San Diego Accident Analysis

Yacun Wang, Huy Trinh, Jianming Geng


Problem Statement
Data
Database Design
Visualization & Result
Future Work

Problem Statement

Main Goal:

- Improve profitability of an insurance company by understanding patterns and factors contributing to accidents in the city of San Diego.

Challenges:

- Identify the relationships between time of day, location, and accident occurrences.
- Determine if driving to a specific location at a specific time of day is risky for drivers.
- Analyze large volumes of car accident data.

Problem Statement

Approaches:

- Perform data exploration, aggregation, visualization, and correlation analysis using SQL queries in PostgreSQL.
- Utilize Neo4j for graph-based analysis to explore relationships between time of day, location, and accidents.
- Integrate the findings from both SQL and Neo4j analysis to gain a holistic understanding of accidents.

Significance + Usage

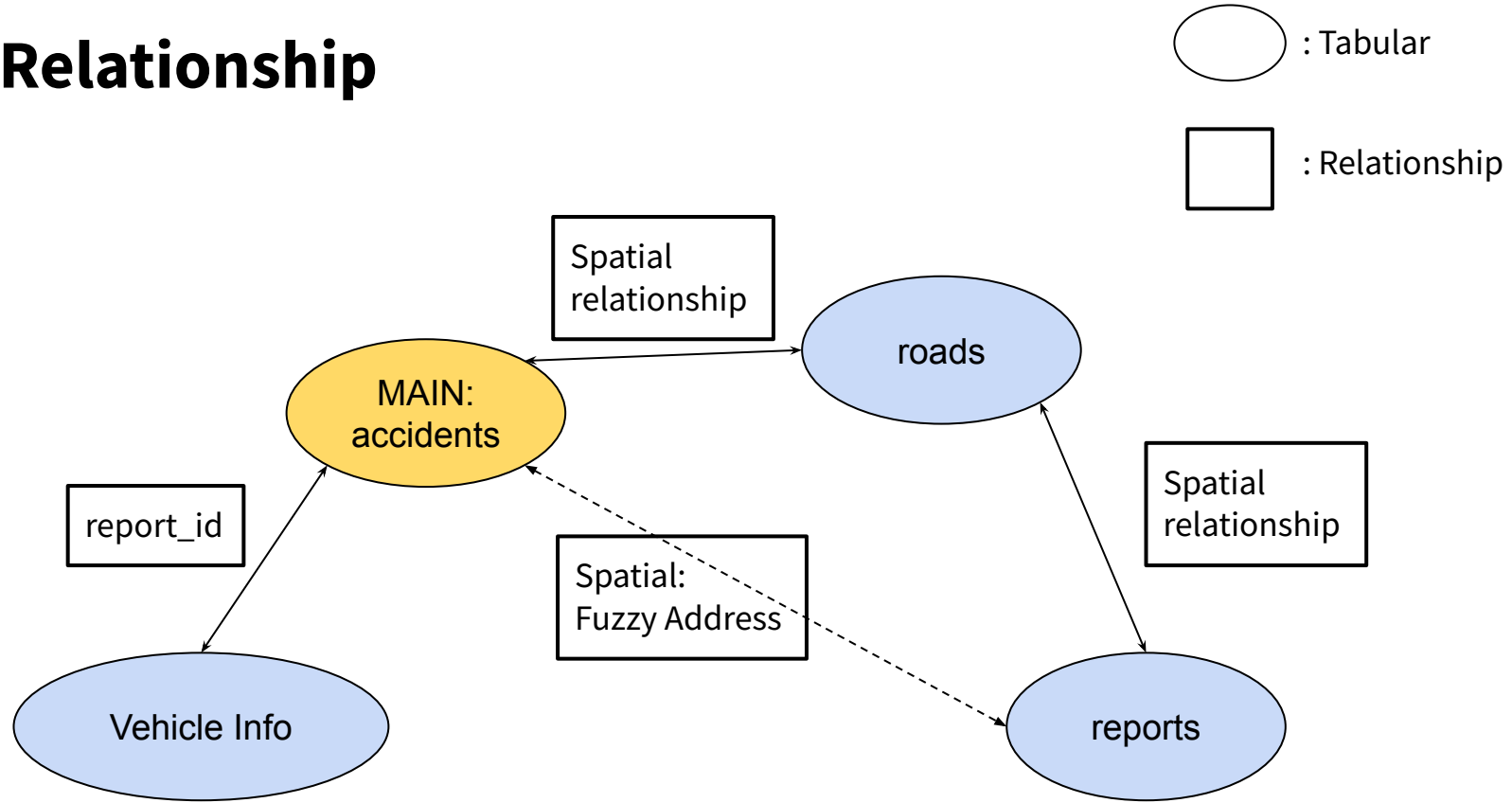
- The insurance company aims to improve its profitability by understanding the patterns and factors contributing to accidents in San Diego.
- The findings will enable the insurance company to optimize operations to achieve long-term profitability and sustainability.


Problem Statement
Data
Database Design
Visualization & Result
Future Work

Data Sources

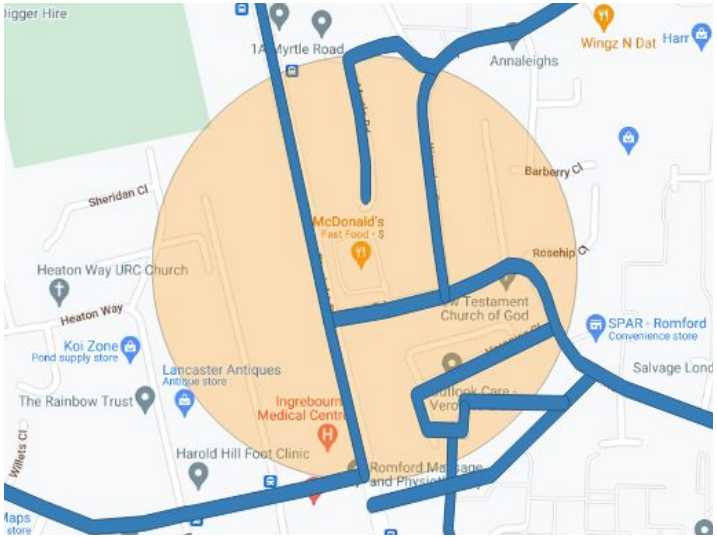
- City of San Diego Traffic Collisions
- City of San Diego Collisions People and Vehicle Involved
- Get It Done Projects
- San Diego Roads

Data Relationship



Data Preprocessing

- Geocoding
  - Get the location (lat, long) from address as POINTS
- Connect spatially close accidents/roads/reports
  - Use geocoded locations as POINTS
  - Use roads as LINES
  - Spatially Join: LINES intersect buffered POINTS
- Data Cleaning
  - Null Values
  - Column Selection
  - Time Formatting
  - etc.



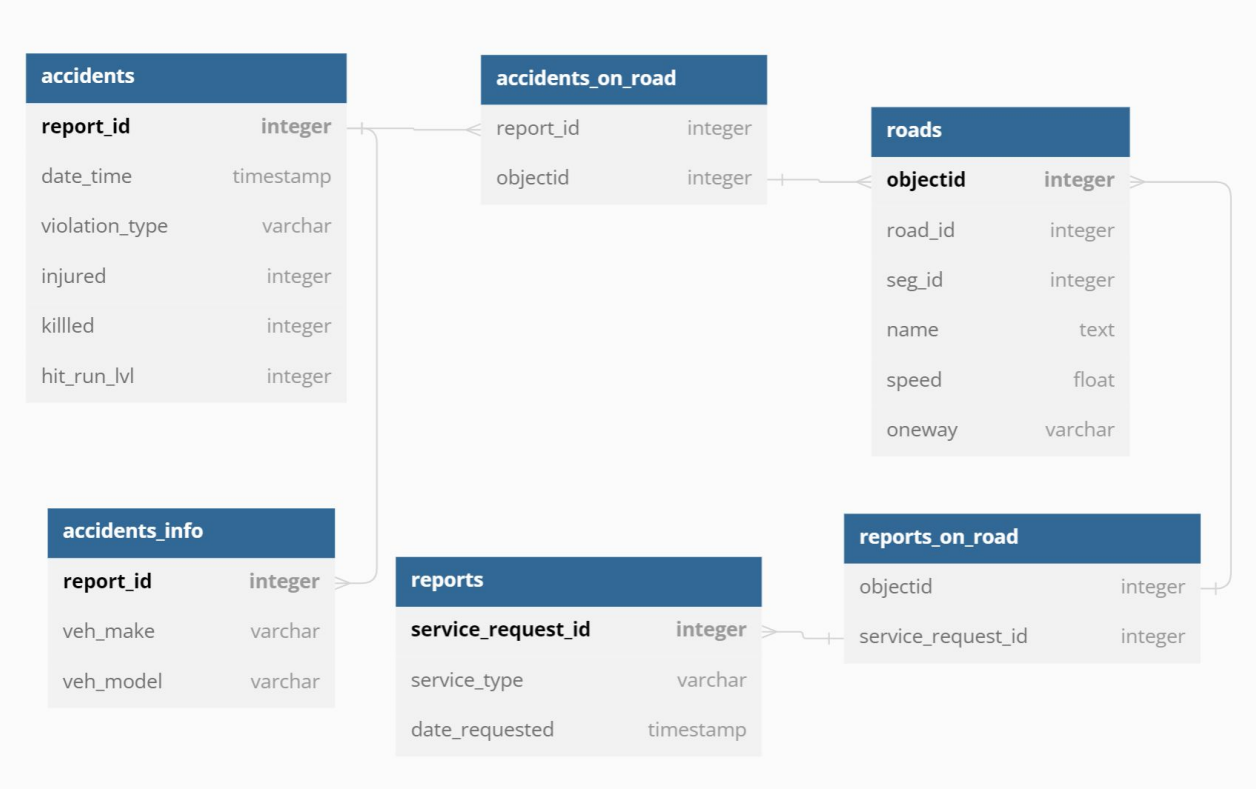
Example


Problem Statement
Data
Database Design
Visualization & Result
Future Work

PostgreSQL

- Fast on Analytics Job (we are not adding or streaming data)
- Can contain multiple columns regarding different attributes
- Row-wise storage suits our purpose of querying across many columns as well as different tables
- Data are simple and easy to store in SQL
- Usage: We put all six tables to a new SQL database on the server

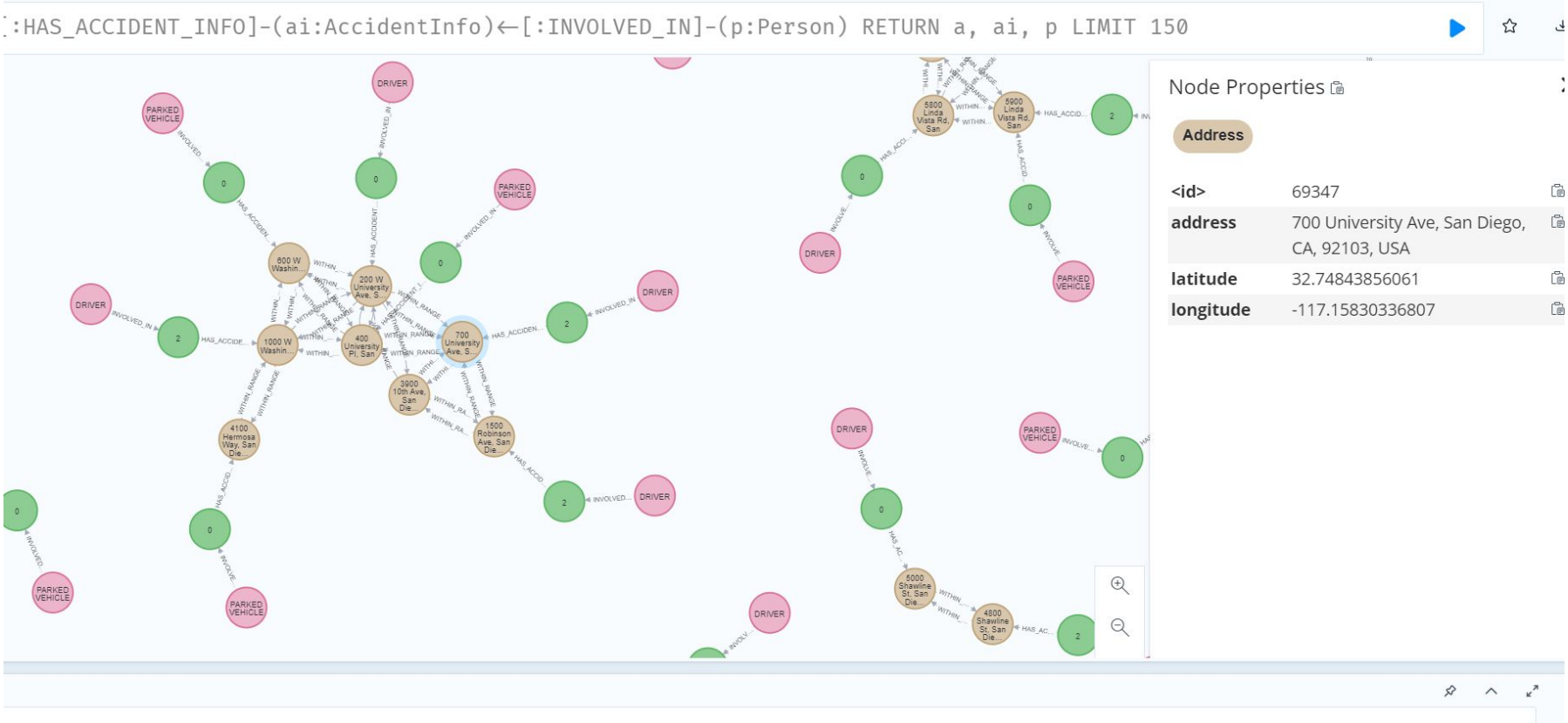
Relational Schema



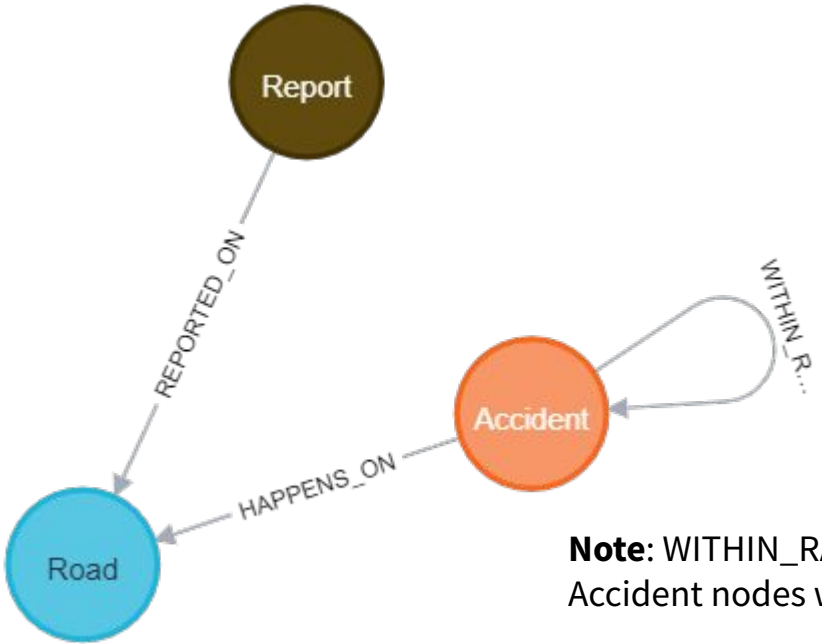
Neo4J Graphs

- Easy to explore highly interconnected and relationship-driven data
- Flexible modeling and more efficient querying of complex relationships than relational joins
- We will explore the road to accident, road to vehicle, accident to accident relationships using Neo4J

Attempted Graph schema

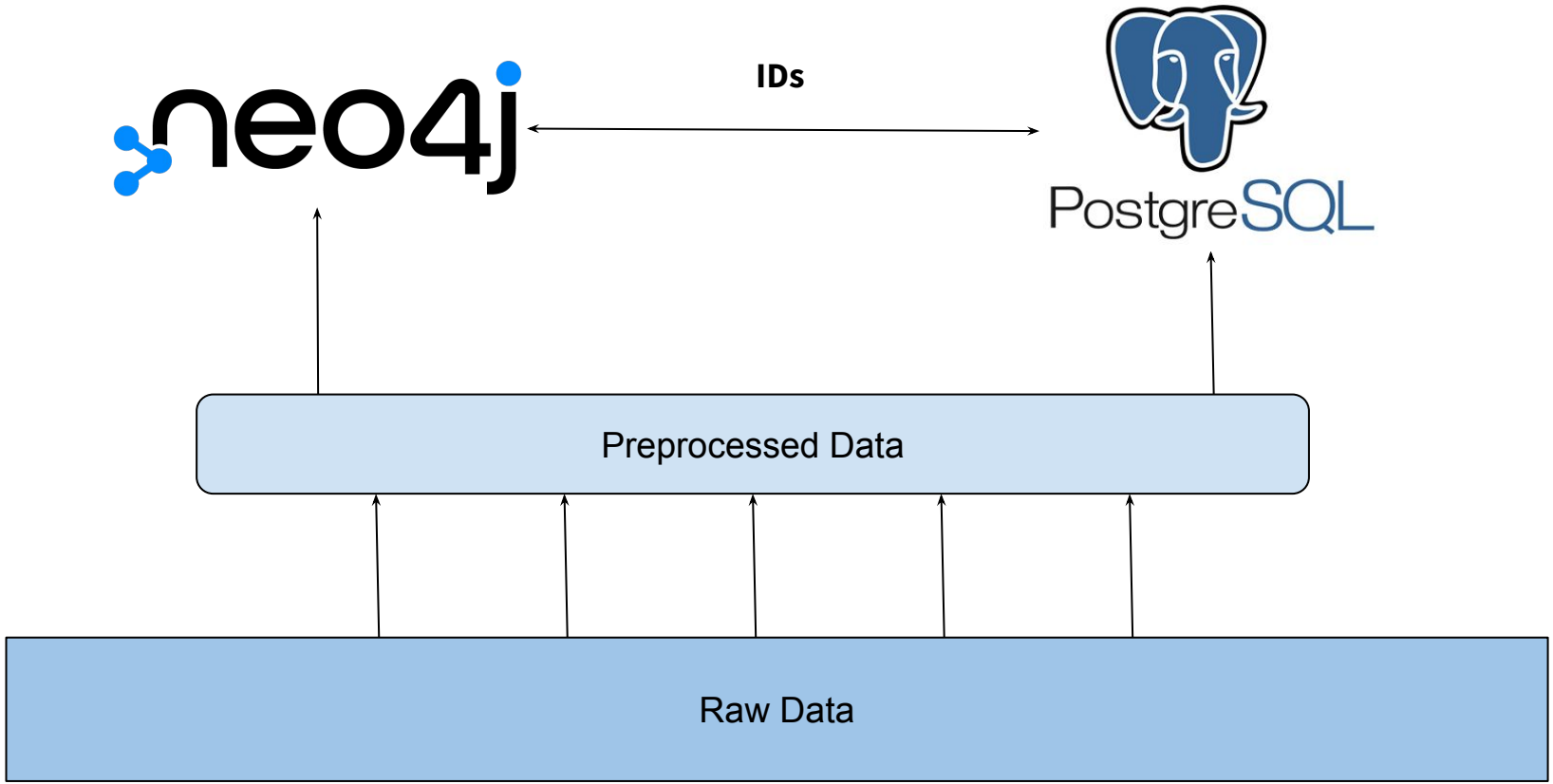


Graph Schema



Note: WITHIN_RANGE exists between distinct Accident nodes when the distance is <= 500 meters

Integrated Both




Problem Statement
Data
Database Design
Result & Visualization
Future Work

Result Types

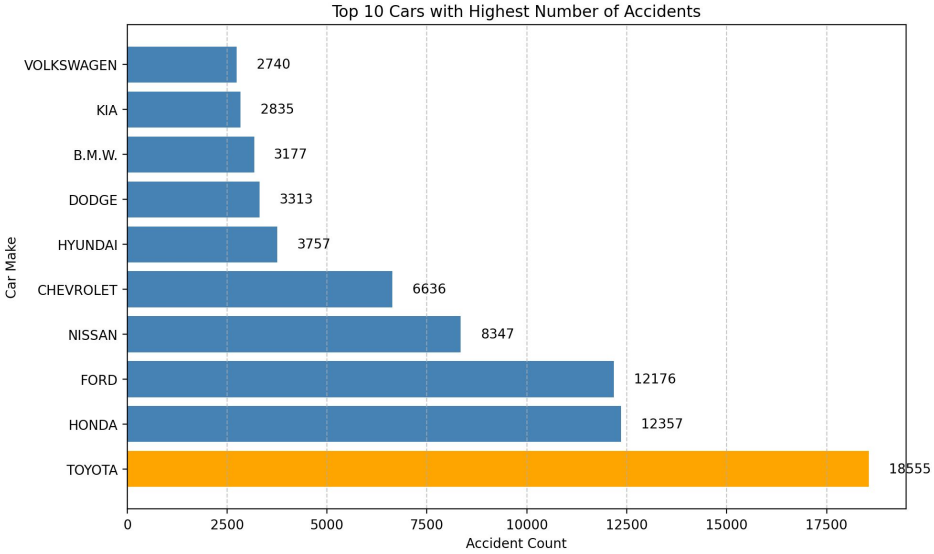
As suggested by the Relational Schema, separated into 4 parts:

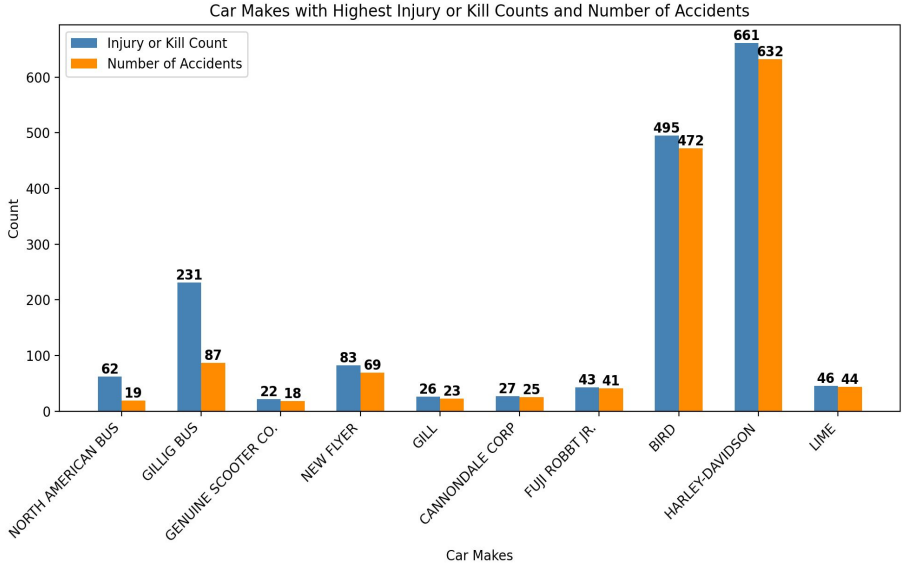
1. Accident Information
2. Accident Vehicle Information
3. Accident Road Information
4. Accident Information in Relation to Get-It-Done Reports

Accident Locations (Community)

- {'num_accidents': 695, 'community': 'City Heights'}
- {'num_accidents': 561, 'community': "Banker's Hill"}
- {'num_accidents': 500, 'community': 'Logan Heights'}

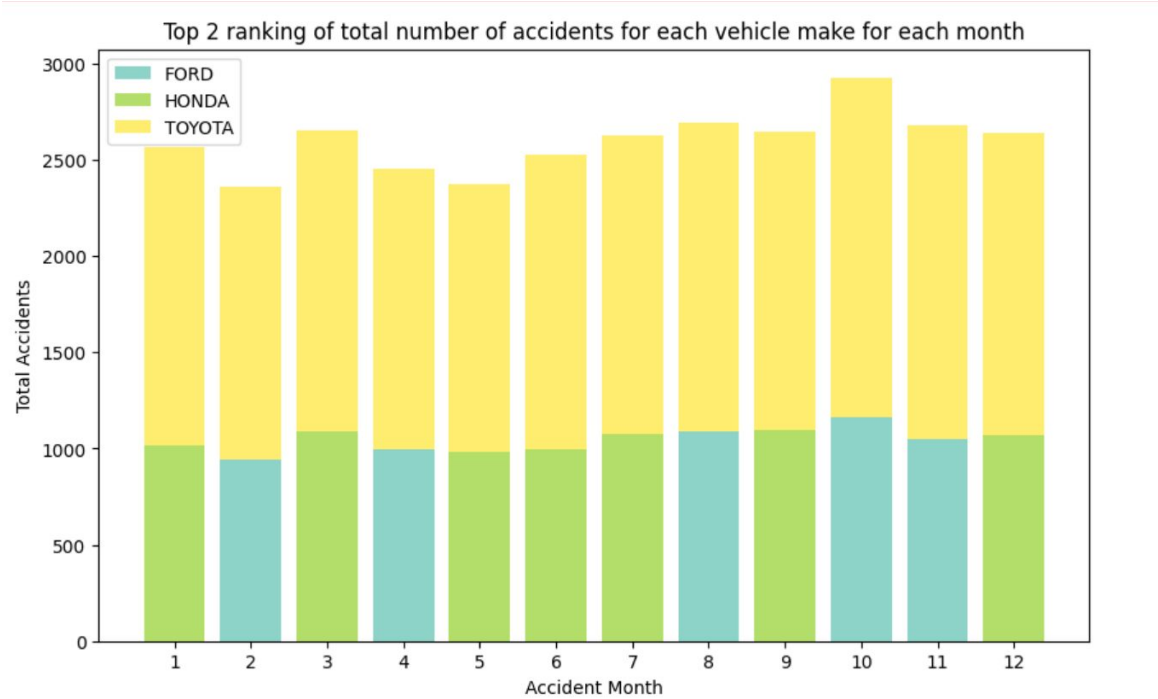
Vehicles and Hurt





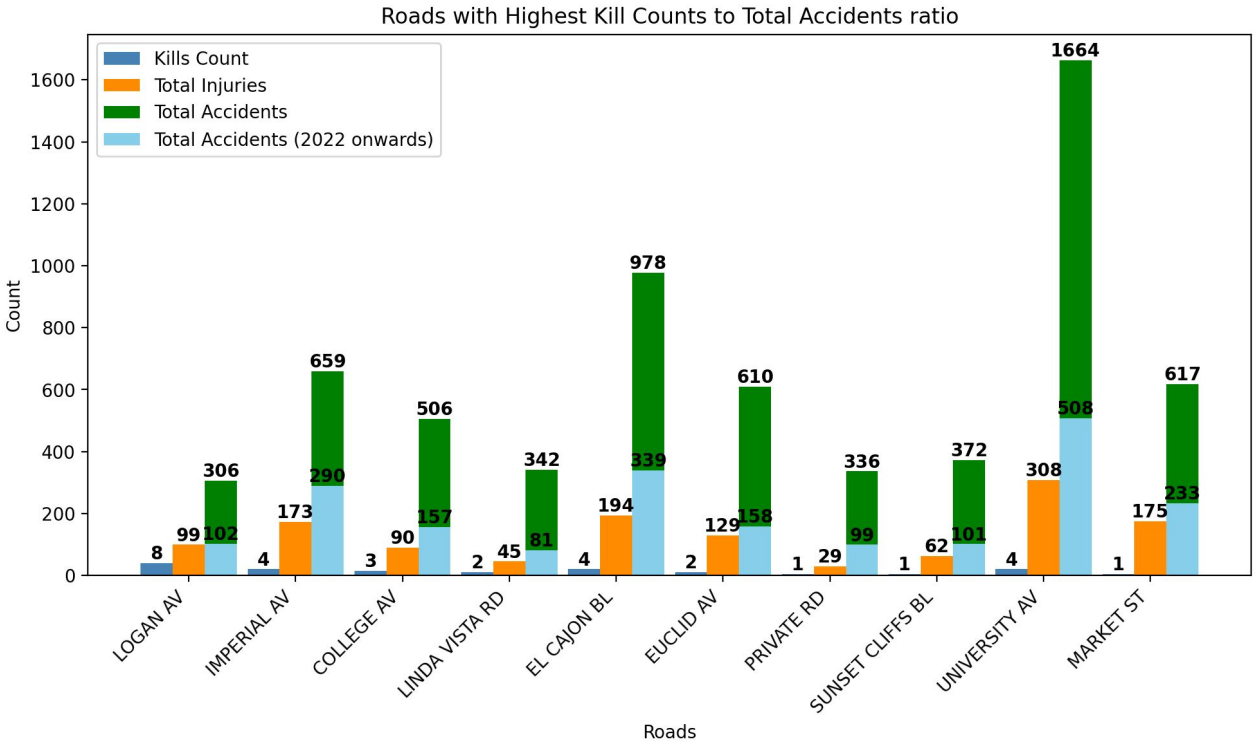
Key Takeaway: While cars contribute to the majority of accidents overall, it is important to note that incidents involving buses and scooters result in a higher number of injuries and fatalities.

Accidents Per Month Per Car Make



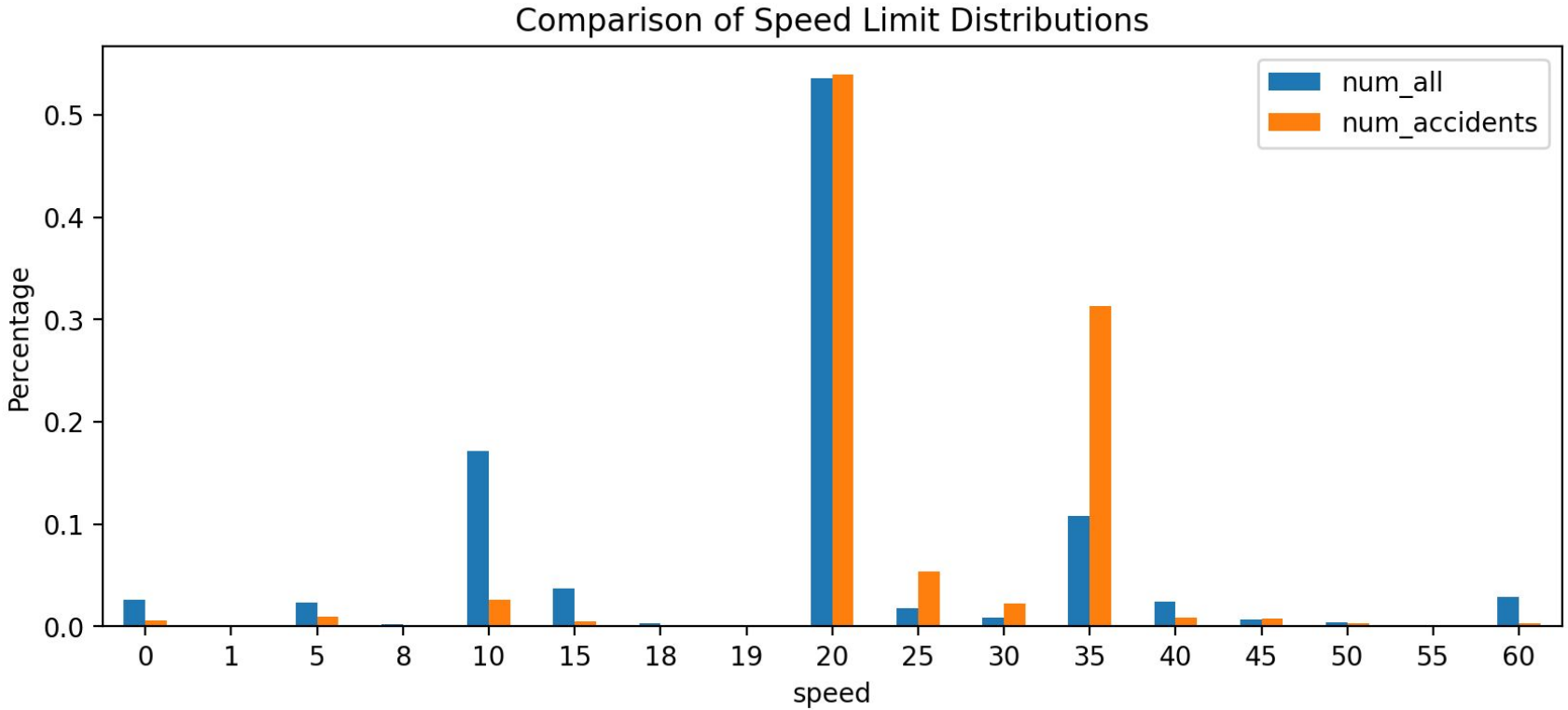
Key Takeaway: seasonal influences on accident occurrences and the vehicle makes most affected during specific times of the year.

Roads with Most Severe Accidents

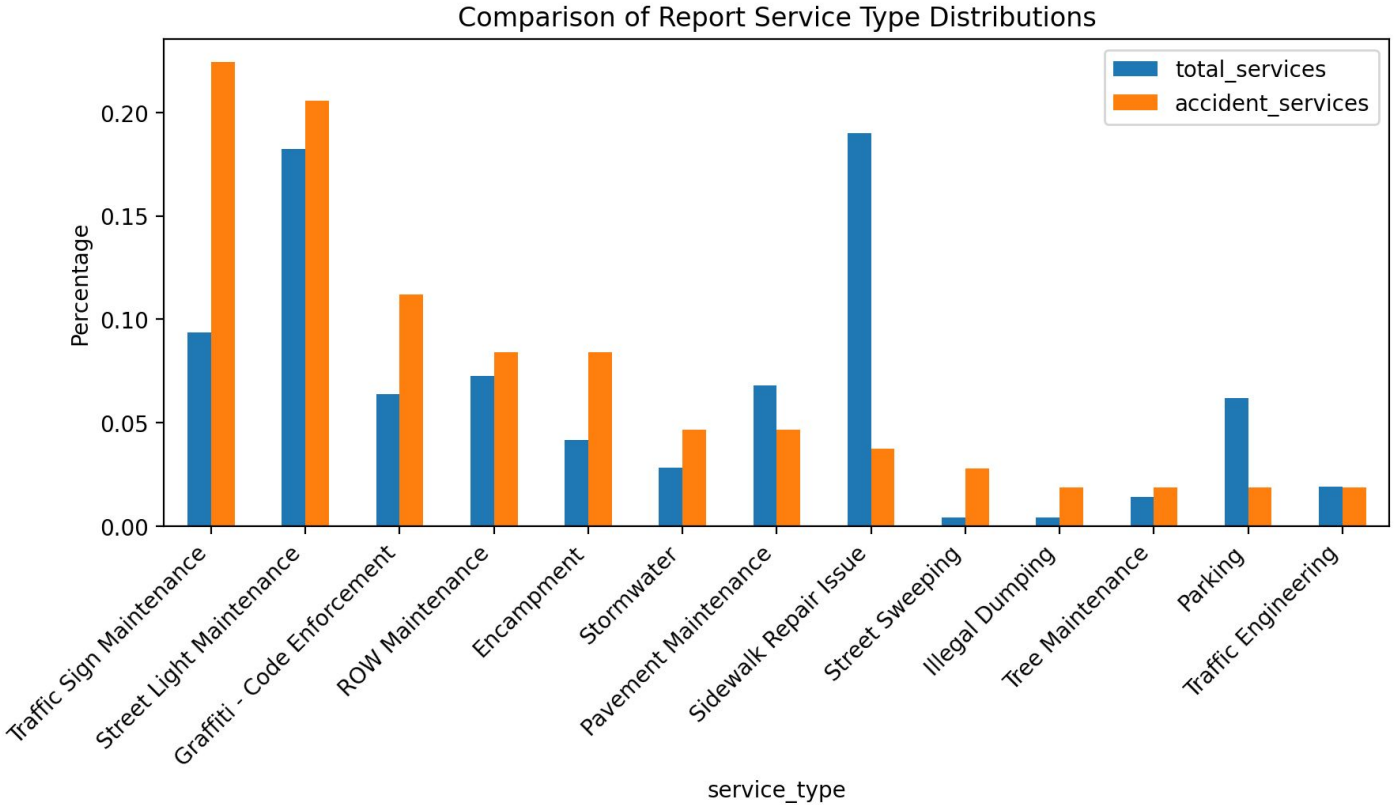


Key Takeaway: The data reveals the roads with the highest kill-to-accident ratios, indicating a higher likelihood of fatal accidents.

Accidents and Speed Limit



Report Types on Risky Roads




Problem Statement
Data
Database Design
Visualization & Result
Future Work

Future Work

- Involve text data: Extracting insightful key words from report descriptions to help with identifying report severity (Apache Solr)
- Involve more street/road features: Consider the effects of different street features and conditions on accidents (Cassandra)
- Cache frequent queries (Redis)

Thank you!

Github Link: https://github.com/jgeng99/San-Diego-Accident-Analysis

Data Sources: See Slide on Data Sources