Members: Alex Shrestha, Josiah Altschuler, Grant Saylor, Musab Ali, Neha Mathur

INST327 Group 8 Final Project Report

Introduction

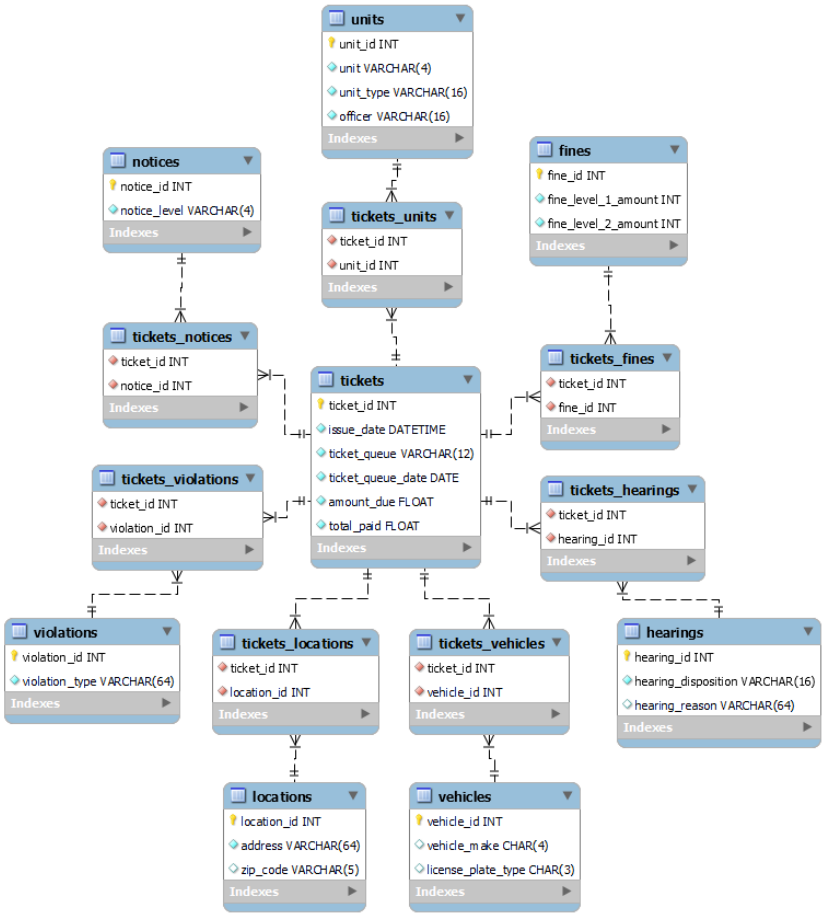
Our database is made up of traffic citations issued by The City of Chicago. The citations include all the parking violations, speeding, and running red lights given on Jan 1st, 2021 from 9 to 11:30 AM. The attributes include Notice Number, Ticket Number, Issue Date, Violation Location, Zip Code, Violation Code, Violation Description, Unit, Unit Description, Officer, Vehicle Make, License Plate Type. Fine Level 1 Amount, Fine Level 2 Amount, Current Amount Due, Total Payments, Ticket Queue, Ticket Queue Date, Notice Level, Hearing Dispo, and Hearing Reason. We decided as a group to choose this topic for our project because we found it the most interesting and useful since it would help the people/government in Chicago. Also, this seemed to be the relatively easiest database to manage/clean up and to create questions on.

Our main audience for this database will be the local government in Chicago, journalists, people living in Chicago, and people traveling in Chicago. The local government would use this database to figure out which streets/intersections they need to update the signs for or hire civil engineers to change up. Journalists can use the data to spread awareness of which areas in Chicago to be aware of so they can inform people to drive safer to not get a citation and to bring awareness of over-policing in certain areas. People living/traveling in Chicago can use this data to inform them of which streets get the most citations so that they know to drive carefully in those areas.

Database Description

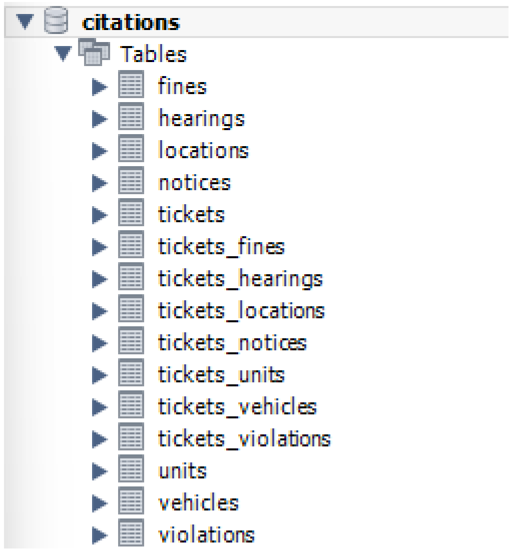
- **Logical Design**

For the logical design of this database, we elected to split all major entities into their own separate tables with separate primary keys – no composites – and connected each table to the central tickets table with linking tables. In total, we have 8 primary tables and 7 linking tables, for a total of 15 tables.



- **Physical Database**

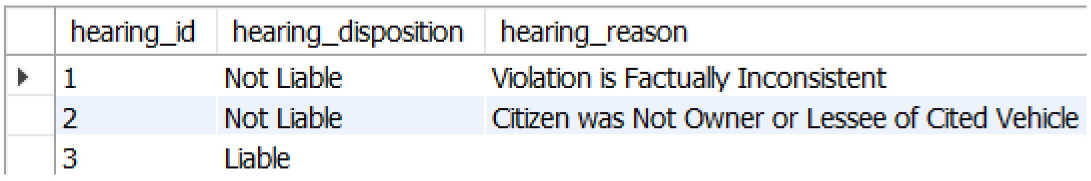
We created a physical database using the sample data below by downloading each table from Google Sheets as a .csv file, and then importing it into MySQL Workbench using MySQL Workbench's Table Data Import feature. The resulting tables can be seen below:



```
▼ citations
  ▼ Tables
    ▶ fines
    ▶ hearings
    ▶ locations
    ▶ notices
    ▶ tickets
    ▶ tickets_fines
    ▶ tickets_hearings
    ▶ tickets_locations
    ▶ tickets_notices
    ▶ tickets_units
    ▶ tickets_vehicles
    ▶ tickets_violations
    ▶ units
    ▶ vehicles
    ▶ violations
```

- **Sample Data**

The primary tickets table contains exactly 500 rows of data. Each of the smaller tables (notices, fines, etc.) contains a varying amount of data, from three rows to hundreds of rows. The smaller tables are that small because they should be, by definition. The hearings table, for example, contains only three rows because there are only three ways that a hearing can go: the driver is found liable; they are found not liable because the violation is factually inconsistent; or they are found not liable because the citizen was not the owner or lessee of the cited vehicle. This table is pictured below:



hearing_id	hearing_disposition	hearing_reason
1	Not Liable	Violation is Factually Inconsistent
2	Not Liable	Citizen was Not Owner or Lessee of Cited Vehicle
3	Liable	

The sample data was obtained by group member Alex Shrestha splitting the original dataset between several sheets in a Google Sheets spreadsheet, and then using the Google Sheets UNIQUE formula to obtain the unique values from each table. These values were then assigned primary keys, allowing us to use very complex Google Sheets formulas to create the linking tables. One such formula can be seen in the box below:

```
Unset
=IFERROR(INDEX($D$2:$D$247, MATCH(C2, $E$2:$E$247, 0)), "")
```

Following this, we used a very complex Python script written by Josiah Altschuler to take data from the dataset and automatically create the 7 primary tables (aside from tickets) and 7 linking tables in our database. These were used to check the accuracy of the data generated in Google Sheets, uncovering a quirk with the way that Google Sheets' UNIQUE function works with multi-column datasets. An excerpt of this code can be found in the box below:

```python
if columnNumber == 1:
  uniqueColumnData1 = list(uniqueValues.values())
  uniqueColumnData1.insert(0, column1)
elif columnNumber > 1:
  uniqueColumnData1 = [column1]
  uniqueColumnData2 = [column2]
  for tuple in uniqueValues.values():
    uniqueColumnData1.append(tuple[0])
    uniqueColumnData2.append(tuple[1])
  if columnNumber == 3:
    uniqueColumnData3 = [column3]
    for tuple in uniqueValues.values():
      uniqueColumnData3.append(tuple[2])
```

- **Views / Queries**

We successfully created five view/queries for our database which together satisfy the project requirements.

View Name	Req. A	Req. B	Req. C	Req. D	Req. E
amount_due_per_violation_location	X	X	X	X	
occurrence_of_violations	X		X	X	
tickets_per_unit	X	X	X	X	
percent_of_tickets_with_hearings	X	X	X	X	X

number_of_tickets_per_specific_vehicle_type	X		X	X	

Changes From Original Design

Since the project proposal submission, one of the first changes we made was updating the scope of our project to include all the columns from our database in order to come up with richer questions and queries. Originally, we excluded columns related to police officers as they included terminology that we couldn't understand, and excluded a column about the type of license plate involved for the same reason, though we cited our genuine belief that it was irrelevant as our reason for removal. Thankfully our group member, Alex Shrestha, was able to locate multiple relevant information resources: the first of which was able to explain almost all of the officer-related columns, as well as the small gaps in our knowledge across the columns we intended to include, and the second of which was able to explain the weird coded language that the Chicago Police use for different types of license plates. The second change we made was changing our data collection time frame from 12 AM - 2:30 PM to 9 AM - 11:30 AM. Third, we decided to update the tables that we will use after we went through the normalization process.

While going through the data cleaning process, we decided to bring back linking tables into our database design, as per one of the query requirements, after originally removing them for the normalization process. As a result, we were able to get rid of the composite keys in our fines table and remove all the foreign keys in our tickets table as those would be moved to their own linking table. Furthermore, the tickets table itself went through a few more changes such as moving the columns vehicle_make and license_plate_type in their own new vehicles table, as well as moving the columns amount_due and total_paid from the fines table into the tickets table as those columns more relate to each ticket rather than the fine.

Database Ethics Considerations

In order to be inclusive and check for biases, we will be analyzing the data to check for regional trends, and thereby demographic data. Racial biases are common within the American justice system, particularly within Black and Hispanic communities, which can cause an overwhelming amount of data to be collected on only certain races. According to a ProPublica analysis, traffic cameras tend to ticket Black and Latino drivers the most. Black and Hispanic zip codes tended to have twice as many traffic tickets as white area codes, and also those zip codes tend to have more traffic cams and more surveillance in general increasing racial disparities in data. Due to these statistics, checking for bias is essential in all human-focused data science particularly crime-related ones like the Chicago database picked for our project. To this end, we created our first view/query, which searches for trends in enforcement by address. We also created a query to determine the number of tickets given by each unit of the Chicago Police Department. By being cautious and aware of biases, we believe that we have a diverse and inclusive database and accompanying queries.

The database we created does not pose any potential copyright, fair use, or other legal concerns. The data we used is traffic violation data from the city of Chicago. All of the data that we are working with is public, meaning anyone can get access to the dataset. The data points that are being used in our database are necessary for us to answer the questions that we posed in our project proposal and in our views/queries to our database. In the event of a data breach, our data will not raise any privacy concerns since the dataset that we will be working with can be accessed publicly through the internet, and therefore the sample data plan does not require us to use any data from a proprietary or closed data source. From an ethical perspective, one concern that we have is that we are looking at data for one day rather than a month or a year's worth of data. Since we only have data for one day, it drastically limits our use case possibilities and limits our ability to make accurate characterizations of long-term trends. The data is also from January 1st, so the frequency of fines may not be representative of an average day in Chicago. Furthermore, we are only looking at about 2 and a half hours of traffic violations, so it further limits our ability to provide accurate analysis.

Lessons Learned

Throughout the course of our database project, we learned several things at different stages of the project. One of the first problems we encountered was data normalization, and we encountered difficulties with normalizing our fines tables. We had two fine-level columns in the given dataset, and we had to contact TA Pekerti to assist us in normalizing that table. The second issue we were confronted with was understanding the meaning of the rows and columns that were in the dataset. Alex had to do a deep dive on the Internet to find a document that described a dataset similar to ours so we could understand what the column titles stood for and what the acronyms and other jargon in the data meant. An example of one such row was Zip Code. We initially thought this column referred to where the driver was caught committing a violation, but after Alex's research, we were able to determine that the zip_code column had data on where the vehicle was registered (i.e. the zip code of the driver's home). The third issue that we came across was when it came to cleaning the dataset, we discovered that it would be very challenging due to us deciding to use not only all columns in the dataset, but also the maximum row count of 500. This would make it essentially impossible to assign primary keys manually, so we needed to use automated solutions like our Google Sheets formulas and Python script in order to create the datasets correctly.

Potential Future Work

One of the main aspects that we would like to improve in our database is the amount of data that is contained within the document. We would like to include demographic information as one of our extensions to the database because we feel that it can generate important queries. One of the queries that we would like to look into is if there are any demographics that are being targeted by the Chicago Police Department. We can also generate queries regarding what demographic has the most traffic cameras near their location, which could possibly indicate a racial bias in the positioning of the cameras. We would also like to add the zip code of the address where the violation occurred because our current dataset does not have that information. Overall, we feel that our dataset has a great foundation and if expanded over a larger amount of time, can be used to find tendencies and insights for Chicago traffic violations.