Connor Segneri

9/3/19

UIUC – Foundations of Data Curation

Relational Schema Design Exercise

# Part 1

The first file is a text file that inventory uses for relating models to styles, power trains, quantities, and individual cars, etc. The file contains 11 columns (not counting the first column that numbers the rows) and 10 rows. There are no column names, and the file is tab delimited. The first column looks like a unique identifier for the row. The second column looks like the year the car was made. The third and fourth column look like the make and model respectively. The rest of the columns look like other car details, with the final column describing the price of the car. This table relates details pertaining to the cars the dealership has in its inventory, where each new car is a new row in the text file.

The second file is a csv (comma-separated values) file that sales uses to relate people to cars, prices, trade-ins, etc. It has 19 columns (again not counting the first column that numbers the rows) and 11 rows, the first row containing the column names for the data. It’s much easier to discern what data is stored in each column when the column name is present. In this file, each row is a sale. Each contains information pertaining to the sale itself (SaleDate, Discount), information about the buyer (LastName, FirstName, MI, Address, City, State, Country), and information about car the car sold (Year, Color, Engine, VIN, MSRP). There are some columns here that seem to be repeated in the inventory text file, specifically columns pertaining to the car sold in the sale (Year, Engine MSRP). I say “seem” to be repeated because without column names in the inventory text file, I would need someone from inventory to help me verify that the columns in the text file are what I think they are. It can’t be assumed just because the data cells hold similar values that the columns in the text file and the csv file represent the same attribute.

The third file is a word document that customer relations uses for relating people to personal information, information about services, warranties, etc. This word document contains a magical assortment of information related to some very magical people. Each person in the file is a collection of four lines, the first describing their name, the second describing their street address, the third describing the rest of their address, and the fourth describing their occupation. Each person’s information is separated by two new lines. Some people, but not all, have a fifth line. This fifth line looks as though it describes their reason for speaking with customer relations, but without speaking to someone from someone from customer relations it would impossible to be sure. All of this information, except for the person’s occupation and reason for speaking with customer relations, can be found in the csv file that sales uses.

# Part 2

Part two can be seen in the **Exercise1Workbook.xlxs** file.

# Part 3

The populated data can be seen in **Exercise1Workbook.xlxs**.

# Part 4

I represented the data into 4 tables; Customers, CustomerRelationVisits, Sales, and Vehicles. I did this because the relations in each individual table pertains to one “thing”, which the table name describes. Tying together the tables using foreign keys allows the 4 tables to share their attributes without duplicating the data across multiple tables.

I left out the column in the csv file that was called “TradeIn”, because I thought it was redundant. If a sale has a “TradeInValue” of zero, then one can assume that there was not a trade in.

I chose a unique column for the primary keys for each table, so that each row could have a unique identifier. All of the tables had unique integers that increased by 1 for each row for their primary keys, except the “Vehicles” table. For this table, I chose the VIN number for the vehicle as the primary key, as this value is already unique.

I don’t think there were any hard decisions. The hardest decision was deciding to leave out a sales location for the “Sales” table. I assumed that there was only one dealership location. If this was not the case, then the I would consider adding dealership details into my relational schema.

My relational schema supports data physical data independence because this is relational abstraction of the data. Any physical schema changes can be applied without changing interactions of the data. My relational schema supports logical data independence because the logical schema can change, and the change remains abstracted from any physical interaction with the data. We need to adjust the mapping between the physical schema and the logical schema to match any physical or logical changes, but as long as that mapping is adjusted, data independence is supported.

My design schema supports the overarching goals of data curation because with my schema in place, data redundancy severely reduces across different teams within the dealership. Having a singular source of truth will clear up a lot of potential confusion. It also simplifies the process for teams to reach out and grab information from other teams. Before, you would need to convert the data in order to perform any relations, as it was scattered across different file types and schemas. Additionally, there is now datatypes that exist to describe that data where there were none before (the text and word files).

I would additionally recommend documenting the schema, and any schema changes that occur. I would then utilize some sort of source control for the data itself, to make sure that there are versions of the data, and versions of the schema that the company could roll back to if a serious issue arose in the current version.