

MySQL

SQL (Structured Query Language) is used to manipulate data stored in relational databases. It is a universal language, thanks to which work on different database systems comes down to issuing the same or similar commands, so-called SQL queries.

SQL was developed in the 1970s at IBM.

In 1986, SQL became an official standard, supported by the International Organization for Standardization (ISO). The last version of the standard (eighth) dates from 2016.

The SQL language has been implemented in most relational database systems, such as:

IBM DB2, Oracle, Microsoft SQL Server, MySQL, MariaDB, SQLite, dBase, InterBase, PostgreSQL, Sybase, Informix, Firebird, MS Access

SQL syntax can be divided into three (or four with SELECT) parts:

- language for defining data structures - DDL (Data Definition Language) - is used for all kinds of operations on tables, such as: creation, modification and deletion,
 - CREATE - creating a structure (base, table, index, user, view, etc.),
 - ALTER - changing the structure (adding a column to the table, changing the data type, etc.)
 - DROP - structure removal
- language for selecting and manipulating data - DML (Data Manipulation Language) - is used to manipulate data placed in tables, allows you to insert data, present, modify and delete data,
 - INSERT – insertion of data
 - UPDATE– change of data
 - DELETE - data deletion
- a language to ensure the security of access to data - DCL (Data Control Language) - is used mainly by database system administrators to grant appropriate permissions to use the database.
 - GRANT - granting permissions to individual objects or globally to the user
 - REVOKE - revoke the indicated permissions
- language for formulating queries to the database - DQL (Data Query Language)
 - SELECT

```
SELECT *  
  FROM emp  
 WHERE salary > 2000  
 ORDER BY years DESC;
```

SQL Data Types

Each column in a database table is required to have a name and a data type.

An SQL developer must decide what type of data that will be stored inside each column when creating a table. The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.

Note: Data types might have different names in different database. And even if the name is the same, the size and other details may be different! **Always check the documentation!**

MySQL Data Types (Version 8.0)

In MySQL there are three main data types: string, numeric, and date and time.

String Data Types

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the maximum string length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large Objects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LOBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data
ENUM(val1, val2, val3, ...)	A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list

Numeric Data Types

Data type	Description
<code>BIT(size)</code>	A bit-value type. The number of bits per value is specified in <i>size</i> . The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1.
<code>TINYINT(size)</code>	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The <i>size</i> parameter specifies the maximum display width (which is 255)
<code>BOOL</code>	Zero is considered as false, nonzero values are considered as true.
<code>BOOLEAN</code>	Equal to <code>BOOL</code>
<code>SMALLINT(size)</code>	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The <i>size</i> parameter specifies the maximum display width (which is 255)
<code>MEDIUMINT(size)</code>	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The <i>size</i> parameter specifies the maximum display width (which is 255)
<code>INT(size)</code>	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The <i>size</i> parameter specifies the maximum display width (which is 255)
<code>INTEGER(size)</code>	Equal to <code>INT(size)</code>
<code>BIGINT(size)</code>	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The <i>size</i> parameter specifies the maximum display width (which is 255)
<code>FLOAT(size, d)</code>	A floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions
<code>FLOAT(p)</code>	A floating point number. MySQL uses the <i>p</i> value to determine whether to use <code>FLOAT</code> or <code>DOUBLE</code> for the resulting data type. If <i>p</i> is from 0 to 24, the data type becomes <code>FLOAT()</code> . If <i>p</i> is from 25 to 53, the data type becomes <code>DOUBLE()</code>
<code>DOUBLE(size, d)</code>	A normal-size floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter
<code>DOUBLE PRECISION(size, d)</code>	
<code>DECIMAL(size, d)</code>	An exact fixed-point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. The maximum number for <i>size</i> is 65. The maximum number for <i>d</i> is 30. The default value for <i>size</i> is 10. The default value for <i>d</i> is 0.
<code>DEC(size, d)</code>	Equal to <code>DECIMAL(size,d)</code>

Note: All the numeric data types may have an extra option: UNSIGNED or ZEROFILL. If you add the UNSIGNED option, MySQL disallows negative values for the column. If you add the ZEROFILL option, MySQL automatically also adds the UNSIGNED attribute to the column.

Date and Time Data Types

Data type	Description
DATE	A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'
DATETIME(<i>fsp</i>)	A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(<i>fsp</i>)	A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition
TIME(<i>fsp</i>)	A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'
YEAR	A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.