INST 327

Section 0102

By: Avram Baicu, Yohannes Gebremariam, Enrique Calle-Matute

Prof. Vedat Diker, Pamela Duffy

Date: 05/10/2022

Team number:1

"Music Friend" Final Report

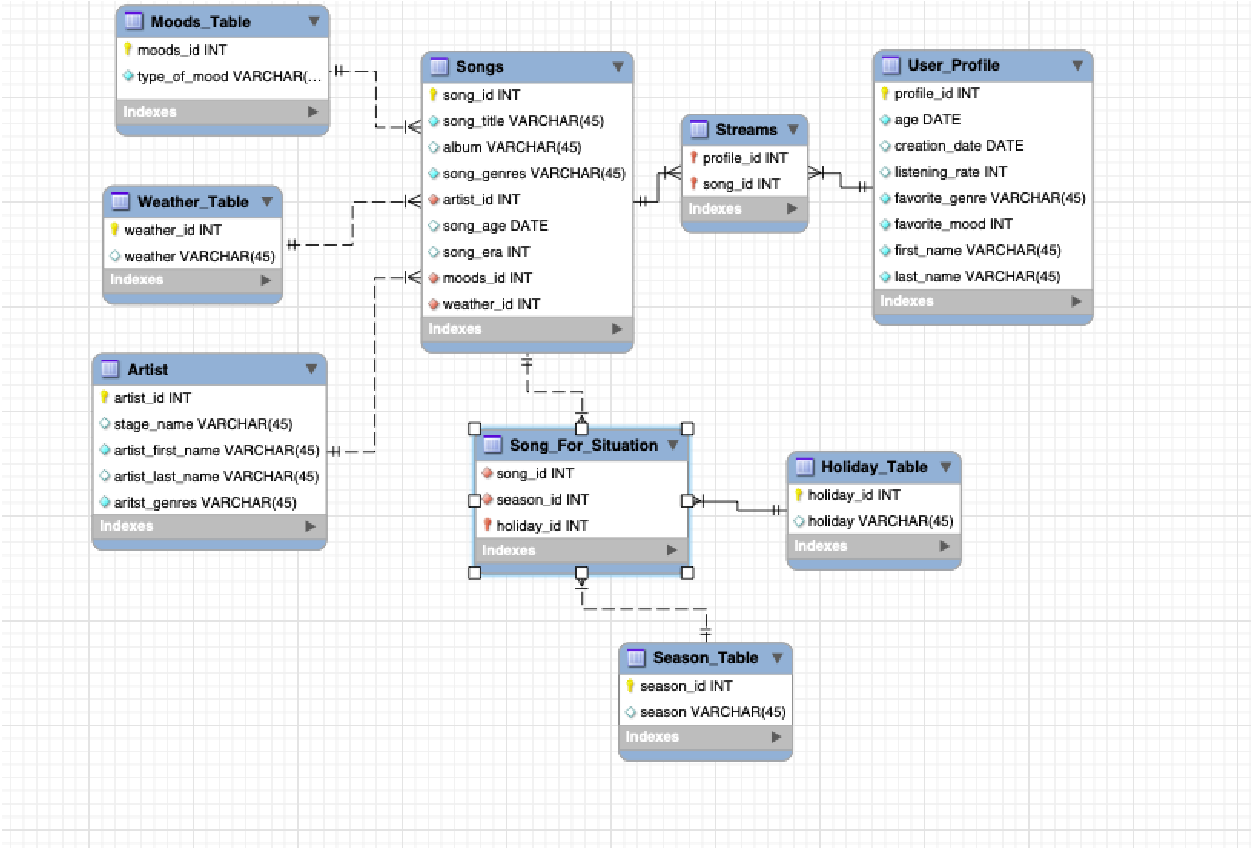
Introduction

For our project, "Music Friend", we choose to develop a database that will play music based on different moods. The various situations that we were able to come up with are personal moods, time of day, weather, seasons, and holidays. This will provide the user with a suggestion and play a similar type of music for them. There are a variety of songs that we were capable of coming up with along with some that had some diverse languages.

We decided on this topic because we wanted to come up with something that is not utilized by bigger organizations. Many people don't have the best experience when it amounts to listening to music and we believe that we can deliver more. All the popular streaming/playing platforms use a random music generator and sometimes someone might want to play the music that only fits the same mood; essentially, we will create a DJ based on how someone feels at the moment or a number of songs for a certain situation. By creating the mood matcher, people will be capable of enjoying the music they want to without having to skip as many times. This will increase how users experience listening to music a lot.

As for changes since the beginning, we've tried covering various songs, but we needed to cut down some of the moods and types to be able to submit a sound project. We were cut down to about half of the manpower as a group. We needed to take out the number of times someone listened to a song. We had to shorten many of the tables like Columbus day for holidays and delete tables like artist genre since it is redundant.

Database Description



- Logical Design (include a PNG of your ERD in this section)

Our logical design consists of our main database (songs) and other tables that are linked or joined. We had to split some of the data apart to be able to develop this database. We wouldn't be capable of adding different music and also have a user profile in the same database. We need to split the tables and link them together because some of the data from the user profile will be sensitive. It will also make it easier to produce data with the songs. As for moods and weather, they could be naturally included since they are different categories of mood. The artist table needs to be split as well because the database will be unreadable if it was together with songs. We didn't need to link any information about the artist directly to the songs which is the reason we didn't construct a linked list. Last, the songs for the situation table represent a similar category, and seasons and holidays overlap which can be useful when producing a database. We needed to make it a linked list because we had to reference both the songs table and link the holiday or season.

Physical Database

- ○ For the AmountOfSongsJohnAndMary query we wanted to link the user to the songs. We filtered all the favorite genres to only show the ones that have the same values. This allowed us to choose the amount of favorite genres and also create a subquery. We then chose to only show users 1 and 3 and ordered them from least to greatest.
- ○ For the **NumOfJohnsSongsForSunnySeason** we linked two tables, and filtered it so it only showed the seasons with the season_id that are less than 4, which singled out summer, and spring and none for the sunny seasons. When importing data, it was able to run smoothly, and our tables coincided very well with the data they were filled with.
- ○ Spotify gives us access to a lot of information about any song or artist out there. This ranges from features describing the Mood of the audio, such as the playlists “party cave”, “sad hour”, and “energy”, through to the features describing the popularity of the artist and song. We also got more advanced information from this apple music, such as the release date, if we want to do a more advanced analysis of the data.

- **Sample Data**
  - ○ We tried to have as much correct data for our sample data as possible. We got our data from our music platforms like Apple music. Some of the songs were from Spotify as well. We created hypothetical data such as fake names for the user profiles that were derived from the Marvel Universe. We also included more names that show that more could be added on as well. When creating the songs, we made sure to include the right dates that the songs were created so that the data would be more accurate. Everything looks up to par for us and we believe that it doesn’t have any errors with our data.
- **Views / Queries**

- What the **AmountOfFavSongsJohnAndMary** query is trying to do is taking a user and songs and it is finding out the number of favorite songs there are for users. Then it is taking only the users with the user profile id that is 1 and 3 and displaying in the order of who has the least amount of favorite songs to the largest amount. It also creates a view that gets replaced each time it is run.This query has met all the requirements.
- **NumOfJohnsSongsForSunnySeason** we wanted to link the table Song _For_Situation and the Season_Table. We were able to cover., SELECT, JOINS, Filtering and Aggregation.
- For our last query, we used artist name to filter out songs by weather condition, we have filtered out songs that are specifically for winter(snow) weather

Views	Select (5)	Joins (4)	Filtering (3)	Aggregation (2)	linking (1)	subquery (1)
AmountOfSongsJohnAndMary(Avram)	x	x	x	x	x	x
NumOfJohnsSongsForSunnySeason(Enrique)	x	x	x	x		
ArtistSongSnowyWeather(Yohannes)	x	x	x			

Changes from Original Design

There has been much change from the original design. The initial design was overall thought over too much with us not realizing that we had to make more significant connections to each other. The modern design has nine tables in total which is exactly the same number of tables as the original design. When trying to work with the original design, we realized we tried to construct too many irrelevant connections, which ended up confusing us as a group. We had to ultimately come to the realization that our original design was just overly complicated to execute for the number of members in our group. We had to split some tables, delete some moods, and delete some tables as well. This way we can develop our database work within the given time so that it also works in general. One of the significant changes we made came by attempting to implement the number of songs that would match the exercise the user may be looking for. As well as the type of music, certain communities listen to. The concept of being able to be inclusive like that sounds beautiful in theory, but in the end, it would take too much data to appropriately represent this table. Both of these tables that were in the concept were eventually scrapped. Our final tables consisted of:

- User_Profile
- Streams
- Songs
- Moods_Table
- Weather_Table
- Artist
- Songs_For_Situation
- Holiday_Table
- Seasons_Table

When it came to mutually deciding which entities/tables not to include in the final submission of this database, it still reflected our project proposal, by including none that we originally planned to. As mentioned earlier, one of the more significant changes we made came from the data within the tables. We decided to be frugal with the data that inhabited the tables, and that, in turn, helped us with the flow and efficiency of the database.

Database Ethics Consideration

When it came to our database ethics consideration, we attempted to not deviate from the original ideas that we mentioned in our initial project proposal. We tried making this a database to be as inclusive as possible. We ended up making this database accessible for all to use, by eliminating tables that may attempt to divide communities and place them into distinct categories. By making the database simple, we could let anyone use it, especially by making it very time and seasonal oriented, including Holidays we can accommodate a larger number of potential users. There, of course, are always concerns when it comes to basing a database on moods as well, but we really do believe that the contents of the database have more benefits that could help rather than encourage any harm. We included songs for as many diverse audiences as we could. We ended up adding as many genres as we can in the given time and included a couple of songs with diverse languages as well.

Another concern that we initially considered at the time of the proposal was considered fair use. By making this only a database that presents the name of titles, we believe we have avoided any legal issues when it comes to fair use. The only concern we would have with fair use would be in the hypothetical scenario that we would want to advertise this database with a song with the database for gain, which is unintended. When it came to user privacy, we did not have a concern with the data we were collecting because our user_profile table did not ask for too much detailed information. We merely collect their first and last name, age, date of creation, favorite mood, their favorite genre, and as well record their listening rate. We also separated the user_profile table from the rest of the database so that their information would be harder to access.

Lessons Learned

When it came to this project, many lessons were learned. One of them is the number of hours it can take to create a database. This includes its functionality, its purpose, and the amount of dynamic thinking in order for it to flow with efficiency. We first had to make sure the ERD is up to standard, and we needed to create all the types of columns in a way that our data will be able to use it. After finishing the ERD, we needed to create a working excel spreadsheet that works with the data given in our project. We had to delete multiple entries for each row and split each data so that everything works. Later we learned how to use our data to give us what we want. The fact that each connection has to be thought out in order to find sufficient need for it, and as well avoid redundancy is incredible.

As a team, we've faced some great struggles. We learned that in order to complete a project, we need to all work together and whenever someone decides not to be involved, the next person has to step up and do their work. We came to realize that sometimes you will have your ego when pursuing a concept, and go with something more feasible. This can go both ways; one has to swallow their ego and do the grunt work in order for everything else to work out as efficiently. The creation of this database did take a lot of organization and care, something that will have to be noted for many more projects to come. We learned that each team member has unique talents, knowledge, and abilities. When all members worked together, we were able to draw on our combined experience, expertise, and talents to complete this project. We believe that collaborating with team members or even different teams should be viewed as a learning opportunity.

Potential Future Work

Music is now largely delivered digitally, a particular management system is required to arrange enormous volumes of material. We can begin by adding more data to our database. Adding more data will allow us to have more concrete examples of our information which will make people enjoy the database even more. Adding back our old tables would help make the data more friendly to the songs they want to listen to. We need to also link the tables so that they would generate a playlist for a user on their specific platform. Most people have a large music collection, and they need a good music library organizer. We often download music from services like iTunes and Spotify. We all know how important it is to have a personal music collection, preferably the finest music organizer that can help us discover our favorite songs quickly. We also need to diversify our songs even further and make our user's information safer.