Hannah Youssef

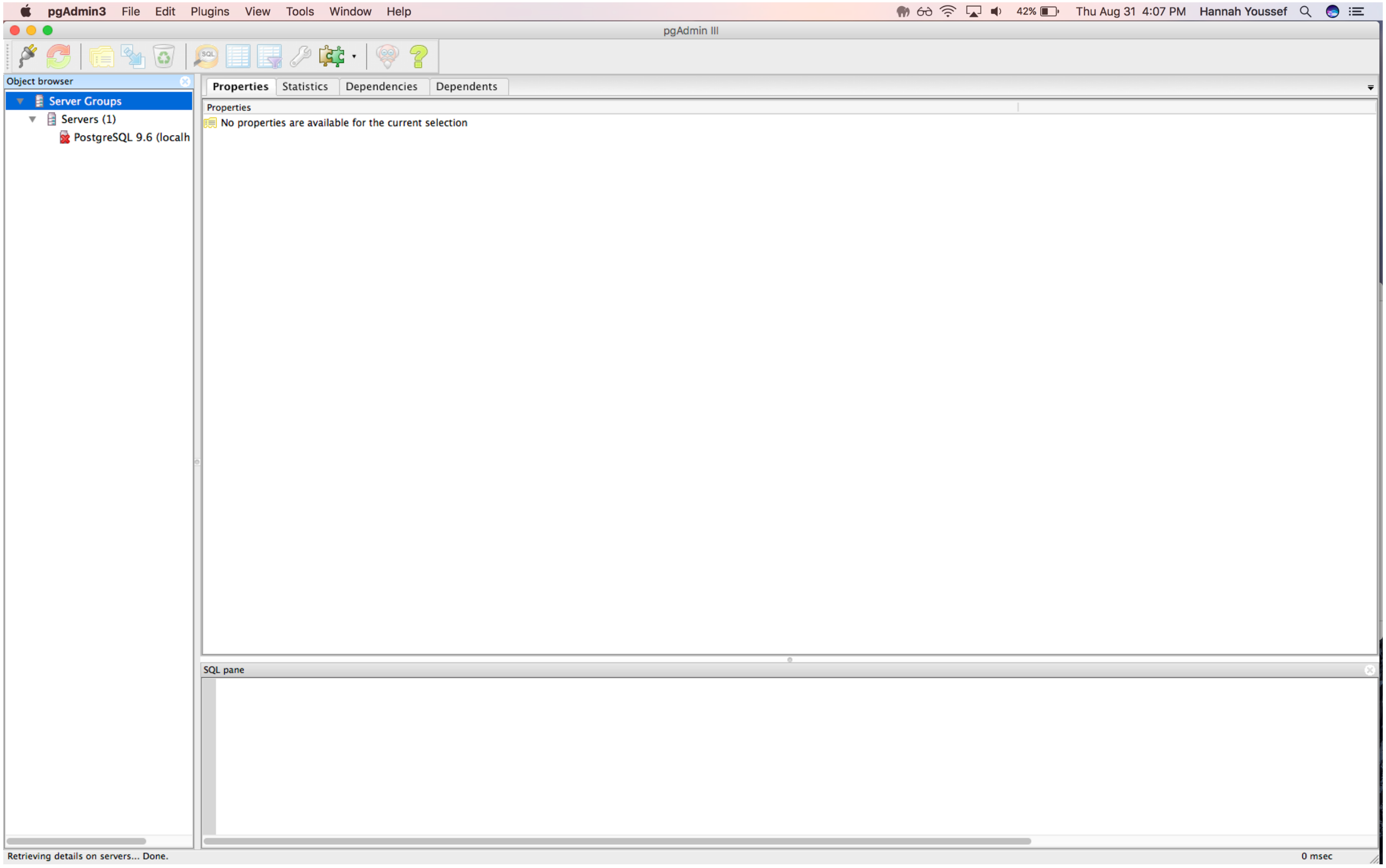
Dr. Alan Labouseur

Database Management

4 September 2017

Lab 1 – PostgreSQL

1. PostgreSQL download screenshot


pgAdmin3 File Edit Plugins View Tools Window Help
42% Thu Aug 31 4:07 PM Hannah Youssef
pgAdmin III
Object browser
Properties Statistics Dependencies Dependents
Properties
SQL pane

Server Groups
Servers (1)
PostgreSQL 9.6 (localh

No properties are available for the current selection

Retrieving details on servers... Done.

0 msec

2. Data vs. Information

A database in use in many hospitals and offices today is a health-care provider database. Some elements of data stored in this database could be a patient's full name, date of birth, gender, phone number, email address, home address, etcetera. These elements of data are stored as attributes in the database, which serve as the column headings of the database table. The database organizes the data into information by organizing data for each patient into a single row with each attribute applying to one patient at a time. Data and information are very different things; information is simply data with context. This fact can be shown using this health-care provider database. One example of a piece of *data* from this database is "10/25/1967". Without any context, this piece of data is useless and has no meaning. An example of *information* from this database is: "Name: John Smith, DOB: 10/25/1967". When providing context about each piece of data, this is now a piece of information that we can use. We now know that John Smith is a patient in this health facility's database whose date of birth is October 25th, 1967. This is now valuable information that we extracted from this database that can be used.

3. Data Models

The hierarchical model was a tree-oriented data model, like semi-structured data. The network model was a graph-oriented, physical-level model. The drawback of the hierarchical model was that it operated at the physical level, which made it very difficult for programmers to write increasingly more difficult code, unlike the more modern models. One shortcoming of the network model is that the generality of graphs was built directly into the model, rather than favoring trees as the other models do. There are many reasons that the relational model is still preferred in DBMS's. One reason is that the relational model provides a simple, limited approach to structuring data, while also being extremely versatile, so virtually anything can be modeled. Another reason is that the relational model provides a limited, yet useful collection of operations on data. These reasons allow users to implement languages such as SQL, and enable programmers to code at higher levels. Considering this, and having knowledge about relational models, I do not think a semistructured data model using XML is an efficient way to store data. Semistructured data resembles trees or graphs, as opposed to tables or arrays, like in the relational model. I understand how storing a large amount of data in a single XML document may seem efficient to some, but the structure of the storage of data in this file is not easy to look at. I do not think it is effective in its storage of data to be quickly looked at and analyzed. The HTML-like tags and paths are not a good way of maintaining a database, as it looks more like programmer's code, rather than a proper database. In my opinion, the relational database is a much more efficient way of storing data that is organized and easy to interpret.