CSC343:

Term Project

Phase 2

November 19th, 2021

Prerak Chaudhari | 1005114760

Jaakulan Subeethakumar | 1005225757

Design Decisions

In phase 1, we proposed to further split the cars-data csv file into 3 separate files: one for miles per gallon, one for horsepower and one for the remaining metrics. This is because some rows are missing a value for either miles per gallon or horsepower and we wish to enforce attribute requirements for all columns in our tables.

The only feedback we got from the TA on our schema was that this change was unnecessary because there is a 1-to-1 relationship between carID (the primary key of the table) and these parameters. Therefore, we will be keeping the csv file as is, and adding nullable columns for MPG and Horsepower to our Parameters table. Given that both mpg and horsepower are never both null for any row in this csv file, we can simply add a where clause to our queries that filters out tuples with a null in the interested column.

Also, we updated the column type of Displacement to be FLOAT because we found a row with a float value for this column.

The updated Parameters table is shown on the next page so that it fits on one page.

Aside from this, the proposed schema from phase 1 is unchanged. We know this schema is good because not only is it easy to understand, but it eliminates redundancies in each relation. Furthermore, each relation has an integer primary key and its foreign keys are also integers. This speeds up comparisons and lookups, compared to using a string key which is present in some relations, because integers are smaller than strings.

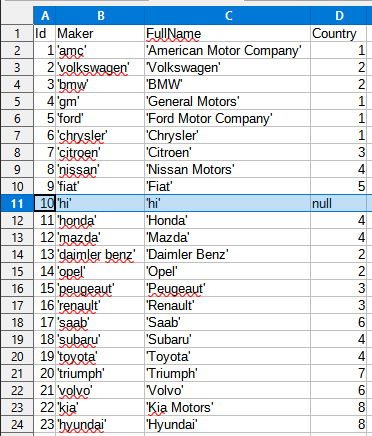
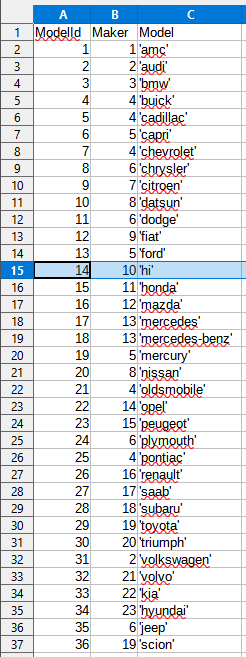
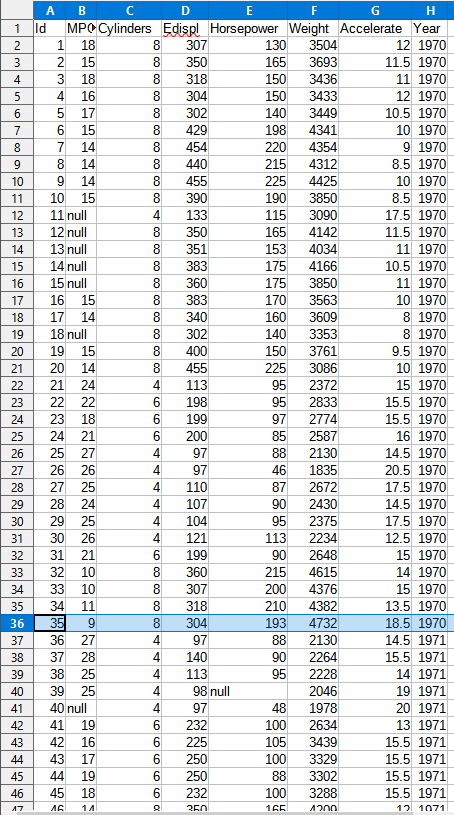
**Parameters**

**(CarID, Cylinders, Displacement, Weight, Acceleration, Year)**

A tuple in this relation represents the other operational parameters of a car.

| Attribute | Description | Type | Required |
| --- | --- | --- | --- |
| CarID | unique identifier for each car | INT | YES |
| MPG | mileage per gallon | FLOAT | NO |
| Cylinders | number of engine cylinders | INT | YES |
| Displacement | engine displacement volume in cubic inches | FLOAT | YES |
| Horsepower | engine power in horsepower | INT | NO |
| Weight | car’s weight in pounds | INT | YES |
| Acceleration | 0-60mph acceleration time in seconds | FLOAT | YES |
| Year | car’s year of production | INT | YES |

Cleaning Process

1. The nonsensical manufacturer ‘hi’ whose country of origin is null was removed from the car-makers csv file to enforce attribute requirements:  
   
2. The corresponding model was removed from the model-list csv file to avoid foreign key violations:  
   
3. There is 1 car of model ‘hi’ in the car-names csv file that had to be deleted to avoid foreign key violations:  
   
4. This car’s metrics also had to be deleted from the cars-data csv file to avoid foreign key violations:  
   
5. In the car-names csv file, there are 2 instances of erroneous whitespace padding that were corrected by removing the whitespace before the opening quotation mark:  
     
   
6. In phase 1, we proposed to replace the Model column in the cars-names csv file with its numerical identifier from the model-list csv file. This is done to reduce redundancy as the model name should not be present in multiple relations. While this may come at a minor performance cost due to additional natural joins, we believe it is good practice to design the schema this way. For example, our first query deals with car manufacturers as we need to determine their country (thus, continent) of origin. This schema design means we don’t need to re-lookup the model name; furthermore, our comparison and lookup times are faster because we are using an integer key versus a string key (integers are smaller than strings).  
     
   This change was accomplished using the pandas library:

| import pandas as pd  carNames = pd.read\_csv('car-names.csv') modelList = pd.read\_csv('model-list.csv')  combined = pd.merge(carNames, modelList, on='Model') df = combined[['Id', 'ModelId', 'Make']].sort\_values(by=['Id'])  df.to\_csv('car-names.csv', index=False) |
| --- |

1. All instances of the word “null” in the cars-data csv file are replaced by an unquoted empty string as per standard CSV format for handling null values:  
   