1. **Executive summary**

We built a system which ranked the community areas in the city of Chicago by a calculated metric called the safety score so that we could help people interested in buying or renting to choose safe neighbourhoods.

Using the Crime Statistics data from the city of Chicago portal, we first trimmed the data, then cleaned the data and finally calculated the safety scores of the various beats and districts in the city.

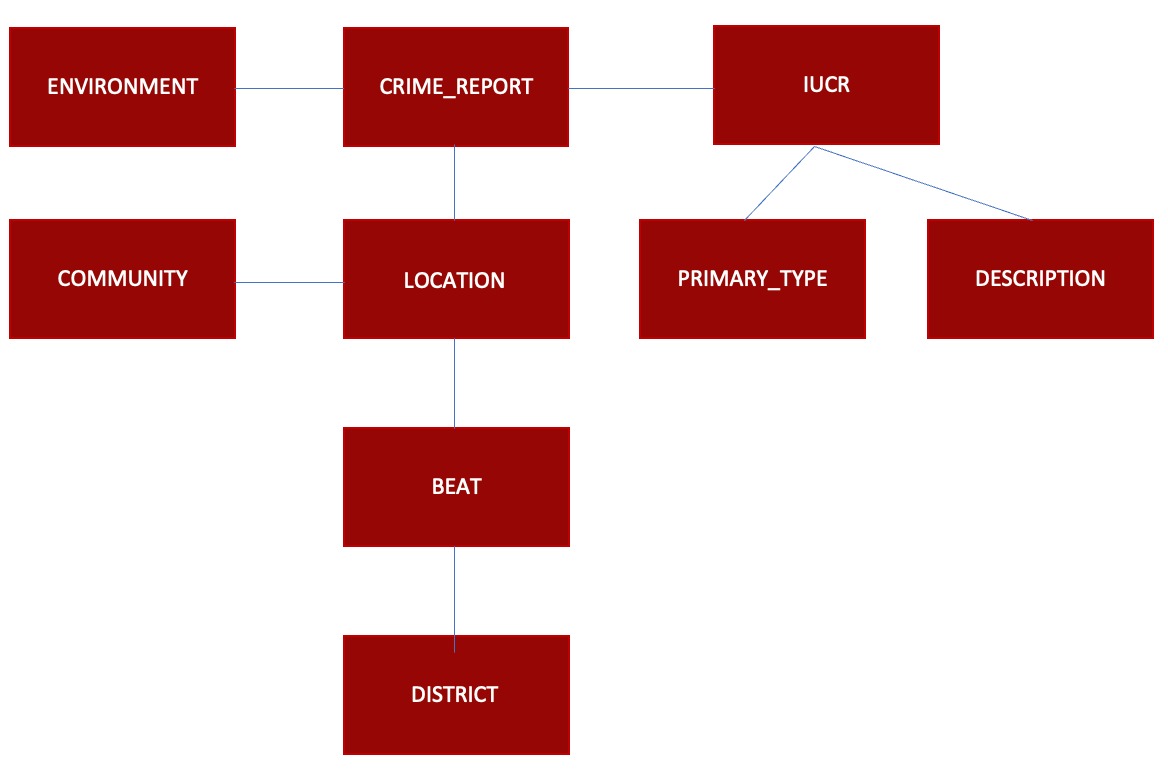
Finally using the calculated scores, we made recommendations about the various areas in the city and discussed the plans of the team on the ways we could improve our model.

1. **Business case and objective(s)**

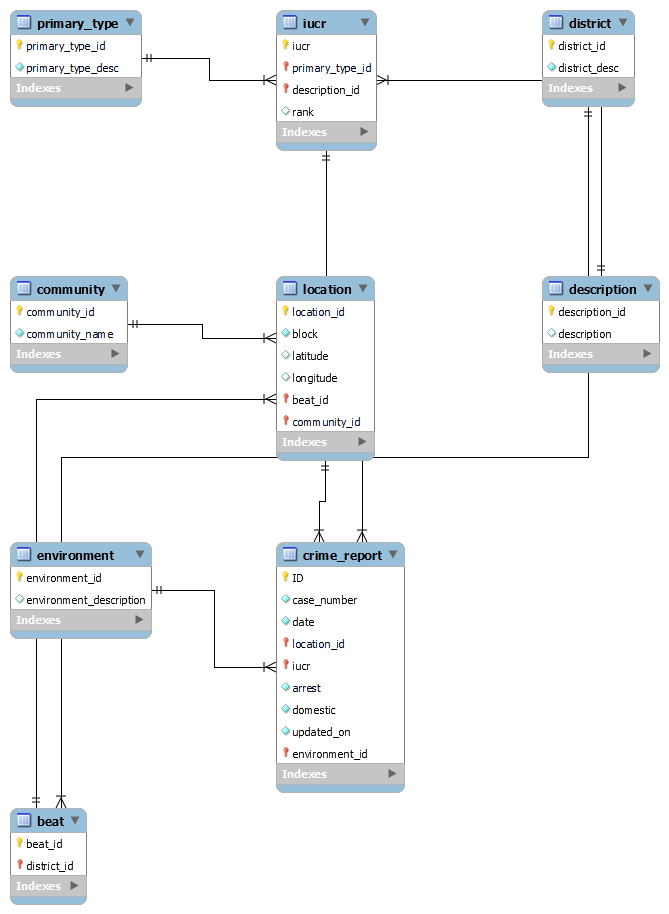
One of the major concerns while buying or renting a place to stay is safety of the area especially in high rise cities like Chicago. Our team, the Real Estate Vigilantes are building a recommendation system, that would help the residents of Chicago find their very own SAFE home.

1. Objectives:
   * Collect, prepare and analyze the crime data
   * Set up the data on Azure, connect with MySQL and Python
   * Create a model to score the areas based on severity of crime on a scale of 0-100.
2. **The problem to be solved and datasets you plan on using**
   * Calculate and map a safety score to each beat and district based on the crime.
   * The data set used was the Crime Statistics provided on the Chicago Data Portal from 2001 to present which is a record of the crimes and their details: [Crime Statistics for Chicago](https://catalog.data.gov/dataset/crimes-2001-to-present)
   * Created look up tables for Community Area and District Mapping using Wikipedia.
3. **Data Models**
4. After manually ranking the crimes in the IUCR table, they are mapped to the crimes by beat and district.
5. The sum of the rank is then normalized between 0-1 by beat and district for day and night.
6. The following formula is then used to calculate the safety score with more weightage given to day scores as compared to night scores as we believed the crimes during the day are more significant in nature like robberies which take place when an owner is generally not present at home.
7. **Conceptual, logical, and physical data models**

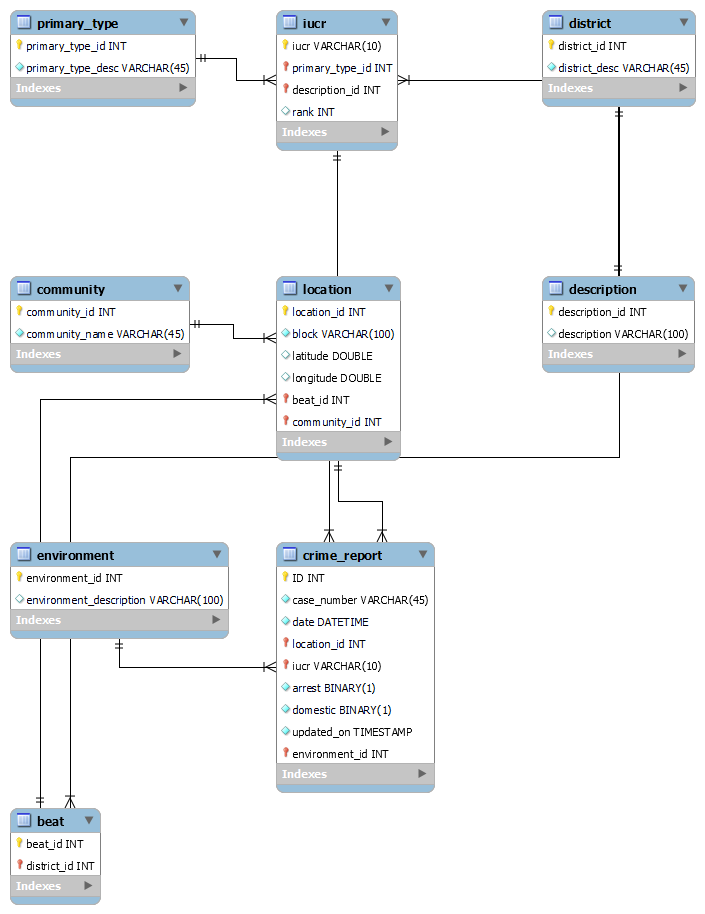
* **Conceptual model**



* **Logical model**

****

* **Physical data model**

****

1. **Relational and Dimensional, (and if applicable No-SQL/Document, and Graph), and rationale behind using one over the other**

(Relational model is provided in point 5). As the purpose of the project is to calculate the scores of the district and beat, we saw that it involved just about 2-3 joins for analysis and the performance wasn’t affected by these when compared to a dimensional model. Therefore, we decided to stick with a relational model.

1. **The model should contain at least 5 tables (more is fine), which are completely Normalized till 3rd normal (if applicable) form and ER diagram provided in the document**

ER diagram provided in point 5, under logical model which is the normalized data till 3NF.

1. **Any denormalization required in the physical model, if yes, why (provide rationale for doing so)**

In order to calculate the safety scores, the rank for the IUCR had to be mapped to every crime based on the beat and district. The following query was used:

*SELECT b.beat\_id, d.district\_id, SUM(i.rank) AS 'sum\_rank\_day' FROM iucr AS i LEFT JOIN crime\_report AS cr ON i.iucr = cr.iucr*

*LEFT JOIN location AS l ON cr.location\_id = l.location\_id*

*LEFT JOIN beat AS b ON b.beat\_id = l.beat\_id*

*LEFT JOIN district as d ON b.district\_id = d.district\_id*

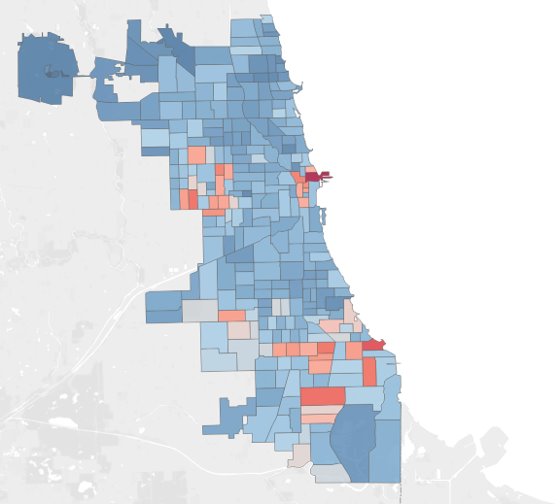
*WHERE year(cr.date) >= 2013 AND HOUR(cr.date) < 16*

*GROUP BY b.beat\_id, d.district\_id*

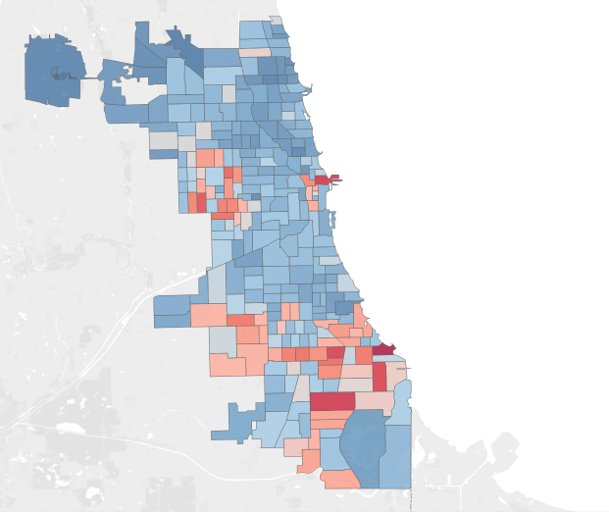
1. **Data Profiling**
   * Current and up to date crime data from the [Chicago Data Portal](https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2) refreshed and updated every 3 days. Every row in the data is a reported crime with **22 columns** and **7.69M rows** before data cleaning.
   * Created look up tables for Community Area and District Mapping using Wikipedia.
2. **Perform high level data profiling and cleaning, document any observations along with sample values such as outliers, data quality anomalies, any aggregations already applied in the data set or you plan on calculating**
   1. **DATA TRIMMING** 
      1. We chose data from 2013 to Present, as the efficacy of older data could not be verified; however, the entire data set was ingested in case the data can be repurposed later.
      2. Unnecessary and repeated columns have been removed:
         1. Ward: It wasn’t up to date with the current Chicago wards mapped out
         2. Location: As it is just a combination of latitude & Longitude, it was removed.
         3. FBI code – It was providing similar information compared to IUCR column so it was dropped.
   2. DATA CLEANING
      1. The dataset is by nature clean. However, missing values were handled by dropping the corresponding entries.
      2. The datetime variables were already formatted to MM/DD/YYYY HH:MM:SS AM/PM
3. **Methodology and various tools used in the process**
   1. Azure database – Azure was used to set up the server load the normalized data.
   2. MySQL Workbench – Used to connect with the server and run SQL commands for checking the tables and analysis.
   3. Python – Using Jupyter notebooks, the data was loaded onto the Azure server with a few commands in Python.
   4. Tableau – Create dashboards and gather insights.
4. **Evaluation and rationale behind using a certain Methodology, tool, and technology**
   1. Azure database – Since the data set is huge and is regularly updated, we would like our dashboards to reflect the same real-time data, hence we believed connecting a cloud service like Azure would provide the necessary tools required to refresh and load the database every 3 days easily.
   2. MySQL Workbench – This enabled our entire team to connect with Azure server easily. All that was required to be shared was the server link. With this, we were able to create the ER diagrams.
   3. Python – It provides multiple packages like sqlalchemy and pandas that help connect with the servers and APIs easily and manipulate this data for analysis with easy and readable commands compared to SQL. It also provided ways to upload the data with minimal commands whereas SQL would have required millions of INSERT commands and may have led to multiple human errors.
   4. Tableau – It is an easy-to-use visualization tool that directly connects with the server, allows a great range of customization thereby helping in providing better insights.
5. **Automation methodology (if any, or recommended) for the End to End pipeline.**

With the help of Python, a script is created to load the recent data onto the server every 3 days. This is done by connecting with the API provided by the Chicago data portal and downloading the latest updated information and pushing it onto the database on Azure.

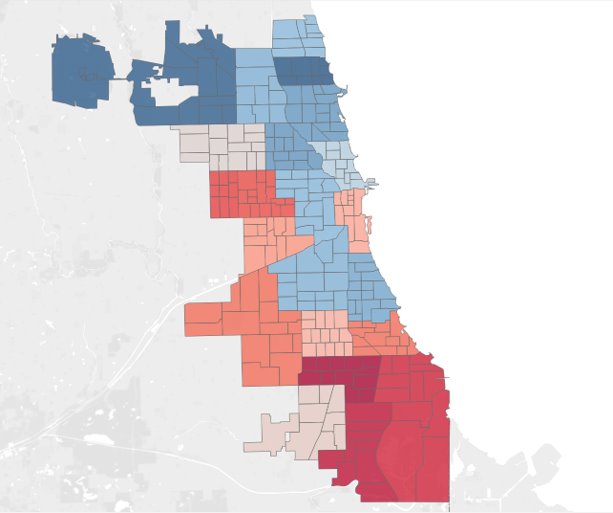
1. **Insights**
   1. Plotting by the count of crimes committed in each beat, shows an interesting map, where generally marked as “unsafe” areas like the south are safer than downtown. This method does not consider the severity of the crime, just that a crime occurred.



* 1. Further, after implementing our model and generating the safety scores, we can see the areas being marked as unsafe based on the severity of the crime.



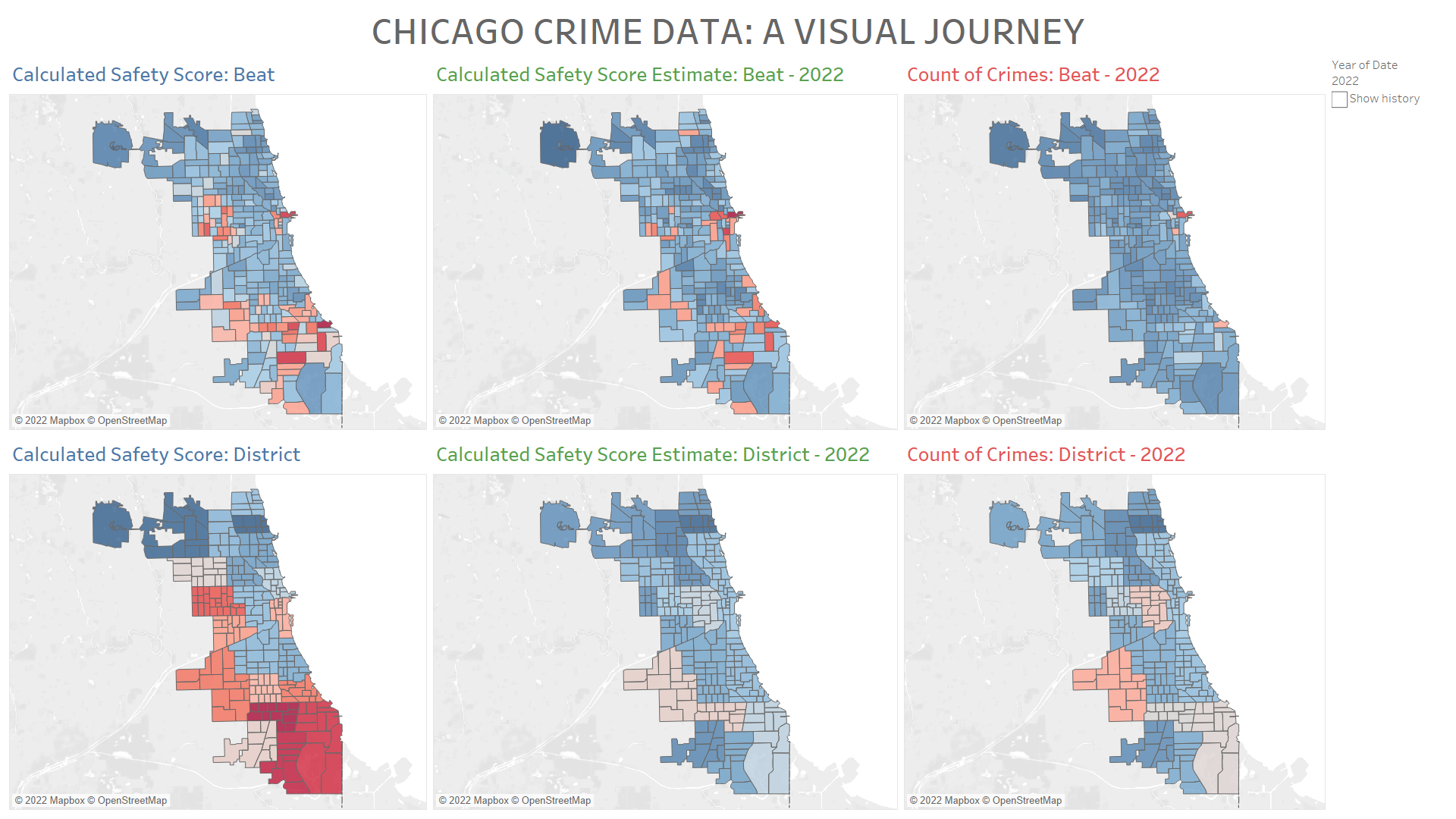
* 1. Districts 16 and 20 and the north of Chicago in general had the highest average scores, while the south of Chicago in general had abysmal safety scores.



1. **Reports and Dashboards**

Please refer point 15 for our reports.

Dashboard:



1. **Any recommendations and lessons learned**
   1. **Recommendations**
      1. Lincoln (District 20) had the Highest Safety score, and Gresham (District 6) had the Lowest Safety score.
      2. Beat no 1655 in Jefferson Park had the Highest Score and Beat no 421 in South Chicago had the Lowest Safety score.
      3. Forest Glen, Jefferson Park, O'Hare, Uptown, Lincoln Square were calculated as the best community areas to live in.
      4. South Shore, Calumet Heights, Avalon Park, Roseland and Pullman were the worst community areas to live in.
   2. **Lessons Learned**
      1. Always understand and plan the hierarchy of your data. We faced issues with modifying our data model due to incorrect assumptions.
      2. While cloud databases can come with significant deployment efforts, in the long term it provides a lot of benefits.
      3. Understanding the history of the data will help in future planning, because we did not follow the changes in Chicago’s police beat system, we had to limit our data to 2013 onwards.