- Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

  We pretty much achieved all the functionality that we proposed in the project proposal, in addition to the basic CRUD requirements, we also included an interactive map on the front page of our website, which aligns with the initial idea. However, one of the things that we need to further improve is the login page and functionality. We only designed a basic login button but did not fully implement it.

- Discuss what you think your application achieved or failed to achieve regarding its usefulness.

  We believe our application successfully met its fundamental goal of displaying railroad derailments by different railroad names and time windows on a map, complete with accident information accessible by hovering the cursor. Furthermore, our stored procedure also displays detailed information on derailment rate trends by cause and operational statistics, highlighting the top 15 railroad companies with the largest traffic volume. All these functionalities achieved the main objective of this project, which is to demonstrate railroad safety and operational statistics using real datasets.

- Discuss if you changed the schema or source of the data for your application.
  We did not change the source or schema of the data.

- Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
  We did change the table implementations a little bit by adding a table to store the many-to-many relationship: one derailment accident may have one or more accident causes. This is the correct way of handling the many-to-many relationship. The way we had before was adding a column in the accident database to store the possible secondary accident cause, which is not the ideal way.

- Discuss what functionalities you added or removed. Why?
  We did not fully implement the login functionality because of time constraints.

- Explain how you think your advanced database programs complement your application.
  We have implemented several constraints on the backend database to ensure the data's cleanliness and integrity. For example, we implemented triggers to make sure the imported new data followed the existing format and also several foreign key constraints among multiple tables to ensure the proper joining operations.

- Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

  Xinhao Liu: The team encountered difficulties when importing data into the MySQL@GCP database. After designing the database table schema in MySQL, we were unable to import all the prepared data (stored in CSV format) into the MySQL database; only 1/10 of the data was imported for three of the five tables. We later figured out that the issue was due to the fact that the remaining 9/10 of the data had a null value in one of the columns which was used to document a many-to-many relationship, and that column was set as a foreign key of another table. MySQL@GCP does not allow empty foreign key values during importation. The team resolved this problem by adding another table to store this information, accommodating the many-to-many relationship.

  Zhuoang Tao: The team struggled to load data from GCP databases using stored MySQL queries and stored procedures. Since the team had no experience building websites before, it took us a long time to understand how to use a front-end button to get the corresponding resources from the backend and databases. We used ChatGPT to generate a template about this functionality and followed the tutorial videos on YouTube to learn how to use and modify this template. We also encountered some obstacles while debugging related codes; however, we improved our teamwork skills during this process. Two teammates debugged the codes; one teammate checked the website functionalities and reported the console log of the localhost website; and one teammate checked whether GCP databases had been modified correctly.

  Aoyang Li: Another problem we encountered during the project is on how to use the router to direct to different pages using buttons. We took a long time to figure out the sequence of <Router>, <Routes>, what content should be between them,

and what components should be outside. We tried various options and finally got what we wanted on the page appearance and functionality.

Haiyang Qin: The problem we encountered during this project was trying to connect the database from the backend to the frontend. Since it is my first time learning to connect the backend to the frontend, I watched some videos and also asked ChatGPT for further help. At first, I did it successfully with one sql query on my PC. Then, I ask my teammates to pull the code from git, here comes the problem. They cannot open the website, or they can open it but cannot load the query results and give the error information: listen EADDRINUSE: address already in use :::3000.
After searching information from Stackoverflow and chatgpt, we know that different users should have different frontend ports. We solved this problem by given each a unique frontend ports and just open backend on ones's PC.

- Are there other things that changed comparing the final application with the original proposal?
  We did not change much from the original proposal. The primary objectives were aligned pretty well.

- Describe future work that you think, other than the interface, that the application can improve on.
  1. Enable CURD for all the tables
  2. Fully implement the login functionality which only administrators can make the update and delete.
  3. Add more interactive plotting to show the trending statistics.
  4. Add more filtering functionality to the left sidebar

- Describe the final division of labor and how well you managed teamwork.
  The team split the work quite evenly and followed the original labor division. It was a great work achieved by everyone!