💻 Data Warehouse Project 2 S1, 2025

Project 2 - Graph Database Design and Cypher Query

This is an **individual** project. Project 2 contributes **20%** to the total assessment of this unit. The submission deadline is **May 23 at 11:59 PM**. Submit a zip file on **LMS**.

This project is designed to practise solving business problems using graph database modelling techniques on a real-world dataset.

Project 2 Dataset

2MB Project2_Dataset.csv

Project 2 Dataset (an data entry error for Crash: 20164024)

2MB Project2_Dataset_Corrected.csv

Project 2 Dataset (corrected)

The dataset is obtained from ARDD: Fatalities—December 2024—XLSX ↗ by removing all missing/unknown values, modifying values, and adding new columns. It has 10490 rows (excluding the header row) and 25 columns, stored in CSV format.

Below is a data dictionary describing the columns.

Field	Description	Format	Comments
ID	Surrogate key	Integer	Unique identifier

Crash ID	National crash identifying number	Text	8 digits, not unique identifier
State	Australian jurisdiction	Text	
Month	Month of crash	Integer	
Year	Year of crash	Integer	2014-2024
Dayweek	Day of week of crash	Text	
Time	Time of crash	Time	hh:mm
Crash Type	Refers to the number of vehicles involved.	Text	
Number Fatalities	Number of killed persons (fatalities) in the crash	Integer	
Bus Involvement	Indicates involvement of a bus in the crash	Text	
Heavy Rigid Truck Involvement	Indicates involvement of a heavy rigid truck in the crash	Text	
Articulated Truck Involvement	Indicates involvement of an articulated truck in the crash	Text	
Speed Limit	Posted speed limit at location of crash	Text	
Road User	Road user type of killed person	Text	

Gender	Sex of killed person	Text	
Age	Age of killed person (years)	Integer	
National Remoteness Areas 2021	ABS Remoteness Structure ↗	Text	
Statistical Areas Level 4 (SA4) Name 2021	Australian Statistical Geography Standard ↗	Text	
National Local Government Areas (LGAs) Name 2024	Australian Statistical Geography Standard ↗	Text	
National Road Type	Geoscape Australia, Transport and Topography	Text	
Christmas Period	Indicates if crash occurred during the 12 days commencing on December 23rd	Text	
Easter Period	Indicates if crash occurred during the 5 days commencing on the Thursday before Good Friday	Text	
Age Group	Standard age groupings used in the Road Deaths Australia monthly bulletin	Text	
	Indicates if crash occurred during		

Day of week	the weekday or weekend. (Note: ‘Weekday’ refers to 6am Monday through to 5:59pm Friday)	Text	
Time of day	Indicates if crash occurred during the day or night	Text	

Graph Database Design, Implementation and Queries

1. Design a property graph for the provided dataset, using the Arrows App ↗ to draw the nodes and relations. Your design should be able to answer **ALL** questions in **item 3** below.

> ⓘ Note: The Arrows App has not been stable recently, but you can use the alternative link ↗ to access it. Please note that the alternative Arrows App doesn't support opening a file from Google Drive, and you cannot save files to Google Drive with this version

2. Implement the property graph in Neo4J: Perform ETL to transfer the provided dataset into CSVs and then import these CSVs into Neo4j.
3. Write Cypher queries to answer the following questions.
    a. Find all crashes in **WA** from **2020-2024** where **articulated trucks** were involved and multiple fatalities (**Number Fatalities>1**) occurred. For each crash, provide the **road user**, **age** of each road user, **gender** of each road user, **LGA Name**, **month** and **year** of the crash, and the **total number of fatalities**.
    b. Find the maximum and minimum age for female and male **motorcycle riders** who were involved in fatal crashes during the **Christmas Period** or **Easter Period** in **inner regional Australia**. Output the following information: **gender**, **maximum age** and **minimum age**. (Hint: Zero results is a meaningful result in itself.)

c. How many young **drivers** (**Age Group = '17_to_25'**) were involved in fatal crashes on weekends vs. weekdays in each state during **2024**? Output 4 columns: **State name**, **weekends**, **weekdays**, and the **average age for all young drivers (Age Group = '17_to_25') who were involved in fatal crashes in each State**.

d. Identify all crashes in **WA** that occurred **Friday** (but categorised as a **weekend**) and resulted in **multiple deaths**, with victims being **both male and female**. For each crash, output the **SA4 name**, **national remoteness areas**, and **national road type**.

e. Find the top 5 SA4 regions where the highest number of fatal crashes occur during peak hours (**Time between 07:00-09:00 and 16:00-18:00**). For each SA4 region, output the **name** of the region and the separate number of crashes that occurred during morning peak hours and afternoon peak hours (Renamed Morning Peak and Afternoon Peak).

f. Find paths with a length of 3 between any two LGAs. **Return the top 3 paths, including the starting LGA and ending LGA for each path**. Order results alphabetically by starting LGA and then ending LGA.

g. **(CITS5504 ONLY)** Find all weekday fatal crashes involving **pedestrians** where either **buses or heavy rigid trucks** were present in **speed zones less than 40 or greater than/equal to 100**. Group these crashes by unique combinations of **time of day**, **age group**, **vehicle type** (bus or heavy rigid truck), and **speed limitation**. For each group, count the number of crashes that occurred. Output a table showing **time of day**, **age group**, **vehicle type**, **crash count**, and **speed limitation**, sorted first by **time of day** (ascending) and then by **age group** (ascending).

> ! The Cypher query performance could be very slow if you use the original dataset for **question f**. You may use a smaller dataset for **question f**. For questions a to e, and g, the Cypher query performance on the original dataset is good.

4. Write Cypher queries for at least two other meaningful queries that you can think of.

> ⓘ **Performance Note:** The Neo4j engine might experience slow query performance with a large dataset, depending on your device's capabilities. If

you experience long query time, consider reducing the dataset size by filtering out records, such as deleting records for specific State/Year, to enhance performance.

Cannot install Neo4j Desktop on your own device: Please use Neo4j Aura for this project. This project does not involve concepts of graph data science.

Submission Requirement: Please submit both the original (unfiltered) CSVs and the filtered (smaller) CSVs in two separate folders. While you can run your Neo4j queries on the smaller CSVs for performance reasons, it's essential to provide the original CSVs for our reference and verification. You have the option to filter records either directly in the original CSVs, within Neo4j using Cypher commands like "DELETE", or through programming languages. **Please determine the most suitable option for your needs.**

In summary, the process:

1. Design your property graph;
2. Perform ETL to transform the provided dataset into CSV files using Excel or other programming language;
3. Import CSVs into Neo4j using Cypher;
4. Write and execute queries. If queries are very slow on your device, filter out some records and try all queries again. Submit both the original (unfiltered) CSVs and the filtered (smaller) CSVs in two separate folders. If your queries run efficiently without filtering, then there's no need to filter out records. In such cases, submit only the original (unfiltered) CSVs.

Submission Guidelines

⚠ In-text citations and end-text references are required in this project report, following the IEEE referencing style ↗.

1. **Script Files**

- **Cypher Scripts in ".txt" Format:** Submit **ONE** text file with **all** Cypher scripts for graph database creation (e.g. create nodes, relationships and properties), data load, and querying (all given queries and your own queries).

- **Other Script Files**: You may use any programming language to create CSV files for nodes and relationships; please submit your scripts in a suitable format.
    - *If you are going to use Excel to create CSV files for nodes and relationships, please ignore the script submission requirement.*

2. **CSV Files:** The CSV files for populating your database.
3. **A PDF contains:**

- Describe/Discuss the design and implementation process with references to graph data modelling techniques to support your design choices.
- A screenshot of your design created in the Arrows app ↗.

ⓘ Note: The Arrows App has not been stable recently, but you can use the alternative link ↗ to access it. Please note that the alternative Arrows App doesn't support opening a file from Google Drive, and you cannot save files to Google Drive with this version

- The code, screenshots and explanations for the ETL process.
    - *Please include key screenshots and/or code in your report, rather than all your code.*
- *If you filter out some data due to the performance issue, specify which data was filtered and explain how you filtered them out (e.g. the smaller dataset for question f).*
- Screenshots of queries (graph database implementation, the given queries and your own queries) and their corresponding results.
    - *Some of the queries may have thousands of results, please do not include the actual results in the PDF - discuss interesting ones with the aid of screenshots.*
- Discuss how Graph Data Science can be applied to at least one practical application with references to the graph algorithm of choice.

ⓘ **NOTE**: All files need to be zipped up in **a single zip file** and submitted to LMS ↗ (Under the **Projects channel** that can be found in the left navigation panel).

Marking Scheme

Scheme	Marks
Graph database design using Arrows app ↗.	5
Discussions of the design choices with pros and cons identified	5
ETL: describes the complete process of transforming the provided dataset to CSVs (including any filtering methods and specific steps taken), including key code, explanations and important screenshots in your PDF file.	5
Graph database Implementation with Cypher code, including code and key screenshots of database stats (#nodes, #relations, labels, and relationships) in your PDF file.	5
Correctly answering the specificed questions using Cypher.	5
Your self-designed queries with Cypher.	5
Discuss how graph data science can be applied to at least one useful application of this graph database.	5
Report quality (the PDF file).	5
Total	**40**

You can interpret the scale of marks as:

5 - Exemplary (comprehensive solution demonstrating professional application of the knowledge taught in the class with initiative beyond just meeting the project requirement)

4 - Proficient (correct application of the taught concepts with clear understandings demonstrated)

3 - Satisfactory (managed to meet most of the project requirements)

2 - Developing (some skills are demonstrated but needs revision)

1 - Not yet Satisfactory (minimal effort)

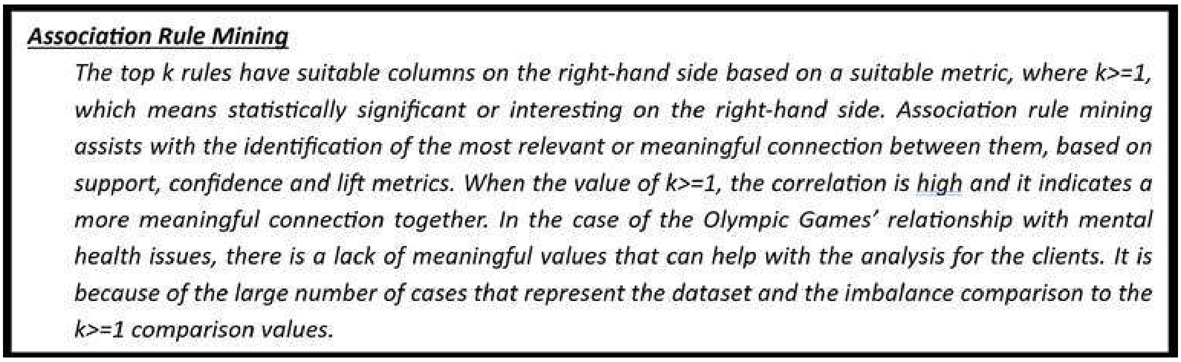
0 - Not attempted.

Can I Use Generation AI?

Informed and educated use of generative AI tools is encouraged and should be declared. If you choose to use generative AI, please document its suggestions in your report and provide a clear rationale for accepting or rejecting each recommendation. While using generative AI is optional, as a data science professional, you are strongly encouraged to develop critical evaluation skills and build familiarity with modern tools that could enhance your productivity.

> ⚠ **NOTE:** Using generative AI without proper citation and documentation is unacceptable.

The following example is a BAD example that may risk receiving a 0 mark.



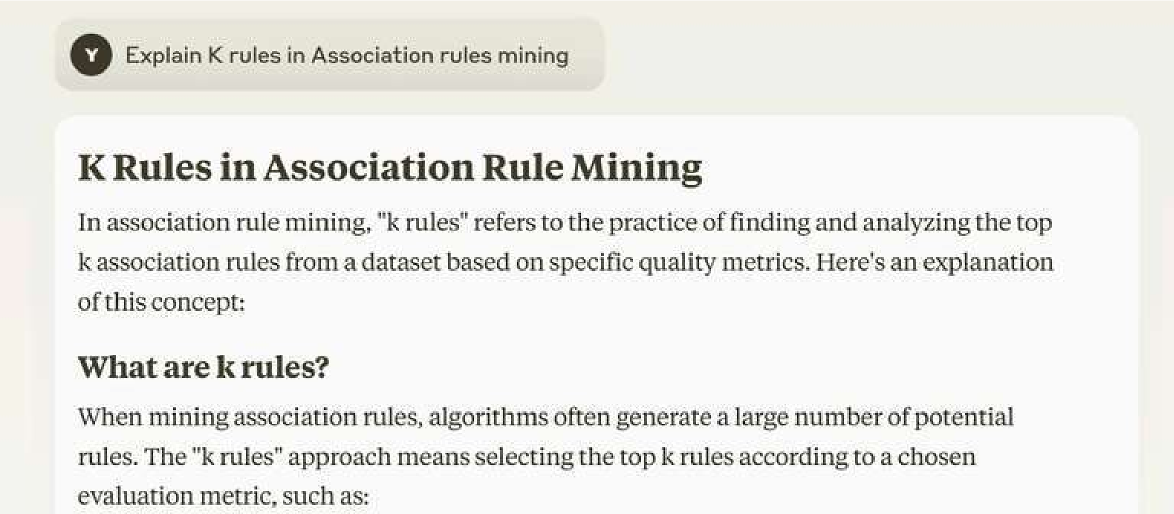
Association Rule Mining

The top k rules have suitable columns on the right-hand side based on a suitable metric, where k>=1, which means statistically significant or interesting on the right-hand side. Association rule mining assists with the identification of the most relevant or meaningful connection between them, based on support, confidence and lift metrics. When the value of k>=1, the correlation is high and it indicates a more meaningful connection together. In the case of the Olympic Games' relationship with mental health issues, there is a lack of meaningful values that can help with the analysis for the clients. It is because of the large number of cases that represent the dataset and the imbalance comparison to the k>=1 comparison values.

Hallucinated wrong answer generated from LLM

In this case, `k>1` means the student should explain more than 1 top association mining rule. However, the student misunderstood the `k` value and the concept of lift. This is a common error when using GenAI, which sometimes generates non-existent theories - a phenomenon called 'AI hallucination ↗'.

The explanations below are correct understandings.



Explain K rules in Association rules mining

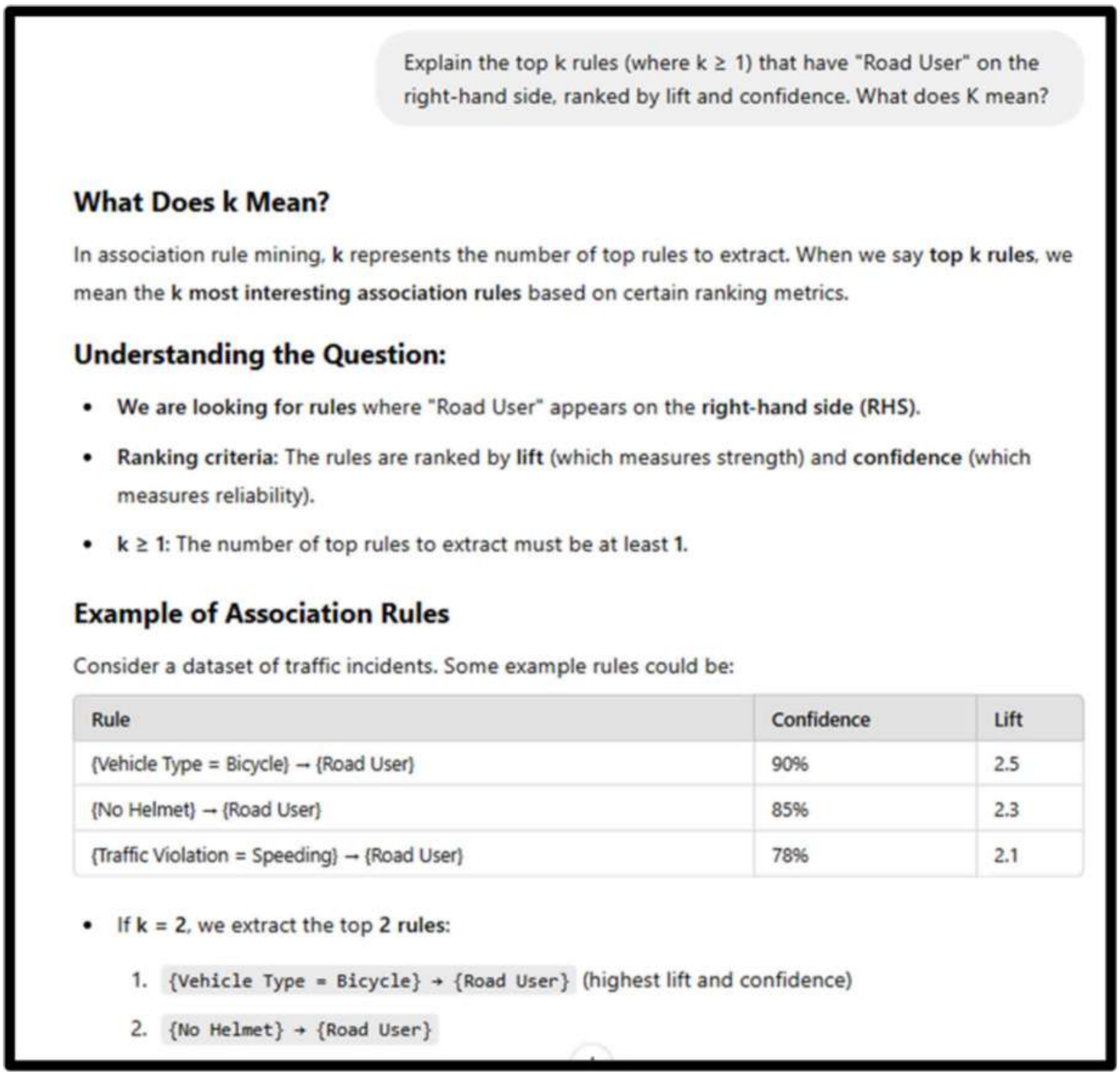
K Rules in Association Rule Mining

In association rule mining, "k rules" refers to the practice of finding and analyzing the top k association rules from a dataset based on specific quality metrics. Here's an explanation of this concept:

What are k rules?

When mining association rules, algorithms often generate a large number of potential rules. The "k rules" approach means selecting the top k rules according to a chosen evaluation metric, such as:

Correct Explanation



Explain the top k rules (where k ≥ 1) that have "Road User" on the right-hand side, ranked by lift and confidence. What does K mean?

What Does k Mean?

In association rule mining, **k** represents the number of top rules to extract. When we say **top k rules**, we mean the **k most interesting association rules** based on certain ranking metrics.

Understanding the Question:

- **We are looking for rules** where "Road User" appears on the **right-hand side (RHS)**.
- **Ranking criteria:** The rules are ranked by **lift** (which measures strength) and **confidence** (which measures reliability).
- **k ≥ 1:** The number of top rules to extract must be at least **1**.

Example of Association Rules

Consider a dataset of traffic incidents. Some example rules could be:

Rule	Confidence	Lift
{Vehicle Type = Bicycle} → {Road User}	90%	2.5
{No Helmet} → {Road User}	85%	2.3
{Traffic Violation = Speeding} → {Road User}	78%	2.1

- If **k = 2**, we extract the top **2 rules:**
  1. `{Vehicle Type = Bicycle} → {Road User}` (highest lift and confidence)
  2. `{No Helmet} → {Road User}`

Correct Explanation

Last updated 1 day ago