



MySQL

database management system



www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

MySQL is the most popular and a free Open Source Relational Database Management System (RDBMS). An RDBMS system stores the data in the form of tables that might be related to each other. MySQL uses Structured Query Language (SQL) to store, manage and retrieve data, and control the accessibility to the data. It is one of the best RDBMS being used for developing web-based software applications.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer.

MySQL works on many system platforms, such as, Linux, macOS, Microsoft Windows, AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM i, IRIX, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64.

This tutorial will give you quick start with MySQL and make you comfortable with MySQL programming.

MySQL Examples

Consider an example table CUSTOMERS created in the MySQL database. This table contains the details of customers like ID, NAME, AGE, ADDRESS, SALARY.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	Hyderabad	4500.00
7	Muffy	24	Indore	10000.00

Since MySQL uses SQL to manage data, it also uses almost all DDL, DML and DCL statements. For instance, the following DML statement lists the records of all customers who are 25 years old.

```
SELECT * FROM CUSTOMERS WHERE AGE = 25;
```

Following records are displayed as a result-set –

ID	NAME	AGE	ADDRESS	SALARY
2	Khilan	25	Delhi	1500.00
4	Chaitali	25	Mumbai	6500.00

MySQL Online Editor

In this tutorial, we provide a MySQL Online Editor which helps you to Edit and Execute the MySQL code directly from your browser. Click on the  icon to run the following MySQL code and retrieve the records matching with the given condition.

```
SELECT * FROM CUSTOMERS WHERE NAME = 'Chaitali';
```

This Online Editor will save you the time to install the MySQL setup in order to execute any query. Try our [MySQL Online Editor](#) now.

Why to Learn MySQL?

MySQL is generally faster, more secure and efficient than other relational databases. Some of world's fastest growing organizations make use of MySQL to efficiently power their high-volume web sites, business-critical systems and packaged software.

However, whether you choose MySQL for your application or not, totally depends on your organization's resources and aim. Usually, MySQL is used by smaller businesses that do not have large data sets, because of its cost efficiency and simple setup.

MySQL Jobs and Opportunities

MySQL professionals are in high demand as more and more organizations are using this open-source database system for efficient data management. If you have the skills, you could earn an average salary of around \$150,000 per year, but it will depend on your location, experience, and job role.

Here are some of the top companies actively looking for MySQL experts for roles like Database Administrator, Database Developer, Database Tester, Data Scientist, ETL Developer, Database Migration Expert, Cloud Database Expert, and more. They need people who can manage and optimize their databases, build data-driven applications, and extract insights from large datasets –

- Google
- Amazon
- Netflix
- Infosys
- Tata Consultancy Services (TCS)
- Tech Mahindra
- Wipro
- Pinterest

- Uber
- Wipro
- Trello
- And many more...

To get started, you can use our user-friendly tutorials, which are designed to help you learn MySQL and prepare for technical interviews or certification exams. You can learn at your own pace, anytime and anywhere.

With the right MySQL skills and knowledge, you can kickstart a rewarding career in the ever-expanding field of data management and analytics. You could be the part of professionals who are driving innovation and data-driven decision-making in some of the world's most renowned companies.

Who should learn MySQL?

This MySQL tutorial has been prepared for beginners to help them understand the basics to advanced concepts related to MySQL database.

Prerequisites to learn MySQL

Before you start doing practice with various types of examples given in this reference, I'm making an assumption that you are already aware about what is database, especially RDBMS and what is a computer programming language.

Frequently Asked Questions about MySQL

Following are very Frequently Asked Questions(FAQ) about MySQL, and this section tries to answer them briefly.

What is MySQL and how it works?

MySQL is a popular open-source relational database management system (RDBMS). It organizes data into tables with rows and columns. Users can interact with MySQL using SQL (Structured Query Language) to perform operations like inserting, updating, and querying data. The system works by processing SQL commands to manage and retrieve data efficiently.

Who developed MySQL?

MySQL was developed by Swedish company MySQL AB, founded by David Axmark, Allan Larsson, and Michael "Monty" Widenius. It was later bought by Sun Microsystems in 2008, which was subsequently acquired by Oracle Corporation in 2010.

How to install MySQL?

You can install MySQL from the MySQL Installer Community, along with other MySQL products you require. The MySQL Installer will allow you to install a certain version of MySQL or you can customize the installation as per your requirements. For more detailed information on how to install MySQL, [click here](#).

What are data types in MySQL?

Since MySQL uses SQL to store and manage the data, the data types used in MySQL are also the same as data types in SQL. Following are three categories of SQL data types.

- String Data types.
- Numeric Data types.
- Date and time Data types.

How to learn MySQL?

Here is the summarized list of tips which you can follow to start learning MySQL.

- Install MySQL database on your computer system.
- Follow our tutorial step by step from the very beginning.
- Read more articles, watch online courses or buy a book on MySQL to enhance your knowledge.
- Try to develop a small software using PHP or Python which makes use of the MySQL database.

How long does it take to learn MySQL?

The time it takes to learn MySQL varies, but basic proficiency can be gained within a few weeks with consistent practice. Mastering more advanced features may take a few months of dedicated learning and hands-on experience. Regular practice and real-world application contribute to the speed of learning MySQL.

What is the latest version of MySQL?

The latest version of MySQL was 8.0. Upgrading to the latest version is recommended for security and feature enhancements.

How to check the MySQL Version in Linux?

To check the MySQL version in Linux, you can use the following command in the terminal:

- `mysql --version`

This command will display the MySQL client version. If you want to check the server version, you can use:

- `mysql -u your_username -p -e "SELECT version();"`

Replace "your_username" with your MySQL username, and you will be prompted to enter your password. After entering the password, the command will display the MySQL server version.

How do I access my MySQL database?

To access your MySQL database, you can use the MySQL command-line client or a graphical user interface (GUI) tool. Here are the basic steps for both:

Using MySQL Command-Line Client

- Open terminal/command prompt.
- Enter: `mysql -u your_username -p`.
- Enter your password when prompted.

Using GUI Tool (e.g., MySQL Workbench)

- Download and install the tool.
- Create a new connection with your details.
- Test the connection.
- Use the GUI to manage your MySQL database.

What are the main features of MySQL?

MySQL is a popular open-source relational database management system (RDBMS) known for its ease of use and scalability. Its main features include support for SQL queries, efficient data storage and retrieval, and robust transaction management, making it suitable for a wide range of applications, from small websites to large enterprise systems. Additionally, MySQL offers strong security measures and a vibrant community for support and development.

How to Start, Stop, and Restart MySQL Server?

To start, stop, or restart the MySQL server, you can use the command-line interface. The exact commands may vary depending on your operating system. Here are instructions for different operating systems:

Windows:

- To Start MySQL Server: Open a command prompt with administrator privileges and run the following command: **net start mysql**.
- To Stop MySQL Server: Open a command prompt with administrator privileges and run the following command: **net stop mysql**.
- To Restart MySQL Server: You can stop and start the MySQL service using the commands mentioned above. Alternatively, you can use the MySQL Notifier or the Services application to restart the MySQL service.

Linux (Ubuntu/Debian):

- To Start MySQL Server: **sudo service mysql start**.
- To Stop MySQL Server: **sudo service mysql stop**.
- To Restart MySQL Server: **sudo service mysql restart**.

macOS:

- To Start MySQL Server: **sudo brew services start mysql**.
- To Stop MySQL Server: **sudo brew services stop mysql**.
- To Restart MySQL Server: **sudo brew services restart mysql**.

These are general commands, and depending on your specific setup, you might need to adjust them. Also, note that on Linux, the service management commands may vary depending on the distribution (e.g., Ubuntu, CentOS).

Remember to replace "mysql" with the actual service name if it's different in your system.

What is MySQL Schema?

A MySQL schema can simply be defined as a blueprint of the database. It stores all the information of the tables, its attributes and entities. As MySQL is a relational database

management system, it is important to have schema as it also represents the relationship between the attributes and entities of multiple tables.

Which is the best place to learn MySQL?

As a beginner, you can use our simple and the best MySQL tutorial to learn MySQL. We have removed all the unnecessary complexity while teaching you these MySQL concepts. You can start learning it now: [Learn MySQL](#).

Copyright & Disclaimer

© Copyright 2025 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Prerequisites to learn MySQL.....	iii
Copyright & Disclaimer.....	vi
Table of Contents	vii
 MYSQL TUTORIAL	 XV
1. MySQL - Introduction	16
2. MySQL - Features.....	19
3. MySQL - Versions.....	21
4. MySQL - Variables.....	25
5. MySQL - Installation	32
6. MySQL - Administration.....	40
7. MySQL - PHP Syntax	44
8. MySQL - Node.js Syntax	49
9. MySQL - Java Syntax.....	53
10. MySQL - Python Syntax	57
11. MySQL - Connection	60
12. MySQL - Workbench.....	64
 MYSQL DATABASES.....	 78
13. MySQL - Create Database.....	79
14. MySQL - Drop Database	85
15. MySQL - Select Database (USE Statement)	92
16. MySQL - SHOW Databases	98
17. MySQL - Copy Database	107
18. MySQL - Database Export.....	114
19. MySQL - Database Import	118
20. MySQL - Database Info	121

MYSQL USERS.....	127
21. MySQL - Create Users.....	128
22. MySQL - Drop User	138
23. MySQL - Show Users.....	146
24. MySQL - Change Password.....	156
25. MySQL - Grant Privileges.....	164
26. MySQL - Show Privileges	178
27. MySQL - REVOKE Statement	190
28. MySQL - Lock User Account.....	201
29. MySQL - Unlock User Account.....	208
MYSQL TABLES.....	215
30. MySQL - Create Tables.....	216
31. MySQL - Show Tables.....	227
32. MySQL - ALTER Command.....	239
33. MySQL - Rename Tables.....	251
34. MySQL - Clone Tables.....	261
35. MySQL - Truncate Table	275
36. MySQL - Temporary Tables	287
37. MySQL - Repair Tables.....	295
38. MySQL - DESCRIBE Tables	304
39. MySQL - Add/Delete Columns	317
40. MySQL - Show Columns.....	331
41. MySQL - Rename Columns.....	343
42. MySQL - Table Locking.....	355
43. MySQL - Drop Tables	364
44. MySQL - Derived Tables	371
MYSQL QUERIES.....	382

45. MySQL – Queries	383
46. MySQL – Constraints	387
47. MySQL - Insert Query	395
48. MySQL - Select Query	407
49. MySQL - Update Query.....	418
50. MySQL - Delete Query	429
51. MySQL - REPLACE Query	440
52. MySQL - Insert Ignore.....	450
53. MySQL – Insert on Duplicate Key Update.....	461
54. MySQL - Insert Into Select.....	475
MYSQL VIEWS.....	489
55. MySQL - CREATE VIEW	490
56. MySQL - Update View.....	500
57. MySQL - Drop View.....	511
58. MySQL - Rename View	521
MYSQL INDEXES	530
59. MySQL - INDEXES.....	531
60. MySQL - Create Index.....	533
61. MySQL - Drop Index.....	542
62. MySQL - Show Indexes	552
63. MySQL - Unique Index.....	564
64. MySQL - Clustered Index	575
65. MySQL - Non-Clustered Index.....	578
MYSQL OPERATORS AND CLAUSES.....	582
66. MySQL - WHERE Clause.....	583
67. MySQL - Limit.....	592
68. MySQL - Distinct Clause.....	604

69.	MySQL - ORDER BY CLAUSE.....	615
70.	MySQL - GROUP BY Clause.....	628
71.	MySQL - Having Clause.....	640
72.	MySQL - AND Operator	650
73.	MySQL - OR Operator	663
74.	MySQL - LIKE Operator	677
75.	MySQL - IN Operator	688
76.	MySQL - ANY Operator.....	701
77.	MySQL - Exists Operator.....	714
78.	MySQL - NOT Operator.....	727
79.	MySQL - NOT EQUAL Operator	742
80.	MySQL - IS NULL Operator	754
81.	MySQL - IS NOT NULL Operator	765
82.	MySQL - Between Operator	776
83.	MySQL - UNION Operator	790
84.	MySQL - UNION vs UNION ALL.....	803
85.	MySQL - MINUS Operator	807
86.	MySQL - Intersect Operator	819
87.	MySQL - Interval Operator.....	830
	MYSQL JOINS.....	841
88.	MySQL - Using Joins.....	842
89.	MySQL - Inner Join.....	850
90.	MySQL - Left Join	861
91.	MySQL - Right Join.....	874
92.	MySQL - Cross Join.....	886
93.	MySQL - Full Join.....	900
94.	MySQL - Self Join	905
95.	MySQL - Delete Join.....	915

96. MySQL - Update Join	924
97. MySQL - Union vs Join	934
MYSQL KEYS	939
98. MySQL - Unique Key	940
99. MySQL - Primary Key	950
100. MySQL - Foreign Key.....	959
101. MySQL - Composite Key	968
102. MySQL - Alternate Key	976
MYSQL TRIGGERS	980
103. MySQL - Triggers.....	981
104. MySQL - CREATE TRIGGER.....	984
105. MySQL - SHOW TRIGGERS.....	992
106. MySQL - DROP TRIGGER.....	1003
107. MySQL - Before Insert Trigger	1009
108. MySQL - After Insert Trigger	1019
109. MySQL - Before Update Trigger	1022
110. MySQL - After Update Trigger.....	1032
111. MySQL - Before Delete Trigger.....	1042
112. MySQL - After Delete Trigger	1056
MYSQL DATA TYPES	1070
113. MySQL - Data Types.....	1071
114. MySQL - VARCHAR.....	1074
115. MySQL - Boolean Datatype	1082
116. MySQL - ENUM	1095
117. MySQL - DECIMAL.....	1107
118. MySQL - INT	1117
119. MySQL - FLOAT	1128

120. MySQL - BIT	1139
121. MySQL - TINYINT	1149
122. MySQL - BLOB	1157
123. MySQL - SET	1162
MYSQL REGULAR EXPRESSIONS.....	1175
124. MySQL - Regular Expressions.....	1176
125. MySQL - RLIKE Operator.....	1182
126. MySQL - NOT LIKE Operator.....	1193
127. MySQL - NOT REGEXP Operator.....	1207
128. MySQL - REGEXP_INSTR() Function	1216
129. MySQL - REGEXP_LIKE() Function	1226
130. MySQL - REGEXP_REPLACE() Function.....	1235
131. MySQL - REGEXP_SUBSTR() Function	1247
MYSQL FULLTEXT SEARCH	1256
132. MySQL - Full-Text Search.....	1257
133. MySQL - Natural Language Fulltext Search	1268
134. MySQL - Boolean Full-Text Search	1278
135. MySQL - Query Expansion Full-Text Search.....	1291
136. MySQL - ngram Full-Text Parser.....	1302
MYSQL FUNCTIONS & OPERATORS.....	1314
137. MySQL - Date and Time Functions.....	1315
138. MySQL - Arithmetic Operators.....	1321
139. MySQL - Numeric Functions.....	1322
140. MySQL - String Functions	1326
141. MySQL - Aggregate Functions	1332
MYSQL MISC CONCEPTS	1334
142. MySQL - NULL Values	1335

143. MySQL - Transactions.....	1345
144. MySQL - Sequences	1367
145. MySQL - Handling Duplicates.....	1379
146. MySQL - SQL Injection	1391
147. MySQL - SubQuery.....	1393
148. MySQL - Comments	1405
149. MySQL - Check Constraint.....	1413
150. MySQL - Storage Engines.....	1428
151. MySQL - Export Table into CSV File.....	1443
152. MySQL - Import CSV into database.....	1456
153. MySQL - UUID	1475
154. MySQL - Common Table Expression (WITH).....	1480
155. MySQL - On Delete Cascade.....	1484
156. MySQL - Upsert.....	1488
157. MySQL - Horizontal Partitioning	1493
158. MySQL - Vertical Partitioning.....	1505
159. MySQL - Cursors.....	1510
160. MySQL - Stored Functions.....	1514
161. MySQL - SIGNAL Statement	1518
162. MySQL - RESIGNAL Statement	1522
163. MySQL - Character Set.....	1536
164. MySQL - Collation	1541
165. MySQL - Wildcards.....	1546
166. MySQL - Aliases	1553
167. MySQL - ROLLUP	1561
168. MySQL - Today Date	1573
169. MySQL - Literals	1580
170. MySQL - Stored Procedure.....	1598

171. MySQL - EXPLAIN	1617
172. MySQL - JSON	1623
173. MySQL - Standard Deviation	1638
174. MySQL - Find Duplicate Records	1642
175. MySQL - Delete Duplicate Records	1657
176. MySQL - Select Random Records	1672
177. MySQL - Show Processlist	1685
178. MySQL - Change Column Type	1698
179. MySQL - Reset Auto-Increment	1710
180. MySQL - COALESCE() Function	1724

MySQL Tutorial

1. MySQL - Introduction

What is a Database?

A database is used to store a collection of data (which can either be structured or unstructured). Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data storages can also be used to manage data, such as files on the file system or large hash tables in memory, but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. In such a database, the data is stored in a structured way with the help of different tables. Relations are established among these tables using primary keys or other keys known as **Foreign Keys**.

A **Relational Database Management System (RDBMS)** is a software that –

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- **Database** – A database is a collection of tables, with related data.
- **Table** – A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** – One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** – A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** – Storing data twice, redundantly to make the system faster.
- **Primary Key** – A primary key is unique. A key value cannot occur twice in one table. With a key, you can only find one row.
- **Foreign Key** – A foreign key is the linking pin between two tables.
- **Compound Key** – A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** – An index in a database resembles an index at the back of a book.

- **Referential Integrity** – Referential Integrity makes sure that a foreign key value always points to an existing row.

MySQL Database

MySQL is a fast, easy-to-use RDBMS and is being used by many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

History of MySQL

Here is the history of MySQL –

- Development of MySQL by Michael Widenius & David Axmark beginning in 1994.
- First internal release on 23rd May 1995.
- Windows Version was released on the 8th January 1998 for Windows 95 and NT.
- Version 3.23: beta from June 2000, production release January 2001.
- Version 4.0: beta from August 2002, production release March 2003 (unions).
- Version 4.1: beta from June 2004, production release October 2004.
- Version 5.0: beta from March 2005, production release October 2005.
- Sun Microsystems acquired MySQL AB on the 26th February 2008.
- Version 5.1: production release 27th November 2008.
- Oracle acquired Sun Microsystems on 27th January 2010.
- Version 5.5: general availability on 3rd December 2010
- Version 5.6: general availability on 5th February 2013
- Version 5.7: general availability on 21st October 2015

- Version 8.0: general availability on 19th April 2018

Before You Begin

Before you begin this tutorial, you should have a basic knowledge of the information covered in our PHP and HTML tutorials.

This tutorial focuses heavily on using MySQL in a PHP environment. Many examples given in this tutorial will be useful for PHP Programmers.

We recommend you check our [PHP Tutorial](#) for your reference.

2. MySQL - Features

MySQL is a type of relational database that stores and manages the data based on Structured Query Language (SQL) queries. Thus, making it a structured database, i.e., the data stored in this relational database is in the form of tables.

It is a fast, easy-to-use RDBMS being used for many small and big businesses, it is developed, marketed and supported by a Swedish Company called MySQL AB.

Features of MySQL

One of the major reasons MySQL is considered one of the most popular relational databases is because of its abundant features. Let us look at them one by one –

Open-Source

MySQL is open-source, which means this software can be downloaded, used and modified by anyone. It is free-to-use and easy-to-understand. The source code of MySQL can be studied, and changed based on the requirements. It uses GPL, i.e. GNU General Public license which defines rules and regulations regarding what can and can't be done while using the application.

Quick and Reliable

MySQL stores data efficiently in the memory ensuring that data is consistent, and not redundant. Hence, data access and manipulation using MySQL is quick. It is considered one of the fastest relational databases with higher productivity as well.

High Performance

MySQL provides comparatively higher performance without affecting its functionality. It also has a very little memory leakage making it memory efficient as well.

Scalable

Scalability refers to the ability of systems to work easily with small amounts of data, large amounts of data, clusters of machines, and so on. MySQL server was developed to work with large databases.

Data Types

It contains multiple data types such as unsigned integers, signed integers, float (FLOAT), double (DOUBLE), character (CHAR), variable character (VARCHAR), text, blob, date, time, datetime, timestamp, year, and so on.

Character Sets

It supports different character sets, and this includes latin1 (cp1252 character encoding), German, Ujis, other Unicode character sets and so on.

Secure

It provides a secure interface since it has a password system which is flexible, and ensures that it is verified based on the host before accessing the database. The password is encrypted while connecting to the server.

Support for large databases

It comes with support for large databases, which could contain about 40 to 50 million records, 150,000 to 200,000 tables and up to 5,000,000,000 rows.

Platform Independent

MySQL can be run on various operating systems including Windows, Linux, macOS etc. It can be used with several programming languages like C, C++, Java, Python, Perl, PHP etc.

Client and Utility Programs

MySQL server also comes with many client and utility programs. This includes Command line programs such as 'mysqladmin' and graphical programs such as 'MySQL Workbench'. MySQL client programs are written in a variety of languages. Client library (code encapsulated in a module) can be written in C or C++ and would be available for clients that have C bindings.

3. MySQL - Versions

Versions are introduced in any product to upgrade by adding extra features and removing unnecessary ones, fixing the bugs etc. The process of versioning is actually important to make the product more efficient with growing technology.

A product is generally released after performing phases of testing: alpha testing, beta testing, gamma testing, and then it is produced once all these tests are passed.

Whenever MySQL is installed, we must choose the version to install along with its distribution format. The latest version of MySQL is **8.0** with its minor version being **8.0.34**. To install the MySQL server, there are two methods: Development release and General Availability.

The Development Release contains all the latest features but it is not recommended to be used in production. The General Availability Release is more of a production release which can be stably used in production.

MySQL Versions

MySQL is a fast and easy-to-use RDBMS and is being used by many small and big businesses. It is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL was first internally released on 23rd May, 1995, and until Oracle acquired Sun Microsystems, version 3.19 to version 5.1 were released.

Version 5.1

The version 5.1 of MySQL had its production release on 27th November, 2008 by adding extra features like event scheduler, partitioning, plugin API, row-based replication, server log tables.

But, version 5.1 contained 20 known bugs that gave out wrong results, along with the 35 bugs from version 5.0. However, almost all of them are fixed as of release 5.1.51. Also, MySQL 5.1 and 6.0 (in alpha test phase) showed poor performance in data warehousing, which could partially be due to its inability to utilize multiple CPU cores for processing a single query.

Version 5.5

MySQL Server 5.5 was a General Availability Release as of December 2010. The improved features in this version include –

- The default storage engine is InnoDB with improved I/O subsystem, which supports transactions and referential integrity constraints.
- Improved SMP support
- Semi-synchronous replication.
- SIGNAL and RESIGNAL statement was added in compliance with the SQL standard.
- Support for supplementary Unicode character sets utf16, utf32, and utf8mb4.

- New options for user-defined partitioning.

Version 5.6

The General Availability of MySQL 5.6 was released in February 2013. New features of this version included:

- Query optimizer has more efficient performance.
- Higher transactional throughput in InnoDB.
- New NoSQL-style memcached APIs.
- Improvements to partitioning for querying large tables.
- Better management of very large tables.
- TIMESTAMP column type that correctly stores milliseconds.
- Improvements to replication.
- Better performance monitoring by expanding the data available through the PERFORMANCE_SCHEMA.
- The InnoDB storage engine also provides support for full-text search and improved group commit performance.

Version 5.7

MySQL 5.7 was made generally available in October 2015. For the minor versions of MySQL 5.7, MySQL 5.7.8 and later, a support for a native JSON data type defined by RFC 7159 by August 2015.

Version 8.0

MySQL Server 8.0 was announced in April 2018, with new improved features. Currently, the minor versions in MySQL 8.0 start from 8.0.0 to 8.0.34. Previous MySQL Server 8.0.0-dmr (as a Development Milestone Release) was announced on 12th September, 2016.

Features Added in MySQL 8.0

The latest version of MySQL is 8.0. Some of the newly features added to it are discussed below:

- **Data dictionary** – In previous MySQL releases, dictionary data was stored in metadata files and non-transactional tables. MySQL now incorporates a transactional data dictionary that stores information about database objects.
- **Atomic Data Definition Language(Atomic DDL) statements** – An Atomic DDL statement combines the data dictionary updates, storage engine operations, and binary log writes associated with a DDL operation into a single, atomic transaction.
- **Upgrade procedure** – Previously, after installation of a new version of MySQL, the MySQL server automatically upgrades the data dictionary tables at the next startup, after which the DBA is expected to invoke `mysql_upgrade` manually to upgrade the system tables in the `mysql` schema, as well as objects in other schemas such as the 'sys' schema and user schemas.

As of MySQL 8.0.16, the server also performs the tasks previously handled by `mysql_upgrade`. In addition, the server updates the contents of the help tables as well. A

new `--upgrade` server option provides control over how the server performs automatic data dictionary and server upgrade operations.

- **Session Reuse** – MySQL Server now supports SSL session reuse by default with a timeout setting to control how long the server maintains a session cache that establishes the period during which a client is permitted to request session reuse for new connections. All MySQL client programs support session reuse. In addition, C applications now can use the C API capabilities to enable session reuse for encrypted connections.
- **Security and account management** – The security is improved greatly and greater DBA flexibility in account management is also enabled.
- **Resource management** – MySQL now supports creation and management of resource groups, and permits assigning threads running within the server to particular groups so that threads execute according to the resources available to the group.
- **Table encryption management** – Table encryption can now be managed globally by defining and enforcing encryption defaults.
- **InnoDB enhancements** – Several InnoDB enhancements were added, like, auto-increment counter value, index tree corruption, mem-cached plug-in, InnoDB_deadlock_detect, tablespace encryption feature, storage engine, InnoDB_dedicated_server, creating temporary tables in temporary tablespace, zlib library etc.
- **Character set support** – The default character set has changed from latin1 to utf8mb4. The utf8mb4 character set has several new collations, including utf8mb4_ja_0900_as_cs, the first Japanese language-specific collation available for Unicode in MySQL.
- **JSON enhancements** – Several enhancements and additions were made to MySQL's JSON functionality.
- **Data type support** – MySQL now supports the use of expressions as default values in data type specifications. This includes the use of expressions as default values for the BLOB, TEXT, GEOMETRY, and JSON data types, which previously could not be assigned default values at all.
- **Optimizer** – Optimizer is enhanced in various ways as well.
- **Improved hash join performance** – MySQL 8.0.23 reimplements the hash table used for hash joins, resulting in several improvements in hash join performance.
- **Common table expressions** – MySQL now supports common table expressions, both non-recursive and recursive.
- **Window functions** – MySQL now supports window functions that, for each row from a query, perform a calculation using rows related to that row.
- **Lateral derived tables** – A derived table now may be preceded by the LATERAL keyword to specify that it is permitted to refer to (depend on) columns of preceding tables in the same FROM clause.
- **Aliases in single-table DELETE statements** – In MySQL 8.0.16 and later, single-table DELETE statements support the use of table aliases.
- **Regular expression support** – Previously, MySQL used the Henry Spencer regular expression library to support regular expression operators.

- **Internal temporary tables** – The TempTable storage engine replaces the MEMORY storage engine as the default engine for in-memory internal temporary tables.
- **Logging** – Logging process is also improved in the newer versions.

In addition to all these, there are many other features added to the new versions of MySQL as well.

Features Deprecated in MySQL 8.0

The following are some of the features deprecated in MySQL 8.0 and may be removed in the future series. Several alternatives mentioned can be used by applications.

- The utf8mb3 character set is deprecated. Please use utf8mb4 instead.
- ucs2, macroman and macce, dec, hp8 are also deprecated. You should use utf8mb4 instead.
- User-defined collations are deprecated.
- sha256_password is deprecated.
- The plugin form of validate_password is still available, but deprecated.
- The ENGINE clause for the ALTER TABLESPACE and DROP TABLESPACE statements is deprecated.
- The PAD_CHAR_TO_FULL_LENGTH SQL mode is deprecated.
- AUTO_INCREMENT support is deprecated for columns of type FLOAT and DOUBLE (and any synonyms).
- The UNSIGNED attribute is deprecated for columns of type FLOAT, DOUBLE, and DECIMAL (and any synonyms)
- FLOAT(M,D) and DOUBLE(M,D) syntax is deprecated.
- The ZEROFILL attribute for numeric data types and the display width attribute for integer data types are deprecated.
- The BINARY attribute is deprecated. However, the use of BINARY to specify a data type or character set remains unchanged.
- ASCII and UNICODE are deprecated (MySQL 8.0.28 and later). Use CHARACTER SET instead, in both cases.

4. MySQL - Variables

In general, variables are the containers that store some information in a program. The value of a variable can be changed as many times as required. Each variable has a datatype specifying the type of data we can store in it such as integer, string, float etc.

- In some programming languages such as Java, C, C++ etc., we need to declare the datatype of a variable before assigning values to it.
- In languages like python, the datatype of a variable is presumed based on the values assigned to it. There is no need of declaring the datatype separately.
- In MySQL, there is no need to declare the datatype. We can simply define a variable with a value using the SET statement.

Variables in MySQL

The main purpose of a variable is to label a memory location(s) and store data in it so that it can be used throughout the program.

The characters we use to declare and define a variables are called literals and a literal can be anything other than special characters, numbers and, reserved keywords.

In MySQL, there are three types of variables. The same is described below –

- User-Defined Variable
- Local Variable
- System Variable

User-Defined Variables

The User-Defined variable allows us to store a value in one statement and subsequently refer to it in another. To do so, MySQL provides SET and SELECT commands to declare a variable. These variable names will have the symbol "@" as a prefix. We can use either = or := symbols depending on the situation. The user-defined data type can be any of the following: integer, decimal, Boolean, etc.

Syntax

Following is the syntax to declare a user-defined variable in MySQL using the SET statement –

```
SELECT @variable_name = value
```

Example

In the following query, we are assigning a value to a variable using the SET statement as follows –

```
SET @Name = 'Michael';
```

Using the SELECT statement, we can display the value of @name variable –

```
SELECT @Name;
```

Output

The output for the query above is produced as given below –

@Name
Michael

Example

Here, we are assigning a value to a variable using the SELECT statement –

```
SELECT @test := 10;
```

Output

On executing the given query, the output is displayed as follows –

@test := 10
10

Example

Let us create a table with the name CUSTOMERS using the following query –

```
CREATE TABLE CUSTOMERS(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    NAME VARCHAR(20) NOT NULL,
    AGE INT NOT NULL,
    ADDRESS CHAR (25),
    SALARY DECIMAL (18, 2)
);
```

Now, let us insert values into the above-created table using the INSERT INTO statement –

```
INSERT INTO CUSTOMERS (NAME, AGE, ADDRESS, SALARY) VALUES
('Ramesh', 32, 'Ahmedabad', 2000.00),
('Khilan', 25, 'Delhi', 1500.00),
('Kaushik', 23, 'Kota', 2000.00),
```

```
( 'Chaitali', 25, 'Mumbai', 6500.00),
( 'Hardik', 27, 'Bhopal', 8500.00),
( 'Komal', 22, 'Hyderabad', 4500.00),
( 'Muffy', 24, 'Indore', 10000.00);
```

The CUSTOMERS table is created as follows –

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	Hyderabad	4500.00
7	Muffy	24	Indore	10000.00

Now, let us declare a variable with the name @max_salary using the SELECT statement to display the maximum salary value from the CUSTOMERS table –

```
SELECT @max_salary := MAX(salary) FROM CUSTOMERS;
```

Then, we will select records from the table where the salary is equal to **@max_salary** variable –

```
SELECT * FROM CUSTOMERS WHERE SALARY = @max_salary;
```

Output

The output for the query above is produced as given below –

ID	NAME	AGE	ADDRESS	SALARY
7	Muffy	24	Indore	10000.00

Local Variables

The MySQL local variable can be declared using the **DECLARE** keyword. When we declare the local variable, the @ symbol is not used as prefix. This variable is a strongly typed variable, which means that we definitely need to declare a data type.

The MySQL DEFAULT keyword can be used while declaring a variable to set the default value of the variable. This is an optional parameter, if we do not define this, the initial value will be **NULL**.

Syntax

Following is the syntax to declare a local variable in MySQL –

```
DECLARE variable_name1, variabale_name2, ...  
  
data_type [DEFAULT default_value];
```

Example

In the following example, we are using the DECLARE statement in a stored procedure.

```
DELIMITER //  
CREATE PROCEDURE salaries()  
BEGIN  
    DECLARE Ramesh INT;  
    DECLARE Khilan INT DEFAULT 30000;  
    DECLARE Kaushik INT;  
    DECLARE Chaitali INT;  
    DECLARE Total INT;  
    SET Ramesh = 20000;  
    SET Kaushik = 25000;  
    SET Chaitali = 29000;  
    SET Total = Ramesh+Khilan+Kaushik+Chaitali;  
    SELECT Total,Ramesh,Khilan,Kaushik,Chaitali;  
END //
```

Now, let us call the stored procedure using the following query –

```
CALL salaries() //;
```

Output

Following is the output –

Total	Ramesh	Khilan	Kaushik	Chaitali
104000	20000	30000	25000	29000

System Variables

The **system variables** are predefined by the MySQL. These contains the data we need, to work with the database. Each MySQL system variable has a default value.

The SET command in MySQL can be used at the runtime to dynamically change the values of the system variables.

There are two variable scope modifiers available for the SHOW VARIABLES command. They are GLOBAL and SESSION.

- The GLOBAL variables are active throughout the lifecycle.
- The SESSION variables can be available only in the current session.

Following is the command to display all the system variables in MySQL –

```
SHOW [GLOBAL | SESSION] VARIABLES;
```

Example

In the following example, let us display the existing global system variables using the SHOW VARIABLES query –

```
SHOW VARIABLES LIKE '%table%';
```

The variables are displayed in the table format as follows –

Variable_name	Value
big_tables	OFF
default_table_encryption	OFF
innodb_file_per_table	ON
innodb_ft_aux_table	
innodb_ft_server_stopword_table	
innodb_ft_user_stopword_table	

innodb_table_locks	ON
innodb_temp_tablespaces_dir	.\#innodb_temp\
innodb_undo_tablespaces	2
innodb_validate_tablespace_paths	ON
lower_case_table_names	1
max_heap_table_size	16777216
old_alter_table	OFF
performance_schema_max_table_handles	-1
performance_schema_max_table_instances	-1
performance_schema_max_table_lock_stat	-1
show_create_table_skip_secondary_engine	OFF
show_create_table_verbosity	OFF
table_definition_cache	2000
table_encryption_privilege_check	OFF
table_open_cache	4000
table_open_cache_instances	16
tablespace_definition_cache	256
temptable_max_mmap	1073741824

temptable_max_ram	1073741824
temptable_use_mmap	ON
tmp_table_size	99614720
updatable_views_with_limit	YES

Now, using the query below, we will fetch the current value of the MySQL "key_buffer_size" variable –

```
SELECT @@key_buffer_size;
```

Output

Following is the output of the above query –

@@key_buffer_size
8388608

=====

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://www.tutorialspoint.com/index.htm>