Marcus Anton Zimmermann

Labouseur Lab One

Short Essay One

Information is data given context. There are many examples of this in banking. Whether considering names, balances, loans, interest rates, or anything else, these values, stripped of context, are nothing more than numbers and letters. However, when arranged within columns and rows of relational database tables that are connected by key values, the data turns into information that we can understand. Banks have to use relational databases in order to keep track of their client's information. For example, you might have a table titled Customer and columns within that table titled Customer ID, Customer name, and Customer address. Additionally, you might have a table titled Account and columns within that table titled Account ID, Account balance, and Account interest rate. Row entries would then be filled with numbers and/or letters corresponding to their respective columns, and a key, maybe one that equates Customer ID and Account ID, would be used in order to relate the two tables. Following this basic system by which data is given relations, and therefore context, creates information that can be interpreted and understood.

Short Essay Two

Pre-relational databases, such as those based on the hierarchical model (tree-based model) and network model, are able to store large amounts of data over a long period of time. In the hierarchical model, there is a root, children, and leaf nodes. In other words, data is represented by hierarchically nested and tagged elements or nodes. The network model is a little different; child nodes can have multiple parents, avoiding duplication. While both systems were a huge leap in database technology, they certainly had a few shortcomings. Consider the following issues: there's no guarantee that data cannot be lost unless it is backed up, there is no direct support for a query language, schema for the

data is limited to the creation of directory structures, and if two users manipulate a file at the same time, one of the user's changes will fail to appear. The relational model, tried and tested over many years, either avoids or overcomes these obstacles by placing data in the rows and columns of tables that are connected by key-values. It's highly popular and has dominated database management for decades. Despite its success; however, new models are still being developed. Some use XML as a data storage model. While appropriate in some contexts, it has many foreseeable issues. This is because it is hierarchical like its pre-relational predecessors. Knowing the flaws that come with the tree-based approach to database management, I would likely set up my database using the relational model.

pgAdmin Up and Running