Iglika Hadjiyska
Database Management
Labouseur
January 25th, 2016
Lab 1: PostgreSQL short essay

Data Models

The most widely used model for databases is the relational database model, which revolutionized how databases were organized and accessed when introduced by Edgar Codd in 1970. However, before the relational model, other data models, such as the hierarchical and network models, were used, although with numerous problems attached. The hierarchical and network models were used within the Information Management System, created by IBM, long before the relational model was released. The models were slightly similar in design although differed in how users retrieved information.

The hierarchical model can be described as resembling a tree in its design. Databases have a root, with numerous children below them and such. It creates a simply organized system but muddles the process of accessing information about the children and the objects within the database. Within the scope of a database of video game players, should the user want to display all of the items acquired by players in the system, that user would have to iterate through all of the players themselves to find the information they want. Also, the system would not account for duplicate or missing information. If an item is created within the game, it would not be shown in the database unless separately created or found by a player. The hierarchical model makes it cumbersome when the records don't necessarily follow the design of the structure, such as when a child record has more than one parent. Such an instance would not be allowed within a hierarchical model.

The network model is slightly different from the previous model in the way that builds from the tree like structure. Children can now have more than one parent if two records share the same attribute. More broadly, the schema, or structure, of the model does not restrict it to following one structure, like hierarchical. This type of model still faces the same kind of problems as its predecessor, such as the duplication of records and the lose of items declared within the game but not in the database, such as a rare, not yet found item.

One of the main problems found with these two models is that they both do not fulfill ACID, a set of properties used to optimize database systems. It insures that should a part of a transaction fail, the entire transaction should fail to save the database from being changed, should there be a power outage and such (Atomicity). It also makes sure that every piece of data added is consistent with the rules of the database (Consistency) and also that changes should be independent of each other and made visible as they are made (Isolation). Finally, the system also insures that should a change or a transaction be made, it can be recovered and recorded should data be lost from a power crash or mistake (Durability). None of the systems have implementations that allow for all of the attributes of ACID. Relational databases however, record transactions as they happen with a right ahead log, create records that must comply with standards before it can be added to the database, allows transactions to be made one after the other, and finally, doesn't allow partial changes to the database. The previous models did not have these implementations included. Furthermore, these rudimentary models also did not work with higher level languages, unlike the relational model which can run with SQL.

Within these relational models, a special mark-up language called Extensible Markup Language is used to organize data and information within the database. This sort of language,

which operates primarily using tags, works well for such data storage because it allows for easy extraction of information. Compared to other models of data storage, such as key value store and document store, do not support ACID, and sometimes have no structure themselves.