Summary – Day 2

MySQL supports several standard SQL data types. Each column can contain only one data type. In MySQL, data types are grouped in different categories:

- Numeric
- Date and time
- String
- Spatial
- JSON

MySQL String Data Types (Text Formats):

In MySQL, string data types usually store data as long strings of text, for example, feedback or product description. Strings can consist of letters, numbers, or binary media files such as images, photos, or audio files.

The MySQL string data types are divided into:

- TEXT
- BLOB
- CHAR and VARCHAR
- BINARY and VARBINARY
- ENUM
- SET

TEXT:

TEXT data type is used to store long-text strings to display information about the table object, such as product descriptions, blog comments, etc. The storage size of the TEXT data type varies from 1 byte to 4 GB. Unlike the numeric data types, you do not have to set a length next to the TEXT data type in the table column. Moreover, TEXT values are not stored in the server's memory but use the disk instead. Therefore, the TEXT data types require +1 additional byte to retrieve data.

Type	Storage	Maximum number of characters	Usage
TINYTEXT	255	255	To store short-text strings such as links, product description or summary
TEXT	64 kB	65535	To store texts such as articles that do not exceed the specified length of characters
MEDIUMTEXT	16 MB	16777215	To store large texts such as whitepapers or books
LONGTEXT	4 GB	4294967295	To store huge texts such as computer programs or applications

BLOB:

Unlike the TEXT data types, which are non-binary string data types, the BLOB data types are binary strings. In MySQL, the BLOB data type represents a binary large object and can be used to store binary media data, such as audio or video links, images, or files.

The BLOB data types, including TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB, have a variable length, i.e. additional bytes (from 1 to 4) are required to store a value length in the column. For example, the maximum size of data to be stored is as follows:

- TINYBLOB => 255 bytes + 1 byte
- BLOB => 65535 + 2 bytes
- MEDIUMBLOB => 16777215 + 3 bytes
- LONGBLOB => 4294967295 + 4 bytes

When comparing BLOB and TEXT data types, it should be noted that BLOB is defined as numeric values, while TEXT – as character strings having a character set. This should be taken into account when comparing and sorting information.

CHAR and VARCHAR:

- In MySQL, the CHAR data types store non-binary strings with a fixed length that reaches 255 characters, while the VARCHAR data types store non-binary strings with a variable length having a maximum size of up to 65535 characters.
- For both data types, you need to set a size parameter in characters (in brackets) when creating a column. The size parameter represents the column length for a CHAR data type and the maximum column length for a VARCHAR data type. For example, CHAR(3) refers to up to 3 characters for the column value.
- The main distinction between the CHAR and VARCHAR data types is a way of storing data. CHAR adds spaces to values on the right to the specified length, for example,

CHAR(3) will be displayed as follows 'table '. VARCHAR outputs the value as it is, without any additional spaces – VARCHAR(3) will be displayed as 'table'.

- It should be noted that when defining a datatype for a phone number in MySQL, VARCHAR is more preferable to integers as sometimes there may be special symbols or characters. In addition, VARCHAR simplifies validation.

BINARY and VARBINARY:

- Though CHAR and VARCHAR seem to be similar to BINARY and VARBINARY data types, they have some differences. BINARY and VARBINARY store binary strings, and length is measured in bytes.

ENUM:

- The MySQL ENUM data types are strings with enumeration values. ENUM allows you to set a list of predefined values and then choose any of them. If you add an invalid value that is not included in the list, you will get an empty string.
- For example, we want to create a table that will store information about the size of women's clothes: small, medium, and large. In the table, we will insert the size column with the ENUM type. It means that this column will take only specified values.

MySQL Numeric Data Types (Number Formats):

MySQL supports numeric data types such as integers, decimals, and floating-point data types:

- Integers represent numbers without fractions and can have SIGNED and UNSIGNED attributes. Usually, they may be used for IDs or counting numbers.
- Decimals represent numbers with fractions and store exact numeric values in the column. They can be signed and unsigned and are usually used for columns that store monetary values. In the comparison with the floating-point numbers, decimals are more accurate.
- Floating-point represent numbers with fractions but do not store exact numeric values. They can be signed and unsigned. Floating-point numeric values use a double-precision 64-bit format or a single-precision 32-bit format to store data. They may lead to a loss of precision during arithmetic operations.

Integer:

Integer data types are numeric values without fractions. MySQL supports the following integer data types:

- TINYINT
- SMALLINT

- INT
- MEDIUMINT
- BIGINT

They can be UNSIGNED, which allow only zero and positive numbers in a column, and SIGNED, which store zero, positive, and negative numbers.

Boolean:

- The boolean data types can only accept either true or false values. In a binary format, true refers to 1 and false – to 0. As a rule, they are used for logical operations.
- MySQL does not have a boolean (or bool) data type. Instead, it converts boolean values into integer data types (TINYINT). When you create a table with a boolean data type, MySQL outputs data as 0, if false, and 1, if true.

Float:

The Float data types represent single-precision approximate numeric values that require 4 bytes for storage.

Double:

The Double data types refer to the floating-point numeric data types and use 8 bytes to store double-precision values. The syntax for the double data type is DOUBLE PRECISION(m,d) where 'm' is the total number of digits and 'd' is the number of digits following the decimal point. For example, DOUBLE(7,5) means that it will store a value with seven digits and five decimals.

Decimal:

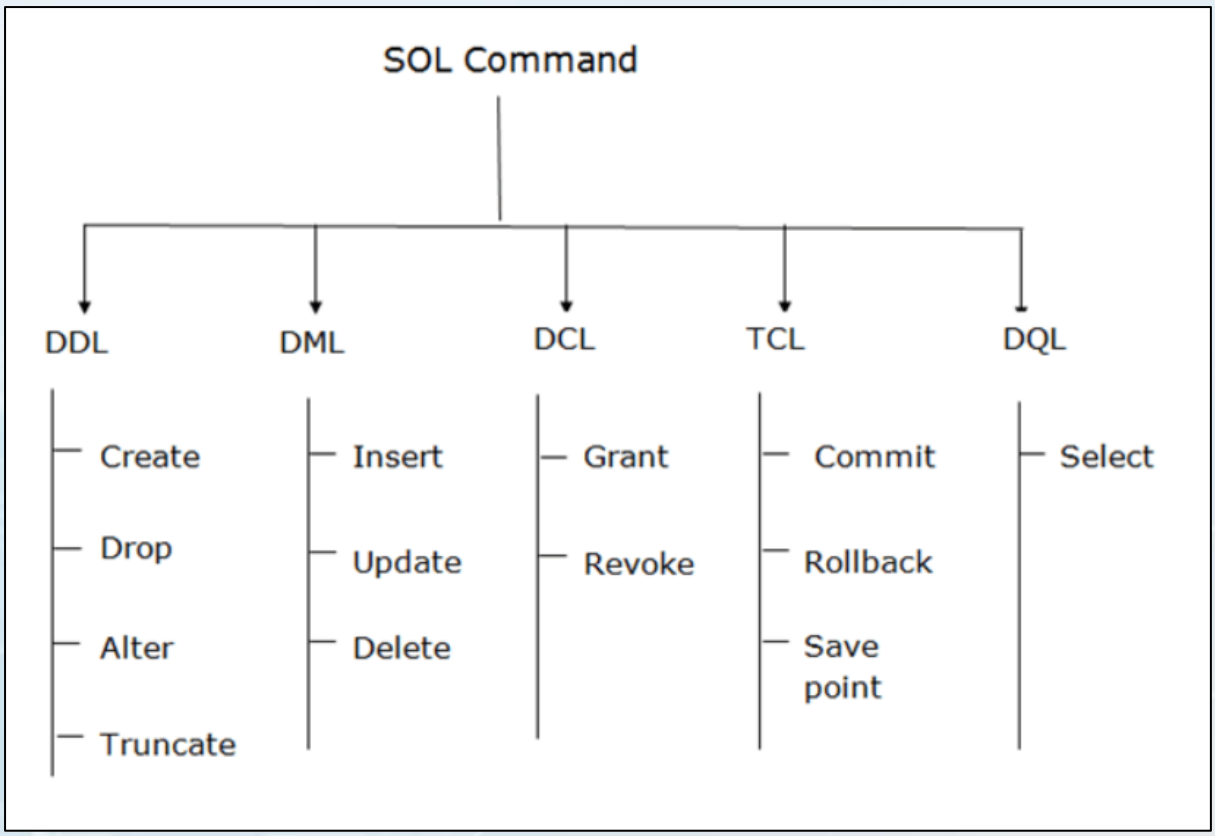
- The DECIMAL data type can be used to store exact and fixed numeric values. When creating a table column, the syntax for the data type is DECIMAL(p,s) where 'p' stands for precision, the maximum number of digits, and 's' stands for scale, the number of digits following the decimal.
- As a result, the main difference between float and double data types is precision (from 0 to 23 for FLOAT, and from 24 to 53 for DOUBLE) and accuracy (up to approximately 7 decimals for FLOAT, and up to approximately 15 decimals for DOUBLE).
- To sum up, decimals are better to use for fixed amounts, such as monetary and financial information (price, salary, etc,), while float and double – for approximate calculations where rounding values might have a negative impact.

MySQL Date & Time Data Types:

Type	Usage	Data type format	Range
DATE	Stores only date information in the table column	YYYY-MM-DD format (year, month, and date)	from '1000-01-01' to '9999-12-31'
TIME	Displays only time	HH:MM:SS format (hours, minutes, and seconds)	from '-838:59:59' to '838:59:59'
DATE TIME	Stores both date and time in the column	YYYY-MM-DD HH:MM:SS (year, month, and date, and hours, minutes, and seconds)	from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
DATE TIME	Stores both date and time values in the column	YYYY-MM-DD HH:MM:SS (year, month, and date, and hours, minutes, and seconds)	from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
TIME STA MP	Stores both date and time values in the column. Conversion of the value from the zone of the connection server to UTC takes place.	YYYY-MM-DD HH:MM:SS (year, month, and date, and hours, minutes, and seconds)	from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC
YEAR	Stores only year values in the column	YYYY (year)	from '1901' to '2155'

MySQL Commands:

- DDL – Data Definition Language
- DQL – Data Query Language
- DML – Data Manipulation Language
- DCL – Data Control Language
- TCL – Transaction Control Language


SOL Command
DDL
DML
DCL
TCL
DQL
Create
Drop
Alter
Truncate
Insert
Update
Delete
Grant
Revoke
Commit
Rollback
Save point
Select