

Stage 6: Project Report

Group 18

Enya Sun, Rushika Kumarswamy, Sriya Mikkilineni, Chen Wang

1. 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).

The main change from our project proposal that our final project reflects is a change in the user interface.

Originally, we had planned to implement an interactive interface design where users can pinpoint data about flights, airports, and airline routes on a visual map. This involved color coding to display differences in delays and cancellations for different origin airports, destinations, and airlines. We also aspired to add a toggling component where more information about the different columns from our data tables would be displayed when a user hovers over a certain airport.

Unfortunately, we were unable to implement these user interface components due to both time limitations and the learning curve. Since this was all of our group members' first time working with SQL and GCP in combination with HTML/CSS, we spent a lot of time trying to thoroughly understand how to use and integrate these different platforms to create one robust and functional project. Our user interface for the final project steered in the direction of a more direct and easy-to-understand interface for users. We included multiple different form fields for users to customize the data that they want to be shown by our interface. Users also are able to add and delete data as they deem necessary and manipulate flight data so that they can receive the information that is relevant to their needs. Once they have made the desired changes/customizations to the data, they can see a table with all of the flight information from our dataset that matches their criteria.

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

The foundation of our database lies in the organizational structure of its contents, and we stored our data with carefully selected tables and coherent relations. This made the process of accessing the contents faster and allows for users to better comprehend the purpose and usage of our application. We were also able to

provide an image of the database relations in our website, which made it more user-friendly. The credibility of our data has remained consistent throughout the proposal to final stages (using real values from 2015 flight delays and cancellations), but we lack a process that checks for the validity of data after performing CRUD operations. This would have been extremely difficult to implement, but it does not necessarily diminish the usefulness of our application as users can individually upload their data into our tables and make more personalized and selective additions/edits that best suit their interests.

Our application works with large amounts of data, and although we were able to implement basic insertion, deletion, and update operations, they were somewhat inefficient. The usefulness of these operations lies in their ability to easily alter large amounts of data, and although these processes were implemented, their execution (mainly through the backend) was somewhat weak. Despite this, we found that adding stored procedures and triggers increased the versatility of our options, and our application would likely benefit from more of these implementations.

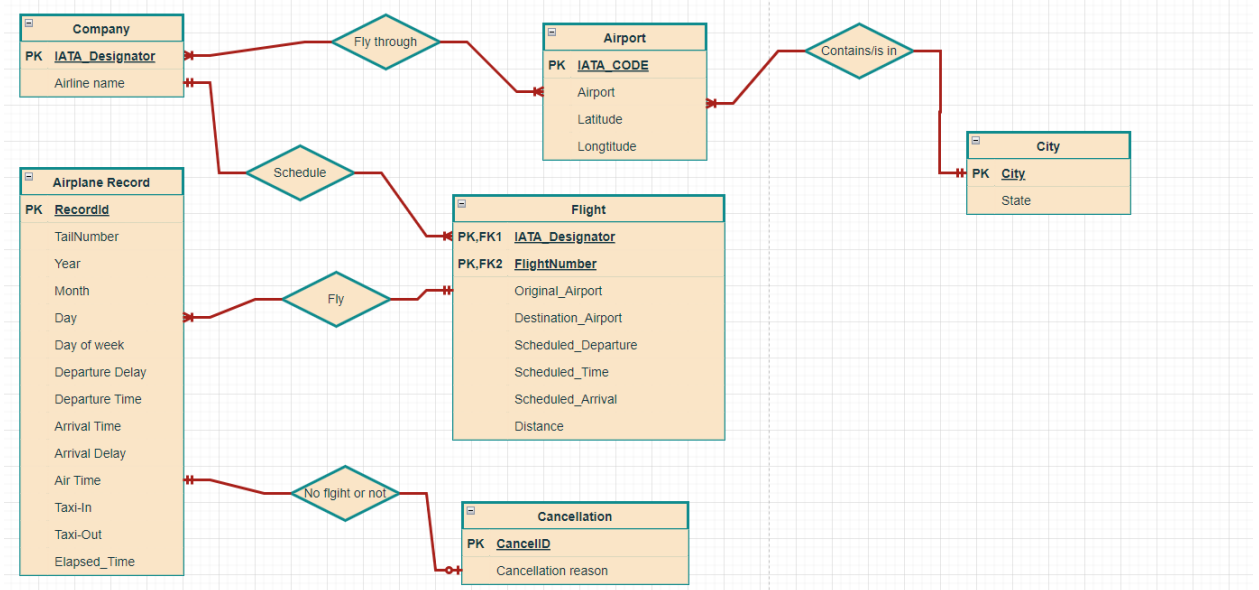
3. Discuss if you changed the schema or source of the data for your application

We changed the schema for our application in the following ways: The first thing we wanted to do was to create a virtual ID as a primary key in each table, then reference other tables by foreign keys. While our logic is sound and usable, the foreign key constraints will limit our data input if there is no primary key in the specific table (i.e. IATA_Designator should be found in the Company table, or you can not input new row data in the Airplane_Record and the Flight table). Because we combined two datasets into a database system, there remains some mismatches that we can't input the data into the GCP. So finally we deleted the foreign key between the Flight table V.S the Company table and the Airplane_Record table V.S the Company table.

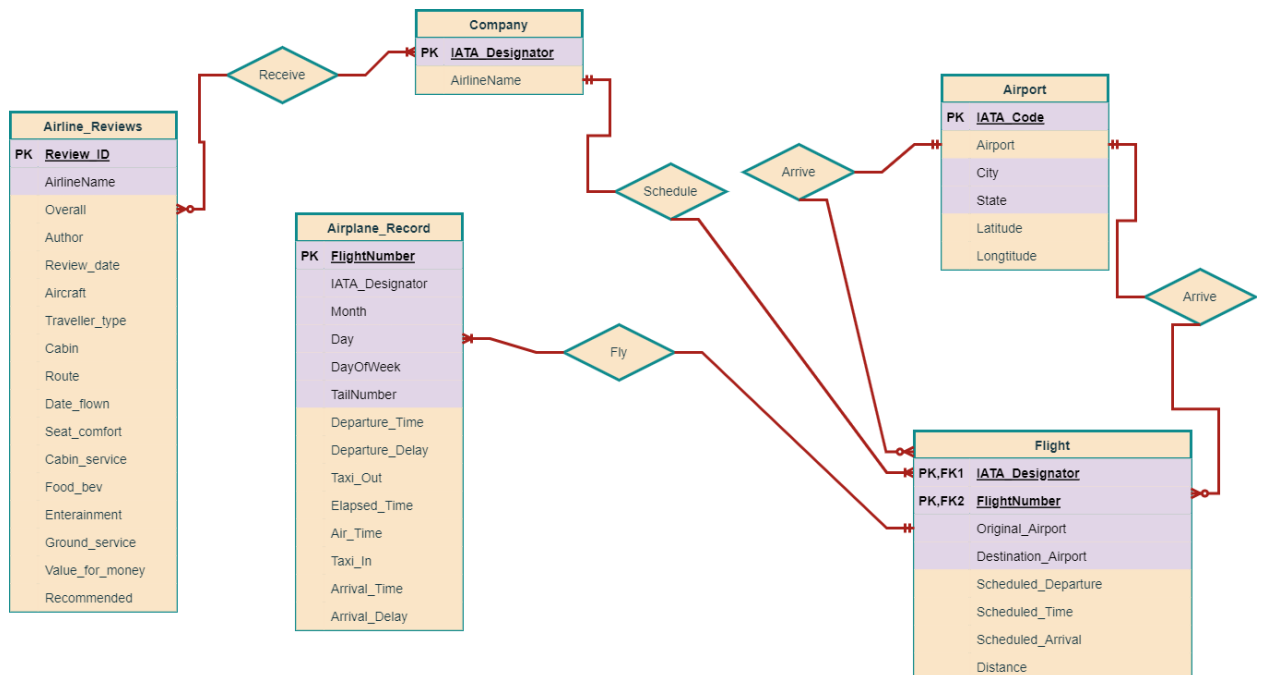
Our source data was also changed such that it was reduced into a more manageable scale of data. It was originally far too much data for us to load onto our GCP due to both time limitations as well as storage limitations with our accounts, so we had to condense the data into a representative yet smaller set of data from the original.

4. Discuss what you changed 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?

This is our first version of the schema.



And this is our latest one.



You can see in the first version, we separated the Airport table into the Airport and the City table, because our TA mentioned that separating the Airport table into two subtables is meaningless. We also introduced another source of data to the Airline_Reviews in order to meet the requirements. But in reality, it will be risky because the two different datasets are not 100% compatible. The second one is a more suitable design because it reduces redundancy between tables and yields itself to a more diverse range of usability than the first one.

5. Discuss what functionalities you added or removed. Why?

We had originally planned for users to be able to add, delete, and edit any raw data through a legible website, and we were able to implement that. Additionally, we added a store procedure and a trigger that calculated the average delay time per airline and automatically accounted for airline specific delays upon inserting or altering data respectively.

We removed the filtering and predictive complements of our application as they were beyond the scope of our current ability, and they did not fit well with our final website presentation. Unfortunately, any updates to data can only be viewed from the backend through manually opening the affected tables, so it we were unable to implement live keyword searches and filters for user inputs. We also decided to accept user inputs as drop-down box choices in order to minimize the possibility of errors, which prevented us from being able to display any filtering. Instead of the live map that we had planned to present on the website, we posted our database structure with all its labeled tables and relations instead. Instead of seeing a map with airline locations connected by flights, users are presented with a more rudimentary version of the raw data, and although this could be confusing to those with no database knowledge, it allows generally allows them to see how they, and subsequently our application, are interacting with data.

6. Explain how you think your advanced database programs complement your application.

Our advanced database programs help users view relevant information in a clear and concise format rather than viewing the entire data at once. It also allows users to update data in a much more visually appealing format that is much more convenient for users. For instance, we added a stored procedure to allow users to view airlines and the corresponding average delay time, which may be incredibly useful while selecting which airlines to fly with if one must book a connecting flight.

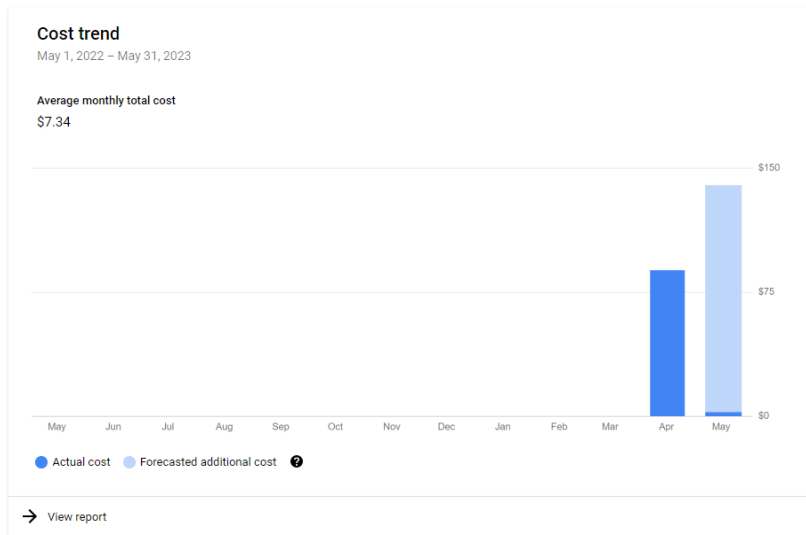
7. **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**

Enya: Figuring how to set up GCP in general was an extremely long process. Though we were provided videos and instruction on what steps to take, we spent much time attempting to debug problems on our own end that we were unable to consult the videos for. This included condensing our data as mentioned below, and we all had to work together to accurately implement the relations we had planned in our proposal into functioning tables.

Rushika: Integrating frontend and backend through example sql queries would be helpful. A lot of additional research and debugging had to be done to figure these things out.

Sriya: The scale of our original dataset proved to be much too large for our application in the initial stages of our project. Originally, the 2015 flights dataset that we used had over 4,000,000 individual rows of data. Trying to load the full range of data onto the platforms that we intended to use for the project, including GCP, was a problem because they did not support such large files of data. We struggled to reduce the dataset as well, since none of our data handling applications, such as Pages or Google Sheets, were able to take in the large dataset for us to clean and manipulate. After many hours of struggling, we ended up having to divide the dataset into around 100 different files with subsets of the data and further filtering out those files so that we were able to capture a representative amount of data for the dataset so that its results wouldn't change but its size would be greatly reduced. With this method, we were able to gather around 15,000 rows of the data to load into our GCP.

Chen: The bill account problem. The legend should teach us how to cancel our GCP account if we do not use it anymore. There is little information I can find from the website. Future teams should be more careful with the way they spend their GCP credits so that the given codes can last the group for the entirety of the project duration.



8. Are there other things that changed comparing the final application with the original proposal?

Another difference between our original proposal and our final application is the scale of our data. Since our original dataset was quite large, we were unable to load all of the data that we wanted onto our application. Thus, we had to condense the dataset to what we felt was still representative of the data without having such a large number of rows.

9. Describe future work that you think, other than the interface, that the application can improve on

One aspect that the application can improve upon is its dataset. Currently, our project relies on a dataset that contains flight, airlines, and airport data from only the year 2015. Involving a wider range of data, such as from the past 5 years up to current years would help our application expand its usability and real-world relevance. For this to work, we would need to collect and clean a large amount of data using different data manipulation techniques in order for the new data to fit our application and its specifications.

In addition, expanding the different data manipulations that users can do from our interface would be a valuable future task to take on for our application. For example, currently, filtering out data with either one or two constraints has to be done in two separate tasks in our application. Integrating these two functions to work as one unit where the user can either input one, two, or more constraints

into the same field so that the filtering process becomes more robust and seamless would improve the usability of our application.

10. Describe the final division of labor and how well you managed teamwork.

Enya: I helped work through the indexing portion in GCP, designing advanced queries with the other members, and testing and implementing the triggers.

Rushika: I helped set up GCP, design relevant and specific SQL queries and meeting the programming requirements of the project.

Sriya: I helped clean our dataset, condense it into a more manageable load of data, and upload it into GCP such that the data matched the schema that we designed. I also helped research and find helpful resources for the project as well as assisted with testing and implementing the stored procedures.

Chen: I am responsible for the coding(include javascript, html code) and maintaining the Github doc file.

Teamwork was well managed within our group members. We met both in-person and through Zoom to discuss our goals for each stage and to divide up the labor. Most of the work was done all together during the meetings, and any extra work was done individually when necessary.