

1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
 - a. Our final project followed the core functionalities that we established at the beginning. Therefore, our final project is not very different from our original proposal. The only changes we made were to efficiently demonstrate the functionalities. We also were unable to collect perfect data (multiple Spotify songs had null genre), and we had to change the display accordingly to showcase what we could infer from the given data. Finally, we did not implement passwords or user profiles for the users.
2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
 - a. Our application achieved all of its objectives of being able to perform CRUD operations, Advanced queries as well as stored procedures and triggers. One thing we could improve is the UI, since we didn't apply CSS to style our application as shown in the original designs. Another aspect which we initially aimed to accomplish was storing user profiles, as mentioned previously, which would have made the application more useful since users would be able to look back upon their saved playlists. However, this was a side feature and the main functionality of the app works smoothly and can hopefully provide value to music lovers.
3. Discuss if you changed the schema or source of the data for your application
 - a. We didn't. We used the Spotify API to get data for the tables.
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?
 - a. For our ER diagram, we reduced the number of tables and added 3 new tables to better connect the databases. Originally, we intended on using multiple keys to connect the tables with each other, however after learning about functional dependencies, we reduced the number of columns needed by removing redundant keys, and created a a better/ more efficient ER diagram (submitted in database design). Furthermore, when implementing the website, we understood that certain tables are only called based on other tables, therefore, we removed some flexibility but increased efficiency by directly connecting certain tables. Our Final design was better as it was efficient and practical.
5. Discuss what functionalities you added or removed. Why?
 - a. We added the ability to view songs added to a playlist on a scatterplot, based on the album popularity, song popularity and release date. This was useful when looking at the correlation between popularity and time, and for

displaying all the new songs added for that genre. We also added the ability to see the the playlists with above avg play times. This was useful to compare the different playlists. A functionality we removed was the delete playlist functionality. We realized that we would be demonstrating the deletions of songs, and we wanted to keep our advanced query (above avg playlist time) output consistent for the demo. Therefore, we chose to not implement this functionality.

6. Explain how you think your advanced database programs complement your application.
 - a. Our advanced queries helped us create interesting visualizations and analyses of Spotify data in addition to our playlist-generating feature. For example, in our stored procedure, we wrote queries that allowed users to visualize the different eras of modern music reflected by the songs in Spotify's database. The results of our query also revealed the fact that many older songs lack accurate release dates and therefore display their upload date instead. Many classical songs were listed as contemporary because of this.
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.
 - a. Sarah: One technical challenge that we encountered was getting our buttons on our website to route to different pages and change what the webpage looked like. We ended up resolving this issue by utilizing sessions to redirect without refreshing the main page, which is how every other update works.
 - b. Nayonika: A major challenge we faced occurred when we were implementing our stored procedure. Our second advanced query was meant to be associated with a second cursor, but this ended up leading to a variety of errors. We ended up realizing that a second cursor was unnecessary since we didn't actually need control statements to produce data relevant to our goal, so we ended up rewriting the query to directly select from the table that stored the values stored by the first cursor.
 - c. Sreyas: A challenge I faced was deadlock when writing the trigger. My event and action were on the same table because of which I could not use my trigger and it caused a deadlock.
 - d. Chirag: One technical challenge we encountered was getting data from the database when the query involved modulo's "%" or date time. For the date time query, we accepted a year value, converted it into a string based on the date time format and passed it. For modulo, we made a separate

string with it and only passed it while executing the query rather than formatting the query string with it.

8. Are there other things that changed comparing the final application with the original proposal?
 - a. The biggest difference between our final application and proposal is the user login aspect that we planned to implement. Originally, we planned on having users create accounts so that their past queries could be stored and their playlists could be connected to an actual Spotify account. If we were to make adjustments to the final application, this would definitely be the biggest improvement to the project.
9. Describe future work that you think, other than the interface, that the application can improve on
 - a. Other than the interface, our application could improve by implementing options to search by song instead of artist, more options for what to base playlists off of, or even real-time updates for the search that is already implemented.
10. Describe the final division of labor and how well you managed teamwork.
 - a. Overall, the work between the team members felt fairly evenly distributed. While each of us had our own focus point, all of our work came together very evenly and in a timely manner. Sarah worked mainly on the front end, Chirag worked on pulling the data and on routing, and Nayonika and Sreyas focused on building the backend, middleware, trigger, and stored procedure, but everyone did put some effort into each section of the project.