Analysis of Road Traffic Safety and Congestion Using Big Data

A Project Report
Presented to
DATA-228
Fall, 2023

By:
ShashiKumar K. Mallikarjuna
Mahamaya Panda
Aradhya A. Rathnakar
Bhavan K. Basavaraju
ReddySaketh R. Chappidi
Date: December 10, 2023

ABSTRACT

Analysis of Road Traffic Safety and Congestion Using Big Data

By:
Shashi Kumar Kadari Mallikarjuna
Mahamaya Panda
Aradhya Alva Rathnakar
Bhavan Kumar Basavaraju
ReddySaketh R. Chappidi

The modern issue of escalating traffic challenges demands a more informed approach to effective solutions. This project aims to closely examine road traffic details in Chicago, focusing on utilizing crash data to provide a comprehensive view of traffic safety and congestion. The crash data encompasses information on every traffic crash within the City of Chicago under the jurisdiction of the Chicago Police Department (CPD). The data, sourced from the electronic crash reporting system (E-Crash) at CPD, is displayed on the data portal without personally identifiable information. Records are added to the portal upon finalization of a crash report or amendments to an existing one in E-Crash. To achieve project objectives, various Amazon web service (AWS) cloud resources, including Amazon Simple Storage Service (S3) buckets, Glue, Redshift, and Athena, are deployed to enable efficient data staging and processing through file drops and Application programming interface (API) integration. The data undergoes meticulous processing to ensure quality and completeness, resulting in a dataset available for exploration and analysis. A user-friendly front-end application, powered by AWS QuickSight offers stakeholders an intuitive interface to explore and visualize street-level information. The application's rich features allow stakeholders to interact with the data, extracting insights relevant to their needs. The project's outcomes are highly relevant for urban planning and public safety. By leveraging historical and real-time data, the project provides a valuable resource for decision-makers to inform choices about traffic management, road safety, and urban planning. The project's approach and outcomes can benefit businesses, academics, and practitioners in transportation and urban planning, offering a framework for addressing road traffic safety in cities.

ACKNOWLEDGEMENT

Sincere appreciation is extended to Professor Andrew Bond and the Teaching Assistant Venkatasai Rao Dheekonda for entrusting the opportunity to work on the project Analysis of Road Traffic Safety and Congestion Using Big Data. Their continuous guidance and unwavering support throughout the duration of the project have been invaluable, and the wealth of knowledge and skills acquired during the course of research is highly acknowledged. Thank you for this opportunity.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Project goals and objectives

Project goals

The project aims to address critical issues in Chicago's road traffic safety and management by leveraging data-driven approaches. The objectives include:

1. **Enhance Road Traffic Safety.** Creating effective data-driven policies aimed at enhancing the road traffic safety of Chicago. Analyze these patterns and propose measures that will reduce the frequency and intensity of road accidents.
2. **Mitigate Traffic Congestion.** To make use of historical, and onsite traffic data and identify patterns of traffic congestion. Adopt smart traffic management techniques aimed at enhancing the flow of traffic and decongesting crucial locations.
3. **Inform Urban Infrastructure Planning.** Present actionable insights that will underpin urban infrastructure decision-making. Efficiently catering to rising city infrastructure needs to enhance road traffic management.

Objectives

The following objectives are proposed to address the challenges of road traffic safety and congestion in Chicago:

1. **Collect and Integrate Comprehensive Data**. Aggregation and integration of multiple datasets such as traffic crashes, road closures, red light violations, and speed violations from the Chicago Data Portal.
2. **Utilize Big Data Analytics**. Leverage an advanced big data analytics approach to analyze the data and extrapolate road traffic-relevant patterns that involve crashes.
3. **Implement Real-Time Interventions**. Establish an instantaneous approach to responding to incidents utilizing intelligence derived from crashes for improved live traffic management.
4. **Optimize Urban Infrastructure**. Employ analytics information in adjusting the existing city’s traffic patterns and urban infrastructure for optimal performance.
5. **Improve Commuting Experience**. Ensure road safety and reduce crash-related delays to improve the general commuting experience among residents. Ensure traffic flows are maintained with minimal inconvenience, reducing transport network shutdowns.
6. **Contribute to Public Safety Efforts**. Contribute useful information for urban planning, and public safety projects by offering insight into issues around traffic safety and congestion as well.

1.2 Problem and motivation

Problem Statement

The urban landscape of Chicago suffers because of the population expansion and consequent increased road traffic with escalated accident rates. Traffic congestion, crashes, and miserable commutes are what locals must endure due to deficiencies in existing transport policies and measures. Contemporary cities are much more complicated than they were before, thus, traditional means are no longer sufficient for improving road safety or regulating traffic. Therefore, an advanced data-driven option is required in place of this.

Motivation

The motivation comes up because of a necessity to overhaul how urban traffic issues are addressed. Traditional techniques of traffic management are no longer effective in the era of an ever-growing population and change in vehicle and pedestrian movements. The project focuses on using the power of analysis of a large amount of data for a deep understanding of accidents, traffic, and jams which aim at making more secure and effective roads. Through this, it seeks to make meaningful contributions to urban planning, public safety, and commuting on Chicago streets.

1.3 Project application and impact

Project Application

The project's primary application lies in the development of a real-time dashboard that offers comprehensive insights into crucial aspects of traffic management, including crashes, traffic jams, and related events. This dynamic and responsive application harnesses the robust data processing capabilities provided by AWS. The project strategically employs tools like S3 for scalable storage, Glue and Glue crawlers for efficient data processing, Redshift for powerful querying, and Athena for interactive analysis. This orchestration of AWS services culminates in visually intuitive and interactive dashboards developed using AWS QuickSight, providing stakeholders with a holistic and immediate understanding of the city's traffic dynamics. Through the integration of these AWS tools, the project ensures not only the real-time processing and analysis of traffic data but also the seamless visualization of key metrics. The result is a user-friendly dashboard that empowers decision-makers, city planners, and other stakeholders to make informed choices promptly. This application serves as a powerful resource for enhancing the overall efficiency of traffic management in Chicago, offering a comprehensive and real-time view of traffic-related events for more effective decision-making and intervention strategies.

Project Impacts

Academic Impact. The project contributes to academia by showcasing the practical application of advanced big data analytics techniques in real-world scenarios. It serves as a case study for incorporating cloud resources for efficient data processing and visualization, providing a valuable learning resource for students and researchers.

Industry Impact. In the industry, the project has a significant impact on the field of urban planning, transportation management, and data analytics. The real-time dashboard offers industry professionals a powerful tool for making informed decisions, optimizing traffic management, and planning infrastructure developments based on real-world insights.

Societal Impact. The societal impact is profound, directly addressing the safety and commuting experience of residents. By providing real-time analysis of crashes and traffic jams, the project contributes to reducing accidents and improving overall road safety. This, in turn, leads to a more efficient and reliable commuting experience, positively influencing the daily lives of the community.

Urban Planning and Infrastructure Development. The insights generated by the project have a direct impact on urban planning and infrastructure development. City planners can use the data to identify high-risk areas, optimize traffic flow, and plan infrastructure projects more effectively, thus contributing to the sustainable development of urban environments.

Public Safety Efforts. The project aligns with public safety efforts by providing a comprehensive view of road safety and congestion patterns. Emergency services can utilize the real-time dashboard to respond more effectively to incidents, reducing response times and improving overall public safety.

Environmental Impact. The reduction in traffic jams and crashes leads to a more efficient flow of traffic, subsequently reducing fuel consumption and emissions. This environmental impact aligns with broader sustainability goals by contributing to a decrease in pollution and carbon footprint associated with vehicular delays and accidents.

1.4 Project results and deliverables

Project Results

The key result of the project "Analysis of Road Traffic Safety and Crashes Using Big Data" is the development of a real-time dashboard system. This system provides comprehensive analyses of various traffic-related aspects, including crashes, traffic jams, and related incidents. The system is underpinned by the robust data processing capabilities of AWS, integrating tools such as S3, Glue, Glue Crawler, Redshift, and Athena. The visualization component is powered by AWS QuickSight, offering an

intuitive and interactive interface for users to explore and understand the traffic crash data of Chicago.

Project Deliverables

Real-Time Dashboard System. The primary deliverable is a fully functional real-time dashboard system. This system will be accessible to stakeholders and users, providing dynamic visualizations and analyses of traffic safety and congestion patterns in Chicago.

Comprehensive Report. A detailed report outlining the project methodology, data sources, analysis techniques, and findings. This report will serve as a comprehensive document for academic and industry reference, detailing the project's processes and outcomes.

Codebase and Documentation. The project codebase, including scripts for data processing, analytics, and dashboard development, will be provided. Accompanying documentation will offer insights into the code structure, dependencies, and usage guidelines for future reference and potential extensions.

User Documentation. Documentation tailored for end-users, providing guidance on navigating the real-time dashboard, understanding visualizations, and extracting meaningful insights. This will enhance the usability of the system for stakeholders.

Visual Reports and Presentations. Visual reports generated by the system, as well as presentation materials summarizing key findings and insights. These materials will facilitate effective communication of project outcomes to stakeholders, including city planners, transportation authorities, and the general public.

Demonstration Prototype. A demonstration prototype showcasing the functionality of the real-time dashboard. This prototype will offer an interactive experience, allowing stakeholders to explore the features and capabilities of the system.

Cloud Resource Configuration Guidelines. Guidelines and documentation on the configuration and setup of AWS cloud resources used in the project. This will assist in replicating and scaling the project infrastructure for similar applications.

Training Materials. Materials for training users and administrators on the operation and maintenance of the real-time dashboard system. This will empower stakeholders to effectively utilize and manage the system.

1.5 Market Research

The growing adoption of advanced safety features in mainstream vehicle segments is expected to drive growth for companies like Mobileye and Velodyne which are at the forefront of computer vision and environment perception technology development. The detailed market research can be seen in the below table.

Table 1
Market Research

Company	Offerings	Partnerships	Market Share
Mobileye	Collision Avoidance System Lane Keeping Assist Adaptive Cruise Control	BMW, Audi, Honda, Nissan	40% in vision-based assisted driving segment
Velodyne Lidar	Alpha Prime Lidar Velabit Lidar Vella Development Kit	Ford, Mercedes-Benz, Hyundai	25% in automotive lidar market

The traffic data analytics project focused on Chicago can potentially benefit automotive safety technology companies like Mobileye and Velodyne Lidar in the following ways:

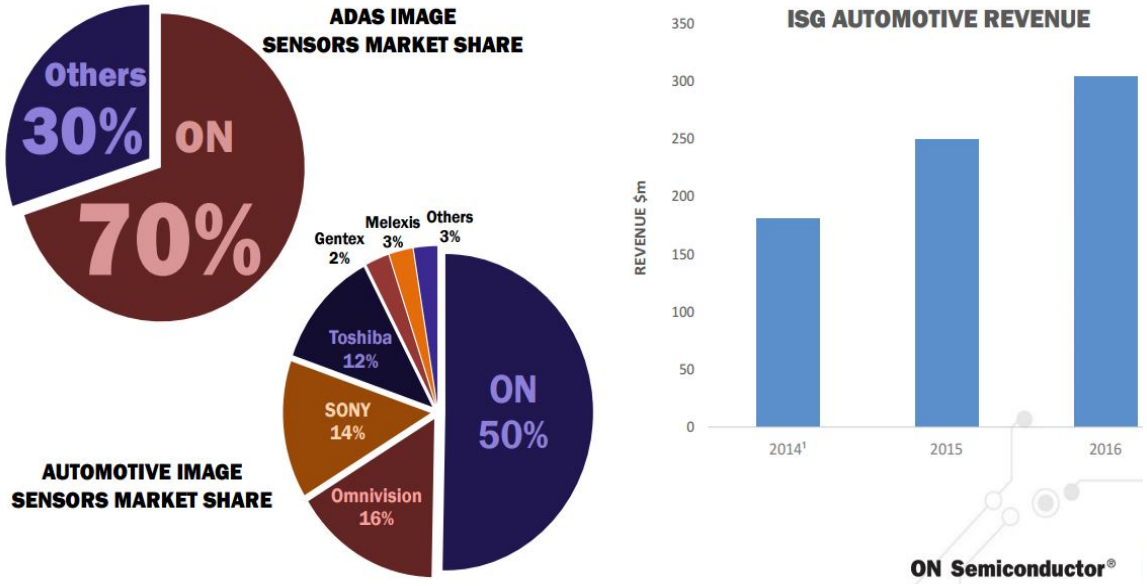
1. **Local Insights.** The granular analysis of crash data patterns, congestion hotspots, and road safety issues for Chicago provides highly relevant local insights. This can help shape safety technologies to address region-specific risks.
2. **Validation Data.** The multi-year crash datasets act as vital validation benchmarks for self-driving and driver assistance systems. Safety improvements can be quantified by comparing before/after datasets.
3. **Infrastructure Mapping.** Detailed traffic infrastructure maps are critical for autonomous navigation. The project will enrich routing, HD mapping, and location data keys for AV testing.
4. **Edge Case Mining.** Rare corner crash cases can be mined to improve the handling of edge scenarios and enhance technology resilience.

5. **Partnership Opportunities.** The traffic analytics platform connects datasets across agencies and infrastructure. Possibilities for further data-sharing partnerships.

This project fulfills crucial data needs - localization, infrastructure intelligence, and continuous validation benchmarks to facilitate the development and deployment of cutting-edge vehicle safety systems for the Chicago area. Tight integration of traffic analytics with emerging automotive tech stacks is mutually beneficial.

Figure 1

Integrating Traffic Analytics for Advanced Infrastructure Intelligence in Chicago



1.6 Project Report Structure

The report begins with an Abstract and Acknowledgement section, followed by a Table of Contents. Chapter 1 introduces the project, including goals, objectives, problem statement, motivation, applications, impacts, results, deliverables, and market research. Chapter 2 discusses the project background, related work, and state-of-the-art technologies and includes a literature survey. Chapter 3 defines the system requirements through a requirements analysis. Chapter 4 describes the system design covering architecture, data/database, interfaces, connectivity, and any design problems/solutions. Chapter 5 summarizes the system implementation, issues faced, and technologies used. Chapter 6 scopes the testing and experiments, and provides results. Finally, Chapter 7

concludes the report with a project summary and discusses future work and improvements.

Chapter 2 Background and Related Work

2.1 Background and Used Technologies

Chicago's urban landscape is negatively impacted by the city's growing population, which increases traffic on the roads and raises the risk of accidents. Due to the shortcomings of the current transportation regulations and procedures, residents must put up with traffic congestion, collisions, and dreadful commutes. Because modern cities are far more intricate than they were in the past, traditional methods of enhancing traffic control and road safety are no longer adequate. Therefore, this must be replaced with an improved data-driven choice.

Data is collected from the Chicago Data Portal and staged inside the AWS S3 bucket. AWS services are used for data processing, data cleansing, data transformation, and visualization. Raw data is converted into data format which is suitable for making suitable analysis. Knowledge of multiple AWS technologies is employed in this project.

Used Technologies

AWS Glue. It is used for running spark scripts to perform data cleansing, transformation, enrichment, and aggregation.

Apache Spark. Apache Spark is used inside AWS Glue jobs for data transformation and to handle large datasets.

AWS Crawler. AWS Crawler connects to the S3 bucket, extracts the metadata, and stores it in a new crawler database.

Amazon S3. It is used for storage purposes. Raw and transformed datasets are stored in it.

AWS Athena. Querying and testing data that is stored in the S3 bucket is done by using AWS Athena.

AWS Redshift. AWS Redshift offers fast query performance and is used as a data warehouse for loading and extracting data.

AWS Quicksight. It is a cloud-based tool that is used for analysis using visualizations.

2.2 State-of-the-art Technologies

Enormous amounts of traffic data are mined by sophisticated computing techniques to provide insights not readily apparent to the human eye, such as the ability to forecast high-risk road stretches, model the sequence of events leading to frequent incidents, and pinpoint systemic trends that underlie accident hotspots.

Data Platforms. The system harnesses a trifecta of advanced traffic analytics solutions, including Inrix, Waycare, and Streetlight Data, to comprehensively address the multifaceted aspects of traffic management and safety. Inrix stands out for its adeptness in gathering and analyzing linked vehicle and probe data, delivering nuanced insights into traffic dynamics. Waycare, a sophisticated traffic management system, seamlessly integrates data from diverse sources, encompassing satellites, lidar, and cameras, providing a holistic view of traffic conditions. Streetlight Data specializes in linked car safety analytics and mobility indicators, adding a layer of depth to the system's analytical capabilities. This strategic amalgamation of cutting-edge technologies ensures a robust and well-rounded approach to traffic analytics, enhancing the system's ability to derive meaningful insights for informed decision-making in traffic management and safety initiatives.

Analytics Software. The system integrates advanced traffic safety solutions, including Citilog, Safety Cloud by AWS, and Predii, to fortify its capabilities in identifying and mitigating potential safety hazards. Citilog employs video analysis to discern and highlight potential safety hazards and near-miss occurrences, enhancing the system's situational awareness. Safety Cloud by AWS leverages machine learning to forecast the likelihood of accidents on highways, providing proactive insights into potential risks. Predii contributes with its predictive software, specializing in forecasting and mitigating collision hotspots. This synergistic utilization of cutting-edge technologies ensures a comprehensive approach to traffic safety, enabling the system to proactively identify, forecast, and address potential safety challenges for a more secure and efficient traffic management ecosystem.

Traffic Safety Systems. The system integrates advanced technologies, including Mobileye Shield+, Nexar Traffic Safety Cloud, and Tome Software Traffic Safety, to bolster its capabilities in real-time traffic safety and collision prevention. Mobileye Shield+ employs innovative real-time alerting technology, acting as a crucial tool to avert collisions between cars, pedestrians, and bikes. Nexar Traffic Safety Cloud harnesses the power of AI and crowdsourced dashcam data, with the overarching goal of reducing collisions through intelligent insights. Tome Software Traffic Safety utilizes data analysis to optimize road layouts and markings, effectively lowering safety hazards. This strategic amalgamation of technologies ensures a multifaceted and proactive approach to traffic safety, leveraging real-time alerts, AI-driven insights, and data-driven road optimization for a safer and more efficient traffic ecosystem.

2.3 Literature Survey

A study [1] addresses the necessity to precisely forecast traffic flow to reduce accidents and congestion, as well as the growing complexity of road networks, especially at crossings. Intelligent transportation systems (ITS) are emerging in different cities, combining technologies to solve transportation problems brought on by the notable growth in traffic. It emphasizes how crucial road safety is to ITS, especially at intersections, which are thought to be the most complicated component of the road network because of the interactions between various users, including cars and pedestrians. Additionally, the introduction emphasizes the difficulties in predicting traffic flow, such as the dynamic, nonlinear character of traffic and the requirement for scalability to manage the massive volumes of data produced by traffic sensors. The substantial amount of traffic data that was gathered over six years in the city of Melbourne is processed and analyzed by the study using big data analytics. To increase the accuracy of traffic flow prediction models and to better comprehend traffic movement, this entails handling and analyzing real-time traffic data.

In this study [2] in Intelligent Transportation Systems (ITS), big data analytics refers to the growing volume and complexity of data produced by multiple sources, including GPS, induction loop detectors, microwave radars, video surveillance, remote sensing, and radio frequency identification data. Three Vs describe Big Data features seen in the increasingly complex data acquired by ITS: volume, variety, and velocity. Advanced technologies like electronic sensor technologies, data transmission technologies, and intelligent control technologies are integrated into the transportation systems to handle the vast and complicated data created by ITS. Big Data systems like Apache Hadoop and Apache Spark are utilized to process the vast and intricate amounts of data produced by ITS. These platforms use parallel computing power and distributed file systems to process data quickly. They can assist with large-scale system optimization and make sense of Big Data. Furthermore, specific Big Data processing frameworks for Intelligent Transportation Systems (ITS) have been created. These include a real-time traffic control platform and tools for determining the average speed and congested areas of a roadway.

The article "Towards Accident and Congestion Prevention" [3] examines the application of Big Data analytics tools in order to improve road safety and obtain insight into traffic accidents. The research employs two sizable datasets, one about casualties and the other about accidents, to assess various classifiers and tackle obstacles including class imbalance and computation time. The objective of the authors is to construct a system that can anticipate and avert traffic incidents in real-time. In doing so, they seek to comprehend how human actions affect traffic flow and safety determinations. The

potential for developing intelligent transportation systems through the analysis of traffic data is also emphasized, as is the significance of location as a vital indicator of human mobility. Furthermore, the study cites prior investigations concerning the prediction and categorization of traffic accidents, underscoring the importance of employing large datasets and sophisticated analytics tools to acquire precise and implementable insights. Based on the feature selection methodologies employed in the research, the casualty type—which comprises a variety of classifications including pedestrian, cyclist, taxi rider, horse rider, tram occupant, and vehicle occupant—is among the most significant predictors of traffic accidents. The paper underscores the substantial impact that human-related attributes, including but not limited to age, sex, and other combined or individual characteristics, have on traffic flow and incidents. Furthermore, the research findings suggest that these attributes about human beings are highly significant predictors within the dataset, underscoring the influence of human conduct on road safety determinations and traffic collisions. Utilizing Big Data analytics tools to mine and analyze massive traffic accident and casualty datasets, including H2O and WEKA. These tools are utilized to perform feature selection, assess classification algorithms, and resolve the issue of imbalanced classes in the datasets. Additionally, aggregation and quality measures are implemented in the study to address the issue of class imbalance and improve the precision of predictions. In general, the article makes significant contributions to the utilization of Big Data analytics in the field of transportation, particularly regarding the prevention of accidents and congestion.

This study [4] discusses the use of disaster data from real-world incidents to improve the safety of autonomous vehicles. This underscores the significance of analyzing accident data to acquire knowledge regarding the circumstances and origins of accidents, thereby aiding in the development of secure and optimal routes for autonomous vehicles. Furthermore, the authors emphasize the importance of incorporating real-world data and user preferences to offer accurate insights into decision-making processes, thereby enhancing user confidence and fostering greater acceptance of autonomous vehicle technology. It addresses the application of Big Data mining and analysis methods to examine real-life accident data, to forecast collisions and congestion in transportation networks. The authors further mention the formation of international coalitions and initiatives to gather and assess data on road accidents, to aid researchers in the development of road safety applications, particularly those for connected and autonomous vehicles. In essence, incorporating empirical accident data into the process of determining secure trajectories for autonomous vehicles may ultimately contribute to the acceptance and safety of autonomous vehicle technology by mitigating the occurrence of accidents, casualties, and fatalities.

This paper [5] discusses the creation of a big data-based vehicle crash analysis program for New York City (NYC). It highlights the importance of open data in transportation research, emphasizing how it fosters innovation, raises standards of accountability, and fosters social trust. The application makes it possible to visualize data about vehicle accidents in New York City during the previous nine years, with an emphasis on identifying patterns in the number of injuries and fatalities among residents of the city's boroughs. Furthermore, the technology facilitates comprehensive study, including causal analysis of the primary causes or significant contributing variables of crashes, to offer insights for reducing future traffic-related fatalities. The methodology uses Python modules including Pandas, Matplotlib, Folium, and Streamlit for data extraction and cleaning, geographic analysis, and data visualization. Big data technologies are used in the study for data extraction, cleaning, and analysis. It speaks specifically about using an API for data extraction and storage. The report also mentions using Streamlit and large data tools to produce visualizations based on the previously mentioned approaches. Although the precise big data technologies are not stated, it is assumed that big data frameworks or tools are used in the data processing and analysis to manage the substantial amount of accident event records that the NYC Police Department has gathered. These technologies handle and analyze large datasets efficiently to obtain insightful information on traffic occurrences in New York City. The study concludes by highlighting the need for open data in transportation research and the creation of a big data platform for NYC auto crash investigation. The tool offers a user-friendly interface that facilitates the direct visualization of data, enabling the observation of trends in individual injuries and fatalities occurring within NYC boroughs. It also facilitates comprehensive research, including causal analysis of the primary causes or significant contributing factors to a crash. The methodology uses Python modules including Pandas, Matplotlib, Folium, and Streamlit for data extraction and cleaning, geographic analysis, and data visualization. The study also details upcoming projects, such as developing a predictive machine learning model using the insights gleaned from data mining and analysis.

This study [6] discusses how to build a Big Data analysis and prediction system for Vehicular Networks (VN) that can be used in real-time to analyze and predict traffic accidents and congestion. The system provides an accurate Estimated Time of Arrival (ETA), predicts accidents and congestion before they occur, and updates ETA in the event of an accident or congestion by using real-time average speed data from vehicle detectors and online streaming data from on-road cars. The system is fast, accurate, and dependable because it is cloud-based and makes use of Lambda Architecture (LA). The benefits of adopting big data analysis in Vietnam to improve traffic efficiency and road safety are also covered in the study, along with the difficulties in putting such a system into

practice. As part of its Big Data analysis, the suggested system uses Distributed Random Forest (DRF) and Naive Bayes (NB) classifiers to anticipate accidents and traffic jams before they occur. The spatiotemporal data for road segments, which includes elements like the number of vehicles, the state of the road surface, the day of the week, and the weather, is analyzed using these classifiers. The system can precisely forecast the possibility of accidents and traffic jams by analyzing this data in real time, enabling preventative or mitigating actions to be implemented. The study concludes with the presentation of a real-time Big Data analysis and prediction system for vehicular networks (VN) to reduce or eliminate traffic jams and accidents. The system can accurately predict accidents and congestion before they happen, update the Estimated Time of Arrival (ETA) in the event of an accident or congestion, and send alerts to participating vehicles by using Lambda Architecture (LA) and streaming spatiotemporal data from vehicles and road infrastructure. The suggested system shows how Big Data analysis in Intelligent Transportation Systems (ITSs) and Vehicular Networks can improve traffic efficiency and road safety.

Chapter 3 System Requirements and Analysis

3.1 Domain and Business Requirements

The system operates on a comprehensive data lifecycle, starting with the real-time collection of diverse traffic-related data such as incidents, road closures, red light violations, speed violations, and historical traffic congestion from Chicago's designated open data portal. To meet this requirement, the system employs AWS cloud resources, utilizing S3 buckets, Glue, Redshift, and Athena for efficient staging and processing of incoming traffic data, seamlessly handling both file drops and API integration. Subsequently, the system capitalizes on advanced big data analytics techniques to analyze the processed data, extracting meaningful patterns and insights about road traffic safety, crashes, and congestion. This integrated approach, from data collection to analysis, ensures a comprehensive understanding of traffic dynamics, enabling informed decision-making and interventions for enhancing road safety and mitigating congestion in Chicago.

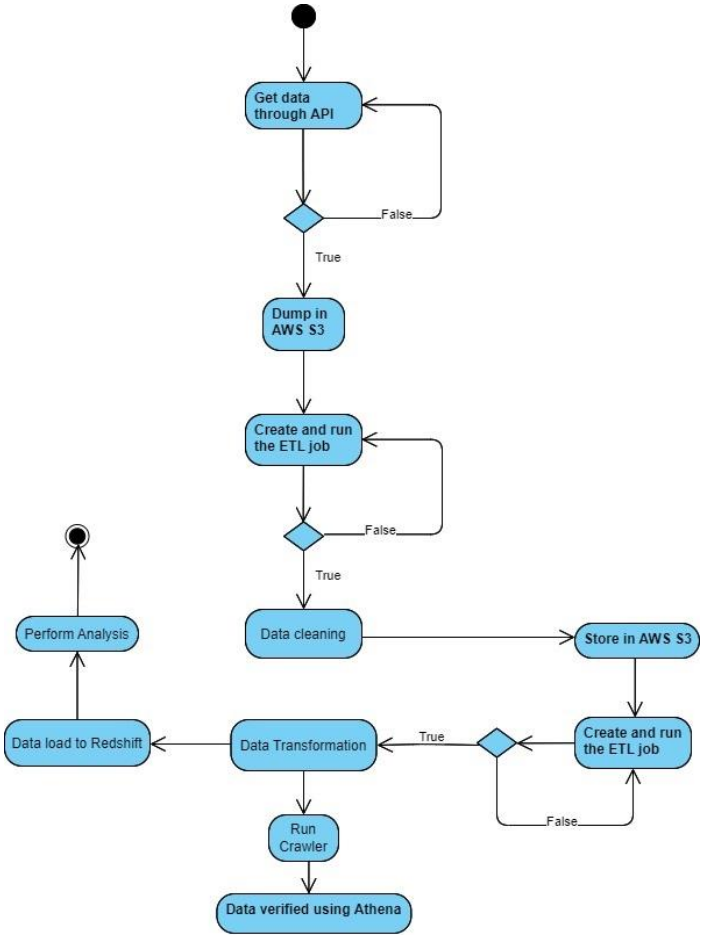
The system is designed to implement real-time interventions based on data-driven insights, fostering enhanced traffic management and safety. To meet the business requirements, the front-end application, powered by AWS QuickSight, Tableau, and Power BI, ensures a user-friendly graphical interface. This interface empowers stakeholders to visualize and explore traffic data, facilitating effective decision-making. The system also addresses the need for comprehensive reporting, generating detailed reports summarizing key findings, insights, and visualizations for various stakeholders, including city planners, transportation authorities, and law enforcement.

Furthermore, the system leverages historical and current data, providing a holistic view of traffic safety and congestion for urban planning and public safety efforts. Scalability is a pivotal aspect, and the system's architecture is designed to scale seamlessly, accommodating increasing data volumes and evolving requirements. To ensure reliability and availability, the system minimizes downtime, providing continuous and uninterrupted traffic monitoring. Robust security measures are implemented to protect sensitive traffic data and ensure compliance with data privacy regulations.

Lastly, the project places a strong emphasis on user training and support. It provides comprehensive training materials and ongoing support for users and administrators, ensuring effective operation and management of the real-time dashboard system. This holistic approach ensures that the system not only meets technical requirements but also aligns with user needs, compliance standards, and the dynamic nature of traffic management. The UML activity diagram can be seen below.

UML Diagram 1

UML Activity Diagram



The UML Class Diagram depicts the relationships and dependencies among key entities in the context of a comprehensive traffic analysis system. The central entity is the "Vehicle," which is associated with two dependent classes: "Model" and "Vehicle History." The "Model" class represents the specific model details of the vehicle, capturing information such as make, year, and specifications. The "Vehicle History" class maintains a record of the historical data associated with the vehicle, including past incidents, maintenance records, and other relevant details.

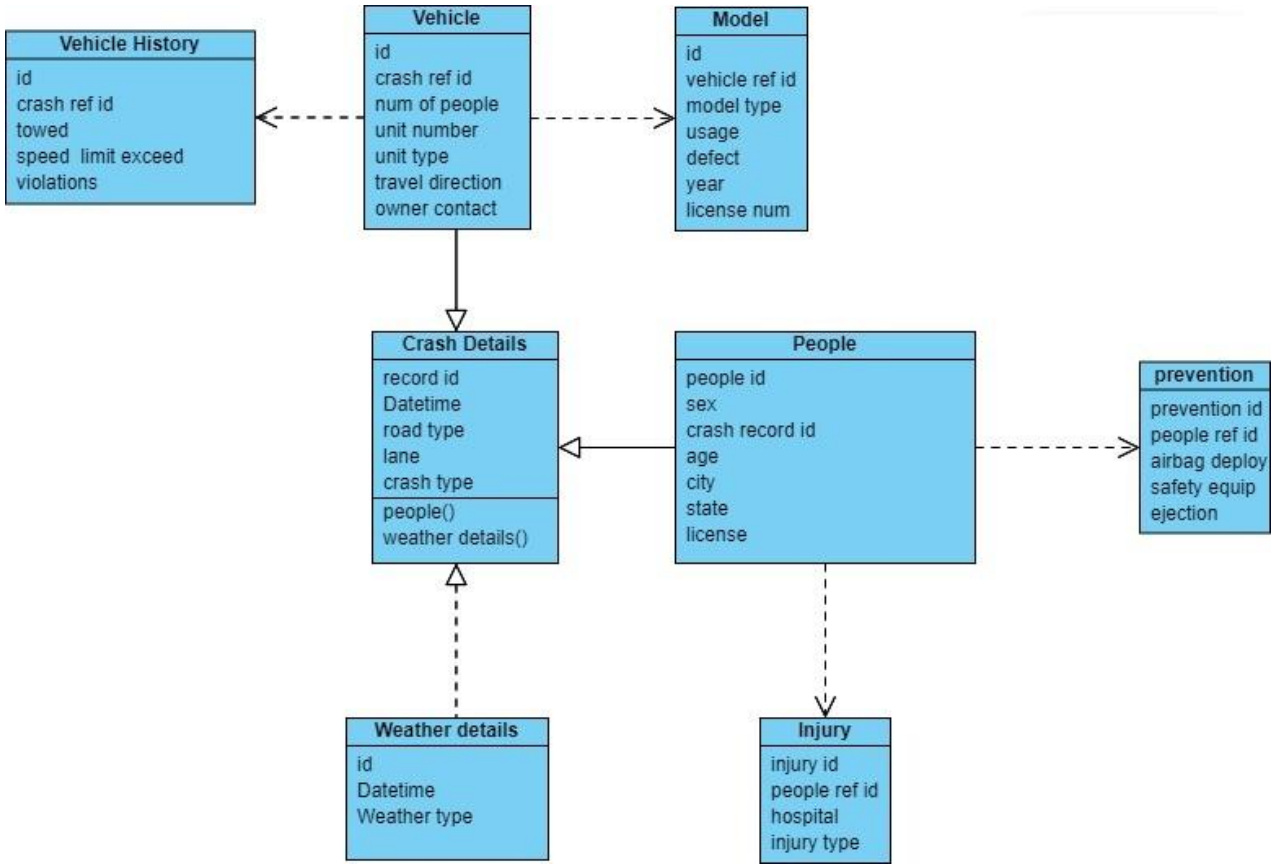
The "Crash Details" class is in generalization with the "Vehicle" class, signifying that crash details inherit properties from the vehicle entity. This relationship implies that crash details are specific instances related to a particular vehicle. Additionally, the "Crash Details" class is associated with the "Weather Details" class, indicating a dependency on weather information during the crash event. This dependency helps in capturing contextual data related to weather conditions at the time of the crash.

The "People" class, on the other hand, is in generalization with the "Crash Details" class, suggesting that details about individuals involved in a crash event are instances of the "People" class. The "People" class is further associated with two dependent classes: "Prevention" and "Injury." The "Prevention" class represents preventative measures or actions taken to avoid or minimize the impact of a crash, while the "Injury" class captures information about injuries sustained by individuals during a crash event.

In essence, this UML Class Diagram illustrates the interrelationships among vehicles, crash details, people, and associated entities, providing a comprehensive view of how these elements are structured and connected within the broader context of a traffic analysis system. The UML class diagram can be seen below.

UML Diagram 2

UML Class Diagram



3.2 Customer-Oriented Requirements

City Planners. City planners are responsible for urban development and infrastructure planning. They require insights into traffic patterns for informed decision-making. The use cases with their descriptions can be seen in the below table.

Table 2

Use Cases and Description of City Planners

Use Case Title	Description
View Comprehensive Reports	City planners can view detailed reports on traffic patterns.
Access Historical Data	City planners can access historical data for long-term planning.
Real-Time Planning	City planners can utilize real-time insights for immediate planning.

Transportation Authorities. Authorities overseeing transportation infrastructure and management. They need real-time data for traffic control and incident response. The use cases with their descriptions can be seen in the below table.

Table 3

Use Cases and Description of Transportation Authorities

Use Case Title	Description
Real-Time Traffic Control	Transportation authorities can monitor and control traffic flow in real time.

Use Case Title	Description
Instant Incident Notifications	Transportation authorities receive instant notifications on traffic incidents.
Historical Data Analysis	Transportation authorities can access historical data for trend analysis.

Law Enforcement. Police and law enforcement agencies use traffic data for monitoring and enforcing traffic laws, ensuring public safety. The use cases with its descriptions can be seen in the below table.

Table 4

Use Cases and Description of Law Enforcement

Use Case Title	Description
Traffic Incident Investigation	Law enforcement can investigate traffic incidents using historical data.
Real-Time Traffic Violation Alerts	Law enforcement receives real-time alerts on traffic violations.
Enforcement Strategy Analysis	Law enforcement can analyze patterns to enhance enforcement strategies.

Public Safety Officials. Officials responsible for public safety and emergency response. They need real-time data during emergencies and incidents. The use cases with their descriptions can be seen in the below table.

Table 5

Use Cases and Description of Public Safety Officials

Use Case Title	Description
Real-Time Emergency Response	Public safety officials can utilize real-time data for emergency response.
Incident Analysis	Public safety officials access historical data for incident analysis.
Collaborative Decision-Making	Public safety officials collaborate with other agencies based on traffic insights.

General Public. Citizens are interested in real-time traffic information for planning commutes and avoiding congestion. The use cases with their descriptions can be seen in the below table.

Table 6

Use Cases and Description of General Public

Use Case Title	Description
Real-Time Traffic Information	The general public can access real-time traffic information through the dashboard.

Use Case Title	Description
Incident Notifications	The general public receives notifications on road closures and incidents.
Alternative Route Planning	The general public can plan alternative routes based on real-time congestion data.

Business Stakeholders (Automobile industry). Businesses planning to design vehicles with integrated navigation systems. They require traffic data to enhance navigation features. The use cases with their descriptions can be seen in the below table.

Table 7

Use Cases and Description of Business Stakeholders

Use Case Title	Description
Traffic-Informed Navigation	Vehicle design businesses integrate real-time traffic data for advanced navigation.
Route Optimization	Navigation systems optimize routes based on current traffic conditions.
Integration Planning	Businesses plan the integration of traffic data into vehicle navigation systems.

3.3 System function requirements

Data Collection and Processing. The Real-Time Data Integration feature involves seamlessly incorporating continuous streams of real-time traffic data from the Chicago portal into AWS Athena and Redshift. This process ensures the availability of processed and integrated data for quick access and analysis. The system is designed to handle incoming data in real-time, maintaining accuracy and relevance, and generating logs or notifications to keep stakeholders informed about the integration status. For Historical Data Import, the system manages periodic batches of historical traffic data, including information on past incidents, traffic patterns, and road closures. It imports this data into AWS S3 and Glue, ensuring proper organization and structure. Validation checks are incorporated to maintain data integrity during the import process. The system generates reports or notifications indicating the success or failure of historical data imports, allowing users to stay informed about the completeness and accuracy of historical datasets.

Table 8

Traffic Data Pipeline Features

Feature	Inputs	Behavior	Outputs
Real-Time Data Integration	Real-time traffic data from Chicago portal	Integrate incoming data into AWS Athena and Redshift	Processed data stored in Athena and Redshift
Historical Data Import	Historical traffic data from Chicago portal	Import historical data into AWS S3 and Glue	Staged and processed historical data in S3 and Glue

Data Analysis and Insights. In Traffic Pattern Analysis, the system employs advanced big data analytics techniques on processed data retrieved from AWS Athena and Redshift. This includes both real-time and historical traffic information. The analysis aims to identify traffic patterns, congestion hotspots, and historical trends, utilizing machine learning algorithms for predictive insights. The system provides detailed visual representations of traffic patterns, offering customizable analysis reports for city planners and transportation authorities. These insights equip stakeholders with actionable information for optimizing traffic flow and planning interventions effectively.

Table 9
Data-Driven Traffic Management Features

Feature	Inputs	Behavior	Outputs
Traffic Pattern Analysis	Processed data from Athena and Redshift	Apply big data analytics for traffic pattern recognition	Insights on traffic patterns, congestion, and incidents
Real-Time Interventions	Real-time insights from analysis	Implement interventions based on analysis results	Real-time adjustments to traffic flow and incident response

Table 10
User Interface and Reporting

Feature	Inputs	Behavior	Outputs
User-Friendly Dashboard	Processed data for visualization	Develop a user-friendly interface for data exploration	Dashboard with interactive visualizations and reports
Comprehensive Reports	Analyzed data for reporting	Generate detailed reports for stakeholders	Reports summarizing key findings and insights

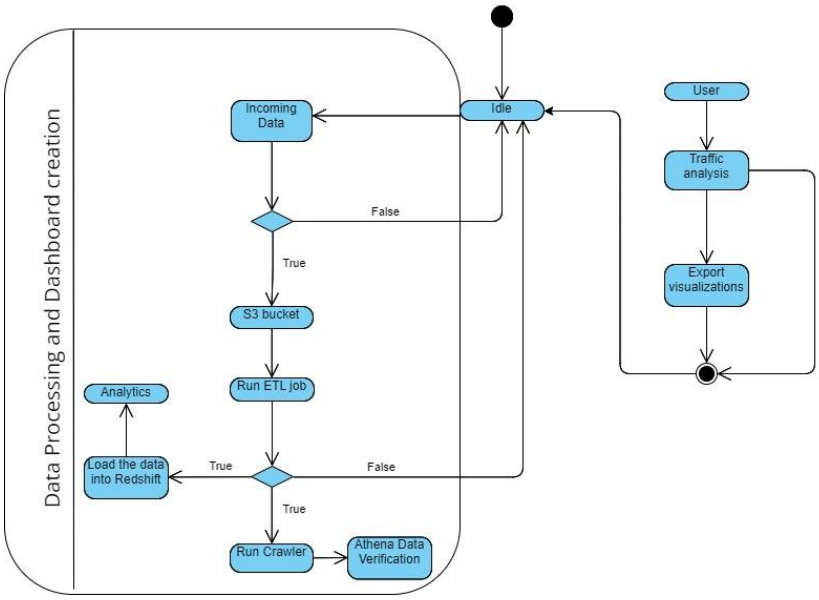
3.4 System Behavior Requirements

The State Machine Diagram portrays the operational flow of the traffic analysis system, featuring an "Ideal State" at the center. On the left side, the system transitions to states such as "Initializing Data Processing" and "Dashboard Creation," representing the initialization of data processing activities and the concurrent creation of user-friendly

dashboards. On the right side, the diagram depicts the transition to the "User Interaction" state, where stakeholders engage with the system to access and analyze processed data. The continuous back-and-forth transitions between user interaction and data processing states highlight the dynamic nature of the system as users explore visualizations and reports. The diagram concludes with the "Ending the Machine" state, symbolizing the termination of system operation after users have completed their interactions, providing a comprehensive visualization of the traffic analysis system's workflow. The UML State machine diagram follows below.

UML Diagram 3

UML State Machine Diagram



The below table shows each state and transition would be accompanied by detailed descriptions, conditions triggering transitions, and potential actions or events associated with each state.

Table 11
Transitions Table

From State	To State	Transition Description
Ideal State	Initializing Data Processing	Transition to initiate data processing.
Initializing Data Processing	Dashboard Creation	Concurrent transition for dashboard creation.
Dashboard Creation	User Interaction	Transition as dashboards are ready for user interaction.
User Interaction	Initializing Data Processing	Transition as users engage with the system for data processing.
User Interaction	Ending the Machine	Transition as users complete interactions, ending system operation.

3.5 System Performance and Non-Function Requirements

Performance. The system is engineered to exhibit highly efficient real-time data processing capabilities, prioritizing minimal latency in the integration and analysis of traffic data. It ensures swift imports of historical data within reasonable time frames, facilitating timely trend analysis crucial for informed decision-making. Optimized traffic pattern analysis algorithms contribute to the system's speed and accuracy, ensuring the generation of actionable insights promptly. This approach ensures that the system not only processes real-time data efficiently but also handles historical data with agility, enabling timely and informed interventions for effective traffic management and safety improvements.

Capacity. The system is meticulously designed for scalability, equipped to seamlessly accommodate the escalating volumes of both real-time and historical traffic data. Robust storage solutions, including AWS S3, Redshift, and Glue, offer ample capacity to handle growing datasets, ensuring the system's ability to scale with evolving data requirements. Scalability measures extend to the user interface, addressing the need to handle increased concurrent user interactions effectively. This comprehensive scalability approach not only guarantees the system's resilience against data growth but also ensures an optimal user experience by accommodating the expanding demands on both the backend infrastructure and frontend interactions.

Availability. The system places a paramount emphasis on maintaining high availability to facilitate continuous real-time data integration and analysis. Scheduled maintenance and updates are strategically planned during low-traffic periods to minimize service disruptions, ensuring uninterrupted operations during critical usage times. In addition, the implementation of redundancy and failover mechanisms serves as a proactive strategy to mitigate potential service interruptions. This comprehensive approach not only guarantees the reliability of the system but also ensures that essential maintenance activities are conducted with minimal impact on the continuous and seamless flow of real-time traffic data integration and analysis processes.

Compliance with Standards. Data integration and processing within the system are meticulously designed to adhere to stringent data protection and privacy regulations, ensuring full compliance with relevant standards. To fortify the security of sensitive data, industry-standard encryption protocols are employed during both transit and storage phases, guaranteeing robust protection against unauthorized access or breaches. Furthermore, the system is aligned with AWS best practices and standards for cloud-based data processing, ensuring not only compliance but also the adoption of industry-leading security measures. This approach establishes a comprehensive framework that prioritizes data privacy, security, and conformity with industry and regulatory standards throughout the entire data integration and processing lifecycle.

Security. The system prioritizes security through the implementation of robust access controls and authentication mechanisms, strategically designed to restrict system access based on user roles. Encryption measures are rigorously applied for both data at rest and in transit, ensuring the utmost confidentiality of sensitive information. To maintain a proactive security stance, the system undergoes regular security audits and assessments, systematically identifying and addressing potential vulnerabilities. This comprehensive approach not only safeguards against unauthorized access but also establishes a dynamic security posture that evolves with emerging threats, reinforcing the system's resilience against security risks throughout its operational life cycle.

User Interface. The user interface of the system is meticulously crafted to deliver an intuitive and user-friendly experience, prioritizing ease of navigation for stakeholders. Customization features are integrated, empowering users to tailor their dashboards and reports to precisely meet their individual needs and preferences. Ensuring a seamless user experience, the system exhibits responsiveness and compatibility across a spectrum of devices and browsers. This approach not only enhances accessibility but also provides stakeholders with a versatile and adaptable platform, facilitating optimal engagement and interaction with the traffic data insights presented through the user interface.

3.6 Context and Interface Requirements

Development Environment. In the development phase, Amazon Web Services (AWS) plays a pivotal role, providing a suite of cloud-based services for data storage, processing, and analysis. Amazon S3 facilitates scalable and secure storage, while Glue and Redshift are utilized for efficient data processing and analysis. Athena, a serverless query service, allows for interactive querying of data directly in S3. For development purposes, AWS QuickSight acts as the development interface, providing an intuitive platform for creating and testing interactive dashboards during the system's development phase on AWS.

Testing Environment. The testing phase continues to leverage AWS services for robust testing of ETL pipelines and queries on sample data. Redshift, with its optimized query performance, ensures thorough testing of data transformations. Athena's serverless nature allows for seamless testing of SQL queries on the sample datasets. This environment ensures the reliability and efficiency of the ETL processes before deployment to the production environment.

Deployment Environment. For the deployment phase, the production databases are hosted on AWS Redshift, ensuring a scalable and high-performance environment for handling large datasets. Dashboards and reports, vital for user interaction and decision-making, are deployed on AWS QuickSight. QuickSight's integration with other AWS services ensures a seamless deployment of visualizations, providing stakeholders with an intuitive interface to explore and interpret the analyzed data.

System Interface Requirements. The system interfaces with users and other systems through key interfaces, ensuring effective communication and interaction. These interfaces include:

User Interface. This facilitates interaction with end-users through intuitive dashboards and reports deployed on AWS QuickSight.

Ingestion Interface. It manages the ingestion of raw data into the system, utilizing AWS services like S3, Glue, and Redshift for efficient processing and storage.

Query Interface. This enables users to interact with the system through SQL queries, leveraging the capabilities of AWS Athena for direct querying of data stored in S3.

Development Interface. AWS QuickSight acts as the development interface, providing an intuitive platform for creating and testing interactive dashboards during the system's development phase on AWS.

User Interface Structure. The intuitive dashboard and report designs of the system exemplify a commitment to data visualization best practices, embodying principles that prioritize clarity and ease of interpretation. The incorporation of robust features such as filtering, sorting, and drilling capabilities empowers users with the tools needed for in-depth data exploration and analysis. This functionality not only enhances the system's versatility but also ensures that users can navigate through complex datasets effortlessly. The inclusion of customizable views is a strategic design choice, allowing users to tailor their visualizations according to their specific needs. This flexibility enables a more focused and personalized experience, ensuring that relevant metrics take center stage for each user, enhancing their overall engagement and understanding.

Moreover, the system places a strong emphasis on user support and accessibility. Comprehensive help documentation and tutorials are thoughtfully provided, fostering a user-friendly onboarding process for new users. This commitment to user education and ongoing support underscores the system's dedication to ensuring that users, regardless of their familiarity with the platform, can leverage its capabilities effectively. Overall, the user interface structure, as depicted in the table, reflects a thoughtful approach to design, where user experience, customization, and support are prioritized to create a robust and user-friendly data exploration and analysis platform.

Table 12

User Interface Structure

Component	Description
Dashboards/Application	The dashboards on QuickSight allow users such as traffic planners, law enforcement agencies, and city officials to visualize key performance indicators (KPIs) and metrics related to traffic crashes with vehicle crashes and management.
Reports	Canned reports in standardized PDF format are available on Amazon QuickSight and can be easily downloaded and shared to convey insights.

API Interfaces. Leveraging ETL APIs, the project seamlessly ingests raw data from source systems into staging layers, orchestrating the efficient loading process into the Amazon Redshift database for further analysis and storage. This streamlined pipeline ensures data integrity and accessibility for robust database management.

Thus, the preliminary user interface provides multiple modalities for users and stakeholders to explore insights as per their needs and devices intuitively and interactively. This facilitates effective decision-making.

3.7 Technology and Resource Requirements

Data Technologies. The project seamlessly integrates various AWS services to manage crash data efficiently. Amazon S3 serves as a secure and scalable storage solution, handling both raw and processed crash data. AWS Glue plays a pivotal role in data cataloging and executing Extract, Transform, Load (ETL) workflows, ensuring the transformation and loading of data into the desired format. AWS Glue Crawler further enhances the process by profiling and cataloging schemas of raw crash data files from S3, populating the metadata repository crucial for downstream data transformation and analytics. Amazon Redshift functions as the designated data warehouse, hosting processed crash attributes in a scalable and high-performance environment. AWS Athena complements the setup by facilitating SQL queries directly on structured crash data stored in S3, contributing to a seamless and powerful data processing and analysis pipeline.

Analytics Technologies. The project leverages Amazon QuickSight for constructing interactive dashboards and reports, offering a user-friendly interface for data exploration. AWS Jupyter Notebook, utilizing Python, complements the process by enabling robust data processing, modeling, and API development. This integrated approach ensures a seamless workflow, combining QuickSight's visualization capabilities with Python's versatility for comprehensive data manipulation and analysis.

Infrastructure. The project incorporates Identity and Access Management (IAM) roles to establish robust authentication and access management, ensuring secure and controlled access to resources. IAM roles play a pivotal role in governing user permissions and enhancing security measures by granting or restricting access based on predefined roles and responsibilities.

Also, a Virtual Private Cloud (VPC) is used to establish a secure network architecture. VPC ensures the isolation and segmentation of resources, creating a private and controlled environment within the cloud infrastructure. This secure network architecture, coupled with IAM roles, fortifies the overall system against unauthorized access, providing a foundation for safeguarding sensitive data and ensuring the integrity of the project's cloud-based infrastructure.

Data Sources. The primary dataset for this project is sourced from the Chicago Data Portal, a comprehensive repository containing crucial information about vehicle

crashes in Chicago. It is accessible through the website http://www.chicago.gov and is owned by Jonathan Levy. The portal is a valuable resource that is curated with data reported by the Chicago Police Department in adherence to the IL Traffic Crash Reporting Form SR1050. The dataset encompasses details about individuals involved, injuries sustained, vehicles implicated, and comprehensive information about each traffic crash.

To harness the potential of this dataset, the project employs a systematic approach. The data, which was last updated on November 28, 2023, undergoes extraction, transformation, and loading (ETL) processes. AWS services such as S3 provide scalable and secure storage, Glue and Athena facilitate efficient ETL processing, and Redshift serves as a robust data warehouse. This orchestrated integration enables seamless storage, processing, verification, and exploration of the crash data, laying the foundation for comprehensive analysis and visualizations.

By leveraging AWS infrastructure and services, the project ensures a dynamic and scalable environment for handling the intricacies of crash data. The extracted, transformed, and loaded datasets serve as a rich source for exploring patterns, identifying trends, and deriving valuable insights that can inform strategies for enhancing road traffic safety and mitigating congestion in Chicago.

Resources. The project's infrastructure is fortified by robust cloud computing capacity, specifically tailored for storage and analytics workloads, ensuring scalability and efficiency. Integrated development environments and operating systems serve as essential development tools, optimizing the coding processes. Hardware resources, encompassing desktops and laptops, facilitate the development phase. The project benefits from a skilled team comprising data engineers, analysts, and visual designers, essential for comprehensive data processing and visualization. Ongoing maintenance and support mechanisms are ingrained in the project framework, guaranteeing the sustained reliability and functionality of the system. This holistic setup ensures the seamless integration of technology, human expertise, and ongoing support for the project's success. The software and hardware specifications are listed below.

Table 13

Software Specifications

Software	Configuration
Operating System (OS)	macOS Big Sur
Integrated Development	AWS Jupyter Notebook

Software	Configuration
Environment (IDE)	

Table 14
Hardware Specifications

Hardware	Configuration
CPU	Apple M1 Pro chip with 8-core CPU
GPU	8-core GPU
RAM	6GB unified memory
Disk Space	500GB SSD storage
Performance	4.1 TFLOPS / 8.1 TFLOPS
Max execution time	12 hours

This set of technologies provides a scalable, secure, and collaborative environment for the project objectives.

Chapter 4 System Design

4.1 System Architecture Design

The project involves a series of key components and their intricate interconnections for comprehensive data analysis. The first step of the process is to obtain data from the Chicago Data Portal. This data includes People Data, Traffic Crash Data, and Vehicle data, which are then staged inside an AWS S3 bucket for further processing.

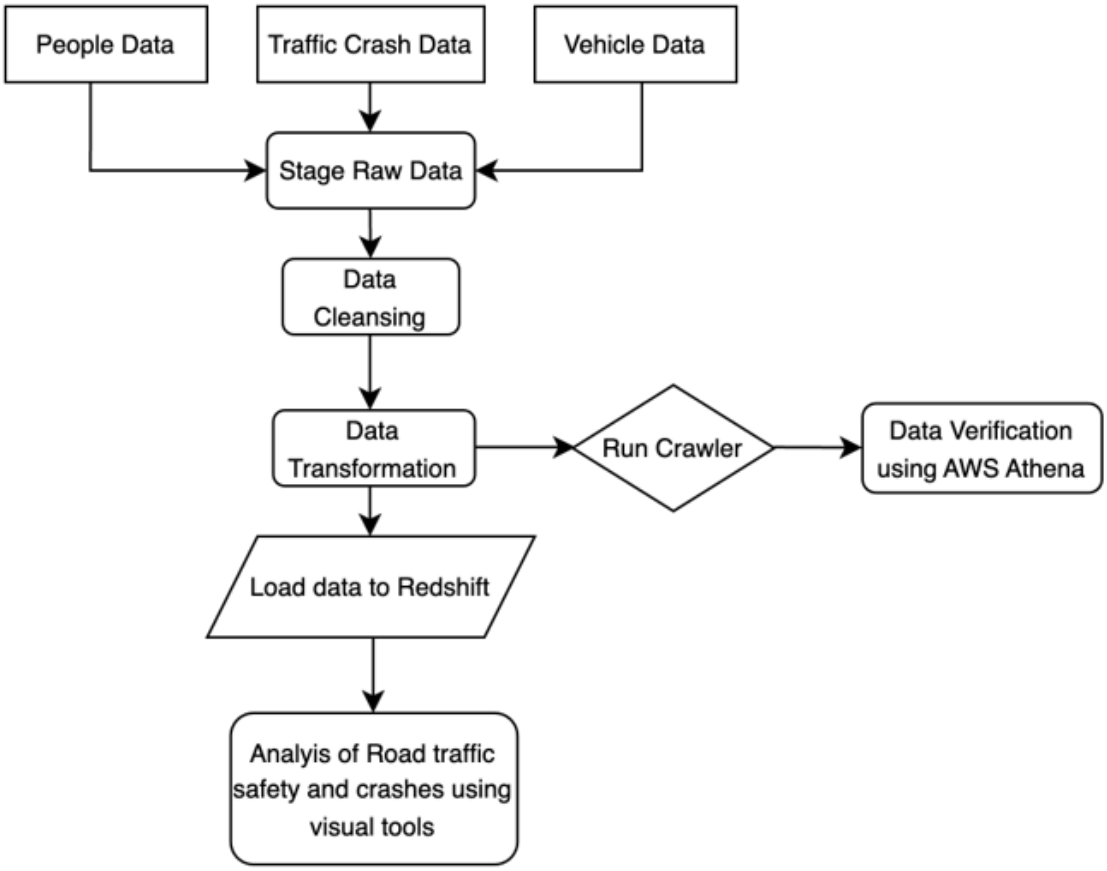
To ensure that the raw data is cleansed and transformed into a format suitable for analysis, AWS Glue jobs are utilized to run spark scripts taking these data as input. This process involves several stages, including data cleansing, data transformation, data enrichment, and data aggregation. The result is a transformed dataset that is ready for analysis.

Once the data has been transformed, an AWS Crawler is run to obtain the schema of the transformed data. The crawler is configured to connect to the S3 bucket and access the transformed data. The crawler then extracts the metadata from the data and stores it in a new crawler database. This database is used to query and test the data in AWS Athena. Once the prepared data has been loaded into AWS Redshift, it acts as a warehouse for the data, enabling all queries and analysis to be conducted according to business requirements. AWS Redshift is a powerful data warehouse that can handle large amounts of data and provide fast query performance. It is also highly scalable, which means that it can easily accommodate new data sources as the project evolves. The analysis and findings of the data are then integrated into AWS Quicksight, which is used to visualize the findings and insights obtained from the data by creating dashboards. AWS Quicksight is a cloud-based business intelligence tool that provides powerful visualization capabilities. It allows users to easily create visualizations, dashboards, and reports that can be shared with others.

Overall, this process enables the data to be collected, cleansed, transformed, and analyzed efficiently and effectively. By utilizing these key components and interconnections, the project can provide valuable insights and findings that can be utilized to drive business decisions and improve outcomes. The system architecture is highly scalable and adaptable, allowing for new data sources to be added and analyzed as the project evolves.

Figure 2

System Architecture Diagram



4.2 System Data and Database Design

The System data is structured under the principles of an OLAP system, which is inherently suitable for analytical processing. This design allows for the data to be processed and analyzed in a manner that is optimized for insights and decision-making. The database is constructed in a star schema architecture, which comprises a central fact table and three dimension tables. This approach is particularly effective for handling large data sets with complex relationships between variables.

The fact table is designed to capture the crash data, which is the main focus of the system. It includes a variety of metrics related to crash events, such as location, time, and severity. The dimension tables, on the other hand, contain information about the people and vehicles involved in the crashes, as well as other relevant contextual data. In addition to the two primary dimension tables, a date dimension table is established at the time of

database design. This table allows for timeline querying and enhances analytical capabilities by providing a hierarchical structure for date-related data. This dimension is essential for conducting temporal analyses, such as identifying trends in crash frequency over time.

All-inclusive, the star schema architecture, and OLAP system design allow for the data to be processed and analyzed efficiently and effectively. The resulting database is intrinsic and amenable to analytical processing, thereby enabling insightful analysis of the underlying phenomena.

Figure 3

Entity-Relationship Diagram

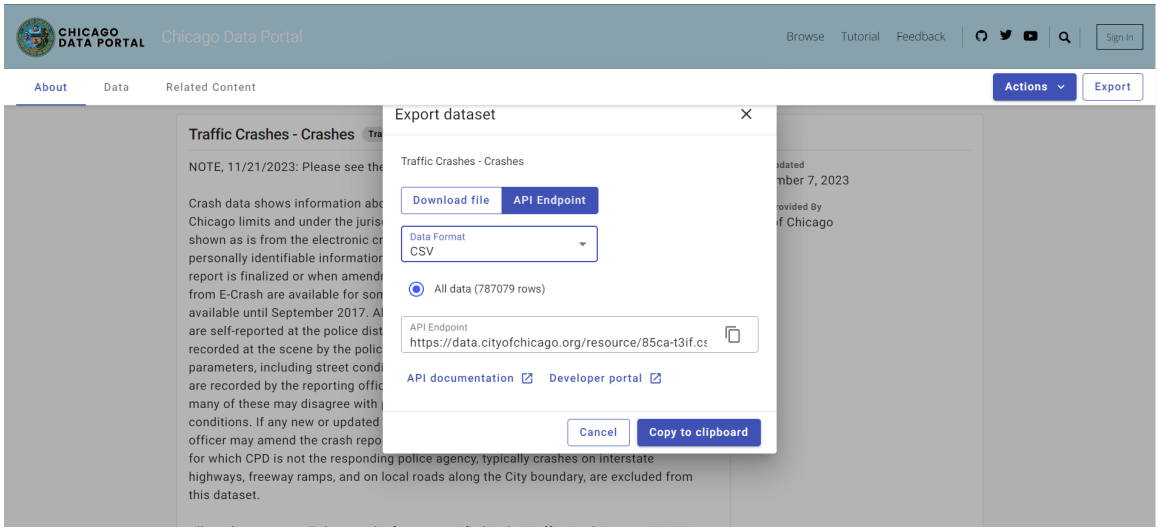
4.3 System Interface and Connectivity Design

External Interfaces. Chicago Data Portal: API calls are made to the Chicago Data Portal for data retrieval. The mechanism involves scheduled or event-triggered requests to fetch the latest data. Data could be retrieved in two formats (i) comma-separated

values (CSV) and (ii) JavaScript Object Notation (JSON). Data has been retrieved in CSV format in this project.

Figure 4

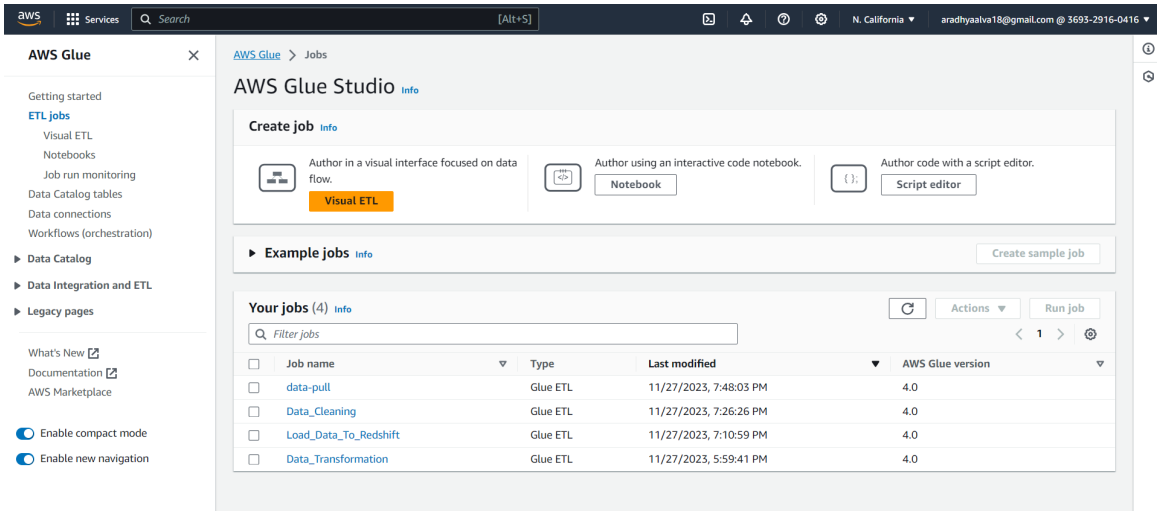
Chicago Data Portal API



AWS Glue. Integration with AWS Glue has been established for data cleansing and transformation. Glue jobs are configured to process incoming data from the Chicago Data Portal. There are four ETL (Extract, Transform, and Load) jobs conducted, each of which is responsible for pulling data from the portal, cleaning the data by removing null values, duplicate values, outliers, and irrelevant data, transforming the data into a standard format, putting them into proper tables as per the database design, and then loading the tables into the warehouse.

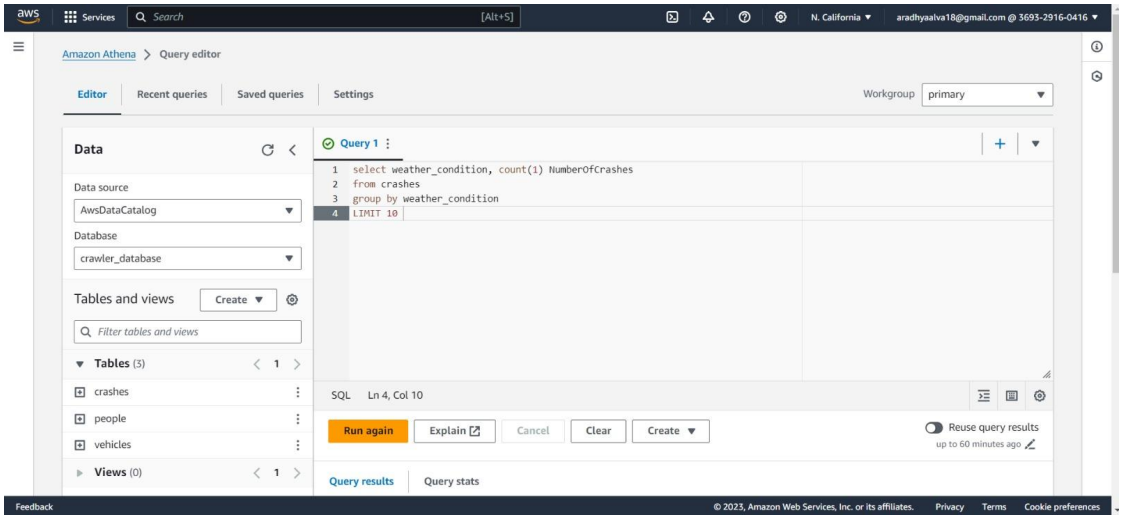
Figure 5

AWS Glue Jobs



User Interfaces. AWS Athena Interface is an interactive query service that AWS provides, which allows users to analyze data stored in Amazon S3 using Structured Query Language (SQL) queries. It is a serverless service, which is particularly useful for ad-hoc analysis, exploring large datasets, and gaining insights from data, all without the need for managing infrastructure or complex data processing pipelines. Athena is leveraged to overlook any underlying patterns of data and test out the integrity of the data obtained.

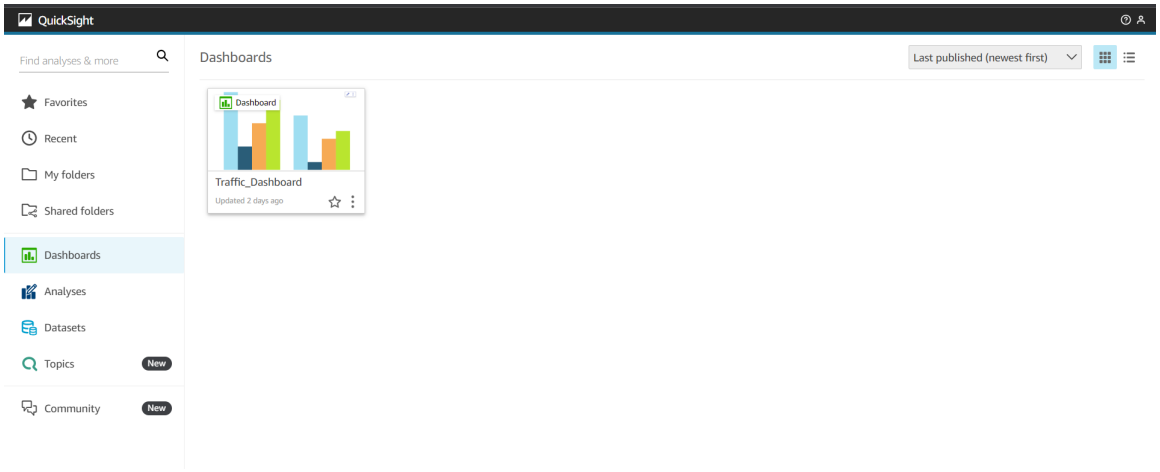
Figure 6
Athena Interface



AWS QuickSight Amazon QuickSight is a cloud-based business intelligence (BI) service provided by Amazon Web Services (AWS), which allows users to create and visualize interactive, sharable dashboards from various data sources. It is designed to provide a user-friendly and cost-effective solution for business intelligence and data visualization, making it easier for organizations to derive insights from their data. It offers a range of data visualization options, including tables, charts, and graphs, as well as the ability to create custom visualizations using APIs and SDKs which are leveraged in the project to create dashboards.

Figure 7

AWS Quicksight Interface



4.4 System User Interface Design

The process of data querying involves a comprehensive approach that is critical to obtaining accurate and useful insights. The operation flow is to first test the data in AWS Athena, a serverless interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. The goal of this step is to ensure that the data is well-structured and ready for further analysis. Once this is done, ad-hoc querying can be performed in the data warehouse. Amazon Redshift, a fully managed, petabyte-scale data warehouse service in the cloud, is utilized for handling large datasets and performing complex queries for data analysis.

The data queried in Amazon Redshift is then integrated into an Amazon Quicksight dashboard, a cloud-powered business intelligence service that enables users to create and publish interactive dashboards and reports. The dashboard allows for further analysis of the data and visualization based on specific business requirements. The dashboard provides a rich array of plots and charts that can be leveraged to present the information clearly and concisely. By comparing different attributes of the data metric-wise, useful insights are obtained from the data and in-depth analysis is performed. Furthermore, the visualization of the data in the Amazon Quicksight dashboard allows for a more comprehensive understanding of the data, enabling informed decision-making.

4.5 System Component API and Logic Design

These include architectural and functional components of the traffic analysis system that indicate the communications and processing patterns among separate elements.

Data Ingestion API

Purpose. This brings in the real-time as well historic traffic data to the system through this API. The system features two endpoints: "/real-time-data" and "/historical-data." The "/real-time-data" endpoint accepts incoming real-time traffic data, while the "/historical-data" endpoint is designed to handle historical batches of traffic data.

Logic. The logic behind these endpoints involves authentication and the organization of incoming data. Upon verification, the system ensures that the data is properly ordered and subsequently enters it into the appropriate data storage, establishing an organized and authenticated flow for both real-time and historical traffic data processing.

Data Processing Logic

Purpose. The primary purpose of the system lies in transforming ingested data into actionable information and discernible patterns. This involves key components such as AWS Athena and Redshift, which collectively execute intricate query operations on historical data.

Logic. The logic of the system is multifaceted, leveraging big data analytics, machine learning algorithms, and historical trend analysis. These components work synergistically to ascertain traffic flow rates, identify congestion patterns, and detect the occurrence of incidents. By applying advanced analytical techniques, the system extracts valuable insights, enabling informed decision-making based on a comprehensive understanding of the processed data.

User Interface (UI) Logic

Purpose. The system serves the purpose of overseeing the front end, functioning as the portal through which users can seamlessly collaborate with the system. Key components in this context include AWS QuickSight, which is responsible for rendering visualizations.

Logic. The underlying logic involves the creation of tailored dashboards that cater to specific user needs. Through interactions with users, the system facilitates the distribution of these customized dashboards, offering an intuitive and user-friendly interface. By leveraging AWS QuickSight, the system not only visualizes data effectively

but also ensures a dynamic and collaborative user experience, enhancing the accessibility and usability of the platform.

Notification Service API

Purpose. The primary purpose of this system is to facilitate real-time alerting by promptly notifying the appropriate individuals. The "/alerts" endpoint plays a central role in this process, serving as the trigger point for incidents and critical events.

Logic. The system's logic revolves around continuous monitoring of data flows, employing a mechanism to flag alarms when certain predefined thresholds are reached. By actively scanning incoming data, the system ensures a proactive response to potential issues, enabling timely notifications that empower users to address incidents and critical events promptly. Through this alerting mechanism, the system enhances its responsiveness to emerging situations, contributing to effective incident management.

4.6 System Design Problems, Solutions, and Patterns

The problems are encountered while designing the project workflow in AWS and they are mentioned as follows:

Data Quality. The data quality challenges are addressed below.

Challenge. The data obtained from the API contained a huge amount of null values, duplicate values, and inconsistent data. Such data would lead to improper analysis and bad-quality output.

Solution. An ETL job is constructed to solve this problem which deals with removing null values and duplicates, handling inconsistent data, and improving the quality of data.

Data Integration.

Challenge. Mapping the data from the S3 bucket into the Redshift database seems challenging because of the difference in data types of attributes.

Solution. An additional layer of transformation is added in the glue job which loads data into the redshift database where the data type of each attribute is compared to the data type of columns present in the redshift schema and then makes the necessary changes to map the data.

Schema Design.

Challenge. Choosing between the star schema and the snowflake schema is challenging. Star Schema often involves denormalization, which means redundant data is stored in dimension tables for the sake of simplicity and query performance. Snowflake

Schema is more normalized, meaning it reduces redundancy by normalizing dimension tables. This can be beneficial for data integrity and storage efficiency.

Solution. The granularity of data and the business requirements for the project are understood. The project required a simpler and straightforward approach toward data where analysis is conducted on a demand basis and comprises more information. Hence, the star schema is chosen.

Metadata Management.

Challenge. Managing metadata about the data (definitions, lineage, etc.) is crucial for understanding the data's context and lineage.

Solution. An AWS Glue Crawler is created to pull the schema from the transformed data and store it inside a crawler database. The crawler is added inside the ETL workflow such that whenever the pipeline is initiated and the data is being pulled, the schema is freshly obtained and up-to-date.

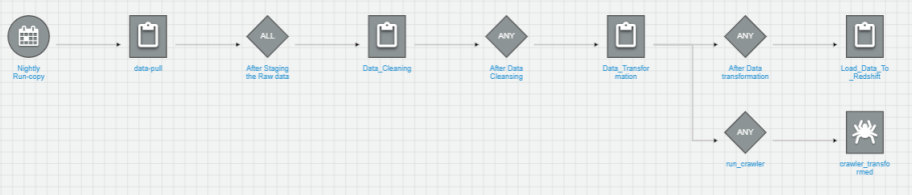
Chapter 5 System Implementation

5.1 System implementation summary

The project begins by starting the ETL workflow, which can be scheduled on a daily, weekly, monthly, or on-demand basis. The workflow is displayed in the figure below, which contains four triggers, four Amazon Glue jobs, and one Amazon crawler. Once the workflow begins, the ETL job 'data pull' is triggered, which pulls the data from the Chicago Data Portal and inserts it into the S3 bucket as per the mentioned path in the glue job. Once this ETL job is completed, and the data is staged in the mentioned S3 bucket, a trigger initiates another ETL job 'Data_Cleaning,' which takes the staged data as input and cleanses the data as per the given instructions and business logic in the spark code.

Figure 8

ETL Workflow



The status of the jobs is visible under AWS Glue monitoring, where the success or failure of the jobs is tracked and shown in the figure below. Once the 'Data_Cleaning' job is completed successfully, it stores the data in the mentioned S3 bucket path. When the ETL job is complete, the trigger 'After Data Cleansing' triggers the ETL job 'Data_Transformation', which deals with standardizing the data and converting it into respective tables that mirror the schema of the database tables present in the data warehouse. Once the transformation is complete and the job succeeds, two respective triggers are executed. The trigger 'After Data Transformation' triggers the ETL job 'Load_Data_To_Redshift', which takes the transformed data as input and loads it into the Redshift database that is already being created. It maps the data from the S3 bucket to the data warehouse. The trigger 'run_crawler' initiates the crawler 'crawler_transformed', which gets the schema of the transformed data and loads it into a crawler database. Catalog tables are generated from the obtained schema, which is then referred to in the mapping job as well. From the Glue Data Catalog tables obtained, the data can be queried and tested out in Amazon Athena to confirm the consistency and integrity of the data.

Figure 9

ETL Jobs Monitoring Window

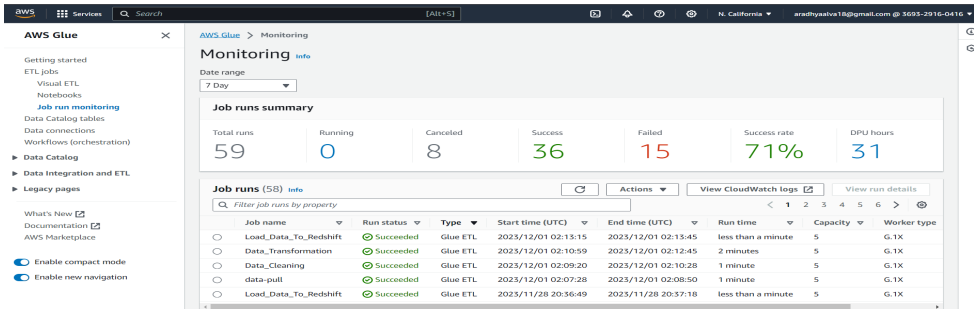


Figure 10

List of Triggers and Status

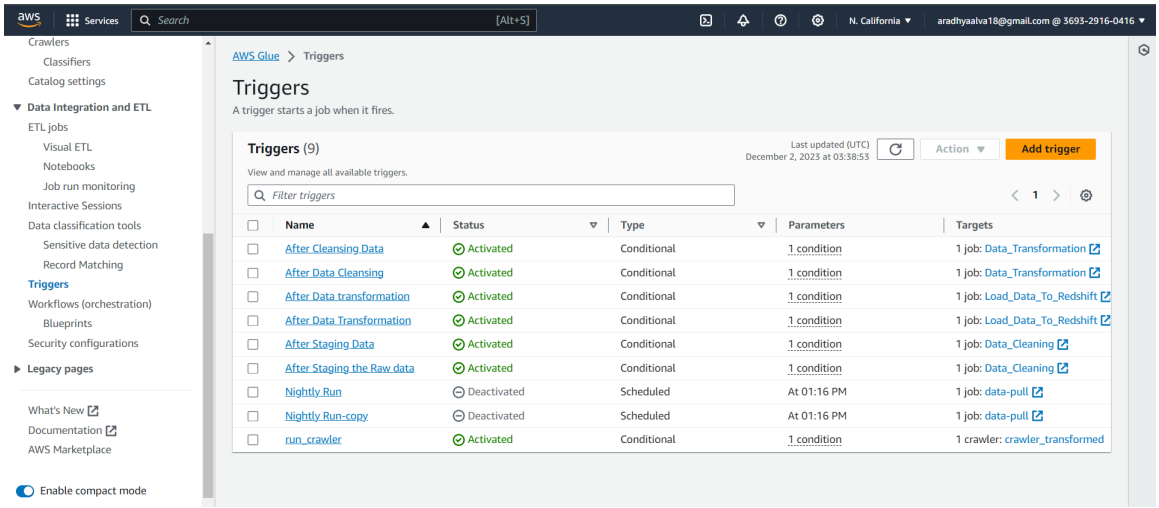


Figure 11

Job Type Breakdown Information

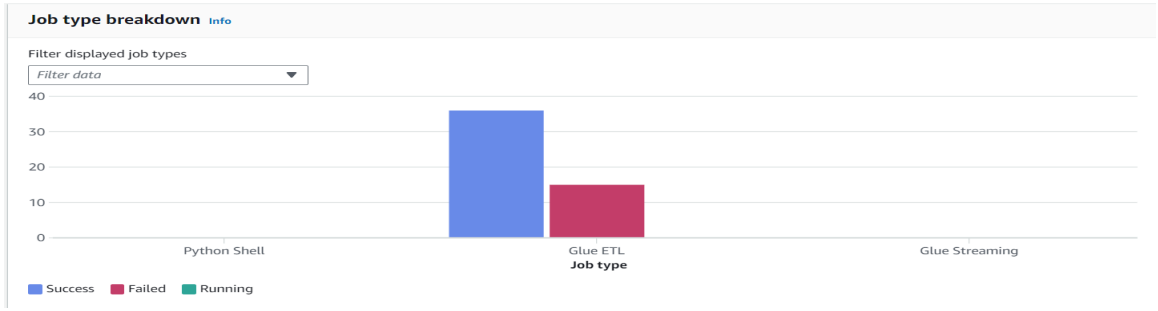


Figure 12

Worker Type Breakdown Information

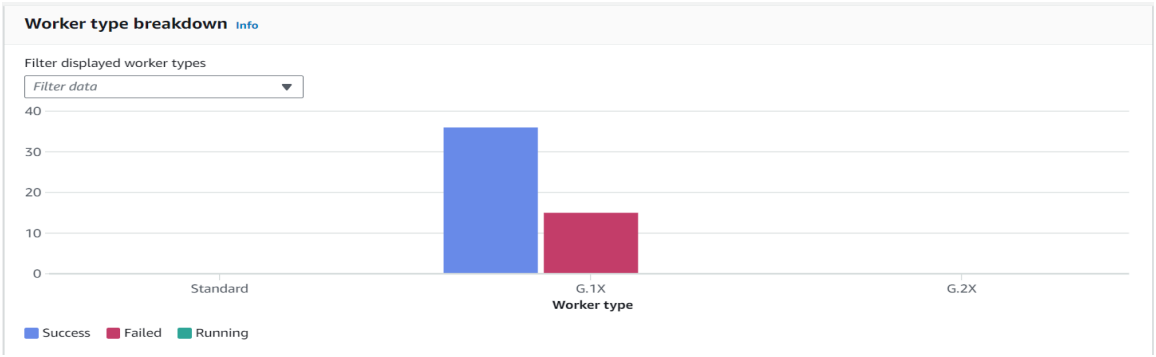


Figure 13

Job Runs Timeline

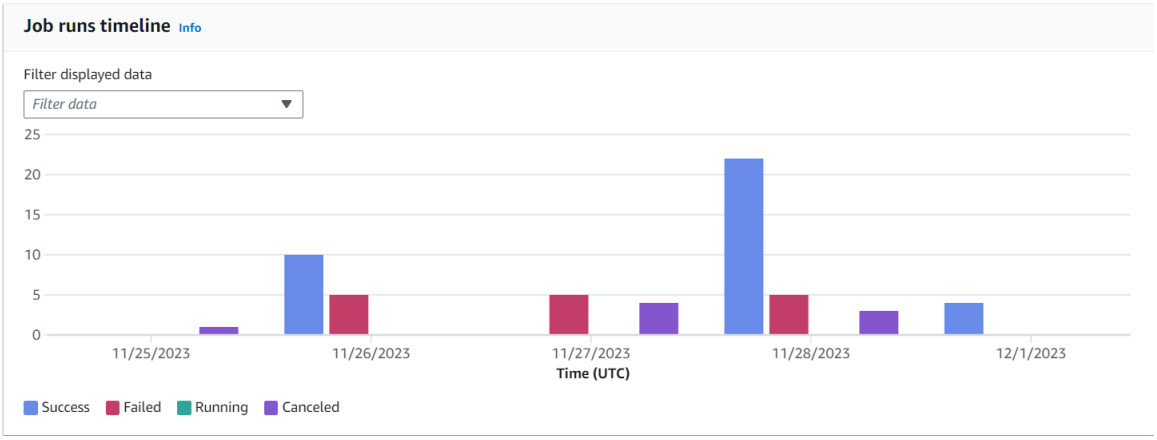
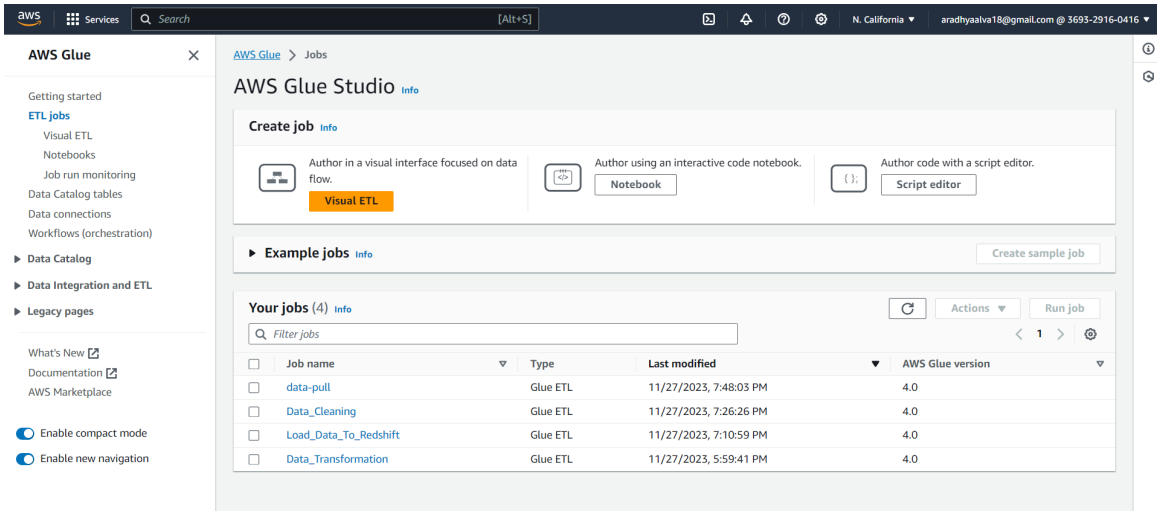


Figure 14

ETL Jobs Created



Below is the code for the ETL job ‘data-pull’ which is written in the AWS Glue provided Jupyter Notebook. A spark session is created and the required libraries are being invoked. The API url is being provided for the necessary input calls to be performed along with the necessary s3 bucket path mentioned in the source code. This job is to pull recent data from the given API and store it in the mentioned s3 bucket. Each time the job is initiated, fresh data is being pulled from the data portal and is being replaced in the s3 bucket. According to the run status, it roughly takes about a minute for the job to succeed.

Figure 15

(ETL) 'data-pull' Job

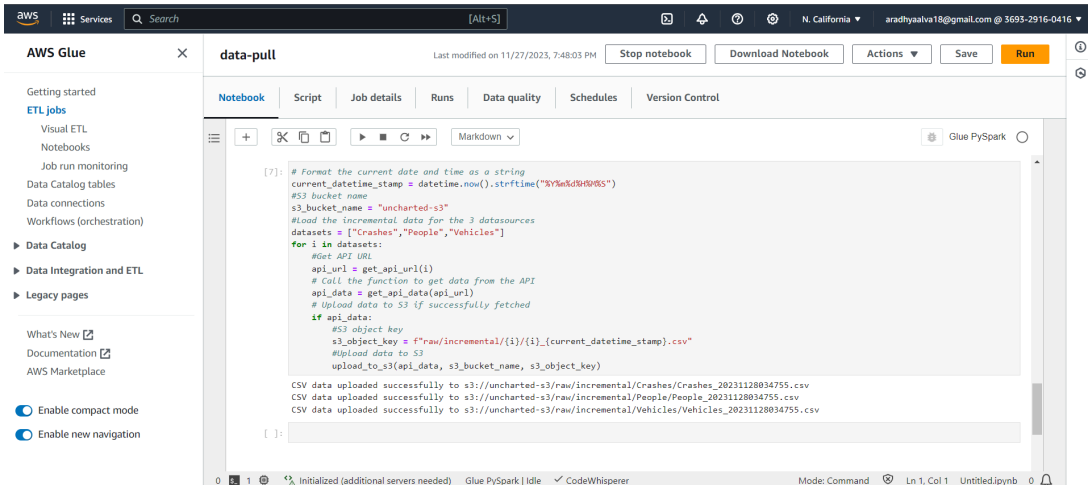
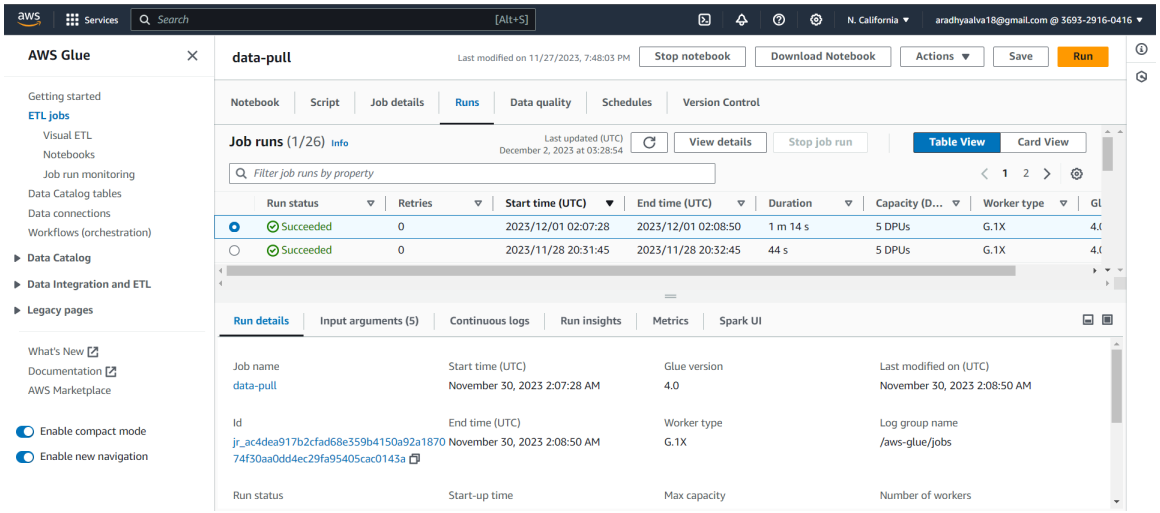


Figure 16

Run Status of 'data-pull'



Below is the code for the ETL job 'Data_Cleansing' where all the null values are handled, duplicate rows are removed, outliers are taken care of and irrelevant data formats are extinguished. The values are being thoroughly sought through and ensured that they are of suitable types. As is visible from the figure below, the property parameters for the particular job are given as follows and are being maintained similarly for all the ETL jobs.

Figure 17

(ETL) 'Data_Cleansing' Job

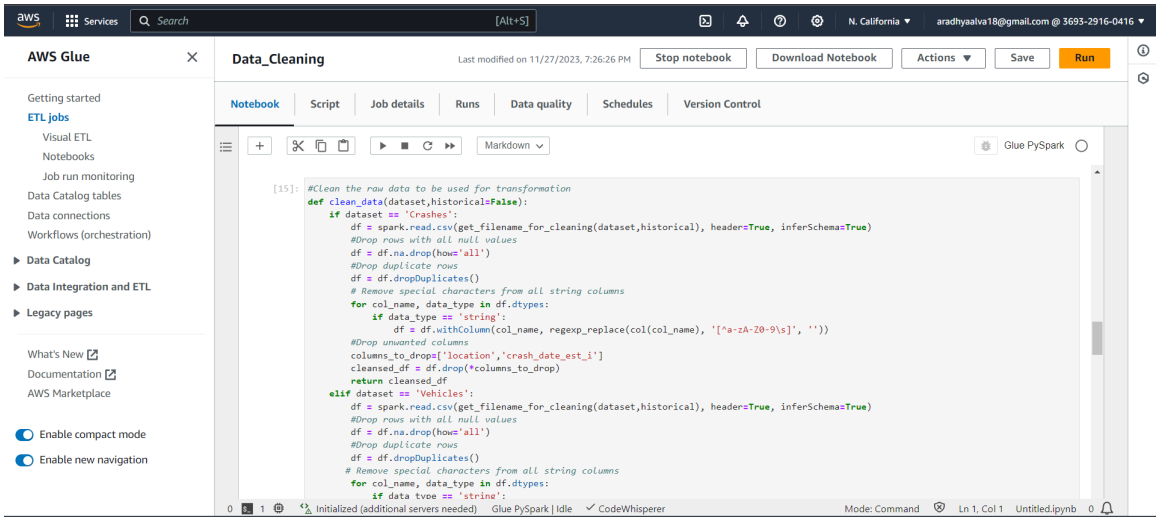


Figure 18

Basic Properties of 'Data_Cleansing'

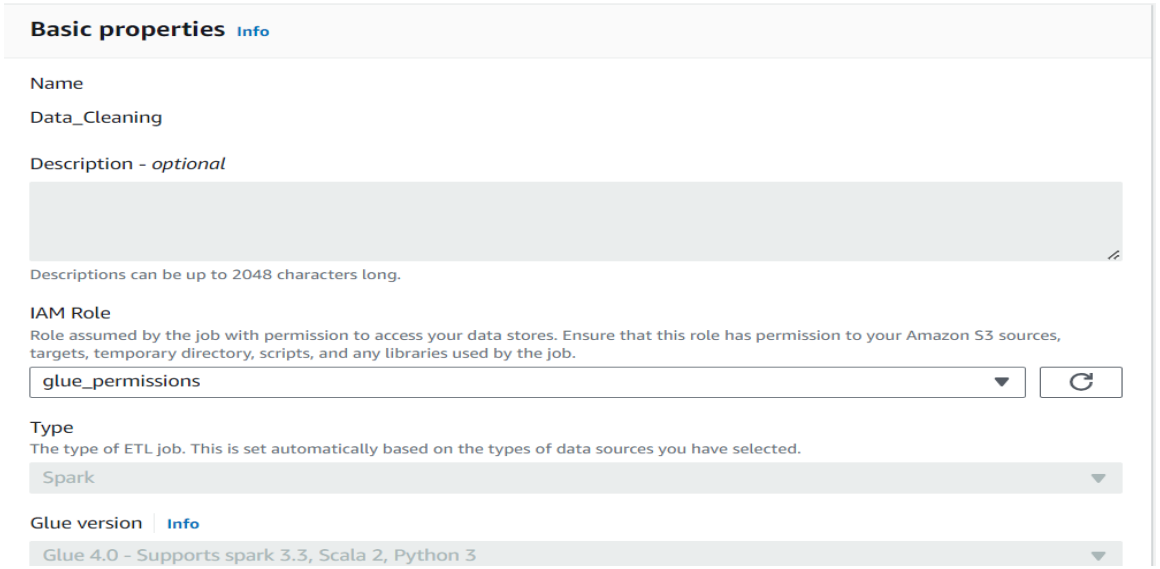


Figure 19

Properties of 'Data_Cleansing' Extended

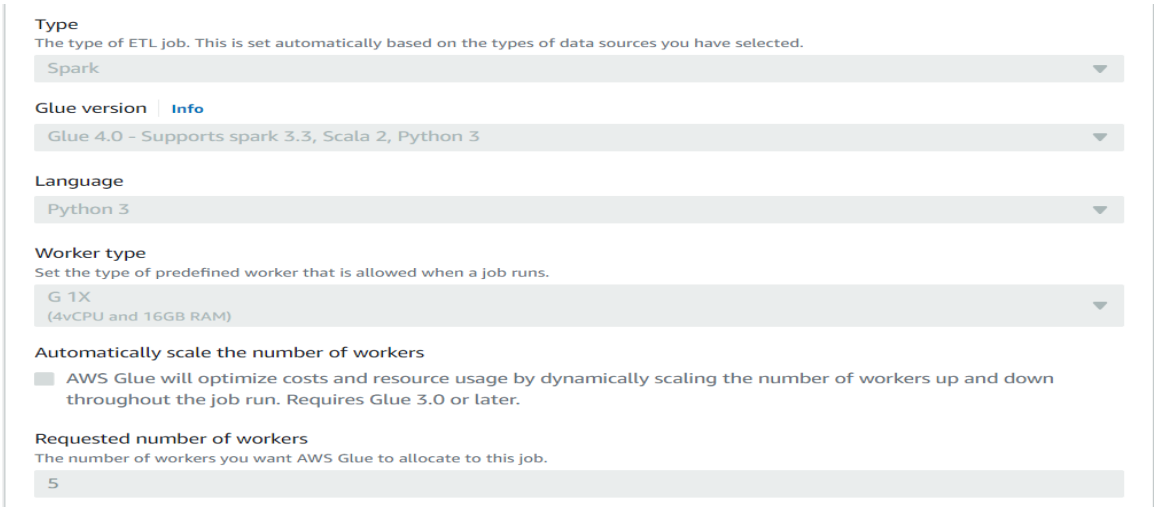


Figure 20

Properties of 'Data_Cleansing' Extended

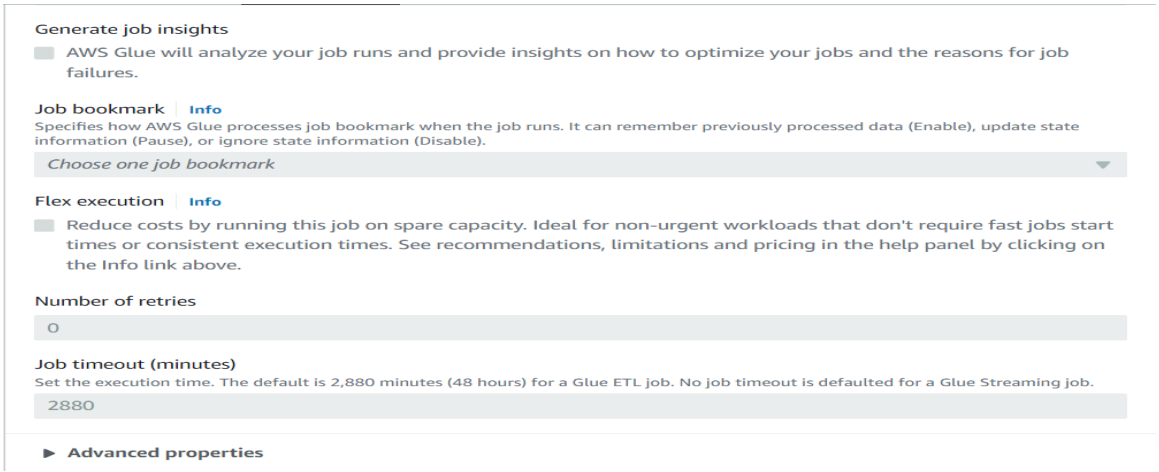
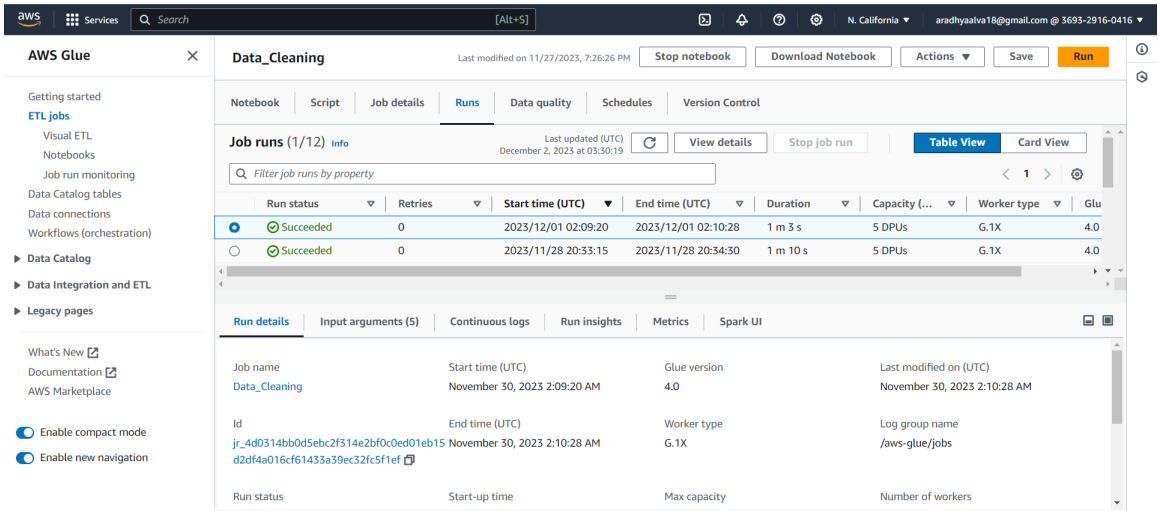


Figure 21

Run Status of 'Data_Cleansing'



Similar to the above jobs and run status 'Data_Transformation' and 'Load_To_Redshift' jobs have also been executed under the same job properties. It is seen that the job utilizes the same worker type which is the 'G.1X' version and uses the capacity of 5 DPUs.

Figure 22

(ETL) 'Load_To_Redshift' Job

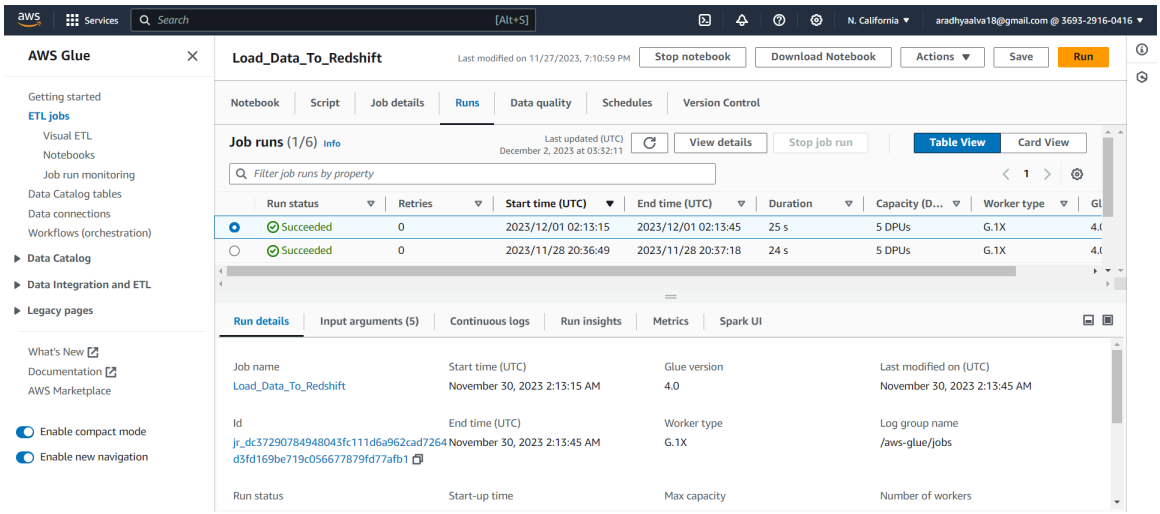
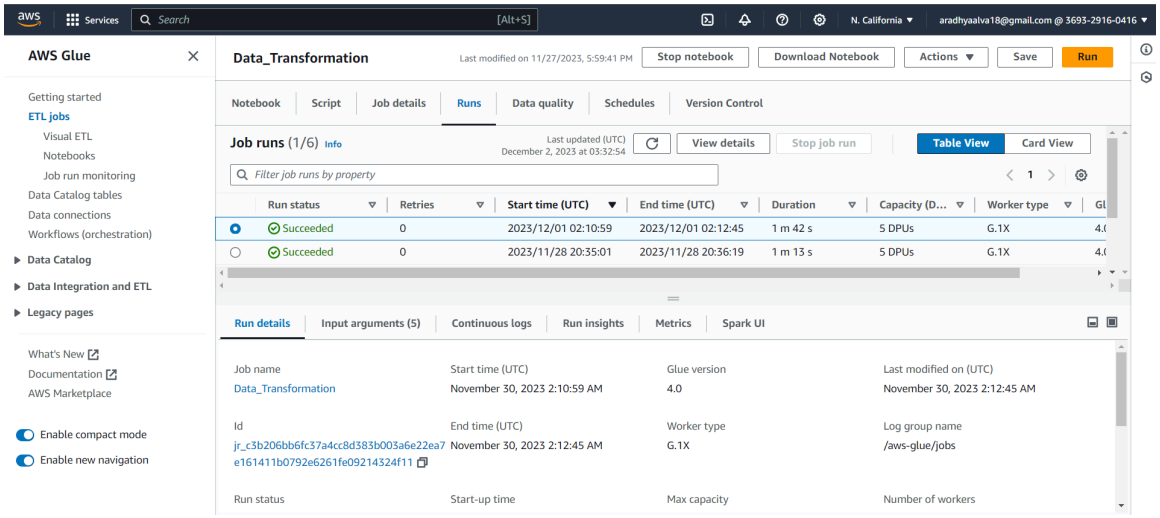


Figure 23

(ETL) 'Data_Transformation' Job



The 'Load_To_Data_Redshift' glue job is a critical component in the data integration process, enabling the mapping of data from the s3 folder to the redshift database. Utilizing the AWS Glue Data Catalog tables, the schema can be transformed and subsequently loaded into the data warehouse. Visual ETL provides an evident flow of data, enabling efficient analysis and processing. The crawler database is selected as the source, while the source key and target key are kept the same to avoid errors and ensure smooth mapping. However, the data types had to be changed to enable proper mapping and avoid conceptual difficulty. In particular, the conversion of data types such as 'bigint', 'double', and 'float' into integer data types helped reduce memory utilization and enhance processing efficiency.

Furthermore, to ensure that the data is stored in an appropriate format for analysis and querying in the data warehouse, the dates are converted from the 'object' data type to 'DateTime'. This conversion enhances the accuracy of the data, making it easier to analyze and query, ultimately increasing the quality of the output. Such changes are critical in protecting the integrity of data, which is essential for the success of any data integration process. By ensuring that the data is correctly transformed and mapped, the 'Load_To_Data_Redshift' glue job plays a vital role in the overall success of the data integration process.

Figure 24

Taking Data from Catalog Tables

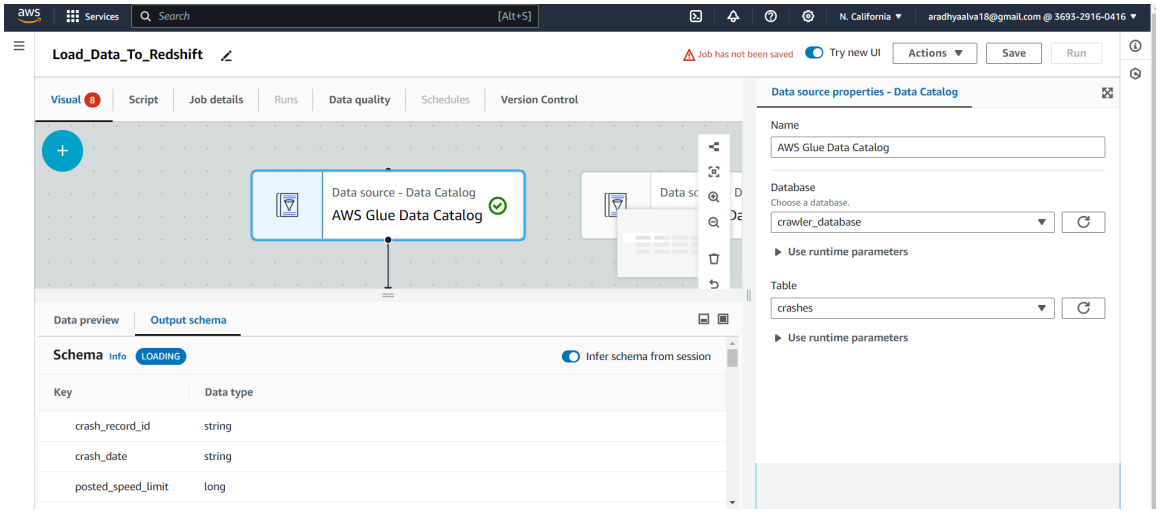


Figure 25

Changing the Schema

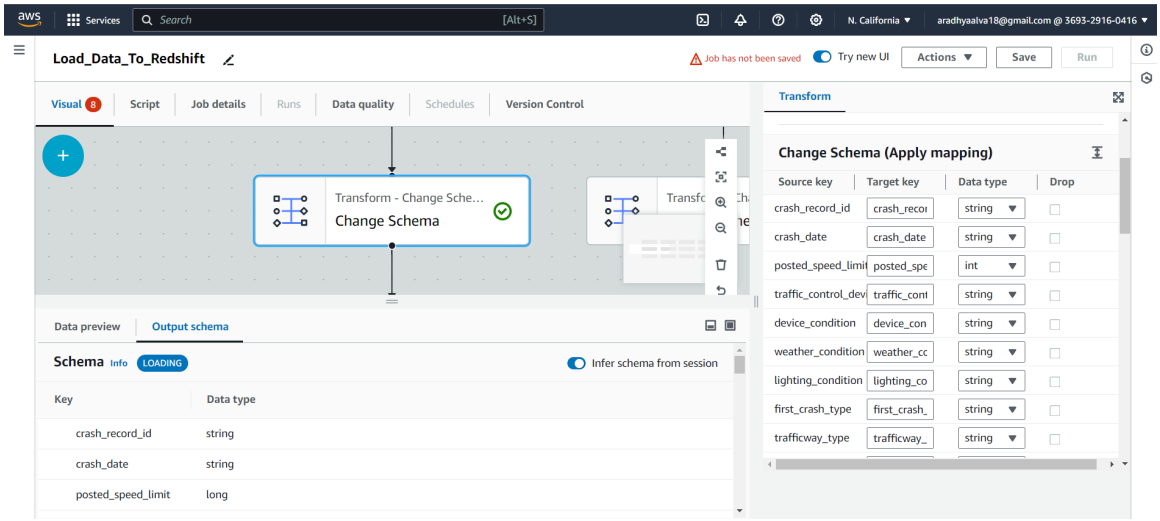


Figure 26

Loading in Redshift

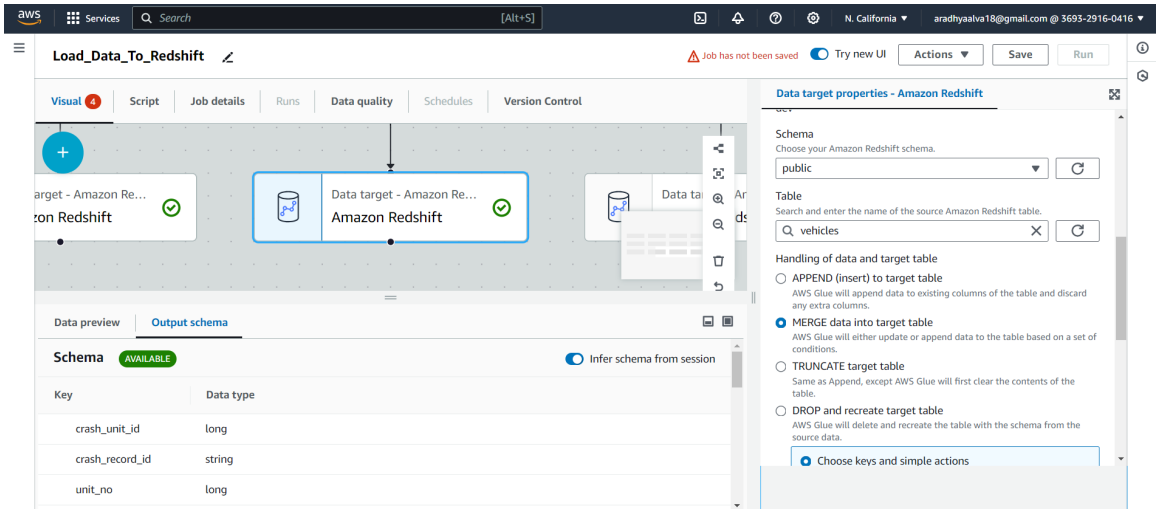
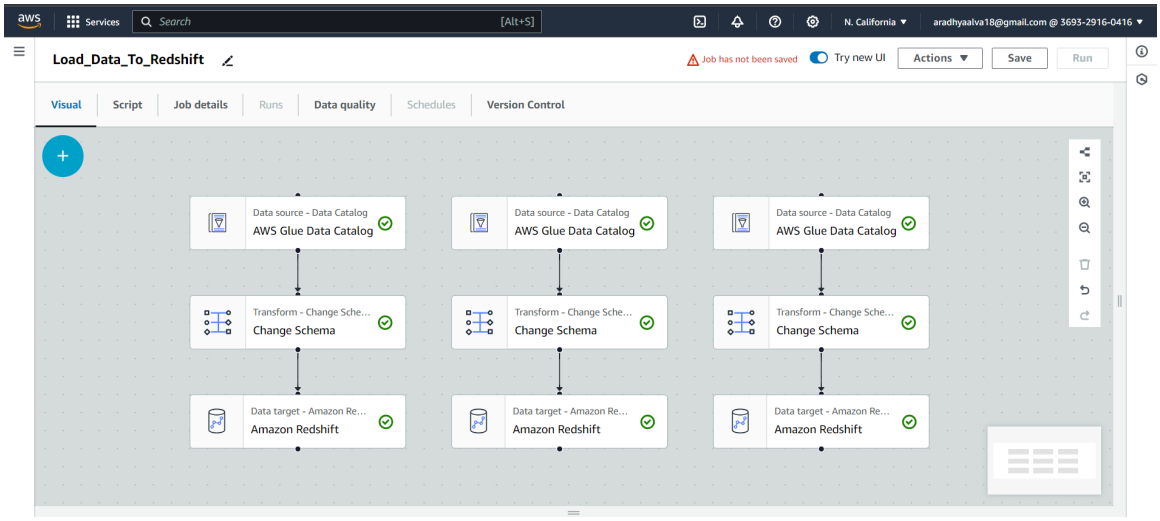


Figure 27

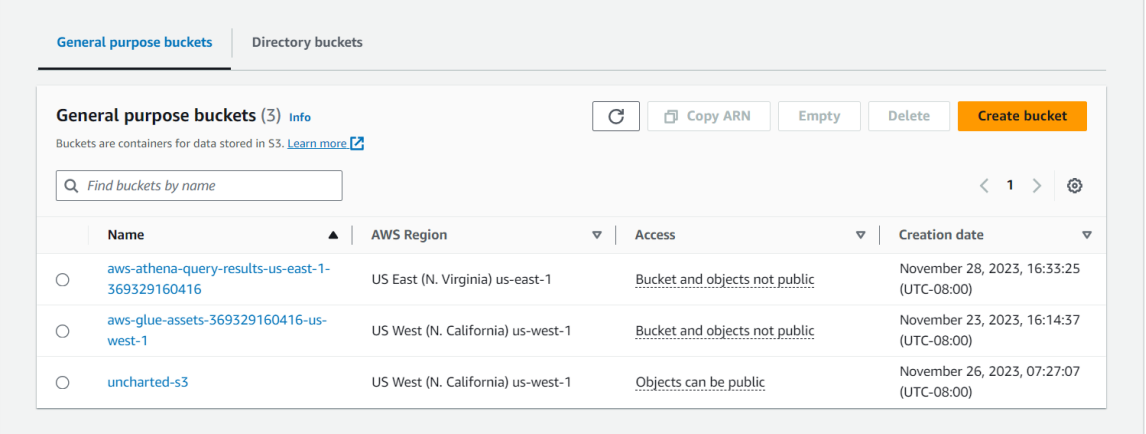
Architecture of Mapping



The figure below provides an overview of the s3 buckets used in the project. There are three buckets in total: one bucket for storing metadata from Athena, one for storing metadata from AWS Glue, and a main bucket named 'uncharted-s3' that serves as the staging layer for the project. The 'uncharted-s3' bucket is the primary bucket used in the project and contains multiple folders, each serving a specific purpose. These folders help organize the data and enable efficient processing. The use of multiple folders also

helps to reduce the complexities involved in managing large volumes of data. The use of these s3 buckets is critical to the success of the project. By storing metadata in separate buckets, the project can achieve better organization and management of data, while the 'uncharted-s3' bucket serves as the primary staging layer, enabling efficient processing and analysis of data.

Figure 28
AWS S3 Interface



The diagram below illustrates the organization of the 'uncharted-s3' bucket, which is used as the primary staging layer for the project. The bucket is divided into several folders, each serving a specific purpose and playing a crucial role in organizing and managing the data effectively.

The 'Athena' folder is used to store the results of queries made in Amazon Athena. It is essential to store the results in a secure location to avoid any loss of information or analysis. The storage of data is divided into three primary folders: 'clean', 'raw', and 'transformed.' The 'raw' folder is used to store all the data pulled from the Chicago Data Portal and stage it in AWS S3 using the ETL job 'data-pull.' Once the data is staged in the raw folder, the ETL job 'Data_Cleansing' takes the data from the raw folder, performs cleansing and stores the data in the 'clean' folder.

The 'clean' folder contains two subfolders, namely 'historical_data' and 'incremental.' The 'historical_data' folder contains data from 2015 to the present, while the 'incremental' folder contains recent data being pulled from the API. Both these folders have three subfolders inside them, namely 'Crashes,' 'People,' and 'Vehicles.' These subfolders categorize the data, making it efficient for analysis and convenient to handle.

The 'transformed' folder contains data that is taken from the 'clean' folder and transformed into business requirements by the ETL job 'Data_Transformation.' This is the

final folder where the ETL job 'Load_Data_To_Redshift' maps the data present in this folder to the schema present in the Redshift database.

All-inclusive, the use of these folders is critical in organizing and managing the data effectively, enabling efficient processing and analysis. The naming conventions used for the folders and subfolders are intuitive and consistent with the requirements they serve, making it easier for users to understand and work with the data.

Figure 29

'uncharted-s3' Bucket Content

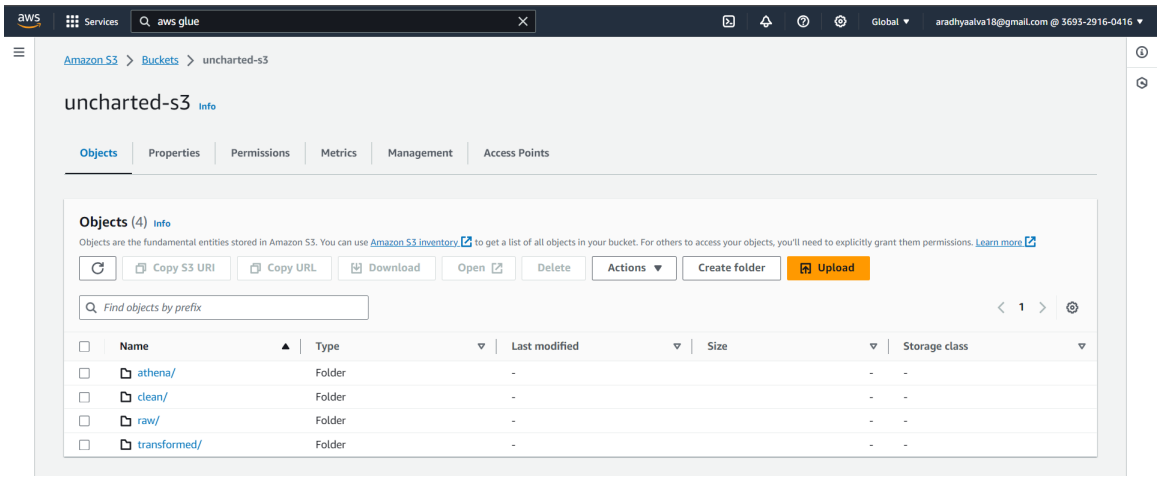


Figure 30

'clean' Folder

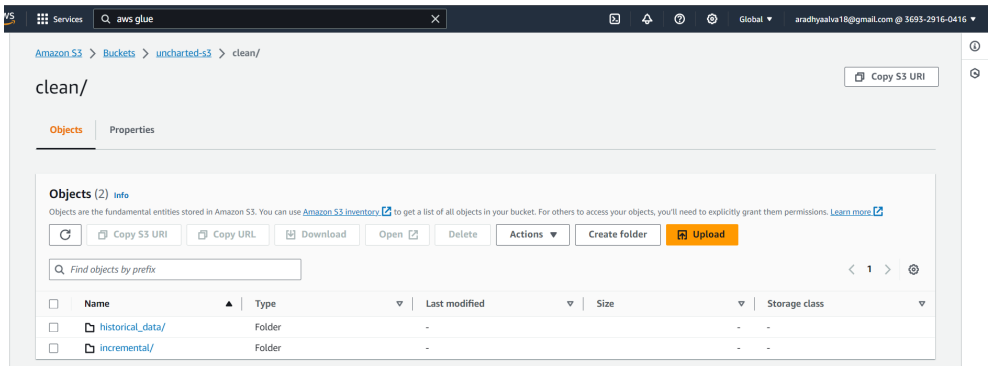


Figure 31

'historical_data' Folder in 'clean' Folder

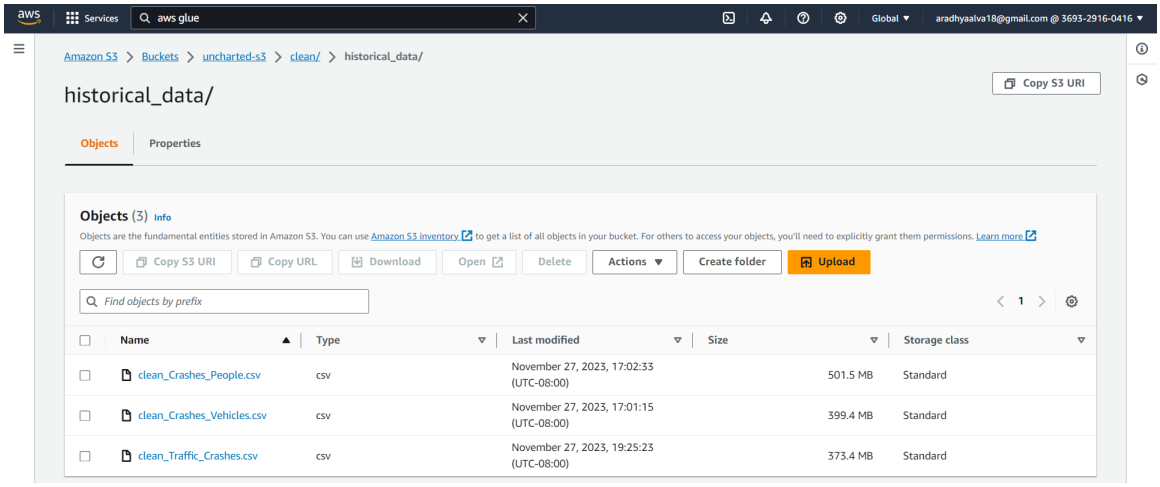


Figure 32

'incremental' Folder in 'clean' Folder

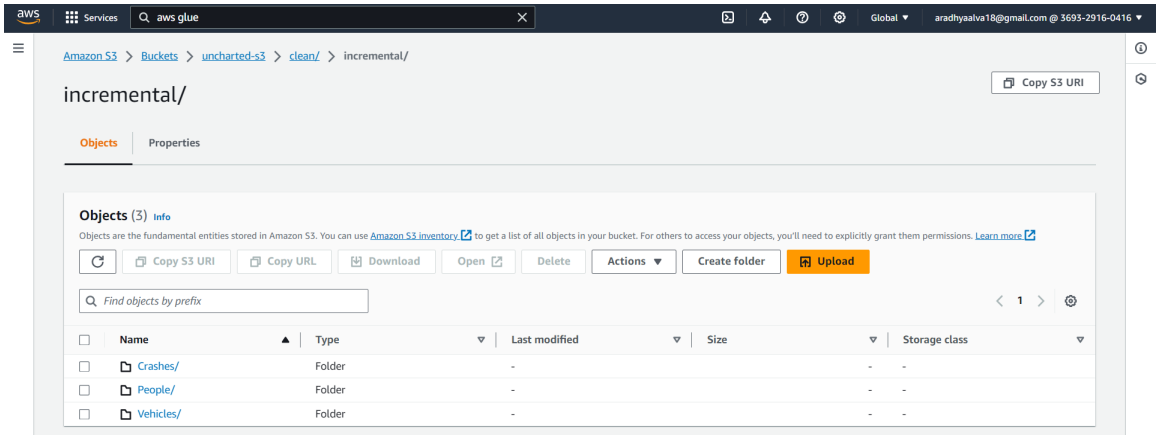


Figure 33

'crashes' Folder

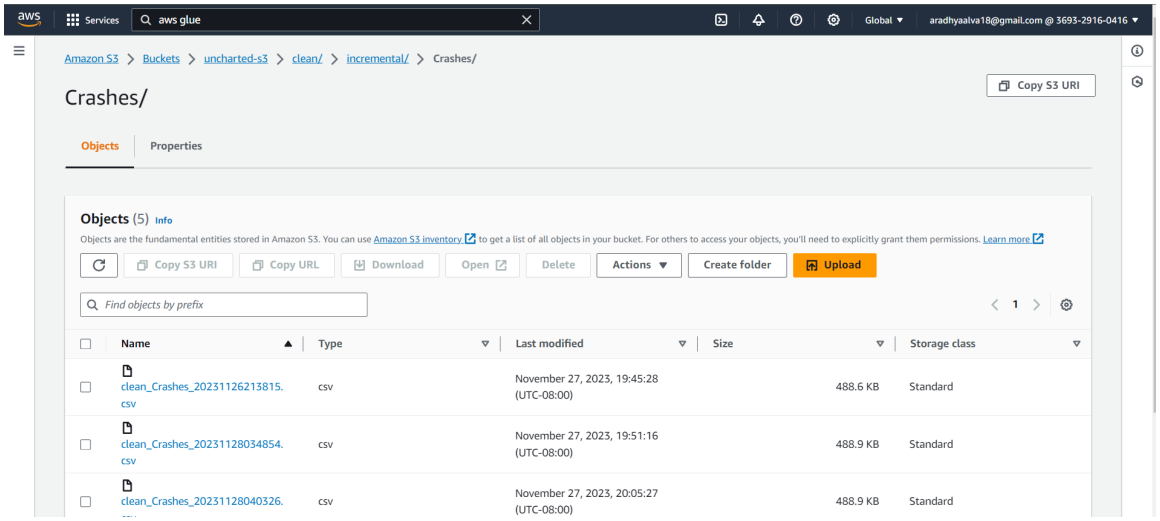


Figure 34

'People' Folder

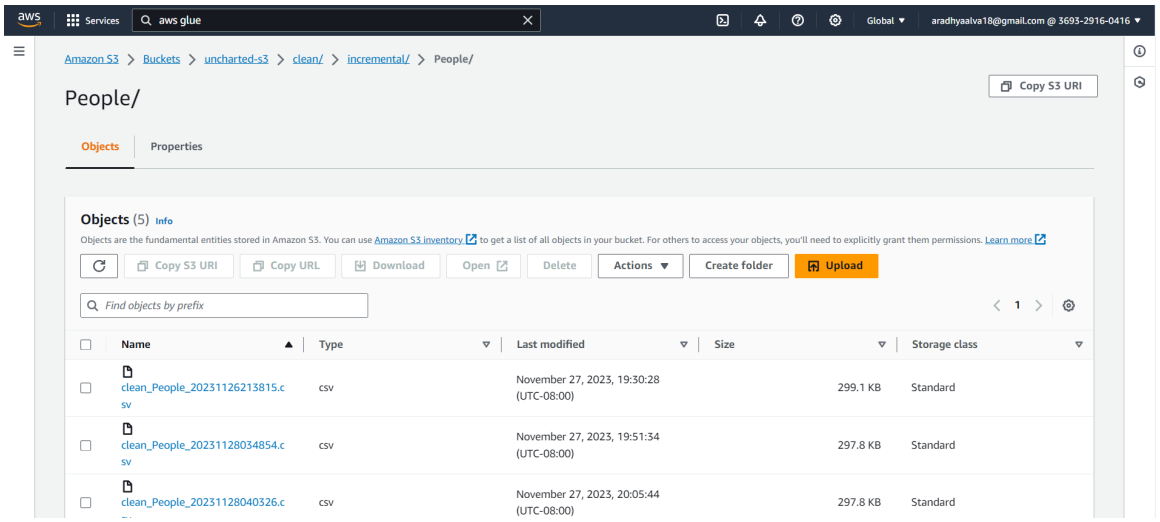


Figure 35

'Vehicles' Folder

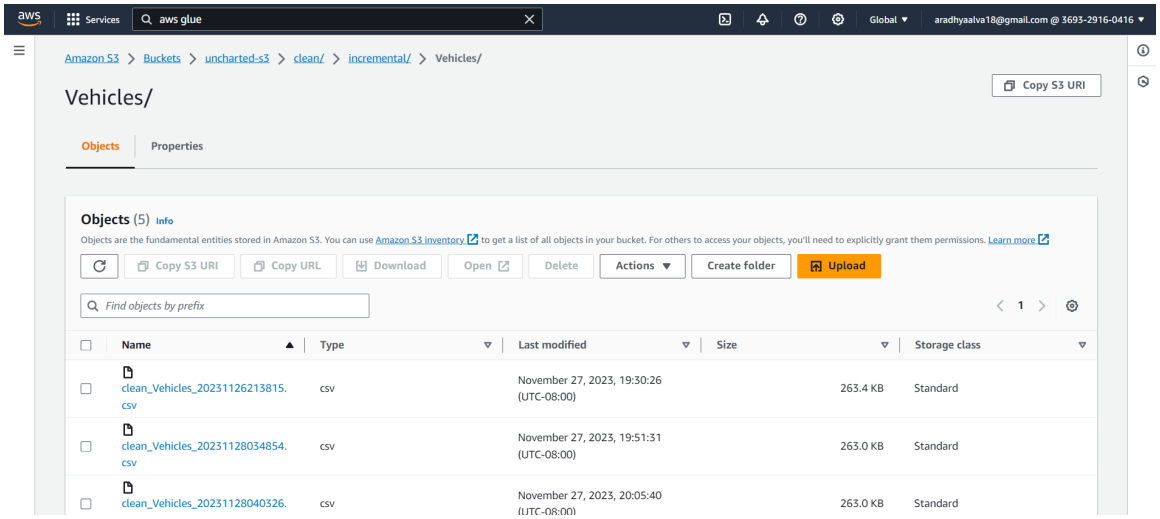
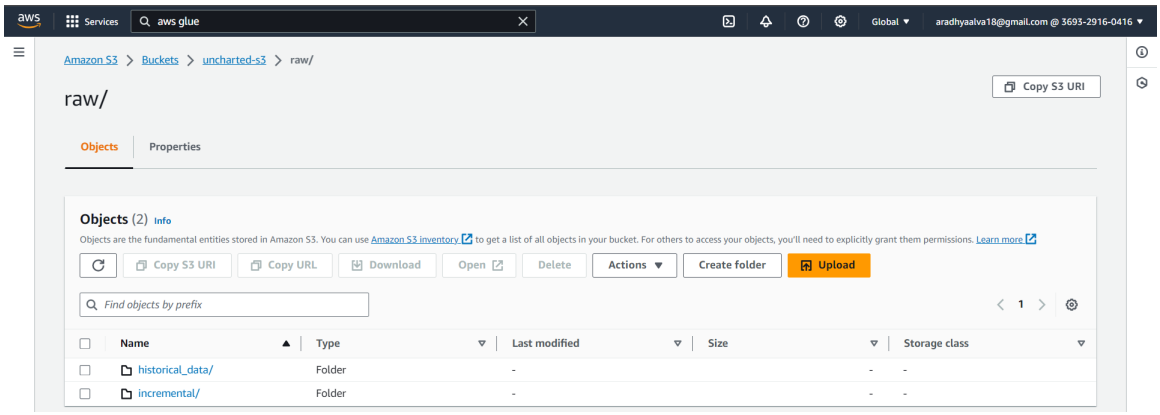


Figure 36

'raw' Folder



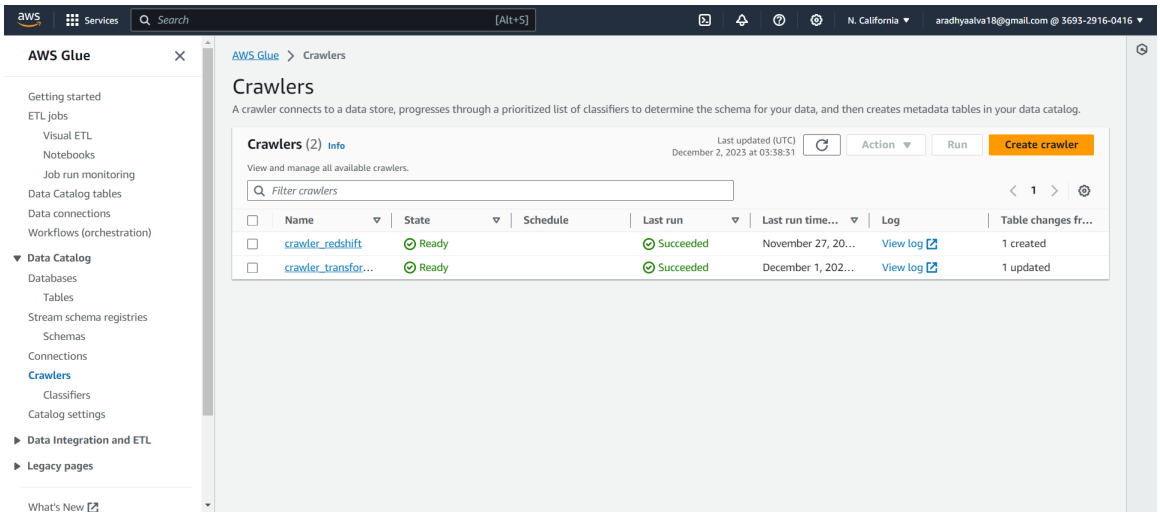
The AWS Crawler is a tool that is widely used in data integration pipelines and is being leveraged to gather the schema of both Redshift database tables and transformed data in the S3 bucket. This crucial step is necessary to facilitate the mapping of the data during the loading process into the data warehouse. By gathering the schema of the data, the crawler provides a comprehensive view of the data structure and its attributes, which can subsequently be used to optimize queries and improve data integration performance.

In addition to schema gathering, the AWS Crawler also creates data catalog tables that store the metadata of the data obtained. The metadata includes information such as column names, data types, and data sources. This data catalog can be utilized to query and test the integrity of the data obtained in AWS Athena - an interactive query service that allows users to analyze data in S3 using standard SQL.

The use of AWS Crawler in data integration pipelines provides several benefits. Firstly, it simplifies the mapping process by providing a comprehensive view of the data structure. Secondly, it automates the process of creating data catalogs, which can be used for future reference or to ensure the integrity of the data. Finally, the combination of AWS Crawler and AWS Athena offers a powerful solution for querying and analyzing data in S3, enabling data-driven decision-making and insights.

Figure 37

Crawler Interface



Two crawlers - 'crawler_transformed' and 'crawler_redshift' -are enabled to gather metadata for the transformed data and the Redshift database tables, respectively.

The 'crawler_transformed' is specifically used to extract the schema of the transformed data. This tool works by scanning the data in the target location and inferring the schema based on the structure and content of the files. Once the schema is extracted, it is used to create a table in the Redshift database that can store the transformed data.

On the other hand, the 'crawler_redshift' is used to extract the schema of the Redshift tables. This tool works by scanning the database and inferring the schema based on the table definitions and data types. The schema extracted by the 'crawler_redshift' is used to ensure that the transformed data is compatible with the database tables.

Once the schema of the transformed data and the Redshift tables is collected, the 'crawler_transformed' is used to create three different tables - 'crashes', 'people', and 'vehicles'. These tables are mapped to the category of data they belong to, and the transformed data is loaded into the corresponding table. This ensures that the transformed data is organized and easily accessible for further analysis and querying.

Figure 38

Properties of 'crawler_transformed'

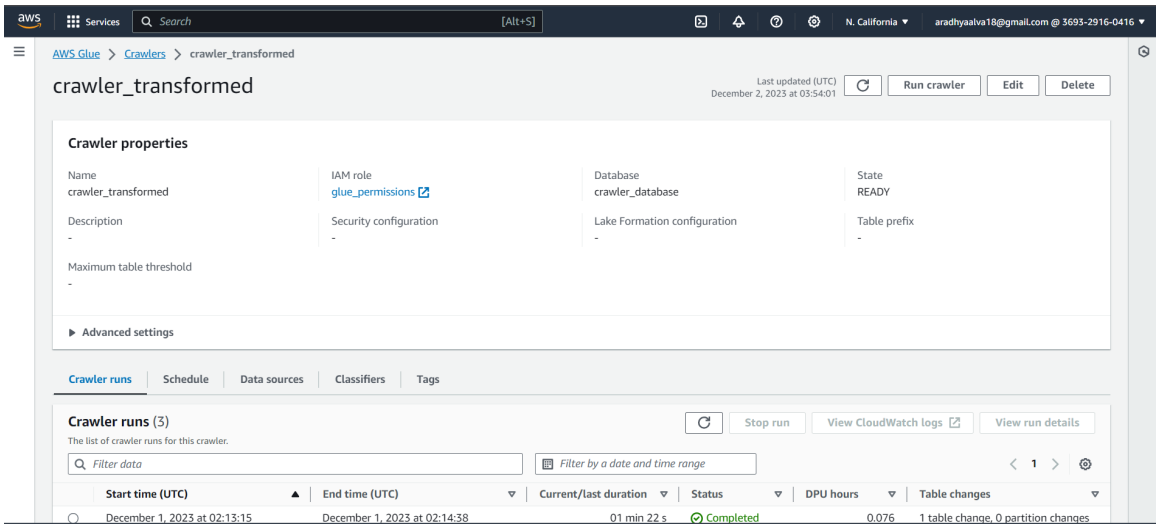
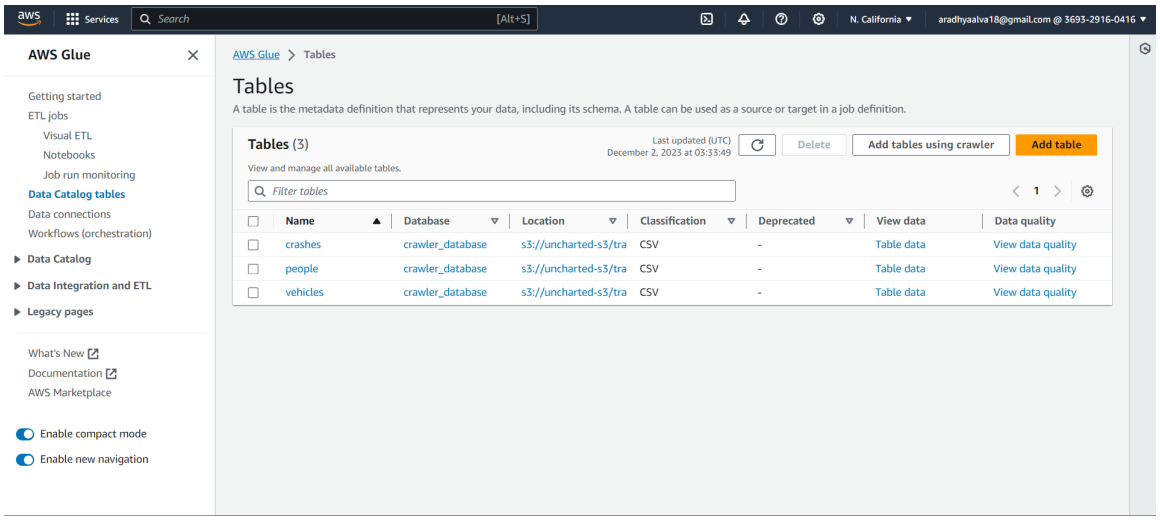


Figure 39

AWS Data Catalog Tables



The schema of each table obtained by running the 'crawler_transformed' on the transformed data is shown in the figures below. These schemas are then monitored closely to analyze their compatibility with the database design and to gather the change requirements necessary to support the design. In particular, any discrepancies or inconsistencies between the transformed data schema and the database design are identified and addressed to ensure that the data can be loaded and queried correctly. By utilizing the crawlers to extract the schema of the transformed data and the database tables, and by monitoring and analyzing the compatibility of these schemas, it is ensured that the ETL process is robust and efficient and that the data is loaded and stored correctly for further analysis and querying.

Figure 40

Schema of 'Crashes'

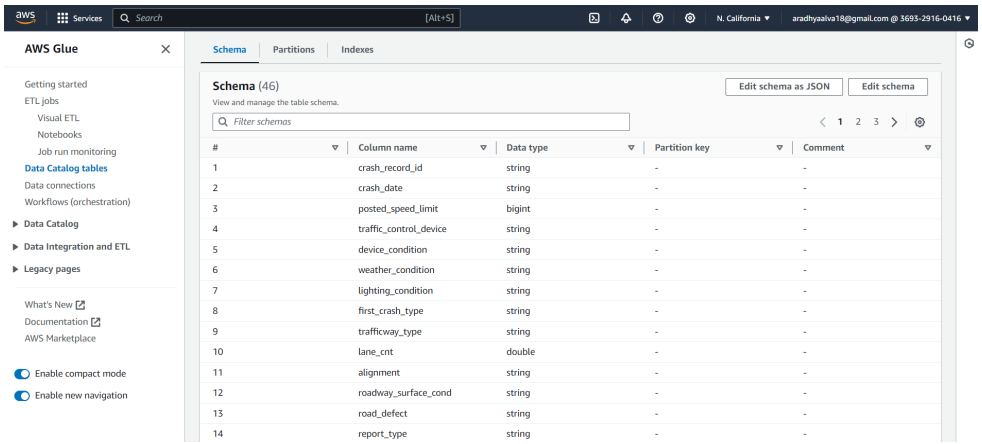


Figure 41

Schema of 'People'

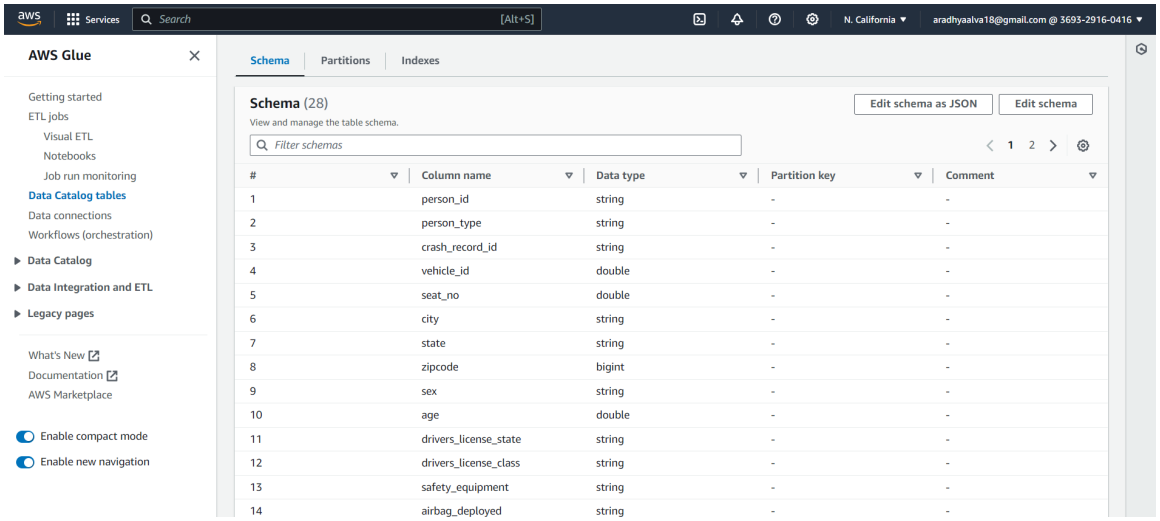
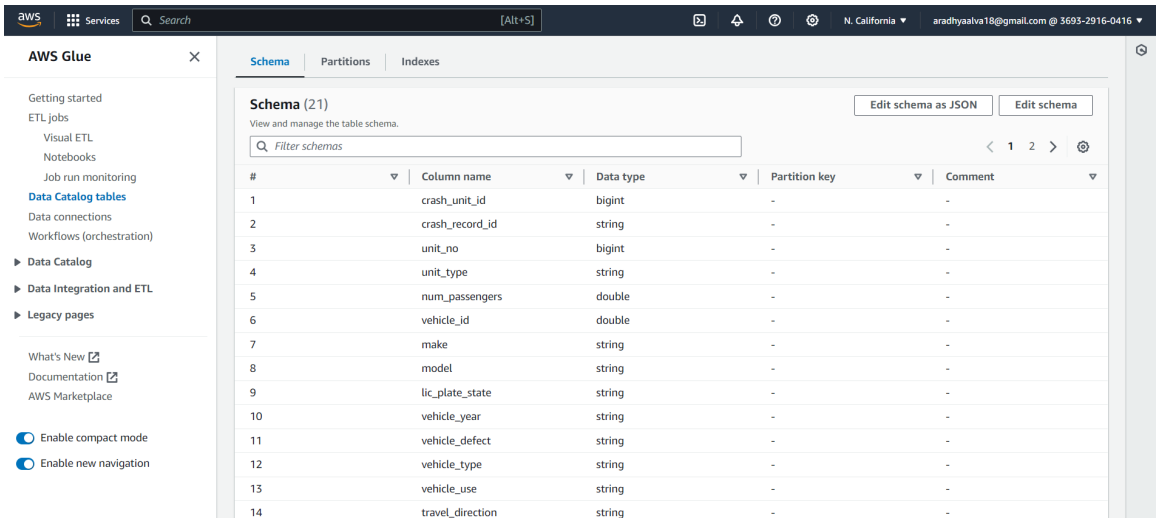


Figure 42

Schema of 'Vehicles'



AWS Redshift is utilized as the data warehouse solution for the project. A new database called 'dev' is created in the 'public' environment, with a schema established for database operations. The configuration of the Redshift cluster is set to limit the number of nodes to two, with 'ra3.4xlarge' being the selected node type. This node type is chosen

based on the requirement for high levels of CPU and memory resources for database operations.

Automated snapshots are enabled to ensure the protection of critical data against accidental loss. However, manual snapshots are not enabled, which could have provided a further layer of security, allowing for specific states of the database to be saved for future reference but is not necessary for the project. The figures below display the properties of the Redshift cluster, including its metric performance and query overview. It is a useful tool for monitoring the performance of the cluster and identifying any issues that could have affected the performance of database operations.

Figure 43

Redshift Cluster

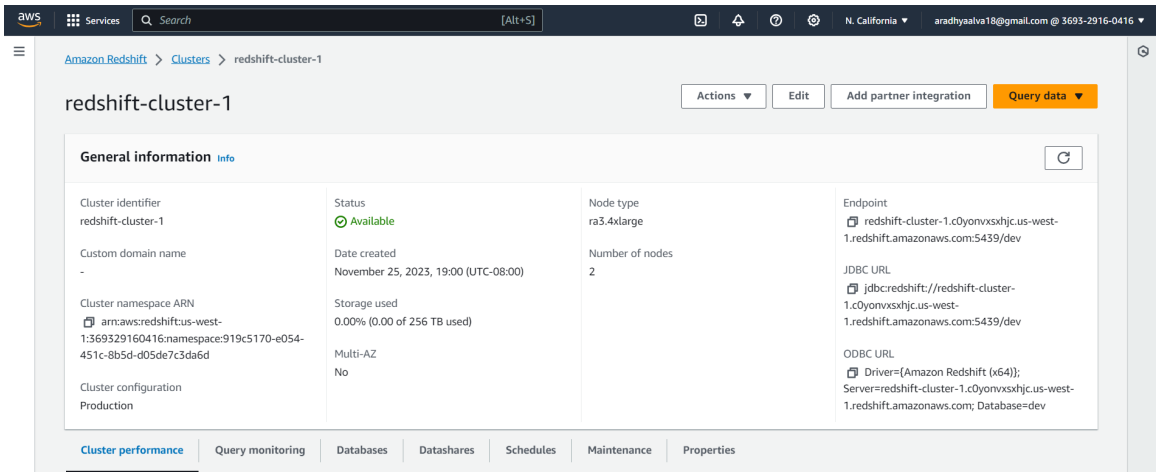


Figure 44
Redshift Dashboard

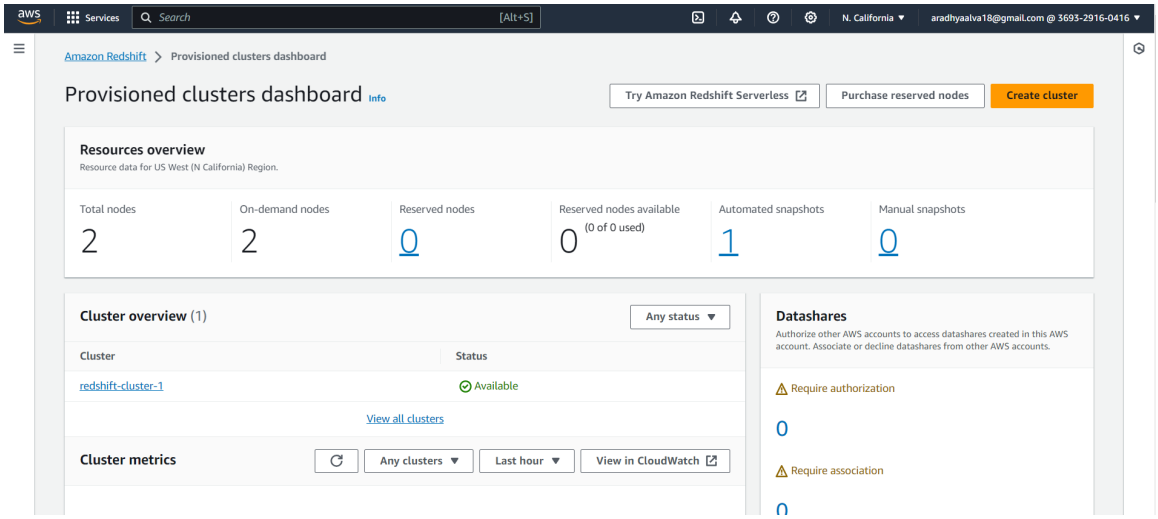


Figure 45
Cluster Metrics

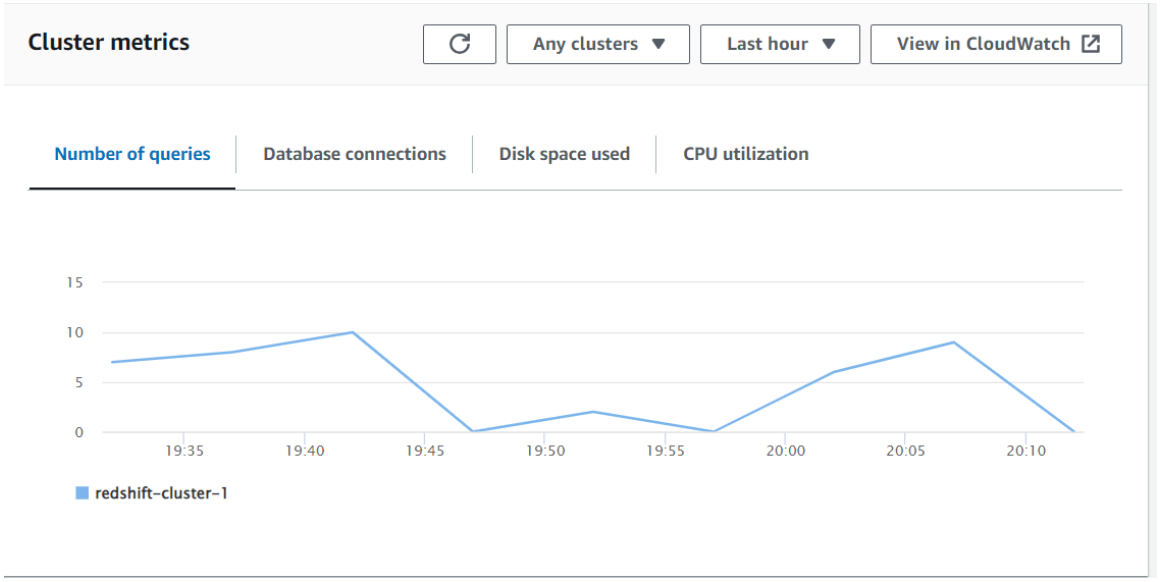


Figure 46
Cluster Metrics Cont'd

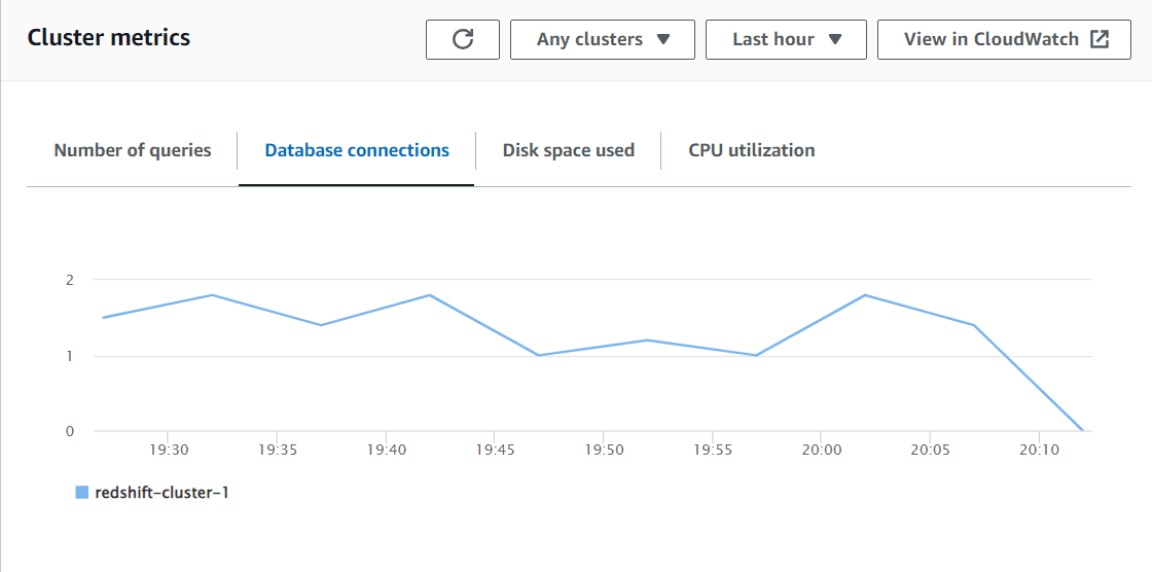
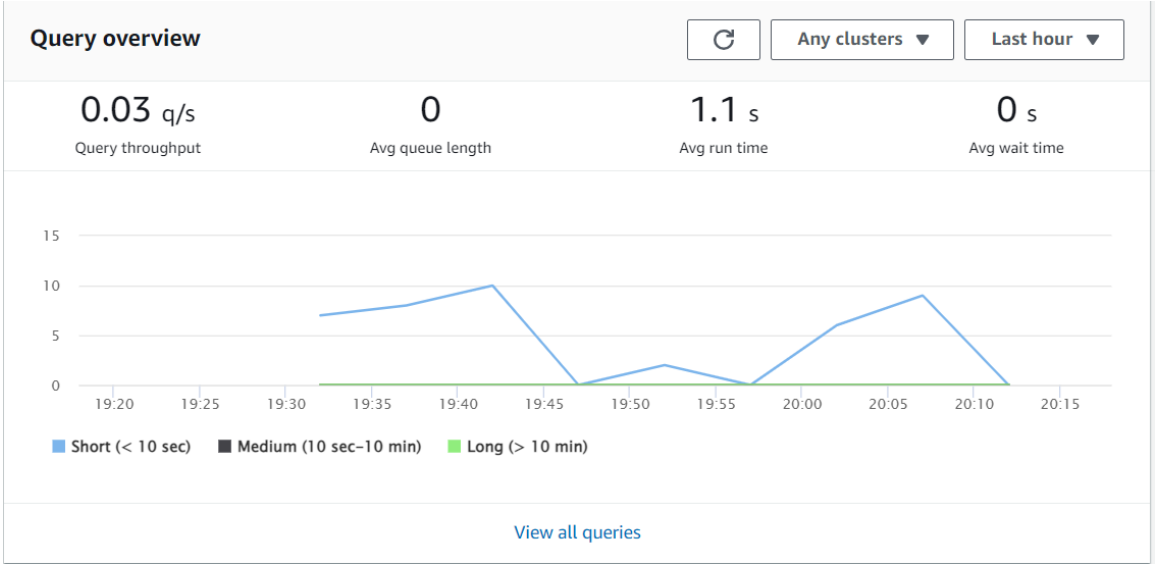


Figure 47
Query Run Review



The figure below illustrates a Redshift database created in the public environment, composed of four tables that describe three dimensions and one fact table. Specifically, the 'vehicles' and 'people' tables are mapped to two of the tables and are treated as

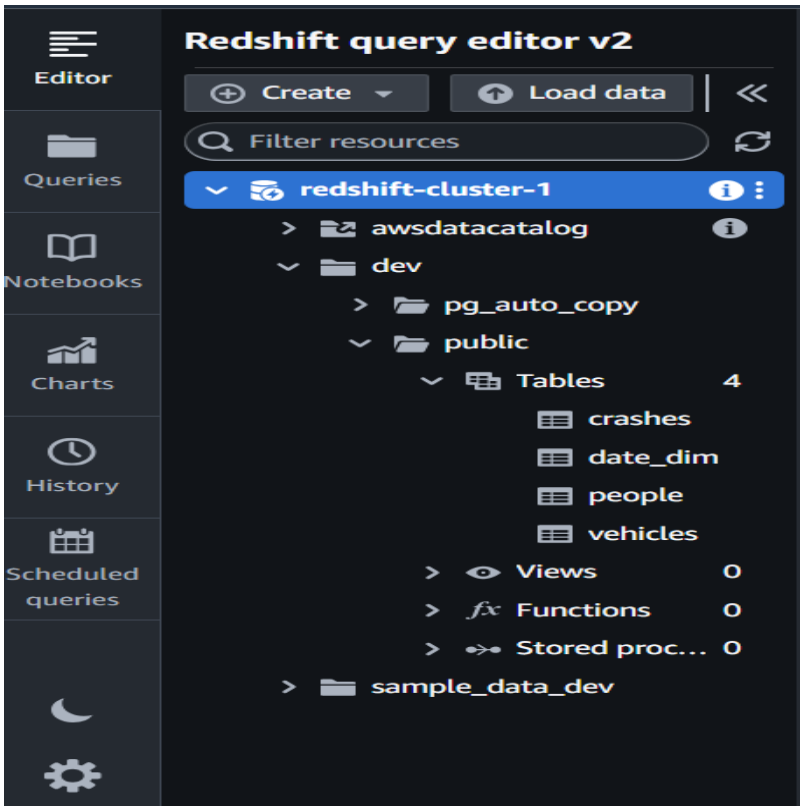
dimensions. The database also includes a date dimension consisting of attributes such as 'Date,' 'Day_of_week,' and 'Holiday,' among others, to emphasize the importance of dates in the database and enhance timeline querying.

The fact table in the Redshift database is the 'crashes' table, which is mapped into the database. This table contains various numerical attributes such as 'no_of_deaths,' 'no_of_critical,' and 'no_of_injured,' which provide information about the severity of the crashes. It is worth noting that each dimension has its primary key which uniquely identifies the instances of the table. On the other hand, the fact table contains a primary key that is a combination of all the primary keys of the dimension tables.

Including the primary keys in the design of the database is crucial as it enables connectivity among the tables and supports referencing. The use of primary keys establishes a relationship between the fact table and the dimension tables, allowing for complex analytical queries that provide insights into the relationship between crashes and the various attributes of the dimensions.

Figure 48

Redshift Database

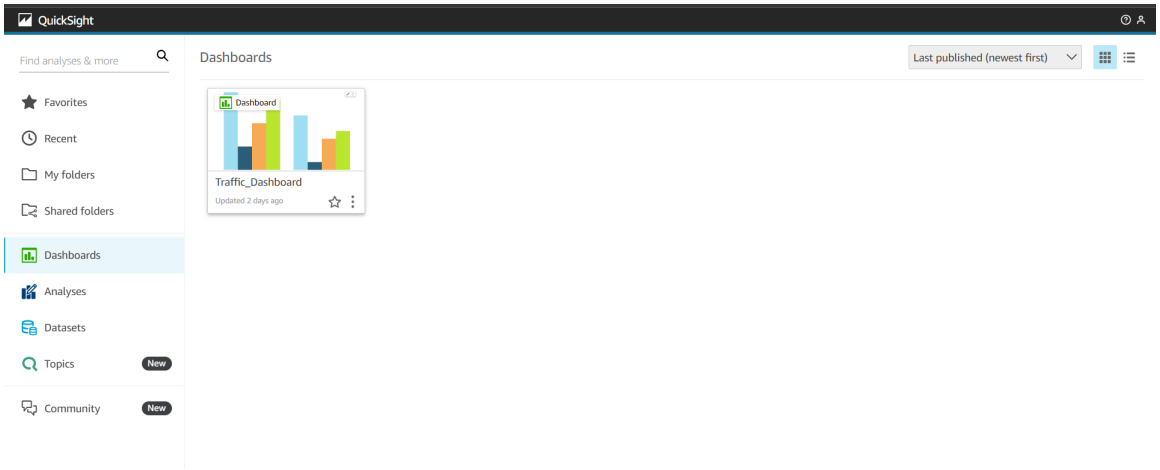


After setting up the Redshift database and ensuring its complete functionality, the next step is to integrate the data with AWS Quicksight for generating visualizations and performing further analysis. Quicksight is a powerful visualization tool provided by AWS that can be easily connected with any AWS service. To begin, a dashboard is created in Quicksight and the data is loaded from the data warehouse. Using Quicksight, meaningful and insightful visualizations are generated that bring out underlying patterns and information from data that may not be visible when presented in tabular form. These visualizations can help users better understand the data and make more informed decisions based on the insights gleaned from the data.

Overall, the integration of the Redshift database with AWS Quicksight is a valuable step in the data analysis process, providing users with a powerful and user-friendly tool for visualizing and exploring data.

Figure 49

AWS Quicksight Dashboard



Proactive measures are taken to manage AWS costs responsibly. AWS cloud services are charged monthly, and it is wise to be mindful of the resources that are being used. Deleting unnecessary resources that are no longer required is a crucial step in optimizing cloud expenditure.

The process of deleting resources involves several steps, including identifying the resources that need to be deleted, removing any dependencies, and finally, deleting the resources themselves. The figures provided below help in visualizing the deletion process. They provide a clear picture of the resources that are used in the project and the steps that are taken to delete them. By following these steps, it is ensured that no unnecessary resources are being paid for.

In conclusion, by being mindful of AWS costs and taking measures to optimize the cloud expenditure, it is ensured that the resources are being used effectively and efficiently. Deleting unnecessary resources is a critical step in this process in managing the finances along with the development of the project.

Figure 50

Redshift Cluster Deletion

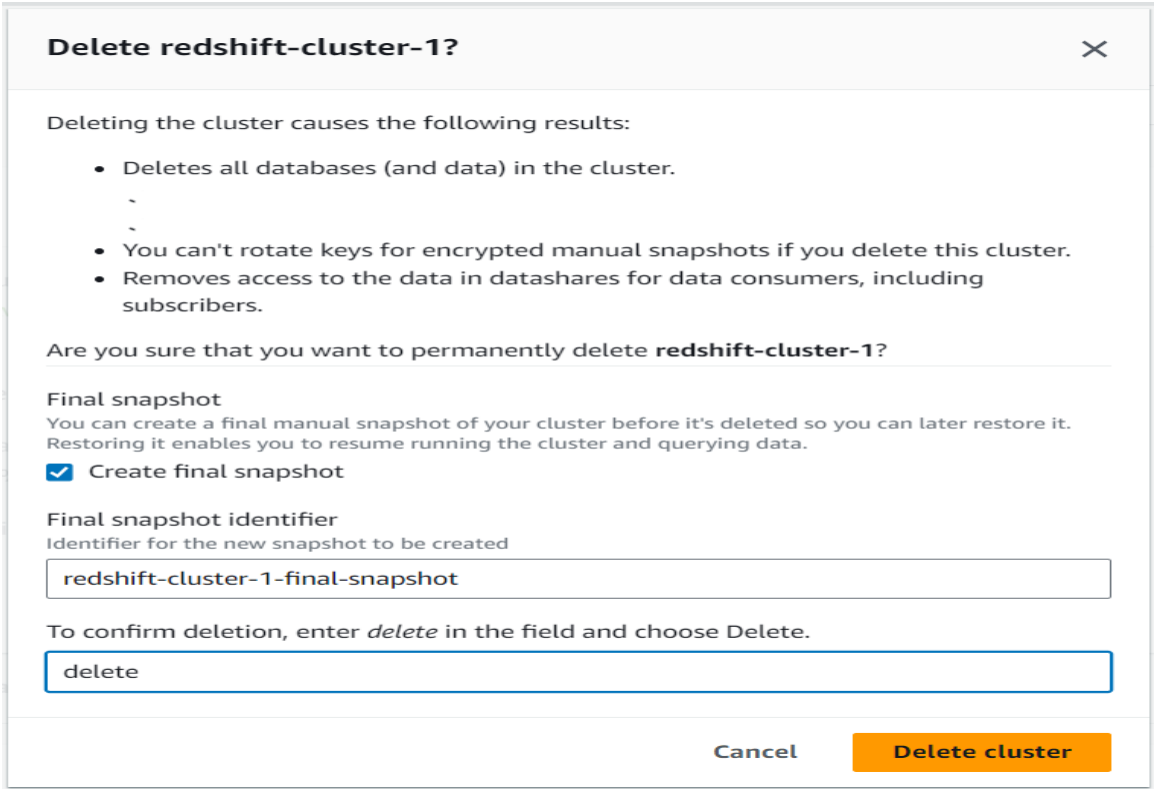


Figure 51

Cluster Deletion Confirmation

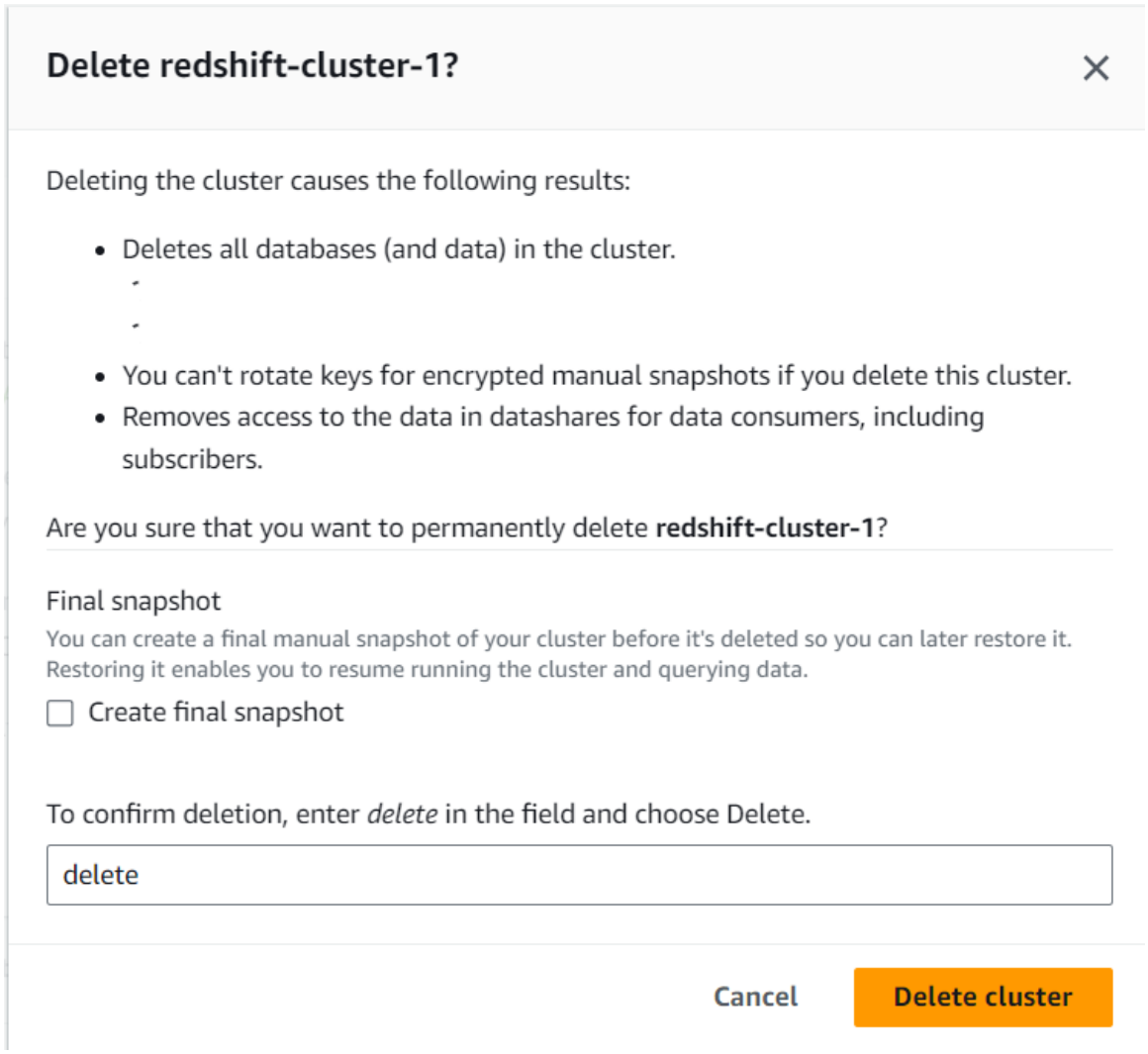


Figure 52
Crawler Deletion

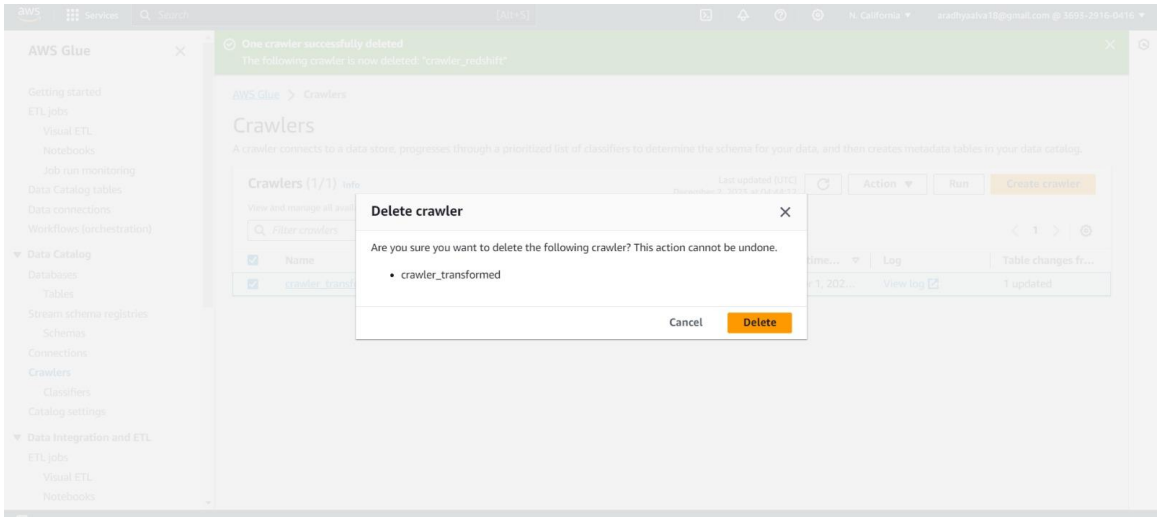


Figure 53
Database Deletion

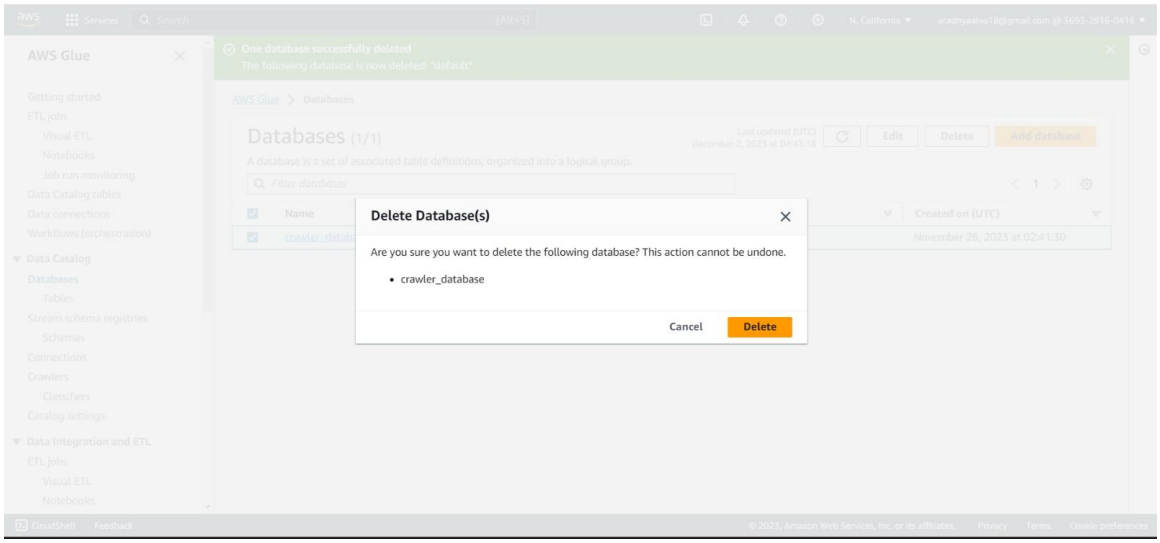


Figure 54
Workflow Deletion

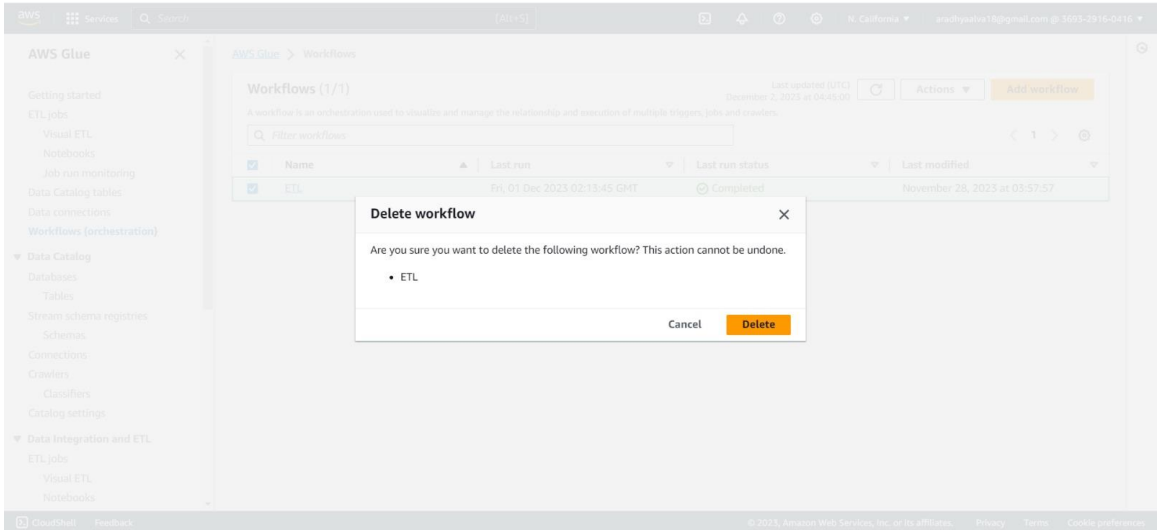


Figure 55
Bucket Deletion

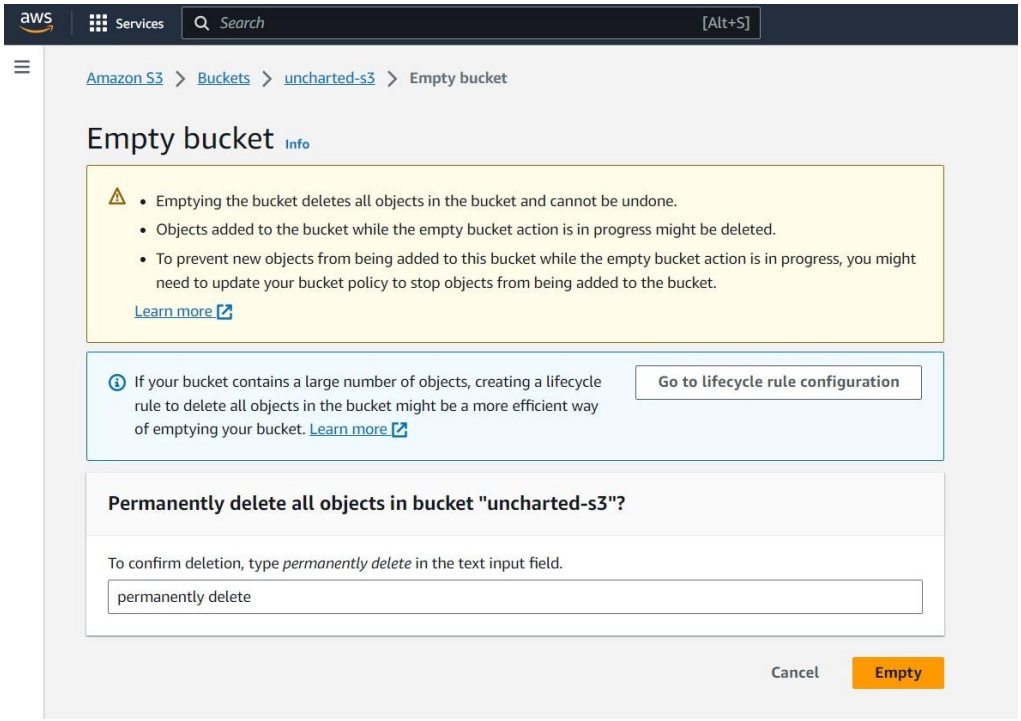
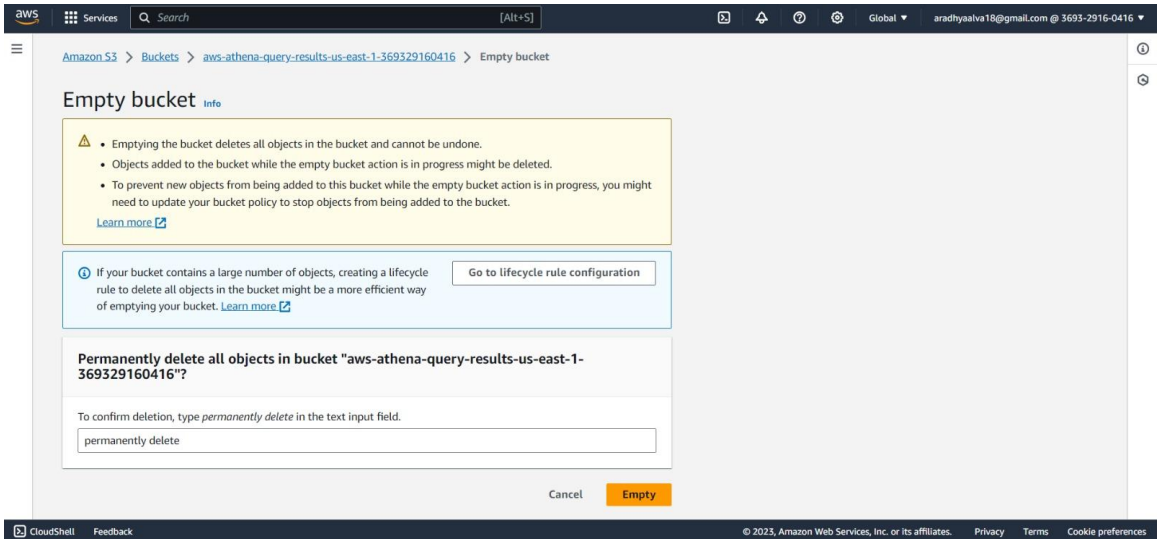


Figure 56

Bucket Deletion (Cont'd)



5.2 System implementation issues and resolutions

During the implementation of the system, a few problems are encountered which are mentioned below along with the solutions:

IAM Role Permissions.

Problem. AWS Glue requires an IAM role that assumes the policies and permissions allowing it to access data in S3 and Redshift databases. Appropriate role permissions enable the glue to take data from s3 buckets as input, process it in the glue jobs, and store it in either S3 buckets or a data warehouse.

Solution. The figures below show the permissions given to the IAM role 'glue_permissions' which enables AWS Glue to access services like S3 bucket and Amazon Redshift to access data in both services.

Figure 57

IAM Role for AWS Glue

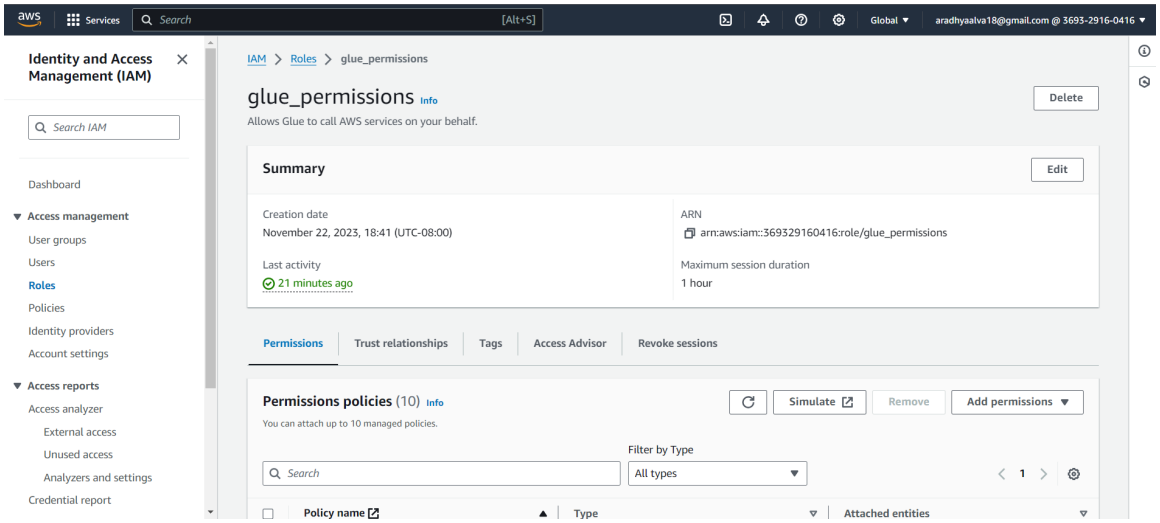


Figure 58

Policies Attached to 'glue_permissions'

Permissions policies (10) Info

You can attach up to 10 managed policies.

Simulate | Remove | Add permissions

Filter by Type

Search | All types | 1

Policy name	Type	Attached entities
AmazonAPIGatewayInvokeFullAccess	AWS managed	2
AmazonRedshiftAllCommandsFullAcc...	AWS managed	2
AmazonS3FullAccess	AWS managed	6
AWSGlueConsoleFullAccess	AWS managed	3
AWSGlueServiceNotebookRole	AWS managed	1
AWSGlueServiceRole	AWS managed	2
AWSLambda_FullAccess	AWS managed	3
AWSLambdaBasicExecutionRole	AWS managed	1
AWSStepFunctionsFullAccess	AWS managed	4
glue_notebook_iam	Customer inline	0

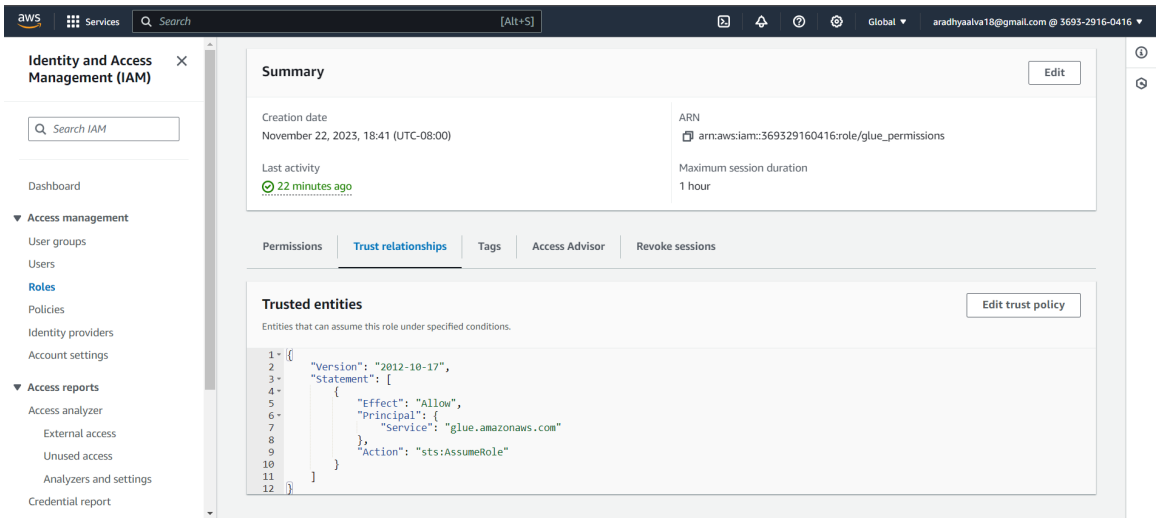
Trust Relationship.

Problem. A trust relationship problem often pertains to challenges with the trust relationship between different entities within a system, such as between domains, accounts, or services. This can manifest as an incorrect or improperly modified trust relationship defined for an IAM role, which can impede the role's ability to assume other roles or permit entities to assume it.

Solution. To resolve a trust relationship problem resulting from an incorrect or improperly modified trust relationship defined for an IAM role, the policy is reviewed and corrected. Specifically, the trust policy should accurately specify the trusted entities that can assume the role. By doing so, the role is able to assume other roles and allow entities to assume it as intended.

Figure 59

Trust Relationship Policy



VPC Endpoint Configuration.

Problem. If the Amazon Redshift cluster is situated in a private subnet and a VPC endpoint is being used for Amazon S3, the endpoint can not be configured correctly. This potentially caused issues that have to be taken care of for connecting Redshift to s3 through Amazon Glue.

Solution. The VPC endpoint is created and associated with the correct route tables. It's vital to ensure that everything is set up correctly to avoid any issues. Additionally, it is verified that the security groups associated with the Redshift cluster allow outbound traffic to the S3 VPC endpoint.

Figure 60
VPC Interface

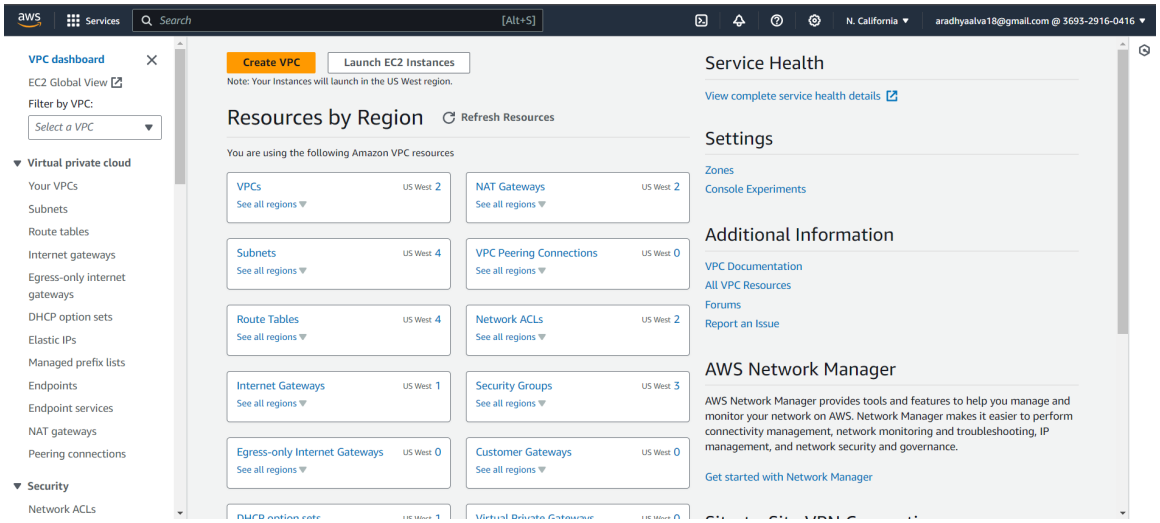


Figure 61
VPC List

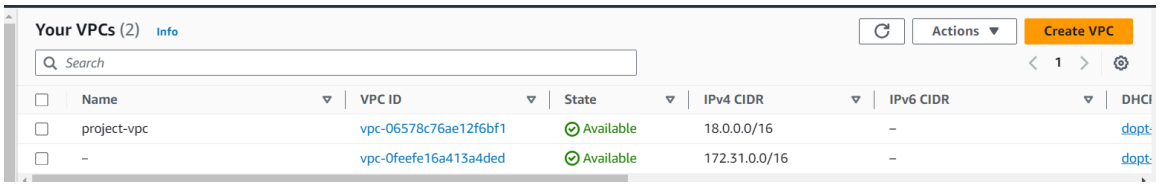


Figure 62
Endpoints Used

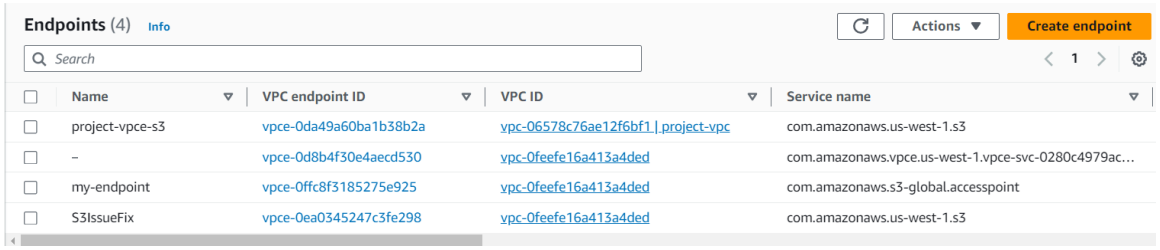


Figure 63
Subnets Used

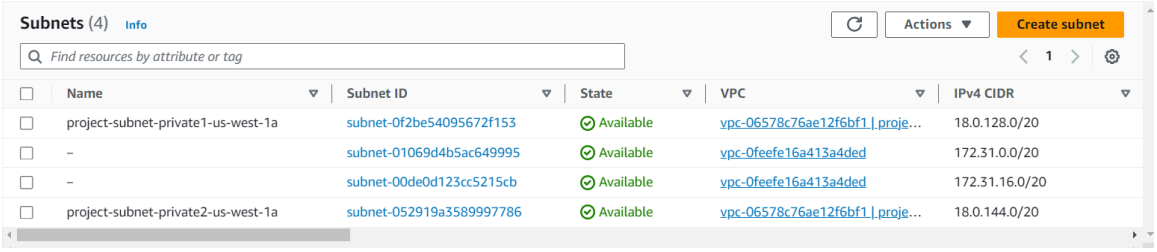
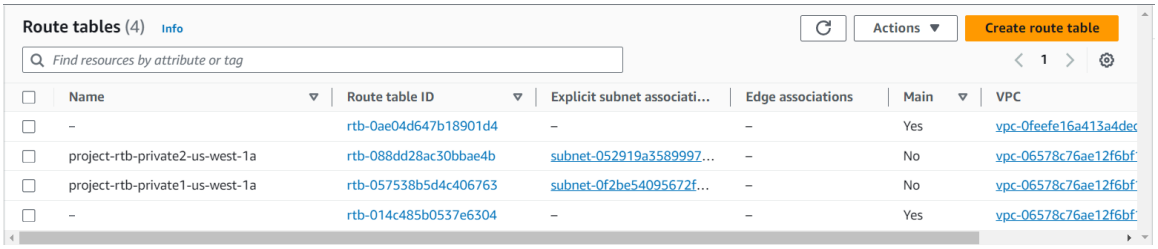


Figure 64
Route Tables Used



NAT Gateway Configuration.

Problem. If the Amazon Redshift cluster is situated in a private subnet and a NAT Gateway is being used for Amazon S3, the gateway might not be configured correctly. This potentially caused issues that have been taken care of for connecting Redshift to S3 through Amazon Glue.

Solution. Confirmed that the NAT gateway is in a public subnet with a route in the private subnet's route table pointing to the NAT gateway for internet access. It is an important step that ensures that the system is set up correctly for internet connectivity. Additionally, it is checked that the route table is associated with the private subnet where the Redshift cluster resides to ensure it has a route to the NAT gateway.

Figure 65
Gateways Used

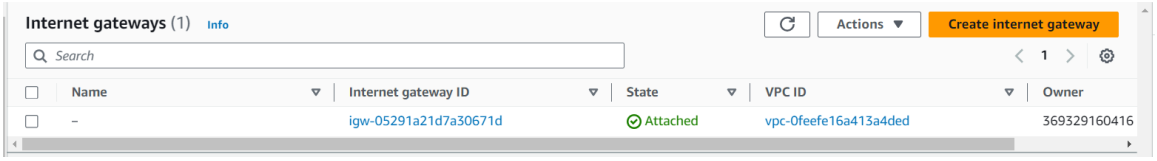
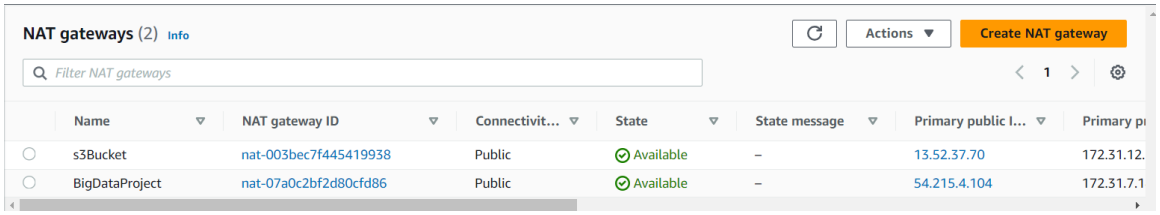


Figure 66

NAT Gateways Used



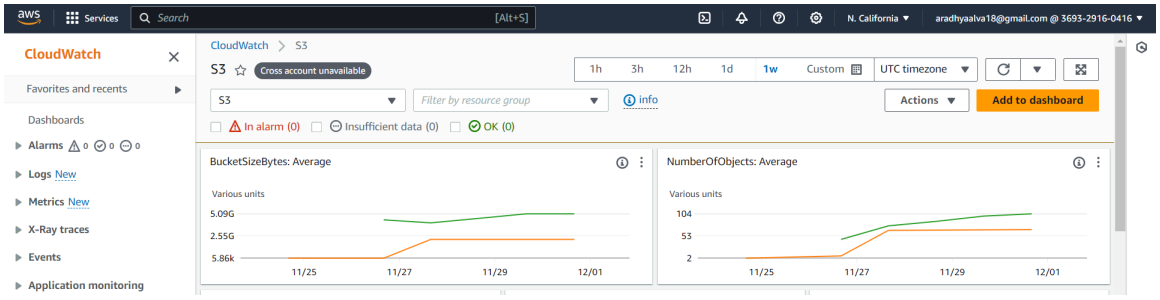
5.3 Used technologies and tools

The following is a list of AWS technologies and tools related to data storage and analysis that are incorporated into the project.

Amazon S3. Amazon S3 is a highly reliable, scalable, and secure object storage service that enables users to store and retrieve data at any time. It is commonly used as a data lake for storing both raw and processed data, as it offers a simple and cost-effective solution for data storage and access. Amazon S3 can store virtually any type of data, including images, videos, documents, and large datasets. It also provides features such as versioning, lifecycle policies, and cross-region replication to ensure data durability and availability. It is used as a staging layer in the project to store raw, cleansed, and transformed data. It is also used to store metadata and the results obtained from Amazon Athena.

Figure 67

S3 Bucket Performance



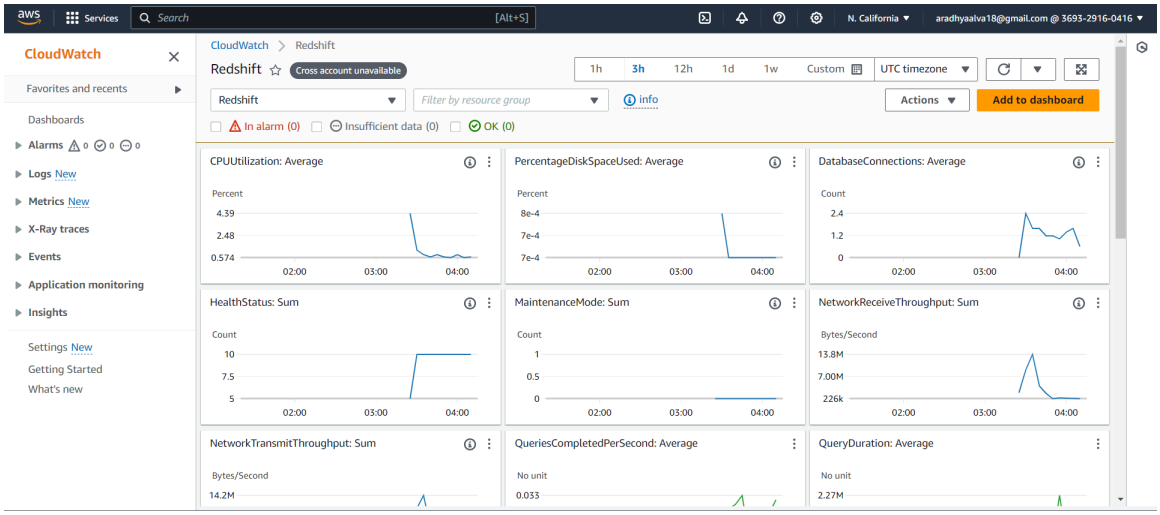
AWS Glue. AWS Glue is a fully managed extract, transform, and load (ETL) service that streamlines the process of preparing and loading data for analysis. It offers a range of features, including a crawler that automatically discovers data metadata, a development environment to create ETL jobs, and a runtime to execute those jobs. With

AWS Glue, users can easily create and maintain data transformation workflows, extract data from various sources, transform it into the desired format, and load it into target data stores for analysis. The service is designed to be scalable, reliable, and cost-effective, making it a popular choice for organizations of all sizes. This service is leveraged to cleanse, transform, and map the data into the Redshift database.

Amazon Redshift. Amazon Redshift is a fully managed data warehouse service that is optimized for high-performance analysis using SQL queries. It is designed to handle large datasets and complex queries, making it a popular choice for organizations that require fast and scalable data analytics. Amazon Redshift can be integrated with various data sources, including Amazon S3, Amazon DynamoDB, and other relational databases, allowing users to consolidate and analyze data from different sources in a single location. The service offers a range of features such as columnar storage, compression, and automatic optimization, which help to improve query performance and reduce costs. Additionally, Amazon Redshift offers flexible pricing options, including on-demand and reserved pricing, making it a cost-effective solution for organizations of all sizes. It is being utilized as the data warehouse for the project and is being loaded as fact and dimensional data.

Figure 68

Redshift Performance



AWS Glue Crawler. AWS Glue Crawler is a component of AWS Glue, which is a managed ETL service that simplifies the process of preparing and loading data for analysis. The crawler connects to the user's source or target data store and proceeds

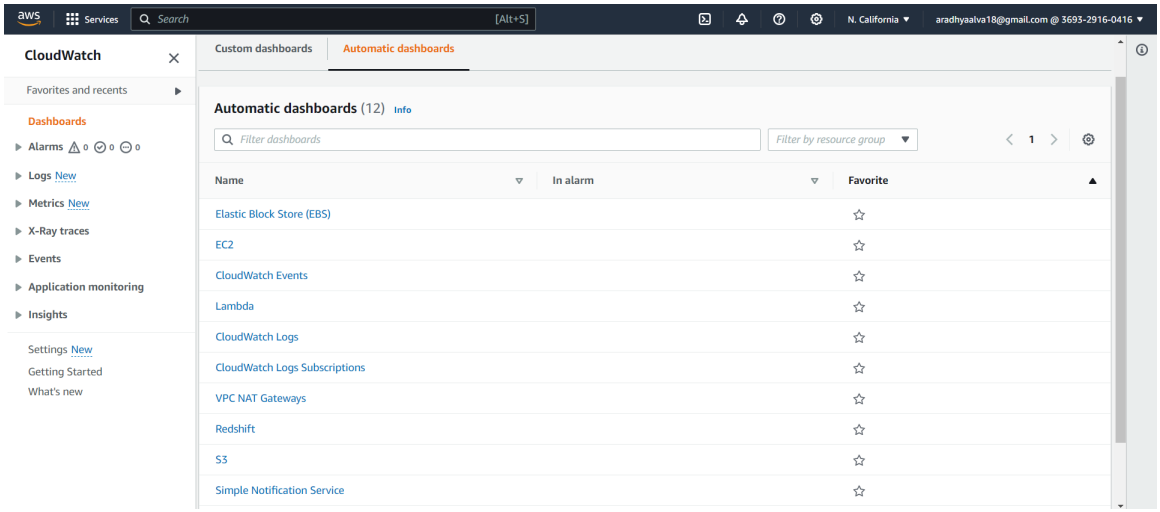
through a prioritized list of classifiers to determine the schema for the data. It then creates metadata tables in the AWS Glue Data Catalog, which can be used by other AWS services for data analysis. The AWS Glue Crawler is designed to automatically discover data metadata, including table and column names, data types, and other relevant information, and create schemas based on that metadata. This eliminates the need for manual schema creation and ensures that the data is correctly formatted for analysis. It is leveraged to get the schema of both the Redshift database and transformed data to store in the data catalog and then used to map the data efficiently.

Amazon Athena. Amazon Athena is an interactive query service that allows users to analyze data directly in Amazon S3 using SQL queries. It eliminates the need for complex ETL jobs, as users can query data directly from its source in S3. This makes it a popular choice for organizations that need to analyze large amounts of data quickly and without the need for complex data transformation processes. Amazon Athena is a serverless service, meaning that users do not need to manage any infrastructure or software. Queries are automatically distributed across a highly scalable and performant infrastructure, enabling users to get fast and accurate results. The service supports a wide range of data formats, including CSV, JSON, and Parquet, and can be easily integrated with other AWS services, such as Amazon QuickSight, to enable data visualization and reporting. It is used to test out the integrity of the data obtained.

Amazon QuickSight. Amazon QuickSight is a fully managed business intelligence service that simplifies the creation and publication of interactive dashboards that incorporate machine learning-powered insights. The service is designed to be easy to use, with a user-friendly interface that allows users to create custom dashboards and visualizations without the need for specialized skills. Amazon QuickSight can connect to various data sources, such as Amazon Redshift and Athena, to visualize and analyze data. The service offers a range of features, including automatic data refresh, drill-downs, and filters, that enable users to interactively explore and analyze their data. Additionally, Amazon QuickSight integrates with other AWS services, such as AWS Lambda and Amazon SageMaker, to enable advanced analytics and machine learning capabilities. The service is highly scalable and cost-effective, making it a popular choice for organizations of all sizes. It is utilized to generate insightful visualizations of the data obtained and integrate it into dashboards.

Figure 69

Cloudwatch Dashboard to Display Technologies Used



Chapter 6 System Testing and Experiment

6.1 Testing and experiment scope

The successful implementation of the project required a well-planned and executed testing process. Hence, the testing process is carefully designed by utilizing AWS Athena for querying the data. Since data integrity is of utmost importance for any analysis and output, the testing process played a critical role in ensuring the project's success.

To eliminate any potential issues that could affect the project's output, special emphasis is placed on ensuring that the data is consistent and of high quality. Although null values are handled, duplicate rows are deleted, and irrelevant data is removed in the initial data cleaning process, it is recognized that performing ad-hoc querying before mapping the data into the data warehouse is essential for in-depth analysis.

To accomplish this, the query editor in Athena is utilized to execute some ad-hoc queries and test the data's integrity. By performing ad-hoc querying, inconsistencies or poor data quality that might negatively affect the analysis and output of the project are identified and eliminated. This step allowed the team to resolve any issues with the data before moving forward with the next phase of the project.

Therefore, the testing process for the project is carefully planned and executed to ensure the data's consistency and high quality. By utilizing the query editor in Athena for ad-hoc querying, any potential issues with the data are identified and resolved before moving forward with the project's next phase.

6.2 Testing and experiment approaches

The figures presented below provide an overview of the data and the queries used to retrieve it. The data has been directly queried using the tables in the catalog database. Specifically, Query 1 provides information on ten different weather conditions and the corresponding number of crashes that occurred during each respective weather condition. Notably, the results are not displayed in chronological order or in order of the number of crashes. The purpose of this query is to test the integrity of the data and its behavior against ad-hoc queries.

In addition, 'Query 2' experiments with join statements to combine tables and generate output. The output displays the number of crashes as per the roadway surface condition, with the list limited to ten. The query editor settings have been configured to store the results of the query in the specified S3 bucket path. Once the query executes, the results are stored in the mentioned S3 bucket path to prevent loss of information and improve analysis.

This approach ensures the preservation of data integrity and minimizes the risk of data loss. By storing the results in the specified S3 bucket path, the data can be easily

retrieved and analyzed, facilitating more effective decision-making. Furthermore, the use of join statements in 'Query 2' enables the combination of tables to provide a richer and more comprehensive dataset for analysis.

As a result, the presented figures demonstrate the importance of reliable data retrieval and analysis to support effective decision-making.

Figure 70

Athena Query 1

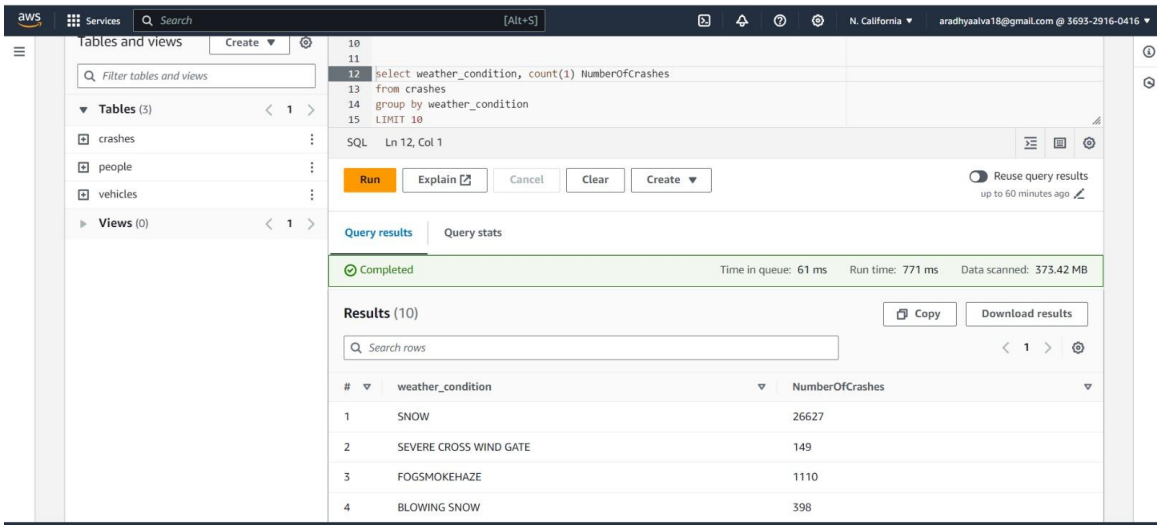


Figure 71

Athena Query 2

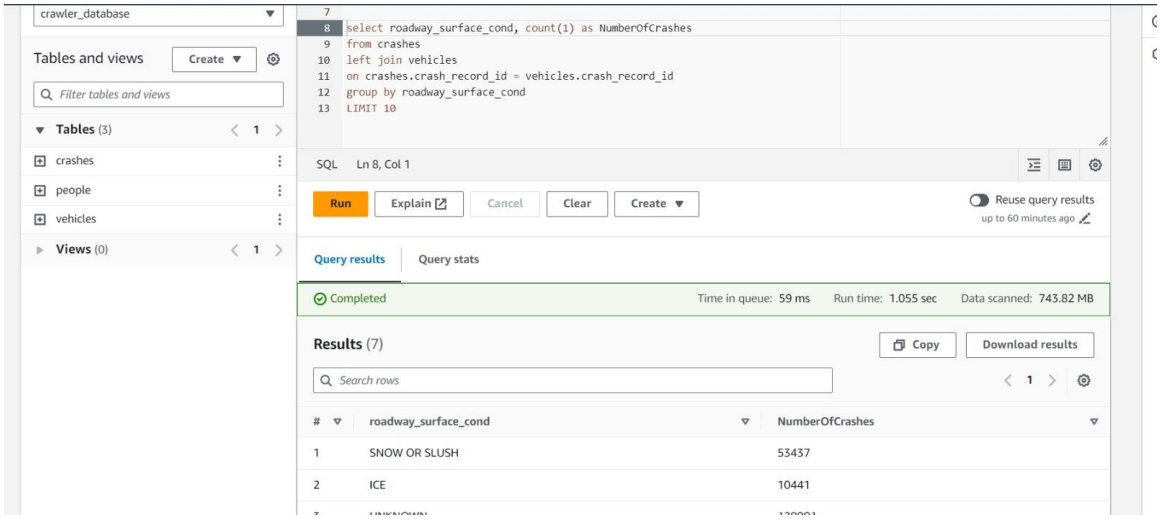


Figure 72

Athena Query Editor Settings

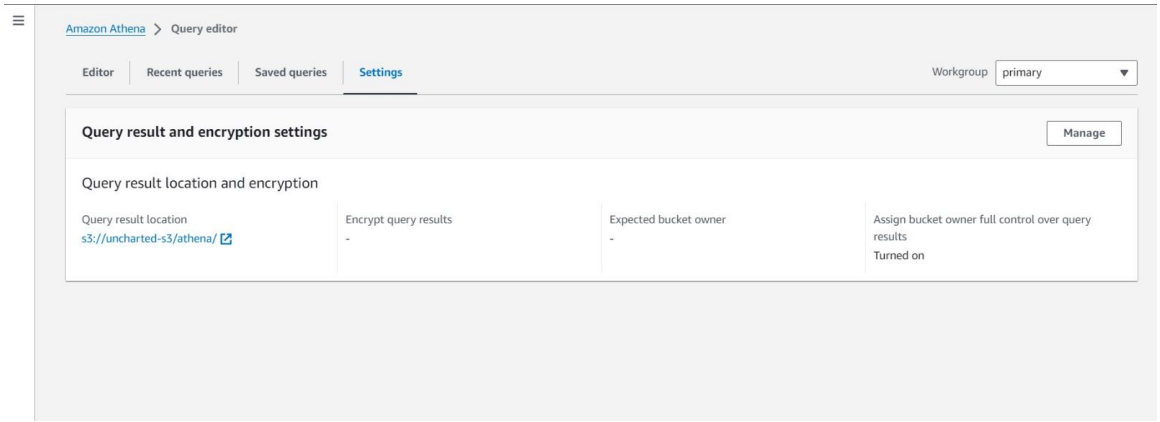
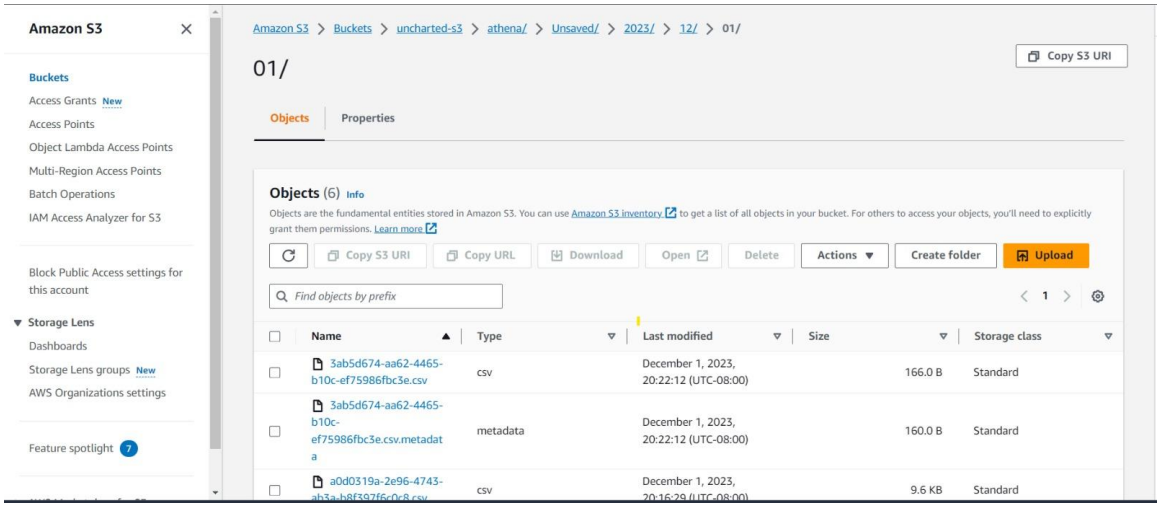


Figure 73

S3 Bucket Location to store Athena Results



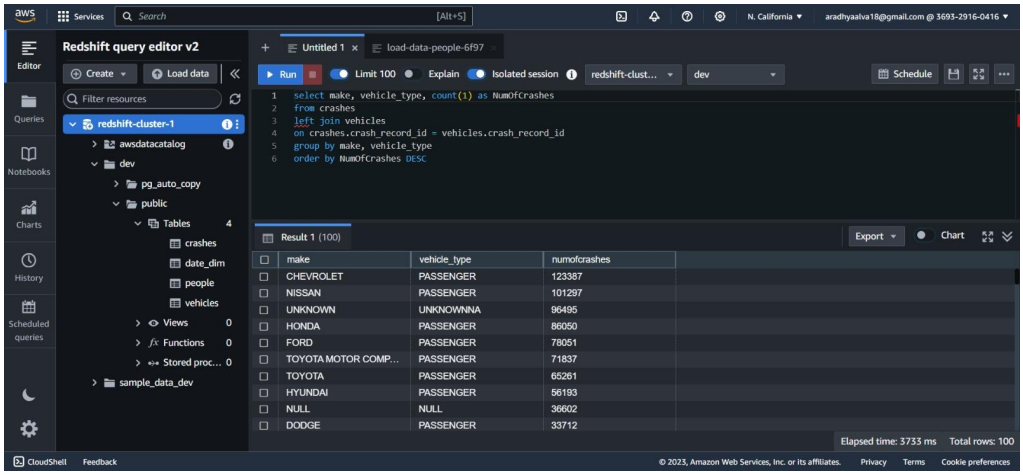
6.3 Testing report (or case study results)

Testing and Experiment

Once the data is loaded into the Redshift database tables, analysis and querying are performed on the data. ‘Query 1’ displays which vehicle companies have suffered the most number of crashes. The list displays the order in descending order which shows the insight that ‘Chevrolet’ vehicles have taken part in the most number of crashes, followed by ‘Nissan’. The insight is helpful to assess the reason and come up with necessary safety guidelines.

Figure 74

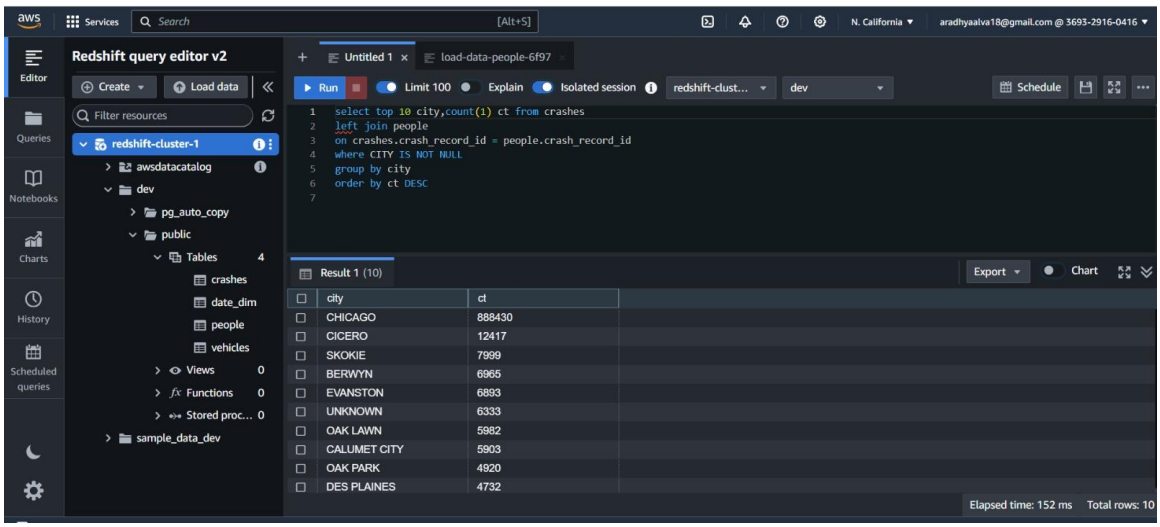
Redshift Query 1



The ‘Query 2’ displays the top 10 cities where the crashes have happened the most and the list is displayed in descending order. From the analysis, it is evident that Chicago suffers the most crashes at about ‘888430’ followed by Cicero at about ‘12417’. The analysis is important to assess which city has the most number of crashes to get in-depth into the matter and solve the root issues.

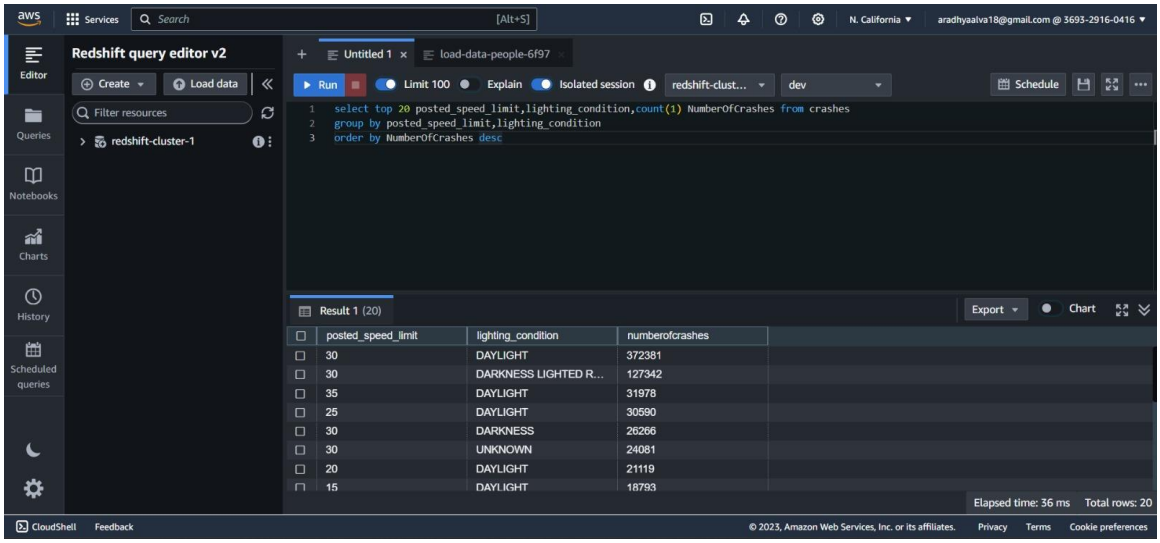
Figure 75

Redshift Query 2



‘Query 3’ displays under what speed limit and lighting conditions the most number of crashes occurred to analyze which set of combinations caused the most number of crashes. According to the query output speed limit of ‘30’ and the lighting condition of ‘Daylight’ have encountered the most number of crashes. It is necessary to get insight into the influence of lighting conditions and the speed limit for the crashes to occur.

Figure 76
Redshift Query 3



The loaded data in the Redshift database is utilized to create different visualizations in AWS Quicksight, that provide various insights into traffic and crash analysis. These visualizations included bar graphs, correlation plots, line graphs, box plots, density plots, and pie charts.

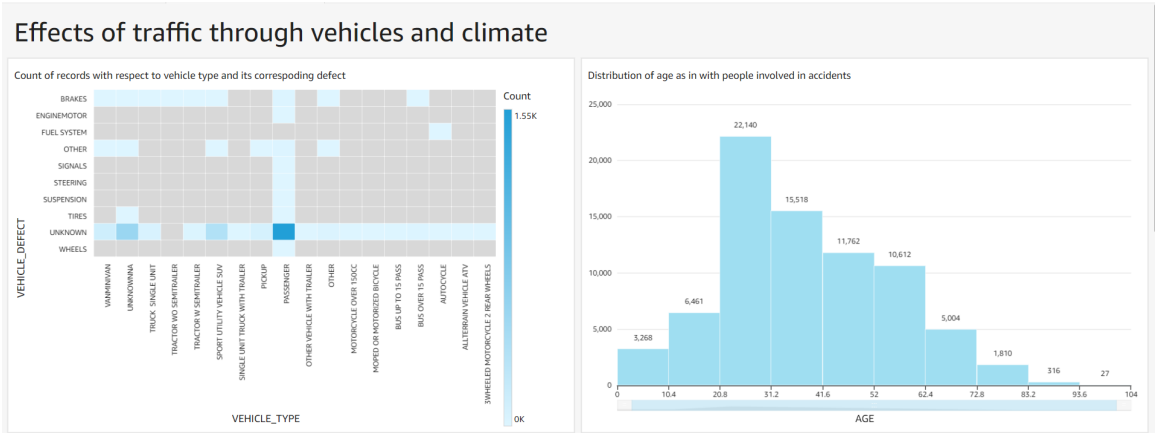
Quicksight Dashboards

Age-related Risk Insights. The visualizations below strongly emphasize the age group between 20.8 to 31.2 as being significantly prone to crashes. This crucial insight sheds light on age-related risk factors and serves as a foundation for targeted interventions and awareness campaigns aimed at enhancing the safety practices of individuals within this demographic.

Vehicle Defects and Crash Analysis. Also, the visualizations below delve into the defects of vehicles involved in crashes, offering a nuanced understanding of potential mechanical and technical issues contributing to accidents. This information becomes instrumental in devising strategies for preventive maintenance and regulatory measures to mitigate the impact of vehicle-related defects on road safety.

Figure 77

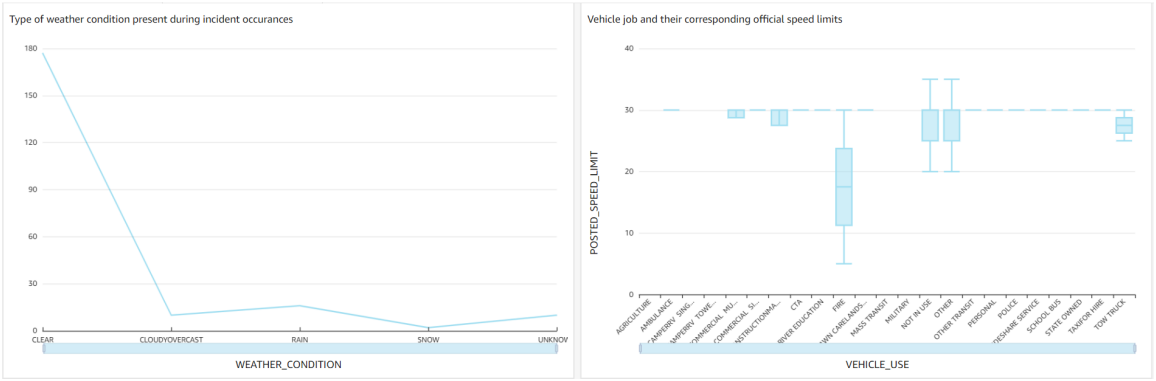
Vehicle Defects and Age-Related Crash Analysis for Comprehensive Traffic Safety Understanding



Weather Conditions and Infrastructure Planning. The analysis of weather conditions influencing crashes presents an opportunity for strategic infrastructure planning. Insights gained from the below visualization can inform the deployment of resources and technologies to mitigate the impact of adverse weather, enhancing overall road safety and minimizing weather-related incidents.

Figure 78

Leveraging Weather Insights for Strategic Infrastructure Planning

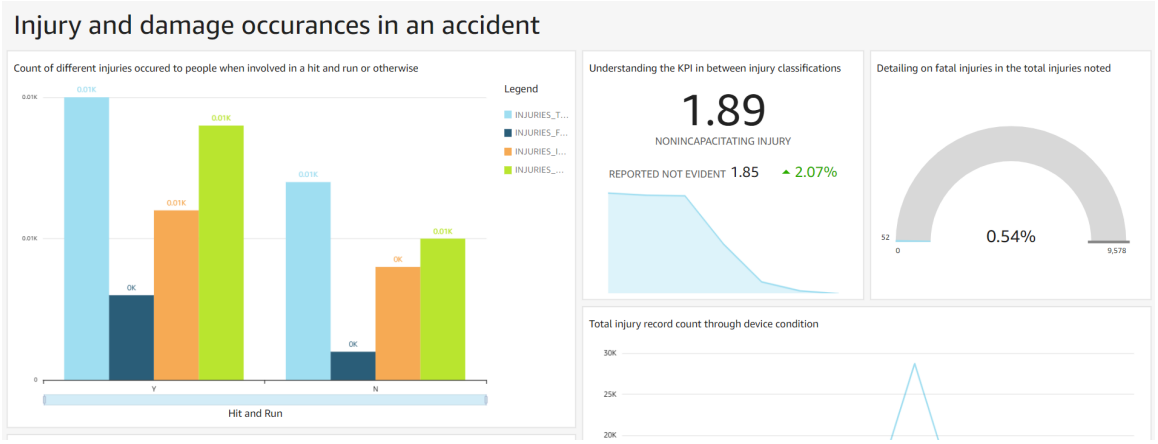


Injury and Damage Analysis for Holistic Safety. The injury and damage reports offer a holistic view of the consequences of crashes. This valuable information aids in developing comprehensive safety measures, including emergency response planning, healthcare resource allocation, and initiatives to reduce the severity of injuries. The correlation between injury types and associated damages provides a nuanced perspective for developing impactful safety interventions.

Overall, these insights below derived from diverse visualizations not only pinpoint areas of concern but also serve as a foundation for evidence-based strategies to enhance road safety. By understanding the multifaceted aspects contributing to crashes, stakeholders can implement targeted measures that address specific risk factors and ultimately contribute to a safer and more secure transportation environment.

Figure 79

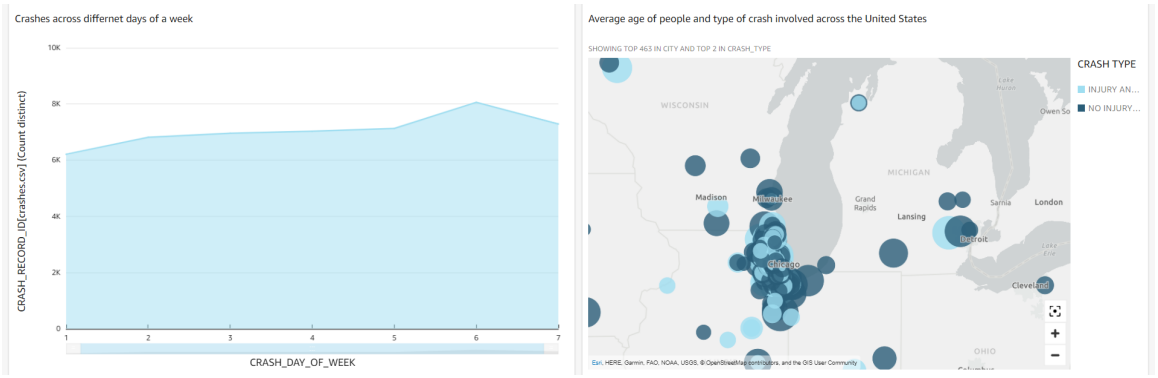
Exploring Relationships Between Injuries and Damages through Interactive Dashboard Visualization



Insights from Weather-Integrated Visualization. The visualization being showcased utilizes a map and intricately layers crash data with corresponding weather conditions. By effectively integrating crash and weather data, the visualization offers a spatial understanding of how variables like rain or snow contribute to increased crash risks. This insight is crucial for formulating strategies to enhance road safety, guiding infrastructure planning, and resource deployment to mitigate the impact of adverse weather conditions on accident occurrences. Thus, the visualization dashboard below delivers a holistic perspective on crashes throughout the United States, integrating temporal, regional, and urban-rural trends for a multifaceted analysis.

Figure 80

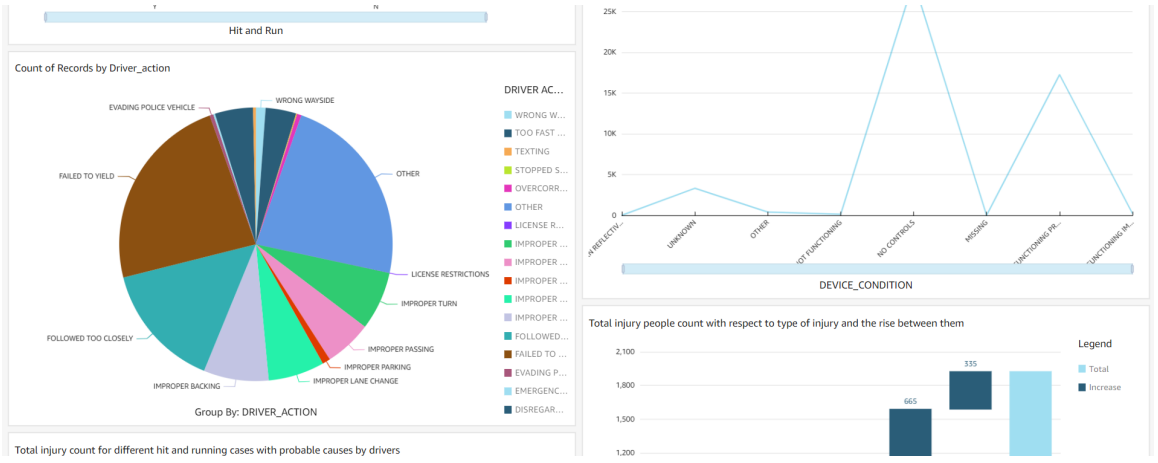
Mapping the Impact of Weather Conditions on Crashes for Enhanced Road Safety Understanding



Interactive Dashboard Dynamics. The dynamic and interconnected dashboard, composed of a donut chart, a line graph, and side-by-side bar graphs, provides a comprehensive tool for in-depth crash analysis. The donut chart below offers an immediate visual comparison of crash distribution across days of the week, while the line graph tracks temporal trends over months or years. The interactivity, facilitated by linked sliders or dropdowns, empowers users to apply specific filters, dynamically updating the entire dashboard. This coordination enables nuanced exploration of crash patterns, uncovering spatial and temporal insights across diverse categories on a unified platform.

Figure 81

Coordinated Multicategory Insights through Interactive Visualizations



In conclusion, the insightful visualizations derived from the comprehensive analysis of the loaded data in the Redshift database serve as valuable tools for addressing traffic challenges and improving road safety. By uncovering patterns related to age-specific crash frequencies, vehicle defects, weather influences, and temporal and spatial trends, these visualizations offer actionable insights for informed decision-making. Leveraging these findings, strategies can be devised to implement targeted interventions, educational campaigns, and infrastructure improvements, ultimately contributing to a reduction in the number of crashes, injuries, and fatalities on the roads. The integration of data-driven approaches, facilitated by the robust analytical capabilities of Redshift, positions this project as a valuable resource for urban planning, traffic management, and public safety initiatives.

Chapter 7 Conclusion and Future Work

7.1 Project summary

Project Overview

The four-month project, spanning from August 2023 to December 2023, is dedicated to analyzing Chicago road traffic crash data using advanced big data technologies. The project aims to extract meaningful insights to enhance road safety and address congestion issues in the city. Leveraging a vast dataset spanning over five years, sourced from the Chicago Police Department, AWS services such as S3, Glue, Redshift, and QuickSight are being used for efficient data storage, transformation, and visualization.

Throughout the project, rigorous data quality checks are being implemented to ensure the completeness and consistency of the analyzed information. The outcome of the project is reflected in the development of an interactive dashboard using AWS QuickSight. This dashboard serves as a powerful tool for various city agencies and community groups engaged in transportation planning. The insights derived from the dashboard analysis empower stakeholders to make informed decisions to enhance road safety and optimize traffic flow within the city. The project's ultimate contribution lies in providing actionable intelligence for policymakers, planners, and community leaders to create a safer and more efficient transportation infrastructure for the residents of Chicago. The source code is checked in Github, and the link to that source code can be found here: https://github.com/shashikumar1998/ChicagoVehicleCrashBigDataAnalaysis.

Status and Results

The project has completed the data cleaning, ingestion, and quality checks. The development of the interactive dashboard aims to offer visualizations, reports, and maps to stakeholders and users. The analysis has already yielded valuable insights into high-risk locations, common accident factors, and patterns. Recommendations based on these findings are presented in front of the entire class, as well as the entire project is documented focusing on data collection, implementation, infrastructure improvements, signal timing adjustments, and enforcement priorities.

Experience and Lessons

The critical importance of thorough data cleaning and documentation is increasingly being recognized and implemented. Stakeholder engagement remains a crucial aspect, ensuring that the dashboard aligns with real needs and that findings can effectively inform decision-making. The team's professional growth is being highly valued as they deploy AWS, perform ETL processes, and develop analytical applications.

Looking ahead, the project's outcomes are expected to contribute significantly to making Chicago's roads safer through informed, data-driven strategies.

7.2 Future work

The traffic management solution is currently being developed to incorporate additional real-time data sources such as traffic cameras and speed sensors. By leveraging these sources, the system can move beyond its current capabilities and delve into predictive analytics and forecasting for congestion and crashes. This expansion enables traffic authorities to proactively address potential issues, optimizing traffic flow and enhancing overall safety. Moreover, machine learning models play a crucial role in identifying crash hotspots and automatically classifying crash types and severity. This application of advanced analytics not only streamlines the response to incidents but also contributes to a deeper understanding of traffic patterns, allowing for more targeted safety measures.

Expanding the geographical scope to include road networks in surrounding suburbs is another crucial avenue for current and future work. This extension provides a more comprehensive view of regional traffic patterns, facilitating a holistic approach to traffic management. Additionally, the incorporation of augmented/virtual reality visualizations into the dashboard offers a more immersive exploration of crash locations and contributing factors. This not only enhances the user experience but also aids in the effective communication of insights to decision-makers. Integrating GPS and location data from ride-sharing/taxi services further broadens the analysis, allowing for a comprehensive assessment of the impact of traffic on commercial operations. Open-sourcing parts of the system or dashboard is a forward-looking step, encouraging collaboration and contributions from the research community, fostering innovation and continual improvement. Lastly, conducting before-and-after studies to quantitatively measure the impact of safety improvements recommended by the project analysis is crucial for refining and optimizing the solution over time, ensuring its ongoing effectiveness in improving traffic management decisions.

References

[1] W. Alajali, W. Zhou and S. Wen, "Traffic Flow Prediction for Road Intersection Safety," 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 2018, pp. 812-820, doi: 10.1109/SmartWorld.2018.00151

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019. doi:10.1109/tits.2018.2815678

[3] H. Al Najada and I. Mahgoub, "Big vehicular traffic Data mining: Towards accident and congestion prevention," 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 2016, pp. 256-261, doi: 10.1109/IWCMC.2016.7577067

[4] H. A. Najada and I. Mahgoub, "Autonomous vehicles safe-optimal trajectory selection based on big data analysis and predefined user preferences," 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2016, pp. 1-6, doi: 10.1109/UEMCON.2016.7777922

[5] A. Saxena and S. A. Robila, "Analysis of the New York City's Vehicle Crash Open Data," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 6017-6019, doi: 10.1109/BigData52589.2021.9672012

[6] H. A. Najada and I. Mahgoub, "Anticipation and alert system of congestion and accidents in VANET using Big Data analysis for Intelligent Transportation Systems," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 2016, pp. 1-8, doi: 10.1109/SSCI.2016.7850097