Project 2 Final

US Accidents Database (link)

Michael Dobson
Luke Garrett

Introduction:

The dataset we chose to implement for this project contains information on all reported accidents throughout the United States between the years of 2016 and 2021. The dataset in the form of a csv identifies each accident by a unique letter number key (Ex: A-1). We chose this dataset because for each respective accident there was a wide breadth of information involving location, severity, weather conditions, and other relevant information. Due to the extensive amount of information available we thought providing an application that could efficiently traverse and analyze the data would be a useful tool. Our predominant goal for the application was first to create an intuitive interface for reading and manipulating the database. We chose to use a PHP based website for this as neither of us had much experience with front-end design and we thought of building off of Dr. Singh's template would be the best way to ease ourselves into front-end development. The application itself provides an easy to navigate interface for interacting with the database itself. Users can look up accidents based on the ID in the database and also insert, update, and delete into the existing database. Additionally the application contains options for users to analyze the data for instance there is a tab that can be accessed to sort certain weather conditions by the average accident severity that occur during those specific conditions.

Overall we believe we have accomplished the goals we initially set out for ourselves when we began working on this project. Our initial goal was to create an application that provided a non-technical user a platform to be able to search, modify, and analyze the database, all of which are accomplished in our final application.

Final Implementation:

Description:

Our final implementation is based on a PHP connection to the MySQL server running the database. Originally using Dr.Singh's php client code as a starting point we found a more suitable graphic template and expanded on Dr.Singh's code to provide more functionality for the accident databases. The application focuses on reading and manipulating the dataset in an efficient manner.

Final Database Design:

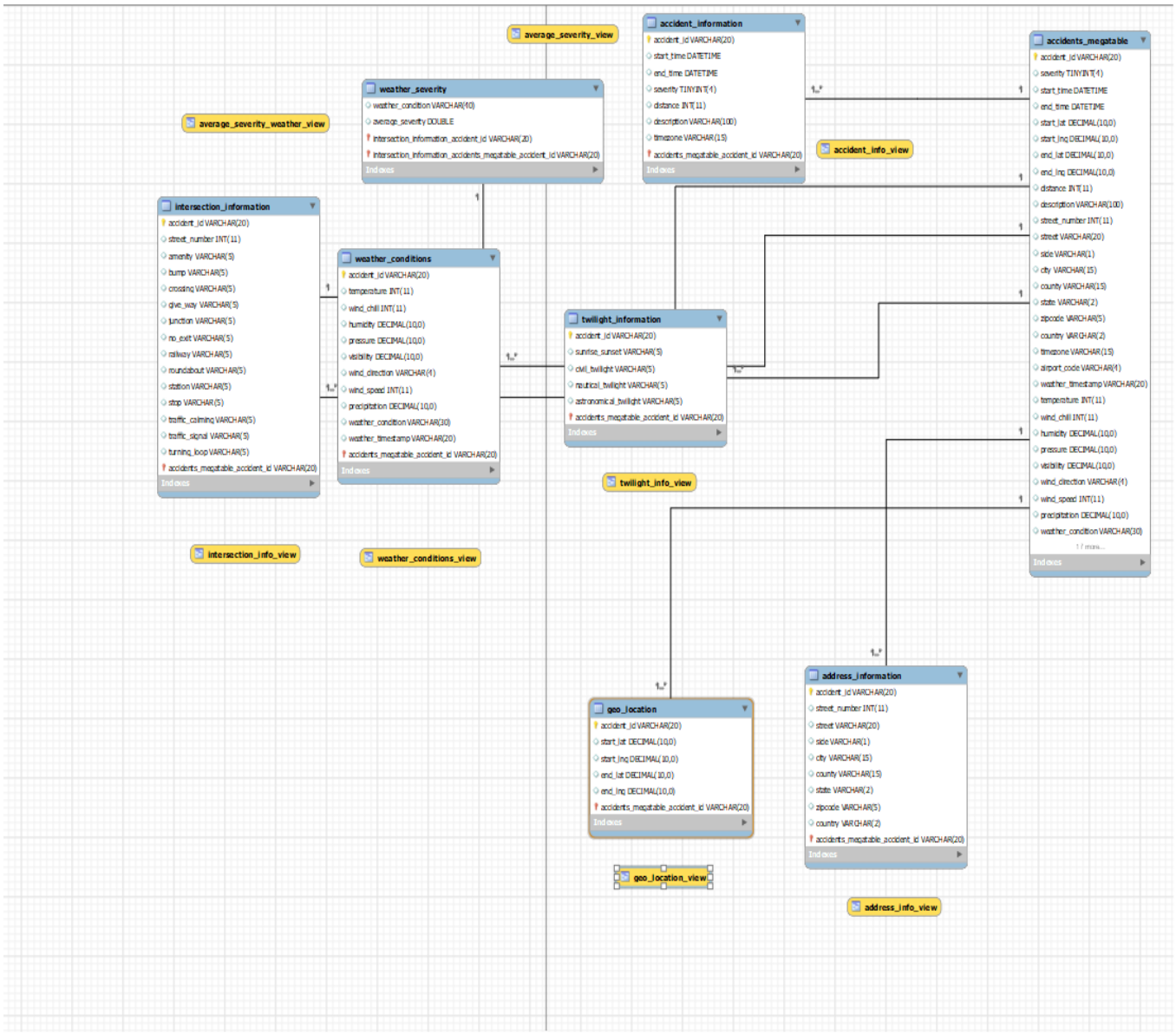
Normalization Process:

The normalization process for this dataset was not nearly as rigorous as normalizing the election data in project 1. There were not many functional dependencies in the database itself as most of the information was singularly dependent on the accident ID primary key. Because of this we were able to obtain a 3NF normalized database by intuitively decomposing the csv into like categories for decomposed tables.

Design:

The primary table in our final design is the accident_information table. This contains the id, start time, end time, severity index, distance, description, and timezone. From this table all subsequent tables that utilize accident id as their primary key, reference the accident id of the accident_information as a foreign key. Next we have the address_information table which contains the residential information of the accident's location. Additionally we have a geo_location table which houses the latitudinal and longitudinal measurements for both the start and end of the accident. Next we have an intersection_information table which houses data with regards to type of intersection or road the accident took place at, for example if this accident occured at a roundabout or stop sign. The next table is the twilight_information table which houses data regarding the sun position. The next table in the database is the weather_condtions table which houses data with regards to the current weather and visibility conditions during the time and place of the accident. The final table is called weather_severity and this table is used for analysis of the database. The weather_severity table is populated by each possible weather description and the average severity of the accidents that occur during said weather condition.

UML:



Testing:

Since the majority of our application focuses on reading and inserting data into the dataset we did not need to implement a large portion of test code. Before implementing queries into the PHP files we did run extensive sample queries on the backend to determine that what we were implementing into the final application was providing the necessary. One thing we noticed when we were implementing the insert function is that a severity could be inserted that is not in the actual possible severity range of an accident (0-4). This would also skew the average severity for the weather_severity table thus we decided to implement triggers on insert that would round the severity to the nearest extreme if it were out of the desired range. As for testing our

procedures we just used sample data from the dataset to determine if they were performing correctly.

Summary of Implemented use cases:

The core implementation focuses on returning and altering data from the us_accidents database. The application allows for intuitive analysis of the database as well as the ability to manipulate and update the data stored in the database.

Functionality:

Search Accident: accessed through drop down menu or icons located at the bottom of home page. Allows user to search for a specific accident by inputting the accident id.

Get Accident and weather information: Accessed through get specific information drop down in the main drop down menu. Returns the attributes of an inputted accident id from the accident_information and weather_conditions tables.

Get Accident information: Accessed through get specific information drop down in the main drop down menu. Returns the attributes of an inputted accident id from the accident_information table.

Get Weather information: Accessed through get specific information drop down in the main drop down menu. Returns the attributes of an inputted accident id from the weather_condition table.

Get Average Weather Severity: Accessed through get specific information drop down in the main drop down menu. Prompts user to input a minimum severity level and returns the weather condition and its average severity of all weather conditions that exceed the average severity input by the user.

Insert an accident: Accessed through the add or edit accidents dropdown from the main dropdown or through the icon at the bottom of the home screen. Prompts user to input the attributes that correspond with the accident_information table and then inserts rows corresponding to the users input.

Update an accident: Accessed through the add or edit accidents dropdown from the main dropdown or through the icon at the bottom of the home screen. Prompts user for accident id and severity to and updates severity to the new value of corresponding to the respective accident id.

Delete an accident: Accessed through the add or edit accidents dropdown from the main dropdown or through the icon at the bottom of the home screen. Prompts user for an accident id to be deleted from the database and deletes corresponding accident id.

Summary Discussion

Status and Stopping point:

The current application is fully functional and achieves all of the goals that we set out for ourselves at the beginning of the project. There is potential to add more types of analysis if we wish to expand past our goals.

Challenges:

The largest challenge we faced when implementing this project was how to deal with front-end development. Neither of us have had anymore than surface level exposure to front-end or web development. Thus there was a steep learning curve involved with creating the application interface and doing so in a manner that would reflect the complexity in our back end design. This obstacle was overcome in a piece by piece manner. We began by just implementing small changes to the template that was provided, as our knowledge expanded we shifted focus to creating an interface that better suited our goals. Towards the end we were faced with challenges as to how to implement certain use cases that did not directly reflect use cases in Dr.Singh's template. This was accomplished via Michael Dobson taking on the brunt of the research and programming necessary to effectively implement these features into the front-end development. Because of this Luke Garrett chose to focus on more of the complex implementations in the back-end to balance out Michael's front-end work load.

Division of Labor:

During the first two milestones tasks on the front-end and back-end were split in an ad hoc manner between the two of us. Neither of us had front-end knowledge and we both were well versed in DBMS implementation for the back-end. Because of this most of our basic back-end implementation was easy to accomplish and we both completed those tasks as they arose. The initial front-end development was done by both of us using Dr.Singh's template as the core for the early implementations of the user interface. As stated in the previous section closer to the end of the project we were tasked with making more complex front-end design implementations and because Michael Dobson had slightly more success with the early front-end manipulation he chose to focus his time and effort into determining the best way to implement our desired front-end changes. To even out the workload during this time, Luke focused on implementing the more complex changes and advanced features in the back-end. The project milestones and reports were broken up relatively evenly with each partner filling out sections that corresponded to the work and changes they had implemented as of that point in that project. Overall we think the work was split it up in as even a manner as we could while still maintaining the integrity of the project.