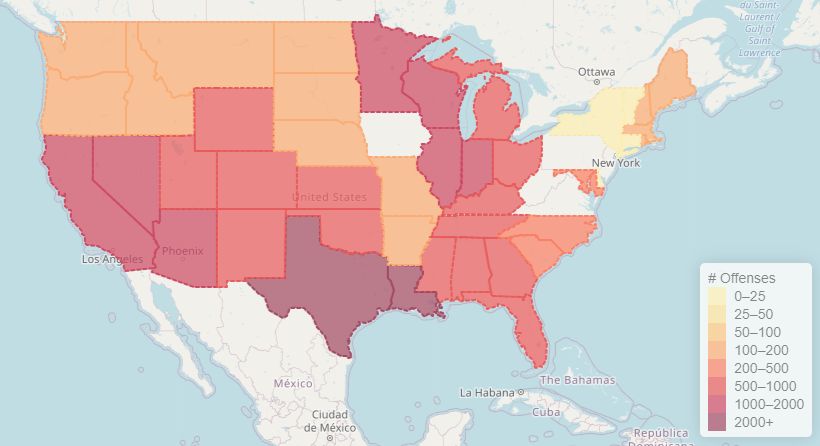
**The Human Trafficking Synopsis**



*Final Report*

Lauren King

Ivan Perez

Jason Gilman

Adam Armendariz

CSCE 310 – Database Systems

Department of Computer Science and Engineering

Texas A&M University

December 4, 2020

**Table of Contents**

[Problem Description](#_kljiyq3i2q6d) 3

[Contributions](#_ou9ysjg36byg) 3

[Database Design](#_porpr8s4z07m) 3

[Entity Relationship Diagram](#_b1o60147xd1l) 3

[Relations](#_6rgzzw3aogdm) 4

[Normalizations](#_p3knguo624t5) 4

[Sample Queries](#_ljctowi3mb6i) 5

[User Interface](#_dh1kc6uqpen2) 5

[Links](#_afcbwudwdp3p) 8

### **Problem Description**

Human trafficking is an issue that still plagues our world today. As of 2016, there was an estimated 40.3 million people affected by human trafficking throughout the world. These issues are not only overseas, but in the United States as well. There are millions of cases of missing and exploited people who have been victims of sex trafficking and forced labor. A factor that assists traffickers is ignorance and staying in the shadows. Most people don’t know where human trafficking occurs and remain mostly blind to the dangers in their own areas. Our project's goal is to gain a better understanding of these statistics and attempt to visualize and contextualize the large amounts of data out there regarding this issue.

### **Contributions**

Lauren King - initial database design

Ivan Perez - database design back end, database initialization, scripting

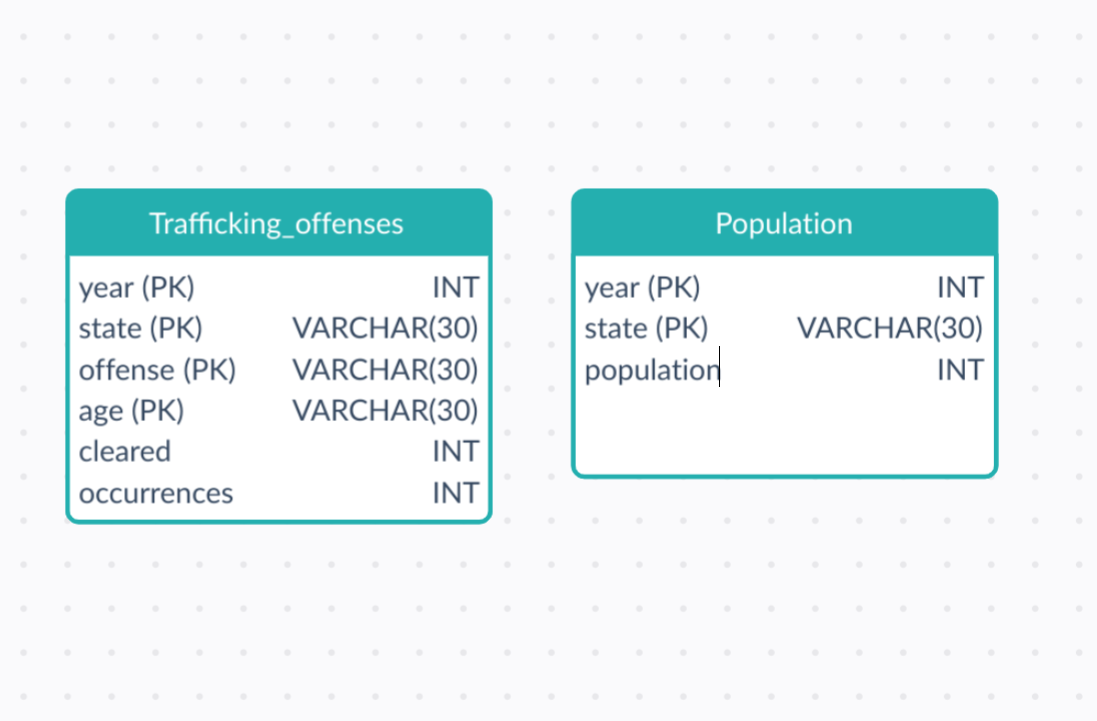
Jason Gilman - front end; UI and visualization of data, documentation

Adam Armendariz - front end; Visualization of data and assist with UI

### **Database Design**

### **Entity Relationship Diagram**

Our team made quite a few changes to our original ER Relationship Diagram. With the given time constraints, we were able to simplify our database alongside our entity relationship diagram given that most of the data that we were originally working with was corrupt upon further investigation. This relationship diagram is at the core the necessity structure we need to achieve our team’s mission statement.

**

### **Relations**

**Trafficking\_offenses**

* year INT
* state VARCHAR(30)
* offense VARCHAR(30)
* age VARCHAR(30)
* cleared INT
* occurrences INT

**Population**

* year INT
* state VARCHAR(30)
* population INT

### **Normalizations**

The original table contained these columns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | State | Offense | Age | Cleared | Population |

The primary key of this table is (year, state, offense, age) because within each year are states and within each state are victims who are either adults or juvenile, these adults and juveniles can be subject to both sexual trafficking and labor trafficking.

This table was split into two tables, ‘population’ and ‘trafficking\_offenses.’

**Population:**

|  |  |  |
| --- | --- | --- |
| Year | State | Population |

**Trafficking Offenses:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | State | Age | Offense | Cleared |

### 

### **Sample Queries**

Most of the queries we use in our application are structured around the user’s inputs. The user has the opportunity to specify the data to be visualized by selecting the years, types of offenses, ages, area of interest, number of occurrences, and number of clearances that will be returned from the database. Since there are many optional input opportunities, we heavily relied on the INTERSECT operators when structuring our queries. This allows us to join two selections together that meet both selection’s criteria while not capturing duplicate entries. An example is as follows:

SELECT \* FROM trafficking\_offenses WHERE year=2014 OR year=2017

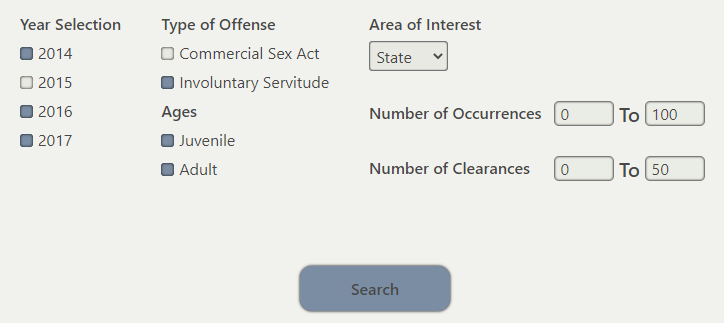
INTERSECT

SELECT \* FROM trafficking\_offenses WHERE age=”adult”

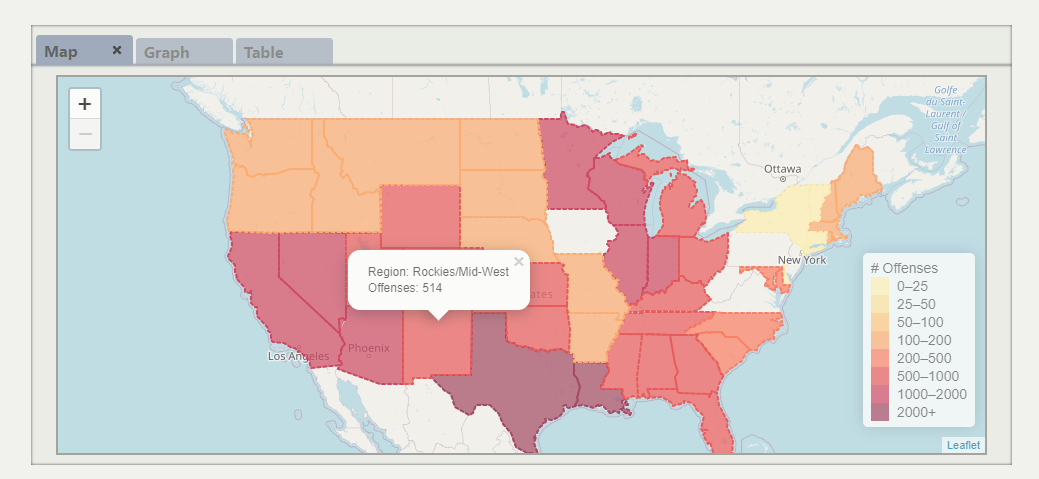
For ease of use in the application, we separated each input into its own selection query, and used the intersect operator to join the data together so that we could visualize the common rows. Most of the queries sent to our database from the application are structured very similarly to the one above, although there are several additional selections to reflect each user input. Our application did not require complex queries to capture the data necessary for visualization. We currently do not have functionality to update, insert, or delete data through our application. Any administrator functions have to be done manually through the SQLite interface.

### **User Interface**

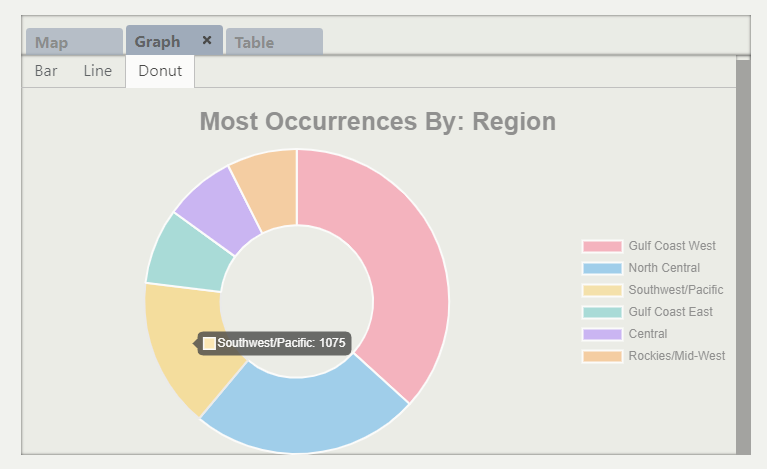
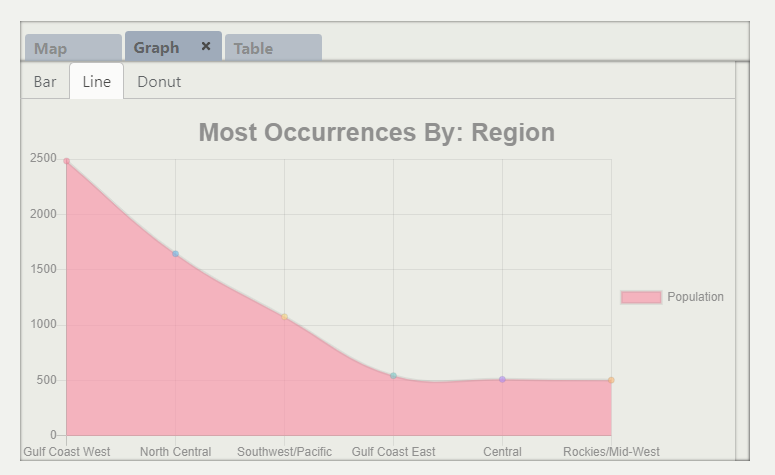
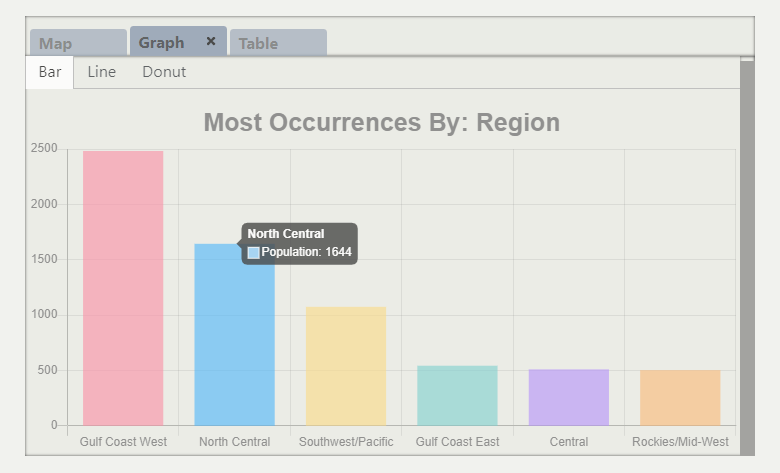
The user interface for our application is separated into input and output sections. Our goal was to design an intuitive interface so that any user could easily understand how to use our application. In order to do this, our input section features a very simple interface, including checkbox inputs, selection inputs, text input, and a search button.



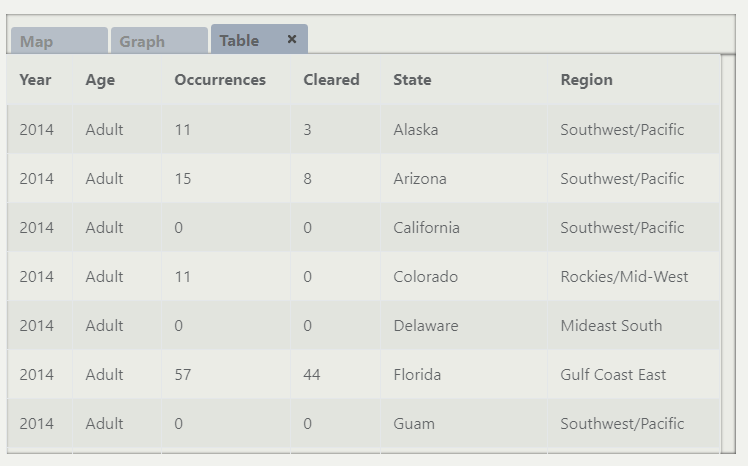
The output section of our user interface is organized into tabs. When the user clicks the search button in the input section, three new tabs will appear in the output section. Each tab will visualize the data that returned from the database query. One of these tabs will display an interactive map. The map is color-coded based on the number of human trafficking offenses that have occured in that region or state, depending on the user’s input. The user can hover over the states or regions to highlight their selection. If the user were to click on a state or region, a pop-up would appear giving the user the details of the number of offenses committed and the name of the state or region. A legend is also displayed, which gives the user a reference to each color's significance.



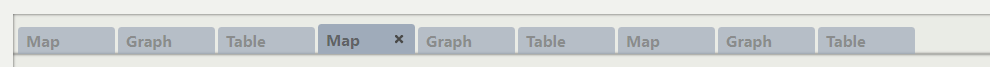
The second tab generated from a user query is an informative graph. The graph displays the same information that is shown on the map, except it is visualized in either a bar, line, or donut graph. Hovering over any of the sections of the graph will inform the user of the number of offenses committed and the name of the state or region.



The third tab generated shows a table of the raw data that was returned from the database query. We provide this information in case the user wants to see the specifics of any data point.



At any point the user can alter their inputs and send a new query to the database. Whenever this happens, three new tabs will be appended to the taskbar. The user is free to switch between any tabs they have generated.



### **Links**

### [Github Repository Link](https://github.com/JasonGilman18/Human-Trafficking-Synopsis)

### [Final Presentation](https://docs.google.com/presentation/d/1Tz1-XVqjsb_neZExVq0nTHx9G4zqrc0VYYIXtoV2QbE/edit?usp=sharing)