Charles Carlson

Wade McTaggart

**Specific Objectives and Questions**

Implement a database that uses spatial queries to allow users to view crimes in Chicago taking place near their location, popular attractions, or across their ward. Spatial queries will be made to collect crime statistics per ward in Chicago.

Potential use cases for this database:

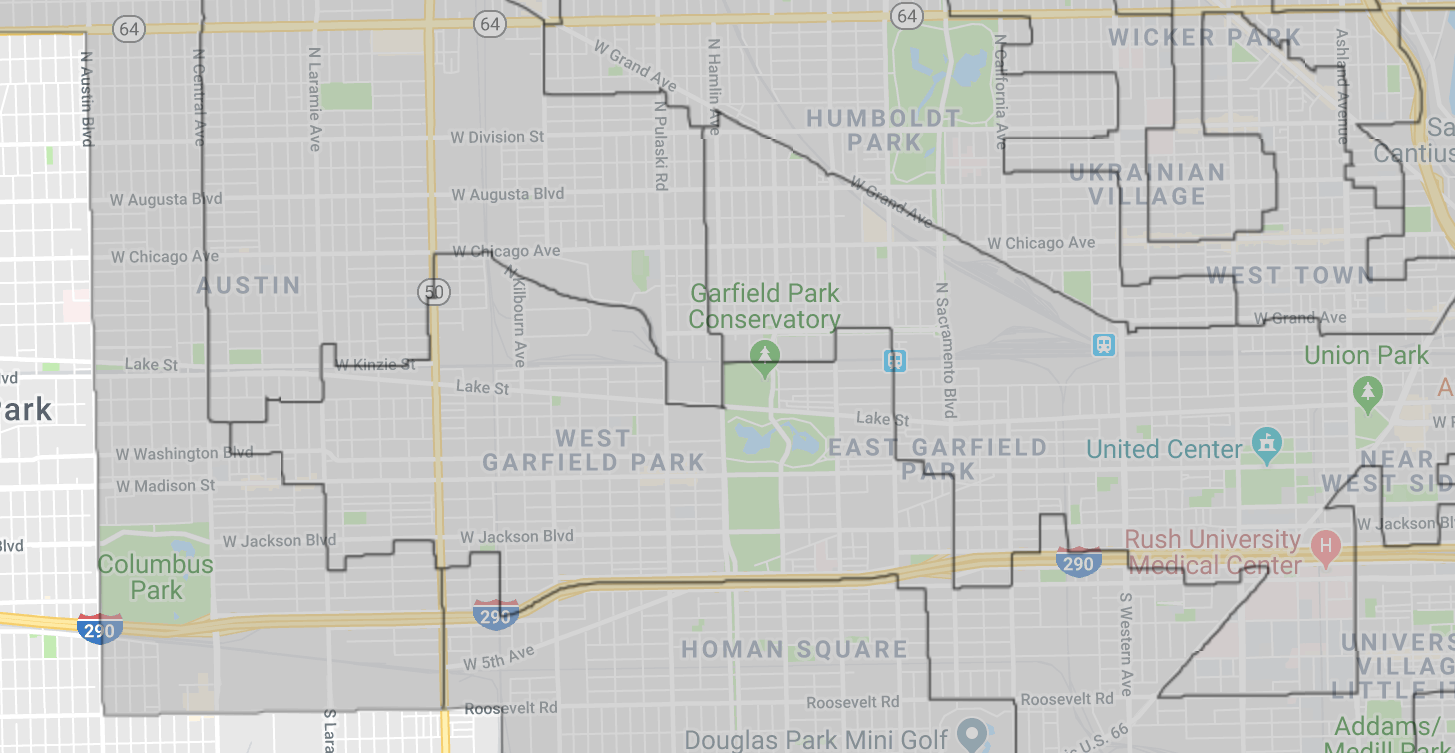
1. Which representative is responsible for the most/least crime-ridden ward.
2. Which ward has the highest/lowest crime rate?
3. How much crime is their near X location I’m planning on visiting?
4. How does my current ward rank against the other wards in Chicago in terms of crime?
5. Can we see some relationship between areas of low socio-economic status and high crime rates?

Justification of database:  
 The database is necessary because the data is not static. New crimes happen every hour, ward lines are redrawn, and representatives are elected. A database allows new information to come in in the form of a stream and allow the application to analyze it. Wards boundaries however, are defined by shapefiles. Each year the wards may be adjusted and their geometry saved in a shapefile. Since this chance is planned and infrequent, a shapefile serves our purposes well.

**Data Collection and Conceptual Design**

We have to main sources of data:

1. <https://www.kaggle.com/currie32/crimes-in-chicago>. This is a free dataset containing hundreds of thousands of crime incidents that took place in Chicago along with coordinate locations. The data includes columns such as:
   1. Date
   2. Crime Code
   3. Block
   4. Location
   5. Violent?
2. 2015 Chicago Wards Shapefile - <https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Wards-2015-/sp34-6z76>. A shapefile providing the polygons of the wards in Chicago (picture on next page)



Kaggle allows us to download the dataset as CSVs, we will be using some Python scripts to format the data so that we only get data within the year of 2015. After applying the 2015 filter we will have:

1. Crimes
2. Crime Types
3. Wards

ER DIAGRAM HERE

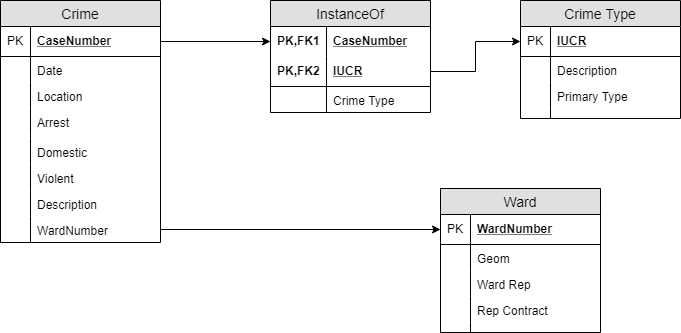
**Implementation**

Primary Keys:

1. CaseNumber - Used to uniquely identify crime rows in the Crime table
2. IUCR - Used to uniquely identify a crime type in the Crime Type table
3. Ward Number - Used to uniquely identify wards in Ward table

Foreign Keys:

1. IUCR - Foreign key to rows of InstanceOf table

****

Integrity Constraints - All data within the database is open source, therefore, all data is accessible to all users

**Application Architecture:**

The application will be a web app hosted in Heroku (free hosting service that uses git). The backend will be in Flask (Python) and the front-end will be in React (JavaScript). The Database will be a Postgres instance hosted (for free) in AWS’s RDS.

An exact mimic of the database will be hosted on Professor Chaglar’s servers, however they will only be accessible when the application is being used within the Universities’ network.

**Example Queries**

“Rank the wards based on crime rate (highest to lowest)”

Select wardRankings.WardNumber as “Ward”, wardRankings.Count as “Count”

From

(Select w.WardNumber as “Ward”, Count(crimes.CaseNumber) as “Count”

From Crime as crimes, Ward as wards

Where ST\_Within(crimes.location, wards.geom)

Group by wards.WardNumber) as wardRankings

Order by wardRankings.Count desc