Data Types

In this lesson we will be looking at how computers and devices store your data and what that means for storage requirements and using resources efficiently while storing data. We will also look at the types of data (or data types) that can be used for all the data that we will be storing in our database.

When it comes to the computers storing your data, it's a good idea to understand how that works.

At the most basic level inside your computer the CPU or Central Processing Unit makes all the important calculations while your computer is running. The only basic unit of communication the CPU understands is 1s or 0s. There are billions of transistors, little gates in your CPU that are either open (a 1) or closed (a 0) and this is how all the data going through the CPU is recognized. This system of ones and zeros is called binary. Using binary there are just the 2 digits either 1 or 0 and each 1 or 0 is called a bit.

8 bits together form a byte. Every character and symbol of data is made up of 8 bits or 1 byte. Here are the bytes--combinations of 1s and 0s that make up the letters in the name Anderson.

ASCII is a character encoding based on the English language. It has 255 characters that correspond to different numbering system.

Each of the characters also corresponds to a binary byte. Control characters like tab, line space, carriage return; then there's printable characters special symbols, upper and lower case letters. Then past 126 to 255 there are extended character like mathematical symbols, other some language symbols, trade symbols etc.

Because the ASCII character set was not enough to cover all the languages, other character sets were developed.

Unicode also known as UTF-8 added onto ASCII so all international languages could be included including Asian language characters. So now some characters in Unicode have to be made up of more than one byte per character. The first 255 are the same as ASCII and only take 1 byte but additional characters in the Unicode set can take up to 3 bytes each. So Unicode is now the standard that you should build your databases in because it can transfer to any language easily.

Latin1 was the old standard that just included the English based languages.

When we enter data into our database, we have to decide what type of data it is and make sure it fits in the categories of data that we will cover here. There are three main types of data.

The first is Characters (also referred to as text or strings) This could be names, titles, even text like social security numbers, phone numbers or zip codes. Any text that won't really need to have math done on them would fit here.

The next type of data is Numbers. This could be decimals or integers or whole numbers or very precise numbers. Anything that could potentially need to be used in math should be stored as a number.

The third data types fit in the category of Dates and Times. Dates are always entered into the database as yyyy dash mm dash dd and times as hour colon mintes colon seconds.

Lets take a closer look at characters. One way to save your text or string data is using the CHAR or character data type. You use CHAR when you know the length in characters of your data. For example we know the length of zip codes, social security numbers and phone numbers so we could use CHAR with any of those. If we had a column that was for the two letter state code we could use CHAR(2) to show that we know data in this column will always be 2 characters long. With CHAR 3 bytes of data are used for each character so in our 2 character state code we would be using 6 bytes total.

Another data type for characters is VARCHAR or variable characters. You would use this data type when you don't know how long a string of characters might be ahead of time. We won't know the length of each name might be entered or titles or addresses; things like that.

If we had a first name column that we were defining we could use VARCHAR with 20 in parenthesis to say that we aren't sure how long the first name might be but we will set aside 20 characters for it just in case. With the variable character data type 1 byte of storage is used for each character plus one byte to keep track of the length. So if our name was Jane we would only use 4 of the 20 possible bytes plus one byte to keep track of the length so Jane would use 5 bytes total even though there were 20 bytes possible. I don't sue CHAR very often even if I know the exact length of characters if I know that VARCHAR will use less storage, I will use it.

Remember with CHAR and VARCHAR you use single quotes surrounding the text when entering the data in a statement.

The next type of data are numbers.

Integers or whole numbers are numbers without decimals. For example, primary keys are commonly integers. There are also types of integers that go by the names BIGINT, MEDIUMINT, SMALLINT, and TINYINT. And as their names suggest they also store integers but the range of how big or how small the numbers can be differs. Usually INT works well with most data. TINYINT does have a good use though for

BOOLEAN type data. BOOLEAN is where you can use have just two possible values, true or false. So TINYINT(1) and BOOLEAN work the same way. A value of 0 would be false and a value of 1 would be true. INT integers take 4 bytes of storage and TINYINT takes 1 byte.

The other type of number is a real number or numbers that can have decimals.

When you use the DECIMAL data type you can store fixed-point numbers, which are numbers that have a fixed number of digits to the right of the decimal point.

If we had DECIMAL(9, 2) we are saying that we can have 9 digits total on the left and right of the decimal point and 2 digits after the decimal point. The 9 digit total is called precision and the 2 digit after the decimal is called scale. So small number up to a number like 5,643,215.22 would fit within the nine total digits and two decimal places. but if it got high enough to go over those digits, say 60 billion you would have to use a data type with a higher precision. The amount of storage needed for these numbers vary according to how big the precision and scale are. Precision tops out at 65 (which is still a really huge number) and scale tops out at 30.

Two other data types fit within real numbers the Double and the float numbers. They both store data that are floating-point numbers (or no fixed number of digits before and after the decimal point). They can handle really large or really small numbers. Double take 8 bytes and fixed takes 4 bytes. These types of numbers are usually needed for very scientific calculations. We won't be using these data types in this course.

The last data types are DATE and TIME.

DATE is just the year month and day and takes 3 bytes. TIME is just the hour minutes and seconds and also takes 3 bytes. DATETIME has both the date and the time together. Remember with DATETIME, you can only enter dates from the years 1970 to 9999. TIMESTAMP will get a similar format to DATETIME but it will get it from the users system (or computer) that they are using. YEAR is the 4 digit year and ranges from 1901-2155.

As you enter these dates and times in statements you need to have single quotes around them. Because dates have dashes in them and the times have colons they are treated like strings, and you have to have the quotes around them as you enter them in the MySQL syntax.

A few other data types to be aware of are ENUM and SET. These are a good when you want to restrict the user to a set of string values.

ENUM stores values that are mutually exclusive. Meaning you can only choose one.

For example if you only wanted them to be able to enter the data yes, no or maybe then you could use those as the values when you set up the ENUM data type. Then the user is restricted to only those choices.

SET also restricts the number of options but the user can choose more than one value is they choose. A good example here would be the possible toppings on a pizza. They can and one to many but are restricted to what toppings are offered.

I like to think of them like radio buttons and checkboxes on input forms. The checkboxes are like SET and the radio buttons where you can only choose one at a time are like ENUM.

BLOB stores the binary output of images, audio and video. This can even include dpf or word files. These can take a lot of storage and sometimes it is easier to use them with other programming languages. We won't use this data type in this course but it is worth knowing what BLOB data is.

So there we go. The ways data is stored and some of the more common data types we can use to set up the way our data is stored.