Tokyo Olympics Wiki Project Report

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).

1.Filter Features: We didn't implement the filter feature.But the search bar itself already can help users find any data they can find in the database.

2.Visualization: We initially decided to make several graphs of search results to create better visualization for the users. But we don't have time to implement it.

3.Data export: Initially we decided to provide an interface for users that they can download the research result from the website in a variety of formats. But we didn't implement the function in the last stage.

4.UI Mockup:We initially decided to show all the search results in a single table. But we find It's not practical or readable to put a large amount of data into a single table, instead we split the search result into several tables according to the category of the dataset such as Athletes, teams, Coaches and so on.

5. Additional data on salary and age: Because the original dataset lacks useful information for every athlete. We create extra data about the salary and age of every athlete to make the dataset more meaningful and provide more useful information to the user.

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

In terms of usefulness this application very effectively allows you to inquire about practically any type of data regarding the olympics. It is very easy to filter data using multiple attributes. For example, you can search for Athletes from the United States in Gymnastics with a first name starting with 'A' very easily. Additionally, it is just as easy to modify the data in the database. One thing our application failed to do is extend the usefulness of the database into a useful visual. For example, one thing that could have been done is to create a chart displaying each country and the number of gold medals achieved in the Olympics that year.

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

In the beginning, we used the original data from Kaggle for the 2021 Olympics in Tokyo. There are several tables in the dataset which are Athletes,Genders,Medals,Coaches and Teams.When we try to create an ER/UML diagram from those dataset. We were met with several problems.

Genders: Our database focuses on the information of every individual athlete, so it's useless to put a extra gender dataset into the diagram which includes the number of female and male numbers in each sport.Instead, we modify the Genders dataset into a Sport dataset which includes the average salary and average age of athletes in different sports. As a result, the information of salary and age should be more practical and useful than provide a gender to the user

Athletes: Since we change the number content of Genders into Sport. We also add an extra row column into the athletes dataset which includes age and salary of each athlete.As a result, It's meaningful to connect Sport dataset and Athletes dataset so that user can easily compare the value of each athlete to the average value to the average value

Medals: We also make the change to the Medals dataset. The Medals dataset only counts the medals by country. But our website wants to focus more on the individual so we modify the dataset into counting how many medals each athlete get during the olympics to make the dataset more realistic and useful

4.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 remove the NOC column from the Coaches table in the ER diagram. Because we find out that NOC is a column shared by many tables. So we don't have to record the same data of NOC in every table which causes the redundant data in the database.We think we can also try to remove the Event and Discipline column from the coaches as well, because the coaches table is connected to the Teams table by primary key Coa_Name and Teams already have the column of Event and Discipline.

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

We added the functionality of editing, adding or deleting data from the database. This makes it extremely useful to update the existing database without causing any issues. We added this for the purpose of being useful, and for the purpose of being able to control any updates on the database using a trigger. The trigger is called whenever an update on the database occurs. For example, if a user would like to insert an athlete from the country "Chicago," the trigger would not allow this because Chicago is not a country.

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

Our advanced database programs play a crucial role in the overall functionality and user experience of our search engine website, which focuses on providing comprehensive

information about athletes, coaches, and teams from the Tokyo Olympics. These programs complement our application in multiple ways:

Efficient Data Management: Our advanced SQL database enables us to efficiently store, organize, and manage the large volume of data from various tables related to the Tokyo Olympics. This ensures that our search engine can access and retrieve relevant information quickly and accurately.

Complex Query Handling: The database programs we use allow us to handle complex queries involving multiple tables and conditions. This capability allows our search engine to deliver precise and relevant results to users, even when they are searching for information that spans across multiple categories or data points. Several tables are connected together

Scalability: As our search engine continues to grow and we add more data sources, our advanced database programs enable us to easily scale our application without compromising performance. This ensures that we can continue to provide a reliable and efficient search experience to our users. Users can easily delete,modify and insert data into the database.

Advanced SQL Queries: The use of advanced SQL Queries ensures that users can implement more complex and advanced search through a button on our website. Which is much more convenient than directly searching.

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.

Yifeng Ni: I meet the problem of not enough data to build up Schema.I need to modify the original data, So I create several fake columns to several tables in order to create a practical source of data and a useful ER/UML application.

Eduardo Martinez: In the original design of the schema, we were met with the challenge of creating tables from relationships between data. This was hard specifically because of the gender column conflicting from table to table, making it difficult to construct relationships. We wanted to keep the gender attribute in the first place, because we believe it is an important data to have in the database. However, our only choice was to remove the column in its entirety to address these issues.

Siddhant Sharma: I ran into a problem trying to get the CRUD functionalities in the search results page to update the tables and still display all the remaining data. I fixed that in the server.js file (the main js file) by have the queries for displaying the data and the search result of the user (from front-end) all in global variables and then passing it along with the specific query for the CRUD functionalities so that we have the data to populate the front end each time a CRUD is performed and still maintain the front-end UI with the related table.

Rohan Vij: I focused a lot of my time writing the advanced queries and stored procedures. This involved a lot of syntax errors and table joining errors that took a lot of time to fix. Some issues were with the return types and the limited amount that each query could return. There were also limits on the tables and ability to join based on keys. Fixing these issues was especially harder when all the code was being done on the server and could not be run locally.

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

One main change from the original proposal was the interface. We essentially changed it to be more user friendly, and more pleasing to the eye. The locations of the search bar, and where you edit the database changed.

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

1. Machine Learning Integration: Implement machine learning algorithms to better understand user preferences, personalize content, and provide more accurate search results. This can help deliver a more tailored user experience.

2. Mobile App Development: Develop a mobile app version of the search engine to provide users with a seamless experience on their smartphones and tablets. This will make it easier for users to access information on-the-go.

3. Event Prediction and Analysis: Use historical data and machine learning techniques to predict event outcomes and analyze trends. Offer insights to users, such as which athletes are expected to perform well or which countries may dominate specific events.

4. Security: Implement strong security measures to protect your SQL database from unauthorized access and potential threats. Use encryption, access controls, and SQL injection prevention techniques to safeguard your data.

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

Yifeng Ni: Mainly works on Conceptual and logical database design. Modifies the data from the original dataset to a more reasonable dataset to suit our database design and completes part of the report in every stage.
Eduardo Martinez: Mainly worked with Yifeng on the Conceptual and Logical design of the Database. Helped construct the relationships and tables for the data, and the application UI mockup.
Siddhant Sharma: Mainly worked on the front-end of the website to get all the necessary CRUD functionalities, Stored procedure and advanced queries working along with connecting with the database hosted on GCP for a successful webpage.

Rohan Vij: handled most of the back-end with the SQL queries including the advanced procedures, queries, creating the tables and establishing the database. Completed all the queries in SQL and dealt with all the GCP parts of the database.

Video Link:
https://drive.google.com/file/d/15EAciMroCsakuiaVrfi3NFp3kdW77s__/view?usp=sharing