Edward Achziger

Database Management

Short essay #1 Data vs. Information - Select a database in use today (real or imagined) and identify the elements of "data" stored therein and describe how the database organizes the "data" into "information". Give contrasting examples of "data" and "information" that illustrate the meaninglessness of "data" without context and organization. Talk about the value the "information" provides once the component data is given context.

The database I mainly use today would be my iTunes library database of music. In this database, the data would be the song title, the song length of time, the album it is a part of, what artist plays the song, the genre, track number, the mp3 file and so on. iTunes acts like a database because it stores the data for one song, along with as many songs as I want to put in my library. It then put this data into information by organizing it for the user in a single spreadsheet view or whatever view the user wants. So when I go in to play a song by Alabama Shakes in my library, it organizes all the data for me so I can easily see read the information of how many songs I have by them and how long the mp3 file will play for when I click on it. If iTunes weren't able to organize all this data into information, going into the iTunes library and trying to play a song by Alabama Shakes would be very challenging and involve scrolling through tons of long unnamed mp3 files and trying to piece together random titles and artists. iTunes gives users all the content they need for enjoying their music and knowing what song they just played, what song they might want to play next, and giving other songs by the same artist. Without this organization, using iTunes to play music would be meaningless because it would be too hard to sift through all that data and make any sense of it.

Short Essay #2: Data Models - Briefly describe the hierarchical and network prerelational data models. Explain their shortcomings in relation to the relational model. Considering this, what do you think of XML as a model for data storage.

Leaving the 1950's and advancing out of the flat file systems, the first data model was the heretical model. In this model designed in 1960, everything branched down, like a corporate later structure. So in an example of my "Database for a Perfect Fantasy Football Team" the first file would be "FantatsyFootball." From there, more folders would route down with all of the positions. And for each position we could draw down more folders with player names and items assigned to each player name. As we can imagine drawing this out the data model would get real messy towards the bottom and would require items to be created multiple times for each player. Also, a player would have to appear twice in the data model if they played multiple positions. This was one of the major shortcomings of the hierarchical model. For data models to be effective, they would have to pass the ACID test. ACID is an acronym for Atomic, Consistent, Independent, and Durable. The hierarchical model failed on all these test and over time proved to be an inefficient data model to work with.

The next data model that was developed working off of the hierarchical would be the IMS Network Model. Similar to the hierarchical model, the Network Model involved one main "FantasyFootball" file that branched down to positions which then could branch down to players and then to more items associated to the players. What differs from with this model is that it eliminates the duplication of players and items branching down because a player name could be assigned to multiple position groups. This allowed the database to be a little more consistent but it still had short comings in the ACID test. A big problem for the Network Model is that it was not Atomic, dude to the fact that in a long line of items if one failed at the bottom, the whole system would fail. Also, these data models were not durable, for example as each execution as they went done the line, an item could be changed, causing it to be disruptions for all the other players before. All these shortcomings paved way for the

relational model, which solved the ACID test by creating a collection of tables in rows and columns. XML is not a model for data storage because XML from my understanding only really works to move the data from database to database. It would fail the ACID test for its lack of an isolation property since it must use other models to actually store the data.

POSTGRES PHOTO