Zachary Tsouprakos

Professor Labouseur

Database Management: Lab 1

http://www.dba-oracle.com/t_object_hierarchical_database.htm

1/23/2016

1) A database that can be seen today is the one that is utilized by phone systems. There is endless amount data in this system, for example, the countless amount of numbers that are put together to make a telephone number. These numbers are meaningless pieces of data which have no context, until they are placed in a specific order which allows them to become information. When these numbers have context behind them, they are able to be utilized to place phone calls. Since there are only so many numbers, the phone system is very unique because the first three numbers are the NPA, or number plan area, (area code) which allow phone numbers to be repeated. For example, the telephone number 845-560-7010 is completely different than 516-560-7010. The database gives the first three pieces of data a little bit more information than the other 7; the area code provides the general location of where this phone number is located. As I stated earlier, data is meaningless unless it is given context, thus resulting in meaningful information. All of these numbers need to be in specific order with a specific area code so it can correlate to the single person that uses that specific telephone number.

2) The hierarchical model is a data model that is used to organize data in tree-like structures. This model was created by IBM and called IMS (information management System). These databases are easy to understand and organize data in a rank-like system, hence the name, hierarchical model. The first piece of information at the top of this model is the easiest to access and one must pass through this piece of data before accessing more information in the database. As one moves deeper into the hierarchical model, the data becomes more specific. Every piece of information besides the very first once is connected upwards to a parent; a piece of information that is connected under a parent is a child. One must pass through the parent to access the child. As for the example in class, if two different parents have the same child, the “child” must be duplicated so that it can be accessed through each parent. This duplication of data can be seen as a negative where there has to be another piece of information added to the database. Also, in the example we used in class there was information that was in the database but not used but either parent, the only way to show this was to create another parent, which does not correctly express the true importance of this data.

IBM also created this network model where there is a slight difference compared the hierarchical model. One difference with model is that duplication is fixed. In the example used in the hierarchical model, the two parents that have a similar child each had their own copy of the child; in the network model, these two parents each use the same child and a closed loop is

created. It is not possible to access each child or individual piece of data by going through multiple paths. This makes the ability to access all of the data within the model more consistently and could possibly even speed up the time to navigate throughout the model. One downside of this model is the inability to express the items that are not used by the parents. The disadvantages of these two models are both comparable, however, the relational model finds a way to provide an organized data model while avoiding these disadvantages. The relational model uses ACID (atomicity, consistency, isolation, durability), which allows this model to be very reliable. Each query in the database is isolated and run one after another which allows the database to work consistently. In the earlier example, duplication and expression were the major problems with these two previous models. In the relational model, we are able to express what each category of data is and can specify which piece of data relates to another, even if one is not used (item D). Also, we are able to choose from existing data and assign each or add it to a parent (player 1 and 2 both having item B).

Using XML diagrams for storage purposes would not be very practical. In my opinion, XML diagrams would get very messy and difficult to navigate when there is a lot of data that needs to be stored. Navigating through the hierarchical XML diagram can be difficult where you must pass through one parent to go the piece of data and so on. Also, it would be a production to add data to a large XML diagram because the entire diagram would need to move or be manipulated, compared to the relational model where it can just be added to its specific box. In larger scale situations, an XML diagram would not be the most efficient diagram to use and I would rather use the relational model for storage purposes.