Carolena Realmuto
Lab 1: PostgreSQL
1/21/17

Data vs. Information:

A popular database used today is Oracle, an object-relational database management system produced by Oracle Corporation. It stores data such as numbers, names, SQL commands, data buffers, and other user information in to a server-side memory-structure called System Global Area. Oracle stores data in the form of tablespaces and physically in the form of data files. There is something called the "data dictionary" that comprises a set of tables and views that map the structure of the database. It contains information such as user info (such as user privileges, integrity constraints, names and datatypes of all columns in tables, etc. Data is notoriously ambiguous if not given context. For example, the number 12312 can mean anything if not given context. Is it a price, is it a measurement, is it a location? We have no idea unless it is given more substance. We can add a $ and then the country to show a specific currency, etc. If we add decimal points to make it 123.12 and then degrees F it becomes a temperature. Furthermore, lets add a location, time and date: Cancun, Mexico at 1:23 on 08/23/16. Now it's a specific temperature of a specific city on a specific day at a specific time. Once data is given context it is now called "information." This is extremely valuable to the database user because now one can interpret and analyze this information. Let's say I have a hundreds of temperature data for Cancun at different times during different days stored in Oracle. I can make a graph for a historical weather trend to hopefully see peak vacation times throughout the year if I'm, let's say, a travel agent. This information can be extremely valuable to my business, but it can only have said value if the original 12312 is given context.

Data Models:

The hierarchical data model, or tree-based model", and the network model requires the programmer to visualize data as its stored. The former has data stored as records (collection of fields containing one value) which are connected to one another through links. The limitation is that this model is inflexible because the relationship is confined to one-to-many. The network model is viewed as a graph in which object types are nodes and relationship types are arcs; it is not restricted to being a hierarchy. It allows each record to have multiple parents and children. The disadvantages are system complexity, operational anomalies, and absence of structural independence. Both models do not support high-level query languages, too. The relational database defeats most of these issues. This new model is based on tables of rows and columns and gives us a two-dimensional way to look at data. It can have multiple relationships such as 1:many, many:1, or many:many, or even 1:1. The database can check for inconsistencies and enforce constraints, support high-level query languages such as SQL and the user can "ask" database questions without writing new code and disrupting its structure and organization. An XML model is similar to the hierarchical data model and can be visually pleasing to the viewer. However, relational databases are better at handling large volumes of data within a system. This data can be updated through transactions that ensure the integrity of the database and the content can be extracted very quickly. XML does not have this management system and using XML with large volumes of data can be inefficient and unreliable. XML unlike relational is self

describing and can be transmitting in its entirety from one party to another. With this said, I would use a relational database to manage large data storage that is updated regularly and I would use XML to deliver formatted data to client programs, etc.