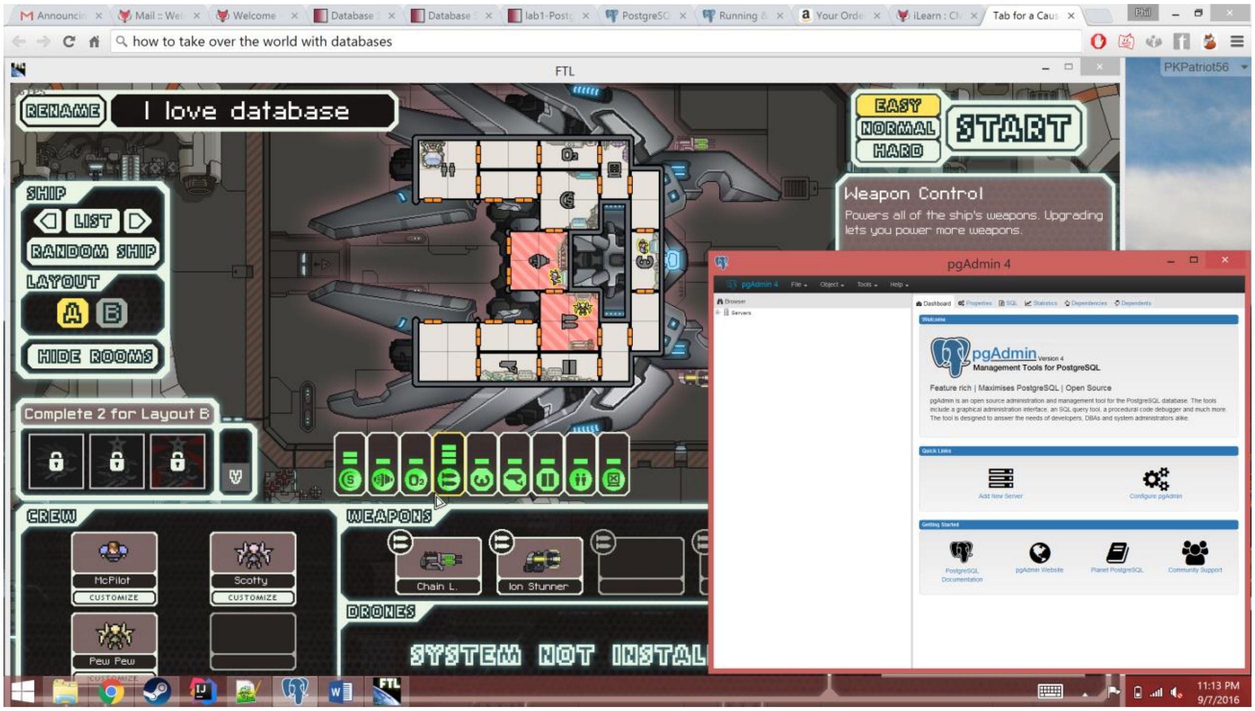
Phil Kirwin

9/7/2016

Lab 1

Proof of successful MySQL installation:



Weapon Control

Powers all of the ship's weapons. Upgrading lets you power more weapons.

pgAdmin Version 4

Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an open source administration and management tool for the PostgreSQL database. The tools include a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

Short Essay 1: Netflix, as one of the largest television and movie streaming services in the world today, naturally has to make clever and efficient use of databases in order to keep track of the massive amount of information that they are responsible for. In this instance, the types of data that Netflix would need to be stored by them would include various names, numbers, and video clips. While this data is meaningless on its own, it then becomes the job of the application used by Netflix to query this database to organize this data into tangible information. For example, the names that would be stored in the database would be assigned to corresponding video clips, which are also provided a run time and a release date from the numbers stored in the database. Another example would be that some names in the database can be related to specific users and thus must be related to a set of payment information that is stored in the database. Without any context, the data that is stored in the database is impossible to access by a general user base, and when context is provided, it allows users to gain information from the Netflix application about specific shows and movies that they are watching, including information as seemingly basic as its name and run time.

Short Essay 2: Pre-relational data models, while very important and certainly useful in the times that they were created, do also suffer from serious drawbacks when compared to the current relational data model. The hierarchical model is based on storing data as a hierarchy, meaning that an object’s smaller components are completely stored within it, and the components of those objects are then also stored within them. As an example, a kitchen would feature itself as at the top of its hierarchy, while the various cabinets in the kitchen would be the next layer, and the contents of each cabinet would be the next layer. The problem with this model, however, is that duplicate data values can exist independent of one another, since the concept of a hierarchy prevents any sort of looping. In the kitchen example, this would be the equivalent of having bowls in two separate cabinets, but having to identify them as totally separate and unique objects because a hierarchy must have two unique objects exist if they are stored in two different places. In an attempt to remedy this problem, the network model was created, which essentially functions the same way as the hierarchical model, except it allows multiple higher tier objects to encapsulate the same instance of a smaller objects thus preventing the need for duplicate entries. As a result, in the kitchen example, bowls could be in two separate cabinets and still be considered the same type of object. However, the network model also has a flaw in that in order to acknowledge the existence of objects that are not attributes of other current objects, a separate, inaccessible object must be created that stores anything that could later be useful. In the kitchen example, this would mean that if none of the cabinets currently had a blender in it, a false cabinet would have to be created to store the blender until a time that a blender was added to a real cabinet. The relational model then solves this problem, for creating separate entities for all data and having another entity made specifically for relating them. In the kitchen example, this would mean that the kitchen, cabinets, and dishes would all be separate entities that other entities would be able to access and connect to one another where necessary. Taking this into consideration, XML seems to be a relatively strong model for data storage compared to relational databases due to its overall similarity to relational databases while still maintaining something of a hierarchy for the sake of user-friendliness. XML still at least seems to overcome the data inconsistency issue of the hierarchical model and has some of the flexibility of the relational database model, but hierarchies in general seems to get much more complicated with large amounts of data, such that a relational database may still be more functional in these cases.