

Team Project Proposal

December 12, 2020

Team 1: Kamran Djourshari, Diego Amores, Vi Le, Guanzhi Chen, Chi-Hao Sheng

Team Project: Final Submission (Report and Database)

Introduction:

For our project, we made a database on Vinyl records. Vinyl records were created in the 1930s, as a method to bring music to every listener's home environment. Vinyl records describe music albums that put different songs together, from one artist, for music listeners to enjoy. With so many different artists available, we thought it would be a good idea to highlight some of the classics/hits over the years. We were motivated to bring this information together into one database for all music listeners to enjoy and deep dive into.

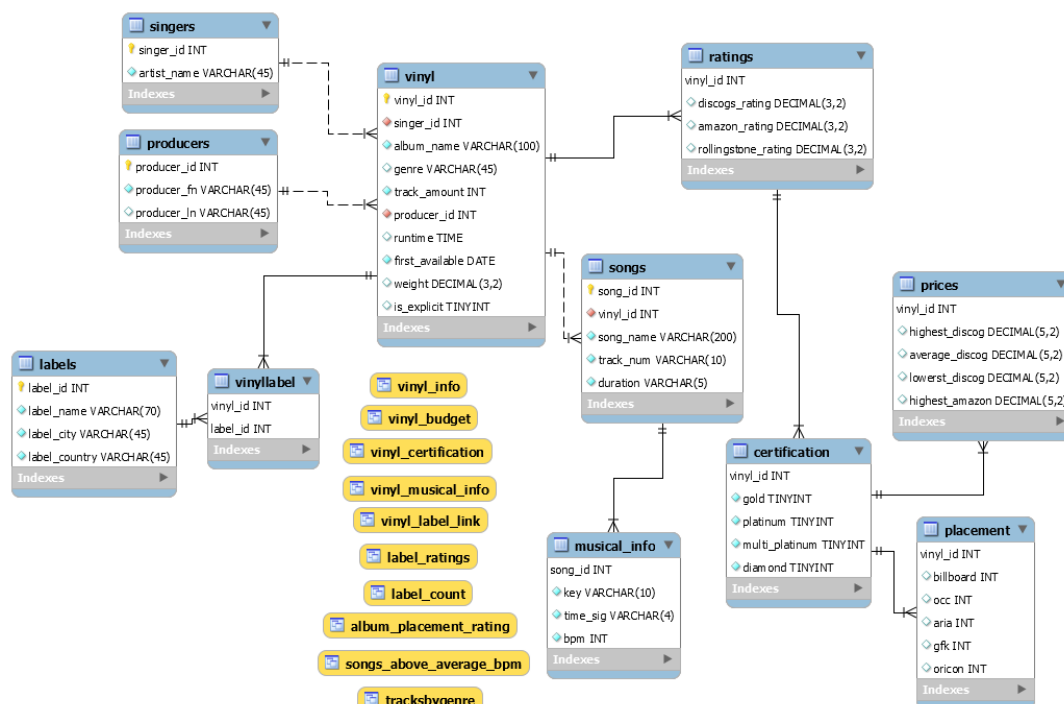
To make this database, we focused specifically on creating multiple different tables that all carry significant information about each vinyl. Using information from websites like Wikipedia, Amazon, Discogs, Disconest, and Music Stack, only the most popular vinyl records within the genres of Rock, Metal, Pop, and Hip-Hop were selected. The database incorporates tables relating to vinyl records including the main table, the songs table, the artist table, the producer table, and the prices table. These tables were used to present interesting information on the most popular vinyl albums. Since our previous progress report, we decided to clean up the database by limiting the number of columns in the main vinyl table as they were not that important for music listeners including columns like "album photo", "language", and "is_discontinued". However, the vinyl records database still covers a variety of different topics, from each of these tables mentioned, including various albums, unique auto-incremented IDs (vinyl, singer, and producers), artists' names, specific songs, run-time, ratings, labels, producers, weight, and more. From these tables, we have multiple different types of relationships linking

back to our main table including a one-to-one relationship, a one-to-many relationship, and a many-to-many relationship, which helps to keep the database clean and normalized so music enthusiasts can easily browse through it. With our database formed we hope that music listeners will be able to explore our views and database and learn more about their favorite vinyl hits from the 1980s and 2010s like with their varying prices and peak placement over the years.

Database Description:

We included 20 iconic vinyl records from the eras of the 1980s and 2010s. The database covers a variety of different topics with 11 different tables included for all music enthusiasts to enjoy. The database and queries serve as a tool for learning interesting facts about vinyl albums that even the biggest music enthusiasts would not have known.

Logical Design:



Our logical design was based on organizing information into categories that would make the most sense for each piece of information that we wanted to convey. We decided that we wanted the information to include data that would be relevant to vinyl collectors as well as those who

wanted to retrieve just simple information about the vinyl. Therefore, we wanted all the information to be easily accessible without dumping too much into the main vinyl table. With this goal, we created 10 tables with 1 linking table. We have a main vinyl table with most of the information and separate smaller tables that hold more niche information that would be more important to people who want to look for it. We also wanted the name of each table to be perfectly representative of the data that we put in so that our database was more readable and intuitive, which is why our database is organized the way it is.

Physical Database:

The database our team created includes 11 tables. The centerpiece of all our tables is the vinyl table. This table connects to 5 other tables, which are: 'ratings', 'songs', 'singers', 'producers', and 'vinyllabel'. Three of these tables' relationships have one-to-many with the vinyl table, which includes: 'vinyllabel', 'ratings', and 'songs. There are 2 tables relationships are many-to-one with the vinyl table: singers and producers. There is one many-to-many relationship with it consisting of the vinyl table and labels table. In this situation, 'vinyllabel' served as the linking table for this many-to-many relationship. For the 'vinyllabel', this table has a many-to-one relationship with the labels and vinyl tables. The songs table has a many-to-one relationship with the musical_info table. The 'ratings' table a many-to-one relationship with the 'certification' table. The last certification has 2 many-to-one relationships with the 'prices' table and 'placement' table.

Sample Data:

For our sample data, we retrieved vinyl record information from various websites such as Discogs, Amazon, Disconest, Music Stack, and Wikipedia. We manually looked for vinyl records with our criteria of vinyl records from the 1980s and 2010s. For collecting data, we looked for metadata that was related to vinyl records, such as vinyl name, artist name, genre,

track amount, runtimes, etc. We also looked for pricing, labels, ratings, and certifications on Amazon and Discogs. To increase the accuracy of our findings, we searched through Wikipedia to confirm our information. Then we proceeded with making CSV files so that we save our data and import it to our database more effectively than coding separate SQL scripts. Here is a sample of our main table, vinyl.

	vinyl_id	singer_id	album_name	genre	track_amount	producer_id	runtime	first_available	weight	is_explicit
▶	1	1	Thriller	Pop	9	1	00:42:19	1982-11-30	0.95	0
	2	2	Dark Side of The Moon	Rock	10	2	00:43:09	1973-03-01	1.14	0
	3	8	Back in Black	Rock	10	3	00:42:11	1980-07-25	0.70	0
	4	7	Greatest Hits	Rock	17	4	00:58:19	1981-10-26	1.35	0
	5	10	Like a Virgin	Pop	9	5	00:43:10	1984-11-12	0.85	0
	6	1	Bad	Pop	10	1	00:48:40	1987-08-31	0.56	0
	7	13	Let's Dance	Rock	8	5	00:39:41	1983-04-21	0.67	0

Views/Queries:

View Name	Req. A Join	Req B. Filtering	Req C. Aggregation	Req D. Linking	Req. E Subquery
vinyl_info	x				
vinyl_certification	x	x			
songs_above_average_bpm	x	x	x		x
label_ratings	x	x	x	x	
album_placement_rating	x		x		x
label_count	x		x		
tracksbygenre			x		
vinyl_budget	x	x			
vinyl_musical_info	x	x			
vinyl_label_link	x			x	
TOTAL: 10	9	5	5	2	2

vinyl_info: This view presents information on all the vinyl albums in the database including the artist, genre, number of songs, runtime, date when released, and its explicitly in a clean and presentable format.

vinyl_certification: A view that highlights the vinyl's name, artist, executive producer, and genre for those that had earned either a Diamond or Multi-platinum reward.

songs_above_average_bpm: A view that retrieves the songs that have a BPM above the average BPM. This view includes the song name, artist name, genre, and BPM.

label_ratings: A query/view emphasizing each label's rating based on Discogs, Amazon, and Rolling Stone reviews. It also includes an aggregate average rating from all three of these sites' ratings.

album_placement_rating: This view simulates a user querying the rating of vinyl records with above-average ratings and their peak positions on the Billboard. The result set shows the relationship between critics' ratings and placement on music charts.

label_count: The view pulls up the labels' name, city, country, and the count of how many times they appear in the database.

tracksbygenre: This query/view shows the average number of songs based on the specific type of genre (Dance, Rock, Hip-Hop, Metal, and Pop).

vinyl_budget: The vinyl budget view allows for users to show vinyl records with specified prices.

vinyl_musical_info: The vinyl musical info list more helpful information about vinyl records such as album name, song name, artist, track #, duration, and what musical key that track is played on.

vinyl_label_link: The vinyl label link, linking 3 of tables that had many-to-many relationships(vinyl, vinyllabel, labels). To be able to put certain values of each table together, so the user can easily access those values in the same section. Those values are the album name, label name, and label id.

Changes from Original Design:

The team has mostly retained the structural design from the original plan. Most changes were made due to project scope readjustment and normalizations.

The first change was reselecting the vinyl records. The selection went from the initial 30 vinyl records released between the 1980s to 2010s to 20 vinyl records released in the 1980s and in 2010s. This decision was made from the need to finish the project on time as the original plan would require an excessive workload which might make meeting the deadline difficult.

The second change involves adding new entities because of normalizations. After the first normalization, a new table called 'vinyllabel' was created to link the vinyl and label table that have a many-to-many relationship. The purpose is to ensure each combination of rows between the two tables could be uniquely identified. After the second normalization, a new table called 'singers' was created because the initial singer column in the vinyl table did not rely entirely on the primary key.

The third change is removing an entity and several columns. The countries of the release table were taken out because it overlaps with the first_available column in the vinyl table. The sound engineer column in the credits table and the studio column in the vinyl table were excluded because the data on them are sometimes obscure and the team agreed they were less relevant to the core purpose of the database. The is_discontinued and language column in the vinyl table was removed because all vinyl records are still in production and are all in English,

therefore it became unnecessary to dedicate individual columns for them. The album_photo column was removed due to technical difficulties and concern for database integrity.

Lessons Learned:

A lot of the challenges we faced were during the process of creating and inserting data into our database. Our team was very good at meeting every week to discuss our plans and work on our project, but that also means we were often working ahead. We would try to work on a part of the project not gone over in lecture and ultimately stress out over not knowing how to do it. For example, when we were inserting data into our database, we had not learned about importing the Excel sheets yet, so we almost made several stored processes to individually insert each row into the database for each table. We were worried about having to do all the stored processes and if affected the morale of the group. Luckily, the next day, we learned about importing the Excel sheets. We learned that it was ok to slow down and work at a more manageable pace.

What we learned ethically is that our database has a bias due to the nature of our topic. The data we included had the most popular vinyl of the 1980s and 2010s, a limit we set to give our database enough variety to contain robust data, but with enough constraint so we wouldn't have to fill it with hundreds of vinyl records. However, choosing the most popular records means that it is not necessarily representative of all the records that existed in those time periods. Out of our 20 records, 5 are by people of color. The most popular vinyl/albums are, not surprisingly, populated with mostly white male artists. This is since there is an inherent bias in society that favors the white male artist, even though there are hundreds of exceptional vinyl made by people of all races. Our database does not reflect this because of our choice to include the most popular vinyl. We learned to make it clear that our database is about the *most popular* vinyl of those decades to avoid an interpretation that could be something along the lines of "people of color were not common in the music scene during these decades".

Potential Future Work:

Future work we came up with for our database included adding a wide variety of vinyl records. Therefore, allowing our user community to expand with different types of vinyl record lovers from all eras. We also came up with adding more vinyl records more effectively instead of importing them manually with a CSV file. For instance, we could use the power of a programming language such as Python to increase our productivity for our database. The python script would allow us to connect to the database and add information from websites such as Discogs and Amazon by using the concept of web scraping/data extraction. Finally, we could also add upcoming vinyl information from various other music services such as Apple Music's and Spotify's APIs, so our database is up to date with the latest information on vinyl records.

References

Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more.

(2019). Amazon.Com. <https://www.amazon.com/>

Discogs - Music Database and Marketplace. (2010). Discogs.Com. <https://www.discogs.com/>

Tryggvason, K. (n.d.). *Disconest - Musical metadata for your Records.* Www.Disconest.Com.

<http://www.disconest.com/>

VINYL RECORDS AND CD MARKETPLACE. (n.d.). Retrieved October 16, 2020, from

<https://www.musicstack.com/>

Wikipedia. (2014). Wikipedia.Org. <https://www.wikipedia.org/>