**Assignment 1. Relational Schema Design**

*1. Write a short narrative description of each file we have given you. What information do they contain? How is the information formatted and organized? What kinds of information is shared among the files? Evidence of in-depth examination of data, overall quality analysis and completeness.*

**1. Input Data Review and Preprocessing**

**1.1 General observations**

For the ease of presentation and to ensure data provenance, i.e. to carefully record all data changes and transformations, I added the following tabs to the Workbook file:

* The original data converted to a tabular format is shown on the *Original Data* tab of the Workbook. It is color-coded to show the data that need modification and missing data.
* I made the changes suggested by color codes on the *Original Data* tab, called this stage preprocessing, and showed the result on the *Changes Documented* tab of the Workbook containing preprocessed input data ready for the final analysis before the import into the database with further optimization.

I can make the following general observation about all the three files:

* The data in the three files were recorded by each department in very different formats, and the files in their current form, without format conversion, cannot be used in one database effectively.
* Information is duplicated among the files:

1. File B contains all the information from File A except for drive and body type (and its own ID column); Make and Model are separate columns in File A and one column in File B.
2. File B also contains all the information from File C except for the Zip Code, Occupation, and Needs Financing columns from File C.

* All the three files have the same number of rows; Files A and C have the same number of columns while their number is significantly greater in File B due to the duplication of data already available in the other two files.

More detailed information about each individual file relating to the data quality, completeness, and preprocessing needed before information can be imported into a database is presented below. Due to the number of data discrepancies identified, it was not easy to stay within the limit of 50-100 words per file.

**1.2 File A *Inventory***

a) Quality of Data

* The file contains information about the car inventory; this is a text file with tab separated values (TSV). “Columns” have no headings. A different number of TSVs may be found in one column (e.g. see entries for make and model) which makes it hard to parse the data automatically and manual intervention is required.
* The Body column contains incomplete or incorrect information about 6 cars and more complete about 4 cars (sedan, hatchback). This can be easily corrected as Tesla Model S is a 5-door liftback (not 4-door as mentioned in File A) and the Fords from File A are SUVs. The only unclear detail is that sometimes SUVs are classified as 3-door SUV or 5-door SUV. This issue needs clarification from the Inventory Department. So far, I left it as 4-door SUV pending clarification.
* The Drive column needs clarification from the Inventory Department in regard to the possibility of showing Tesla (now AWD) and Ford (4WD) as one value of 4x4 instead of two (AWD and 4WD). It doesn’t look that this dealership sells trucks with more than two axles.
* The VIN column seems to be a problem: with a few exceptions, VINs are not 17 characters long (as required according to Carfax.com and Wikipedia), and several different lengths are used (to say nothing about the fact that all the VINs are fake, but that was done for the purpose of the exercise, I guess).
* The Inventory does not indicate if a certain car is still part of the inventory or is sold. In fact, all of the cars in this list are sold according to File B except, maybe, for File A record ID#1 whose status is somewhat unknown as there is no Purchase Price in File B.
* Otherwise, the records seem to be OK.

b) Documenting Changes Before Import to Database

* Converted data to tabular format
* Added meaningful column headings based either on File B or common sense.
* Added the Sold column.
* Completed the Body column

**1.3 File B *Sales***

a) Quality of Data

The file contains information about car sales; it is provided in the CSV format (comma separated values) which can be easily converted into a tabular format (e.g. when opened in MS Excel or using other automated methods). These are the same cars mentioned as inventory in File A. Some data is incomplete, namely:

* Incomplete **record ID#3**: price missing. Needs clarification (deal is not closed or the data were not properly recorded?)
* Incomplete **record ID#5:** this is Ford Explorer Eddie Bauer Edition according to the File A; can be easily corrected.
* Incomplete **record ID#6**: a) State missing; both Bloomington IL and Bloomington IN are close enough to be possible cities of residence; this can be easily corrected as according to File C the state should be IL; b) Information on the discount OR trade in is missing. Seems to be a sloppy record. Needs clarification.
* Potentially wrong **record ID#7**: a July 2015 sales date for a 2016 model (too early for next year’s models?). May need clarification
* Incomplete **records ID#9 and ID#10**: Country missing; can be easily corrected as both places seem to be and actually are (verified) in Illinois, USA
* Zip codes are missing in addresses (can be found in File C)
* Same problem with the VIN column as in File A.

General observations about File B:

* It is not clear what happens with trade-in cars: they don’t seem to become part of the inventory (needs to be clarified with the department);
* White is the most preferred car color;
* Harry Potter lives in Chicago IL; I didn’t know that.

b) Documenting Changes Before Import to Database

* Records ID#5, 9, and 10 were corrected.
* For simplicity purposes, the MI column was combined with the First Name column. I can’t think of a situation when someone would need a search based on middle names.
* The excessive Repeat Customer column was removed: the data is already shown in the Discount column (you can run a search by the Discount column in order to find all Repeat Customers)
* The excessive Trade In column was removed: all data is shown in the Trade In Value column; NULL or a special code can mean no trade in, and another special code, e.g. 1111, can mean there was a trade in, but the exact value is unknown. This will work to get meaningful search results.
* Having four columns instead of one for address may be excessive from the standpoint of reserved space on disk, but I left them because one may need to run a specific search by City, State, or Country.
* The Model column was split into two column: Make and Model (as in File A Inventory).

**1.4 File C *Customer Relations***

a) Quality of Data

The file contains data about customers in a .docx file (MS Word) in the TSV format. Not easy to convert into the tabular format automatically

* These are exactly the same customers who purchased vehicles according to File B.
* There are no “column” headings; I came up with them based on the other two files.
* There is no ID column.
* Harry Potter teaches at UIC; I didn’t know that.

b) Documenting Changes Before Import to Database

* Added the ID column.
* Added meaningful column headings.
* Tried to unify the values in the Needs Financing column.

**4. Schema and Relations**

*How did you decide to represent the data in the way that you did?*

All possible data modification / corrections mentioned above in Section 1 were introduced in order to make the data as consistent and complete as possible. Different database relations were updated based on these modifications, based on each other and common sense for all the issues mentioned above in Section 1.

My database has three relations: *Car inventory*, *Sales*, and *Customers*. Currently, there are 10 tuples in each relation and a varying number of attributes. Most of the attributes in the relations are of the varchar type which is a variable alphanumeric type including *Zip Code* because zip codes can contain letters in foreign countries (and we have attribute *Country*) and even including *State* because this field can be used to record a region in a foreign country which the customer is from (again, since we have attribute *Country*). The lengths of varchar attributes are tentative and may be adjusted based on practical need.

All attributes with a limited range of values were assigned the type *Selection from a list of varсhars*. All attributes related to prices have the type *Money*. Attribute *Year* from relation *Car inventory* is of the *Smallint* type as it is sufficient to represent realistic years. Attribute *Sold* from the same relation is of the *Boolean* type (sometimes called *Bit*). Attribute *Sale Dat*e from relation *Sales* is of the *Smalldatetime* type as it is sufficient to represent values for this attribute.

Appropriate restrictions were introduced for some attributes, namely for those that cannot be Null because a tuple will make no sense in this case.

All the primary keys for the three relations are cross-referenced in Relation 2; however, depending on the specific usage requirements for the database, it is possible to introduce foreign keys in Relations 3 and 1 to relate them to each other and to Relation 2. For a moment, I thought of having a composite primary key for Relation 2 *Sales* that would consist of the two foreign keys: *Customer* and *Car*. But then I realized that, theoretically, one customer may have two or more transactions for the same car if, for example, the previous transactions were incomplete for any reason, and the dealership may want to keep this information in the database separately. That is why I introduced a separate primary key in Relation 2 that consists of transaction codes.

Also, I decided not to use attribute *VIN* as the primary key in relation *Car inventory* because all VINs are incorrect, contain forbidden letters, are of unacceptable varying length, and require additional validation.

Open issues mentioned above in Section 1 still need clarifications from respective departments.

*Did you leave out any information? If so, why?*

Below is a more detailed description of all the changes made in each relation (both deletions and additions):

* **Relation 1** is modified File A: all attribute names were added at the preprocessing stage above; attribute *Car* *ID* was modified to show a unique car ID code – this is the primary key; attribute *Sold* (type = Boolean) is added to facilitate search for cars that are sold or that are still part of the inventory; attributes are rearranged for a more logical representation of data.
* **Relation 2** is based on File B: attribute names were provided as part of File B or modified based on common sense; attribute *Transaction* *ID* was modified to show a unique transaction ID code – this is the primary key. The following data optimization was implemented: File B columns *Repeat Customer* and *Trade In* were removed as containing duplicate information from File B columns *Discount* and *Trade-In Value*, respectively (see the preprocessing stage description); File B columns *Model*, *Year*, *Color*, *Engine*, *VIN*, *MSRP* were all replaced with one attribute *Car* (type = string), which is a foreign key associated with Relation 1, because this information is already provided in Relation 1 and belongs there logically; File B columns *LastName*, *FirstName*, *MI*, *Address*, *City*, *State*, *Country* were all replaced with one attribute *Customer,* which is a foreign key associated with Relation 3, because this information is already provided in Relation 3 and belongs there logically.
* **Relation 3** is modified File C: all attribute names were added at the preprocessing stage above; attribute *Customer* *ID* was added to show a unique customer ID code – this is the primary key; data entries for attribute *Needs Financing* were unified at the preprocessing stage above.

*Why did you choose certain things as attributes? As keys?*

I chose certain things as attributes and keys in an attempt to optimize the data in order to avoid data duplication that can lead to data manipulation anomalies / loss of data integrity.

*What were the hardest decisions you had to make in this design process?*

The hardest step for me during the design process was to determine how to distribute attributes among the three relations and optimize them to avoid data redundancy and related data manipulation anomalies / loss of data integrity), and also to clean File B as it is big and contains a lot of inconsistencies and duplicate information.

*How does your schema design support data independence?*

As it was mentioned on numerous occasions during the lectures, my schema supports data independence by ensuring abstraction from transient details of storage and indirect interaction with the stored data via the relational representation that is mapped to the actual storage representation. The logical schema can change without affecting interactions with the data which means it is possible to add new data constructs now without having an impact on programs (logical data independence). Also, it is possible to modify storage methods without such impact meaning that the physical schema can change without affecting interactions with the data (physical data independence). Operations on data can now be defined formally. All interactions with the data are conducted in terms of relations (containing tuples, attributes, and values); subsequently, this can be translated into data storage instructions.

The original input data were taken from different sources and were provided in very different formats. Now they are integrated and can be used more easily by all data sources (departments). This definitely makes it easier to check data for validity and quality and improves the overall efficiency of the company’s operations though a unified standard approach to handling data.

*How may your schema design support the overarching goals of data curation*

Since data curation is concerned with all aspects of the management of data, my schema design supports data curation by making this management easier and more effective on the bases of the qualities described above. As far as the areas of curatorial activities are concerned, my schema supports:

1) Collection as once you have an appropriate data model, you have a better idea about what data should be collected and how it should be integrated;

2) Organization through an appropriate data model and standards;

3) Storage because the data has physical independence making storage more reliable;

4) Preservation as my schema ensures the understanding of the semantics of data elements and makes it easier to audit it;

5) Ease of Access and retrieval through appropriate data organization;

6) Identification because the schema contains identifiers and facilitates in developing methods of identification;

7) Integration through schema alignment by accommodating data from multiple sources;

8) Reproducibility though documenting data processing and analysis;

9) Sharing though a uniform format for all the departments;

10) Communication because well-organized and effectively integrated data serves as a good basis for analysis and insights;

11) Provenance through the identification of inputs, modifications, and actions responsible for data values;

12) Modification, Compliance, and Workflow indirectly through a more clear structure of the database based on the appropriate schema which makes it easier to understand and analyze data and take appropriate actions.

*Which curation activities could enhance or sustain the database for future discovery and use for new purposes?*

The curation activities that could enhance or sustain the database for future discovery and use for new purposes may include:

* Preservation;
* Discoverability;
* Access;
* Workflow;
* Identification;
* Further Integration (in case of new data and data types);
* Reproducibility;
* Provenance (in case of new data and data types);
* Modification and management of change;
* Security (access control).

*What additional activities would you recommend?*

In the future, I would recommend to implement:

1. Data validation - clarify all the open issues mentioned in Section 1 hereof in order to make the data more understandable and reliable;
2. Access/revision control (which may include adding additional attribute(s)) to ensure that data can be changed only by authorized employees and that the history of such changes is recorded in the database.