

Lab 1 – Setting up PostgreSQL

Short Essay 1

One example of a modern-day database is one that we use right here at Marist College: the database that stores which students, faculty, and staff have access to which areas of campus by swiping in with their campus ID card. This database would – hypothetically, since I do not have access to the database design – contain the users' names, campus-wide ID numbers, their affiliation with the school (Student, Staff, Faculty, etc.) and department (CS faculty, IT staff, etc.), as well as a list of booleans or other way of storing whether or not a user can access any given room that can be swiped into on campus.

With context for the database – column names such as “Name,” “CWID,” “Affiliation,” “Department,” “Access Level,” etc. – we know that this is a database for maintaining the physical security of the college. Without context, this data could look like just a list of names, numbers, and booleans or “yes/no” fields that mean almost nothing, and we would not be able to tell whether the numbers are campus-wide ID numbers, phone numbers, or social security numbers. Once the data is given context, its value is in the purpose it serves: the security of Marist students, staff, and faculty, as well as physical property (equipment, books, etc.) that the college does not want stolen.

Short Essay 2

In the hierarchical database model, the data is organized and shown in a hierarchical tree, and relationships between data points are shown as connections between the nodes of the tree. As such, each level of the hierarchy represents a different kind of data – such as the example we looked at in class, where the root of the tree was the game world, and the level below that contained all the players in the game world, and the level below that contained all the players' items. The main difference between the hierarchical database model and the network model is that trees in the hierarchical model are not allowed to have closed loops or cycles, and the network model can have these closed loops.

The problem with models using trees or networks is that it is hard to express a data item that has no relationship to the others – such as the example we saw in class, in which no player in a game had picked up a particular item. In that example, we had to put an data item “Unassigned” on the same level of the hierarchy as players, which would make it so that if the players were read from the tree, it would look like there is a player called “Unassigned,” and the game would have to be hard-coded to say that “Unassigned” is not a player.

Considering the disadvantages of using a hierarchical tree or network as a database model, using XML as a model for data storage would not be a good idea, because trees and networks are not ACID compliant, so it would not be suitable for a large enterprise's database. Also, it could be difficult to check a database based on XML for consistency, as you would have to traverse the entire tree algorithmically to ensure that there are no closed cycles or loops (assuming that our XML database would be using a hierarchical tree model, rather than the network model).

Screenshots

