**Music Co.**

Kennesaw State University, Spring 2022

IS3280 – Data Management

Instructor Mark Sims

May 2, 2022

Contents

[**Project** 3](#_Toc102398515)

[Members 3](#_Toc102398516)

[**Executive Summary** 3](#_Toc102398517)

[**Business Rules** 3](#_Toc102398518)

[**ER Diagram** 5](#_Toc102398519)

[**Data Dictionary** 5](#_Toc102398520)

[**Normalization** 6](#_Toc102398521)

[All Attributes 6](#_Toc102398522)

[Functional Dependencies 6](#_Toc102398523)

[Normalized Tables 7](#_Toc102398524)

[**Tables** 7](#_Toc102398525)

[**Inserts** 8](#_Toc102398526)

[**Queries** 9](#_Toc102398527)

[**Forms and Reports** 11](#_Toc102398528)

[**Reflections** 13](#_Toc102398529)

[Sam 13](#_Toc102398530)

[Fletcher 14](#_Toc102398531)

[**References** 15](#_Toc102398532)

[**Additional Resources** 16](#_Toc102398533)

# **Project**

The topic of this project is using database management to organize a music inventory and create curated playlists for music lovers. The database will be created using Access.

## Members

* Sam Olusanya: solusany@students.kennesaw.edu
* Fletcher West: gwest13@students.kennesaw.edu

# **Executive Summary**

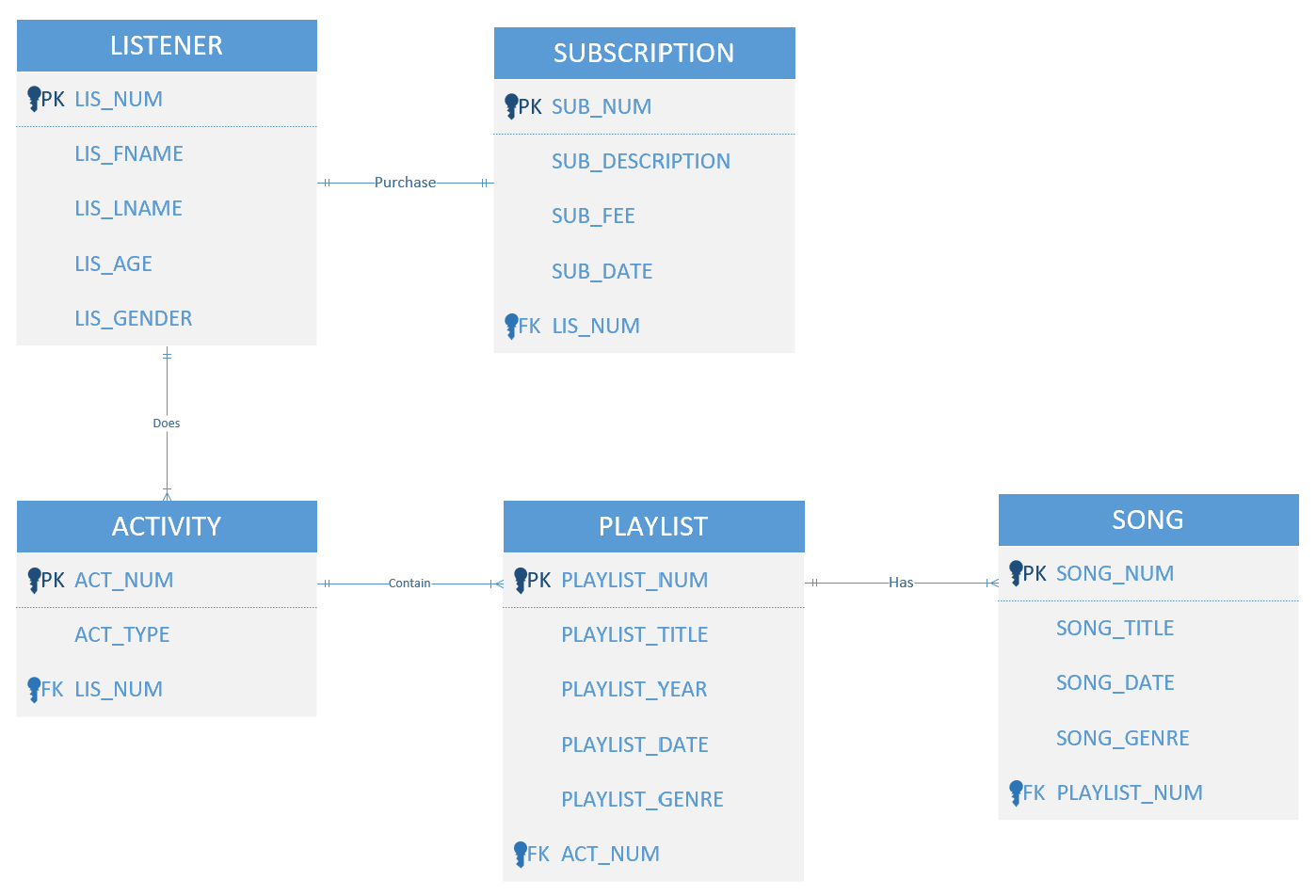
Music Co. is an organization that creates curated music playlists. Presently, Music Co. has attracted a wide variety of listeners with diverse tastes in music spanning several genres. This increase in listeners has necessitated an update to the organization’s record-keeping technology. Specifically, Music Co. requires a database to gather and retain general membership information regarding their musical interests. Additionally, Music Co. needs to organize playlists according to songs, and activities being performed.

Music Co. staff will update the membership list, forms, and membership directory. Potential members will be permitted to enter their information. Staff will be responsible for organizing all data involving members, music, and playlists.

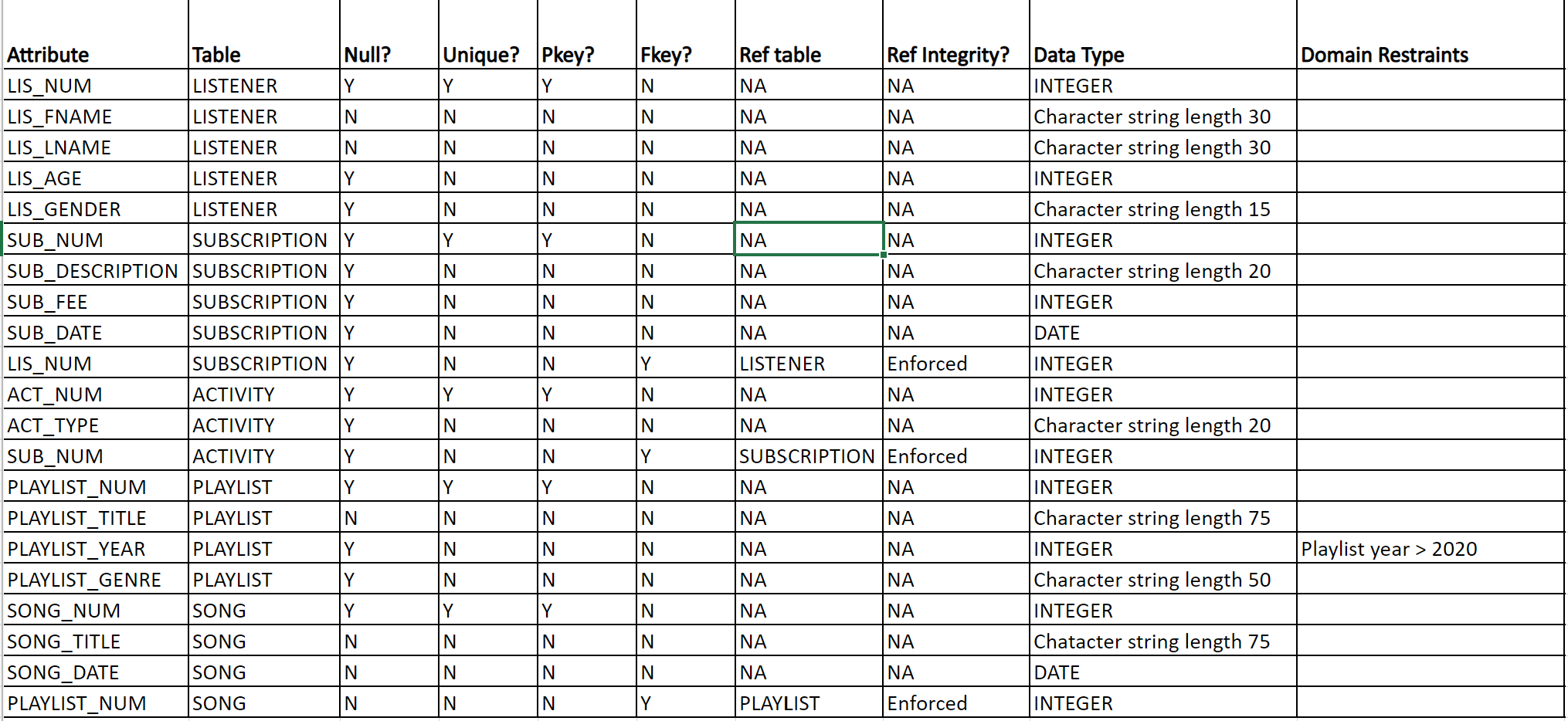
# **Business Rules**

* Music Co. retains general information about its listeners which includes:
  + A unique number assigned to each listener
  + Basic demographic information
  + Music interests - Includes 10 genres
  + Types of activities performed
* Each playlist will be identified by a unique number
* Each playlist may have many songs.
* Music Co. will create playlists and develop a listing for each one. Activities map a listener to a playlist. An activity may contain many playlists and many activities may be done by a listener.
* A subscription can be purchased by a listener and a listener may purchase one subscription
* Records will include members, activities, dates, songs, titles, genres, subscriptions, fees, and playlists.

# **ER Diagram**



# **Data Dictionary**



# **Normalization**

## All Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| LIS\_NUM | LIS\_FNAME | LIS\_LNAME | LIS\_AGE |
| LIS\_GENDER | SUB\_NUM | SUB\_DESCRIPTION | SUB\_FEE |
| SUB\_DATE | ACT\_NUM | ACT\_TYPE | PLAYLIST\_NUM |
| PLAYLIST\_TITLE | PLAYLIST\_YEAR | PLAYLIST\_GENRE | SONG\_NUM |
| SONG\_TITLE | SONG\_DATE |

## Functional Dependencies

**Rule**: A listener number is assigned to each listener to uniquely identify the listener along with demographic information

**LIS\_NUM → LIS\_FNAME, LIS\_LNAME, LIS\_AGE, LIS\_GENDER**

**Rule**: Listeners purchase a subscription

**LIS\_NUM → Subscription**

**Rule**: A subscription number is assigned to uniquely identify each subscription type

**SUB\_NUM → SUB\_DESCRIPTION, SUB\_FEE, SUB\_DATE**

**Rule**: An activity number is assigned to uniquely identify each type of activity

**ACT\_NUM → ACT\_TYPE**

**Rule**: Activities contain playlists

**ACT\_NUM → Playlist title**

**Rule**: A playlist number is assigned to uniquely identify each playlist and other information

**PLAYLIST\_NUM → PLAYLIST\_TTITLE, PLAYLIST\_YEAR, PLAYLIST\_GENRE**

**Rule**: A playlist may have many songs

**PLAYLIST\_NUM → Song titles**

**Rule**: A song number is assigned to uniquely identify each song and its release date

**SONG\_NUM → SONG\_TITLE, SONG DATE**

## Normalized Tables

LISTENER(LIS\_NUM, LIS\_FNAME, LIS\_LNAME, LIS\_AGE, LIS\_GENDER)

SUBSCRIPTION(SUB\_NUM, SUB\_DESCRIPTION, SUB\_FEE, *LISTENER\_NUM*)

ACTIVITY(ACT\_NUM, ACT\_TYPE, *LISTENER\_NUM*)

PLAYLIST(PLAYLIST\_NUM, PLAYLIST\_TITLE, PLAYLIST\_YEAR, PLAYLIST\_GENRE, *ACT\_NUM*)

SONG(SONG\_NUM, SONG\_TITLE, SONG\_DATE, *PLAYLIST\_NUM*)

The data model is in 3NF. Each table is identified by a unique primary key. Each non-keyed attribute is dependent on the whole key. Knowing the SUB\_NUM lets us know the listener. Each primary key is a determinate for all attributes included in the table. For example, SUB\_NUM determines SUB\_DESCRIPTION, and LIS\_NUM determines LIS\_FNAME so if we know the LIS\_NUM, we can identify the subscription they purchased.

# **Tables**

1. CREATE TABLE LISTENER (

LIS\_NUM INT PRIMARY KEY,

LIS\_FNAME VARCHAR(30) NOT NULL,

LIS\_LNAME VARCHAR(30) NOT NULL,

LIS\_AGE INT,

LIS\_GENDER VARCHAR(15));

1. CREATE TABLE SUBSCRIPTION (

SUB\_NUM INT PRIMARY KEY,

SUB\_DESCRIPTION VARCHAR(20),

SUB\_FEE INT,

SUB\_DATE DATE,

LIS\_NUM INT,

FOREIGN KEY (LIS\_NUM) REFERENCES LISTENER(LIS\_NUM));

1. CREATE TABLE ACTIVITY (

ACT\_NUM INT PRIMARY KEY,

ACT\_TYPE VARCHAR(20),

SUB\_NUM INT,

FOREIGN KEY (LIS\_NUM) REFERENCES LISTENER(LIS\_NUM));

1. CREATE TABLE PLAYLIST (

PLAYLIST\_NUM INT PRIMARY KEY,

PLAYLIST\_TITLE VARCHAR(75) NOT NULL,

PLAYLIST\_YEAR INT CHECK (PLAYLIST\_YEAR > 2020),

PLAYLIST\_GENRE VARCHAR(50))

FOREIGN KEY (ACT\_NUM) REFERENCES

ACTIVITY(ACT\_NUM));

1. CREATE TABLE SONG (

SONG\_NUM INT PRIMARY KEY,

SONG\_TITLE VARCHAR(75) NOT NULL,

SONG\_DATE DATE,

PLAYLIST\_NUM INT,

FOREIGN KEY (PLAYLIST\_NUM) REFERENCES

PLAYLIST(PLAYLIST\_NUM));

# **Inserts**

1. INSERT INTO LISTENER VALUES (101, 'JOSHUA', 'JACKSON', '20', 'MALE');
2. INSERT INTO LISTENER VALUES (102, 'CINTHIA', 'CHO', '19', 'FEMALE');
3. INSERT INTO LISTENER VALUES (103, 'LEXY', 'VALDEZ', '23', 'FEMALE');
4. INSERT INTO LISTENER VALUES (104, 'ADAM', 'BRANSON', '27', 'MALE');
5. INSERT INTO LISTENER VALUES (105, 'MARSHA', 'DOVE', '35', 'FEMALE');
6. INSERT INTO LISTENER VALUES (106, 'FELIX', 'PETERSON', '22', 'MALE');
7. INSERT INTO LISTENER VALUES (107, 'SARAH', 'VALAZQUEZ', '56', 'FEMALE');
8. INSERT INTO LISTENER VALUES (108, 'XAVIER', 'WALTON', '25', 'MALE');
9. INSERT INTO LISTENER VALUES (119, 'SHAWN', 'TILLS', '32', 'MALE');
10. INSERT INTO LISTENER VALUES (110, 'AMY', 'ZEN', '29', 'FEMALE');
11. INSERT INTO SUBSCRIPTION VALUES (1001, 'Basic', '2', '2021-01-01', 102);
12. INSERT INTO SUBSCRIPTION VALUES (1002, 'Regular', '3', '2021-01-01', 103);
13. INSERT INTO SUBSCRIPTION VALUES (1003, 'Premium', '5', '2021-01-02', 101);
14. INSERT INTO ACTIVITY VALUES (1, 'Study', 1001);
15. INSERT INTO ACTIVITY VALUES (2, 'Party', 1003);
16. INSERT INTO ACTIVITY VALUES (3, 'Workout', 1001);
17. INSERT INTO PLAYLIST VALUES (1510, 'Playlist 1', 2021, 'ROCK');
18. INSERT INTO PLAYLIST VALUES (1511, 'Playlist 2', 2021, 'SOUL');
19. INSERT INTO PLAYLIST VALUES (1512, 'Playlist 3', 2021, 'HIP HOP/RAP');
20. INSERT INTO PLAYLIST VALUES (1513, 'Playlist 4', 2021, 'POP');
21. INSERT INTO PLAYLIST VALUES (1514, 'Playlist 5', 2021, 'AFROPOP');
22. INSERT INTO PLAYLIST VALUES (1515, 'Playlist 6', 2021, 'LATIN-POP');
23. INSERT INTO PLAYLIST VALUES (1516, 'Playlist 7', 2021, 'FOLK');
24. INSERT INTO PLAYLIST VALUES (1517, 'Playlist 8', 2021, 'CLASSICAL');
25. INSERT INTO PLAYLIST VALUES (1518, 'Playlist 9', 2021, 'COUNTRY');
26. INSERT INTO PLAYLIST VALUES (1519, 'Playlist 10', 2021, 'R&B');
27. INSERT INTO SONG VALUES (54321, 'Changes', '2019-08-12', 1510);
28. INSERT INTO SONG VALUES (54324, 'Death Punch', '2004-06-15', 1510);
29. INSERT INTO SONG VALUES (54325, 'No Cap', '2018-09-08', 1510);
30. INSERT INTO SONG VALUES (34341, 'Chop Money', '2020-02-25', 1511);
31. INSERT INTO SONG VALUES (34342, 'Vamos Azul', '2020-11-09', 1511);
32. INSERT INTO SONG VALUES (34366, 'Symphony', '1900-05-01', 1513);
33. INSERT INTO SONG VALUES (34367, 'For the Longest', '2002-04-03', 1513);
34. INSERT INTO SONG VALUES (34368, 'Summers Eve', '2005-04-22', 1513);
35. INSERT INTO SONG VALUES (34369, 'Dragons Lair', '2011-05-12', 1513);
36. INSERT INTO SONG VALUES (44392, 'Paris', '2010-09-23', 1515);
37. INSERT INTO SONG VALUES (44397, 'Nascar', '2010-11-11', 1515);
38. INSERT INTO SONG VALUES (59237, 'Bama Whiskey', '2006-11-14', 1515);
39. INSERT INTO SONG VALUES (61388, 'Lovin You', '1983-03-26', 1516);
40. INSERT INTO SONG VALUES (61353, 'Legend', '2000-04-05', 1516);
41. INSERT INTO SONG VALUES (61354, 'Courtside', '2016-03-18', 1516);

COMMIT;

# **Queries**

1. SELECT ACT\_TYPE FROM ACTIVITY;
2. SELECT PLAYLIST\_GENRE AS [All Genres] FROM PLAYLIST;
3. SELECT LIS\_FNAME, LIS\_LNAME, LIS\_AGE, LIS\_GENDER FROM LISTENER ORDER BY LIS\_LNAME;
4. SELECT SONG\_TITLE AS [All Songs], SONG\_GENRE FROM SONG ORDER BY SONG\_GENRE;
5. SELECT DISTINCT SUB\_DESCRIPTION AS [Subscription], SUB\_FEE FROM SUBSCRIPTION ORDER BY SUB\_FEE;
6. SELECT SONG\_TITLE, SONG\_DATE FROM SONG ORDER BY SONG\_DATE DESC;
7. SELECT COUNT(ACT\_TYPE) AS [Number Of Activities] FROM ACTIVITY;
8. SELECT COUNT(LIS\_NUM) AS [Number of Listeners] FROM LISTENER;
9. SELECT COUNT(PLAYLIST\_NUM) AS [Number of Playlists] FROM PLAYLIST;
10. SELECT SONG\_GENRE, COUNT(\*) AS [Number of Songs in Genre] FROM SONG GROUP BY SONG\_GENRE;
11. SELECT ACT\_TYPE AS Activity, PLAYLIST\_TITLE AS Playlist FROM ACTIVITY INNER JOIN PLAYLIST ON ACTIVITY.ACT\_NUM = PLAYLIST.ACT\_NUM ORDER BY ACT\_TYPE;
12. SELECT ACT\_TYPE, COUNT(\*) AS [Count of Listeners] FROM ACTIVITY GROUP BY ACT\_TYPE ORDER BY COUNT(\*) DESC;
13. SELECT LISTENER.LIS\_LNAME, LISTENER.LIS\_FNAME, SUBSCRIPTION.SUB\_DESCRIPTION, SUBSCRIPTION.SUB\_FEE

FROM LISTENER RIGHT JOIN SUBSCRIPTION ON LISTENER.LIS\_NUM = SUBSCRIPTION.LIS\_NUM

ORDER BY SUBSCRIPTION.SUB\_DESCRIPTION, LISTENER.LIS\_LNAME;

1. SELECT PLAYLIST.PLAYLIST\_TITLE, SONG.SONG\_TITLE

FROM PLAYLIST RIGHT JOIN SONG ON PLAYLIST.PLAYLIST\_NUM = SONG.PLAYLIST\_NUM

ORDER BY PLAYLIST.PLAYLIST\_TITLE;

1. SELECT LISTENER.LIS\_FNAME, LISTENER.LIS\_LNAME, ACTIVITY.ACT\_TYPE

FROM LISTENER RIGHT JOIN ACTIVITY ON LISTENER.LIS\_NUM = ACTIVITY.LIS\_NUM

ORDER BY ACTIVITY.ACT\_TYPE, LISTENER.LIS\_LNAME;

1. SELECT ACTIVITY.ACT\_TYPE, SONG.SONG\_TITLE

FROM (ACTIVITY RIGHT JOIN PLAYLIST ON ACTIVITY.ACT\_NUM = PLAYLIST.ACT\_NUM) RIGHT JOIN SONG ON PLAYLIST.PLAYLIST\_NUM = SONG.PLAYLIST\_NUM

ORDER BY ACTIVITY.ACT\_TYPE;

1. SELECT LISTENER.LIS\_FNAME, LISTENER.LIS\_LNAME, SUBSCRIPTION.SUB\_FEE, SUBSCRIPTION.SUB\_DESCRIPTION

FROM LISTENER RIGHT JOIN SUBSCRIPTION ON LISTENER.LIS\_NUM = SUBSCRIPTION.LIS\_NUM

WHERE (((SUBSCRIPTION.SUB\_FEE)>3))

ORDER BY LISTENER.LIS\_LNAME;

1. SELECT LISTENER.LIS\_FNAME, LISTENER.LIS\_LNAME, SUBSCRIPTION.SUB\_FEE, SUBSCRIPTION.SUB\_DESCRIPTION

FROM LISTENER RIGHT JOIN SUBSCRIPTION ON LISTENER.LIS\_NUM = SUBSCRIPTION.LIS\_NUM

WHERE (((SUBSCRIPTION.SUB\_FEE)=3))

ORDER BY LISTENER.LIS\_LNAME;

1. SELECT LISTENER.LIS\_FNAME, LISTENER.LIS\_LNAME, SUBSCRIPTION.SUB\_FEE, SUBSCRIPTION.SUB\_DESCRIPTION

FROM LISTENER RIGHT JOIN SUBSCRIPTION ON LISTENER.LIS\_NUM = SUBSCRIPTION.LIS\_NUM

WHERE (((SUBSCRIPTION.SUB\_FEE)<3))

ORDER BY LISTENER.LIS\_LNAME;

# **Forms and Reports**

Graphical user interface

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

# **Reflections**

## Sam

During the project, I discovered interesting tidbits, one of which was the auto-number data type for the primary keys. This helped with the issue of adding new listeners because access assigned automatic numbers to each listener. I also discovered the lookup field for the foreign keys which helped with the same problem, in that it allowed for quicker linking.

Creating the database for Music Co was simple and straightforward. The issue we needed to solve did not require too many queries, however, I ran into an issue with joining some of the tables. An inner join would join the tables, but it would not display the data from either table. It would only display the table headers. A solution was to use a left join or a right join but that only displayed the data from one table. I later discovered that the issue had to do with setting the primary keys to auto-number. The joins worked after reverting the auto-numbered primary keys back to manual entries.

With this project, there was an issue of how we could both work on the database. Access does not allow multiple users simultaneous access to the database, so our solution was to maintain a copy of our database on our individual computers. Any changes we made would then be saved to our Microsoft Teams files.

## Fletcher

I had a good time working on this project with Sam. I didn’t run into too many issues but one that stuck out was when I was creating queries. I have never worked in Microsoft Access before, so it was hard to figure out what to do. I ended up watching a video on how to do it and that solved all my problems. Overall, it was a fun project, and I learned a lot about Access and creating databases.

Like Sam said, there was an issue with being able to work together. Access didn’t allow us to work together in real-time, so it was a bit difficult getting things done. We had to take turns working on the project and then upload what we had done in teams. While this ended up working, it was inconvenient and probably cost us time.

# **References**

*Access: Getting Started*. (2016, April 8). YouTube. https://www.youtube.com/watch?v=VIVrhhEIFR8&list=PLpQQipWcxwt-IWBqprrYVk53qPo53ZxMw&index=6

*SQL Tutorial*. (2022). W3schools. https://www.w3schools.com/sql/default.asp

# **Additional Resources**

* [executivesummary.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/executivesummary.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [BusinessRules.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/BusinessRules.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [erd.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/erd.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [relationalschema.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/relationalschema.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [fd.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/fd.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [datadictionary.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/datadictionary.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [sqlexamples.pdf](https://kennesaw.view.usg.edu/content/enforced2/2511085-CO.430.IS3280.16081.20224/CourseProject/filesonIS/sqlexamples.pdf?_&d2lSessionVal=gXDn614g4XMBJKf9SRx9612BR&ou=2511085)
* [Using Access to Create Forms and Reports -video demo - Data Management Section W01 Spring Semester 2022 CO (usg.edu)](https://kennesaw.view.usg.edu/d2l/le/content/2511085/viewContent/39189936/View?ou=2511085)