

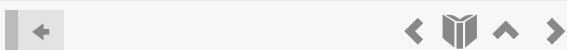


The world's most popular open source database

[Contact MySQL](#)

[Login](#) | [Register](#)

Documentation



[Home](#) [Documentation Home](#)

MySQL 8.0 Reference Manual

- [Preface and Legal Notices](#)
- ▼ [General Information](#)
 - [About This Manual](#)
- ▼ [Overview of the MySQL Database Management System](#)
 - [What is MySQL?](#)
 - **[The Main Features of MySQL](#)**
 - [History of MySQL](#)
- [What Is New in MySQL 8.0](#)
- [Server and Status Variables and Options Added, Deprecated, or Removed in MySQL 8.0](#)
- [MySQL Information Sources](#)
- [How to Report Bugs or Problems](#)
- [MySQL Standards Compliance](#)
- [Credits](#)
- [Installing and Upgrading MySQL](#)
- [Tutorial](#)
- [MySQL Programs](#)
- [MySQL Server Administration](#)

[MySQL 8.0 Reference Manual](#) / ... / [The Main Features of MySQL](#)

version 8.0 ▼

1.2.2 The Main Features of MySQL

This section describes some of the important characteristics of the MySQL Database Software. In most respects, the roadmap applies to all versions of MySQL. For information about features as they are introduced into MySQL on a series-specific basis, see the “In a Nutshell” section of the appropriate Manual:

- MySQL 8.0: [Section 1.3, “What Is New in MySQL 8.0”](#)
- MySQL 5.7: [What Is New in MySQL 5.7](#)
- MySQL 5.6: [What Is New in MySQL 5.6](#)

Internals and Portability

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms. See <https://www.mysql.com/support/supportedplatforms/database.html>.
- For portability, configured using **CMake**.

- › Security
- › Backup and Recovery
- › Optimization
- › Language Structure
- › Character Sets, Collations, Unicode
- › Data Types
- › Functions and Operators
- › SQL Statements
- › MySQL Data Dictionary
- › The InnoDB Storage Engine
- › Alternative Storage Engines
- › Replication
 - MySQL Shell
- › Using MySQL as a Document Store
- › InnoDB Cluster
- › MySQL NDB Cluster 8.0
- › Partitioning
- › Stored Objects
- › INFORMATION_SCHEMA Tables
- › MySQL Performance Schema
- › MySQL sys Schema
- › Connectors and APIs
- › Extending MySQL
- › MySQL Enterprise Edition
 - MySQL Workbench
- › MySQL on the OCI Marketplace
- › MySQL 8.0 Frequently Asked Questions
- › Error Messages and Common Problems
- › Indexes
 - MySQL Glossary

- Tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool (<http://developer.kde.org/~sewardj/>).
- Uses multi-layered server design with independent modules.
- Designed to be fully multithreaded using kernel threads, to easily use multiple CPUs if they are available.
- Provides transactional and nontransactional storage engines.
- Uses very fast B-tree disk tables (`MyISAM`) with index compression.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Uses a very fast thread-based memory allocation system.
- Executes very fast joins using an optimized nested-loop join.
- Implements in-memory hash tables, which are used as temporary tables.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

Data Types

- Many data types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, `FLOAT`, `DOUBLE`, `CHAR`, `VARCHAR`, `BINARY`, `VARBINARY`, `TEXT`, `BLOB`, `DATE`, `TIME`, `DATETIME`, `TIMESTAMP`, `YEAR`, `SET`, `ENUM`, and OpenGIS spatial types. See [Chapter 11, Data Types](#).



Related Documentation

[MySQL 8.0 Release Notes](#)

[MySQL 8.0 Source Code Documentation](#)



Download this Manual

[PDF \(US Ltr\)](#) - 40.6Mb

[PDF \(A4\)](#) - 40.7Mb

[PDF \(RPM\)](#) - 40.4Mb

[HTML Download \(TGZ\)](#) - 10.6Mb

[HTML Download \(Zip\)](#) - 10.7Mb

[HTML Download \(RPM\)](#) - 9.2Mb

[Man Pages \(TGZ\)](#) - 239.9Kb

[Man Pages \(Zip\)](#) - 343.7Kb

[Info \(Gzip\)](#) - 3.9Mb

[Info \(Zip\)](#) - 3.9Mb



Excerpts from this Manual

- Fixed-length and variable-length string types.

Statements and Functions

- Full operator and function support in the SELECT list and WHERE clause of queries. For example:

```
1  mysql> SELECT CONCAT(first_name, ' ', last_name)
2      -> FROM citizen
3      -> WHERE income/dependents > 10000 AND age > 30;
```

- Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions (COUNT(), AVG(), STD(), SUM(), MAX(), MIN(), and GROUP CONCAT()).
- Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax.
- Support for aliases on tables and columns as required by standard SQL.
- Support for DELETE, INSERT, REPLACE, and UPDATE to return the number of rows that were changed (affected), or to return the number of rows matched instead by setting a flag when connecting to the server.
- Support for MySQL-specific SHOW statements that retrieve information about databases, storage engines, tables, and indexes. Support for the INFORMATION_SCHEMA database, implemented according to standard SQL.
- An EXPLAIN statement to show how the optimizer resolves a query.
- Independence of function names from table or column names. For example, ABS is a valid column name. The only restriction is that for a function call, no spaces are permitted between the function name and the "(" that follows it. See [Section 9.3, "Keywords and Reserved Words"](#).

- You can refer to tables from different databases in the same statement.

Security

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Scalability and Limits

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width for InnoDB tables is either 767 bytes or 3072 bytes. See [Section 15.22, “InnoDB Limits”](#). The maximum index width for MyISAM tables is 1000 bytes. See [Section 16.2, “The MyISAM Storage Engine”](#). An index may use a prefix of a column for CHAR, VARCHAR, BLOB, or TEXT column types.

Connectivity

- Clients can connect to MySQL Server using several protocols:
 - Clients can connect using TCP/IP sockets on any platform.
 - On Windows systems, clients can connect using named pipes if the server is started with the named_pipe system variable enabled. Windows servers also support shared-memory connections if started with the shared_memory system variable enabled. Clients can connect through shared memory by using the --protocol=memory option.
 - On Unix systems, clients can connect using Unix domain socket files.

- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages. See [Chapter 28, Connectors and APIs](#).
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections. For example, you can use MS Access to connect to your MySQL server. Clients can be run on Windows or Unix. Connector/ODBC source is available. All ODBC 2.5 functions are supported, as are many others. See [MySQL Connector/ODBC Developer Guide](#).
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available. See [MySQL Connector/J 5.1 Developer Guide](#).
- MySQL Connector/NET enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools. Developers can build applications using their choice of .NET languages. MySQL Connector/NET is a fully managed ADO.NET driver written in 100% pure C#. See [MySQL Connector/NET Developer Guide](#).

Localization

- The server can provide error messages to clients in many languages. See [Section 10.12, “Setting the Error Message Language”](#).
- Full support for several different character sets, including `latin1` (cp1252), `german`, `big5`, `ujis`, several Unicode character sets, and more. For example, the Scandinavian characters “å”, “ä” and “ö” are permitted in table and column names.
- All data is saved in the chosen character set.

- Sorting and comparisons are done according to the default character set and collation. It is possible to change this when the MySQL server is started (see [Section 10.3.2, “Server Character Set and Collation”](#)). To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many different character sets that can be specified at compile time and runtime.
- The server time zone can be changed dynamically, and individual clients can specify their own time zone. See [Section 5.1.14, “MySQL Server Time Zone Support”](#).

Clients and Tools

- MySQL includes several client and utility programs. These include both command-line programs such as [mysqldump](#) and [mysqladmin](#), and graphical programs such as [MySQL Workbench](#).
- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the [mysqlcheck](#) client. MySQL also includes [myisamchk](#), a very fast command-line utility for performing these operations on MYISAM tables. See [Chapter 4, MySQL Programs](#).
- MySQL programs can be invoked with the `--help` or `-?` option to obtain online assistance.

[◀ PREV](#) [HOME](#) [UP](#) [NEXT ▶](#)

