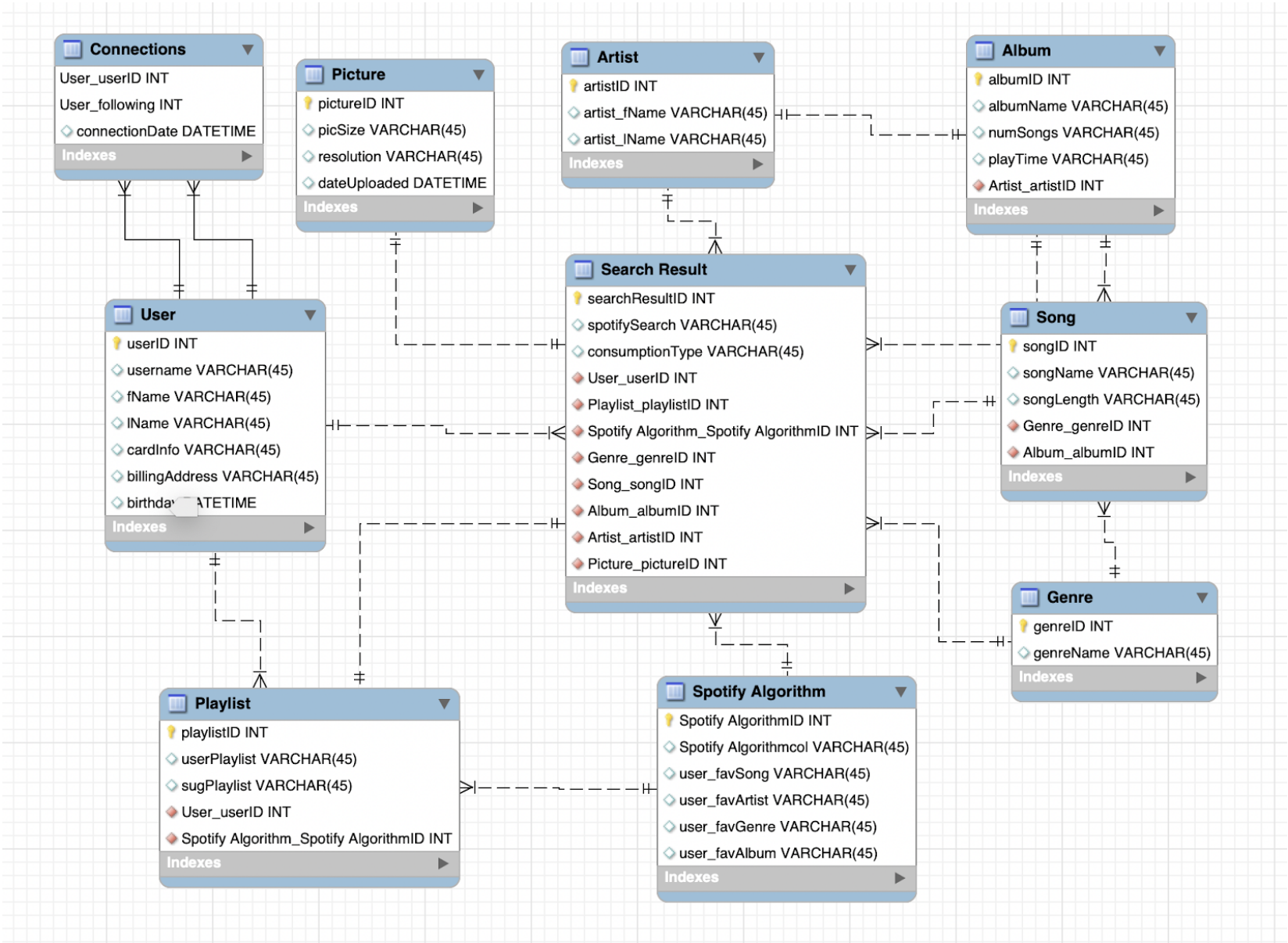
Database Analytics Project Over Spotify

By The Algorithms

Elena Abbott, Manning Edson, Grant Harney, & Cole Phillips

Database Model:



Introduction:

Our database model is for the app Spotify. Spotify takes information from each user to personalize the music listening experience. In our database, we created ten entities that contain data from the app. The goal of our model is to take the data from a group of users and then suggest a playlist for that user that is based on each users' search result. The other entities contain baseline data that is contained within the app such as songs, albums, artists, and user information. One of the entities, Spotify Algorithm, contains the data for each users' preferences, and works with the search result table to enhance the personalization of the suggested playlist.

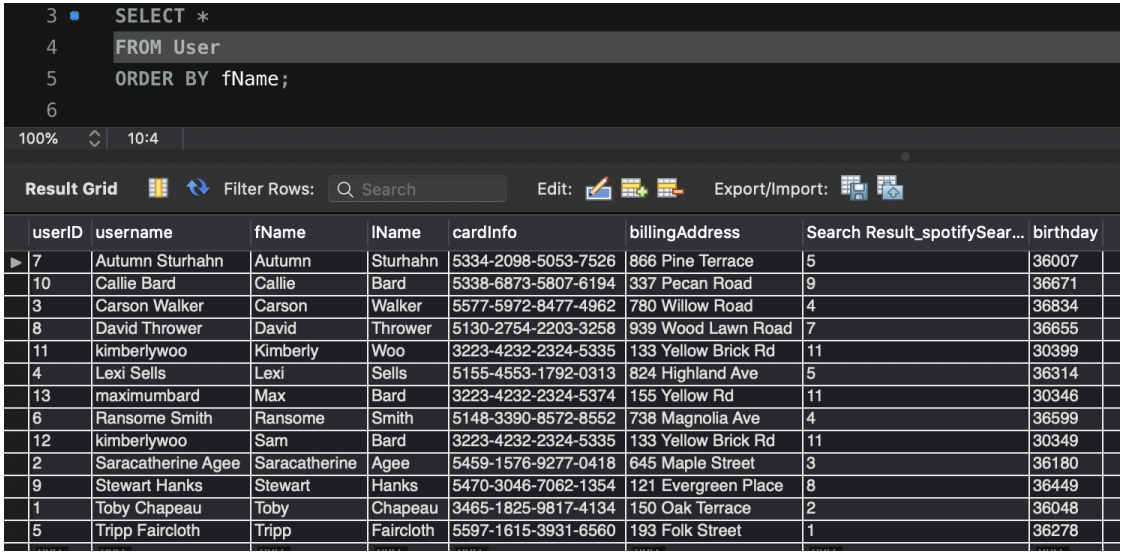
Feature Checklist:

Feature	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Multiple Table Join				X	X		X	X		X	
Subquery					X						
Correlated Subquery				X							
GROUP BY/ GROUP BY with HAVING							x				
ORDER BY	X							X			
DIVIDE											X
IN or NOT IN						X			X		
Built in function (AVG, CONCAT, etc.)					X		X		X		
REGEXP		X									
EXISTS/NOT EXISTS			X								

Queries:

1. This query is designed to show all the information of all users in alphabetical order of first names.

```
SELECT *
FROM User
ORDER BY fName;
```



```
3  SELECT *
4  FROM User
5  ORDER BY fName;
6
```

100% 10:4

Result Grid Filter Rows: Search Edit: Export/Import:

userID	username	fName	lName	cardInfo	billingAddress	Search Result_spotifySear...	birthday
7	Autumn Sturhahn	Autumn	Sturhahn	5334-2098-5053-7526	866 Pine Terrace	5	36007
10	Callie Bard	Callie	Bard	5338-6873-5807-6194	337 Pecan Road	9	36671
3	Carson Walker	Carson	Walker	5577-5972-8477-4962	780 Willow Road	4	36834
8	David Thrower	David	Thrower	5130-2754-2203-3258	939 Wood Lawn Road	7	36655
11	kimberlywoo	Kimberly	Woo	3223-4232-2324-5335	133 Yellow Brick Rd	11	30399
4	Lexi Sells	Lexi	Sells	5155-4553-1792-0313	824 Highland Ave	5	36314
13	maximumbard	Max	Bard	3223-4232-2324-5374	155 Yellow Rd	11	30346
6	Ransome Smith	Ransome	Smith	5148-3390-8572-8552	738 Magnolia Ave	4	36599
12	kimberlywoo	Sam	Bard	3223-4232-2324-5335	133 Yellow Brick Rd	11	30349
2	Saracatherine Agee	Saracatherine	Agee	5459-1576-9277-0418	645 Maple Street	3	36180
9	Stewart Hanks	Stewart	Hanks	5470-3046-7062-1354	121 Evergreen Place	8	36449
1	Toby Chapeau	Toby	Chapeau	3465-1825-9817-4134	150 Oak Terrace	2	36048
5	Tripp Faircloth	Tripp	Faircloth	5597-1615-3931-6560	193 Folk Street	1	36278

2. This query pulls all album names that start with the letter ‘F’.

```
SELECT AlbumName
FROM Album
WHERE AlbumName REGEXP'^F';
```

```
26    SELECT AlbumName
27    FROM Album
28    WHERE AlbumName REGEXP'^F';
29
```

100% 28:28 2 errors found

Result Grid Filter Rows: Search

AlbumName
Fine Line
folklore
Future Nostalgia

3. This query selects names of artists with only a first name. The last name is filled with a space as a placeholder.

SELECT username
FROM User
WHERE lName = ‘ ‘;

```
12    SELECT artist_fName
13    FROM Artist
14    WHERE artist_lName = ' ';
15
```

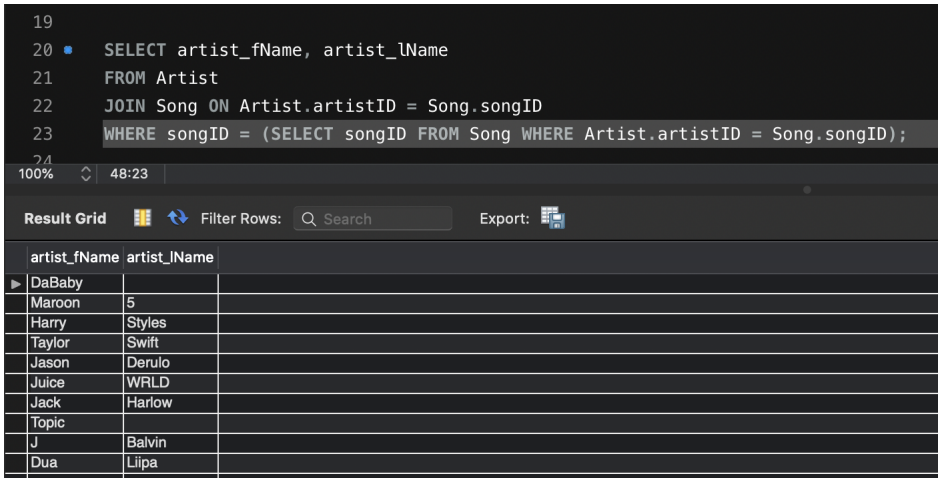
100% 8:12 1 error found

Result Grid Filter Rows: Search

artist_fName
DaBaby
Topic

4. This query selects the artist's first and last name if the artistID and the songID match.

```
SELECT artist_fName, artist_lName
FROM Artist
JOIN Song ON Artist.artistID = Song.songID
WHERE songID = (SELECT songID FROM Song WHERE Artist.artistID = Song.songID);
```



```
19
20 •  SELECT artist_fName, artist_lName
21    FROM Artist
22    JOIN Song ON Artist.artistID = Song.songID
23    WHERE songID = (SELECT songID FROM Song WHERE Artist.artistID = Song.songID);
24
```

100% 48:23

Result Grid Filter Rows: Search Export:

artist_fName	artist_lName
DaBaby	
Maroon	5
Harry	Styles
Taylor	Swift
Jason	Derulo
Juice	WRLD
Jack	Harlow
Topic	
J	Balvin
Dua	Liipa

5. This query grabs the length of the song and the artist's first and last name if the song's length is longer than the average song length.

```
SELECT songLength, artist_fName, artist_lName
FROM Song, Artist
WHERE Song.songID = Artist.artistID
AND songLength > (SELECT AVG(songLength) FROM Song);
```

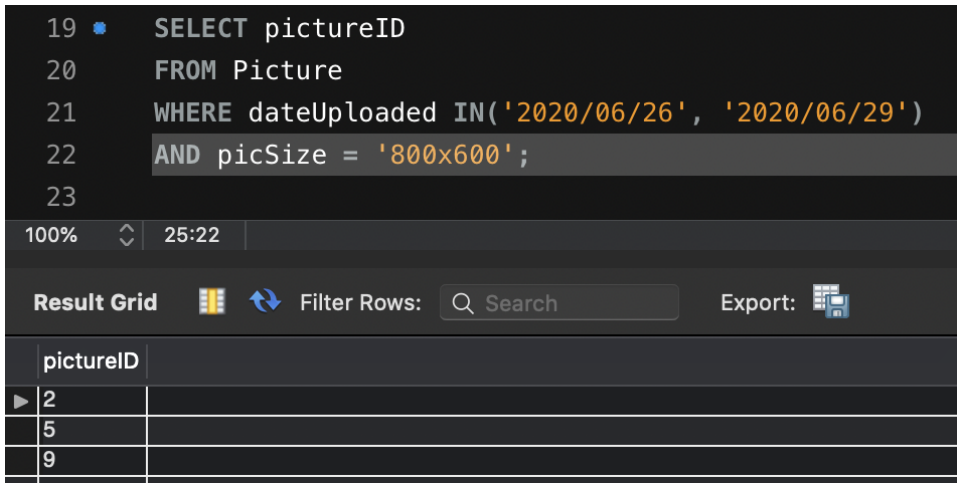
```
16 •  SELECT songLength, artist_fName, artist_lName
17    FROM Song, Artist
18    WHERE Song.songID = Artist.artistID
19    AND songLength > (SELECT AVG(songLength) FROM Song);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

songLength	artist_fName	artist_lName
4:12	Maroon	5
4:10	Taylor	Swift
4:32	J	Balvin

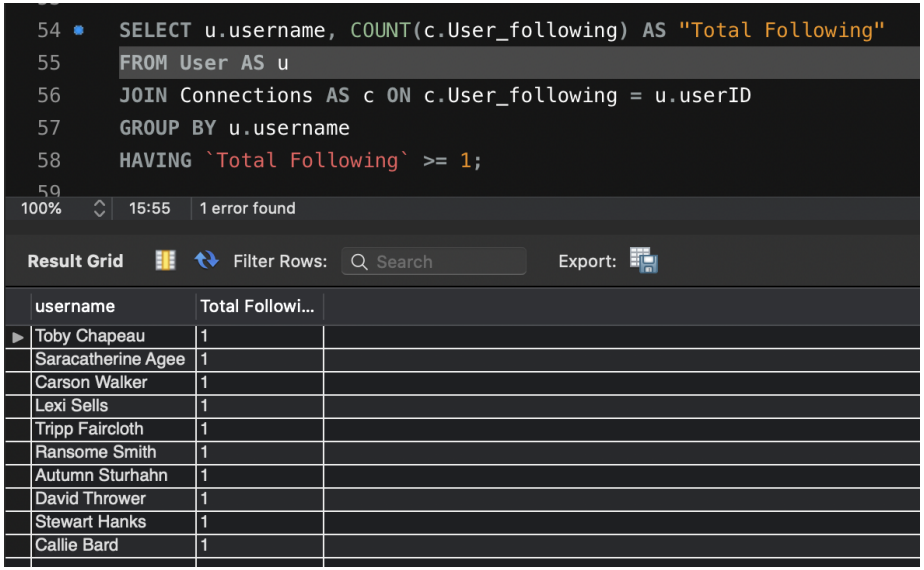
6. This query selects the pictureID if the date uploaded is either June 26th or 29th in the year 2020 and has a picture size of 800x600.

```
SELECT pictureID
FROM Picture
WHERE dateUploaded IN('2020/06/26', '2020/06/29')
AND picSize = '800x600';
```



7. Find the username and number of followers associated with the account when the user follows one or more accounts.

```
SELECT u.username, COUNT(c.User_following) AS "Total Following"
FROM User AS u
JOIN Connections AS c ON c.User_following = u.userID
GROUP BY u.username
HAVING `Total Following` >= 1;
```



```
54  SELECT u.username, COUNT(c.User_following) AS "Total Following"
55  FROM User AS u
56  JOIN Connections AS c ON c.User_following = u.userID
57  GROUP BY u.username
58  HAVING `Total Following` >= 1;
59
```

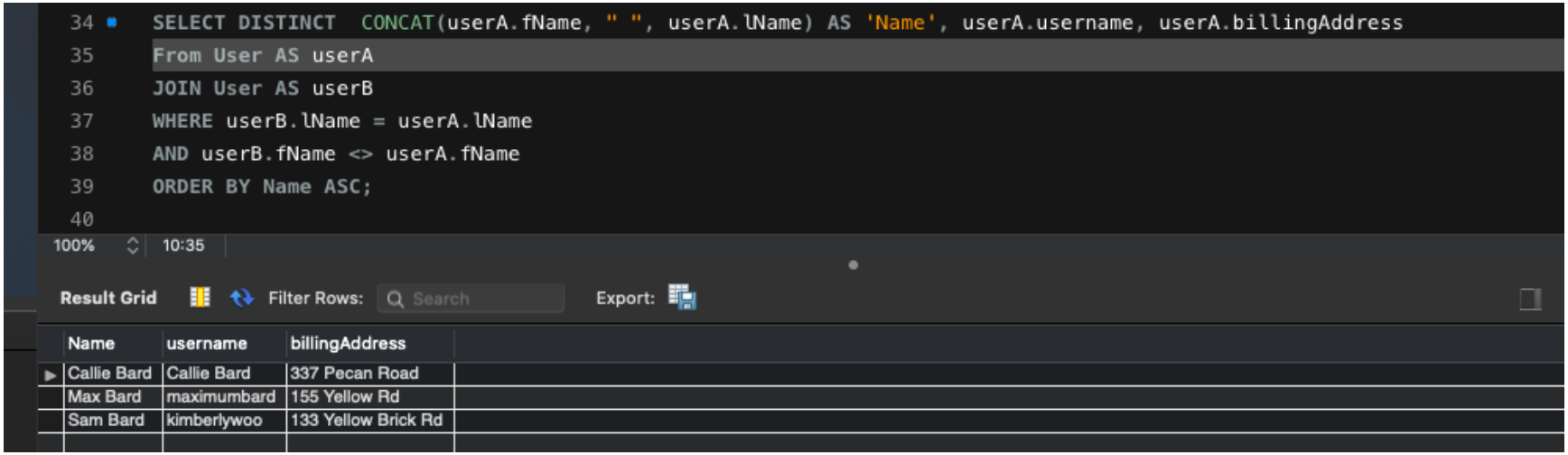
100% 15:55 1 error found

Result Grid Filter Rows: Search Export:

username	Total Followi...
Toby Chapeau	1
Saracatherine Agee	1
Carson Walker	1
Lexi Sells	1
Tripp Faircloth	1
Ransome Smith	1
Autumn Sturhahn	1
David Thrower	1
Stewart Hanks	1
Callie Bard	1

8. Find the full name, username, and billing information of the users who have the same last name.

SELECT DISTINCT CONCAT(userA.fName, " ", userA.lName) AS 'Name', userA.username, userA.billingAddress
From User AS userA
JOIN User AS userB
WHERE userB.lName = userA.lName
AND userB.fName <> userA.fName
ORDER BY Name ASC;



9. Find the song name and song length of songs that are greater than 3 minutes and less than 4 minutes, starting with the letter “W.”

SELECT songName, songLength
FROM Song
WHERE songLength IN (3)
AND songName LIKE ('W%');

```
41
42 •  SELECT songName, songLength
43     FROM Song
44     WHERE songLength IN (3)
45     AND songName LIKE ('W%');
46
```

100% 1:46

Result Grid | Filter Rows: Search | Export:

songName	songLength
Watermelon Sugar	3:54
Wishing Well	3:23
WHATS POPPIN	3:35

10. This query grabs the album name when a song is released by itself, and not as a composed album. When the play time of an album is divided by a single song length and equals 1, then the song must be the only one from the album released yet; this is known as a solo release.

```
SELECT albumName,(playTime/songLength) AS 'Solo_Release'
FROM Album
JOIN Song ON Album.playTime = Song.songLength
WHERE Album.playTime = Song.songLength;
```

```
32 •  SELECT albumName,(playTime/songLength) AS 'Solo_Release'
33     FROM Album
34     JOIN Song ON Album.playTime = Song.songLength
35     WHERE Album.playTime = Song.songLength;
36
```

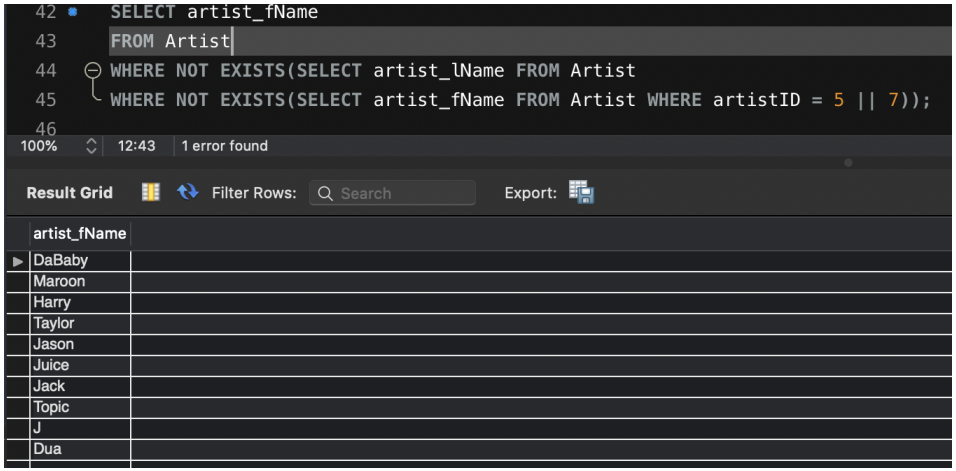
100% 36:35

Result Grid | Filter Rows: Search | Export:

albumName	Solo_Release
UN DIA (ONE DAY)	1

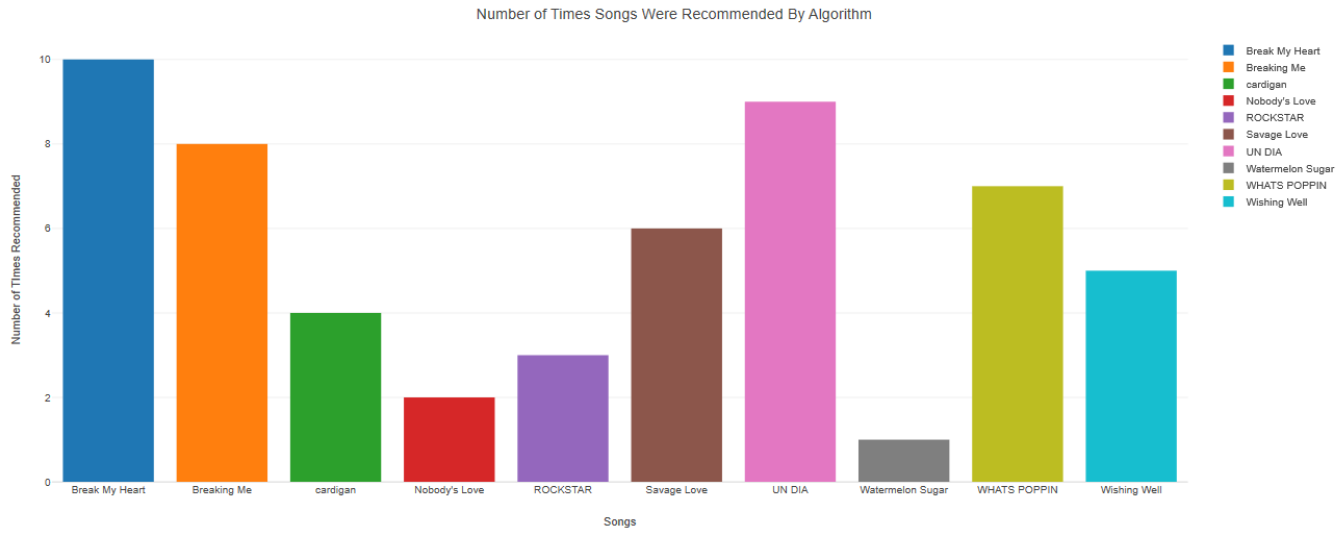
11. This will select the artist's first name when an artist's last name does not exist if a first name does not exist for artistIDs 5 and 7. Since these artistIDs have first names attached to them, there is no null column to be selected, and all last names are returned.

SELECT artist_fName
FROM Artist
WHERE NOT EXISTS(SELECT artist_lName FROM Artist
WHERE NOT EXISTS(SELECT artist_fName FROM Artist WHERE artistID = 5 || 7));

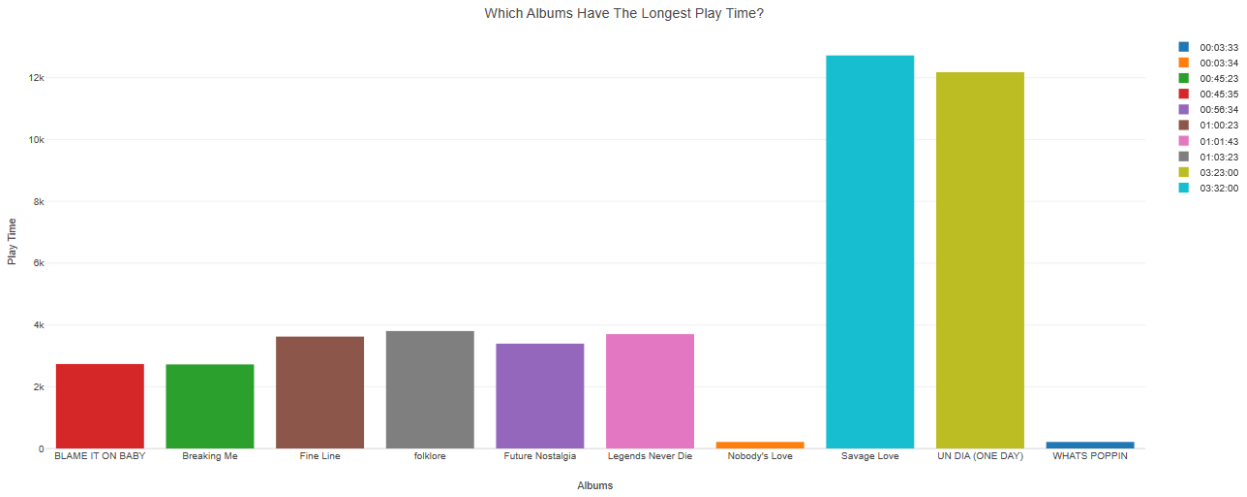


Database Analytics Report

How many times did the Spotify algorithm recommend certain songs?



Which Albums have the Longest Play Time?



What Pixel Size is Most Common for Album Covers?