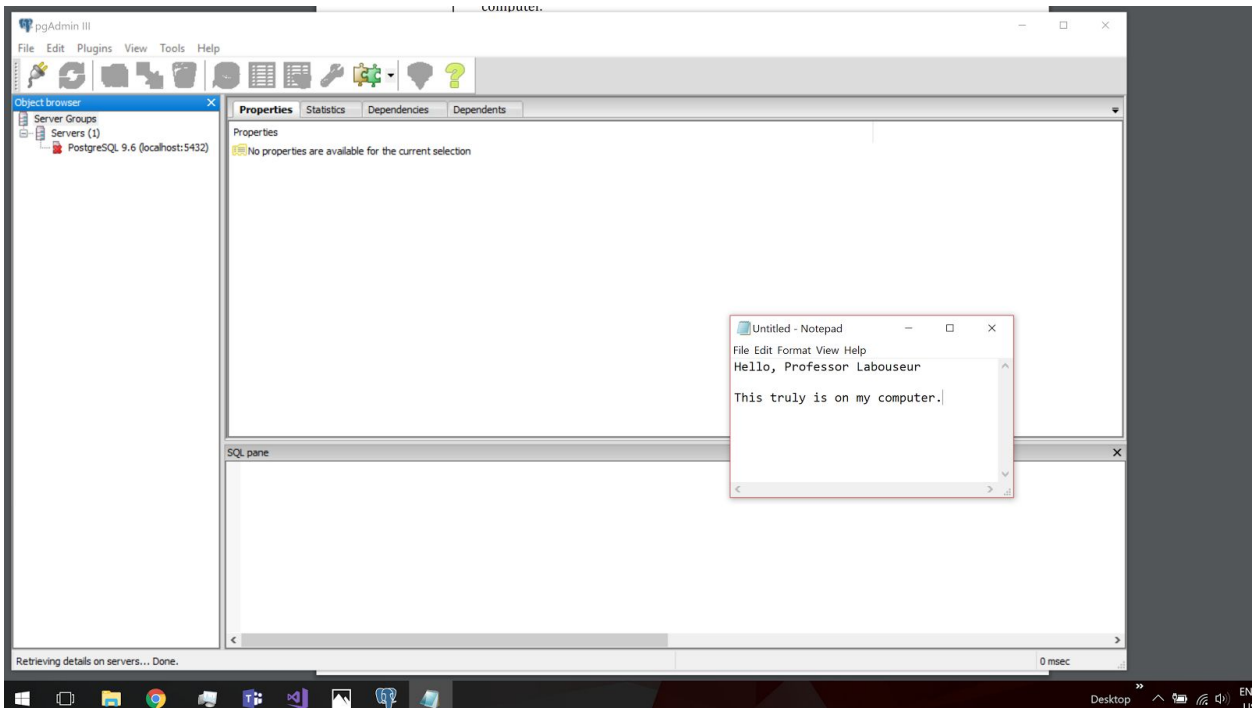
pgAdmin 3



Short Essay: Data Versus Information

In terms of Facebook, their database must store plenty of data that can be interpreted as information. For example, it would hold several strings for values such as first and last names, email addresses, passwords, hometowns, current places of residence, short bios, and status updates. It would also hold other types of values, for one's date of birth (and thus, their age), number of 'Likes', photos, and connections to other members. Alone, a single photo may mean nothing to the majority of people who would see it. This is true for strings like "Faith", "Mazzone", "Anika", "Sharmi", "Geneva, Switzerland", as well as the number of connections one has (624), or the number of 'likes' and 'comments' with a string like "Brunch for Two! Look how perfect this coffee is!".

Facebook allows us to give context to all of our data to produce full information. An image file and a few strings may be entirely indiscernible, but registering the user that posted the photo as well as the other tagged user and other data could show someone that "Faith Mazzone" and "Anika Sharmi" were likely meeting in "Geneva, Switzerland", and the lacklustre photo taken from overhead of a light meal is an image of a meal they shared. Each extra piece of data further richens the information on that specific event and how the general community feels about it.

Facebook, and advertisers, can also draw their own conclusions from the data given. If there are several photos linked to both “Faith Mazzone” and “Anika Sharmi”, one may assume that they are friends in real life and not just acquaintances as the ‘Friends’ list truly implies. Depending on the tagged locations, one may even be able to track their standard meeting points or physical paths in their daily lives. An advertiser may be able to take the number of likes a user makes, as well as the number of photos or status updates, and compare them to the timestamps of each to find how active one particular user is, and the most common times of day they use the site. They can use this to build more information for general purposes “users aged 19-21 are most likely to use the site from 12-2PM and 5PM-12 on weekdays, and are particularly active on weekends where they log in at least once an hour from 10PM-2AM”, and can track the level of engagement (posting photos, sharing videos, or ‘liking’ other content) for each to help target their advertising. This can also be used to extrapolate which physical locations are most popular, who is connected to whom, “popular” users, and other forms of social tracking. This context adds meaning to the otherwise pointless data values.

Short Essay - Data Models

Two common *models* (as opposed to systems) before the Relational Model were the Hierarchical and Network Data Models, the latter being an alteration to the former in an attempt to “fix” some of the most common problems. The Hierarchical Model is similar to a tree graph, with the highest layer at the top. Each piece of information can only have one ‘parent’, but one parent can have many ‘children’. You would have to rely on documentation to hold the information, and IBM’s Information Management System used ISAM in the background of this model. This model has downsides, however, as it means that depending on the use, some data may have to be repeated if used twice. This is redundant, illogical, and occasionally costly if memory and space are issues. To try and combat this, the Network Data Model allowed one item to have multiple parents. This seems logical at first - it removes the need for repetition and the nodes should logically be able to connect to two things at once. However, this can create cycles, and one should not be able to logically move “up” the tree. It causes excess confusion.

Regardless, neither of these models address the scenario that a particular node does not need any branches at that time. The information (and all of its potential connections need to be stored and accessible, but are not being used at that time. This is not uncommon, and neither the hierarchical model nor the network data model account for this. A potential solution would be to create another node that branches all of the unused nodes, but this can throw off data management (such as showing three players, when in reality there are only two plus a placeholder with the unused items).

To my understanding, XML is decent with file storage. However it appears that it is common that XML storage takes up more space than necessary, and with less payoff. It seems that though XML is quite human-friendly in terms of readability, it takes more effort for the computer to read

it which quickly becomes counterproductive. It is also my understanding that it is not particularly quick nor easy to retrieve large amounts of data, making XML less suitable as a database itself, but potentially manageable for file storage.