Database Systems

CMPT 308 • Spring 2015

Lab 1: PostgreSQL - 20 points

Goals

- Install PostgreSQL and pgAdmin
- Practice getting around in the PostgreSQL and pgAdmin environments.
- Practice getting around in and using GitHub.
- Explains some key concepts we covered in class.
- Get some easy lab points.

Instructions

1. Download PostgreSQL from http://www.postgresql.org and install it on your computer.
2. Short essay: Data vs. Information - Select a database in use today (real or imagined) and identify the elements of "data" stored therein and describe how the database organizes the "data" into "information". Give contrasting examples of "data" and "information" that illustrate the meaninglessness of "data" without context and organization. Talk about the value the "information" provides once the component data is given context.
3. Short Essay: Data Models - Briefly describe the hierarchical and network pre-relational data models. Explain their shortcomings in relation to the relational model. Considering this, what do you think of XML as a model for data storage?

Resources

- PostgreSQL - http://www.postgresql.org
- Installation guides - https://wiki.postgresql.org/wiki/Detailed_installation_guides

Submitting

Take a screen shot of the pgAdmin tool up and running on your computer. Be creative as to how you will show that it's on your computer. Submit the screen shot and your essays as a PDF document. (Only PDFs will be accepted. Seriously.)

Push your work to your GitHub repository **before** the due date (see syllabus). Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) check-in messages. Then e-mail Alan the url of your GitHub repository.

Joseph Strauss
CMPT 308 Lab – 1

Data vs. Information

Our textbook defines a database as: "… nothing more than a collection of information that exists over a long period of time…"[1] However, this poses the next question, "What is information?" One of Webster's Dictionary definitions for information is: "knowledge obtained from investigation, study, or instruction"[2] If we relate data and information using the relational model, we can think about tables. If we have a table of data, we have columns that distinguish themselves from each column and we have rows that are unique entries where at least one column entry is different from all other rows. Therefore, no two rows are exactly identical. The collection of characters that are entered into the rows are the data. In itself, while the data is the catalyst to all information, it has no meaning. It is an intellectual expression that is input into a field. The first form of information is the column identifiers; this is because the column is what quantifies the expression in the field. The second form of information is much more subjective; the table(s) may be used by those who have access in order to find answers to questions or determine relations of data. Thus data is the expression in the fields and information is the column's meaning as well as the meaning defined by the table(s) reader.

An everyday example of a database is my household monthly budget. At the beginning of each budget cycle my wife and I determine how much money will be allotted to each category for the month. Although some categories fluctuate or are one-time, most categories are consistent. These monthly tables are linked to a yearly table that calculates total spending and tracks changes in spending trends. The input into each "cell" is just a number. The number 215 is data. In itself, this number has no value to us; however, if this number is under the column of "Gas", this is a piece of information. This tells us that we need to pull $215 out of the bank and place it into a "gas" envelope this month. The yearly table will show us that we have been spending more than $215 each month in gas. This gives us a vital piece of information that suggests, "We need to increase our gas budget". Although 215 is just a three digit number, it its context in the table, it is a link in a trend that provides valuable information that helps us make an informed decision.

[1] (Garcia-Molina, Ullman and Widom)
[2] (Merriam-Webster)

Data Models

The first modern data model was the Hierarchical data model. The basic premise is that all data is in a parent, child, and ancestor relationship. For example, at the root of the structure would be a game. The child of that may be players and their children might be attributes. This model solve a couple of problems. First, it supported physical data independence. This means that the game can be ported to a different system and the file structure will stay intact because the hardware and software are not dependent on each other. Second, unlike a flat-file system, there is now a relationship between the data. However, there are major drawbacks to this model. One major drawback is that there is still inconsistency in the data. This is because there must be duplication of data in the model. For example, if the “Mjolnir” is a child to all players but the game developer decides to remove the hammer, it must be deleted from each character. If one is missed, then that player has a non-existent item attributed to them. Another major flaw is that if a parent of a child is removed a piece of data can lose its link to the rest of the model.

The next model invented was the Network Pre-Relational Model. This is exactly what it sounds like. The idea was to remove duplicates in a model by creating a “web” where one attribute can have many parents or many children. This way if a link is removed, it is universally removed from the data set. Although this solves the problem of duplicates, it still has all the other inherent flaws of the Hierarchy. There is a set structure that must be known ahead of time and removing attributes may still leave nodes lost from the model

The relational model is different because there is no child parent relationship. Instead information is placed into row and columns each row is not dependent on the other. Tables can be used together in one-to-many and many-to-one relationships. This means that if a table is updated, the update will propagate to the other tables that are in relation.

XML does not provide a robust consistent model for data. XML does help to provide a framework for how data is stored. The language focuses on a parent-child relationship. XML does work well with displaying data from a database. The headings in XML work well with the columns of a table.

Works Cited

Garcia-Molina, Hector, Jeffery D. Ullman and Jennifer Widom. *Database Systems: The Complete Book (Second Edition)*. Upper Saddle River: Pearson Prentice Hall, 2008.

Merriam-Webster. Merriam-Webster. Web. 21 Feb. 2015. <http://www.merriam-webster.com/dictionary/information>.


pgAdmin III
File Edit Plugins View Tools Help
Object browser
Server Groups
Servers (1)
PostgreSQL 9.4 (localhost:5432)
Databases (1)
postgres
Catalogs (2)
Event Triggers (0)
Extensions (2)
Schemas (1)
Slony Replication (0)
Tablespaces (2)
Group Roles (0)
Login Roles (1)
Properties Statistics Dependencies Dependents
Property Value
Name postgres
OID 12135
Owner postgres
ACL
Tablespace pg_default
Default tablespace pg_default
Encoding UTF8
Collation English_United States.1252
Character type English_United States.1252
Default schema public
SQL pane
Query - postgres on postgres@localhost:5432
File Edit Query Favourites Macros View Help
postgres on postgres@loca
SQL Editor Graphical Query Builder
Previous queries Delete Delete All
Scratch pad
IMG_0225.JPG - Windows Photo Viewer
File Print E-mail Burn Open
master
Sync
Filter repositories
CMPT308_Spring2015
History
Droning about GitHub
5 days ago by Joseph Strauss
Added .gitattributes & .gitignore...
5 days ago by Joseph Strauss
Droning about GitHub
Joseph Strauss 0662f6d GitHub Revert Collapse all
Here we watched a 30min tutorial about Computer spyware ;)
SQL scripts\Class_01.sql 11
@@ -0,0 +1,11 @@
1 + create table people(
2 + name text,
3 + favoriteColortext,
4 + favoriteNumber int
5 +)
6 +
7 + insert into people (name, favoriteColor, favoriteNumber)
8 + value('Alan', 'blue', 007)
9 +
10 + select *
11 + from people
Recycl
Intel(R) Update
Git Shell
GitHub
Performan...
5:21 PM
2/21/2015