

**VIETNAM NATIONAL UNIVERSITY  
UNIVERSITY OF ECONOMICS AND LAW**



**Business Intelligence & Decision Support System**

**Project: CLOUD-ENABLED DATA WAREHOUSING  
AND VISUALIZATION FOR UK TRAFFIC  
ACCIDENT ANALYSIS: A MICROSOFT AZURE-  
BASED APPROACH**

**Class:** 242MI3303

**Lecturer:** MSc. Le Ba Thien

**Student:** **Group 02**

Lê Phạm Quỳnh Anh – K224161803

Nguyễn Quỳnh Anh – K224161805

Vũ Quốc Khách – K224161821

Tiêu Đăng Vinh – K224161842

Nguyễn Ngọc Phương Vy – K224161844

*TP. HCM, March 20<sup>th</sup> 2025*

## **Acknowledgement**

We wish to convey our heartfelt appreciation to everyone who provided support and guidance throughout the project titled "Cloud-Enabled Data Warehousing and Visualization for UK Traffic Accident Analysis: A Microsoft Azure-Based Approach."

First and foremost, we extend our deepest gratitude to our lecturer, Le Ba Thien, for his unwavering guidance, valuable insights, and enthusiastic encouragement. The cloud-based BI solutions shared through his lectures have deepened our understanding and equipped us with essential, enriching knowledge for our future academic journey. We are profoundly thankful for the invaluable lessons imparted throughout the course and for his support in enabling us to successfully complete our final project to the best of our abilities.

We also express our gratitude to those who accompanied us, offered assistance, shared their expertise, and provided high-quality feedback that helped refine our project. Additionally, we are thankful to the UK Department for Transport for supplying the dataset, which served as a vital resource, allowing us to translate theoretical knowledge into practical application.

Lastly, we appreciate our family and friends for their constant encouragement and understanding during this endeavor. Their support has been a significant driving force behind our completion of this project.

This project's success owes much to the collective efforts and contributions of everyone involved, and we are truly grateful for the chance to apply our skills and knowledge to such a impactful and rewarding undertaking.

Group 02

Ho Chi Minh City, March 20, 2025

## **Commitment**

We hereby declare that the project "Cloud-Enabled Data Warehousing and Visualization for UK Traffic Accident Analysis: A Microsoft Azure-Based Approach " is the result of the group's independent research and work, without copying or using any unauthorized documents. All references and data sources are fully and validly cited.

We guarantee that all information, analysis results, and conclusions in the project are performed honestly and accurately. All stages from data collection, data warehouse design, ETL implementation, to multidimensional analysis are performed according to the methods and techniques learned in the Data Warehouse and Integration subject.

We also pledge to be responsible for the quality and accuracy of this work and are ready to explain any issues related to the project content if requested.

Group 02

Ho Chi Minh City, March 20, 2025

## **Tables of Content**

<b>Tables of Content.....</b>	<b>4</b>
<b>List of Figures.....</b>	<b>7</b>
<b>List of Tables.....</b>	<b>10</b>
<b>CHAPTER 1. INTRODUCTION.....</b>	<b>11</b>
1.1. Overview and Challenges .....	11
1.2. Objectives.....	12
1.3. Object and scope.....	12
1.3.1. Objects .....	12
1.3.2. Scope.....	13
1.4. Business context & Requirement.....	13
1.5. KPIs .....	16
1.5.1. Total Accidents .....	16
1.5.2. Total Fatalities.....	16
1.5.3. Serious Accident Rate (SAR).....	17
1.5.4. Average Speed Limit (AVGSpeed).....	17
1.6. Technologies used.....	17
1.6.1. Data Ingestion and Storage .....	18
1.6.2. Data Processing and Transformation.....	19
1.6.3. Data Storage and Management.....	20
1.6.4. Monitoring and Notification .....	20
1.6.5. Data Visualization and Analysis.....	21
1.7. Project structure.....	22
<b>CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND</b>	<b>24</b>

2.1. Literature review.....	24
2.2. Theoretical background.....	26
2.2.1. Data warehouse .....	26
2.2.2. Cloud-Based Data Warehousing.....	27
2.2.3. Business Intelligence.....	29
2.2.4. Microsoft Azure .....	30
<b>CHAPTER 3. DATA PREPARATION AND DATA MODELING.....</b>	<b>32</b>
3.1. Data source .....	32
3.1.1. Data collection .....	32
3.1.2. Data description.....	32
3.2. Data Transformation .....	41
3.3. Data Modeling.....	52
3.3.1. Relationship.....	52
3.3.2. Dimension tables and Fact tables.....	54
<b>CHAPTER 4. EXPERIMENTING WITH THE ETL PROCESS ON AZURE FUNCTIONS .....</b>	<b>61</b>
4.1. ETL process on Azure Functions.....	61
4.1.1. Az Copy to blob storage.....	63
4.1.2. Azure Functions Deployment on Azure Portal.....	65
4.2. Bronze layer data ingestion.....	67
4.2.1. Raw-to-bronze data pipeline design.....	67
4.2.2. Schedule data updates .....	70
4.2.3. Send an error report email.....	72
4.3. Silver layer data ingestion.....	73

4.4. Gold layer data ingestion .....	76
4.5. Data Governance .....	81
4.6. Load data using Power Automate .....	83
<b>CHAPTER 5. DATA VISUALIZATION .....</b>	<b>85</b>
5.1. Data Modeling.....	85
5.2. Dashboard Strategy .....	86
5.3. Dashboard Visualization.....	88
5.3.1. KPIs .....	88
5.3.2. Page 1 - Accident Overview .....	90
5.3.3. Page 2 - Uncontrollable Causes.....	94
5.3.4. Page 3 - Controllable Causes .....	98
5.4. Recommendation.....	102
5.4.1. Uncontrollable Causes.....	103
5.4.2. Controllable Causes .....	103
<b>CHAPTER 6. CONCLUSION.....</b>	<b>105</b>
6.1. Conclusion.....	105
6.2. Limitation .....	106
6.3. Future works .....	106
<b>REFERENCES .....</b>	<b>108</b>

## List of Figures

Figure 2.1. Data Warehouse Process Overview (Source: Corporate Finance Institute) ...	26
Figure 2.2. Microsoft Azure data warehousing architecture (Source: Microsoft learn) ...	31
Figure 3.1. Accident Severity Distribution.....	44
Figure 3.2. Accidents Distribution by Day of the Week .....	44
Figure 3.3. Accidents Distribution by Road Type.....	45
Figure 3.4. Accidents Distribution by Light Conditions .....	45
Figure 3.5. Accidents Distribution by Weather Conditions.....	46
Figure 3.6. Accidents by Month .....	46
Figure 3.7. Accidents by Hour.....	47
Figure 3.8. Accident Severity & Speed Limit.....	47
Figure 3.9. Relationship Between Number of Vehicles and Number of Casualties .....	48
Figure 3.10. Accidents by Weather Conditions and Severity Level.....	48
Figure 3.11. Accidents by Road Surface Conditions and Speed Limit .....	49
Figure 3.12. Accidents by Junction Type and Urban/ Rural Area.....	49
Figure 3.13. Combination of Road Type, Severity, and Number of Casualties .....	50
Figure 3.14. Accidents by Weather Conditions, Light Conditions, and Severity .....	50
Figure 3.15. Speed Limit and Urban/ Rural Area.....	51
Figure 3.16. Accidents by Geographic Clusters.....	51
Figure 3.17. Police Attendance at the Scene and Accident Severity .....	52
Figure 4.1. Business Intelligence Solutions on the Microsoft Azure Platform .....	62
Figure 4.2. Azcopy tool settings screen.....	63
Figure 4.3. Task Scheduler tool settings screen .....	64
Figure 4.4. Project structure in the VS Code environment.....	65
Figure 4.5. Result of deploying functions to Azure Portal .....	66
Figure 4.6. List of functions in VS Code deployed via the Function App RawBronzeSilverGoldlayer .....	67
Figure 4.7. Raw to Bronze Pipeline Design .....	67
Figure 4.8. Metadata “Processed” .....	69

Figure 4.9. Timer Trigger setting.....	71
Figure 4.10. Sendgrid API key .....	72
Figure 4.11. Sendgrid error .....	73
Figure 4.12. Sendgrid loading successfully.....	73
Figure 4.13. Design a pipeline to ingest data from the storage account into the silver layer. .....	74
Figure 4.14. The data in the silver layer within the SQL database .....	75
Figure 4.15. Design a pipeline to ingest data from the silver layer into gold layer.....	76
Figure 4.16. Dim_AccidentSeverity.....	79
Figure 4.17. Dim_LightConditions .....	79
Figure 4.18. Dim_Police.....	79
Figure 4.19. Dim_RoadSurfaceConditions .....	80
Figure 4.20. Dim_RoadType.....	80
Figure 4.21. Dim_UrbanorRuralArea.....	80
Figure 4.22. Dim_WeatherConditions.....	80
Figure 4.23. Fact_Accidents.....	81
Figure 4.24. Table function_logs recording logs in SQL database.....	82
Figure 4.25. Add rows to a dataset in PowerBI when an item is created in SQL Server..	83
Figure 4.26. Upload Dataset Successfully in Power Automate platform .....	84
Figure 5.1. Data model .....	85
Figure 5.2. Accident Overview Dashboard .....	90
Figure 5.3. Total Accident Trend (2010 - 2015).....	91
Figure 5.4. Detail total accident trend over years.....	91
Figure 5.5. Total Accidents by Severity Type .....	92
Figure 5.6. Number of Casualties by Severity Type .....	92
Figure 5.7. Speed Limit by Severity Type .....	93
Figure 5.8. Total accident by road type .....	94
Figure 5.9. Uncontrollable Causes .....	94



Figure 5.10. Map displaying the total number of accidents by location in the United Kingdom.....	95
Figure 5.11. Number of Casualties by Location .....	95
Figure 5.12. Top 5 Weather Conditions with the Most Accidents.....	96
Figure 5.13. Road Surface Condition Table .....	97
Figure 5.14. Top 3 Road Surface Conditions with the Most Accidents .....	97
Figure 5.15. Controllable Causes Dashboard .....	98
Figure 5.16. Total Accidents by Light Conditions .....	99
Figure 5.17. Number of Casualties by Light Conditions.....	99
Figure 5.18. Total Accidents by Road Type.....	100
Figure 5.19. Number of Casualties by Road Type .....	101
Figure 5.20. Top 10 Police Forces with the most accidents .....	102

## List of Tables

Table 1.1. Traffic Accident Analysis Framework.....	14
Table 3.1. Data summary .....	33
Table 3.2. Data dictionary.....	36
Table 3.3. Relationship between dimensional and fact tables.....	52
Table 3.4. Dim_Date.....	54
Table 3.5. Dim_Time .....	54
Table 3.6. Dim_LightConditions .....	54
Table 3.7. Dim_Police.....	55
Table 3.8. Dim_RoadType.....	55
Table 3.9. Dim_AccidentSeverity.....	56
Table 3.10. Dim_RoadSurfaceConditions.....	57
Table 3.11. Dim_WeatherConditions .....	58
Table 3.12. Dim_UrbanorRuralArea .....	59
Table 3.13. Fact_Accidents.....	59
Table 5.1. Dashboard Strategy.....	86
Table 5.2. KPIs trend through time.....	88

# CHAPTER 1. INTRODUCTION

*This chapter outlines the project's aim of utilizing Microsoft Azure for cloud-based data warehousing and visualization of UK traffic accident data. It addresses the limitations of traditional data management approaches and emphasizes the necessity for real-time, scalable solutions. The project incorporates Azure Synapse Analytics, Data Factory, and Power BI to streamline data ingestion, processing, and visualization. By consolidating accident data from 2010 to 2015, it empowers stakeholders to analyze trends, identify high-risk areas, and foster data-driven decision-making to enhance road safety efforts.*

## 1.1. Overview and Challenges

Cloud computing has transformed how organizations store, process, and analyze data. Business Intelligence (BI) solutions leverage these advancements to convert raw data into actionable insights. At the core of BI, data warehouses integrate structured and semi-structured data, enabling efficient storage, retrieval, and analysis. Microsoft Azure provides a robust ecosystem for cloud-based data warehousing, offering services like Azure Synapse Analytics, Azure Data Factory, and Power BI to support real-time data processing and visualization.

In the government sector, data-driven decision-making is crucial, particularly in traffic safety and accident analysis. In the UK, road accidents cause significant human and economic losses. Government agencies, urban planners, and law enforcement rely on accident data to implement safety measures and optimize infrastructure. However, traditional data management systems struggle with fragmented data, slow processing, and limited real-time insights, hindering timely interventions.

This project addresses these challenges by developing a cloud-enabled BI solution on Microsoft Azure. A centralized data warehouse will store accident data, ensuring scalability and structured analysis. Azure Functions will automate data ingestion, triggered by time-based or event-based processes, maintaining up-to-date reporting. Power BI dashboards will visualize accident patterns, severity trends, and high-risk locations, providing clear, actionable insights for stakeholders. This approach enhances data

accessibility, efficiency, and decision-making, supporting the UK government's efforts to improve road safety.

## **1.2. Objectives**

This project aims to build an end-to-end cloud-based BI system for UK traffic accident analysis using Microsoft Azure. The key objectives include:

- a. Developing a cloud-based data warehouse
  - Centralize accident data for structured analysis.
  - Implement a BI solution on the Azure Functions platform to perform ETL processes across three database layers following the Data Lakehouse architecture.
- b. Implementing event-driven data processing
  - Apply event-driven triggers and scheduled updates to maintain real-time information and automate the data ingestion process.
  - Leverage Azure Functions triggers to optimize the automated ETL workflow across all three layers.
  - Leverage Azure Functions for automated data updates.
  - Apply time-based and event-based triggers to maintain real-time insights.
- c. Building interactive dashboards
  - Design Power BI visualizations aligned with business needs.
  - Provide stakeholders with actionable insights on accident trends and high-risk locations.

By integrating cloud computing, data warehousing, and BI visualization, this project delivers a scalable, automated, and interactive analytics platform, enhancing traffic accident monitoring and supporting data-driven policy decisions in the UK.

## **1.3. Object and scope**

### **1.3.1. Objects**

The primary object of this study is the UK traffic accident data from 2010 to 2015. This dataset serves as the core subject of analysis, enabling the exploration of accident trends, contributing factors, and high-risk locations. By leveraging Microsoft Azure for cloud-

based data warehousing and visualization, this project aims to enhance data accessibility and support data-driven decision-making for traffic safety improvements.

### **1.3.2. Scope**

- Time Scope: The project focuses on UK traffic accident data recorded between 2010 and 2015. This timeframe provides sufficient historical data to analyze accident patterns and trends over multiple years.
- Space Scope: The study covers all road networks in the UK, including urban and rural areas, motorways, and local roads. This broad geographic coverage ensures a comprehensive understanding of accident distribution across different locations.

### **1.4. Business context & Requirement**

Traffic accidents pose a significant challenge to road safety in the UK, impacting both public health and transportation efficiency. To address this issue, the UK government requires a comprehensive understanding of accident patterns and contributing factors to develop effective policies and interventions. This dashboard is designed to provide key stakeholders with data-driven insights into the frequency, severity, and causes of road accidents across different locations, road conditions, and environmental factors.

The primary audience for this dashboard includes government authorities and policymakers, who can utilize the data to develop targeted road safety policies and allocate resources effectively. Additionally, law enforcement agencies, including the police and emergency services, can leverage accident trends to enhance traffic law enforcement and optimize response strategies. Finally, urban planners and infrastructure managers can analyze accident hotspots and road conditions to inform future infrastructure improvements and mitigate risks.

By consolidating accident data into an interactive and visual format, this dashboard aims to support data-driven decision-making, enabling stakeholders to implement proactive measures that enhance road safety and reduce casualties.

*Table 1.1. Traffic Accident Analysis Framework*

<b>Goal</b>	<b>Objective</b>	<b>Business questions</b>
<i>Understanding Traffic Accidents</i>	<ul style="list-style-type: none"> <li>● Identify the role of light conditions in accident numbers and casualties.</li> <li>● Evaluate the impact of road types on accident frequency and severity.</li> <li>● Determine which police forces handle the most accidents to optimize resource allocation. Summarize accident trends by time, location, and severity.</li> <li>● Provide key metrics to help stakeholders track accident patterns.</li> </ul>	<ol style="list-style-type: none"> <li>1. Is the total number of accidents increasing or decreasing over time?</li> <li>2. What is the distribution of accidents by severity (slight, serious, fatal)?</li> <li>3. How many casualties occur at each severity level?</li> <li>4. How does speed limit correlate with accident severity?</li> <li>5. Which road types have the highest number of accidents?</li> </ol>
<i>Uncontrollable Factors in Traffic Accidents</i>	<ul style="list-style-type: none"> <li>● Analyze the impact of location (urban/rural) on the number of accidents</li> </ul>	<ol style="list-style-type: none"> <li>1. Which areas have the highest accident rates?</li> <li>2. How do casualty</li> </ol>

	<p>and casualties.</p> <ul style="list-style-type: none"> <li>● Assess the influence of weather conditions and road surface conditions on accident frequency and severity.</li> </ul>	<p>numbers compare between urban and rural areas?</p> <ol style="list-style-type: none"> <li>3. What weather conditions contribute to the highest number of accidents?</li> <li>4. How do different road surface conditions impact accident frequency?</li> <li>5. Which road surface conditions are associated with the most severe accidents?</li> </ol>
<i>Controllable Causes</i>	<ul style="list-style-type: none"> <li>● Identify the role of light conditions in accident numbers and casualties.</li> <li>● Evaluate the impact of road types on accident frequency and severity.</li> <li>● Determine which police forces handle</li> </ul>	<ol style="list-style-type: none"> <li>1. How do light conditions impact the number of accidents?</li> <li>2. Which light conditions are associated with the highest number of casualties?</li> <li>3. Which road types</li> </ol>

	the most accidents to optimize resource allocation.	<p>have the highest number of accidents?</p> <p>4. Which road types are associated with the most casualties?</p> <p>5. Which police forces handle the most accidents?</p>
--	---	---

## 1.5. KPIs

### 1.5.1. Total Accidents

- Definition: Total Accidents (TA) refers to the overall number of accidents occurring within a specific time period. This includes all types of accidents, irrespective of severity.
- Purpose: This KPI is designed to monitor the frequency of accidents. By tracking the total number of accidents, organizations can identify patterns, pinpoint high-risk times or locations, and assess the need for enhanced safety measures.
- Measure Calculation in Power BI  
TotalAccidents = `COUNT(Fact_Accidents[Accident_Index])`

### 1.5.2. Total Fatalities

- Definition: Total Fatalities refers to the total number of deaths resulting from accidents over a specific period. This includes fatalities from various types of accidents.
- Purpose: This KPI focuses on the severity of accidents by tracking fatalities. Monitoring this metric is vital for evaluating the effectiveness of safety protocols and interventions designed to reduce fatal accidents.
- Measure Calculation in Power BI:  
TotalFatalities =



```

CALCULATE(
    SUM(Fact_Accidents[Number_of_Casualties]),
    Fact_Accidents[AccidentSeverityKey] = 1
)

```

### 1.5.3. Serious Accident Rate (SAR)

- Definition: The Serious Accident Rate (SAR) measures the occurrence of serious accidents within a given period. This includes both fatal and serious accidents, offering insight into the overall severity of accidents in an area.
- Purpose: SAR is a critical KPI for assessing how often serious accidents occur in relation to the total number of accidents. It helps evaluate the effectiveness of safety measures and can inform decisions regarding speed regulations or infrastructure improvements.

- Measure Calculation in Power BI:

SARRate=

$\text{Fact\_Accidents}[\text{TotalSevereAccidents}] / \text{Fact\_Accidents}[\text{TotalAccidents}]$

### 1.5.4. Average Speed Limit (AVGSpeed)

- Definition: The Average Speed Limit (AVGSpeed) represents the typical speed limit set across roads or zones within a defined area.
- Purpose: This KPI is used to evaluate the general traffic control and safety measures. It helps assess if speed limits are appropriately set in relation to road safety standards and regulations.

- Measure Calculation in Power BI:

$\text{AVGSpeed} = \text{AVERAGE}(\text{Fact\_Accidents}[\text{Speed\_Limit}])$

## 1.6. Technologies used

This research project employs a sophisticated array of technologies, predominantly within the Microsoft Azure ecosystem, to construct and operate a data engineering pipeline focused on analyzing road traffic accident data in the United Kingdom from 2010 to 2015.

The pipeline facilitates the extraction, transformation, and loading (ETL) of data, transforming raw datasets into structured, actionable insights through a scalable and automated workflow. The following sections provide a detailed overview of the technologies utilized, highlighting their roles and contributions to the project's objectives.

### **1.6.1. Data Ingestion and Storage**

The data processing journey commences with a systematically designed ingestion architecture centered around Azure Blob Storage. This cloud-based storage solution functions as the primary repository for raw CSV traffic accident records, establishing the foundation upon which subsequent processing operations are built. Azure Blob Storage offers exceptional scalability characteristics—capable of accommodating datasets ranging from megabytes to petabytes—alongside durability features that ensure data integrity through redundant storage mechanisms. These properties make it particularly well-suited for managing large-scale transportation datasets that may grow substantially over time.

The architecture employs AzCopy, Microsoft's specialized command-line utility, to facilitate the secure transfer of data from local environments to the cloud storage infrastructure. AzCopy implements sophisticated data transfer protocols that optimize performance through concurrent operations and resumable transfers, ensuring efficient migration of potentially large accident record datasets. This utility's integration with Azure Active Directory authentication mechanisms further enhances security during data transmission processes.

PowerShell scripting technology serves as an automation layer by encapsulating AzCopy commands within executable scripts. This implementation reduces manual intervention requirements and minimizes potential human errors during data migration procedures. The scripts incorporate error handling mechanisms and logging capabilities, creating a robust framework for reliable data transfers. The Windows Task Scheduler technology further enhances this automation by orchestrating the execution of these PowerShell scripts according to predetermined temporal patterns—typically configured for daily execution at

midnight—thereby establishing consistent and reliable data acquisition rhythms that maintain pipeline operational continuity.

### **1.6.2. Data Processing and Transformation**

At the computational heart of the processing infrastructure lies Azure Functions, implementing a serverless architecture that dynamically responds to system events. This technology eliminates the need for manual server management while providing automatic scaling capabilities that adjust computational resources according to processing demands. Azure Functions, developed using the C# programming language, employ multiple trigger mechanisms that initiate processing operations based on specific conditions. Timer Triggers enable scheduled operations according to predefined temporal patterns, while Blob Triggers detect and automatically process newly uploaded datasets. This event-driven approach enables seamless movement of data through the bronze (raw data), silver (cleaned and transformed data), and gold (analytically optimized data) processing layers.

Within this computational framework, the CsvHelper library provides sophisticated parsing capabilities that transform raw CSV data into structured .NET objects representing accident records. This library implements advanced features including header mapping, type conversion, and error handling during the parsing process. The implementation addresses potential data quality issues such as missing values, inconsistent formatting, and special character handling, ensuring robust transformation of raw data into structured formats suitable for subsequent processing stages.

The SqlBulkCopy class enhances the efficiency of database operations by facilitating optimized batch insertions of transformed data. This technology implements specialized data transfer mechanisms that minimize database transaction overhead and optimize memory utilization during large-scale data insertions. The implementation includes calibrated performance parameters configured to process 10,000 records per batch with a timeout threshold of 300 seconds, establishing a balance between processing efficiency and system resource utilization.

### **1.6.3. Data Storage and Management**

Azure SQL Database functions as the central data repository, implementing a cloud-based relational database system that combines traditional SQL Server capabilities with cloud-specific features including automatic scaling, high availability, and integrated security mechanisms. The database architecture is structured according to Star Schema principles, which organize data into fact tables (containing quantitative measurements) and dimension tables (containing descriptive attributes) to optimize analytical queries through denormalization and simplified join operations. This database implementation separates data across the processing layers, creating a progressively refined data structure that culminates in the gold layer specifically optimized for complex analytical operations. The bronze layer preserves raw data in its original form, maintaining historical integrity while enabling data lineage tracking. The silver layer contains cleaned and transformed data with standardized formats, consistent naming conventions, and resolved anomalies. The gold layer presents analytically optimized data structures with pre-aggregated metrics, derived attributes, and optimized query paths designed to support business intelligence operations.

### **1.6.4. Monitoring and Notification**

SendGrid represents a significant component of the operational monitoring infrastructure within this data engineering architecture. As a cloud-based electronic messaging service, SendGrid has been integrated through its corresponding .NET library to establish automated communication channels regarding ETL process statuses. The implementation facilitates bidirectional information flow between the system and operational stakeholders, providing critical visibility into pipeline execution states. When exceptional conditions arise during data processing—such as syntax inconsistencies during CSV parsing operations—SendGrid automatically generates and distributes error notifications to designated recipients. Conversely, upon successful completion of ETL processes, the system transmits confirmation messages, thereby establishing a comprehensive awareness of operational states. This notification mechanism substantially enhances operational

oversight by reducing detection latency for potential issues and enabling rapid remedial intervention when necessary.

The monitoring infrastructure incorporates ILogger as its fundamental event documentation mechanism. Implemented within the Azure Functions computational layer, ILogger establishes systematic recording capabilities for pipeline execution events across multiple severity classifications, including informational entries, warnings, and critical errors. This logging architecture operates in real-time, capturing execution details contemporaneously with processing operations. The resulting documentation follows a structured data format and undergoes persistent storage within the Azure SQL Database environment, specifically within designated repositories such as the `function_logs` table. This structured approach to event documentation serves multiple operational purposes, including establishing comprehensive audit trails for compliance verification, providing diagnostic information essential for technical troubleshooting procedures, and generating temporal data that supports performance analysis and optimization efforts. The implementation of ILogger thus represents a critical element in maintaining operational transparency and technical governance throughout the data engineering workflow.

#### **1.6.5. Data Visualization and Analysis**

Power BI serves as the analytical interface within this architecture, implementing a business intelligence platform that connects directly to the gold layer data repository. This technology provides advanced data visualization capabilities through an intuitive interface that enables stakeholders to explore complex datasets without requiring advanced technical skills. The implementation includes interactive dashboards and visualizations that present accident trends, geographical distributions, temporal patterns, and key performance indicators related to traffic safety. The analytical layer transforms complex data into comprehensible insights through visualization techniques including geospatial mapping of accident locations, temporal heat maps showing accident frequency patterns, and interactive filters that enable multidimensional analysis of contributing factors. These analytical capabilities support evidence-based decision-making processes for

transportation safety policy development, infrastructure planning, and resource allocation for accident prevention initiatives.

## **1.7. Project structure**

### **Chapter1: Introduction**

Chapter 1 provides an overview of the project, outlining its objectives, scope, and challenges while defining the business requirements and key performance indicators (KPIs). It also reviews the technologies used and relevant literature to establish a foundation for the project. The focus is on setting the context and expectations for the data processing pipeline.

### **Chapter2: Theoretical Background và Literature review**

This section covers foundational theories and prior research relevant to the project, including concepts like Lakehouse architecture, Data Warehouse, and Visualization techniques. It reviews existing studies to contextualize the project's approach to data management and analytics. The chapter builds a theoretical framework to support the technical implementation.

### **Chapter 3: Data Preparation and Data Modeling**

Chapter 3 focuses on exploring the dataset, applying data cleaning and transformation techniques to gain the most comprehensive overview of the data landscape. From there, a data model is designed in a star schema format to be implemented in the data warehouse at the gold layer.

### **Chapter 4: Experimenting with the ETL Process on Azure Functions**

In chapter 3, technical aspects are explored deeply through data handling, including data sources, collection methods, and descriptions, alongside the Exploratory Data Analysis (EDA) plan for transformation. It details the pipeline designs for ingesting data across raw-to-bronze, bronze-to-silver, and silver-to-gold layers, incorporating tools like Azure Blob Storage and triggers. The chapter emphasizes the technical groundwork for efficient data flow and error management.

### **Chapter 5: Data Visualization**

This chapter focuses on the practical execution of the project, covering data modeling techniques, dashboard development strategies, and visualization approaches using Power BI. It describes how transformed data from the gold layer is leveraged to create actionable insights. The goal is to bridge the theoretical pipeline with tangible outcomes for end-users.

## **Chapter 6: Conclusion**

Finally, chapter 5 summarizes the project's achievements, highlights its limitations, and proposes future enhancements. It reflects on the pipeline's effectiveness and dashboard utility while suggesting areas for improvement or expansion. The chapter wraps up the project with a forward-looking perspective.

## **CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND**

*This chapter delves into the theoretical framework and existing research pertaining to traffic accident analysis, cloud-based data warehousing, and business intelligence (BI) solutions. It reviews previous studies on accident trends, risk factors, and predictive analytics while also highlighting the limitations inherent in traditional data management practices. Moreover, it discusses essential concepts in cloud computing, data engineering, and visualization techniques, with a particular focus on Microsoft Azure's role in contemporary BI architectures. This chapter establishes a foundation for the project's methodology by aligning best practices in data processing, storage, and analytics with practical applications in traffic safety.*

### **2.1. Literature review**

The study of traffic accidents has undergone significant methodological transformations over the past several decades. Early research predominantly employed rudimentary statistical techniques to quantify accident rates and identify elementary risk factors. Initial analyses were largely descriptive, focusing on accident frequencies, injury severities, and the basic categorization of incidents. Over time, however, the discipline has transitioned toward more comprehensive and sophisticated methodologies. Researchers now integrate predictive analytics, spatial-temporal modeling, and machine learning techniques to understand the complex interplay between environmental, vehicular, and human factors that contribute to road incidents. Early research by Haddon (1972) established the foundational matrix for conceptualizing accidents as multi-factorial events occurring across temporal phases. This paradigm shift catalyzed subsequent investigations into the complex interplay between human factors, vehicular characteristics, and environmental conditions.

In the British context, seminal work by Quimby et al. (1999) examined the spatial distribution of accidents across the UK road network, identifying critical correlations



between infrastructure design and collision frequency. Their findings, substantiated by longitudinal studies conducted by the Transport Research Laboratory (TRL), revealed significant regional variations in accident patterns, necessitating localized intervention strategies. More recently, Stipdonk and Berends (2008) applied advanced statistical methods to disaggregate accident data by severity levels, demonstrating that fatal and non-fatal accidents often exhibit distinct causal mechanisms and should be analyzed separately. The integration of geographical information systems (GIS) into accident analysis, pioneered by Erdogan et al. (2008), has facilitated the identification of high-risk road segments through spatial clustering techniques. This methodological advancement has been particularly valuable for UK transportation authorities in prioritizing infrastructure improvements. Concurrent developments in machine learning applications, as documented by Alkheder et al. (2017), have enabled the development of predictive models that forecast accident probabilities with increasing accuracy, thereby informing proactive safety measures.

Within the domain of traffic accident analysis, Yannis et al. (2013) demonstrated the efficacy of centralized data repositories in harmonizing heterogeneous data sources, including police reports, hospital records, and insurance claims. Their research underscored the importance of data quality assurance protocols and standardized extraction-transformation-loading (ETL) processes in maintaining analytical integrity. Building upon these principles, Thakuriah et al. (2016) proposed architectural frameworks specifically designed for transportation data ecosystems, emphasizing the need for scalable storage solutions capable of accommodating ever-increasing data volumes and varieties. The emergence of cloud-based data warehousing represents a paradigmatic shift in how transportation data is managed and analyzed. Empirical studies by Abaker et al. (2016) have documented substantial improvements in computational efficiency and analytical flexibility when migrating from on-premises infrastructure to cloud platforms.

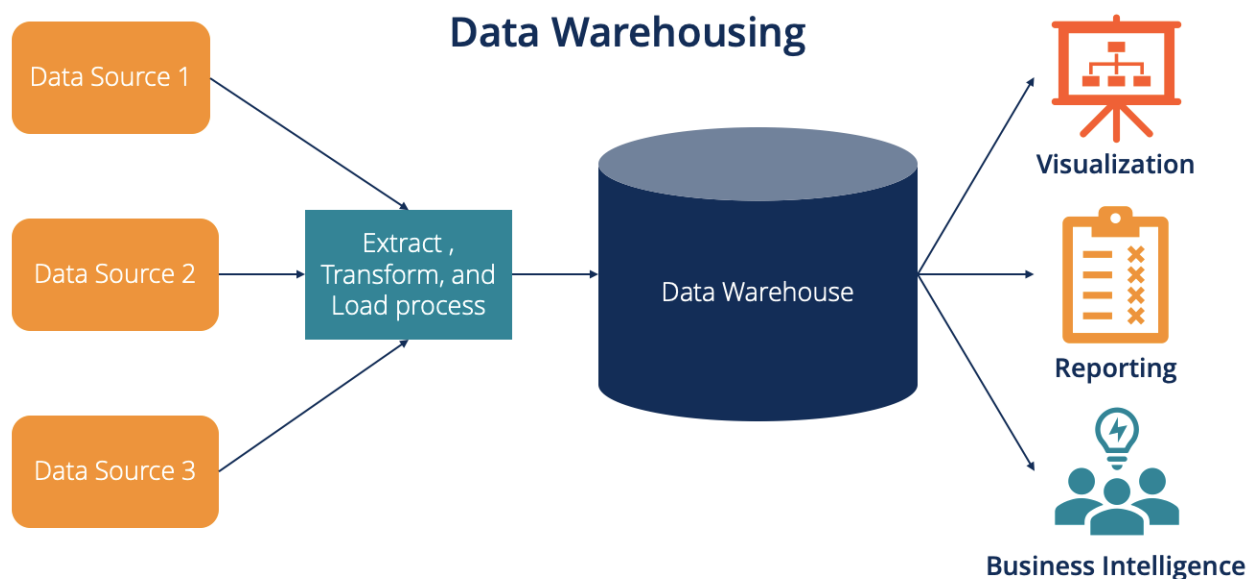
The adoption of cloud computing technologies within public sector organizations has been examined extensively in the literature. Theoretical frameworks proposed by Buyya et al. (2009) delineate the fundamental characteristics of cloud services, including on-demand

self-service, broad network access, resource pooling, rapid elasticity, and measured service. These principles have been progressively applied to government data management strategies, as documented by Janssen and Joha (2011) in their analysis of public sector cloud implementation challenges.

## 2.2. Theoretical background

### 2.2.1. Data warehouse

A data warehouse is a specialized system designed to facilitate decision-making processes by providing a unified, structured, and analyzable repository of data. According to W. H. Inmon, widely regarded as the "father of the data warehouse", the concept emphasizes four defining characteristics: subject-oriented, integrated, time-variant, and nonvolatile. Each attribute contributes to the warehouse's ability to support complex analytical queries and long-term strategic decision-making.



*Figure 2.1. Data Warehouse Process Overview (Source: Corporate Finance Institute)*

A subject-oriented data warehouse focuses on organizing data around specific business areas or subjects, such as customers, sales, or inventory, rather than adopting a transaction-centric approach like operational databases. This structure simplifies analysis by presenting information in a format that aligns with business needs. The integration of data within the warehouse ensures consistency by consolidating and reconciling diverse sources into a

unified schema. This process involves careful cleaning and transformation to eliminate inconsistencies, redundancies, and format disparities, ultimately creating a coherent dataset. Time variance is another crucial characteristic of a data warehouse, as it retains historical data to facilitate longitudinal analysis. Unlike operational systems that primarily store current data and often overwrite it, a data warehouse preserves time-stamped records to support trend analysis, forecasting, and the evaluation of historical performance. This temporal aspect is vital for identifying patterns and correlations over extended periods, which are essential for strategic planning. Nonvolatility in a data warehouse guarantees that once data is loaded, it remains stable and unaltered, except for controlled updates. This characteristic provides a consistent and reliable foundation for reporting and analysis, assuring users that the data reflects a specific point in time without being affected by frequent transactional changes. The process of constructing and managing a data warehouse, known as data warehousing, encompasses multiple phases. It begins with data extraction from various sources, followed by transformation into a uniform, analyzable format, and concludes with loading the processed data into the warehouse. This Extract, Transform, Load (ETL) process is foundational to data warehousing, ensuring that the data is both accurate and aligned with the warehouse's schema.

### **2.2.2. Cloud-Based Data Warehousing**

Data Warehouse as a Service (DWaaS) is a managed cloud service model that enables organizations to leverage the insights, data consistency, and various advantages of a data warehouse without the burden of building, maintaining, or managing its infrastructure. In this model, the cloud service provider handles the setup, configuration, management, and maintenance of the hardware and software resources necessary for the data warehouse.

As data continues to grow in diversity, volume, and velocity, the modernization of data warehouses has become a necessity. While on-premises data warehouses have long been foundational to enterprise business intelligence (BI), they come with significant hardware and software costs, and require ongoing maintenance.

Cloud-based data warehouse services facilitate the collection, storage, and processing of data, allowing organizations to effectively meet their cloud data management needs and gain easy access to critical information. Furthermore, these services offer the ability to instantly scale resources up or down in response to real-time demand, making them more cost-effective compared to traditional on-premises solutions. At the same time, DWaaS minimizes the upfront investment and administrative burden typically associated with conventional data warehousing. Customers are primarily responsible for providing the data and paying a fee for the managed service.

The conceptual foundation of cloud-based data warehousing represents an evolution of traditional data management paradigms. Kimball and Ross (2013) articulated the fundamental principles of dimensional modeling that continue to underpin contemporary warehouse designs, emphasizing the importance of fact and dimension tables organized to optimize analytical queries. The extension of these principles to cloud environments introduces additional considerations related to distributed storage, parallel processing, and dynamic resource allocation.

The theoretical framework proposed by Inmon (2005) regarding enterprise data warehousing provides valuable context for understanding cloud-based implementations. His concept of the "Corporate Information Factory" provides a robust architecture for integrating diverse data sources into a cohesive analytical environment. This model has been adapted and refined for cloud platforms by later researchers and practitioners.

Within Azure-specific architectures, the lambda and kappa patterns described by Marz and Warren (2015) have gained significant prominence. These theoretical models address the challenges of processing both batch and streaming data within a unified framework—a capability particularly relevant for traffic accident analysis, which involves both historical datasets and real-time information streams. The implementation of these patterns in Azure environments typically leverages services such as Data Factory for orchestration, Databricks for processing, and Synapse Analytics for storage and querying.

### **2.2.3. Business Intelligence**

Business intelligence (BI) encompasses a range of technological processes designed to collect, manage, and analyze organizational data in order to generate insights that inform business strategies and operations. Business intelligence analysts play a crucial role in transforming raw data into valuable insights that guide strategic decision-making within the organization. BI tools empower business users to access various types of data such as historical, current, third-party, and in-house data along with semi-structured and unstructured data like social media content. By analyzing this information, users can gain a clearer understanding of business performance and identify the best course of action moving forward.

The theoretical understanding of business intelligence capabilities has been significantly enhanced by the development of maturity models that describe the progressive evolution of analytical capabilities within organizations. The model proposed by Davenport and Harris (2007) defined a continuum from descriptive to prescriptive analytics, providing a framework for evaluating the sophistication of analytical approaches. In the context of traffic safety, this progression corresponds to the advancement from basic accident reporting to predictive risk assessment and automated intervention recommendation.

Building upon this foundation, the Business Analytics Capability Framework developed by Cosic et al. (2012) identified four key dimensions of analytical maturity: governance, culture, technology, and people. This holistic perspective emphasizes that effective implementation of advanced analytics depends not only on technological infrastructure but also on organizational factors that support data-driven decision-making.

The specific application of these concepts to public sector organizations has been explored by Klievink et al. (2017), who identified unique challenges and opportunities for government agencies adopting advanced analytics. Their research highlighted the importance of addressing institutional barriers, privacy concerns, and cross-agency collaboration all factors highly relevant to traffic accident analysis in governmental contexts.

#### **2.2.4. Microsoft Azure**

The theoretical principles underlying Microsoft Azure's data services reflect modern methodologies in distributed computing, data parallelism, and service-oriented architecture. The polyglot persistence model, as described by Sadalage and Fowler (2012), provides a valuable conceptual framework for understanding Azure's diverse storage solutions, including relational databases, NoSQL alternatives, and data lakes. This model acknowledges that varying data types and usage patterns often necessitate specialized storage technologies, a principle particularly relevant in the context of analyzing heterogeneous traffic accident data.

Azure Data Factory serves as a practical application of the extract, transform, load (ETL) paradigm within a cloud-native environment. According to Vassiliadis and Simitsis (2009), modern ETL processes must address both technical requirements for data integration and business needs for governance and auditability. Azure Data Factory expands upon this framework by integrating workflow orchestration capabilities, enabling the creation of complex data processing pipelines that span multiple services and environments. This service embodies the theoretical principles outlined in data flow architectures by Abadi et al. (2003), which highlight the significance of declarative specifications for data transformations and movements.

Azure Synapse Analytics exemplifies the theoretical concept of massively parallel processing (MPP) as articulated by DeWitt and Gray (1992). By distributing computational workloads across multiple nodes, it achieves significant performance enhancements for complex analytical queries. This architecture ensures efficient processing of large accident datasets while maintaining interactive response times for analytical users. The foundational theoretical model of columnar storage, formalized by Stonebraker et al. (2005), supports substantial compression ratios and improved query performance, particularly beneficial when working with multidimensional traffic accident data.

Azure Functions represents a practical interpretation of the serverless computing paradigm as described by Baldini et al. (2017). This approach emphasizes event-driven, stateless computation with automatic scaling capabilities. It aligns seamlessly with the theoretical

framework of microservices which advocates for breaking down complex processing tasks into distinct, independent units of functionality. In the realm of traffic accident analysis, Azure Functions allows for the implementation of specialized data processing routines that can be activated by specific events, such as the arrival of new data or scheduled analyses. Azure Data Lake Storage is built upon the theoretical principles of schema-on-read architectures. This approach diverges from traditional relational models by postponing schema enforcement until the time of query execution, enabling more flexible ingestion of diverse data sources. In the context of traffic accident analysis, this paradigm supports the integration of structured accident records with semi-structured sensor data and unstructured contextual information, thereby creating a robust analytical foundation.

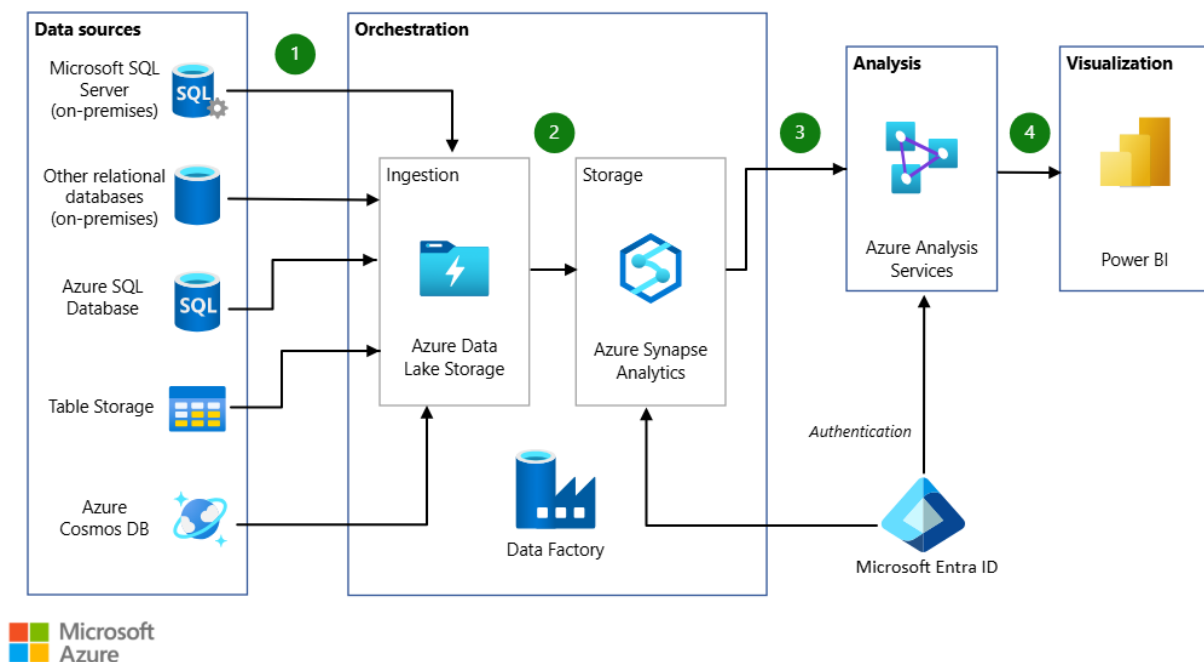


Figure 2.2. Microsoft Azure data warehousing architecture (Source: Microsoft learn)

## CHAPTER 3. DATA PREPARATION AND DATA MODELING

*Chapter 3 focuses on preparing and modeling road traffic accident data in the United Kingdom from 2010 to 2015, sourced from the UK Department for Transport with over 876,497 records and 32 attributes. The data transformation process includes creating binary flags for missing data, standardizing variables, temporal aggregation, spatial clustering, and deriving new variables, enabling the identification of trends such as peak accident hours and geographic hotspots. The modeling section establishes dimension tables (e.g., Dim\_Date, Dim\_RoadType) and a fact table (Fact\_Accidents), facilitating multidimensional analysis to support evidence-based road safety strategies.*

### **3.1. Data source**

#### **3.1.1. Data collection**

The dataset, obtained from the UK Department for Transport and available on platforms like Kaggle and data.gov.uk, constitutes a comprehensive collection of road traffic accident data gathered over an 11-year period from 2005 to 2015. Each year's data is represented in a separate CSV file. Comprising over 876,497 records, this dataset offers a solid foundation for analyzing patterns, trends and contributing factors related to road accidents across the United Kingdom. The extensive time frame allows for longitudinal studies, enabling researchers to evaluate changes in accident frequency, severity and environmental conditions over time, potentially in response to policy interventions, infrastructure improvements, or shifts in societal behavior. For this project, we focus exclusively on the data from 2010 to 2015.

#### **3.1.2. Data description**

The dataset comprises 32 attributes that encompass a diverse array of variables. By utilizing these attributes, researchers can explore vital questions, such as the relationship between road conditions and the severity of accidents, the effects of speed limits on casualty rates and the distribution of accidents in urban versus rural settings. Within the broader academic landscape, this dataset enhances the evolving field of big data analytics



in transportation studies. Its open-access nature promotes reproducibility and collaboration, while the real-world implications highlight the significance of data-driven strategies in efforts to reduce road traffic fatalities.

*Table 3.1. Data summary*

No.	Field Name	Description	Type
1	Accident_Index	Accident identifier	String
2	Location_Easting_OSGR	Local coordinate in the UK (X)	String
3	Location_Northing_OSGR	Local coordinate in the UK (Y)	String
4	Longitude	Longitude	String
5	Latitude	Latitude	String
6	Police_Force	Police unit	Int
7	Accident_Severity	Severity level	Int
8	Number_of_Vehicles	Number of vehicles damaged in the accident	Int
9	Number_of_Casualties	Number of casualties	Int
10	Date	Date of the accident	Date
11	Day_of_Week	Day of the week	Int
12	Time	Time of the accident	DateTime

13	Local_Authority_(District)	Name of the local district where the incident occurred	Int
14	Local_Authority_(Highway)	Name of the main road where the accident occurred	String
15	1st_Road_Class	Road classification	Int
16	1st_Road_Number	Road number	Int
17	Road_Type	Road type	Int
18	Speed_limit	Speed limit	Float
19	Junction_Detail	Junction details	Int
20	Junction_Control	Junction control	Int
21	2nd_Road_Class	Road classification	Int
22	2nd_Road_Number	Road number	Int
23	Pedestrian_Crossing-Human_Control	Control of pedestrian crossing by humans	Int
24	Pedestrian_Crossing-Physical_Facilities	Physical facilities for pedestrian crossing	Int
25	Light_Conditions	Lighting conditions	Int
26	Weather_Conditions	Weather conditions	Int

27	Road_Surface_Conditions	Road surface conditions	Int
28	Special_Conditions_at_Site	Special conditions	Int
29	Carriageway_Hazards	Carriageway hazards	Int
30	Urban_or_Rural_Area	Urban or rural area	Int
31	Did_Police_Officer_Attend_Scene_of_Accident	Did a police officer attend the accident scene?	Int
32	LSOA_of_Accident_Location	Geographic area of the accident location	String

The UK road traffic accident data schema also represents an advanced approach to capturing the multifaceted nature of traffic safety incidents. This schema integrates spatial-temporal coordinates, administrative classifications and contextual factors, allowing for a thorough analysis of accident patterns. The spatial dimension is represented through both Ordnance Survey grid references and standard geographic coordinates, while the temporal aspect is documented via date and time fields. This dual-coordinate system enhances compatibility with various mapping tools and facilitates time-based pattern analysis.

The schema implements a three-level severity classification system (fatal, serious, slight) that strikes a balance between detail and analytical utility. This classification, alongside casualty and vehicle counts, delivers critical metrics for prioritizing safety interventions. Notably, the schema includes extensive documentation of road infrastructure context—such as road classification, configuration, junction characteristics and pedestrian facilities. This detailed infrastructure data allows for an analysis of how design elements affect accident occurrence and severity.

Environmental conditions are recorded through fields detailing lighting, weather and road surface conditions. These factors often interact with human and vehicle elements to impact accident risk.

The urban/rural classification provides essential geographical context, as accident patterns vary significantly between these environments. Additionally, the incorporation of administrative data supports integration with socioeconomic and demographic datasets.

The schema's comprehensive structure reflects the complex interplay of factors influencing road safety outcomes. By encompassing this broad range of variables, the dataset facilitates evidence-based policy development and targeted interventions aimed at improving road safety throughout the United Kingdom.

*Table 3.2. Data dictionary*

Column Name	Description	Integer Value Meaning
Accident_Index	Unique identifier for each accident	Unique ID
Location_Easting_OSGR	Easting coordinate of the accident location (OS grid reference system)	Geographical coordinate
Location_Northing_OSGR	Northing coordinate of the accident location (OS grid reference system)	Geographical coordinate
Longitude	Longitude of the accident location	Geographical coordinate
Latitude	Latitude of the accident location	Geographical coordinate

Police_Force	Code representing the police force that recorded the accident	Varies by region
Accident_Severity	Severity level of the accident	1 = Fatal, 2 = Serious, 3 = Slight
Number_of_Vehicles	Number of vehicles involved in the accident	Count of vehicles
Number_of_Casualties	Number of casualties in the accident	Count of casualties
Date	Date of the accident	Date format (YYYY-MM-DD)
Time	Time of the accident	Time format (HH:MM)
Local_Authority_(District)	District-level administrative area code	Area code
Local_Authority_(Highway)	Highway management authority	Area code
1st_Road_Class	Classification of the main road where the accident occurred	1 = Motorway, 2 = A(M) Road, 3 = A Road, 4 = B Road, 5 = C Road, 6 = Unclassified/Not recorded
1st_Road_Number	Number of the main road	Road number

Road_Type	Type of road where the accident occurred	1 = Single carriageway, 2 = Dual carriageway, 3 = Other classified road types, 6 = One-way street, 7 = Special/other road types, 9 = Unspecified/other
Speed_limit	Speed limit at the accident location (mph)	Speed value
Junction_Detail	Type of junction	-1 = Not recorded, 0 = Not a junction, 1 = Roundabout, 2 = Mini-roundabout, 3 = T-junction, 5 = Crossroads, 6 = Complex junction, 7 = Skewed junction, 8 = Other junction types, 9 = Unspecified
Junction_Control	Type of junction control	-1 = Not recorded, 0 = No control, 1 = Stop sign, 2 = Give way, 3 = Traffic signal, 4 = Other control types
2nd_Road_Class	If the accident occurred at a junction, this is the	Same as 1st_Road_Class

	classification of the second road involved. If not at a junction, this value may be empty or meaningless.	
2nd_Road_Number	Number of the second road (if applicable)	Road number
Pedestrian_Crossing-Human_Control	Type of human pedestrian control	0 = No control, 1 = Controlled by a person (e.g., traffic officer), 2 = Other control type
Pedestrian_Crossing-Physical_Facilities	Type of pedestrian crossing infrastructure	0 = No crossing facility, 1 = Pedestrian crossing markings, 4 = Pelican crossing, 5 = Puffin crossing, 7 = Footbridge or subway
Light_Conditions	Light conditions at the time of the accident	1 = Daylight, 4 = Darkness with street lighting, 5 = Darkness without street lighting, 6 = Darkness with no lighting, 7 = Darkness with unknown lighting status

Weather_Conditions	Weather conditions at the time of the accident	-1 = Not recorded, 1 = Fine (no high winds), 2 = Fine with high winds, 3 = Rain (no high winds), 4 = Rain with high winds, 5 = Snow (no high winds), 6 = Snow with high winds, 7 = Fog/mist, 8 = Other conditions, 9 = Unspecified
Road_Surface_Conditions	Condition of the road surface	-1 = Not recorded, 1 = Dry, 2 = Wet/Damp, 3 = Snow, 4 = Ice, 5 = Flooded
Special_Conditions_at_Site	Special conditions at the accident site	-1 = Not recorded, 0 = No special conditions, 1 = Road surface defective/uneven, 2 = Foreign object or slippery surface, 3 = Other obstructions, 6 = Other unusual conditions, 7 = Unspecified



Carriageway_Hazards	Hazards on the carriageway	-1 = Not recorded, 0 = No hazards, 1 = Road surface defective/uneven, 2 = Foreign object or slippery surface, 3 = Other hazards, 6 = Other unusual conditions, 7 = Unspecified
Urban_or_Rural_Area	Area type where the accident occurred (urban or rural)	1 = Urban, 2 = Rural
Did_Police_Officer_Attend_Scene_of_Accident	Did a police officer attend the accident scene?	-1 = Not recorded, 1 = Yes, 2 = No, 3 = Unknown
LSOA_of_Accident_Location	Lower-layer Super Output Area (LSOA) of the accident location	Area code

### 3.2. Data Transformation

Data transformation represents a crucial step in the data preprocessing journey where raw data is systematically modified to make it more suitable for analysis. When we first collect data which is our traffic accident records. It often comes in a format that isn't immediately ready for analysis. Data transformation involves converting this raw data into more useful forms that will help us uncover meaningful patterns and relationships. Think of data

transformation as similar to translating a book from an unknown language into a mother language. The content remains essentially the same, but it becomes accessible and useful. In our traffic accident analysis, we implemented several important transformation approaches:

- Creation of binary flags for missing data: Rather than simply discarding the 10 records with missing spatial coordinates, we created a binary flag system to mark these instances. This approach preserves valuable information contained in other fields while transparently tracking data quality issues. It's similar to marking a page in a book that has a printing error—we acknowledge the imperfection but keep the rest of the content.
- Standardization of categorical variables: We addressed inconsistencies in categorical fields like the Lower-layer Super Output Area (LSOA), where some records contained empty strings instead of proper area codes. By implementing another binary flag, we maintained analytical transparency while ensuring our analysis wouldn't be skewed by these anomalies.
- Temporal aggregation: Raw timestamp data was transformed into more meaningful temporal units (hours, days of week, months) to facilitate pattern identification across different time scales. This transformation allowed us to discover the pronounced peaks in accident frequency during morning and evening commutes.
- Spatial clustering: Geographic coordinates were transformed through clustering techniques to identify accident hotspots, converting raw location data into actionable insights about high-risk areas.
- Creation of derived variables: We analyzed relationships between existing variables (like vehicle counts and casualty numbers) to generate new insights about accident characteristics, effectively transforming individual data points into relationship metrics.

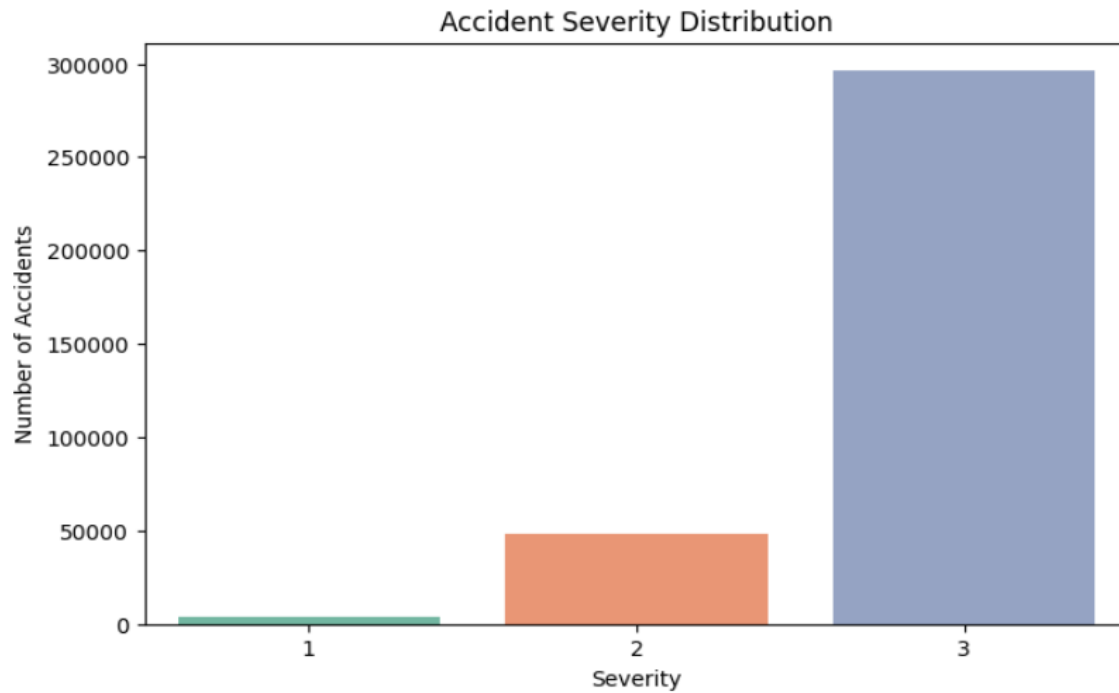
These transformation techniques were not merely technical exercises, they directly enabled our key findings. The binary flag system for missing coordinates allowed us to maintain all 10 records with incomplete location data, preserving valuable information about

accident severity, vehicle involvement and timing that would otherwise have been lost. Our temporal transformations revealed critical patterns in accident distribution across different time periods, highlighting the relationship between commuting hours and accident frequency. The spatial transformations uncovered significant variation in accident density across local authority districts, pinpointing specific hotspots that warrant targeted safety interventions. By handling the LSOA field inconsistencies transparently, we maintained data integrity while maximizing the available information for geographic analysis.

These transformation approaches underscore a fundamental principle in data analysis: the goal is not to create a perfectly clean dataset by eliminating all problematic records, but rather to transform raw data in ways that preserve as much useful information as possible while accounting for limitations. Our transformation strategy embraced the inherent messiness of real-world accident reporting systems. Instead of pursuing an idealized but smaller dataset, we transformed the data to work with its imperfections, resulting in a more comprehensive and nuanced understanding of traffic safety patterns. This approach allows for more robust and realistic conclusions, acknowledging that in real-world applications, data will rarely be perfect, but can still yield valuable insights when transformed appropriately.

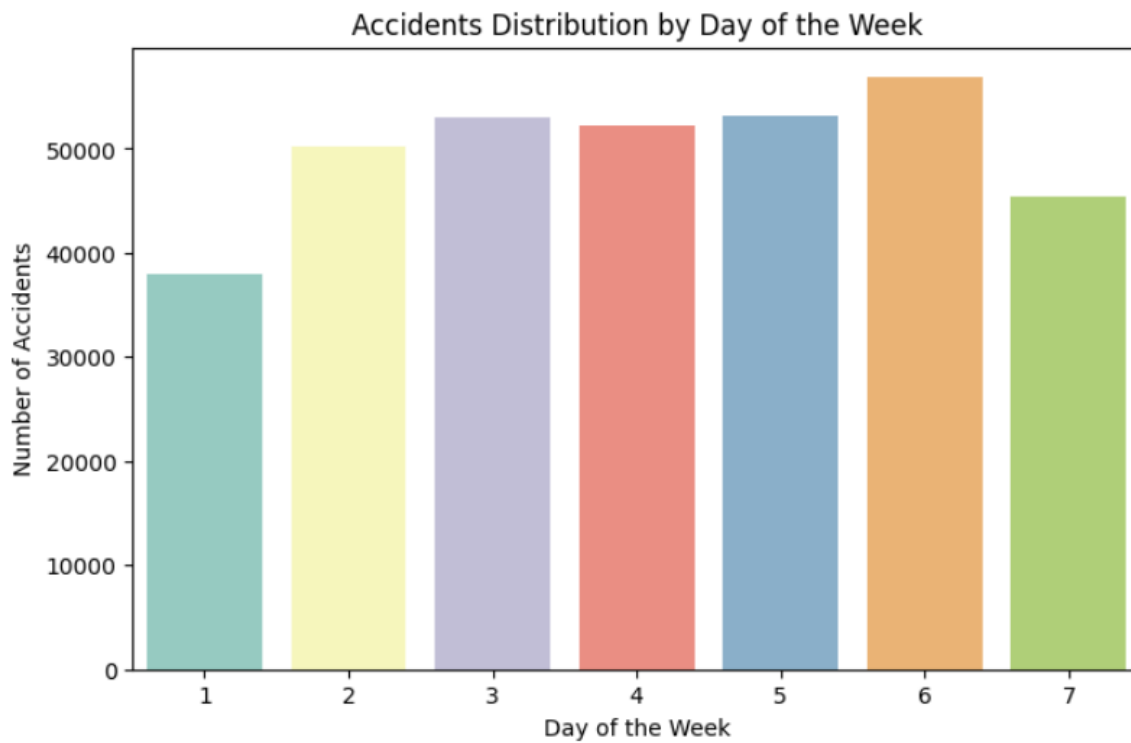
Here are some visualizations to know more about the dataset after the transformation:

- Accident Severity



*Figure 3.1. Accident Severity Distribution*

- Day of Week



*Figure 3.2. Accidents Distribution by Day of the Week*

- Road Type

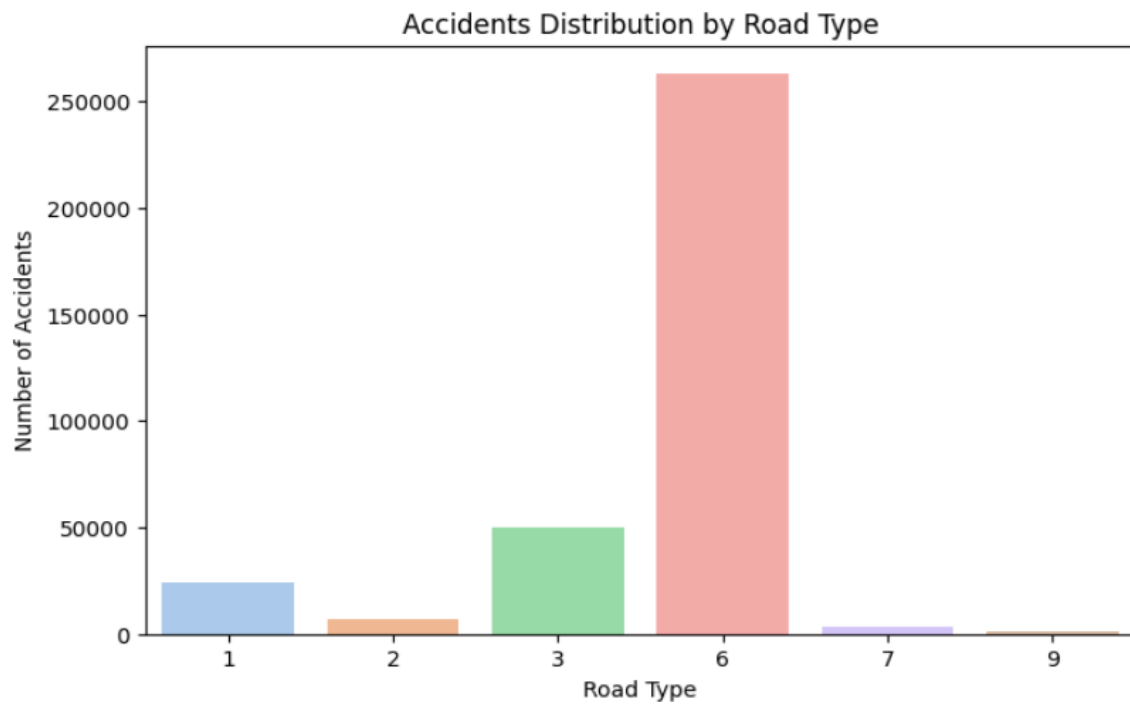


Figure 3.3. Accidents Distribution by Road Type

- Light Conditions

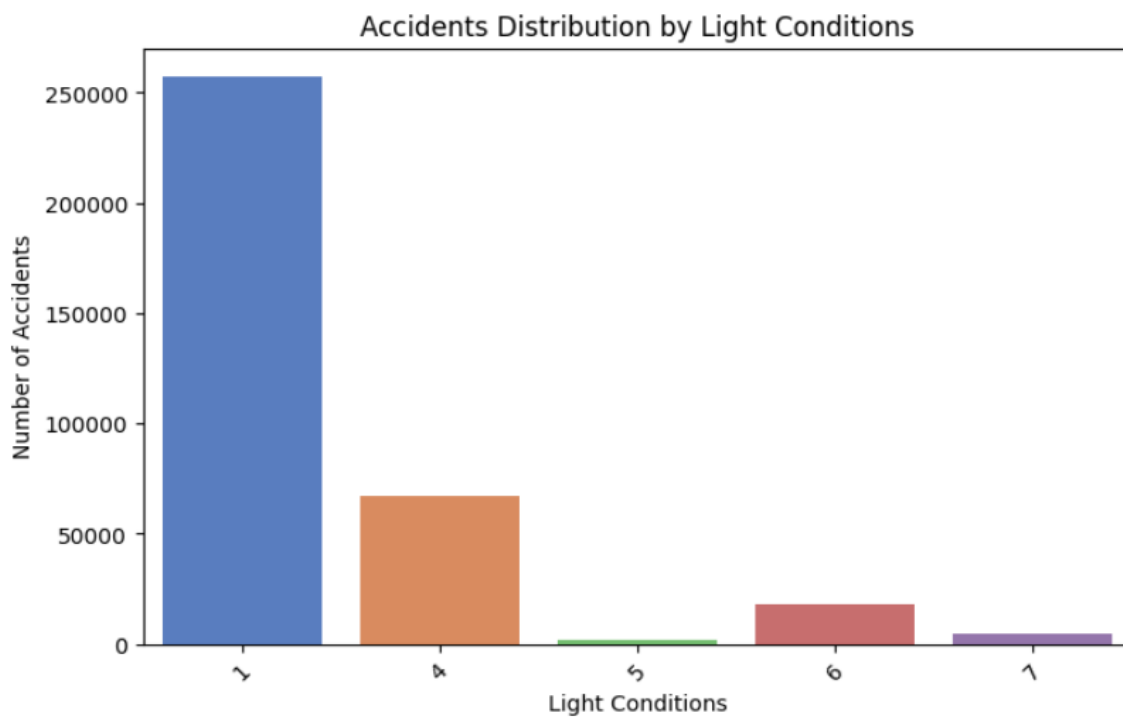


Figure 3.4. Accidents Distribution by Light Conditions

- Weather Conditions

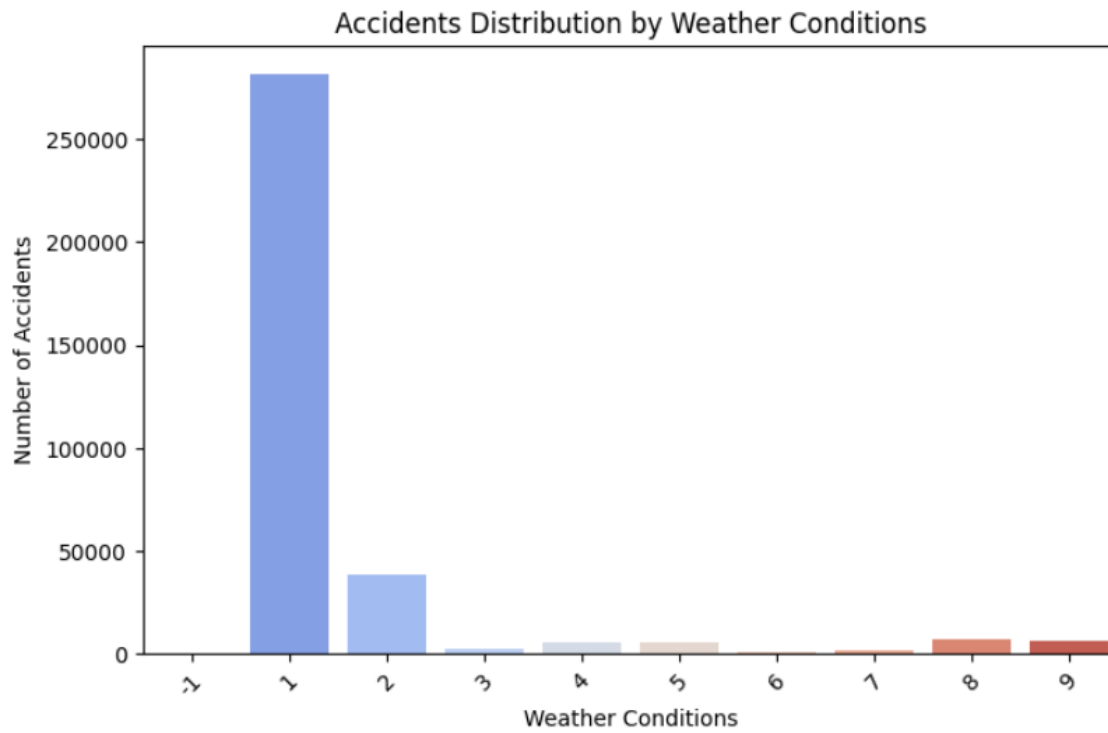


Figure 3.5. Accidents Distribution by Weather Conditions

- Accidents by Month

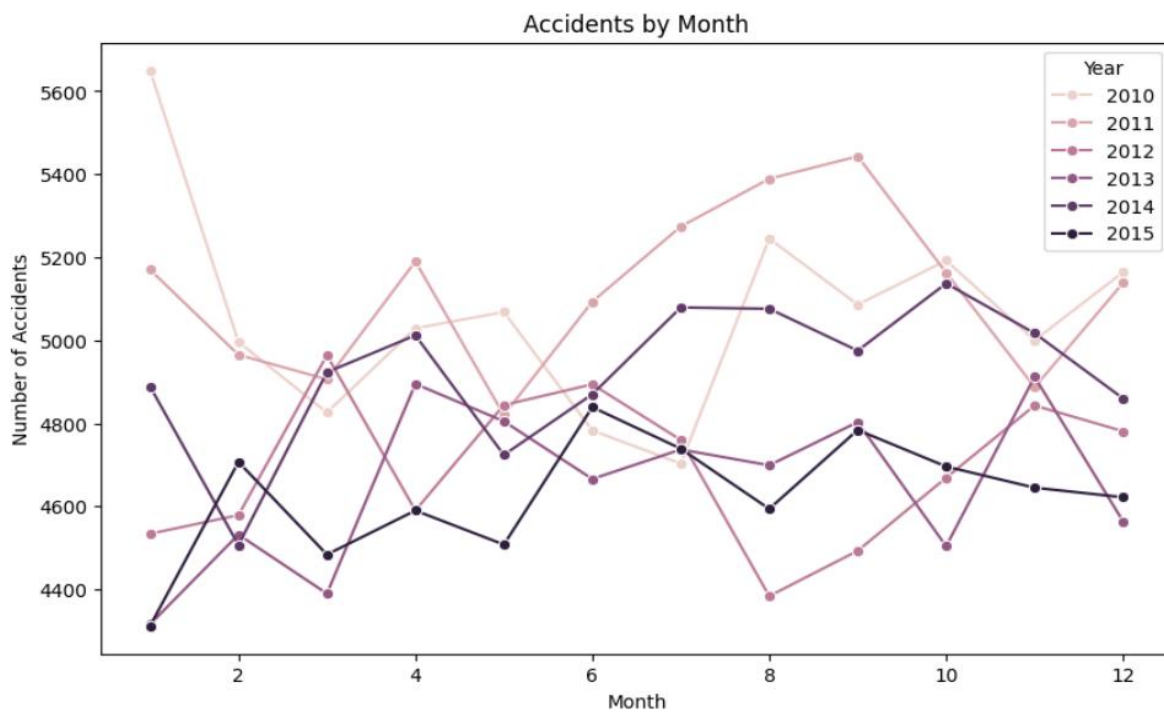
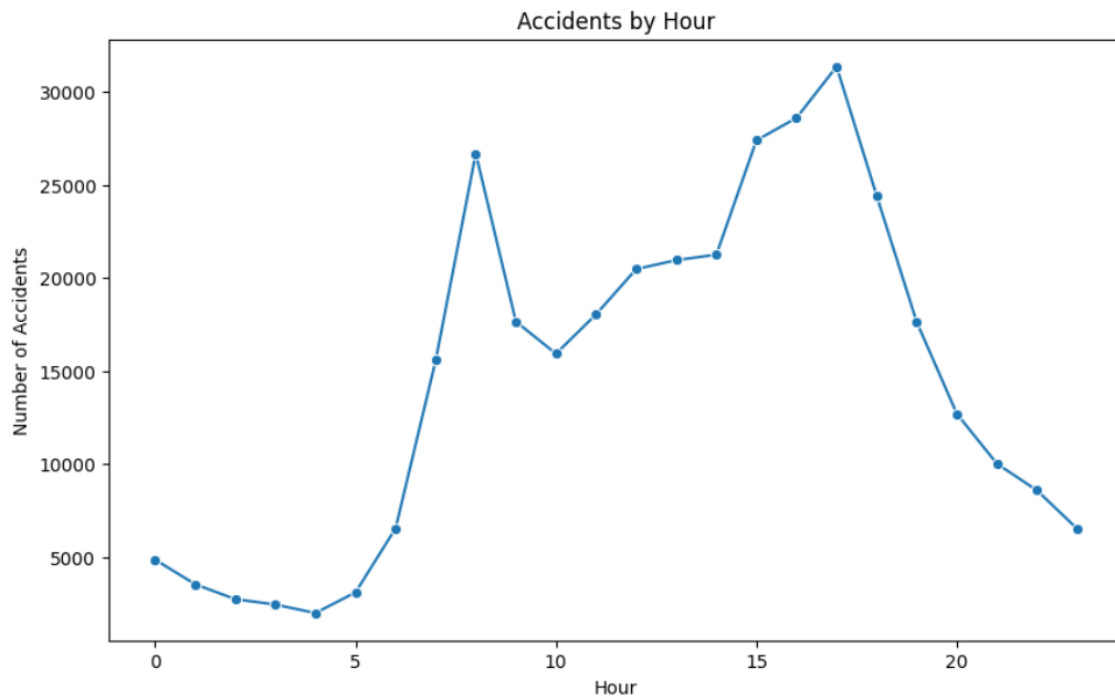


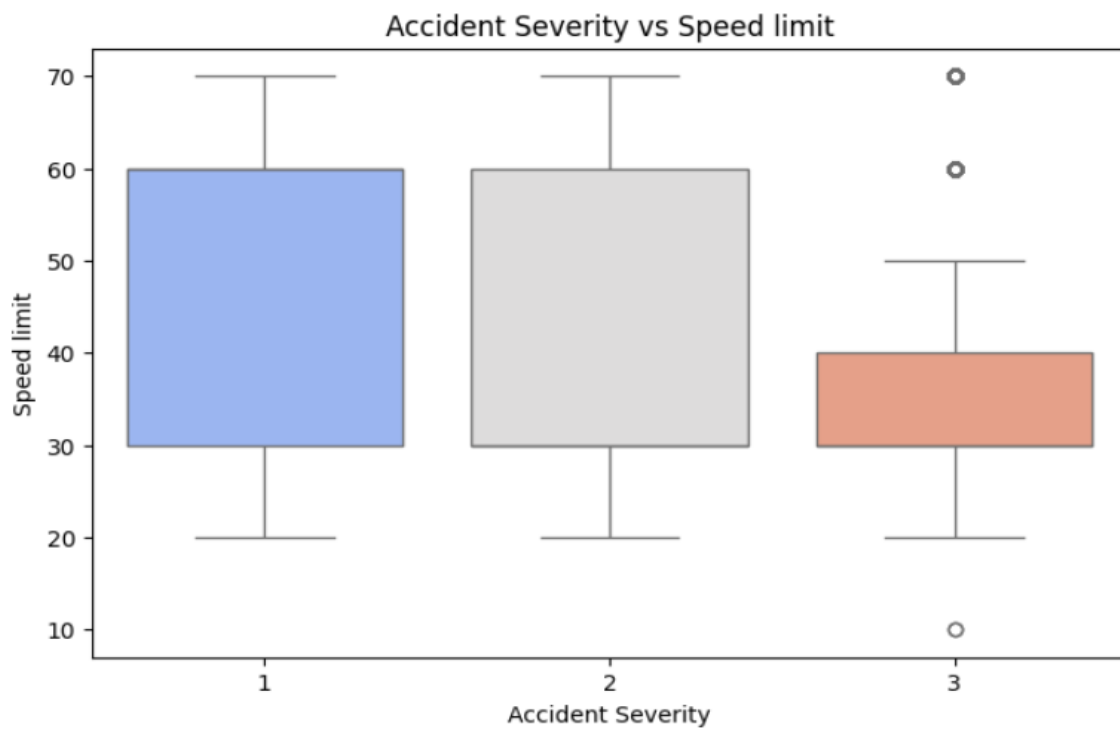
Figure 3.6. Accidents by Month

- Accidents by Hour



*Figure 3.7. Accidents by Hour*

- Accident Severity vs Speed Limit



*Figure 3.8. Accident Severity & Speed Limit*

- Number of Vehicles vs Number of Casualties

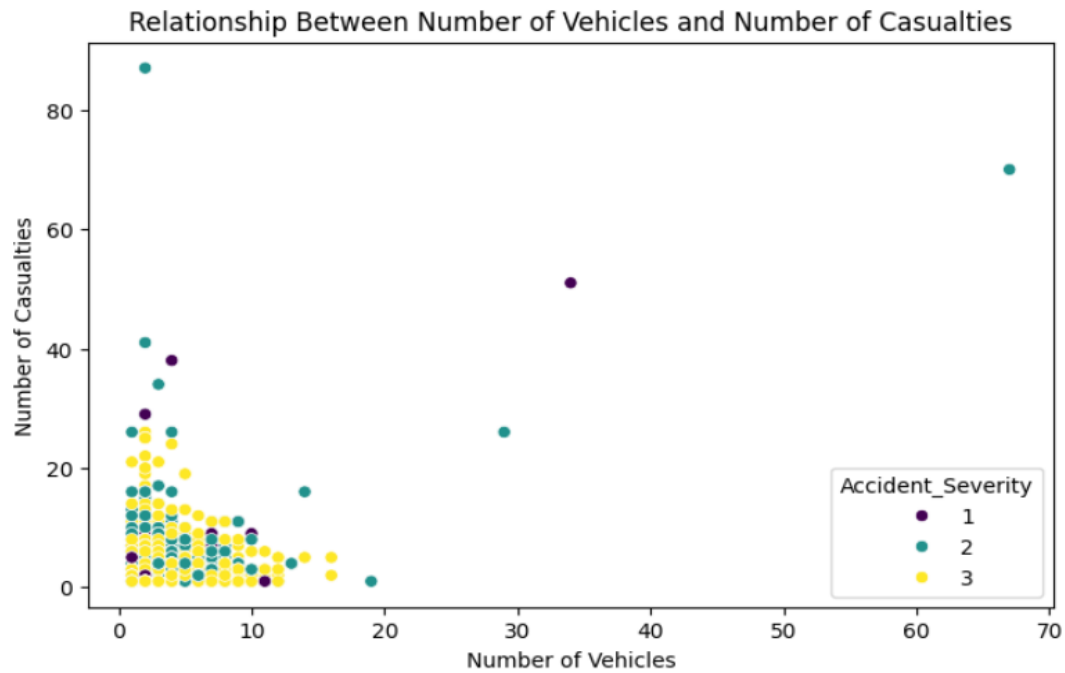


Figure 3.9. Relationship Between Number of Vehicles and Number of Casualties

- Accidents by Weather and Severity

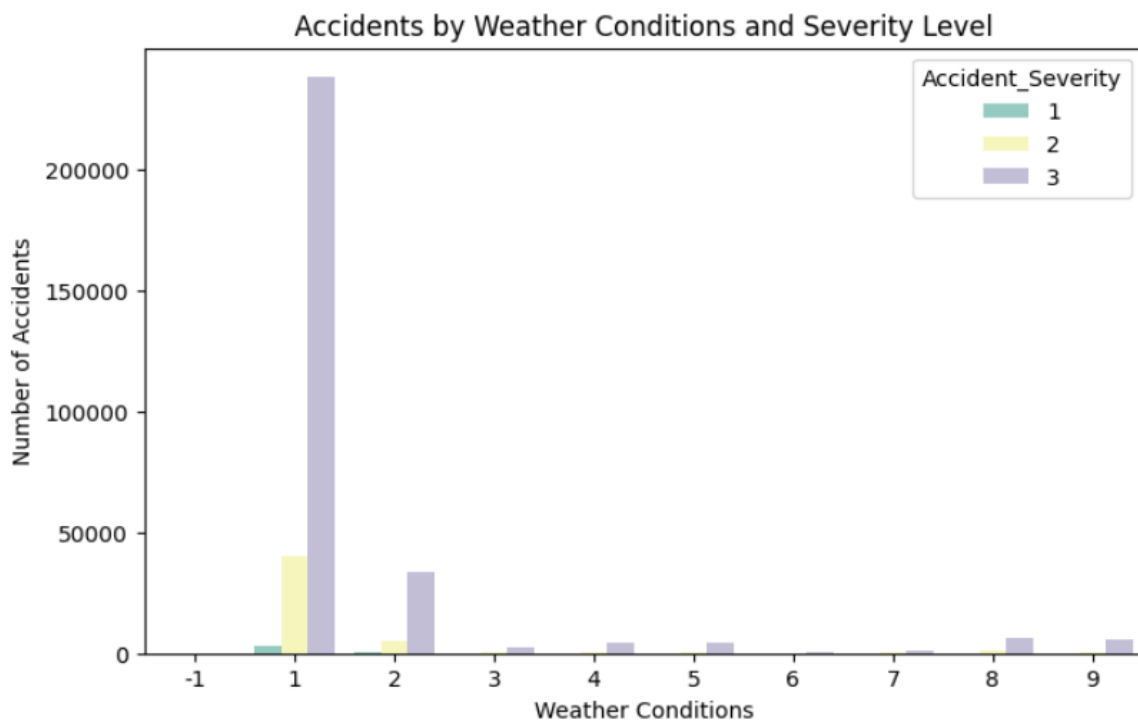


Figure 3.10. Accidents by Weather Conditions and Severity Level



- Accidents by Road Surface and Speed Limit

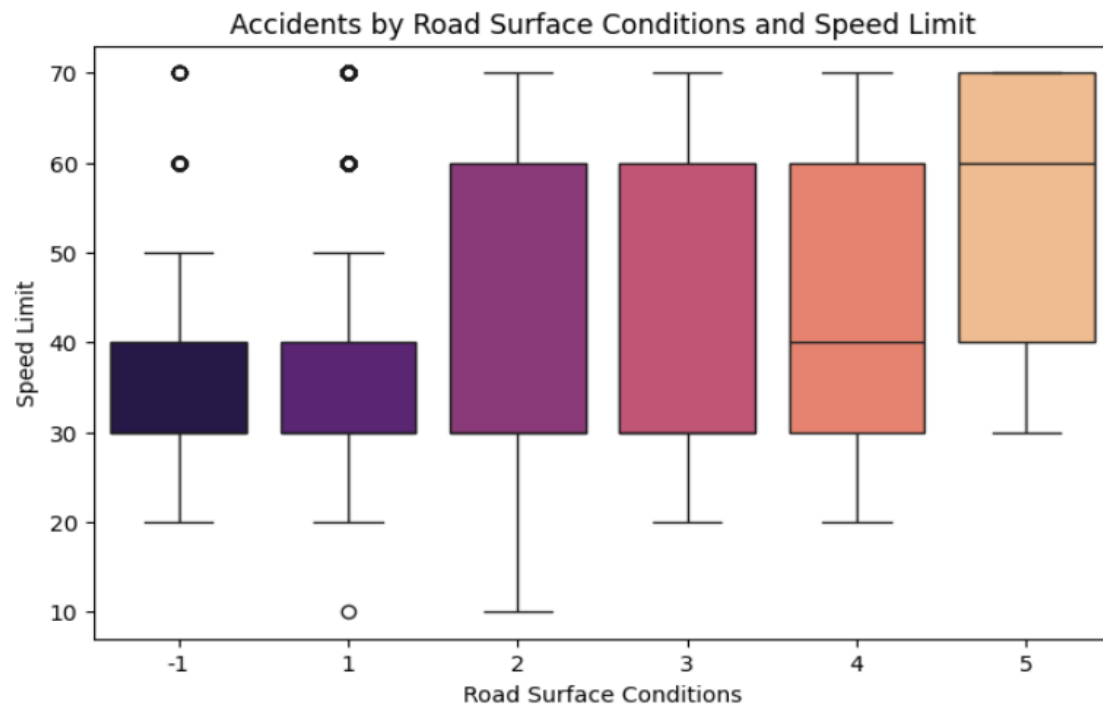


Figure 3.11. Accidents by Road Surface Conditions and Speed Limit

- Accidents by Junction Detail and Urban/Rural Area

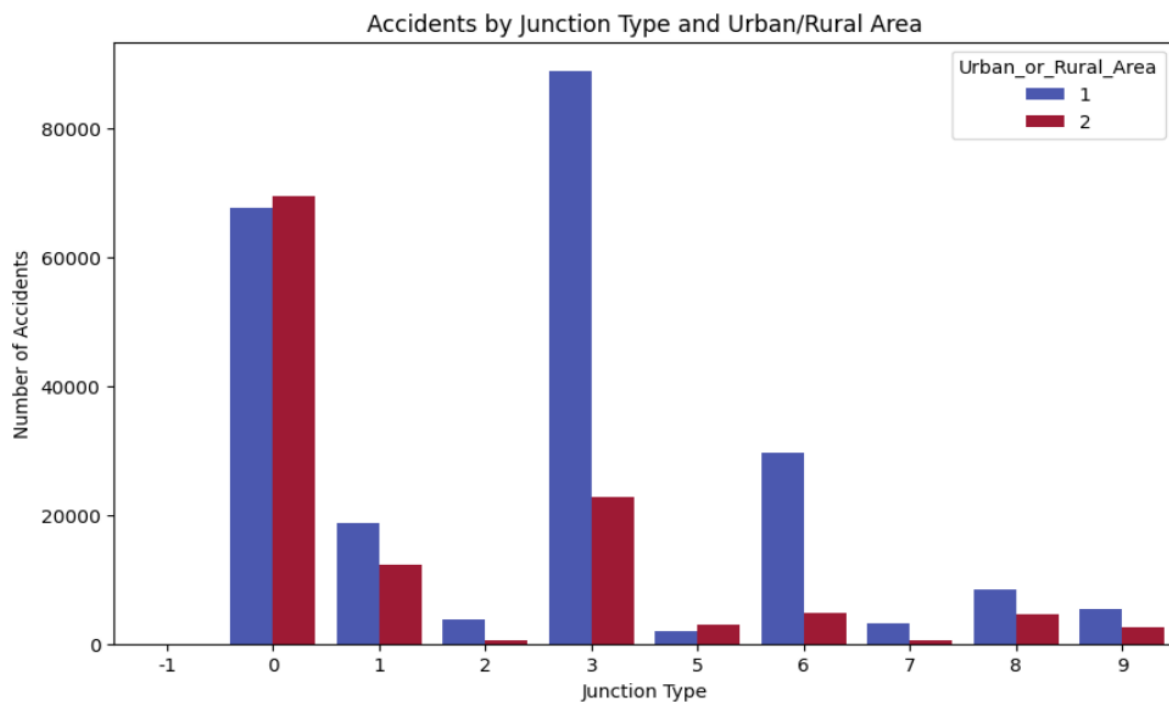


Figure 3.12. Accidents by Junction Type and Urban/Rural Area

- Severity, Casualties and Road Type

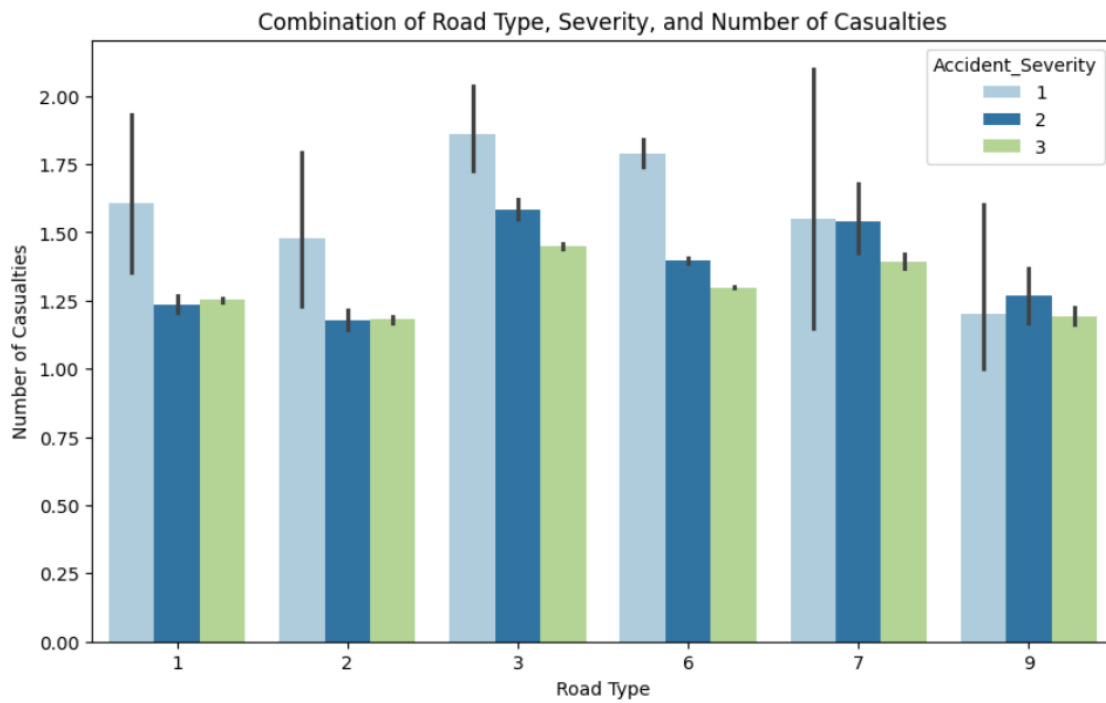


Figure 3.13. Combination of Road Type, Severity, and Number of Casualties

- Weather, Light Conditions and Severity

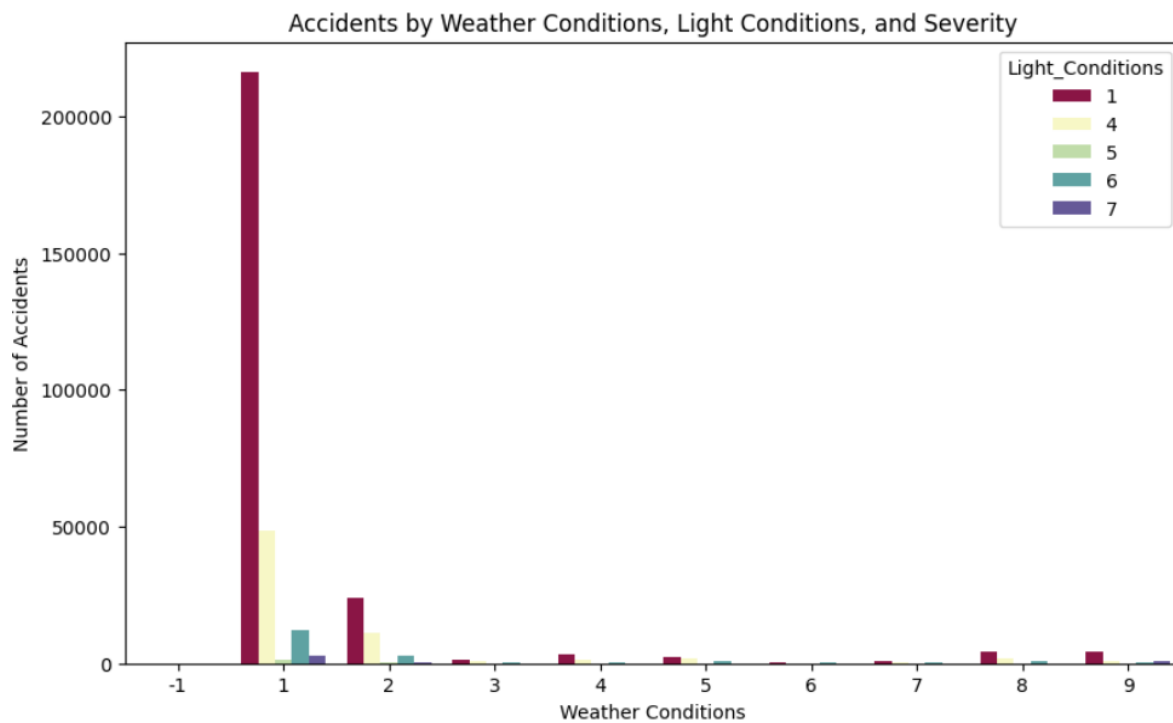
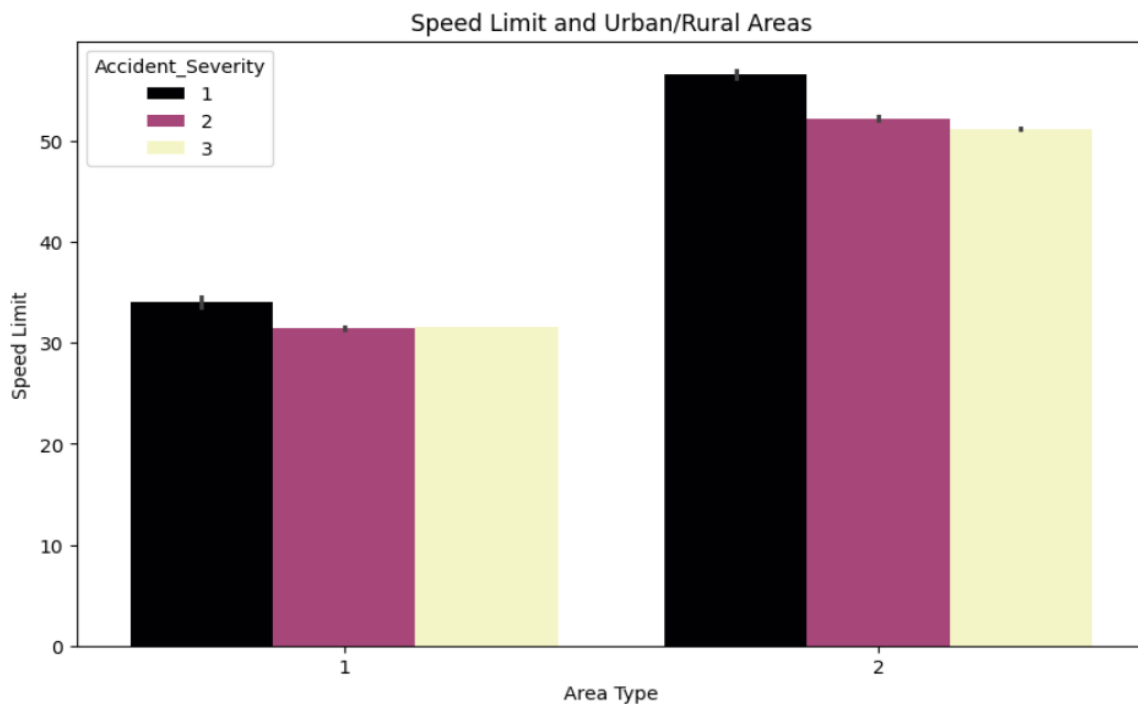


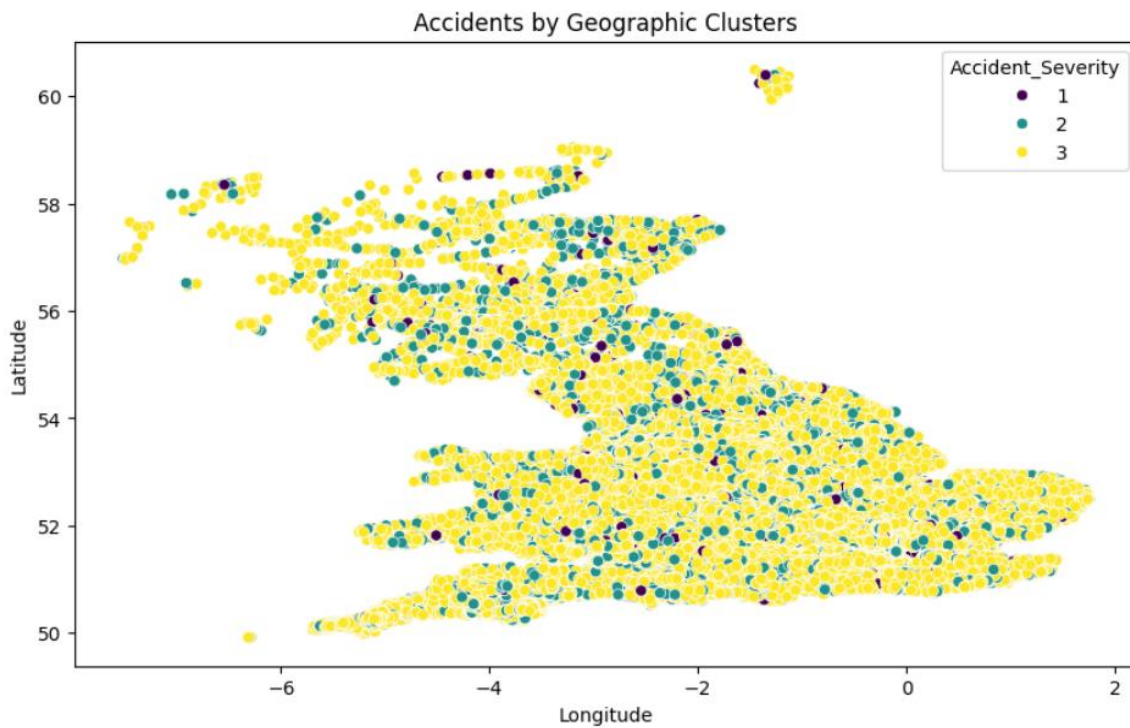
Figure 3.14. Accidents by Weather Conditions, Light Conditions, and Severity

- Speed Limit and Urban/Rural Areas



*Figure 3.15. Speed Limit and Urban/Rural Area*

- Accidents by Geographic Clusters



*Figure 3.16. Accidents by Geographic Clusters*

- Police Attendance and Severity

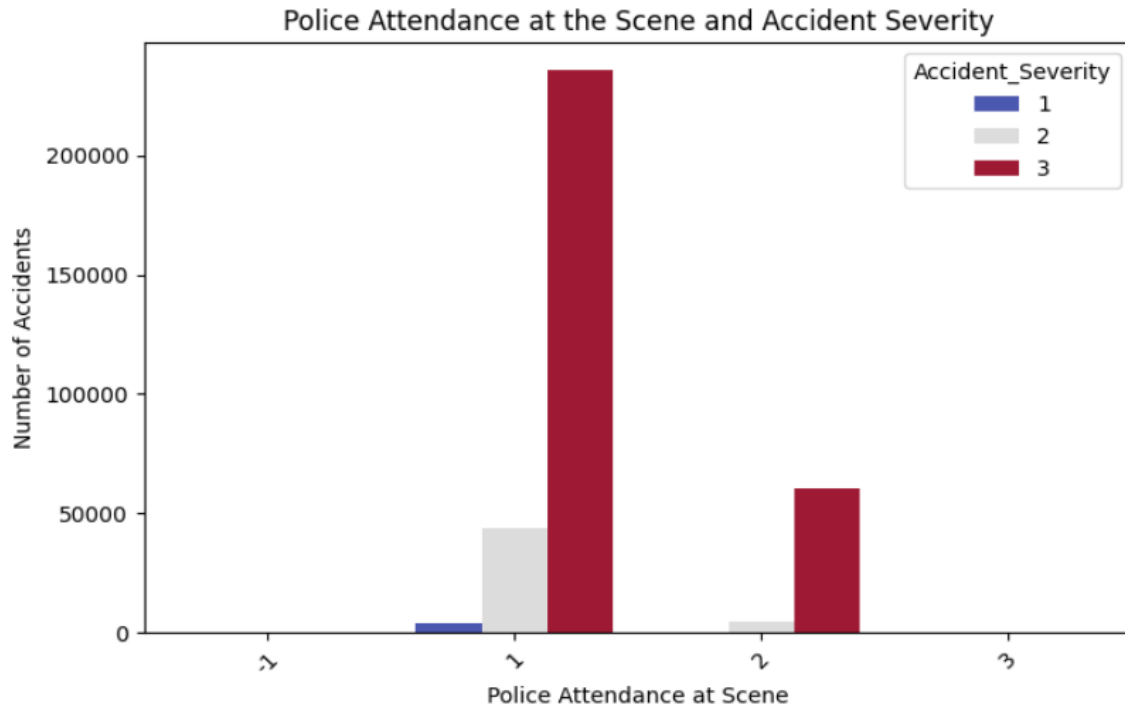


Figure 3.17. Police Attendance at the Scene and Accident Severity

### 3.3. Data Modeling

#### 3.3.1. Relationship

Table 3.3. Relationship between dimensional and fact tables

No.	Relationship	Type	Description
1	Dim_Date → Fact_Accidents	1 - n	One date can have multiple accidents, but each accident occurs on a single date.
2	Dim_Time → Fact_Accidents	1 - n	One time entry (hour and minute) can be associated with multiple accidents, but each accident has a specific time.

3	Dim_LightConditions Fact_Accidents	→	1 - n	One light condition type can be linked to multiple accidents, but each accident has one specific light condition.
4	Dim_Police Fact_Accidents	→	1 - n	One police force can record multiple accidents, but each accident is recorded by a single police force.
5	Dim_RoadType Fact_Accidents	→	1 - n	One road type can be associated with multiple accidents, but each accident occurs on a specific road type.
6	Dim_AccidentSeverity Fact_Accidents	→	1 - n	One severity level can be assigned to multiple accidents, but each accident has only one severity level.
7	Dim_RoadSurfaceConditions → Fact_Accidents		1 - n	One road surface condition can be linked to multiple accidents, but each accident has a specific road surface condition.
8	Dim_WeatherConditions Fact_Accidents	→	1 - n	One weather condition type can be linked to multiple accidents, but each accident occurs under a specific weather condition.
9	Dim_UrbanorRuralArea Fact_Accidents	→	1 - n	One urban/rural area classification can be linked to multiple accidents, but each accident occurs in one specific area type.

### 3.3.2. Dimension tables and Fact tables

*Table 3.4. Dim\_Date*

Column Name	Data Type	Description
Date	DATE	Surrogate key for date
Year	INT	Numeric value of year
Month	INT	Numeric value of month
Month Name	NVARCHAR(4000)	Name of the month (Jan, Feb...)
Day_Of_Month	INT	Day of the month (1-31)
DayName	NVARCHAR(4000)	Name of the day (Monday, Tuesday...)

*Table 3.5. Dim\_Time*

Column Name	Data Type	Description
Time	NVARCHAR(4000)	Surrogate key for time
Hour	INT	Hour component (0-12)
Minute	INT	Minute component (0-59)

*Table 3.6. Dim\_LightConditions*

Column	Data Type	Description
LightConditionsKey	INT (PK)	Surrogate key for Light Conditions
Light_Conditions	INT	Code representing light conditions
Description	NVARCHAR(4000)	1 = Daylight, 4 = Darkness with street lighting, 5 = Darkness without street lighting, 6 = Darkness with

		no lighting, 7 = Darkness with unknown lighting status
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)
Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)

*Table 3.7. Dim\_Police*

Column Name	Data Type	Description
PoliceForceKey	INT (PK)	Surrogate key for Police Force
Police_Force	INT	Code for the police force recording the accident
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)
Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)

*Table 3.8. Dim\_RoadType*

Column Name	Data Type	Description
-------------	-----------	-------------

RoadTypeKey	INT (PK)	Surrogate key for Road Type
Road_Type	INT	Code representing the type of road
Description	NVARCHAR(4000)	1 = Single carriageway, 2 = Dual carriageway, 3 = Other classified road types, 6 = One-way street, 7 = Special/other road types, 9 = Unspecified/other
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)
Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)

*Table 3.9. Dim\_AccidentSeverity*

Column Name	Data Type	Description
AccidentSeverityKey	INT (PK)	Surrogate key for Accident Severity
Accident_Severity	INT	Code for accident severity



Description	NVARCHAR(4000)	1 = Fatal, 2 = Serious, 3 = Slight
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)
Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)

*Table 3.10. Dim\_RoadSurfaceConditions*

Column Name	Data Type	Description
RoadSurfaceConditionsKey	INT (PK)	Surrogate key for Road Surface Conditions
Road_Surface_Conditions	INT	Code representing surface conditions
Description	NVARCHAR(4000)	-1 = Not recorded, 1 = Dry, 2 = Wet/Damp, 3 = Snow, 4 = Ice, 5 = Flooded
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)

Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)
--------	-----	---------------------------------------

*Table 3.11. Dim\_WeatherConditions*

Column Name	Data Type	Description
WeatherConditionsKey	INT (PK)	Surrogate key for Weather Conditions
Weather_Conditions	INT	Code representing weather
Description	NVARCHAR(4000)	-1 = Not recorded, 1 = Fine (no high winds), 2 = Fine with high winds, 3 = Rain (no high winds), 4 = Rain with high winds, 5 = Snow (no high winds), 6 = Snow with high winds, 7 = Fog/mist, 8 = Other conditions, 9 = Unspecified
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)

Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)
--------	-----	---------------------------------------

*Table 3.12. Dim\_UrbanorRuralArea*

Column Name	Data Type	Description
UrbanRuralAreaKey	INT (PK)	Surrogate key for Urban/Rural area
Urban_or_Rural_Area	INT	1 = Urban, 2 = Rural
Start_Date	DATE	Effective start date (SCD Type 2)
End_Date	DATE NULL	Effective end date (NULL if current record)
Status	BIT	1 = Active, 0 = Inactive (SCD Type 2)

*Table 3.13. Fact\_Accidents*

Column Name	Data Type	Description
AccidentIndex	NVARCHAR(4000) (PK)	Unique accident identifier from source
PoliceKey	INT (FK)	Foreign key to Dim_Police
LightConditionsKey	INT (PK)	Foreign key to Dim_LightConditions
RoadTypeKey	INT (PK)	Foreign key to Dim_RoadType

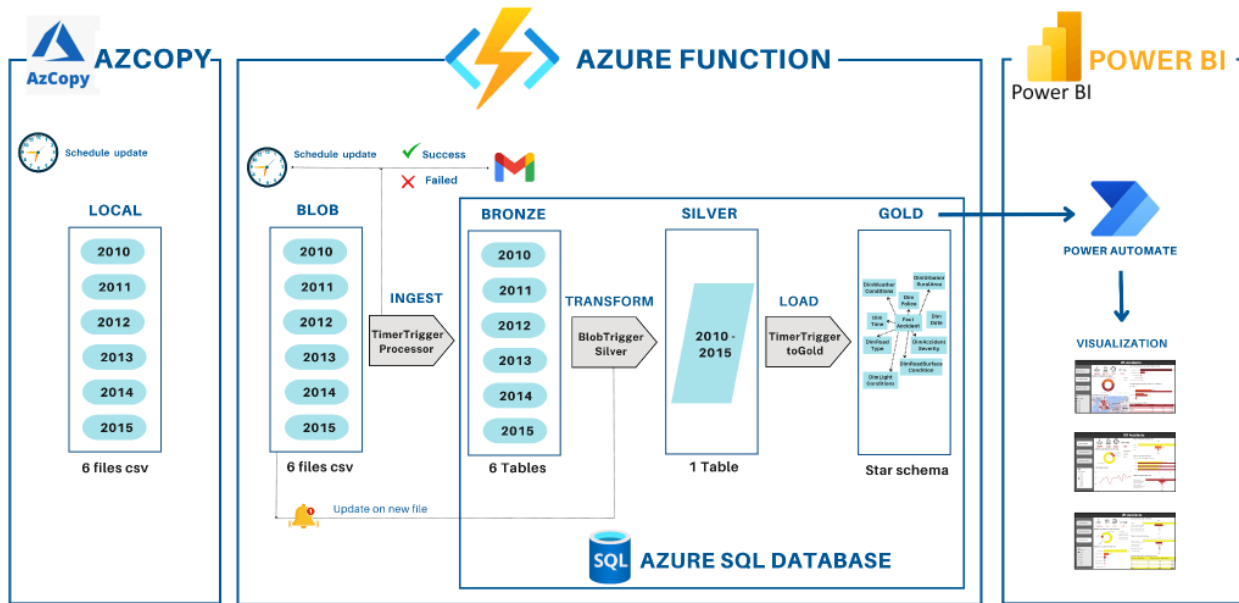
AccidentSeverityKey	INT (PK)	Foreign key to Dim_AccidentSeverity
RoadSurfaceConditionsKey	INT (PK)	Foreign key to Dim_RoadSurfaceConditions
WeatherConditionsKey	INT (PK)	Foreign key to Dim_WeatherConditions
UrbanRuralAreaKey	INT (PK)	Foreign key to Dim_UrbanRuralArea
Longitude	FLOAT	Geographic longitude
Latitude	FLOAT	Geographic latitude
Local_Authority_District	INT	District authority code
Local_Authority_Highway	NVARCHAR(4000)	Highway authority name
Date	DATE	Accident date
Time	NVARCHAR(4000)	Accident hour and minute
Number_of_Vehicles	INT	Number of vehicles involved
Number_of_Casualties	INT	Number of casualties
Speed_Limit	INT	Speed limit (mph)

## **CHAPTER 4. EXPERIMENTING WITH THE ETL PROCESS ON AZURE FUNCTIONS**

*This chapter details the ETL (Extract, Transform, Load) process implemented in Microsoft Azure for UK traffic accident data. It describes the data ingestion from Azure Blob Storage, the transformation process utilizing Azure Functions, and the structured storage within Azure SQL Database, organized into bronze, silver, and gold layers. Notable automation techniques, including event-driven triggers and scheduled processing, facilitate real-time updates. Additionally, the chapter addresses data cleaning, schema design, and performance optimization, establishing a robust foundation for efficient data analysis and visualization in subsequent phases of the project.*

### **4.1. ETL process on Azure Functions**

The BI solution for the data of traffic accidents in the UK is implemented on the cloud computing platform Microsoft Azure. The execution process applies maximum automation through scheduled events starting from local, from when the data files are stored on the device's hard drive. Overall, the implementation process will include 3 main phases: (1) Ingest from local to blob storage (2) ETL all data into three architectural layers in the SQL database and the final phase (3) Upload data to PowerBI and proceed with visualization.



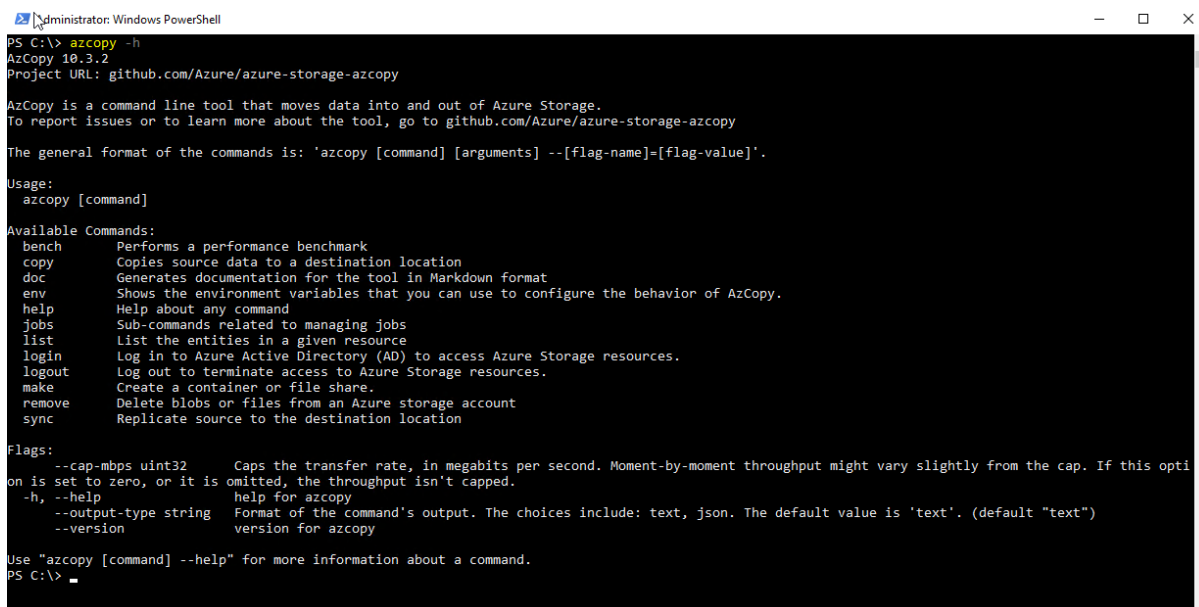
*Figure 4.1. Business Intelligence Solutions on the Microsoft Azure Platform*

- (1) Ingest from local to blob storage: in this phase, the data will be updated based on a fixed date and time set up in advance at local and uploaded to the blob storage account.
- (2) ETL all data into three architectural layers in the SQL database and the final phase: this is the most complex phase in the data ingestion process, as it requires processing data through multiple layers, plus setting up some triggers to enhance the ETL process. The ETL process will be fully executed through Azure Functions and conclude with the recording of data into the SQL database. It starts with the bronze layer where raw data is stored; the data in the bronze layer will be updated according to a pre-scheduled time, and additionally, during the data loading process, email notifications are sent to confirm whether the loading process is completed or if any errors occur. When the data enters the silver layer, it must undergo several transformation techniques to ensure the data is standardized and properly structured; the silver layer is also equipped with trigger operations to automatically load data whenever a new file is updated. Finally, the data when entering the bronze layer will follow the data warehouse structure, which is the star schema.
- (3) Upload data to PowerBI and proceed with visualization: this phase focuses on deep analysis and understanding of current and past insights visualized through dashboards.

Following that, valuable recommendations and insights are derived, while also addressing the business questions posed from the beginning.

#### 4.1.1. Az Copy to blob storage

To efficiently and securely transfer data to Azure Blob Storage, the system uses AzCopy, a powerful command-line tool developed by Microsoft. AzCopy supports high-speed data transfers between Azure storage services, ensuring data security and seamless integration with other Azure services. This tool not only optimizes transfer performance but also supports several security features, such as encryption and authentication using secure methods.

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The terminal shows the command `azcopy -h` being executed, which displays the AzCopy version (10.3.2) and the project URL (github.com/Azure/azure-storage-azcopy). It then provides a detailed description of AzCopy as a command-line tool for moving data between Azure storage services. The terminal lists available commands such as `bench`, `copy`, `doc`, `env`, `help`, `jobs`, `list`, `login`, `logout`, `make`, `remove`, and `sync`, each with a brief description. It also lists flags like `--cap-mbps`, `--help`, `--output-type`, and `--version` with their respective functions. The terminal ends with the prompt `PS C:\>`.

```
Administrator: Windows PowerShell
PS C:\> azcopy -h
AzCopy 10.3.2
Project URL: github.com/Azure/azure-storage-azcopy

AzCopy is a command line tool that moves data into and out of Azure Storage.
To report issues or to learn more about the tool, go to github.com/Azure/azure-storage-azcopy

The general format of the commands is: 'azcopy [command] [arguments] --[flag-name]=[flag-value]'.

Usage:
  azcopy [command]

Available Commands:
  bench      Performs a performance benchmark
  copy       Copies source data to a destination location
  doc        Generates documentation for the tool in Markdown format
  env        Shows the environment variables that you can use to configure the behavior of AzCopy.
  help       Help about any command
  jobs       Sub-commands related to managing jobs
  list       List the entities in a given resource
  login      Log in to Azure Active Directory (AD) to access Azure Storage resources.
  logout     Log out to terminate access to Azure Storage resources.
  make       Create a container or file share.
  remove     Delete blobs or files from an Azure storage account
  sync       Replicate source to the destination location

Flags:
  --cap-mbps uint32  Caps the transfer rate, in megabits per second. Moment-by-moment throughput might vary slightly from the cap. If this option is set to zero, or it is omitted, the throughput isn't capped.
  -h, --help         help for azcopy
  --output-type string  Format of the command's output. The choices include: text, json. The default value is 'text'. (default "text")
  --version          version for azcopy

Use "azcopy [command] --help" for more information about a command.
PS C:\>
```

*Figure 4.2. Azcopy tool settings screen*

After downloading and extracting the AzCopy tool, users need to install and configure it to use it conveniently without having to type the full path every time. This is achieved by configuring the PATH environment variable for AzCopy. Configuring this allows users to run AzCopy from any directory on the system without needing to specify the installation path.

To transfer data to Azure Blob Storage, users need to have access to an Azure storage account. One of the most popular and convenient authentication methods is using a SAS Token (Shared Access Signature). A SAS Token provides temporary access to Azure

services without requiring the primary account credentials, helping to protect the account from security risks and reducing the potential for misuse.

With AzCopy, users can transfer data to Azure Blob Storage using simple commands. To make the process more convenient, users can save these commands into a PowerShell script, which automates the transfer tasks and minimizes errors due to manual execution. Moreover, to ensure that the data transfer occurs automatically and on schedule, users can configure Task Scheduler on the Windows operating system. This allows AzCopy to automatically perform data transfer tasks at predefined times, such as backing up or synchronizing data periodically, without user intervention. Specifically, the system will be configured to run at 12:00 AM every day and although the time to transfer data depends on the file size, with the current setup, each transfer will take approximately 20 to 30 seconds.

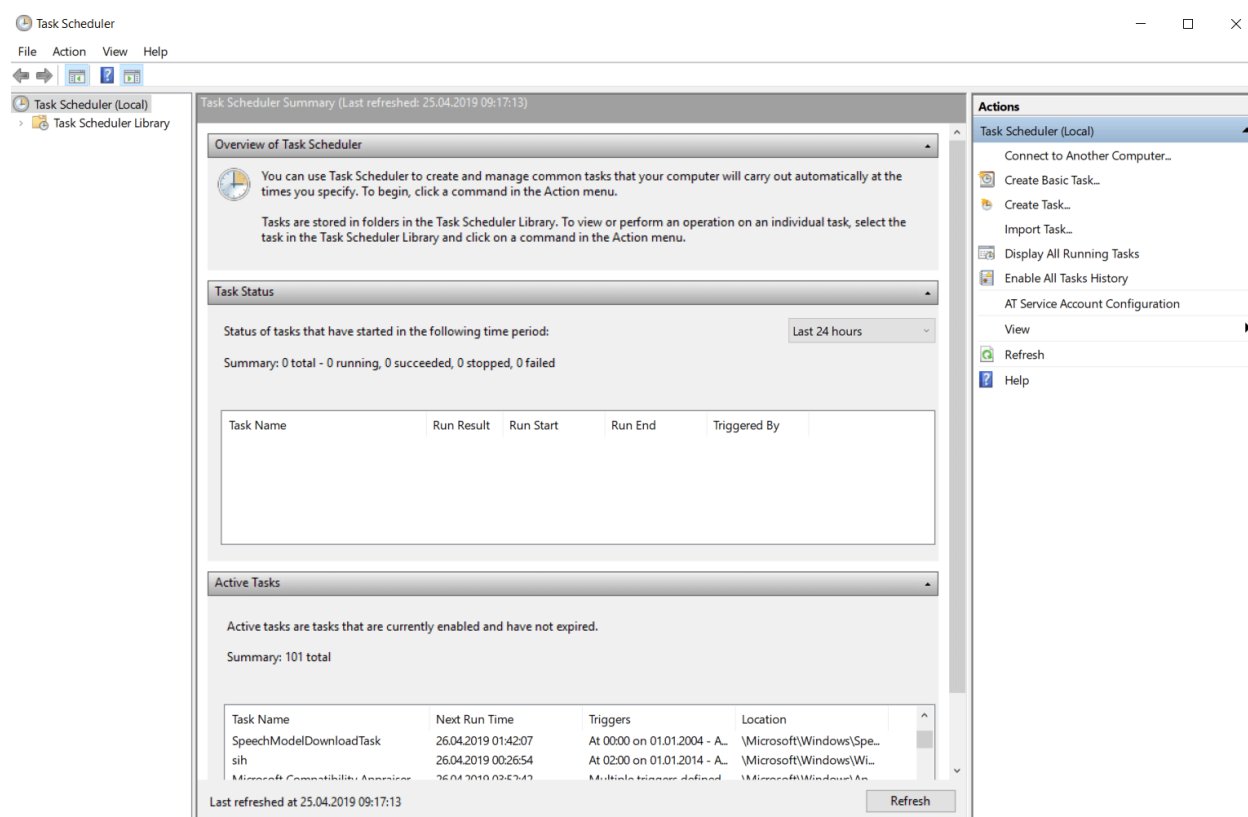


Figure 4.3. Task Scheduler tool settings screen



#### 4.1.2. Azure Functions Deployment on Azure Portal

To start the ETL process using the Azure Function tool, creating a FunctionApp on the Azure Portal is mandatory. The FunctionApp named RawBronzeSilverGoldlayer is created and will link directly to the storage account; here, the author group selects the hosting option for the function app as Consumption. After successfully creating the function app on the Azure Portal environment, the author group uses the local environment, VS Code, to ingest data into the 3 layers in the database. Therefore, three functions—TimerBlobProcessor, BlobTriggerSilver, and TimeTriggertoGold—are created in the local VS Code environment, with each function acting as a pipeline responsible for the data ingestion process into the bronze, silver, and gold layers respectively, corresponding to each listed function.

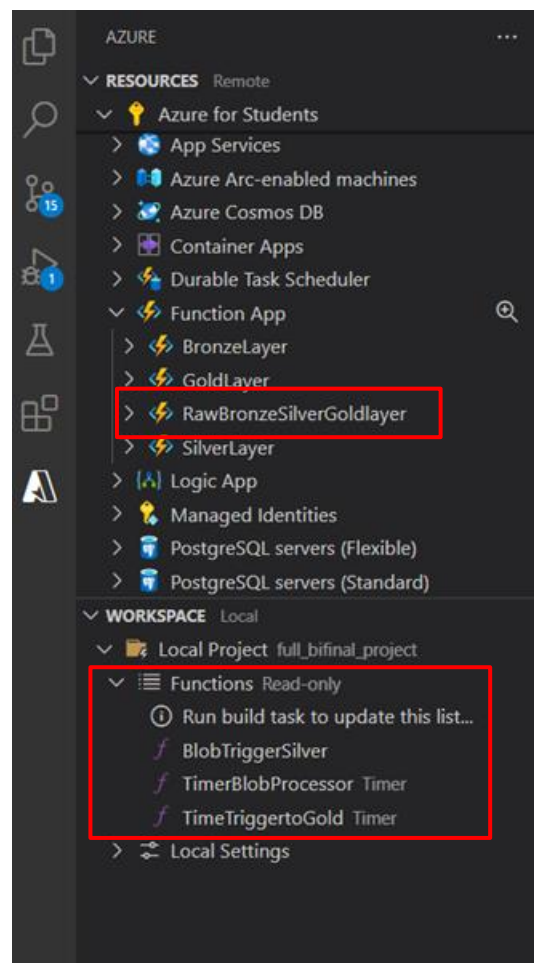
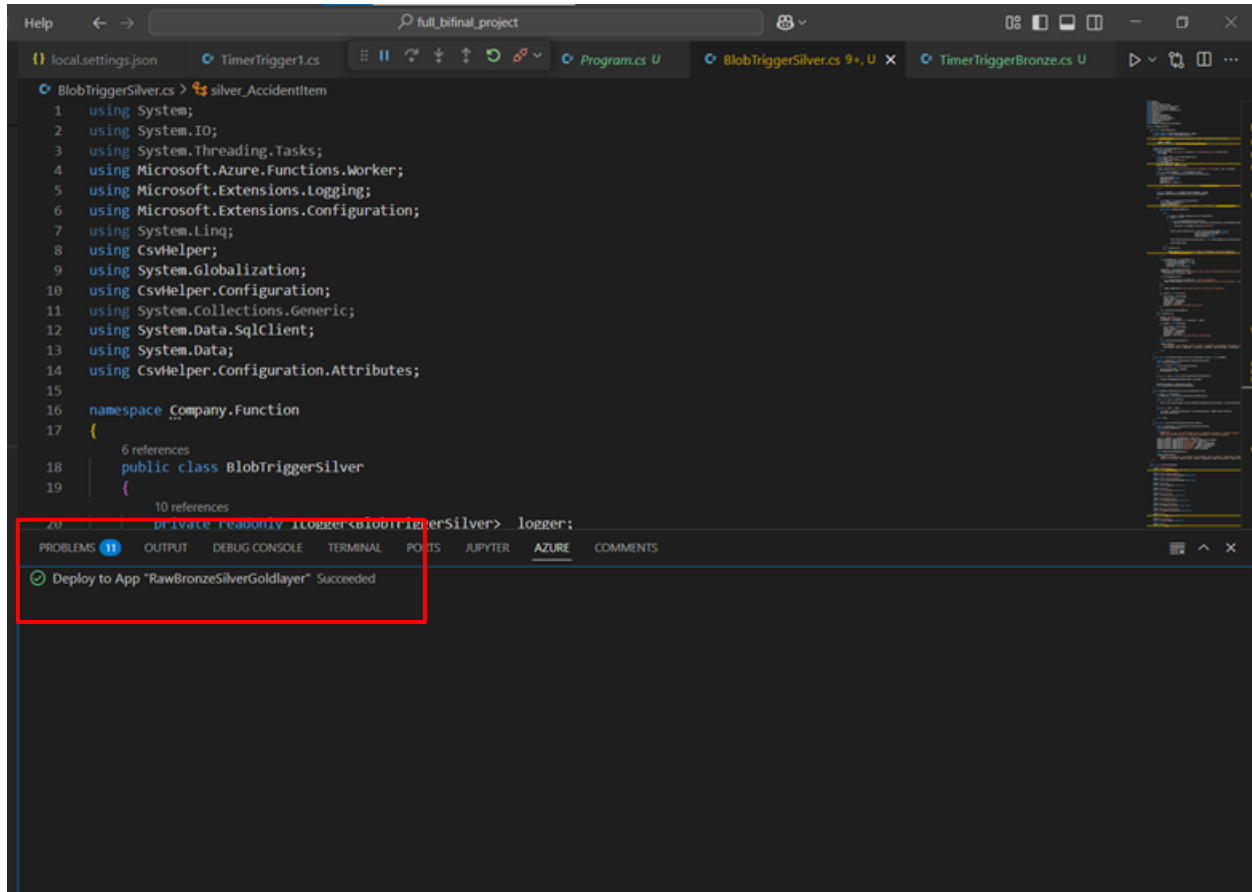


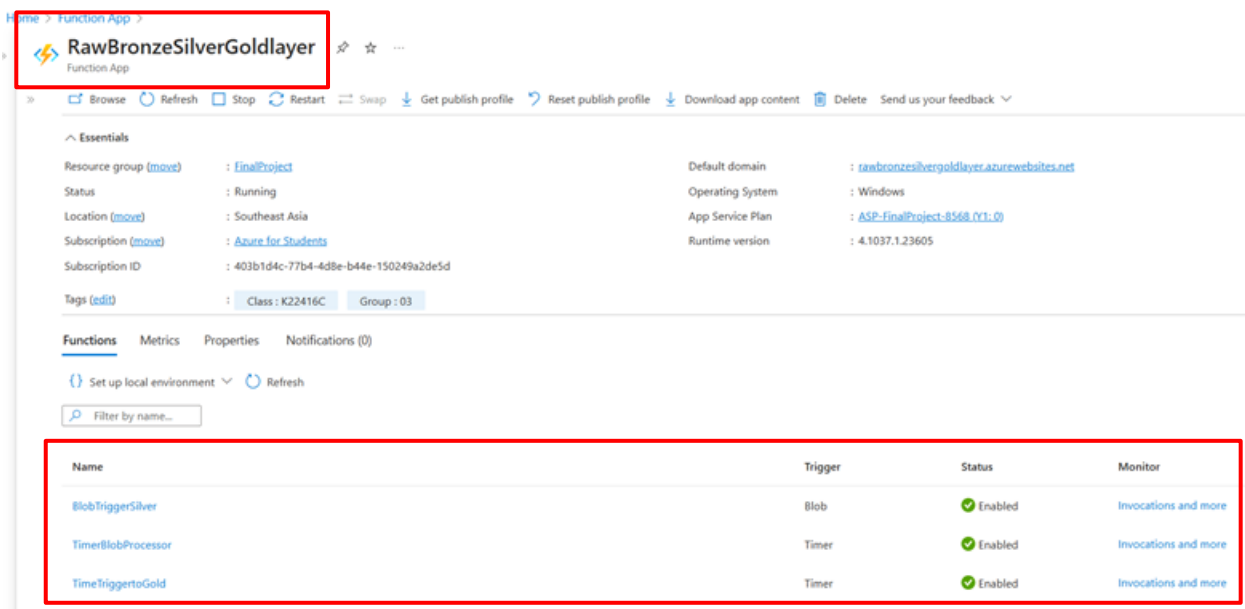
Figure 4.4. Project structure in the VS Code environment

When successfully executed in the local environment, the three functions responsible for each layer will be deployed to the Azure Portal via the initially created Function App, RawBronzeSilverGoldlayer.



*Figure 4.5. Result of deploying functions to Azure Portal*

The displayed result shows that the three functions created in the VS Code environment have been successfully deployed to the production environment with an enabled status. The ETL process on the cloud computing platform is carried out automatically thanks to the characteristics established for each function.

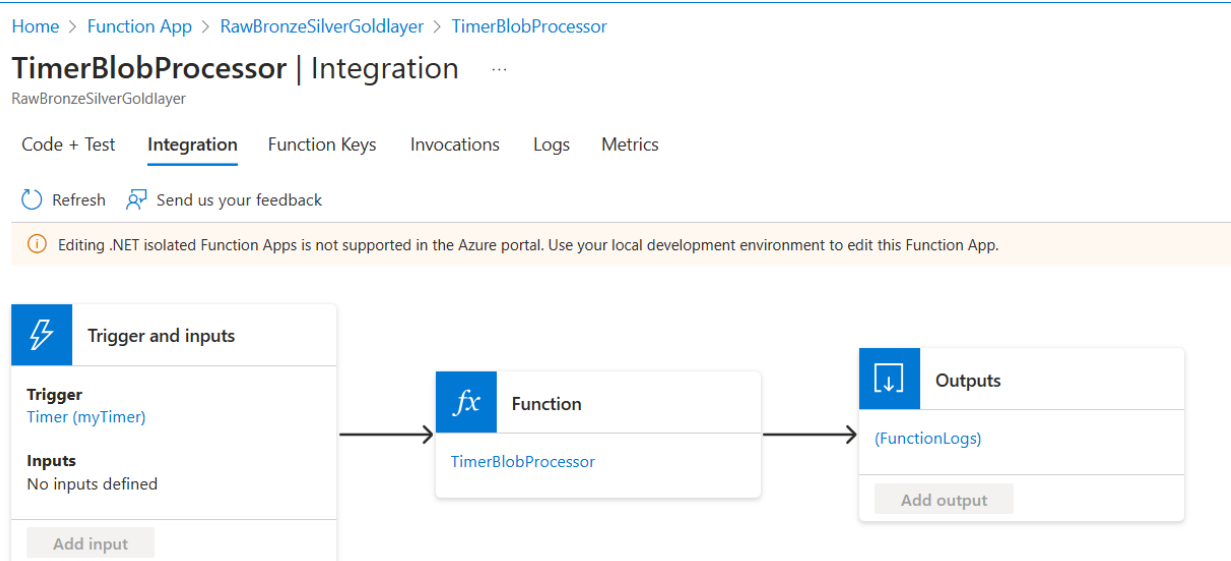


*Figure 4.6. List of functions in VS Code deployed via the Function App RawBronzeSilverGoldlayer*

Next, the author group will delve into details and specifically explain each function corresponding to the data ingestion process for each layer in the Azure SQL database.

## 4.2. Bronze layer data ingestion

### 4.2.1. Raw-to-bronze data pipeline design



*Figure 4.7. Raw to Bronze Pipeline Design*

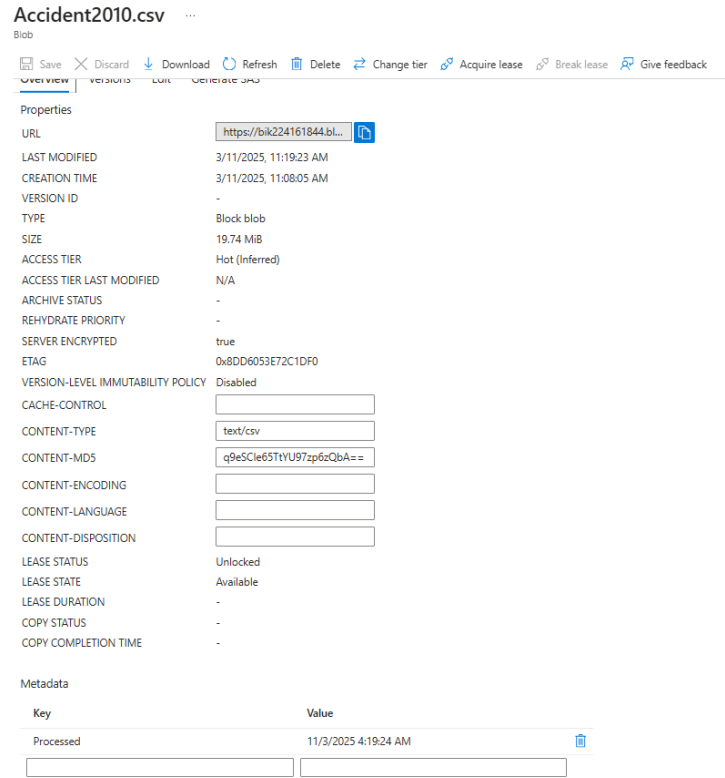
After the data has been periodically loaded into the Azure Storage Account container of employees, the ingestion process from raw to bronze officially begins. The raw-to-bronze pipeline is an automated data processing workflow deployed on Azure Functions to load and process data from the Azure Storage Account into the Bronze layer of the data lake architecture.

Built using the C# programming language, the authors have designed the pipeline to process CSV files containing yearly data from 2010 to 2015 and store them in the cloud-based SQL Azure Database. The main objective is to ensure that data is loaded automatically and accurately, providing a raw database foundation for subsequent processing steps. The specific structure of the pipeline is as follows:

- The bidssfinal container in the Azure Storage Account serves as the input data source in this pipeline, storing traffic accident data files processed annually from 2010 to 2015, corresponding to six input data files. By applying Blob processing techniques to prevent duplicate processing and ensure consistency in the workflow, each blob, once read and processed, is marked with a key-value pair attached as Metadata, where:

- Key: "Processed".
- Value: The current timestamp when the blob is processed, formatted as a string from `DateTime.UtcNow`.

Thus, in the next execution, if a blob already has the "Processed" metadata, the pipeline will automatically skip it and move on to another blob. This optimization enhances performance and ensures that data is not processed multiple times, playing a crucial role in maintaining the consistency of the Bronze layer.



*Figure 4.8. Metadata “Processed”*

Next, the authors utilize the CsvHelper library, a powerful and flexible tool in .NET, to read and map CSV data. Specifically, the process begins by extracting the CSV file content from the blob stream. Each data row is then converted into an `AccidentItem` object, representing a traffic accident record. Additionally, mapping definitions between CSV headers and `AccidentItem` properties are established, supporting various data types (string, float, integer) through the `AccidentItemMap` attribute. This combination enables flexible error handling, ensuring that the pipeline operates without interruptions.

To insert data from the `AccidentItem` object list into the newly created SQL table, the authors employ `SqlBulkCopy`, an optimized method in .NET for handling large-scale data processing. Before using `SqlBulkCopy` to insert data into SQL, the list of `AccidentItem` objects must be converted into a `DataTable` format, as `SqlBulkCopy` requires tabular structured input data. The key parameters of `SqlBulkCopy` are configured as follows:

- `BatchSize` is set to 10,000 records per batch. This value is chosen to balance processing performance and system resource consumption. With each batch of

10,000 records, the pipeline can efficiently load data without overwhelming the database.

- Timeout, or the maximum wait time for each insertion operation, is set to 300 seconds (5 minutes). This duration is sufficient to handle large data batches, especially as the number of records in the CSV files increases, while ensuring that the pipeline does not hang indefinitely in case of issues.
- Retry mechanism enhances reliability by implementing a retry strategy in case of a timeout error. Specifically, if an insertion fails due to a timeout, the pipeline will wait 5 seconds before retrying, with a maximum of three attempts. This approach minimizes the risk of database disruptions, such as unstable connections. If all three retries fail, the error is logged, and a notification is sent for manual intervention.

Additionally, in this pipeline, the authors adopt the full-load technique when loading records into the database. Specifically, if a table already exists, the pipeline deletes the old table and recreates a new one within the bronze schema. Generally, performing a full load may introduce technical and system resource challenges. However, this pipeline optimizes both technical efficiency and data loading time by leveraging SqlBulkCopy, significantly reducing the time required to load data into the bronze layer. Under stable connection conditions, loading a dataset into the database takes only one minute. This demonstrates that the full-load approach in this pipeline remains optimal while saving time and resources for completing other steps in the project.

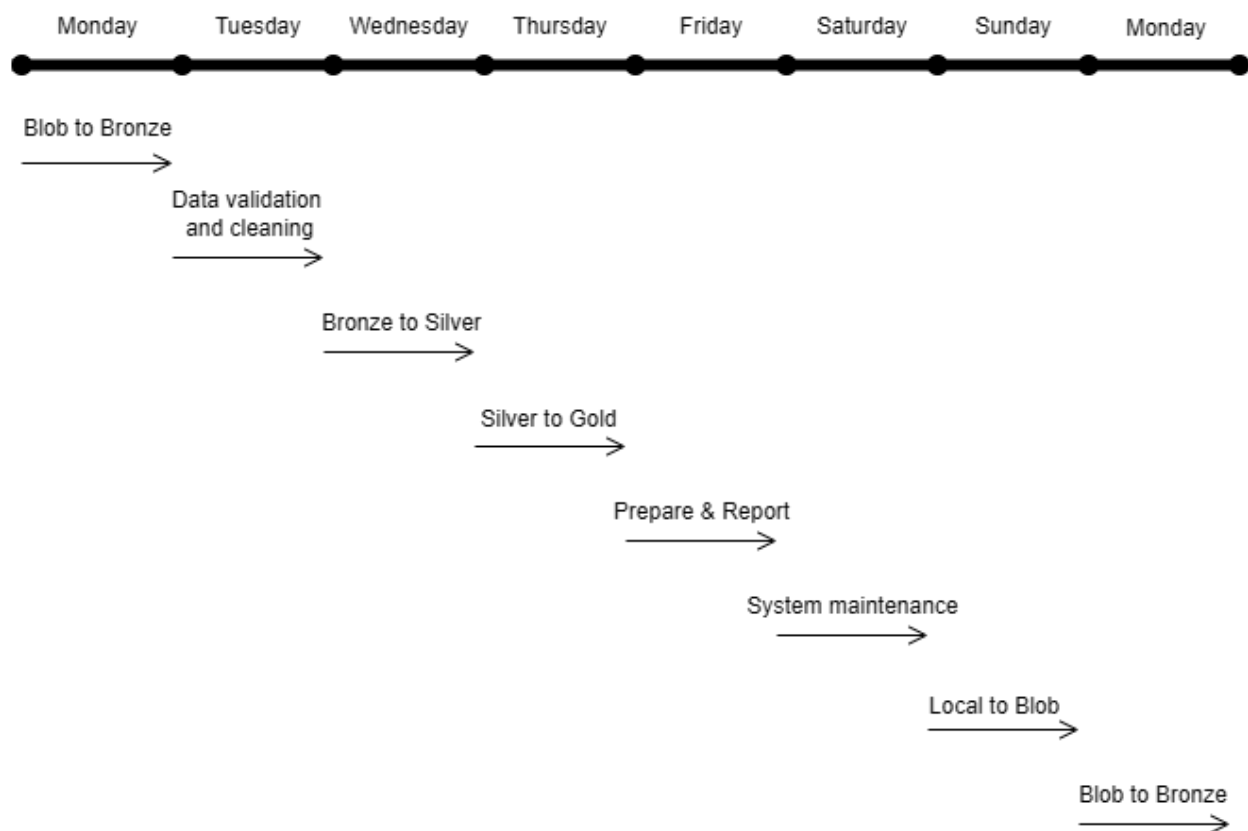
#### **4.2.2. Schedule data updates**

At this point, the processing flow from raw data sources to the database has been completed. The next challenge to address is ensuring real-time processing and automation in a context where data is abundant and continuously updated. These are among the core requirements in pipeline development.

To meet these demands, the authors leverage the Timer Trigger mechanism to automate the periodic data processing from Azure Blob Storage and store it in the Bronze layer of the data lake. Timer Trigger is a key component of Azure Functions, enabling functions to execute on a scheduled basis using CRON expressions or fixed time intervals.

To determine the most suitable schedule for configuring the Timer Trigger function for optimal efficiency, the authors analyzed the following factors:

- Business requirements and data characteristics: Traffic accident data for different regions in the UK is updated frequently, allowing stakeholders to track accident trends and severity levels in each area.
- System performance evaluation: By assessing system performance and fine-tuning execution timing across the entire pipeline, the authors determined an optimal schedule for the Timer Trigger to run once a week on Mondays. The specific schedule is as follows:



*Figure 4.9. Timer Trigger setting*

After determining an appropriate execution schedule, the Timer Trigger is configured as follows:

- Cron expression configuration: "0 0 1 \* \* 1" corresponds to execution at 1:00 AM every Monday.

- TimerInfo parameter setup: The TimerInfo myTimer parameter is used to log the activation time (DateTime.UtcNow), enabling execution history tracking and facilitating pipeline activity monitoring.

With this setup, the Timer Trigger allows the pipeline to run periodically without user intervention, ensuring that data from Azure Blob Storage is processed continuously and in a timely manner. This is particularly crucial for systems requiring frequent data updates, such as traffic accident analysis, which supports data-driven management decisions.

Although some limitations exist—such as execution time constraints and a fixed frequency—Timer Trigger remains a powerful solution due to its automation, reliability, and scalability in a serverless environment. Implementing enhancements, such as adjusting execution frequency and incorporating dynamic data handling mechanisms, can further optimize pipeline performance, making it better suited for real-world data processing requirements.

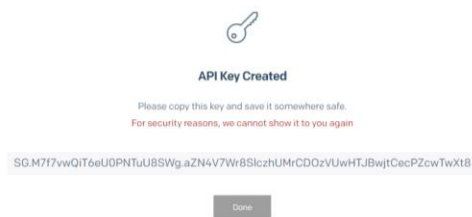
#### 4.2.3. Send an error report email

The pipeline is integrated with SendGrid, a cloud-based email delivery service, to send notifications via email in case of errors or upon successful completion of data processing. SendGrid serves as a monitoring tool, enabling administrators to efficiently track the pipeline's status.

SendGrid is integrated into the pipeline through the SendGrid library in the .NET environment, with its logic embedded in the Run function.

SendGrid Configuration:

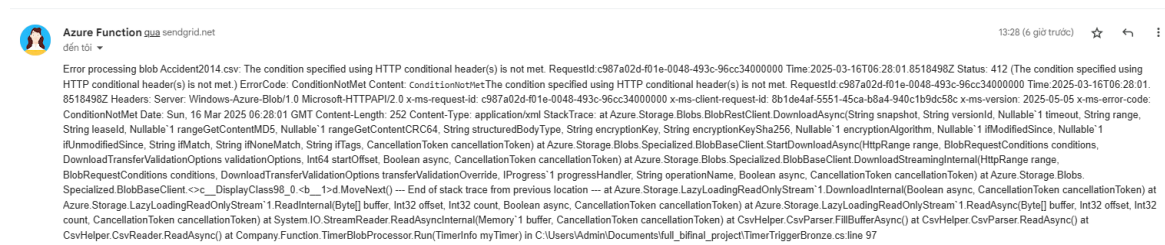
- Create an environment variable for SendGridApiKey, which is retrieved from the environment variables defined in the Application Settings of Azure Functions.



*Figure 4.10. Sendgrid API key*



- In case of an error (e.g., CSV parsing failure), an email is sent containing detailed error information:



*Figure 4.11. Sendgrid error*

- Upon successful pipeline execution, a confirmation email is sent with details on the number of records processed.



*Figure 4.12. Sendgrid loading successfully*

Despite limitations such as network connectivity issues and email quota restrictions, SendGrid remains a reliable and flexible solution. It helps ensure smooth pipeline operation and enables decision-makers to respond promptly to any issues that arise during pipeline execution.

### 4.3. Silver layer data ingestion

The data in Azure Blob Storage and the data in the bronze layer share similar characteristics and significance. Therefore, during the process of moving data into the silver layer, the source will directly point to Azure Blob Storage—where the raw data is stored. Based on the evaluations and EDA (Exploratory Data Analysis), during the ingestion of data into the silver layer, the authors perform several data transformation operations to ensure that the data in the silver layer is complete, consistent, properly structured, and clean. The Azure Function responsible for ingesting data into the silver layer is BlobTriggerSilver, and the data ingestion process into the silver layer will consist of three distinct and prominent segments.

Home > RawBronzeSilverGoldlayer > BlobTriggerSilver

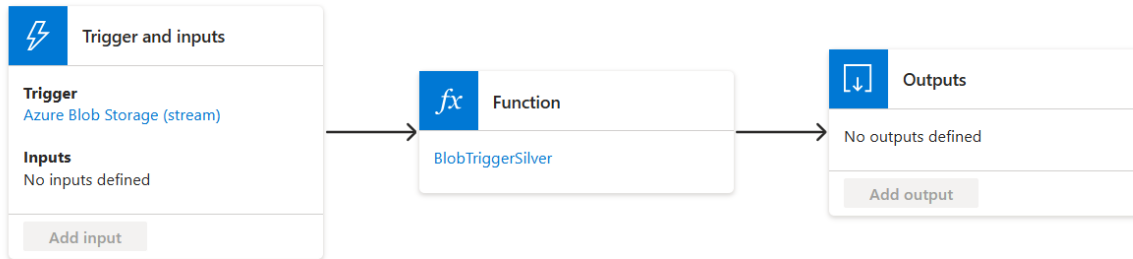
## BlobTriggerSilver | Integration ...

RawBronzeSilverGoldlayer

Code + Test **Integration** Function Keys Invocations Logs Metrics

Refresh Send us your feedback

Editing .NET isolated Function Apps is not supported in the Azure portal. Use your local development environment to edit this Function App.



*Figure 4.13. Design a pipeline to ingest data from the storage account into the silver layer.*

The Azure Blob Storage Trigger is designed to activate the BlobTriggerSilver function. The trigger detects new files uploaded to the container in Blob Storage. Each time a new file is uploaded, the BlobTriggerSilver function initiates the data ingestion process. Here, the trigger acts equivalently to an input: whenever a CSV file is uploaded to the “bidssfinal” container, it serves as the input for the function.

Once activated by the trigger, the BlobTriggerSilver function identifies the parameter findings—the CSV file name—and the connection, which is the connection string to Blob Storage. Instead of passing the entire content of the CSV file as a string or byte array, the BlobTriggerSilver function passes the data as a Stream object. Similar to the data ingestion process for the bronze layer, CsvReader is used to read data from the CSV file, which is then mapped to the predefined AccidentItem class.

Valid data is recorded into the silver.accident1015 table within the silver layer. Unlike the storage method in the bronze layer, data in the silver layer is ingested into a single table using BulkInsertToSql. This is a method for bulk inserting data into an Azure SQL database. A connection is established between SqlBulkCopy and the silver.accident1015 table, followed by mapping columns from the AccidentItem class to the target table. This

produces a list of records containing all objects from AccidentItem after being read from the CSV. Finally, this list of records is converted into a DataTable. Converting it into a DataTable allows SQL to recognize the format and successfully copy the data into the Azure SQL database.

>>

Query 1 × Query 2 ×

Run Cancel query Save query Export data as Show all Open Copilot

Results Messages

Search to filter items...

Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force
200501BS00001	525680	178240	-0.19117	51.489096	1
200501BS00002	524170	181650	-0.211708	51.520075	1
200501BS00003	524520	182240	-0.206458	51.525301	1
200501BS00004	526900	177530	-0.173862	51.482442	1
200501BS00005	528060	179040	-0.156618	51.495752	1
200501BS00006	524770	181160	-0.203238	51.51554	1
200501BS00007	524220	180830	-0.211277	51.512695	1
200501BS00009	525890	179710	-0.187623	51.50226	1
200501BS00010	527350	177650	-0.167342	51.48342	1
200501BS00011	524550	180810	-0.206531	51.512443	1
200501BS00012	526240	178900	-0.182872	51.494902	1
200501BS00014	526170	177690	-0.184312	51.484044	1

*Figure 4.14. The data in the silver layer within the SQL database*

The data, after being cleaned and transformed, has been recorded in the SQL database. Compared to the original data, the data in the silver layer has undergone certain changes. Specifically:

- The "Date" column has been standardized to the format yyyy-MM-dd.
- Two new columns have been added: "LSOA\_of\_Accident\_Location\_missing" to flag missing values in the LSOA\_of\_Accident\_Location column, and "Location\_Data\_Missing" to indicate if any of the four location-related columns—Location\_Easting\_OSGR, Location\_Northing\_OSGR, Longitude, or Latitude—contain null values.
- Rows where all four columns—Location\_Easting\_OSGR, Location\_Northing\_OSGR, Longitude, and Latitude—are simultaneously missing have been removed. These are identified as accidents with undetermined locations,

which are deemed meaningless for long-term analysis, prompting the team to eliminate them.

## 4.4. Gold layer data ingestion

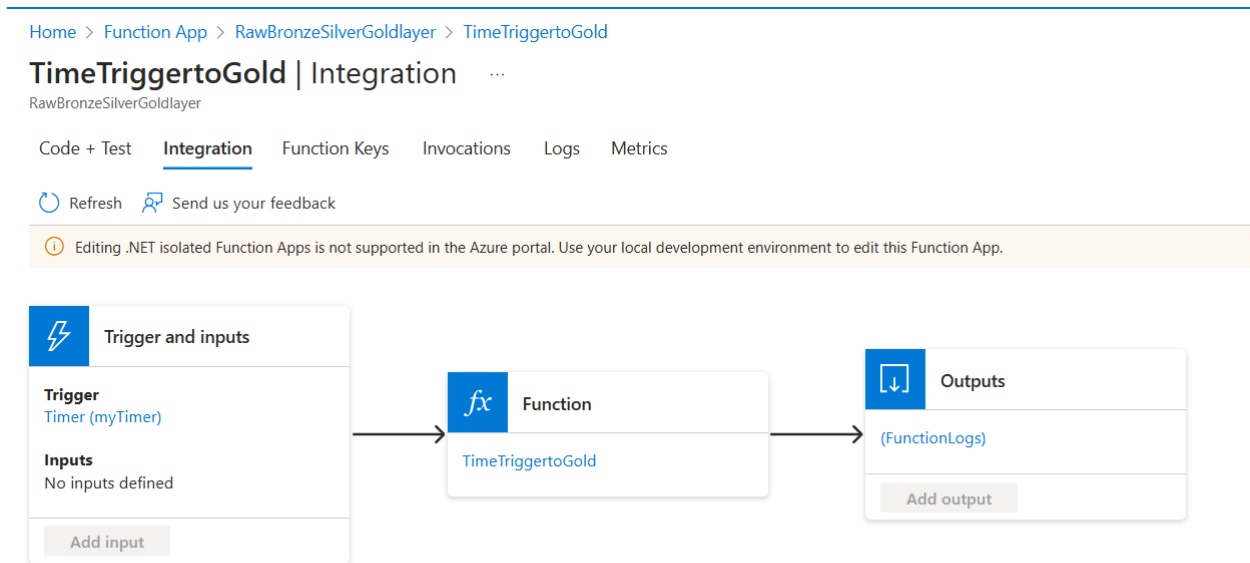


Figure 4.15. Design a pipeline to ingest data from the silver layer into gold layer

The process of ingesting data from the Silver Layer to the Gold Layer plays a crucial role in standardizing and optimizing the data for analysis and reporting. The data is structured using the Star Schema model, which enhances query performance and ensures that the data is available for reporting systems. This process frequently involves the use of Azure Functions, particularly with triggers like Timer (myTimer), which automatically activate processes such as data transformation and loading at defined intervals. The outputs of these functions, including logs and results, are efficiently captured, offering valuable insights for continuous monitoring and validation of the data pipeline.

In this case, the data is extracted from the accident1015 table in the Silver Layer and categorized into Dimension tables, which store descriptive information, as well as a Fact table to store actual event data in the Gold Layer. A key aspect of this process is the use of the Slowly Changing Dimension Type 2 (SCD Type 2) method, which preserves historical changes in the data. Instead of directly updating existing records, the system retains

previous data by setting the End\_Date and marking the record as inactive (Status = 0). Simultaneously, a new record is inserted with a new Start\_Date, an active status (Status = 1), and the End\_Date set to NULL, indicating that this is the current valid record. This approach allows the system to track changes in factors like lighting conditions, weather, road type, or accident severity within the Dimension tables while maintaining the integrity of historical data.

After updating the Dimension tables, the system removes outdated records from the Fact table to ensure that only the most current data is stored. The process of loading data into the Fact\_Accidents table uses foreign keys from the Dimension tables, ensuring that each accident event contains all the necessary information. To automate the ingestion process from the Silver Layer to the Gold Layer, the system utilizes an Azure Function with a Time Trigger.

The Time Trigger in Azure Functions is configured using a CRON expression, providing flexibility in defining execution frequency. This trigger activates based on a predefined schedule, checking for updates in the Silver Layer and transforming the data into the Gold Layer. When the trigger is activated, the Azure Function, named TimeTriggertoGold, connects to the database through a secure connection string in Azure and executes the ETL (Extract, Transform, Load) operations, which include:

- Extracting data from the Silver Layer.
- Checking and updating Dimension tables following the SCD Type 2 principle.
- Removing old data and loading new data into the Fact\_Accidents table.

To enhance performance, this function utilizes bulk insert operations when updating tables, reducing processing time for large datasets. Additionally, the system integrates error logging and monitoring, allowing tracking of any failures occurring during execution.

### **Results after running the system**

After executing the ingestion process from the Silver Layer to the Gold Layer, the system has successfully transformed and structured the data according to the Star Schema model. The Dimension tables have been updated using SCD Type 2, ensuring that any changes in accident-related attributes are recorded while preserving historical data. The

Fact\_Accidents table now contains the most recent accident records, linked with the latest dimension data to support efficient querying and analysis.

### **Updated Dimension Tables**

Each Dimension table has been updated to reflect the latest changes while maintaining historical records:

- Dim\_AccidentSeverity: Stores accident severity categories, ensuring past classifications are preserved while tracking new changes.
- Dim\_LightConditions: Records variations in lighting conditions at accident locations over time.
- Dim\_WeatherConditions: Maintains historical weather conditions associated with accidents.
- Dim\_RoadType: Tracks changes in road classification, supporting traffic pattern analysis.
- Dim\_RoadSurfaceConditions: Stores surface conditions of accident sites, useful for infrastructure planning.
- Dim\_Police: Logs the police force units that responded to accidents.
- Dim\_UrbanorRuralArea: Differentiates accidents occurring in urban versus rural settings.

These tables enable detailed trend analysis, allowing authorities to monitor how accident conditions evolve and identify potential risk factors.

### **Fact\_Accidents Table**

The Fact\_Accidents table has been refreshed to include:

- The latest accident data, ensuring reports and dashboards reflect up-to-date statistics.
- Accurate references to Dimension tables, preserving the integrity of relationships within the Star Schema.
- A structured format optimized for Power BI visualization, enhancing analytical efficiency.

## Dim\_AccidentSeverity

Results Messages

🔍 Search to filter items...					
AccidentSeverityKey	Accident_Severity	Description	Start_Date	End_Date	Status
1	1	Fatal	2025-03-12		True
2	3	Slight	2025-03-12		True
3	2	Serious	2025-03-12		True

Figure 4.16. Dim\_AccidentSeverity

## Dim\_LightConditions

Results Messages

LightConditionsKey	Light_Conditions	Description	Start_Date	End_Date	Status
1	1	Daylights	2025-03-12		True
2	4	Darkness with street li...	2025-03-12		True
3	5	Darkness without stre...	2025-03-12		True
4	6	Darkness with no ligh...	2025-03-12		True
5	7	Darkness with unkno...	2025-03-12		True

Figure 4.17. Dim\_LightConditions

## Dim\_Police

Results Messages

PoliceForceKey	Police_Force	Start_Date	End_Date	Status
1	33	2025-03-12		True
2	43	2025-03-12		True
3	93	2025-03-12		True
4	36	2025-03-12		True
5	53	2025-03-12		True

Figure 4.18. Dim\_Police

## Dim\_RoadSurfaceConditions

Results		Messages			
RoadSurfaceConditio...	Road_Surface_Condit...	Description	Start_Date	End_Date	Status
1	1	Dry	2025-03-12		True
2	4	Ice	2025-03-12		True
3	3	Snow	2025-03-12		True
4	5	Flooded	2025-03-12		True
5	-1	Not recorded	2025-03-12		True

Figure 4.19. Dim\_RoadSurfaceConditions

## Dim\_RoadType

Results		Messages			
RoadTypeKey	Road_Type	Description	Start_Date	End_Date	Status
1	1	Single carriageway	2025-03-12		True
2	3	Other classified road t...	2025-03-12		True
3	9	Unspecified/other	2025-03-12		True
4	2	Dual carriageway	2025-03-12		True
5	6	One-way street	2025-03-12		True

Figure 4.20. Dim\_RoadType

## Dim\_UrbanorRuralArea

ResultsMessages

Search to filter items...

UrbanRuralAreaKey	Urban_or_Rural_Area	Start_Date	End_Date	Status
1	1	2025-03-12		True
2	2	2025-03-12		True

Figure 4.21. Dim\_UrbanorRuralArea

## Dim\_WeatherConditions

ResultsMessages

Search to filter items...

WeatherConditionsK...	Weather_Conditions	Description	Start_Date	End_Date	Status
1	1	Fine (no high winds)	2025-03-12		True
2	3	Rain (no high winds)	2025-03-12		True
3	4	Rain with high winds	2025-03-12		True
4	5	Snow (no high winds)	2025-03-12		True

Figure 4.22. Dim\_WeatherConditions



## Fact\_Accidents

Results

Messages

Search to filter items...

Accident_Index	PoliceForceKey	LightConditionsKey	RoadTypeKey	AccidentSeverityKey	RoadSurfaceConditionsKey	WeatherConditionsKey	UrbanRuralAreaKey
201001BS70003	12	1	5	2	6	8	1
201001BS70004	12	2	5	2	6	7	1
201001BS70006	12	1	5	2	1	1	1
201001BS70007	12	2	1	2	1	1	1

Results

Messages

Longitude	Latitude	Local_Authority_District	Local_Authority_Highway	Date	Time	Number_of_Vehicles	Number_of_Casualties	Speed_Limit
-0.164002	51.484087	12	E09000020	2010-01-11	07:30	2	1	30
-0.195273	51.509212	12	E09000020	2010-01-11	18:35	1	1	30
-0.20311	51.507804	12	E09000020	2010-01-12	10:22	2	1	30
-0.198858	51.513314	12	E09000020	2010-01-02	21:21	2	1	30

Figure 4.23. Fact\_Accidents

### 4.5. Data Governance

The logging system in all three Azure Functions is designed to track the entire execution process, from start to finish, including both successful and failed cases. All three functions utilize a basic FunctionLog structure, but the implementation and log details vary depending on the purpose of each function.

- Real-time logging: The ILogger is used to record logs instantly (info, warning, error) during processing, enabling progress tracking and immediate issue detection.
- Structured log storage: FunctionLog entries are created and stored in the `function\_logs` table in the Azure SQL database, ensuring long-term retention of activity history for later analysis.
- Error handling: When exceptions occur, the log captures detailed error information.
- Log formatting and return: Logs are serialized in JSON format and returned in the function's output, facilitating integration with other systems if needed.

The logging system in the three Azure Functions—BlobTriggerSilver, TimeTriggertoGold, and TimerBlobProcessor—employs the FunctionLog class with a consistent structure, including columns such as FunctionName, Status, TriggeredBy, RecordCount, Timestamp, and Message, ensuring uniformly formatted log information. The ILogger feature logs in real time with different levels, such as LogInformation for general information,

LogWarning for warnings, and LogError for errors, allowing for immediate progress monitoring and issue detection. All logs from the three functions are recorded in the function\_logs table.

A key difference lies in the logging approach: BlobTriggerSilver records logs by directly inserting them via InsertFunctionLog, whereas TimeTriggertoGold and TimerBlobProcessor perform indirect insertion through SqlOutput. Consequently, BlobTriggerSilver does not return an output, while TimeTriggertoGold and TimerBlobProcessor return outputs in the form of function logs via gold\_OutputType and OutputType, respectively. Effective log management across all three functions includes detailed error-handling mechanisms, capturing exception details and failure statuses in the logs, which supports efficient troubleshooting and analysis. These features establish a robust and consistent logging foundation throughout the data processing stages.

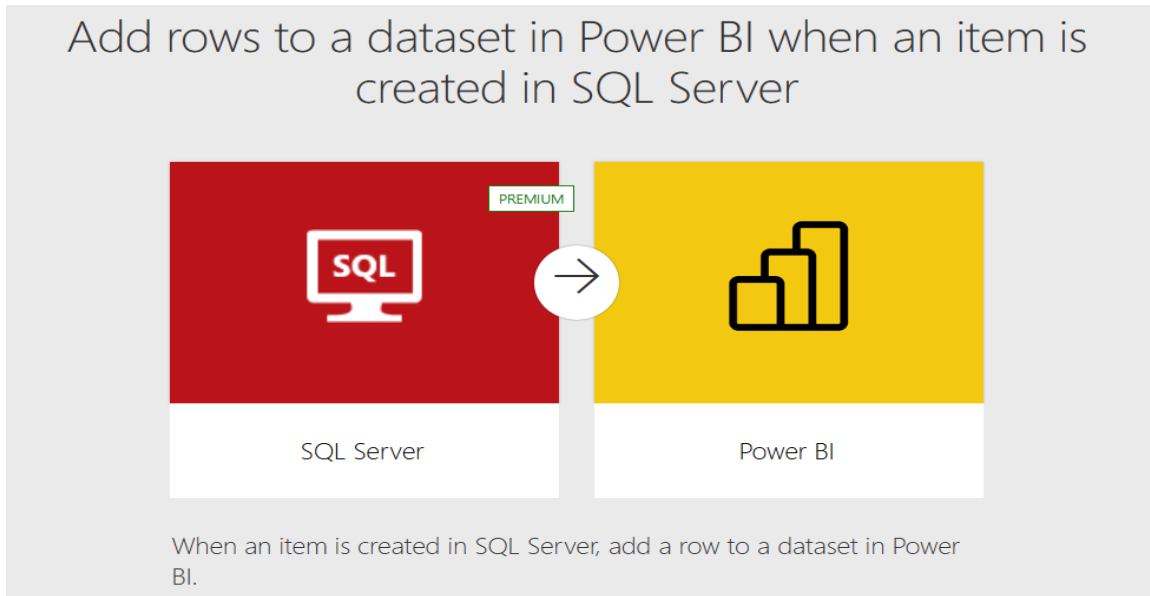
Id	FunctionName	Status	TriggeredBy	RecordCount	Timestamp	Message
340	TimerBlobProcessor	Failed	TimerTrigger	0	2025-03-16T05:52:04.0000000	Error processing blob Accident2012.csv: The condition specified using HTTP conditional header(s) is r
341	TimerBlobProcessor	Failed	TimerTrigger	0	2025-03-16T05:52:02.0000000	Error processing blob Accident2009.csv: The condition specified using HTTP conditional header(s) is r
342	BlobTriggerSilver	Failed	BlobTrigger	145571	2025-03-16T05:52:00.7330000	Error processing blob Accident2012.csv: Violation of PRIMARY KEY constraint 'PK_accident1015': Can
339	TimerBlobProcessor	Success	TimerTrigger	0	2025-03-16T05:52:00.0000000	Processed 0 blobs successfully
338	BlobTriggerSilver	Failed	BlobTrigger	145571	2025-03-16T05:51:53.5430000	Error processing blob Accident2012.csv: Violation of PRIMARY KEY constraint 'PK_accident1015': Can
337	BlobTriggerSilver	Failed	BlobTrigger	145571	2025-03-16T05:51:38.9630000	Error processing blob Accident2012.csv: Violation of PRIMARY KEY constraint 'PK_accident1015': Can
336	BlobTriggerSilver	Success	BlobTrigger	163554	2025-03-16T05:51:28.2700000	Processed blob Accident2009.csv successfully
335	TimerBlobProcessor	Success	TimerTrigger	0	2025-03-16T05:50:00.0000000	Processed 0 blobs successfully
334	TimeTriggertoGold	Success	TimerTrigger	1075136	2025-03-16T05:48:00.0000000	Processed 1075136 records successfully
332	TimerBlobProcessor	Success	TimerTrigger	0	2025-03-16T05:48:00.0000000	Processed 0 blobs successfully
333	TimeTriggertoGold	Success	TimerTrigger	1075136	2025-03-16T05:47:33.0000000	Processed 1075136 records successfully
331	TimerBlobProcessor	Success	TimerTrigger	0	2025-03-16T05:47:14.0000000	Processed 0 blobs successfully

Figure 4.24. Table function\_logs recording logs in SQL database

The recorded logs include both successful and failed cases, along with related information such as the number of records, execution time, and returned messages from all three functions, stored in the `function\_logs` table in the SQL database.

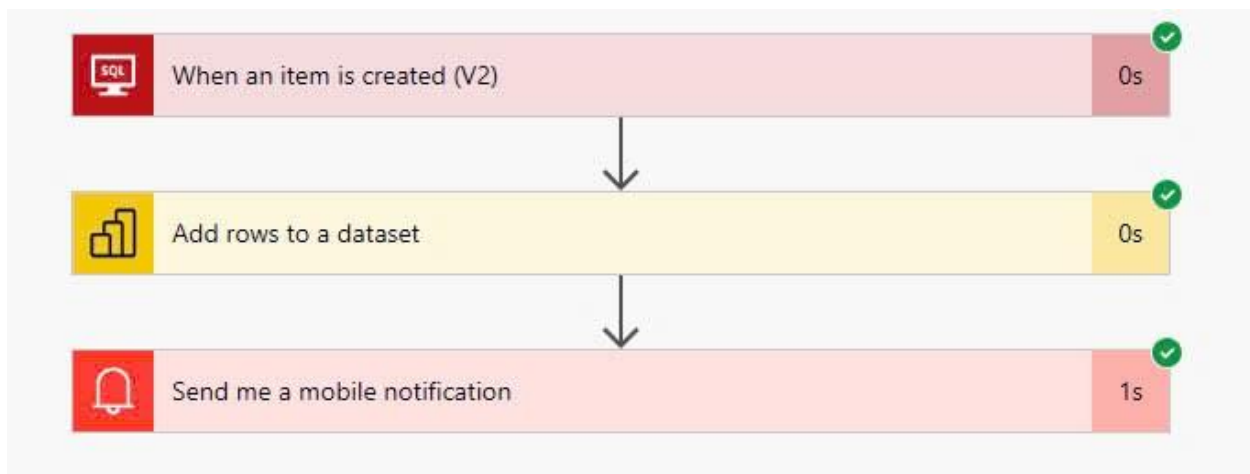
#### 4.6. Load data using Power Automate

The authors utilize a pre-built template provided by Power Automate. This template supports adding rows to the dataset used in Power BI whenever a new entry is created in SQL Server.



*Figure 4.25. Add rows to a dataset in PowerBI when an item is created in SQL Server*

In this setup:



*Figure 3. 1: Power Automate flow*

- The "When an item is created in SQL Server" component connects directly to the data lake and links to tables in the Gold layer, which includes Dim and Fact tables. Whenever a new record is detected in the Gold layer, the pipeline is triggered.
- The "Add rows to a dataset in Power BI" component directly connects to the dataset used for dashboard visualization in Power BI. When a change is detected in the SQL Server data source, this component is activated by mapping the corresponding columns from the SQL Database to the Power BI dataset.



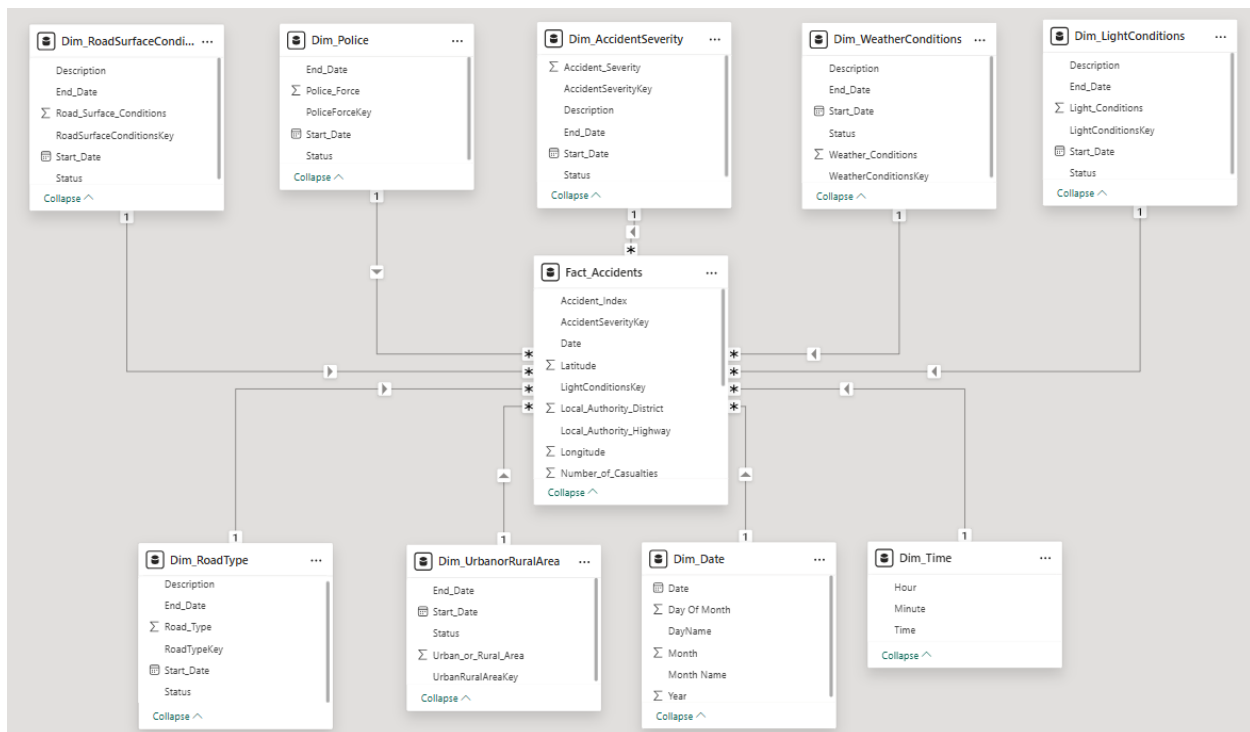
*Figure 4.26. Upload Dataset Successfully in Power Automate platform*

- Finally, a mobile notification feature is integrated to provide real-time alerts whenever the pipeline runs, ensuring prompt awareness of updates or potential issues.

## CHAPTER 5. DATA VISUALIZATION

*This chapter focuses on data visualization using Power BI, transforming processed traffic accident data into interactive dashboards. It outlines the data modeling approach, dashboard design strategy, and key visualizations, including trends in accidents, severity distribution, high-risk locations, and contributing factors such as weather and road conditions. By delivering clear and actionable insights, these visualizations assist stakeholders in analyzing patterns, identifying risks, and making data-driven decisions to enhance road safety in the UK.*

### 5.1. Data Modeling



*Figure 5.1. Data model*

Based on the information from section 3.3.5 on Gold layer data ingestion, which defines the relationship between dimensional and fact tables, the data modeling in Power BI has been identified and structured accordingly. Using a star schema architecture, the Fact-Accidents table serves as the central table storing events and linking to dimension tables. All dimension tables have a one-to-many relationship with the fact table. In addition, our team also created Dim\_Date and Dim\_Time by M language in Power Query tables to

enhance time-based analysis and visualization. Dim\_Date includes time - related data such as Date, Day of Month, DateName, Month, MonthName, Year, covering the period from January 1 to December 31, 2015. Dim\_Time provides more granular time details, including hours, minutes, and seconds within a day, enabling precise tracking of accident occurrences throughout the day.

## 5.2. Dashboard Strategy

*Table 5.1. Dashboard Strategy*

Dashboard	Requirements	Business questions and related chart
Accident overview	<ol style="list-style-type: none"> <li>1. Analyze accident trends based on time, location, and severity.</li> <li>2. Deliver key metrics to assist stakeholders in monitoring accident patterns.</li> </ol>	<ol style="list-style-type: none"> <li>1. Is the total number of accidents increasing or decreasing over time? → Total Accidents Trend (Line Chart)</li> <li>2. What is the distribution of accidents by severity (slight, serious, fatal)? → Total Accidents by Severity Type (Bar Chart)</li> <li>3. How many casualties occur at each severity level? → Number of Casualties by Severity Type (Donut Chart/Pie Chart)</li> <li>4. How does speed limit correlate with accident severity? → Speed Limit by Severity Type (Bar Chart)</li> <li>5. Which road types have the highest number of accidents? → Total Accidents by Road Type (Bar</li> </ol>

		Chart)
Uncontrollable Causes	<ol style="list-style-type: none"> <li>1. Evaluate how location (urban vs. rural) affects accident occurrences and casualty rates.</li> <li>2. Examine the impact of weather and road surface conditions on accident frequency and severity.</li> </ol>	<ol style="list-style-type: none"> <li>1. Which areas have the highest accident rates? → Total Accidents by Location (Map)</li> <li>2. How do casualty numbers compare between urban and rural areas? → Number of Casualties by Location (Donut Chart)</li> <li>3. What weather conditions contribute to the highest number of accidents? → Top 5 Weather Conditions with the Most Accidents (Bar Chart)</li> <li>4. How do different road surface conditions impact accident frequency? → Top 3 Road Surface Conditions with the Most Accidents (Bar Chart)</li> <li>5. Which road surface conditions are associated with the most severe accidents? → Road Surface Condition Table (Table)</li> </ol>
Controllable Causes	<ol style="list-style-type: none"> <li>1. Analyze the influence of light conditions on accident rates and</li> </ol>	<ol style="list-style-type: none"> <li>1. Which police forces handle the most accidents? → Top 10 Police Forces with the Most Accidents (Table)</li> </ol>






















	<p>casualty numbers.</p> <p>2. Assess how different road types affect accident frequency and severity.</p> <p>3. Identify the police forces managing the highest number of accidents to enhance resource allocation.</p>	<p>2. How do light conditions impact the number of accidents? → Total Accidents by Light Conditions (Bar Chart)</p> <p>3. Which light conditions are associated with the highest number of casualties? → Number of Casualties by Light Conditions (Donut Chart)</p> <p>4. Which road types have the highest number of accidents? → Total Accidents by Road Type (Bar Chart)</p> <p>5. Which road types are associated with the most casualties? → Number of Casualties by Road Type (Bar Chart)</p>
--	--	---

### 5.3. Dashboard Visualization

#### 5.3.1. KPIs

*Table 5.2. KPIs trend through time*



Year	KPIs			
All	 876470	 18194	 38.48	SAR Rate 15%
2010	 154414	 3256	 38.88	SAR Rate 14%
2011	 151474	 3314	 38.54	SAR Rate 15%
2012	 145571	 2938	 38.50	SAR Rate 15%
2013	 138660	 2948	 38.53	SAR Rate 15%
2014	 146322	 2898	 38.24	SAR Rate 15%
2015	 140029	 2840	 38.18	SAR Rate 15%

Between 2010 and 2015, a total of 876,470 accidents occurred, including 18,194 fatal cases. The average speed of accident-causing drivers was 38 km/h, indicating that excessive speed was not a major factor. Serious accidents accounted for 15% of the total. In 2010, both total accidents and fatalities reached their peak, whereas 2013 recorded the lowest figures. However, the overall trend shows a steady decline in accident numbers, fatalities, and vehicle speed, suggesting that government initiatives have been effective in enhancing traffic safety. Despite this progress, the proportion of serious accidents remained unchanged, highlighting the need for more comprehensive policies and stricter enforcement to further improve road safety.

### 5.3.2. Page 1 - Accident Overview

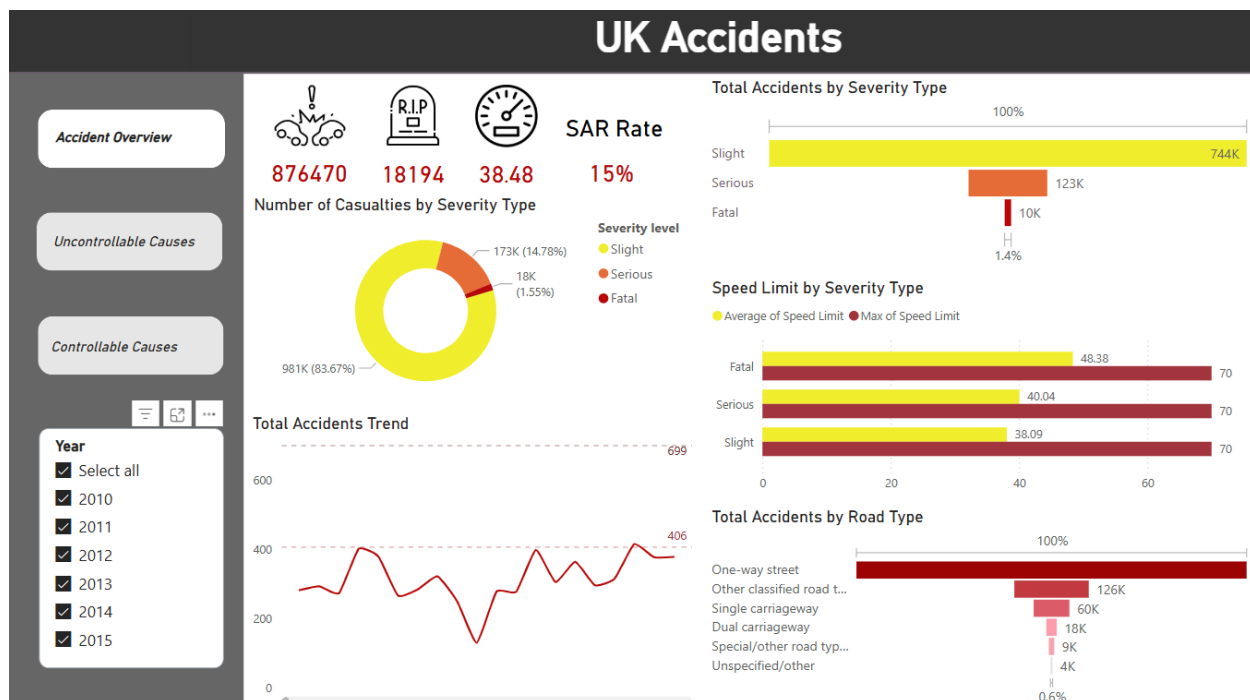


Figure 5.2. Accident Overview Dashboard

The Dashboard Overview provides a comprehensive visualization of road accidents in the UK, highlighting key contributing factors. Specifically, the analysis categorizes accident causes into two main groups: Uncontrollable and Controllable factors.

#### a) Is the total number of accidents increasing or decreasing over time?

The following line charts illustrate the trend in accident numbers over time. All charts

indicate that the number of accidents fluctuates throughout the year.

- *Total Accident Trend (2010 - 2015)*

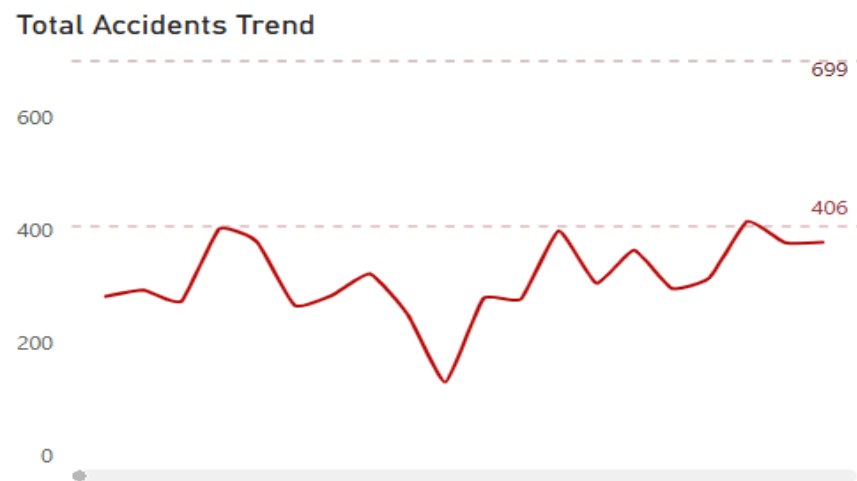


Figure 5.3. Total Accident Trend (2010 - 2015)

The following line charts illustrate the trend in accident numbers over time. All charts indicate that the number of accidents fluctuates throughout the year.

- *Total Accident Trend (2010)*

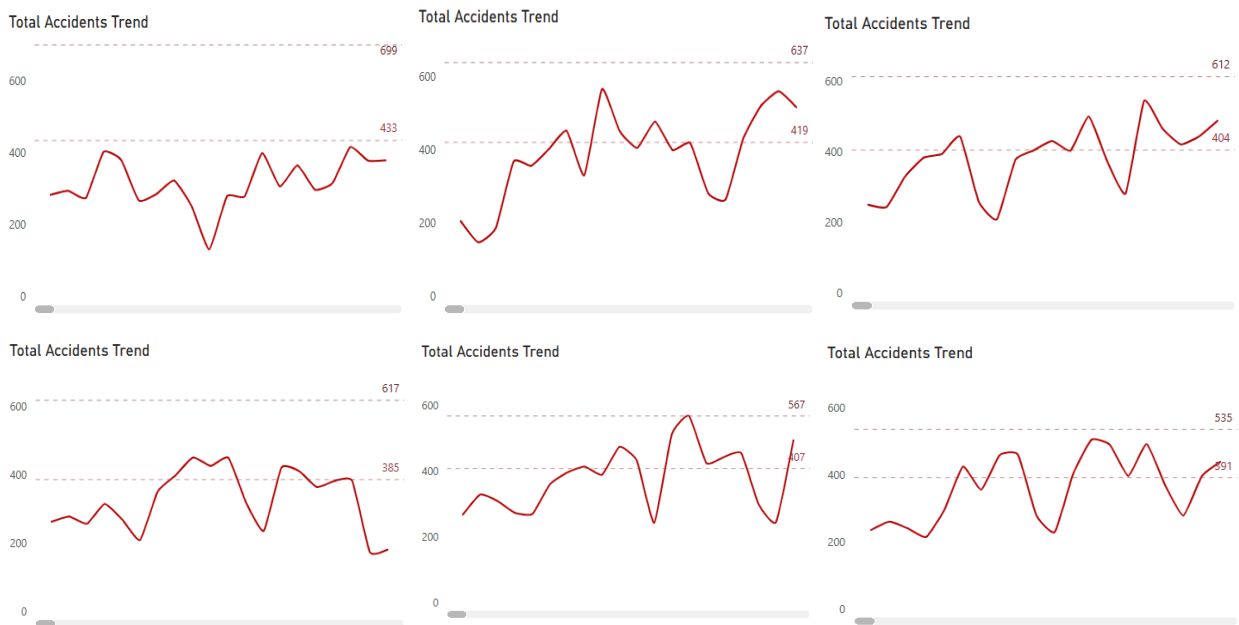
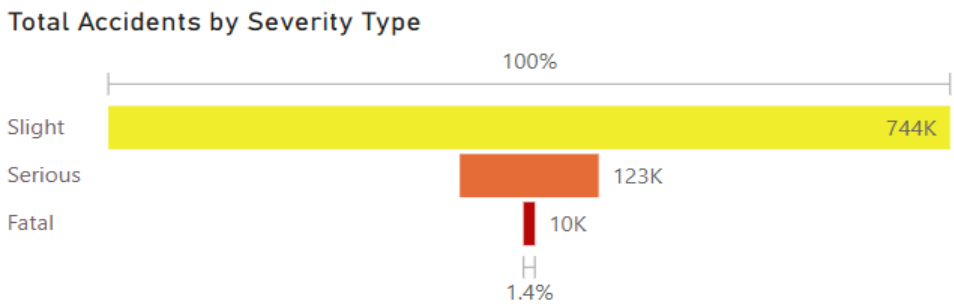


Figure 5.4. Detail total accident trend over years

All six charts exhibit significant fluctuations, particularly during mid-year periods such as summer and winter holidays, when accident numbers surge. The highest peaks each year

show a declining trend, decreasing from 699 to 535, indicating that peak accident levels are gradually reducing over time. Overall, the data reveals seasonal variations in accident numbers, yet by the end of each year, figures tend to stabilize around the average. The decreasing peak values suggest that accident control measures have been increasingly effective.

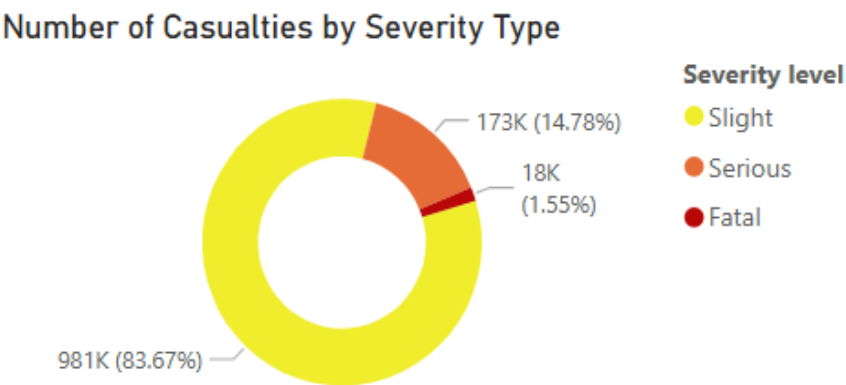
**b) What is the distribution of accidents by severity (slight, serious, fatal)?**



*Figure 5.5. Total Accidents by Severity Type*

The bar chart provides an overview of the distribution of road accidents in the UK across three severity levels. The majority of incidents are minor accidents, accounting for 744,000 cases, while serious and fatal accidents make up a smaller proportion. However, the 10,000 fatal accidents (1.4%) remain a significant figure, underscoring the severity of these incidents despite their lower occurrence.

**c) How many casualties occur at each severity level?**



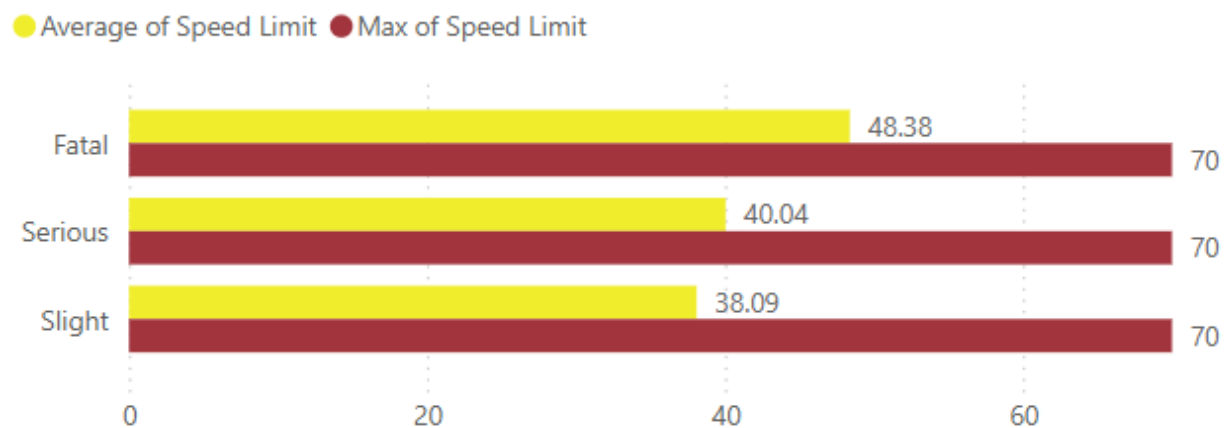
*Figure 5.6. Number of Casualties by Severity Type*

A deeper analysis of accident impact reveals the number of individuals affected across

three severity levels: Slight, Serious, and Fatal. The data highlights that the majority of casualties are minor injuries (83.67%), while serious and fatal injuries account for 14.78% and 1.55%, respectively. Although fatal accidents are less frequent, each incident can still result in severe consequences. This underscores the importance of prioritizing medical and emergency response resources in high-risk areas to ensure timely intervention and reduce casualty numbers.

#### d) How does speed limit correlate with accident severity ?

##### Speed Limit by Severity Type



*Figure 5.7. Speed Limit by Severity Type*

The chart illustrates that at the same maximum speed limit of 70 km/h, accidents involving casualties tend to occur at a significantly higher average speed (48.38 km/h) compared to serious accidents (40.04 km/h) and minor injury accidents (38.09 km/h). This suggests that higher driving speeds may increase accident severity. To mitigate this issue, stricter speeding penalties, increased surveillance through speed cameras, and enhanced monitoring of high-risk roads, such as one-way streets (as highlighted in the following chart), should be considered to improve traffic safety.

### e) Which road types have the highest number of accidents?

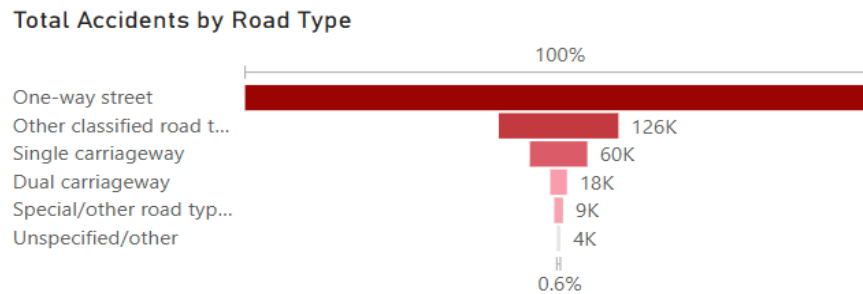


Figure 5.8. Total accident by road type

The horizontal bar chart illustrates the total number of accidents occurring across different road types. Notably, one-way streets have a significantly higher number of accidents compared to other road types. Following this, "Other classifier road type" records nearly 126,000 accidents, while single carriageways account for 60,000 accidents. When combined with insights from other dashboard visualizations, this data provides a comprehensive view of accident trends, including frequent accident locations, severity levels, and speed limits, helping to identify key risk factors and areas requiring enhanced traffic management.

### 5.3.3. Page 2 - Uncontrollable Causes

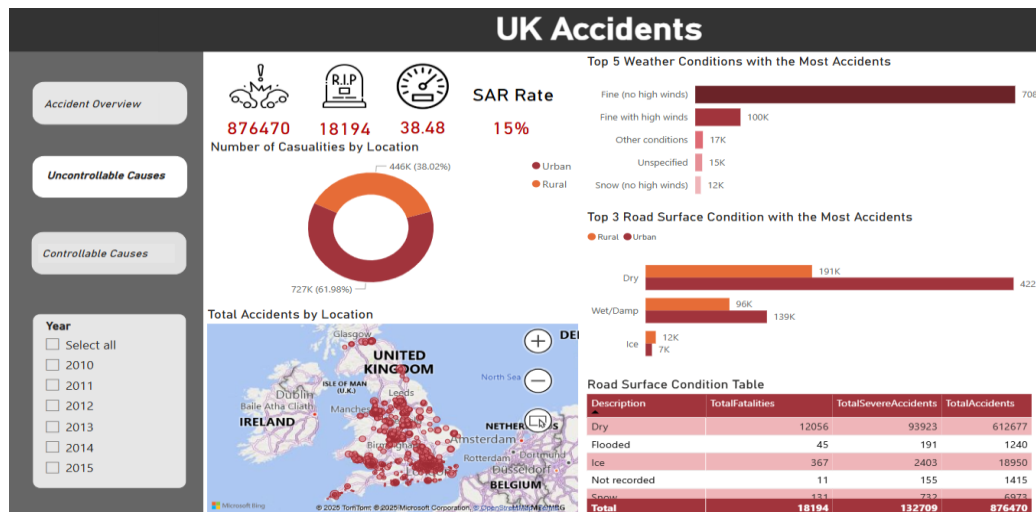


Figure 5.9. Uncontrollable Causes

The second dashboard focuses on visualizing and analyzing uncontrollable factors contributing to road accidents in greater detail. The specific insights are outlined below.

**a) Which areas have the highest accident rates?**

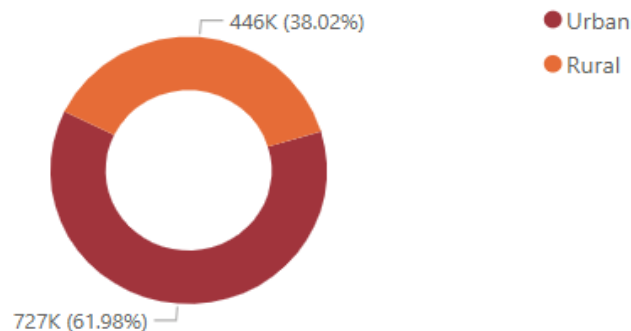


*Figure 5.10. Map displaying the total number of accidents by location in the United Kingdom*

The map visualizes the total number of accidents by location across the United Kingdom. Each red dot represents an accident, with its size and density indicating the concentration of incidents in that area. The highest accident clusters are found in highly populated regions such as London, Birmingham, Manchester, and other major cities. The elevated accident rates in these urban areas are largely influenced by high traffic density, increased vehicle flow, and the complexity of road networks.

**b) How do casualty numbers compare between urban and rural areas?**

**Number of Casualties by Location**

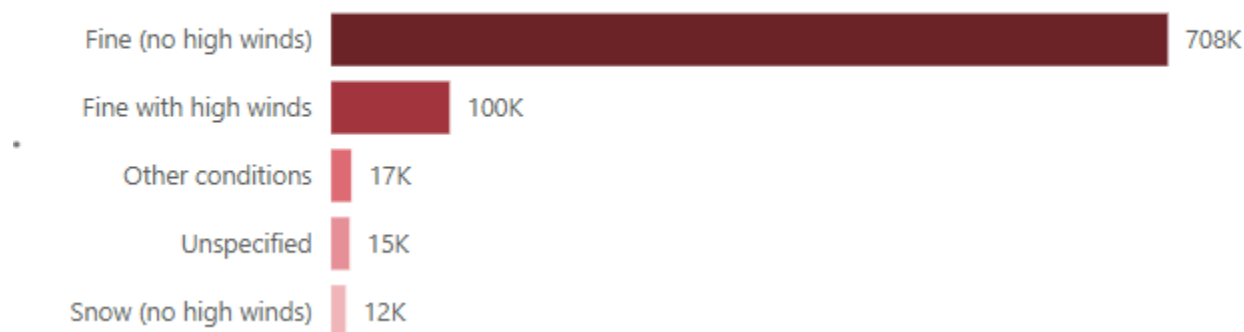


*Figure 5.11. Number of Casualties by Location*

The pie chart reveals that 61.98% of accidents occur in urban areas, while the remaining 38.02% take place in rural areas. Combined with the map above, this highlights that high traffic density in urban regions contributes to a larger number of accidents and casualties. However, the significant proportion of rural accidents suggests that these areas also require attention to improve road safety.

**c) What weather conditions contribute to the highest number of accidents?**

**Top 5 Weather Conditions with the Most Accidents**



*Figure 5.12. Top 5 Weather Conditions with the Most Accidents*

A deeper analysis of weather conditions during accidents suggests that weather is not a primary cause. The horizontal bar chart shows that 708,000 accidents occurred under clear weather with no strong winds (Fine, no high wind). As weather conditions worsened, the number of accidents decreased, with 100,000 cases in strong winds and only 12,000 in snowy conditions without wind. This indicates that human factors and infrastructure may play a more significant role in causing accidents. To further explore this, the following charts analyze infrastructure-related aspects in greater detail.



**d) How do different road surface conditions impact accident frequency?**

[< Back to report](#)

ROAD SURFACE CONDITION TABLE

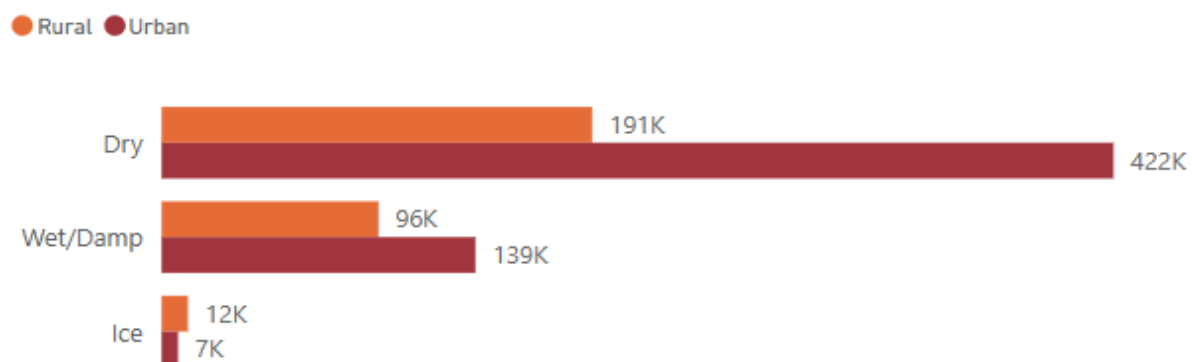
Description	TotalFatalities	TotalSevereAccidents	TotalAccidents
Dry	12056	93923	612677
Flooded	45	191	1240
Ice	367	2403	18950
Not recorded	11	155	1415
Snow	131	732	6973
Wet/Damp	5584	35305	235215
<b>Total</b>	<b>18194</b>	<b>132709</b>	<b>876470</b>

*Figure 5.13. Road Surface Condition Table*

The table presents the number of accidents across different road surface conditions. The majority of accidents occurred on dry roads (approximately 612,000 cases), followed by wet or damp roads (235,215 cases) and icy roads, which had significantly fewer incidents (18,900 cases). Although accidents are more frequent on dry roads, the fatality rate is higher on wet or icy roads. This suggests that while dry roads see a greater number of accidents, slippery road conditions increase the risk of fatal outcomes, highlighting the need for extra caution in adverse weather conditions.

**e) Which road surface conditions are associated with the most severe accidents?**

**Top 3 Road Surface Condition with the Most Accidents**



*Figure 5.14. Top 3 Road Surface Conditions with the Most Accidents*

The horizontal bar chart provides a deeper analysis of the three most accident-prone road conditions, based on the table in Section 3.4. Dry roads recorded the highest number of accidents (422K in urban areas, 191K in rural areas). Wet/damp roads followed, with 139K accidents in urban areas and 96K in rural areas. Icy roads had the lowest accident count (7K in urban areas, 12K in rural areas). While dry roads see the most accidents, incidents on slippery or icy roads tend to be more severe, emphasizing the need for enhanced safety measures under hazardous driving conditions.

5.3.4. Page 3 - Controllable Causes

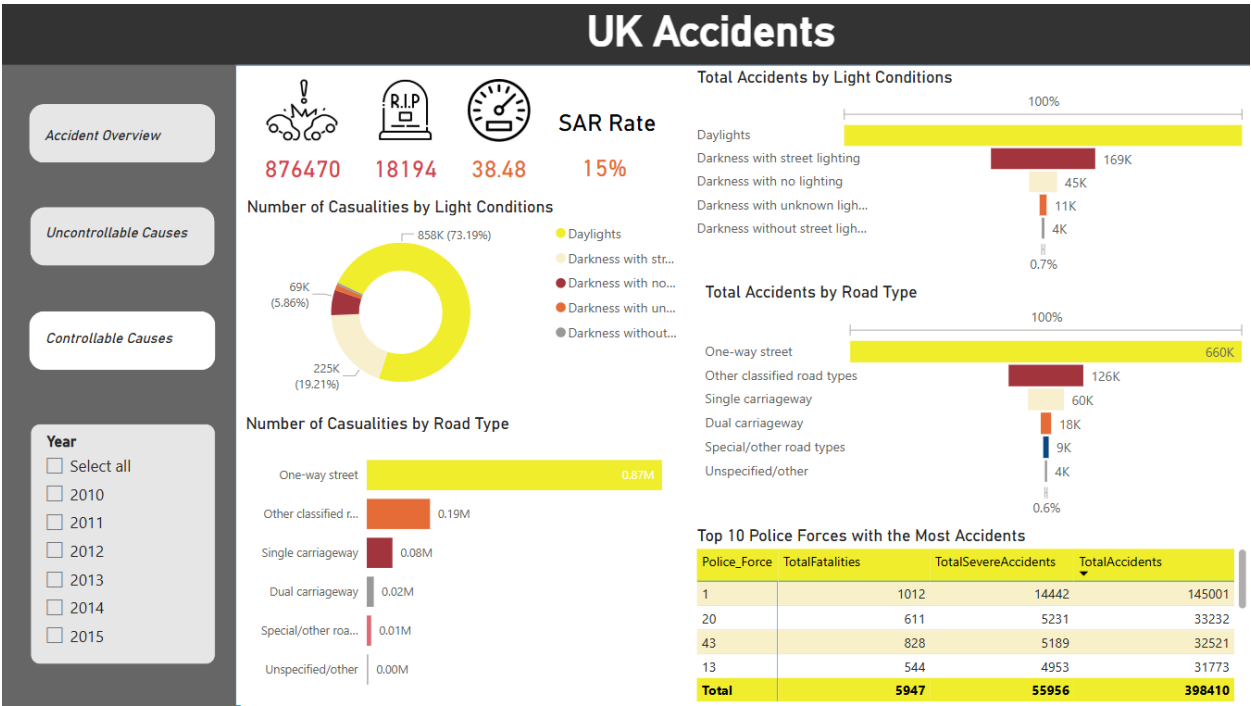


Figure 5.15. Controllable Causes Dashboard

The third dashboard provides a detailed analysis of **controllable factors** influencing road accidents, including **lighting conditions, road infrastructure, and police resources**. The following sections will present a more in-depth examination of these factors.

### a) How do light conditions impact the number of accidents?

#### Total Accidents by Light Conditions

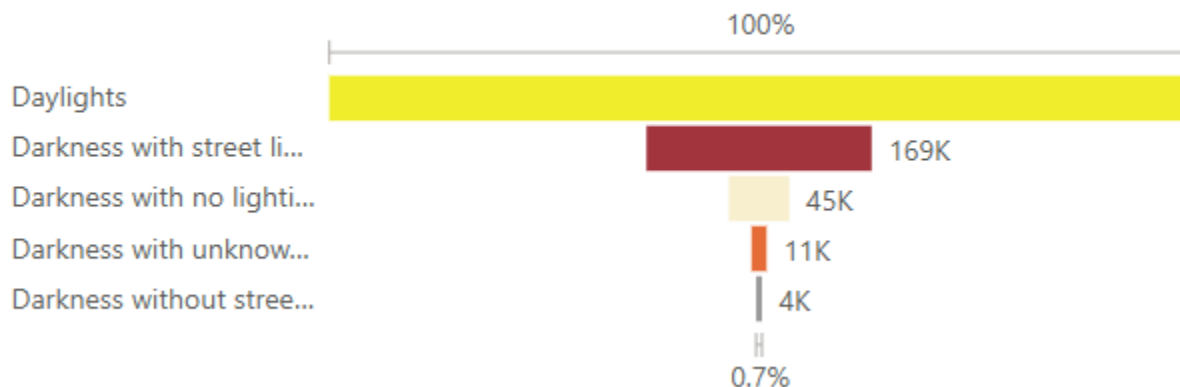


Figure 5.16. Total Accidents by Light Conditions

The chart presents different lighting conditions and the total number of accidents occurring under each type. Daylight accounts for the highest proportion of accidents, followed by accidents under street lighting conditions (169K). Accidents in dark areas without sufficient lighting are significantly lower. However, this also highlights that certain areas lack adequate lighting for road users. To enhance road safety, authorities should assess and install proper street lighting in specific locations identified in the chart, ensuring better visibility and reduced accident risks, especially in poorly lit areas.

### b) Which light conditions are associated with the highest number of casualties?

#### Number of Casualties by Light Conditions

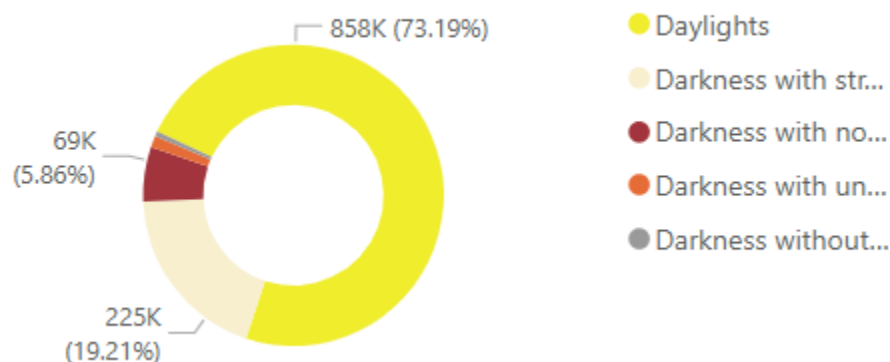
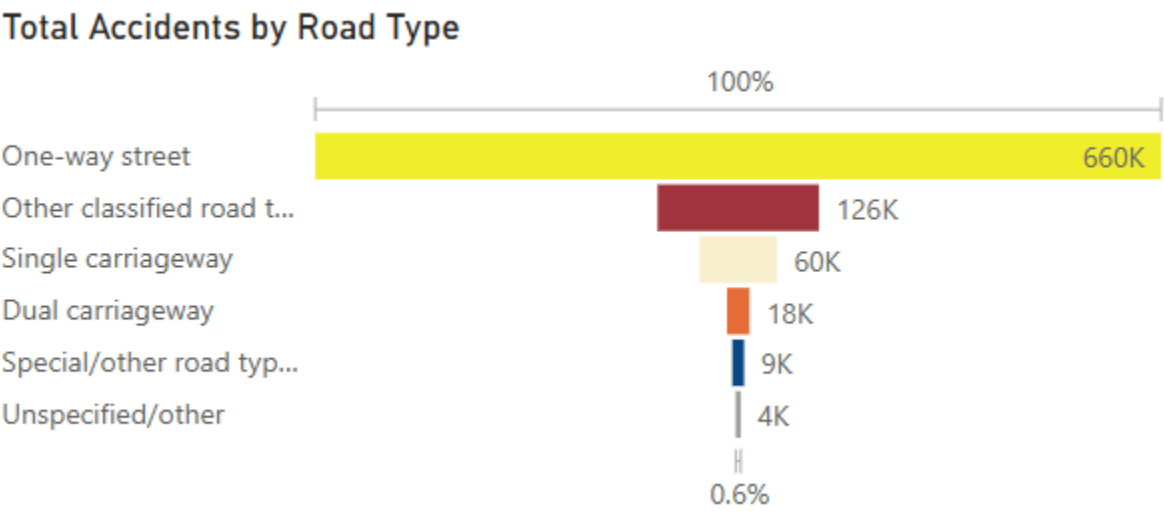


Figure 5.17. Number of Casualties by Light Conditions

The pie chart illustrates that the majority of accidents occur during daylight hours (73.19%), followed by accidents under street lighting conditions (19.21%). Accidents in dark conditions without street lighting account for a significantly lower proportion. While daytime accidents are more frequent, this may be attributed to higher traffic volume. However, nighttime accidents, particularly in areas without street lighting, tend to be more severe. This highlights the importance of enhanced nighttime visibility, improved street lighting infrastructure, and increased driver awareness to mitigate risks in low-light conditions.

**c) Which road types have the highest number of accidents?**

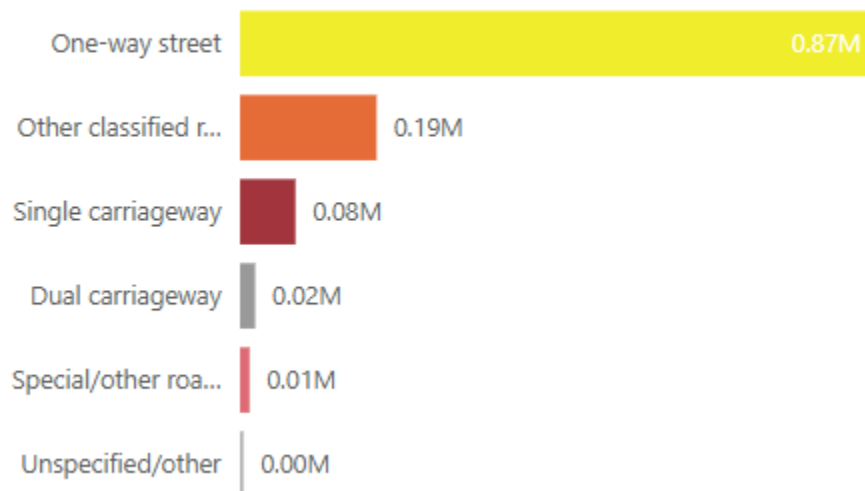


*Figure 5.18. Total Accidents by Road Type*

The highest number of accidents occurred on one-way streets (660K cases), followed by other classified road types (126K cases) and single carriageways (60K cases). One-way streets typically have high traffic density, faster vehicle speeds, and frequent traffic violations such as wrong-way driving and running red lights. These factors contribute significantly to accident occurrences, emphasizing the need for stricter traffic enforcement, improved road signage, and public awareness campaigns to enhance road safety.

**d) Which road types are associated with the most casualties?**

**Number of Casualties by Road Type**



*Figure 5.19. Number of Casualties by Road Type*

The chart above illustrates the number of casualties resulting from traffic accidents, categorized by different road types. In alignment with the "Total Accidents by Road Type" chart, one-way streets account for the highest number of casualties, approximately 0.87 million, significantly surpassing other road types. This finding reinforces the need for targeted safety measures, such as enhanced traffic regulations, stricter speed enforcement, and improved road infrastructure on one-way streets to mitigate the impact of accidents and reduce casualties.

### e) Which police forces handle the most accidents?

<a href="#">&lt; Back to report</a>		TOP 10 POLICE FORCES WITH THE MOST ACCIDENTS		
Police_Force	TotalFatalities	TotalSevereAccidents	TotalAccidents	
1	1012	14442	145001	
20	611	5231	33232	
43	828	5189	32521	
13	544	4953	31773	
46	548	3448	29511	
44	428	5862	27917	
6	470	3963	25887	
47	552	5044	24482	
50	540	3427	24292	
4	414	4397	23794	
<b>Total</b>	<b>5947</b>	<b>55956</b>	<b>398410</b>	

*Figure 5.20. Top 10 Police Forces with the most accidents*

The table provides an overview of the top 10 police forces managing the highest number of traffic accidents. Notably, Police Force No.1 recorded the greatest number of incidents, with 145,001 cases, alongside the highest figures for fatalities and serious accidents. Following this, Police Force No.20 ranks second, albeit with a significantly lower number of cases, though it remains among the most affected regions. Collectively, these 10 police forces documented a total of 398,410 accidents, including 5,947 fatalities and 55,956 serious accidents. These statistics highlight the substantial burden on specific law enforcement units, emphasizing the need for targeted interventions and enhanced traffic safety measures in high-risk areas.

#### 5.4. Recommendation

From this analysis, decision-makers can take proactive actions to reduce traffic accidents, enhance road safety, and protect lives. Specific recommendations include:

#### **5.4.1. Uncontrollable Causes**

Serious accidents frequently occur in major cities, where high population density and heavy traffic flow, especially during rush hours and holiday seasons, contribute to increased accident risks.

- Notably, most accidents happen under favorable weather conditions, indicating that weather is not the primary cause of road accidents. However, severe accidents can still occur in adverse weather, emphasizing the need for enhanced safety measures and driver awareness to ensure caution even in good weather. Additionally, improved warning systems and preventive actions should be implemented to mitigate risks during poor weather conditions.
- Similarly, most accidents take place on dry roads, suggesting that road surface conditions are not the primary factor behind accidents. Nevertheless, the risk of serious accidents remains high on slippery or icy roads, underscoring the importance of driver behavior and traffic conditions. Raising public awareness about safe driving practices on all road surfaces is crucial. Moreover, preventive measures and hazard warnings should be strengthened on high-risk roadways to reduce accident severity.

#### **5.4.2. Controllable Causes**

- Most accidents occur during the day and at night with streetlights, when lighting conditions are optimal. A smaller number of accidents happen under poor lighting conditions, which can be effectively controlled by regular inspections, maintenance, or additional lighting installations to improve visibility.
- One-way streets have a high concentration of injury-related accidents, highlighting the need for greater investments in traffic safety. This includes speed control measures, awareness campaigns, and infrastructure improvements to enhance road safety.
- Given their key role in accident response, police forces should be reinforced, with a focus on monitoring high-risk locations and investigating severe accidents.

Additionally, increased surveillance and patrolling during peak accident hours can further improve road safety.



## CHAPTER 6. CONCLUSION

*This chapter concludes the project by outlining its findings, limitations, and potential future directions. It highlights the effectiveness of the Azure-based business intelligence solution in analyzing traffic accidents in the UK, with a focus on improvements in data accessibility, automation, and visualization. Limitations noted include challenges related to data quality, real-time processing capabilities, and scalability. Looking ahead, suggested enhancements encompass real-time analytics, the integration of machine learning, and overall system optimization. The chapter underscores the project's significant contribution to data-driven decision-making in road safety and proposes potential advancements that could extend its impact.*

### 6.1. Conclusion

The project effectively showcases the design and implementation of a comprehensive Business Intelligence (BI) and Decision Support System (DSS) framework. The automation of data ingestion and transformation through Azure Functions has proven successful in extracting raw data from Azure Blob Storage, processing it in layered stages, and subsequently loading it into structured environments. This layered architecture, comprising Bronze, Silver, and Gold levels, ensured that data quality was upheld while addressing the complexities inherent in real-world datasets. By adopting a star schema design, the data was efficiently organized into fact and dimension tables, which not only optimized query performance but also facilitated multi-dimensional analyses of temporal trends, spatial distributions, environmental conditions, and infrastructural factors influencing road safety. The integration with Power BI further enhanced decision support by providing stakeholders with interactive dashboards that promote evidence-based policy making, effective resource allocation, and prompt emergency response. Overall, the project highlights the substantial potential of data-driven methodologies in tackling complex challenges in traffic safety, illustrating how modern BI tools and cloud-based architectures can transform raw data into actionable insights.

## **6.2. Limitation**

The project encountered several limitations. The reliance on CSV-based datasets and historical records led to issues with missing or inconsistent data, particularly regarding spatial coordinates and categorical fields. While techniques such as binary flags were implemented to address these challenges, the inherent imperfections in the source data hindered our ability to achieve higher analytical precision.

The system primarily employed scheduled triggers for batch processing, which, although effective for periodic updates, may not be ideal for situations requiring real-time or near real-time analytics. This limitation could impact the responsiveness of the decision support framework in dynamic environments. Additionally, as data volumes continue to expand, concerns around scalability arise, particularly with processing capabilities and overall system performance. The current architecture may require further optimization or a shift toward more scalable, serverless solutions. Moreover, despite the inclusion of logging mechanisms and alert systems, there is potential for enhancement in error-handling processes and in ensuring comprehensive monitoring during high-load periods.

## **6.3. Future works**

Building on the foundations established by this project, several avenues for future research and development are envisioned. One notable area for improvement involves the incorporation of advanced data cleansing techniques and the integration of additional data sources, such as IoT sensor data and real-time social media feeds, to provide a more comprehensive view of traffic dynamics.

Transitioning from batch processing to real-time data ingestion and analytics presents another promising opportunity. By exploring event-driven architectures and stream processing frameworks, the system could attain near real-time decision support capabilities. Furthermore, integrating predictive analytics and machine learning algorithms could enable proactive identification of accident hotspots and enhance the accuracy of traffic incident forecasting, enriching the decision support system with anticipatory insights.

Additional enhancements in system scalability and cloud optimization are vital to accommodate growing data volumes and increasingly complex analytical demands. Investigating more advanced serverless architectures or distributed computing solutions would ensure that the platform remains high-performing. Finally, reinforcing the error-handling framework with more sophisticated monitoring tools and adaptive retry mechanisms could significantly improve system resilience, allowing for timely detection and resolution of issues during periods of high load.

In conclusion, while the project has established a robust foundation for a comprehensive BI & DSS system for traffic accident analysis, addressing these limitations and pursuing the proposed future enhancements will be critical for evolving the system into a more dynamic, scalable, and predictive platform for public safety and policy decision-making.

## REFERENCES

- [1] Haddon Jr, W. (1972). A logical framework for categorizing highway safety phenomena and activity. *Journal of Trauma and Acute Care Surgery*, 12(3), 193-207
- [2] Quimby, A., Maycock, G., Palmer, C., & Buttress, S. (1999). *The Factors the Influence a Driver's Choice of Speed: A Questionnaire Study* (Vol. 325). Crowthorne, UK: Transport Research Laboratory.
- [3] Stipdonk, H., & Berends, E. (2008). Distinguishing traffic modes in analysing road safety development. *Accident Analysis & Prevention*, 40(4), 1383-1393.
- [4] Erdogan, S., Yilmaz, I., Baybura, T., & Gullu, M. (2008). Geographical information systems aided traffic accident analysis system case study: city of Afyonkarahisar. *Accident Analysis & Prevention*, 40(1), 174-181.
- [5] Alkheder, S., Taamneh, M., & Taamneh, S. (2017). Severity prediction of traffic accident using an artificial neural network. *Journal of Forecasting*, 36(1), 100-108.
- [6] Dupont, E., Papadimitriou, E., Martensen, H., & Yannis, G. (2013). Multilevel analysis in road safety research. *Accident Analysis & Prevention*, 60, 402-411.
- [7] Tilahun, N., Thakuriah, P. V., Li, M., & Keita, Y. (2016). Transit use and the work commute: Analyzing the role of last mile issues. *Journal of Transport Geography*, 54, 359-368.
- [8] Kass-Hout, T. A., Xu, Z., Mohebbi, M., Nelsen, H., Baker, A., Levine, J., ... & Bright, R. A. (2016). OpenFDA: an innovative platform providing access to a wealth of FDA's publicly available data. *Journal of the American Medical Informatics Association*, 23(3), 596-600.
- [9] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
- [10] Janssen, M., & Joha, A. (2011). Challenges for adopting cloud-based software as a service (saas) in the public sector.
- [11] Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons.

- [12] Inmon, W. H. (2005). Building the data warehouse. John Wiley & sons.
- [13] Warren, J., & Marz, N. (2015). Big Data: Principles and best practices of scalable realtime data systems. Simon and Schuster.
- [14] Davenport, T. H., & Harris, J. G. (2007). The architecture of business intelligence. Competing on analytics: The new science of winning.
- [15] Cosic, R., Shanks, G., & Maynard, S. (2012). Towards a business analytics capability maturity model.
- [16] Klievink, B., Romijn, B. J., Cunningham, S., & de Bruijn, H. (2017). Big data in the public sector: Uncertainties and readiness. Information systems frontiers, 19(2), 267-283.
- [17] Sadalage, P. J., & Fowler, M. (2012). Introduction to Polyglot Persistence: Using Different Data Storage Technologies for Varying Data Storage Needs. Introduction to Polyglot Persistence: using Different data Storage Technologies For Varying Data Storage Needs.
- [18] Vassiliadis, P., & Simitsis, A. (2009). Extraction, Transformation, and Loading. Encyclopedia of database systems, 10, 14.
- [19] Abadi, D. J., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Lee, S., ... & Zdonik, S. (2003). Aurora: a new model and architecture for data stream management. the VLDB Journal, 12, 120-139.
- [20] DeWitt, D., & Gray, J. (1992). Parallel database systems: The future of high performance database systems. Communications of the ACM, 35(6), 85-98.
- [21] Stonebraker, M., Çetintemel, U., & Zdonik, S. (2005). The 8 requirements of real-time stream processing. ACM Sigmod Record, 34(4), 42-47.
- [22] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. Research advances in cloud computing, 1-20.