installer. For a detailed explanation for each step of the installation wizard, see [MySQL Installer for Windows](#).

If you have chosen to configure MySQL as a Windows service during the installation process, which is the default option (see [Windows Service](#) for details), the MySQL server will start automatically after the installation process is completed.

Detailed information regarding Windows installation, including alternative installation methods and instructions for troubleshooting, can be found in [Installing MySQL on Microsoft Windows](#).

- **OS X.** The recommended way for installing MySQL on OS X is to use the OS X installer package. See [Installing MySQL on OS X Using Native Packages](#) on how to download and run the installer package, and how to start the MySQL server afterward.

Detailed information regarding installation on OS X can be found in [Installing MySQL on OS X](#).

- **Other platforms.** For installations on other platforms (for example, FreeBSD and Solaris), as well as installation methods not covered above, see [Installing and Upgrading MySQL](#).

Connecting to the MySQL Server with the mysql Client

Once your MySQL server is up and running, you can connect to it as the superuser `root` with the `mysql` client.

- On Linux, enter the following command at the command line terminal (for installation using generic binaries, you might need to go first to the `bin` folder under the base directory of your MySQL installation):

```
shell> mysql -u root -p
```

- On Windows, click **Start, All Programs, MySQL, MySQL 5.7 Command Line Client**. If you did not install MySQL with the MySQL Installer, open a command prompt, go to the `bin` folder under the base directory of your MySQL installation, and issue the following command:

```
C:\> mysql -u root -p
```

You are then asked for the `root` password, which was assigned in different manners according to the way you installed MySQL. The installation and initialization instructions given above already explain the `root` password, but here is a quick summary:

- For installations using the MySQL Yum repository, MySQL SUSE repository, or RPM packages directly downloaded from Oracle, the generated `root` password is in the error log. View it with, for example, the following command:

```
shell> sudo grep 'temporary password' /var/log/mysqld.log
```

- For installations using the MySQL APT repository or Debian packages directly downloaded from Oracle, you should have already assigned the `root` password yourself; if you have not done that for some reason, see the "Important" note [here](#) or [How to Reset the Root Password](#).
- For installations on Linux using the generic binaries followed by a data directory initialization using `mysqld --initialize` as discussed in [Initializing the Data Directory Manually Using mysqld](#), the generated `root` password is displayed in the standard error stream during the data directory's initialization:

```
[Warning] A temporary password is generated for root@localhost:
iTag*AfrH5ej
```

Note

Depending on the configuration you used to initialize the MySQL server, the error output might have been directed to the [MySQL error log](#); go there and check for the password if you do not see the above message on your screen. The error log is a file with a `.err` extension, usually found under the server's data directory

(the location of which depends on the server's configuration, but is likely to be the `data` folder under the base directory of your MySQL installation, or the `/var/lib/mysql` folder).

If you have initialized the data directory with `mysqld --initialize-insecure` instead, the `root` password is empty.

- For installations on Windows using the MySQL Installer and OS X using the installer package, you should have assigned a `root` password yourself.

If you have forgotten the `root` password you have chosen or have problems finding the temporary `root` password generated for you, see [How to Reset the Root Password](#).

Once you are connected to the MySQL server, a welcome message is displayed and the `mysql>` prompt appears, at which SQL statements are to be sent to the server for execution:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.13 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

At this point, if you have logged in using a temporary `root` password that was generated during the installation or initialization process (which will be the case if you installed MySQL using the MySQL Yum repository, or using RPM packages or generic binaries from Oracle), reset your `root` password with the following statement:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

Until you reset your `root` password, you will not be able to exercise any of the superuser privileges, even if you are logged in as `root`.

Here are a few useful things to remember when using the `mysql` client:

- Client commands (for example, `help`, `quit`, and `clear`) and keywords in SQL statements (for example, `SELECT`, `CREATE TABLE`, and `INSERT`) are case insensitive.
- Column names are case sensitive. Table names are case sensitive on most Unix-like platforms, but case insensitive on Windows platforms. Case sensitivity during string comparison depends on the character collation you use. In general, it is a good idea to treat all identifiers (database names, table names, column names, etc.) and strings as case sensitive. See [Identifier Case Sensitivity](#) and [Case Sensitivity in String Searches](#) for details.
- You can type your SQL statements in multiple lines by pressing `Enter` in the middle of it. Typing a semi-colon (`;`) followed by an `Enter` ends an SQL statement and sends it to the server for execution; the same happens when a statement is ended with `\g` or `\G` (with the latter, returned results are displayed vertically). However, client commands (for example, `help`, `quit`, and `clear`) do not require a terminator.

To disconnect from the MySQL server, type `QUIT` or `\q` at the client:

```
mysql> QUIT
```

Some Basic Operations with MySQL

Here are some basic operations with the MySQL server. [SQL Statement Syntax](#) explains in detail the rich syntax and functionality of the SQL statements that are illustrated below.

Showing existing databases. Use a [SHOW DATABASES](#) statement:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

Creating a new database. Use a [CREATE DATABASE](#) statement:

```
mysql> CREATE DATABASE pets;
Query OK, 1 row affected (0.01 sec)
```

Check if the database has been created:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| pets |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Creating a table inside a database. First, pick the database in which you want to create the table with a [USE](#) statement:

```
mysql> USE pets
Database changed
```

The [USE](#) statement tells MySQL to use `pets` as the default database for subsequent statements. Next, create a table with a [CREATE TABLE](#) statement:

```
CREATE TABLE cats
(
  id          INT unsigned NOT NULL AUTO_INCREMENT, # Unique ID for the record
  name       VARCHAR(150) NOT NULL,                # Name of the cat
  owner      VARCHAR(150) NOT NULL,                # Owner of the cat
  birth      DATE NOT NULL,                         # Birthday of the cat
  PRIMARY KEY (id)                                 # Make the id the primary key
);
```

Data types you can use in each column are explained in [Data Types](#). [Primary Key Optimization](#) explains the concept of a primary key. What follows a `#` on each line is a comment, which is ignored by the `mysql` client; see [Comment Syntax](#) for other comment styles.

Check if the table has been created with a [SHOW TABLES](#) statement:

```
mysql> SHOW TABLES;
+-----+
| Tables_in_pets |
+-----+
| cats           |
+-----+
1 row in set (0.00 sec)
```

[DESCRIBE](#) shows information on all columns of a table:

```
mysql> DESCRIBE cats;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |
| name  | varchar(150)       | NO   |     | NULL    |                |
| owner | varchar(150)       | NO   |     | NULL    |                |
| birth | date                | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Adding records into a table. Use, for example, an [INSERT...VALUES](#) statement:

```
INSERT INTO cats ( name, owner, birth) VALUES
( 'Sandy', 'Lennon', '2015-01-03' ),
( 'Cookie', 'Casey', '2013-11-13' ),
( 'Charlie', 'River', '2016-05-21' );
```

See [Literal Values](#) for how to write string, date, and other kinds of literals in MySQL.

Retrieving records from a table. Use a [SELECT](#) statement, and "*" to match all columns:

```
mysql> SELECT * FROM cats;
+-----+-----+-----+-----+
| id | name   | owner  | birth   |
+-----+-----+-----+-----+
| 1  | Sandy  | Lennon | 2015-01-03 |
| 2  | Cookie | Casey  | 2013-11-13 |
| 3  | Charlie | River  | 2016-05-21 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

To select specific columns and rows by a certain condition using the [WHERE](#) clause:

```
mysql> SELECT name FROM cats WHERE owner = 'Casey';
+-----+
| name |
+-----+
| Cookie |
+-----+
1 row in set (0.00 sec)
```

Deleting a record from a table. Use a [DELETE](#) statement to delete a record from a table, specifying the criterion for deletion with the [WHERE](#) clause:

```
mysql> DELETE FROM cats WHERE name='Cookie';
Query OK, 1 row affected (0.05 sec)

mysql> SELECT * FROM cats;
```

```

+-----+-----+-----+-----+
| id | name   | owner | birth   |
+-----+-----+-----+-----+
| 1 | Sandy  | Lennon | 2015-01-03 |
| 3 | Charlie | River  | 2016-05-21 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

Adding or deleting a column from a table. Use an **ALTER TABLE...ADD** statement to add a column. You can use, for example, an **AFTER** clause to specify the location of the new column:

```

mysql> ALTER TABLE cats ADD gender CHAR(1) AFTER name;
Query OK, 0 rows affected (0.24 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

Use **DESCRIBE** to check the result:

```

mysql> DESCRIBE cats;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name  | varchar(150)        | NO   |     | NULL    |                |
| gender| char(1)             | YES  |     | NULL    |                |
| owner | varchar(150)        | NO   |     | NULL    |                |
| birth | date                | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

SHOW CREATE TABLE shows a **CREATE TABLE** statement, which provides even more details on the table:

```

mysql> SHOW CREATE TABLE cats\G
***** 1. row *****
      Table: cats
Create Table: CREATE TABLE `cats` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(150) NOT NULL,
  `gender` char(1) DEFAULT NULL,
  `owner` varchar(150) NOT NULL,
  `birth` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

```

Use **ALTER TABLE...DROP** to delete a column:

```

mysql> ALTER TABLE cats DROP gender;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE cats;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
| name  | varchar(150)        | NO   |     | NULL    |                |
| owner | varchar(150)        | NO   |     | NULL    |                |
| birth | date                | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

See the [Tutorial](#) for more instructions on how to work with the MySQL server.

Other Important Tasks to Perform

Create more user accounts. `root` is a superuser account for administration of the MySQL server which should not be used for general operations. On how to create user accounts of various kinds, see [Adding User Accounts](#).

Configure MySQL to be managed with systemd. If you have installed MySQL on a systemd platform using generic binaries and want it to be managed with systemd, see [Managing MySQL Server with systemd](#).

Troubleshooting

The following are resources for troubleshooting some problems you might run into:

- [Troubleshooting Problems Starting the MySQL Server](#)
- [Troubleshooting Problems Connecting to MySQL](#)
- [How to Reset the Root Password](#)
- [Common Errors When Using MySQL Programs](#)

Other Helpful Resources

- [MySQL 5.7 Reference Manual](#)
- [Download MySQL Community Server](#)
- [Installing and Upgrading MySQL](#)
- [Using Option Files](#)
- [Tutorial](#) on using the `mysql` client program to work with MySQL
- [What Is New in MySQL 5.7](#)
- [Using MySQL as a Document Store](#)

Legal Notices

Copyright © 1997, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

