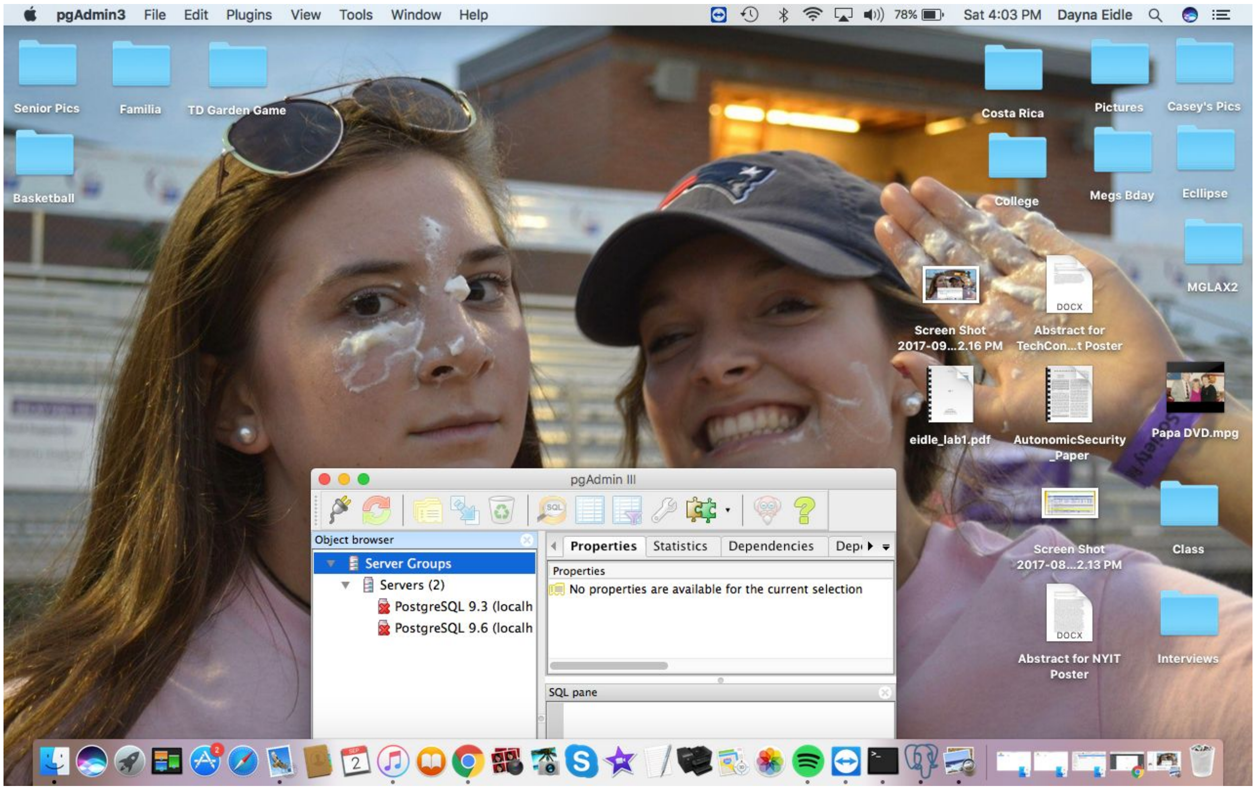
Marist College

Lab 1

Dayna Eidle
Database Management
Professor Alan Labouseur
Due Date: September 4, 2017

Screenshot of pgAdmin III on my computer
(That’s my friend and me as my background so I would hope this wasn’t on someone else’s computer)



No properties are available for the current selection

Short Essay #1: Data vs. Information

Planet Fitness has a database of all the members at their gym. In this database, each person has a list of things describing them and their membership. The database would have sections for things like first name, last name, phone number, age, email address and type of membership (regular or Gold Card). Each member has their own section in the database with all of their member details grouped together. By grouping the details together, each person now has their own profile. That way, if something needs to be looked up about a specific member, like their membership type, it’s easier and more efficient to find what you’re looking for.

An example of data would be a random list of ages. However, once you say that that list of ages is the ages of the members of Planet Fitness (continuing with the previous example), it gains value. Now, one could use this information to figure out the average age or distribution of ages of the people who work out at Planet Fitness. Having that information allows the Planet Fitness marketing department to be able to create ads targeted towards the majority age group.

Short Essay #2: Data Models

The hierarchical data model is setup in a root and branch (or parent and child) type design. There is a root at the top with branches and nodes off of that creating a second level. Then, depending on how big the model is, there can be more branches and nodes off of the nodes created in the second level (therefore making the second level nodes roots to the third level nodes). In this model, each node has only one root being linked to it. The network pre-relational data model is similar to the hierarchical data model in that it has different levels to it. However, in this model, nodes can have multiple roots being linked to them. So, two roots on the second level can both be linked to the same node on the third level, whereas in the hierarchical model, each root would have to have its own separate link to the node on their respective sides of the model.

The problem with both of these models is the case of having an element that doesn’t have a specific place at the moment. In both the hierarchical and the network model, each node belongs to a root, but if there is a node that doesn’t belong to a root, there is nowhere in the diagram for that node to be. This is where the relational model fits in. The relational model provides the opportunity to create a space for elements without direct connections to other elements. Additionally, the relational model shows the specific quantified relationships between different elements within the database (e.g. one-to-many, many-to-one). This helps to give a more detailed picture of the relationships and database as a whole. When it comes to using XML as a model for data storage, to me it looks like a jumbled, diagram-version of the relational model. It shows all of the relationships but doesn’t seem as organized as other models. Also, with there not being as much structure to it, it could quickly get even more difficult to read if new relationships were added. Because of this, I don’t feel that it is a suitable model to use for data storage.