SQL DATATYPES

CLASS OUTLINE

- What are Datatypes?
- Type of Datatypes
  - Numeric
  - Date and Time
  - Character/ String
  - Miscellaneous

DATATYPES

- A data type is a **classification of data** that determines
  - the kind of **operations** that can be performed on it,
  - the type of **values that can be stored** in it,
  - and the **size of memory** required to store it.
- There are various data types in programming languages, **including integer, floating-point, character, Boolean, string etc**. Each data type has its own set of rules and operations that can be performed on it.

DATATYPES

- Properly defining the fields in a table is important to the **overall optimization of your database**.
- For better optimization, one should **use only the type and size of field** you really need to use. For example, don't define a field as 10 characters wide if you know you're only going to use 2 characters.
- MySQL uses many different data types broken into the following categories:
  - Numeric
  - String
  - Date and Time
  - Miscellaneous

DATATYPES

- **Numeric data types:**
  - INT (integer)
  - BIGINT (large integer)
  - FLOAT (floating-point number)
  - DOUBLE (double-precision floating-point number)
  - DECIMAL (decimal number)
- **String (Character) data types:**
  - CHAR (fixed-length character string)
  - VARCHAR (variable-length character string)
  - TEXT (long character string)

DATATYPES

- **Date and time data types:**
  - DATE (date)
  - TIME (time)
  - DATETIME (date and time)
  - TIMESTAMP (timestamp)
- **Other data types:**
  - BINARY String types
  - VARBINARY (variable-length binary data)
  - BLOB (binary large object)
  - BOOLEAN or BOOL (Boolean value)

These are some of the commonly used data types in MySQL, and there are other data types as well, such as **ENUM**, **SET**, and **JSON**.

NUMERIC DATA TYPES:

MySQL uses all the standard ANSI SQL numeric data types.
The following list shows the **common numeric data types** and their descriptions:

- Integer range for **SIGNED** numbers(both positive and negative numbers can be stored).

Data Type	Storage Size	Minimum Value	Maximum Value
TINYINT	1 byte	-128	127
SMALLINT	2 bytes	-32768	32767
MEDIUMINT	3 bytes	-8,388,608	8,388,607
INT	4 bytes	-2,147,483,648	2,147,483,647
BIGINT	8 bytes	-9,223,372,036,854,770,000	9,223,372,036,854,770,000

NUMERIC DATA TYPES:

- Integer range for **UNSIGNED numbers** (only positive integers(including zero) can be stored).

Data Type	Storage Size	Minimum Value	Maximum Value
TINYINT	1 byte	0	255
SMALLINT	2 bytes	0	65,535
MEDIUMINT	3 bytes	0	16,777,215
INT	4 bytes	0	4,294,967,295
BIGINT	8 bytes	0	18,446,744,073,709,500,000

NUMERIC DATA TYPES:

Floating (Approximate Values) Point Types

They can not be defined as UNSIGNED(meaning can hold both positive and negative values).

- **FLOAT(*M, D*)** - FLOAT is a single-precision floating-point data type. The FLOAT data type in MySQL has a precision of 24 bits(4 bytes) or approximately 7 decimal digits. This means that it can store approximate numeric values with up to 7 significant digits, both before and after the decimal point.
- **DOUBLE** OR **REAL(*M, D*)** - A double-precision floating-point number that cannot be unsigned. A precision from 24 to 53 results in an 8-byte double-precision DOUBLE column.

Note: Here **(*M, D*),** means that values can be stored with up to M digits in total, of which D digits may be after the decimal point.

NUMERIC DATA TYPES:

Fixed (Exact Values) Point Types

They can not be defined as UNSIGNED(meaning can hold both positive and negative values).

- **DECIMAL** OR **NUMERIC(*M, D*)** - DECIMAL is a fixed-point decimal data type that **allows for exact numeric values.** It can store numeric values with a specified precision(M) and scale(D), up to a maximum of 65 digits of precision and 30 digits of scale. The storage requirement for DECIMAL varies based on the precision and scale specified.

Note: It's important to choose the appropriate data type based on the requirements of your application. If you need exact numeric values, then DECIMAL is the way to go. If you need to store approximate numeric values and need higher precision, then DOUBLE is better than FLOAT. However, if storage space is a concern, then FLOAT might be a better choice.

DATE AND TIME DATA TYPES:

- **DATE** - A date in **`YYYY-MM-DD`** format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.
- **DATETIME** - A date and time combination in **`YYYY-MM-DD HH:MM:SS`** format, between 1000- 01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.
- **TIMESTAMP** - A timestamp between midnight, January 1, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 **`YYYYMMDDHHMMSS`**.
- **TIME** - Stores the time in **`HH:MM:SS`** format.
- **YEAR** - Stores a year value in the format **`YYYY`.**

STRING & OTHER DATA TYPES:

MySQL provides several data types **for storing string values**:

1. **CHAR**(*M)***:** Stores a **fixed-length string of characters** up to a maximum length of 255 characters.
   For example, **`Mumbai`** can be stored in **CHAR(6)** datatype as its character length is 6.
2. **VARCHAR**(*M)***:** Stores a **variable-length string of characters** up to a maximum length of 65,535 characters.
   For example, **`Mumbai`** can be stored in any VARCHAR(M) datatype where the value of M is greater than or equal to 6 like **VARCHAR(6)** or **VARCHAR(255).**
3. **TEXT:** This data type is used **to store large character strings**, such as **paragraphs** or **documents**, that exceed the maximum length of a VARCHAR column. When you declare a TEXT column, you do not need to specify its length, as MySQL will allocate only the necessary storage for each value, up to a maximum of 65,535 bytes.

The four TEXT types are **TINYTEXT**, **TEXT**, **MEDIUMTEXT**, and **LONGTEXT**. These differ only in the maximum length of the values they can hold.

STRING & OTHER DATA TYPES:

Note:

- In case you provide a string of less character length MySQL will pad the string with **spaces** to make it the maximum length.
- It's important to note that because **CHAR columns** always use the maximum amount of storage, they can **use up more disk space** than **VARCHAR columns**, which are variable-length.

Other strings data types:

ENUM and SET are two specific data types in MySQL that are **used to store lists of values**

1. **ENUM:** ENUM is a data type that allows you to create a list of values that a column can hold. For example, you could create an ENUM column for a "gender" field that only allows the values "male" and "female".
2. **SET:** This data type is **used to store a list of predefined values**, each represented by a string, where **multiple values can be selected at once.** SET columns are often used to store data with multiple options, such as hobbies or interests.

STRING & OTHER DATA TYPES:

Binary strings are sequences of binary digits (bits)**.**

Datatypes for storing Binary strings values:

1. **BINARY:** Stores a fixed-length binary string up to a maximum length of 255 bytes.
2. **VARBINARY:** Stores a variable-length binary string up to a maximum length of 65,535 bytes.
3. **BLOB:** A BLOB is a binary large object that can hold a variable amount of data. BLOB columns are similar to TEXT columns, but they store binary data instead of character data. It is generally used to store data **such as images, audio files, or program code**, that cannot be represented as text.

The four BLOB types are **TINYBLOB**, **BLOB**, **MEDIUMBLOB**, and **LONGBLOB**. These differ only in the maximum length of the values they can hold.

STRING & OTHER DATA TYPES:

Other data types of MySQL:

1. **BOOL** or **BOOLEAN:** In MySQL, the BOOL data type is a synonym for the **TINYINT(1)** data type, which stores a single boolean value that can be either true or false. The values that can be stored in a TINYINT(1) column are 0 and 1, where **0 represents false** and **1 represents true.**
2. **JSON:** In MySQL, the JSON data type is a way to store and manipulate JSON (JavaScript Object Notation) data in a database. JSON is a lightweight data interchange format that is **widely used for web applications and APIs.**
   You can use various MySQL functions to manipulate and query JSON data, such as JSON_OBJECT(), JSON_ARRAY(), and JSON_EXTRACT().

THANK YOU