Team Project Final Report

INST123-0103: Databases for All | Song Library Database

University of Maryland, College Park

INTRODUCTION

The database that we created in our team project takes the form of a song library. As a group, we enjoy many different types of music and felt that creating a database around this topic would be very informative to a variety of users. In addition, we feel that music is something that can be enjoyed by anyone, no matter your background, giving our database a wide range of potential audiences.

The music industry is vast, garnering over billions of dollars in revenue each year, and with the rise of streaming services, higher amounts of music are more easily accessible than ever before. Knowing this, we decided to focus on the most popular song each year for the past 50 years within our database in order to grasp the overall music consumption within the United States, while using industry sources to collect data. We compiled different attributes of each song within our database in order to create a collection of data that cannot be easily found in a similar format anywhere else online. Doing this makes our song library database both unique and effective at conveying the intended information.

Our database consists of four tables: song_stats, song_credits, song_perfrormance, and song_album. Each table has 50 rows in total - one for each top-performing song for the last 50 years. The attributes and formatting details for each table are shown within the database description section (sample data). Together, they create a set of data that describes the evolution of the music industry and the progression of popular music in the United States over the last 50 years. Considering its various properties, this database would not only be of use to general music fans, but also industry professionals and analysts who aim to learn more about music consumption. Our goal with this project was to create an organized database that provides a diverse amount of musical information to its users, which we feel that we have achieved while completing it throughout the semester.

DATABASE DESCRIPTION

Physical Database

group_09_songs_backup.sql

group_09_songs_queries.txt

Sample Data

1) Sample of the song_stats table within our database. In total, it has 50 rows of data.

song_id [PK] bigint	song_title character varying (100)	date_released date	year_top_song integer	song_length_seconds integer	song_genre character varying (100)
1	Heat Waves	2020-06-29	2022	238	Psychedelic pop
2	Levitating	2020-10-01	2021	203	Electro-disco
3	Blinding Lights	2019-11-29	2020	201	New wave
4	Old Town Road	2019-04-05	2019	157	Country rap
5	God's Plan	2018-01-19	2018	198	Pop-rap
6	Shape of You	2017-01-06	2017	234	Pop
7	Love Yourself	2015-11-09	2016	233	Pop
8	Uptown Funk	2014-11-10	2015	270	Funk-pop
9	Happy	2013-11-21	2014	235	Soul
10	Thrift Shop	2012-08-27	2013	235	Pop-rap
11	Somebody That I Used to Know	2011-07-05	2012	244	Alternative
12	Rolling in the Deep	2010-11-29	2011	228	R&B
13	Tik Tok	2009-08-07	2010	200	Dance-pop
14	Boom Boom Pow	2009-02-22	2009	252	EDM
15	Low	2007-10-09	2008	230	Crunk

2) Sample of the song_credits table within our database. In total, it has 50 rows of data.

lead_artist_id [PK] bigint	song_id bigint	lead_artist character varying (50)	lead_songwriter character varying (50)	lead_producer character varying (50)	music_label character varying (50)
1	1	Glass Animals	Dave Bayley	Dave Bayley	Wolf Tone
2	2	Dua Lipa	Dua Lipa	Stephen Kozmeniuk	Warner Records
3	3	The Weeknd	Abel Tesfaye	Max Martin	Republic Records
4	4	Lil Nas X	Montero Hill	Kiowa Roukema	Columbia Records
5	5	Drake	Aubrey Drake Graham	Ronald Nathan LaTour Jr.	OVO Sound
6	6	Ed Sheeran	Edward Sheeran	Edward Sheeran	Asylum Records
7	7	Justin Bieber	Justin Bieber	Benjamin Joseph Levin	Def Jam
8	8	Mark Ronson	Mark Ronson	Mark Ronson	Columbia Records
9	9	Pharrell Williams	Pharrell Williams	Pharrell Williams	Black Lot Music
10	10	Macklemore & Ryan Lewis	Ben Haggerty	Lewis	Macklemore LLC
11	11	Gotye	Wally De Backer	Wally De Backer	Eleven
12	12	Adele	Adele Adkins	Paul Epworth	XL Recordings
13	13	Kesha	Kesha Sebert	Dr. Luke	RCA Records
14	14	The Black Eyed Peas	William Adams	will.i.am	Interscope Records
15	15	Flo Rida	Tramar Dillard	DJ Montay	Atlantic

3) Sample of the song_performance table within our database. In total, it has 50 rows of data.

song_id bigint	spotify_streams bigint	certified_units_sold bigint	billboard_weeks integer
1	2422569377	5000000	91
2	1723520142	4000000	77
3	3561144749	10000000	90
4	1421542819	17000000	45
5	2180280044	15000000	36
6	3467556499	10000000	59
7	2054664252	9000000	41
8	1709685583	11000000	56
9	1186010640	11000000	47
10	981482505	10000000	49
11	1394216029	14000000	59
12	1395684472	8000000	65
13	913226525	8000000	38
14	319556019	5000000	33
15	824148691	13000000	40

4) Sample of the song_album table within our database. In total, it has 50 rows of data.

song_id bigint	album_name character varying (100)	album_release_date date	number_songs integer	certified_units_sold bigint
1	Dreamland	2020-08-07	16	500000
2	Future Nostalgia	2020-03-27	13	1000000
3	After Hours	2020-03-20	14	3000000
4	7 EP	2019-06-21	8	2000000
5	Scorpion	2018-06-29	27	5000000
6	%	2017-03-03	16	5000000
7	Purpose	2015-11-13	14	5000000
8	Uptown Special	2015-01-13	11	1000000
9	GIRL	2014-03-03	11	500000
10	The Heist	2012-10-09	15	5000000
11	Making Mirrors	2011-08-19	12	2000000
12	21	2011-01-24	11	14000000
13	Animal	2010-01-01	14	3000000
14	The E.N.D.	2009-06-03	15	2000000
15	Mail on Sunday	2008-03-18	14	390000

Queries

Query 1: Which music labels have sold the most total albums among the top-performing songs?

```
SELECT song_credits.music_label, SUM(song_performance.certified_units_sold) AS
units_per_label
FROM song_performance
JOIN song_credits ON song_performance.song_id = song_credits.song_id
GROUP BY music_label
ORDER BY SUM(song_performance.certified_units_sold) DESC;
```

Query 2: Which top-performing songs that were released before Spotify was created have the most Spotify streams?

```
SELECT song_stats.song_title, song_stats.year_top_song, song_performance.spotify_streams
FROM song_stats
JOIN song_performance ON song_stats.song_id = song_performance.song_id
WHERE song_stats.year_top_song <= 2005
ORDER BY song_performance.spotify_streams DESC;
```

Query 3: Which top-performing songs have a song length more than 30 seconds away from the average, and how many weeks did they chart on billboard?

```
SELECT song_stats.song_title, song_stats.song_length_seconds,
song_performance.billboard_weeks
FROM song_stats
JOIN song_performance ON song_stats.song_id = song_performance.song_id
WHERE song_stats.song_length_seconds < (
        (SELECT AVG(song_length_seconds) FROM song_stats) - 30
) OR song_stats.song_length_seconds > (
        (SELECT AVG(song_length_seconds) FROM song_stats) + 30
);
```

Query 4: Which artists were the lead songwriter on their top-performing song?

```
SELECT song_stats.song_title, song_credits.lead_artist, song_credits.lead_songwriter
FROM song_stats
JOIN song_credits ON song_stats.song_id = song_credits.song_id
WHERE song_credits.lead_artist = song_credits.lead_songwriter;
```

Query 5: Which top-performing songs sold greater than 1,000,000 units as a song and album?

```
SELECT song_stats.song_title, song_performance.certified_units_sold,
song_album.album_name, song_album.certified_units_sold
FROM song_stats
JOIN song_performance ON song_stats.song_id = song_performance.song_id
JOIN song_album ON song_stats.song_id = song_album.song_id
WHERE song_performance.certified_units_sold > 1000000 AND
song_album.certified_units_sold > 1000000;
```

Query Description	**Requirement A** (4 queries)	**Requirement B** (3 queries)	**Requirement C** (2 queries)	**Requirement D** (1 query)
Query 1	✓		✓	
Query 2	✓	✓		
Query 3	✓	✓	✓	✓
Query 4	✓	✓		
Query 5	✓	✓		

CHANGES FROM ORIGINAL DESIGN

Over the course of developing this project, the original design for our proposal has changed in many ways as a result of trial and error regarding formatting our database and collecting data.

In our initial proposal, we had proposed to create four tables for our song library database. The names and general ideas of these tables stuck, however, the attributes of each have changed since the outline stage. We kept the original four columns for our song_stats table but added a column for year_top_song, which describes which year a specific song was named the top performing song of the year. For our song_credits table, we changed the artist, songwriter, and producer columns to only display the lead role in each position to make our data appear more succinct. For our song_performance table, we removed the on-demand streams column and changed our peak_billboard column to billboard_weeks, which describes how many weeks the song charted on Billboard. Lastly, in our song_album table, we added a column to display how many units each album sold. These changes were made based on which data was available to collect from industry sources, and to show more diverse and informative data within our database. For instance, we thought billboard_weeks was more informative to how each song performed than billboard_peak, and we removed on-demand streams since we believed spotify_streams was very similar and this data was not easily accessible online for each song.

As we collected data, we had to stray away from our proposed plan and pull from additional industry sources to secure all of our data points. In addition, we decided to manually enter our data into a shared file that we could each individually download to allow for easy collaboration in data collection and analysis. Each change we made to our original proposal helped form our song library into a well-developed and meaningful database.

LESSONS LEARNED

While completing our database project, we learned of the importance of properly organizing our database. When outlining our project in the initial proposal, we unknowingly proposed to create tables that contained some irrelevant and unvarying data that would have undermined the relevance of our database. By dividing and categorizing our potential data in a more meaningful way as we moved forward with our project, we were able to reformat our tables in a way that allowed us to organize our data better. More specifically, it led us to adding relevant attributes into our database tables (such as billboard_weeks, year_top_song, certified_units_sold) to ensure our database completely encapsulated the concept of a song library.

As a group, we also learned how to properly collect and manage data. As we found and entered data into our database as we completed data collection, we strayed from the original

plan. Our original proposal greatly underestimated the amount of sources we would need to find our intended data, in addition to the process we would utilize to record it. We ended up adding more industry trusted sources and recording the data in a format that allowed us to download it as a CSV file to easily view the data and perform analysis within pgAdmin. This improved our data management and analysis skills overall.

POTENTIAL FUTURE WORK

There are endless possibilities for future work regarding our song library database and its applications. Since we are storing data regarding the top performing song of each year, we could continue to grow our database by following up for the next few years. This would allow for further evaluation of the growth and evolution of the music industry as time goes on. Not only that, but we could increase the number of top-performing songs for each year to add to our existing database to garner further insight on the general public's music consumption.

We could also expand the amount of characteristics and attributes we store in our database regarding each individual song. This includes adding data from other streaming platforms besides Spotify, such as Apple Music and SoundCloud, to gain a higher understanding of the United States's listener base. Further, we could add additional attributes to create a more expansive database, such as a column identifying if a song was a solo or collaboration, if the song has remixes (and how many), the lead composer for each song, among other categories.

Not only that, but additional databases could be created that link and relate to the data in our song library. Since our database focuses more on how music performs in the United States, other databases could be created to cover the top-performing songs in various countries. This would provide further information on how the music industry is consumed on a global scale, and create a collection of databases that encapsulate worldwide music consumption.