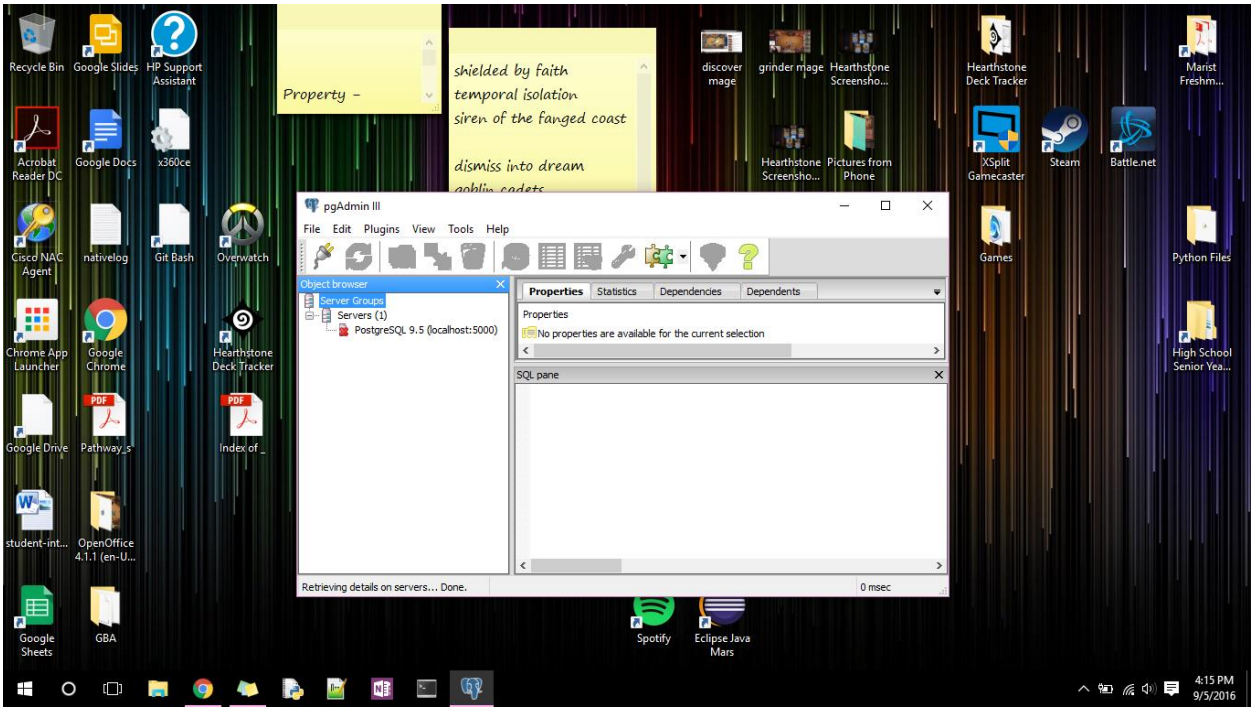
Danny Mulick, Lab 1, Database Management, Fall 2016



Short Essay 1

Databases in the real world have the innate ability to translate raw data into information. This is done through providing connections between pieces of data, creating context for it. Two good examples of this that are common in today's world would be the database for the website TCGPlayer.com, and directory for Marist College.

TCGPlayer uses its database to store data about each card from Magic's history, in the form of the name of the card, its price throughout its print run, the mana cost of the card, the stats of it if it happens to be a creature, and of course the type of card; among other pieces of information. At this point, all of those bits of data coalesce into a complete entity, one sole card. Alone, the data is entirely meaningless; we would have bits saying something close to "0.30, 0.50, 6.00, 200.00" or another field filled with "3RRR". To the common person, these are random numbers and letters that have no relation. Yet, due to the database that TCGPlayer has, we can collect and catalogue it all to lead to a batch of information that allows the average player to gain insight about the card as a whole.

The other example of a database transforming data into information is Marist College's student and staff directory. The directory runs off the LDAP database, which holds information on each person who has held a Marist account. The directory funnels the data which represents the name, phone number, and current location for offices on campus, and for people, holds data which corresponds to their current position if applicable, phone extension, email, and office for staff or faculty. Together, these form one entity that yields information about a person or office.

Modern relational databases are not just collections of data, but houses that draw connections between the bits of data to create information. The entire reason why we have databases to do this is to make the life of the user easier. We review information to draw conclusions based off of it, which denotes its value. Users want to look at data and have it make sense, have context that allows them to understand why we have it.

Short Essay 2

Humanity has used various methods to attempt to organize data and information for millennia. Recently, we have begun to utilize databases to do this for us. Prior to the modern relational database model, we used one of two other models; hierarchical or network. Both of these models each had their own shortcomings, which are why we ultimately replaced them with newer relational models, ultimately coming to XML as the next step towards ease of use.

Starting with the hierarchical model, we organized data in a tier system, putting child fields below the parent fields. This limited the ways we could actually access the data; meaning only the direct parents of a data field could access it, and that no two parents could access the same child element, because that child element could not have multiple parents. The first limitation meant that to completely access the data we wanted, we would have to navigate down the entire tree until we found the leaf that held it. This was overly cumbersome and labor intensive as you had to know the exact path the reach your destination. The second limitation meant that for two data fields to have the same child element, two of that child element would have to exist. This of course led to duplicated fields, which is one of the cardinal sins of database management. In a perfect database, there is only one record that houses a value of data. The network model made strides to fix the second limitation of the hierarchical model.

The network model, although it shared some of the issues that came with the hierarchical model, such as the difficulty in navigating to child elements, did offer a small improvement to the problems with relating multiple parents to one child element. Because the network model actually does mimic a network, nodes can have multiple connections across multiple levels, allowing the parents to match to a single child. Removing the duplicity of the child items is a clear upgrade, allowing the network model to more closely resemble a relational database. However, it still loses when compared to the relational database model, as it cannot allow for easy reading of data from any point in the database.

The XML system, or Extended Markup Language, is used to better organize data into structures to better interpret it, and form information from reading it. I personally believe that XML is going to be one of the next steps in database models and that its implementation would help the general public better understand them. Previous models were clunky and difficult to understand without a background in the field. XML databases can clearly say what each field means by labeling them, and sorting them into objects to contain the field's relationships. However, it can have shortcomings, such as the inability to efficiently read the data without an external command to loop through each object and return the

requested fields. The user must command XML to look into each collection of data, find the field labelled as the one they want, and then tell it to return that field's value and repeat for each other collection. Another modern language, SQL, or Structured Query Language, allows the user to just request the collection that houses all of the fields with the required label, skipping the looping step entirely. This lets the user commit less time to code, and still get the desired result.

Database models have been evolving ever since mankind began to utilize them to store data and form information from the relations between that data. In their early iterations, such as the hierarchical and network models, data was hard to access without navigating directly to it, and allowing multiple paths to the same field was not allowed. Multiple fields with duplicate entries would have to exist to allow for differing relations of data. Modern models and languages remedy these issues by readily building in commands and ways to report back collections of data, such as XML and SQL. Eventually the models and systems for databases will evolve and grow even more, making it easier for the user to draw conclusions based on the information stored within them, and lead to greater discoveries from them.