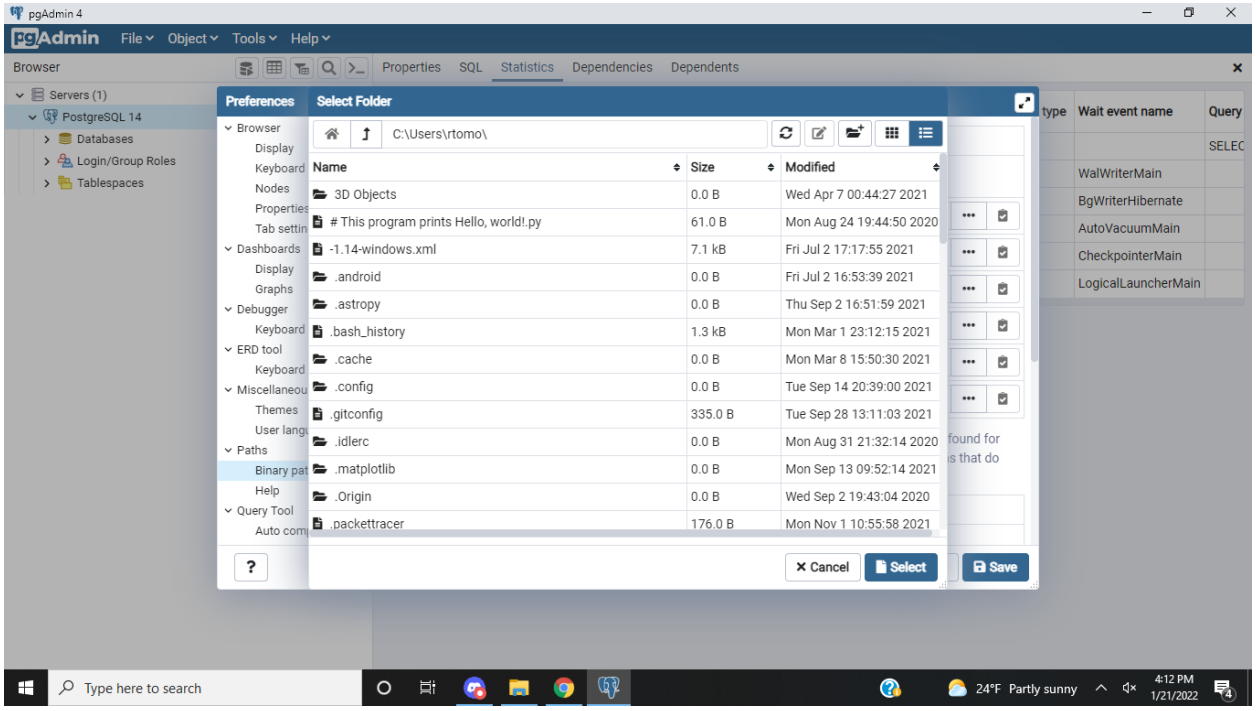
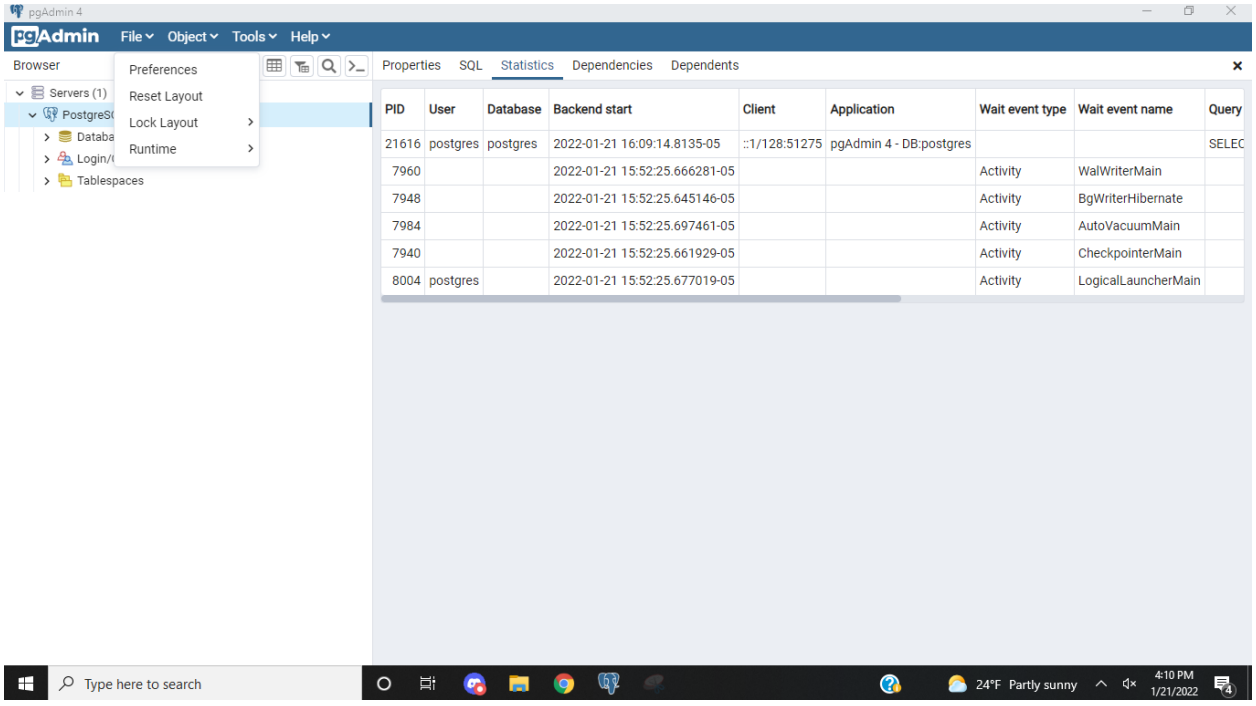
Raymond Tomo
01/21/22

Lab 1- Installing PostGre

Screenshots from the PostGre!





Short Essay #1: Data vs. Information

To help discuss data vs. information, a hypothetical scenario is an inventory system for a fast-food franchise. While there might be numbers of data such as 42, 3, or 100 floating around in the system, the tables containing columns and rows help give these numbers meaning. For instance, when placed into a table with the row being fries and the column being a number of shipments, that 42 now means that there are 42 shipments of fries waiting to be used. Likewise, that 3 could be placed next to the 42 in a subsequent square, under the column “Weeks until the next shipment of fries” and we now know as the owner of the business that the next shipment of fries is due in 3 weeks. These numbers have now become information, as the owner of the business can now make decisions due to having information, or data given the proper context.

Without this, the owner cannot decide if they need to order new fries or when the next shipment is due if they were to just have the numbers floating around in some nebulous space. For instance, if they needed to know the information about their fry inventory and when the next shipment is due, 42 or 3 will be meaningless to help them, as it is just data. This will lead them to make uninformed decisions based on instinct and not supported by actual data. Compare this with the situation where the data is given context, and that manager is able to deduce and make informed decisions about whether or not they will need to order more or fewer fries. Information will help someone make a decision, while data will help someone make a disaster. It is why these two terms may sound similar but are not the same.

Short Essay #2:

A hierarchical data model is almost like a tree. For the sake of demonstration, we can think of the records or bubbles as leaves and the links from one record to another as branches. Starting from one main network, each of the links linking one record to another imply a relationship, with the record closer to the top as being a superset to the subsets lying beneath them. Any records on the same layer of this tree model can be considered within the same layer of organization. However, while this system is great for establishing smaller relationships, it is inferior to a relational data model as you cannot link between records on the same level, and some records may be nebulous or require multiple connections to demonstrate two different types of relationships.It also doesn't have the same flexibility as a relational data model.

A network data model is an improvement to the relational data model, as it allows for objects on the same layer to be linked to each other. Similar to the hierarchical data model, when displayed it will look like a strict tree This means there will be less redundancy as a result, instead of requiring a massive amount of subsets in order to display all of the nuances between relationships. In this system, the lack of clutter also makes it easier to piece together multiple relationships on the same level. However, overall, it is still inferior to the relational model, as some structures or databases will not have the same simplified tree hierarchy displayed here, and may be too complex to map out using such a method.

When it comes to XML, I think it does a good job of being functional and being able to fit many of the needs a system has. Through XML, a user or organization can simply query or call data based on prewritten code, allowing a person to optimize the data in a way that will minimize the interference experienced by mismatches in relational information that sometimes occurs with object-relational databases. I think it exemplifies why relational systems are good while simplifying them down to their base components in an efficient way. However, I feel as though if the system is not set up properly using XML, that things could instead be more bogged down if they are not handled correctly.