Aria

*India Davis*  
*College of Sciences*  
*Computer Science*

Montgomery, AL  
[idavis4@aum.edu](mailto:idavis4@aum.edu)

Kayla Quarles  
*College of Sciences*

*Computer Science*

Montgomery, AL

[kquarle1@aum.edu](mailto:kquarle1@aum.edu)

*Abstract*— In today’s digital era, music streaming has become deeply integrated into everyday life. Aria enables users to create, customize, and manage playlists that reflect their current mood. It allows users to add or remove songs, oversee their music libraries, and tailor their experience based on personal preferences. The app is powered by a MySQL database, drawing inspiration from the data architectures of major music platforms. An examination of existing services reveals that Apple Music utilizes FoundationDB and CloudKit for backend operations, while Spotify transitioned from PostgreSQL to Apache Cassandra to support its extensive data needs, particularly for time-based data tracking. YouTube Music relies on MySQL in conjunction with Vitess to scale effectively, and it incorporates caching and disaster recovery tools to maintain performance. Pandora stands out with its Music Genome Project, which uses detailed musical attributes analyzed by experts to offer personalized listening experiences. Aria’s database design includes structured tables for users, artists, albums, songs, genres, playlists, and liked songs. These tables are connected through foreign keys to ensure efficient data organization and relationship mapping. Developing Aria provided valuable insights into database planning, design, and user-centric structuring. Future enhancements aim to introduce song recommendations, playlist sharing and liking, and listening history tracking. Overall, the project strengthened practical SQL skills and highlighted the significance of thoughtful database architecture.

Keywords—music streaming, user experience, database design, database tables, MySQL database

# Introduction

In today’s generation, Music streaming has had an impact on all our lives, with users wanting to discover a new genre in music or wants to upgrade their playlist with some new songs from their favorite genre. Our app, Aria, will allow the user to create the perfect playlist to suit their mood. Aria will let our users create, delete, and update their playlist. The users can also add songs to playlists that they already love. They will also manage their account with all their songs, albums, and playlists.

# Literature Review

Many music applications we use today use database management to allow their users to store, manage, create, and delete songs into their music library and playlist.

## Apple’s data base applications

Apple uses many applications to manage Apple music. Apple Music uses the application “Foundation DB” for the backend development to manage the wide number of songs and created playlist in their application to satisfy their user’s needs. They also use Cloud Kit, which is their own database service that is for apple applications [1].

## Spotify’s datbase applications

Over the years, Spotify has evolved which database application they have used. The first database system that Spotify uses PostgreSQL but later uses Apache Cassandra to manage their data [2]. Spotify’s databases provide a fast and effective way for users to interact with Spotify. Cassandra allows Spotify to store copious amounts of data by allowing the developers to create wide rows with multiple columns acting as one key. Cassandra also stores “time-series” data, which is best for knowing how long a user must pay for their Spotify subscription [2].

## YouTube Music’s Databasae Applications

YouTube uses a sophisticated backend to manage its massive video storage and streaming needs. The platform’s infrastructure is built around MySQL, scaled horizontally using Vitess for handling high traffic. Videos are processed with batch jobs that handle encoding, thumbnails, and metadata, while advanced compression (VP9 & H.264) ensures efficient storage. YouTube also employs caching with Memcache and node coordination via Zookeeper. To manage the growing demands, YouTube implemented database sharding and master-slave replication and added disaster recovery measures to ensure uptime and data safety across global data centers [3].by

## Pandora Music Genome Project

The Music Genome Project (MGP) is the foundation of Pandora's personalized radio experience. It involves the analysis of songs using up to 400 musical characteristics by trained musicologists with a deep understanding of music theory. The project spans all genres and eras, from classical to contemporary music. The MGP is continually updated and helps Pandora tailor music stations based on individual listener preferences, ensuring a personalized listening experience. The project uses rigorous methods, including redundant analysis and quality control, without relying on automated processes like machine-listening [4].

Our mission for integrating these ideas for Aria in MySQL by creating individual tables for the user information, user playlist, their songs on they have added to their playlist, individual songs and their artist and what albums they are from. Aria will track users liked songs for a user-friendly experience so they can easily access past songs that are outside of their playlist.

# Methodology

For Aria, we have implemented our work into MySQL as our relation database management system. Our database consists of these tables. user, artist, album, song, genre, playlist and liked songs.

## A. User table

The users table stores the user’s first and last name. The table also stores the email they used to sign up and the password they created. We chose to make the last name not required for the individuals who chose to only display and store their first name.

## B. Artist table

The artist table contains the artist id that is created when a new artist is added into the database and to uniquely identify them. The table also stores the artist's first and last name. We chose to make the last name not required since there are many artists who only have a first name.

## C. Album table

The album table stored the unique album id for identification of each album and for easy access. The table also stores the title of the album, the foreign key reference of the artist id to reference that artist who released the album. The release date of the album.

## D. Song table

The song table stores the information of each song that is added into the Aria database. It consists of the song id, song name, and the foreign keys of album id, genre id, and the playlist id to connect the album, genre, and playlist table. The table also contains the song length and its own release date.

## E. Genre table

The genre table stores the genre id of each different genre and the name of the genre

## F. Playlist Table

The playlist table contains the unique playlist id, the name of playlist created by the user, and the foreign key of the user id to reference to the user table.

## G. Liked songs table

The liked songs table stores the foreign keys user id and song id to reference which user liked a certain song. It also has the like song Boolean variable identify which songs that user liked or did not like.

By implementing these tables in Aria Music database can efficiently manage and organize the relationships and attributes between the user, artist and their album and song, and the user’s playlists offering an across-the-board structure for a music app that allows for a user personalized experience and dynamic content management.

# Conclusiion

In this project, we built the aria music data to structure music, artists, users, and playlist. This project has helped us learn how important it is to keep information clear, connected and simple for the user to understand. While our database works adequately for basic needs there is always work for improvement. Our future research plan for this project is adding song recommendations, allowing users to share playlist and like other user’s playlist, and tracking the listening history. Creating this project has been a learning experience for us because of the hands-on experience with building a database from scratch. This project tests our knowledge of SQL and learned how important it’s to plan out the connection and steps it takes to create certain steps in the project. The project has helped strengthen our technical skill and gave us real world experience with working with a database.

# References

1. “Cloud -iCloud” Apple Developer, <https://developer.apple.com/icloud/cloudkit/> . (accessed Mar. 31, 2025).
2. B. Edstrom, “In praise of ‘boring’ technology,” Spotify Engineering,Feb. 25, 2013. <https://engineering.atspotify.com/2013/02/in-praise-of-boring-technology/> (accessed Apr 02, 2025).
3. J. Wallis, “What is Apache Cassandra & Why DOes Spotify Use It?,” intuji, Jun. 14, 2023. <https://intuji.com/what-is-apache-cassandra-why-spotify-uses-it/> (accessed Apr. 02, 2025).
4. Shivang, “YouTube database – How does it store so many videos without running out of storage space?,” Scaleyourapp, Dec. 15, 2019. <https://scaleyourapp.com/youtube-database-how-does-it-store-so-many-videos-without-running-out-of-storage-space/#:~:text=YouTube%27s%20backend%20microservices%20are%20written,for%20horizontal%20scaling%20of%20MySQL>. (accessed Apr 02, 2025).
5. Pandora, “About The Music Genome Project,” Pandora internet radio. <https://www.pandora.com/corporate/mgp.shtml> (accessed Apr 02, 2025).

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**