1.3: Introduction to Data and Databases

Introduction

What is data?

To the chemist data are the measured or counted values that can be collected or produced to understand relationships of observable or computed phenomena that are germane to the practice of science (both empirical and computational). To the chemist there are different types of data that are defined by how the data is generated, like the mass or temperature of a sample, or the spectra of a compound. This data is often stored on a computer in a file or database, and can be subsequently processed through various software programs.

To the computer scientist or software program data has a different meaning in that there are different data types that represent how the computer stores information. That is, a computer does not store a measured phenomena like the temperature of a sample, but a digital data type, a representation of the temperature that a software agent can interact with. For example, a letter of the alphabet would be a different type of data than a number, because you can not do arithmetic calculations on letters like you do on numbers.

Cheminformaticians need to understand both meanings of the concept of data, and in this section we will introduce how computers store data, and the different types of data from the perspective of programming and software agents. Then we will move onto data in the chemistry sense of the word.

What is a database?

Databases are a way computers store information in a manner that can be retrieved. You use databases all the time. Do you realize that as you read this web page you are using a database? Yes, this web page is not a digital file like a MS Word document that saves the information like a sheet of paper, but instead the web browser is displaying information that was pulled from a database as the page is loaded. That is, LibreText is a Wiki that is hosted on the MindTouch knowledge management platform and the information you see is drawn from a database when the page is loaded. Webpages that are pulled from databases are often called dynamic web content, and those that are files are called static web content. Of course databases can store different types of information, and this class will be using databases that store information related to chemical compounds. But it is important to realize that the use of databases in the twentieth century are pervasive, and you are actually using a database right now, as you read this webpage.

How do databases store information?

Databases store data, which is the representation of information through a binary code that computing machines can read. A bit is the smallest binary value with two possibilities, 0 or 1. This data needs to be stored on a physical medium so the machine can read it. In the old days data was stored on punch cards (figure 1.3.1), which allowed for a binary representation of each position, which could be either punched or not punched (bitten or not bitten). If each location of memory is allowed a certain number of bits, then you can generate different combinations, and give those different combinations different meanings.

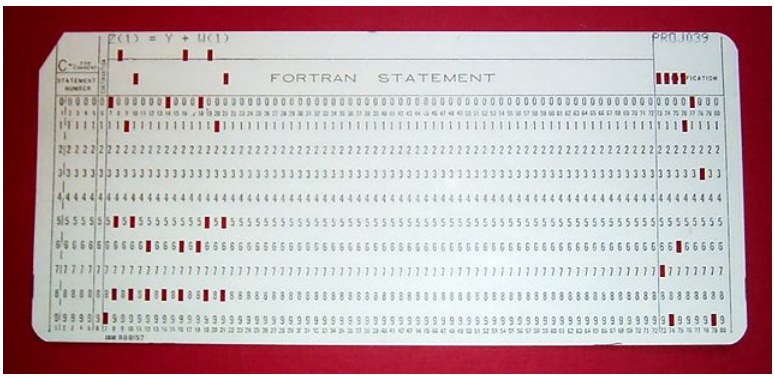

Figure 1.3.1: Old Fortran punch card, one of the earliest computer based means for storing data (image credit: By Arnold Reinhold - CC BY-SA 2.5, https://commons.wikimedia.org/w/inde...p?curid=775153)

A quick look at these possibilities shows that n bits gives 2^n possible combinations.

- 1 bit has two (2^1)possibilities : 0 or 1, and so can represent two different things
- 2 bits has four (2^2) possibilities: 00, 01, 10, or 11, and so can represent four different things

- 3 bits has 8 (2^3) possibilities: 001, 010, 100, 011, 101, 101, 110, 111, and so can represent 8 different things.
- 8 bits has 8 (2^8) possibilities, which is 256, ranging from 00000000 to 11111111, and we won't write them all down here.
- n bits has (2^n)

A byte of data is defined as 8 bits and so has (2^8) or 128 values (which run for 0 to 127). In the early days of computers 8 bit chip of memory was common and the American Standard Code for Information Interchange (ASCII) was developed and based on the 8 bit byte, which shows one set of code that allows computers to interact with a keyboard to store information. Note the first 32 ASCII characters are unprintable codes used to control devices, and the remaining 156 characters are used to store symbols like numbers and the letters of the alphabet. The full list can be found at https://www.ascii-code.com/ and here are some examples. It should be noted that there are other codes besides the ASCII codes.

binary byte	meaning		binary byte	meaning
00100001	!		00000000	null (not printable)
00100101	%		00000010	start of text
00110001	1		00000011	end of text
01000001	A		00001101	carriage return
01100001	a		11111111	ÿ

The take home message here is that everything is stored on the computer in the form of a binary bit, be it a text document, picture, molecular structure data or a spectral file. Each of these represent a different data type and so when you interact with the database, you need to know what type of data is stored, and then use software that can "read" that type of data. Likewise, if you write some simple script to interact with data, you need to recognize the data type you are interacting with, for example, you can do math with numbers, but not letters, and so a number needs to be a different data type than a letter.

Today we do not use punch cards but still store data as a binary representation on a physical device that can be electronically read, like magnetic tape, hard drives, flash drives, SSD (Solid State Disk) and the like. The way magnetic based storage devices work is through the North-South alignment of the magnetic field, where one of these (N-S) would be given the value of 1, and the other (S-N) would be the 0. If you are interested in learning how a hard drive works there is a real good 6 minute video on Nick Parlante's computer science page from Stanford. Flash drives and SSDs have no moving parts and are not based on magnetism, but represent ones and zeros by the ability of tiny channels (gates) within a transistor to be able to conduct (1) or not conduct (0) electricity. It should be noted that after 10-20 years flash drives can lose their memory. In fact, surprisingly magnetic tape is the longest lasting digital storage, although it is the slowest to use.

Data Types

There are multiple data types and when a programming language communicates with a computer it must specify the data type so that it can be properly handled. In fact different programming languages may handle data types differently. When you input data this is often done through a data field, and the data field will specify the type of data. We will look at a some of the very basics here. We are introducing these here because if you define a variable, you need to define what type of data it is. For example, a word is different than a number, and so if you define a variable, the software needs to know if it is a number or a letter. In this class your school will use either R or Python, which may handle data type definitions slightly differently, but in the end, they are doing the same things. We will now look at several different types of data. Note, in both R and Python you can change a variable data type after you first define it, while in many other programs you can not.

Numeric Types

These can be used in calculations. There are two basic types of numbers that we need to be aware of, integers and floating numbers. This is important because the way computers store these data types is different. If you think back to your general chemistry, there were exact numbers and measured numbers. Exact numbers were integers and measured numbers had precision, and used a decimal point.

Integers

Integers are exact numbers and do not have a decimal point. Integers are stored directly as binary values, but the first bit is used to indicate the sign of the number (plus or minus). So a 32 bit (4 bytes) chip could represent 2^{31} different positive or negative integers,

Floating point

Floating point numbers are stored like scientific notation, where part of the bit represents the mantissa, which is the number being multiplied by 10 to a power (and represents the precision), and another part of the bit represents the exponent.

Character Types

These are alphabetical characters. They are often called a string literal and when defined in code need to be placed into double quotations.

Other Types

There are sort of two other types of data. The first are data types used in programming, like time and Boolean logic data types. The second are files that are used to store data, and not used in programming.

Date & Time

There are a variety of date and time formats and these are actual data types.

Boolean (Logic)

Boolean data are data types that have two possible values, true or false. These can be used with Boolean logic operators like AND, OR and NOT, along with comparative operators, like equals, not equal, less than, less than or equal, greater than and greater than or equal. These are very important in programming because they allow computers to make logical decisions.

Specific file types

You can also store files in a database. These can be image files, pdf files, chemical structural data files, spectral files, etc.

Databases

In the old days of punch cards one would physically store the card deck in a filing cabinet and if you wanted to perform a calculation on the data you would physically have to retrieve the cards and load them into submit it to the computing machine.

Figure 1.3.2: March 21, 1957 image of people working on an IBM type 704 electronic data processing machine at Langley Research Center (Image credit: NASA)

Types of Databases

Flat-file Database

A flat file database is the simplest type of database and consists of a data table where the columns are fields and the rows are records. So in the following table, each row has the record of a chemical, which has the data fields that describe attributes of that chemical like its name, number of atoms, melting point, GHS pictogram and molecular formula. A flat file database is essentially of the same structure as a data table in a book like the CRC Handbook on Chemistry and Physics, except that you can search it like a webpage

It is important to note that each field is of a specific data type. The name and molecular formula are string literals (a string of characters), the number of atoms are integers, the melting points are floating point numbers and the pictograms are image files. When you create a field in a database you must identify the type of data stored in it.

Name	Number of atoms	melting point (^oC)	GHS pictogram	molecular formula
n-butane	14	-138.2		C_4H_{10}
Isobutane	14	-138.3		C_4H_{10}
benzene	12	5.5		C_6H_6

Table$PageIndex2$: The structure of a flat-file database. What is important to see here is that each record has different fields associated with it, and those fields need to identify the type of data they contain (you can not upload an image file to an integer field).

There are some shortcomings to the flat-file database, in that values may not be unique, that is, isomers like n-butane and isobutane would have the same number of atoms and molecular formulas, or that a name may have synonyms, and you may have searched that above file for 2-methypropane and missed it. Of course you could have a new record for each synonym, but that would be very inefficient.

Relational-Database

These are the most common types of databases used online. A relational database is like a table with an index number for each record, and you correlate the index number instead of the field value.

n	Name		na	Number of atoms		mp	melting point (^oC)		gp	GHS pictogram		mf	molecular formula
n1	n-butane		na1	14		mp1	-138.2		gp1			mf1	C_4H_{10}
n2	Isobutane		na2	14		mp2	-138.3		gp2			mf2	C_4H_{10}
n3	benzene		na3	12		mp3	5.5		gp3			mf3	C_6H_6

Table$PageIndex3$: Making 5 relational tables of the data in table 1.3.2, each with its own unique index number.

We now treat the flat-file data table as 5 different tables, each with a different index number (n,na,mp,gp & mf), and set up a relationship so the record of the first chemical is not defined by the value within the field, but by its index value. So the first

relationship is identified by the index values of n1, na1, mp1, gp1 & mf1, which relate to their respective field values (the record of row 1 in table 1.3.2). So for example, the number of atoms in n-butane and isobutane are defined by na1 and na2, not 14 and 14, ie., we are relating different things. Also, if we wanted to include all the synonyms, we only have to include their index value, and when we show the record, all of them show.
