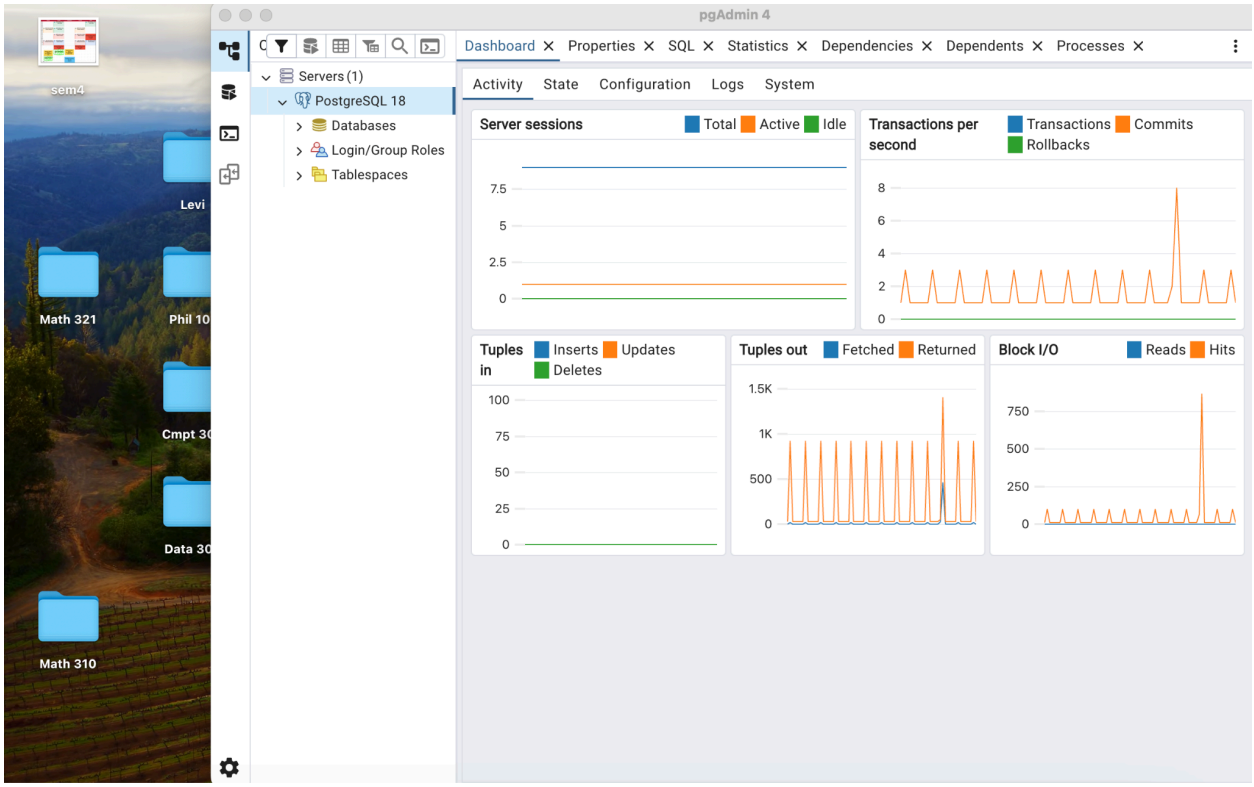
https://github.com/LeviDiamond999/CMPT-308/tree/main

1.



2. An element in a database is a single value. For example let's say you had a database of Olympic sprinters that looked like this (see table 1.1), some of the elements in this database would be (123, 207, 195, Male, 100m) which are highlighted in yellow. Another way to think about it is that there are 6 elements in the column “Name” highlighted in red. You can organize this database and turn it into information by sorting it by height that way you are informed of which sprinters are the tallest. Data is the collection of numbers or strings stored under a specific column. Data is useless if it is not connected to other pieces of data in the same row. For example the number 170 means nothing on its own until we see that it is connected to an ID in the same row and the column title weight (highlighted in green). Information is when you use the data to convey a message. If we wanted to see who the tallest sprinter in the 400m competition was we would filter through the Competition column and take the rows with the elements “400m” and then sort them by the elements in column Height.

This process will provide us the information on who is the tallest 400m sprinter. 1.1 Table of Olympic Sprinters (https://en.wikipedia.org/wiki/Main_Page)

Name	ID	Weight	Height	Sex	Competition
Usain Bolt	123	207	195	Male	100m
Noah Lyles	124	170	180	Male	200m
Sydney McLaughlin-Levrone	125	134	175	Female	400m
Marita Koch	126	137	171	Female	400m
Michael Johnson	127	175	185	Male	400m
Evelyn Ashford	128	117	165	Female	100m

3. Before relational DBMS there were single app data systems. These early systems would require you to feed hole-punched paper into the computer and only track one data type. An electric ballot for example was a machine that they used. These early systems lacked flexibility to track different types of data, you could only send results via mail stored on a disk and they were expensive. All of those reasons also made the data very hard to change because of all the resources and time it would take to re-do and make a new file. Relational models made things a lot easier because you could track different types of data, and ask questions without needing to rewrite an entire program.

> XML is great because it is flexible allowing you to store and structure data. It is human and machine readable.