Assignment 1: Data Curation using the Relational Model

1. Understanding the Data – Narrative Descriptions

Because some of the tables lacked column headers, I created files within the workbook with the name “ORIGINAL DATA ___” for each file. This is so when I reference columns in my descriptions it is clear.

1.1. Inventory

The inventory file, after some clean up and adjustment can be broken down into the following fields:

Field	Description
ID	Consecutive numbering of data
VIN	Unique ID for each Vehicle
Year	Year of Model
Make	(Ford, Tesla, Toyota, etc.)
Model	Model from Maker (Make: Ford -> Model: Flex)
EngineDetails	Extra Spec Details
DriveWheels	4WD, AWD, etc.
Color	Color of Vehicle
NumberofDoors	Number of Doors
BodyStyle	Sedan, Hatchback, etc.
Engine	Internal Combustion, Electric, Hybrid
MSRP	Suggested cost of vehicle

The overall quality of this data was poor. To begin with, there are no column headers that signify what each columns data is holding. This is a major issue as although the inventory manager may understand it, any newly onboarded members or managers from the other departments would have a difficult time deciphering the data. Data is also missing or incorrect from various locations within the dataset. Toyota vehicles are missing engine details, and one Toyota Corolla has a duplicate “Hybrid” marking in the Engine details when this value should only be located in the Engine column. Body Style (Sedan, Hatchback) are only labelled for the Toyotas but none of the other vehicles. Drive Wheels for the 2019 Ford Flex is marked with both 4WD and AWD. As for the MSRP column, each of these prices is surrounded with quotations and spaces which would hinder any ability to query data. Another note is that, in my opinion, the ID column is redundant. Having a consecutive listing of the data points adds nothing to the

data because these values do not correspond to the other files. For example, there is a similar ID column in Sales, but “1” corresponds to Ford Flex SEL in Inventory while “1” corresponds to the sale of a Tesla Model S. These IDs would simply cause confusion when trying to join these data sets together.

1.2. Sales

The sales file, which required no clean up, can be broken down into the following fields:

Field	Description
ID	Consecutive numbering of data entries
LastName	Last Name
FirstName	First Name
MI	Middle Initial
Address	Address of Client
City	City
State	State
Country	Country
SaleDate	Date of Sale
Model	Model AND Make of vehicle
Year	Year of vehicle
Color	Color of vehicle
Engine	Internal Combustion, Electric, Hybrid
VIN	Unique ID for each vehicle
MSRP	Suggested cost of vehicle
Discount	Type of Discount applied (Blank if none)
TradeIn	Whether trade in was done during sale
TradeInValue	Value of train in
PurchasePrice	Price vehicle was purchased
RepeatCustomer	Whether customer was a repeat

The overall quality of the sale data is a lot better. The sales manager has clear column names for almost all of the columns (MI is a bit confusing, MiddleName is a more sensible name). It is stored as comma separated values (csv), which is a very commonly used data file. Some data is not populated, for example one state, a few country, and a purchase price are missing. There are some redundant columns, for example, if this dealership is located in the Illinois area, the chances they sell to someone with an

address outside of the USA is almost 0. Therefore having a Country column fully populated with USA is a waste of storage. RepeatCustomer is also a redundant column. Within Discount, “Repeat Customer” is listed as one of the reasons for the discount. Effectively the data for this column would be covered by the Discount column. Another note is that the Make and model are combine into one column, when they should be split into a Make and Model column like the inventory file. Aside from these redundancies & missing values, the overall accuracy and structure of the data is solid, especially compared to the inventory data. It is important to note that the VIN, Make/Model, Year, Color, Engine, & MSRP are the same as the Inventory file. Ideally, we would join these two files through the VIN number.

Finally, it is a bit suspicious that Draco M. Malfoy does not have a purchase price from his sale. I would inquire with the sales employee how this transaction was committed & where the information of the sale is.

1.3. Customer Relations

The customer relations file, which required no clean up, can be broken down into the following fields:

Field	Description
LastName	Last Name
FirstName	First Name
MI	Middle Initial
Address	Address of Customer
City	City of Address
State	State of Address
Country	Country of Address
ZipCode	Zip code of Address
Profession	Customer’s profession
CustomerNeeds	Needs of customer

This data is by far the most complete, but lacks structure greatly. There are no column names or any structure at all, it is simply placed in a word doc separated by tabs and new lines. The redundancies with the Country column that were discussed from the Sales table apply here as well. ZipCode is an additional field not listed in either of the other tables. The issue with MI not being a clear column name remains the same here (I assigned the name MI to keep consistency with the Sales table). Another flaw I noticed

was the CustomerNeeds column. It has “Needs loan” and “Needs financing” and “Inquiry into financing options”, which are too similar/vague to be effective data. The column itself is a proper metric to track, but the inputs should be descriptive yet concise. A data flaw that I would personally want cleaned up is the Profession column. Yes, it is descriptive in that it allows us to extrapolate the budget/price point each customer would purchase a vehicle at, but it needs to be clearer (i.e. high medium or low budget, etc.). There is also no primary key that would connect this table to the other tables, although I do not believe it directly needs a PK to the Inventory, a common ID between Sales & Customer would be very helpful.

Overall, the data lacks clear Primary Keys, uniform data types and entries, has redundancies, and is missing some data.

2. Database Schema Design

Schema design in attached XLSX workbook, see “Schema Design”

3. Example Tables

Example Tables in attached XLSX workbook, see “Table Design”

4. Database and Schema Design Process

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

The thought process I followed when designing this schema was to maximize data storage efficiency & while still capturing all key data & storing it in a structurally sensible manner. This means some tables have new data types, previously blank data now have choices specified in the schema, redundant columns and data removed and/or shifted to different columns based on sensible data structure, etc. Details regarding this will be discussed in the upcoming sections.

4.2. Did you leave out any information? If so, why?

Yes, columns were removed and shifted throughout this process.

To begin, we can discuss data that was simply removed due to its redundancy or its purposelessness.

Inventory:

- The “ID” column in the original Inventory file, which was simply a consecutive numbering of the datapoints was removed. It provided no purpose except for a count of our data.
- Removed the data value “Hybrid” from the Engine Details column as it was a duplicate entry and in the wrong location.

Sales:

- Any location related information was taken out. Sales data does not require a client’s address, and more specifically this data is already in the Customer Relations table.
- Any car data was also removed, such as Make & Model, MSRP. All of this can be accessed through a join on the VIN_ID
- Lastly, all the name information was removed and replaced by a Customer_ID. This is because it is duplicate data that more correctly belongs to the Customer Relations table, and by creating a Customer_ID we can still join and access that information for each sale.

Customer Relations:

- Country column was removed from this table because as an Illinois based car dealership, it is highly unlikely they would be making overseas sales. Thus, we are storing an all “USA” column of data which is a waste of data storage.
- The last column, CustomerNeeds, was not necessarily removed but instead standardized. Instead of a description of their needs a simple flag of LoanRequested was created, that tells whether the client inquired or needs a loan, vs no loan requested. This is more efficient from a data perspective & allows for easier querying in the future.

4.3. Why did you choose certain things as attributes? As keys?

Key Information:
Primary Key = ☆
Secondary Key = ✪

Inventory

Field	Reasoning/Description
☆ VIN_ID	This is the PK of the table, and for an inventory table this is the most sensible PK to use as it uniquely identifies each vehicle in the inventory.
Year	Year of vehicle is a valid inventory attribute so was kept.
Make	Maker of vehicle is a valid inventory attribute so was kept.

Model	Model of vehicle was a sensible inventory attribute so was kept.
Model_Details	This column was utilized to further describe the model of the vehicle. I originally thought about combining it with the Model column, but this way vehicles with the same Model but different specs can still be queried easily.
Drive_Wheels	Drive wheel type is a valid inventory attribute so was kept.
Vehicle_Color	Vehicle color is a valid inventory attribute so was kept.
Num_Doors	Number of doors was a valid inventory attribute so was kept, but the datatype was altered to be an INT to allow for better querying.
Body_Style	Body style of vehicle is a valid inventory attribute so was kept.
Fuel_Type	Fuel Type, referring to type of engine/power of vehicle, is a valid inventory attribute so was kept.
MSRP	MSRP of vehicle is a valid inventory attribute so was kept.
Sold	This Boolean flag was added for the specific reason that not all inventory data will always be sold vehicles. In this set of data it was, and it is important to keep track of vehicles purchased in case of future maintenance, returns, etc. But if a vehicle has been sold already and is still in the inventory, there is a chance the same vehicle could be sold twice, which would cause extreme issues with customer relations. Thus having this Boolean to track a vehicles Sold status will negate this from occurring.

Sales

Field	Reasoning/Description
☆ **Customer_ID**	Customer_ID is a custome made ID to identify customers. It is simply a concatenation of the first and middle initials followed by a last name and separated by a "-" delimiter. This is so that we could remove all the name information while still having a way to connect to the Customer Relations table. Either VIN_ID or Customer_ID would have been fine PKs, but Customer_ID was chosen as the "Sale is to the customer".
✪ **VIN_ID**	VIN_ID was kept in the sales table and chosen as a secondary key. This is because we need a way to connect to the Inventory table to get further details regarding the vehicle sold when

	querying, and this also allows us to connect across from Customer Relations -> Inventory.
SaleDate	Date of sale with datatype DATE was kept as date of sale is a sensible attribute in the sales table.
Discount	Discount was a sensible sales attribute so was kept.
TradeIn	TradeIn is a sensible sales attribute so was kept, although it was altered to a BOOLEAN type to help with future querying.
TradeInValue	TradeInValue is a sensible sales attribute so was kept, but the datatype was altered to a FLOAT type with 2 decimals as to help with future querying.
PurchasePrice	PurchasePrice is a sensible sales attribute so was kept, but the datatype was altered to a FLOAT type with 2 decimals as to help with future querying.

Customer Relations

Field	Reasoning/Description
☆ **Customer_ID**	Customer_ID is a custome made ID to identify customers. It is simply a concatenation of the first and middle initials followed by a last name and separated by a "-" delimiter. This is so that we could remove all the name information while still having a way to connect to the Customer Relations table. Either VIN_ID or Customer_ID would have been fine PKs, but Customer_ID was chosen as the "Sale is to the customer".
LastName	VIN_ID was kept in the sales table and chosen as a secondary key. This is because we need a way to connect to the Inventory table to get further details regarding the vehicle sold when querying, and this also allows us to connect across from Customer Relations -> Inventory.
FirstName	Date of sale with datatype DATE was kept as date of sale is a sensible attribute in the sales table.
MiddleInitial	Discount was a sensible sales attribute so was kept.
Address	TradeIn is a sensible sales attribute so was kept, although it was altered to a BOOLEAN type to help with future querying.
City	TradeInValue is a sensible sales attribute so was kept, but the datatype was altered to a FLOAT type with 2 decimals as to help with future querying.

State	PurchasePrice is a sensible sales attribute so was kept, but the datatype was altered to a FLOAT type with 2 decimals as to help with future querying.
ZipCode	ZipCode is a sensible sales attribute so was kept
Profession	Profession is a sensible sales attribute so was kept, but it is used as a reference for the created Household_Earnings column.
LoanRequest	Was originally a nameless column with some description regarding the financial needs of the client, and was polished into a BOOLEAN attribute to represent whether a loan was inquired about or requested such that querying in the future is easier.

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

A couple things come to mind regarding this, namely how to track sold vs. unsold inventory & what data to keep in which table.

Name & Address information makes sense in a sale, as we want to know the information of the customer that we have sold a vehicle to. But this felt more suited in the customer relations table so it was removed from the sales table, while still being accessible through the PK.

Tracking sold inventory correctly was a business case I felt that was not considered in the data. Let's say we have 5 new cars that come into inventory, and our sales team is trying to sell these vehicles. What if two sales members sell the same car to different individuals? That would destroy any trust the customer base has in this business. Thus, tracking the sales in the inventory would allow us to know whether a vehicle can be sold, and in case a salesmen is unsure whether a vehicle can be sold, they can simply query the data for all the vehicles that are not sold yet. To do this we have a Sold column in our inventory with a BOOLEAN datatype, so it is both memory efficient & allows us to access the information we need.

4.5. How does your schema design support data independence?

This schema supports data independence because if new columns are added it will not affect the user view. The structure is built in such a way that all data is accessible across all tables as long as the correct joins are used.

For example:

If we wanted to query for the make and model of a vehicle that customer's that are doctors have bought, we can write something like:

```
SELECT cr.LastName, cr.FirstName, cr.MiddleInitial, cr.Profession, i.Make, i.Model
FROM Inventory i, Sales s, CustomerRelations cr
WHERE i.VIN_ID = s.VIN_ID
    s.Customer_ID = i.Customer_ID
    i.Sold = TRUE
    cr.Profession LIKE "%doctor%"
```

We can see that data is accessible across all tables while the data is still independent to its table. If a new field was added to any of these tables a user would be able to access these new columns while it still not affecting their view of the other columns.

4.6. How may your schema design support the overarching goals of data curation (revisit objectives and activities of Week 1)?

This data supports the overarching goals of data curation in many ways:

- Organization: this model has data in locations that are sensible within the context of the tables
- Storage: By removing redundant columns and changing datatypes to more efficient ones, we are maximizing storage use while still preserving all important data
- Discoverability: By creating sensible primary and secondary keys, we can access data across tables without any issue

Some of the goals/activities of data curation are not necessarily applicable in this assignment as this is not a fully accessible database, but the main principles that it can affect have been considered throughout the process of design.

4.7. What are the pros and cons of your schema design?

Pros:

- Sensible PK & SK selections
- Datatypes are more efficient
- Tables are legible (Column headers are easily distinguishable as to what the data present is)

Cons:

- Would require team to follow new data standards

- Team would have to learn database access query languages (MySQL)

4.8. Which curation activities could enhance or sustain the database for future discovery and use for new purposes? What additional activities would you recommend?

Security is a big component that needs to be considered with this data. Namely, there is an entry for one Draco Malfoy that lists a sale with no purchase price. This is highly suspicious, as 10s of thousands are simply missing from the system. Did someone tamper with this data? Was it accessed inappropriately? Security precautions & standards need to be in place such that something like this does not happen again.

Modification is a component I think needs to be considered with this DB. The team originally had very unstructured data, and a methodology needs to be established so this data cleansing remains to the same standard it is now at.

Reproducibility will be important in the future. This business has very little data now, but in the future, there will be more data, more columns, more keys, etc. that will be added to the data. When this happens, we need to ensure that we consistently query the correct data and that we are joining our tables correctly, or we can start to confuse and lose our handle on the data.