Tadd Bindas

Database Management

9/1/16

Lab 1

Essay 1:

One database that is in use today is the Marist database. This database stores tables that hold student ID numbers, meal plan swipes, Marist money, thrifty cash, and other items. There is a primary table that contains student's cwid number, email address, contact info, and has a primary, foreign key relationship with other tables using the student's cwid number. The database organizes information based on categories such a meal plan, housing, and Marist Money (to name a few). These tables are all linked to the table of student names via the CWID (a primary to foreign key relationship). The Marist database takes in data through the swipe of the student's ID card and turns the data into information through various queries. An example of this is the swipe of a student ID card. When a student swipes an ID card to try to gain access to a building, the ID card reader pulls the CWID from the card and eventually finds its way to the database and a query is run to check if the data is in the database. If it is in the database a true is returned and the person is allowed into the building, if it is not in the database the student it no allowed in. The database sorts each individual doorway's access into a separate table with a list of many cwids that are allowed in. This table turns the data, the cwid numbers programmed in, into information, a list of which people have access to a certain doorway. As a reminder, random numbers are data, but not information because we do not have any context. Information is data with context. The number 20062536 could be anything, its data. However, if it is inside of the

CWID row inside the freshman student’s contact table, it is now information. Data is meaningless unless it has context. 20062536 is just a number, but with context it is a useful pin that is the identification number of a student. An example of how context can give data power is knowing someone’s social security number. If someone tells you a random nine-digit number, you could assume it is anything. However, if you determine it is a social security number, then you have the power to steal someone’s identity. The number becomes information.

Essay 2:

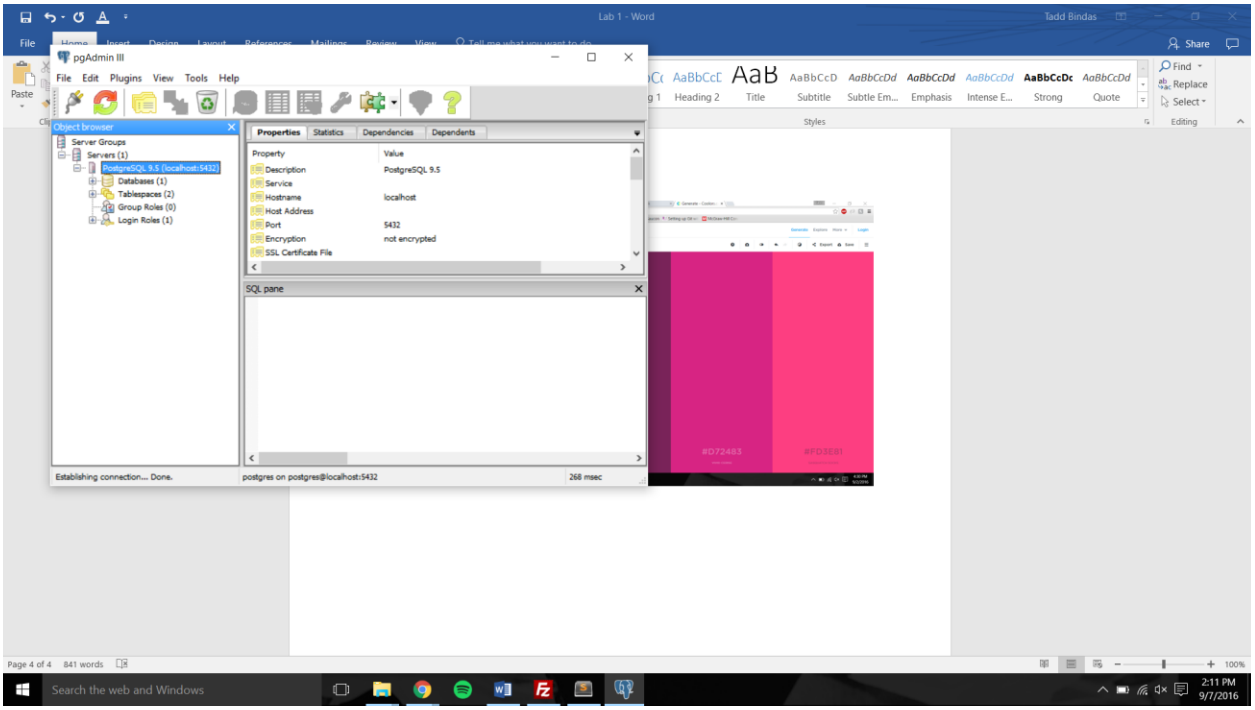
The hierarchical network data model was one of the first ways that coders classified data in a database. The model was a family tree where there was a top node with many children. It was built using COBOL, had physical data independence, and sent a man to the moon. Below is an example of the hierarchical model (I was unable to draw lines, so I used bullet points to determine which nodes were higher up the model).

- Game (root node)
  - Player 1 (child node)
    - Id (node)
    - Items
    - Name
  - Player 2 (child node)
    - ID
    - Name
    - Items

However, despite its successes, the hierarchical network tool still needed to be upgraded. It caused many data duplications that resulted in many errors. In the example I gave, the items node could have had the same data inside, which would cause confusion if called by a line of code. The eventual successor to the hierarchal method was the network model. The network model was a family tree model almost identical to the hierarchical model, however it fixed the issue of duplication. Below is an example of the network model.

- Game
    - Player 1
        - A
        - B
    - Player 2
        - B (points to the same B that player 1 does)
        - C
    - Shop
        - C (Same as P2)
        - D

Node B is shared by both Player 1 and 2 and C is shared by the shop and Player 2. The only weakness to this, is if there was a game made in this fashion, a player could make his/her name “Shop” and steal all of the items in the game. This model needed abstraction, which brought about the relational model which is the current standard for data models. Considering the history of the relational model, and how data models evolved through the past, I believe that the relational model is the future of database technology. I do not believe XML is the future. Using SQL and a select statement is an easy way to collect the necessary data that is inside of a database. Using XML to find data would result in walking the entire document tree until the match of the data is found. Using XML is not as efficient at all. XML is also not efficient with inserting items. If you gave a new item to a player in an XML database, and want to index your XML files, you would have to rebuild the entire index after the insertion. In a SQL relational database model, a new row is just added. These downsides to XML are the reason I believe the relational database model is the future of database technology.



Server Groups
Servers (1)
PostgreSQL 9.5 (localhost:5432)
Databases (1)
Tablespaces (2)
Group Roles (0)
Login Roles (1)

Property	Value
Description	PostgreSQL 9.5
Service	
Hostname	localhost
Host Address	
Port	5432
Encryption	not encrypted
SSL Certificate File	

Establishing connection... Done.
postgres on postgres@localhost:5432
268 msec