**\*\*Answers 1 and 2 were derived from information found on W3Schools.com.**

**Question: What are ten different data types MySQL provides?**

**Answer:** W3Schools showcases three main branches of MySQL data types: string data types, numerical data types, and data and time data types. There are numerous data types under each of these main types.

STRING Data types include: **char, text, enum, and set**.

NMERIC Data types include: **bit, boolean, integer, float, decimal**

DATE/TIME Data Types include: **timestamp**, etc.

**Question: How is each data type you described used, and what makes it unique?**

**Answer:**

**Char:** Tells the computer that anything that comes after this is a string of fixed length. When char is typed like this: char(3), this means that the string is three characters long. Strings with the “char” designation can have a length from 0 to 255 characters. (The default for this type of data is 1.)

**BINARY** **is similar to char** in that its default is 1 and it stores string data too. **The difference between BINARY and Char** is that **BINARY** stores binary byte strings and the size (given by the number following **BINARY** (like this: **BINARY(3)**) determines the column length in bytes.

**TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB** also hold binary data (objects). **TINYBLOB** holds up to 255 bytes of data, whereas **BLOB** holds 65,535 bytes of data, whereas **MEDIUMBLOB** holds up to 16,777,215. **LONGBLOB** holds up to 4.294,967,295 bytes of data.

**Varchar differs from char because** **varchar** tells the computer that the following string is variable in length and can have a length as much as 65,535 characters, just like **BLOB** and **TEXT**.

**Text:** (from dev.mysql.com ) **TEXT** is a character string. **TEXT** is similar to **BLOB** data in that the amount of data they store is the same. Consider: **TINYTEXT** holds up to 255 characters. **TEXT** holds up to 65.535, **MEDIUMTEXT** holds 16,777,215, and **LONGTEXT** holds up to 4.294,967,295 characters.

**Enum:** Like a drop down with numbers, this string object, can have only 1 value. The ENUM list can have up to 65,535 values in the list, but if a value is chosen that isn’t on the list, blank will be the return result. ENUM(value1, value 2, etc.) is how this is typed in.

**Set:** A set is a string object that consists of up to 64 values chosen from a list of values. For example, if I had a number range from 0-100 and chose the odds, the odds would be my set. (SET is formatted like ENUM.)

**Bit:** (from [www.javapoint.com](http://www.javapoint.com) ) Allows users to store bit values. A bit will only store a 0 or a 1, but can store up to 64.

**Boolean: (also BOOL)** Like in Java, Boolean stores true or false. In MySQL, nonzero values are true, and zero is false.

**Integer: (also INT)** Stores up to 255 integer values from-2,147,483,648 to 2147483647 (signed) and from 0 to 4294967295 (unsigned). Format is **INT(size).**

**SMALLINT** can only hold up to 255 integers. The ranges for integers it can hold are: -32768 to 32767 (signed) and 0 to 65535 (unsigned).

**MEDIUMINT** can hold up to 255 integers. The ranges for integers it can hold are: -8388608 to 8388607 (signed) and 0 to 16777215 (unsigned).

**BIGINT** can hold up to 255 integers. The ranges for integers it can hold are: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (signed) and 0 to 18,446,744,073,709,551,615.

**Float(p):** When (p) is 0-24, MySQL chooses **FLOAT** for the data type. If (p) is 25-53, the chosen data type is **DOUBLE**. When the data type is FLOAT, that means the number is a rational number (able to be divided by another number).

**decimal: (also DEC)** This sets the number of digits after the decimal point. So in a ‘Dollars’ column notation would look like this: DEC(4, 2). What this notation means is that there are 4 digits in the number and 2 are behind the decimal. With this data type, up to 65 numbers are allowed total with a max of 30 behind the decimal. The default size is 10 and the default value of numbers behind the decimal (d) is 0.