Spotify Data Trends

INFO330 Group 9:

Meg Balfrey, June Mi Hong, Terra Sumi Shrestha, Maansi Surve

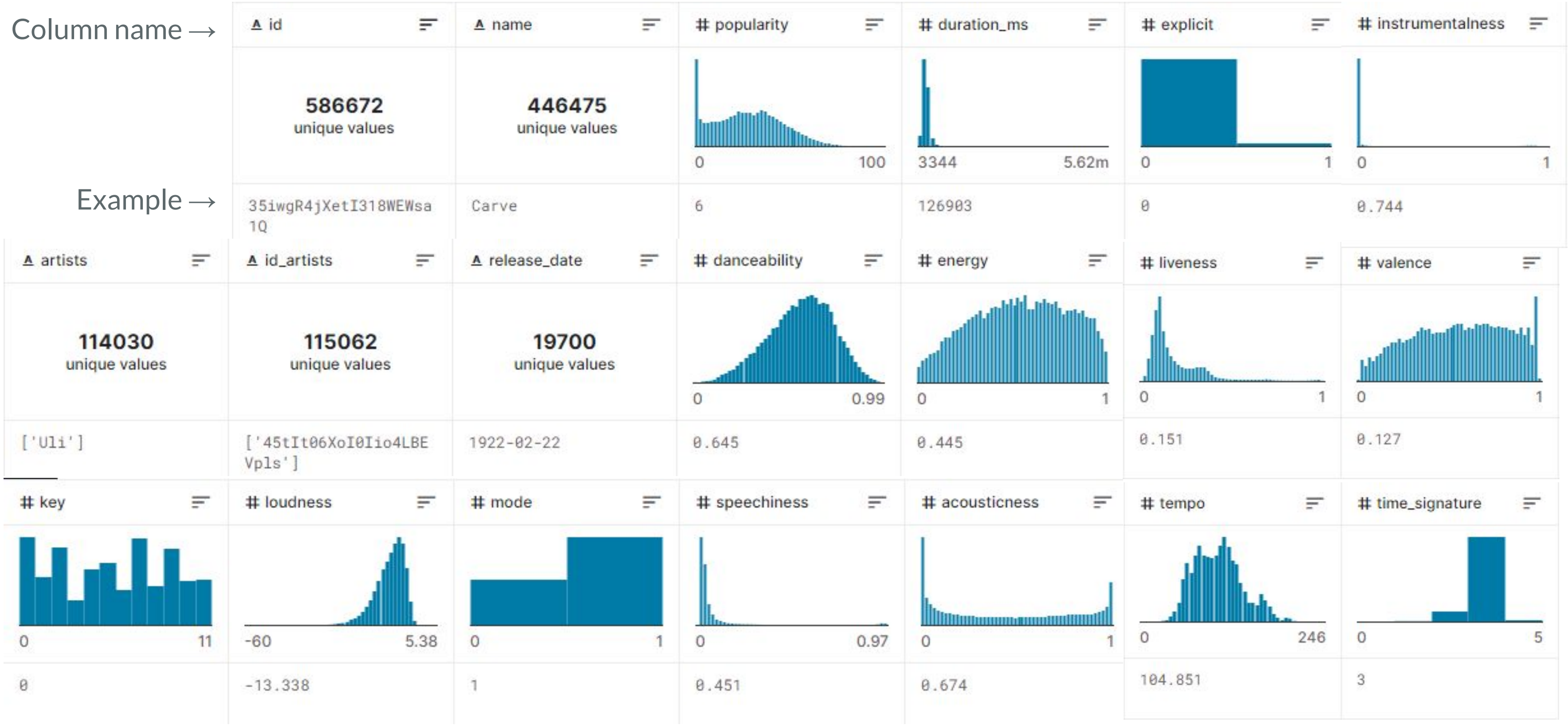
Our Dataset

- Spotify Data from Kaggle
- Covers data from the year 2000-present
- Basic track data includes song length, tempo, name, and release date
- Observational track data includes danceability, energy, speechiness, liveliness, and valence
- Also includes artist data, including name, genre, followers, and popularity

Data Examples - Tracks

Column name →

Example →

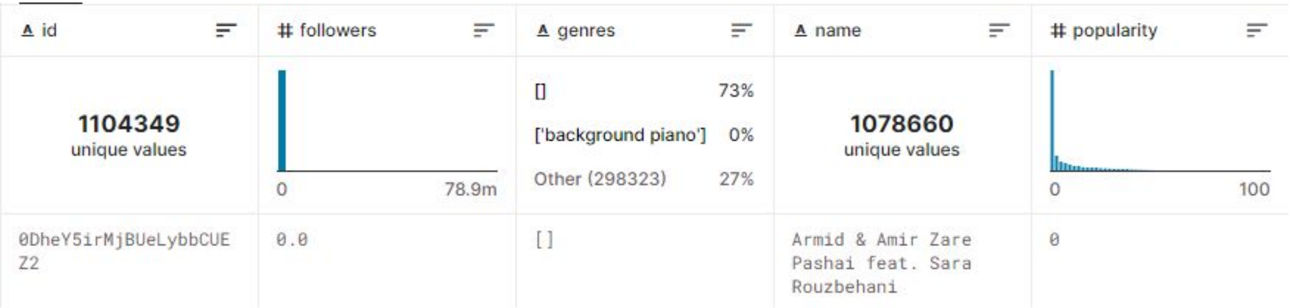


id	name	popularity	duration_ms	explicit	instrumentalness
586672 unique values	446475 unique values	0 – 100	3344 – 5.62m	0 – 1	0 – 1
35iwgR4jXetI318WEWsa1Q	Carve	6	126903	0	0.744

artists	id_artists	release_date	danceability	energy	liveness	valence
114030 unique values	115062 unique values	19700 unique values	0 – 0.99	0 – 1	0 – 1	0 – 1
['Uli']	['45tIt06XoI0Iio4LBEVpls']	1922-02-22	0.645	0.445	0.151	0.127

key	loudness	mode	speechiness	acousticness	tempo	time_signature
0 – 11	-60 – 5.38	0 – 1	0 – 0.97	0 – 1	0 – 246	0 – 5
0	-13.338	1	0.451	0.674	104.851	3

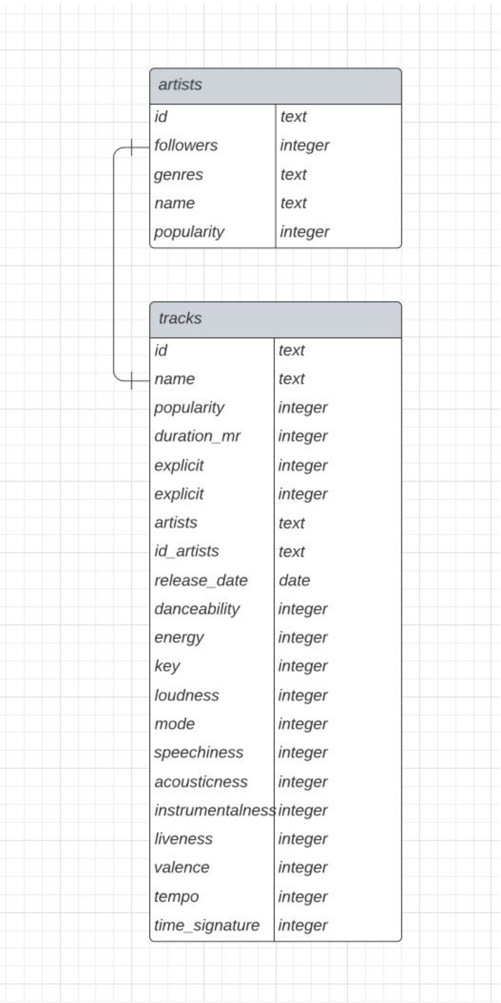
Data Examples - Artists



Column name →	id	followers	genres	name	popularity
	1104349 unique values	0 78.9m	[] 73% ['background piano'] 0% Other (298323) 27%	1078660 unique values	0 100
Example →	0DheY5irMjBUeLybbCUEZ2	0.0	[]	Armid & Amir Zare Pashai feat. Sara Rouzbehani	0

Our Schema (Overview)

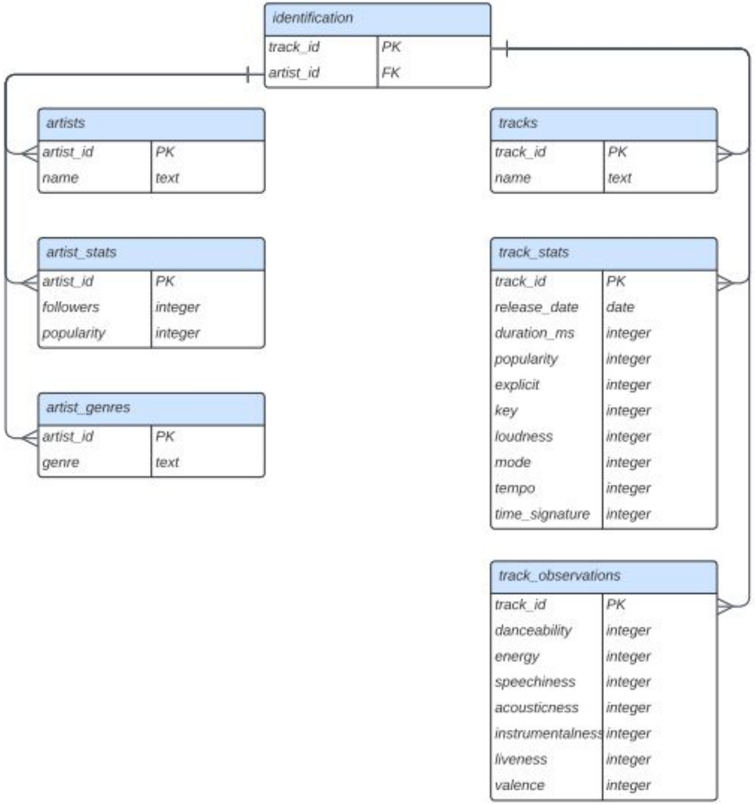
- We started with two tables
  - Artists and tracks
- Identified artist_id as the PK in the two tables
  - Used to convert to 3NF



artists	
id	text
followers	integer
genres	text
name	text
popularity	integer

tracks	
id	text
name	text
popularity	integer
duration_mr	integer
explicit	integer
explicit	integer
artists	text
id_artists	text
release_date	date
danceability	integer
energy	integer
key	integer
loudness	integer
mode	integer
speechiness	integer
acousticness	integer
instrumentalness	integer
liveness	integer
valence	integer
tempo	integer
time_signature	integer

Our Schema (Overview)



identification	
track_id	PK
artist_id	FK

artists	
artist_id	PK
name	text

artist_stats	
artist_id	PK
followers	integer
popularity	integer

artist_genres	
artist_id	PK
genre	text

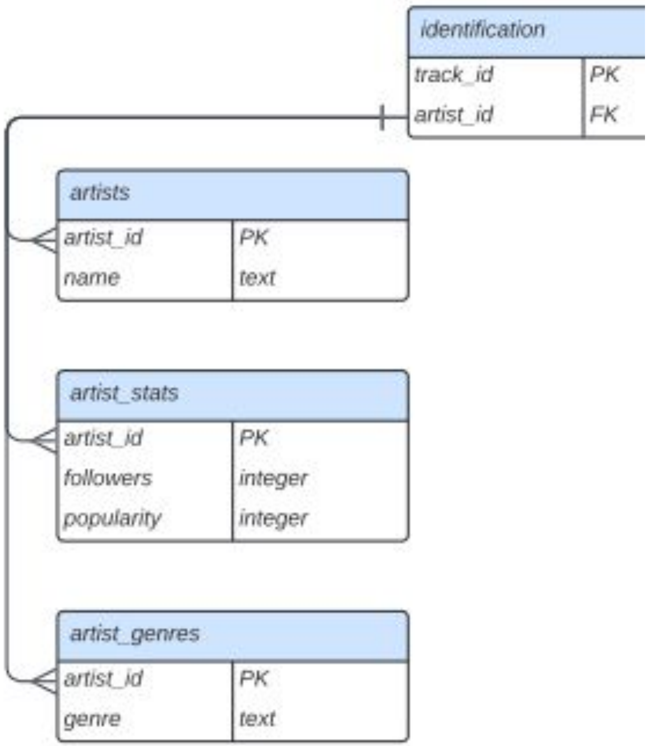
tracks	
track_id	PK
name	text

track_stats	
track_id	PK
release_date	date
duration_ms	integer
popularity	integer
explicit	integer
key	integer
loudness	integer
mode	integer
tempo	integer
time_signature	integer

track_observations	
track_id	PK
danceability	integer
energy	integer
speechiness	integer
acousticness	integer
instrumentalness	integer
liveness	integer
valence	integer

- We divided the data to separate transitive and partial dependencies
- Aimed to reduce as many redundant rows as possible
- Clarified keys

Our Schema (Breakdown)



identification	
track_id	PK
artist_id	FK

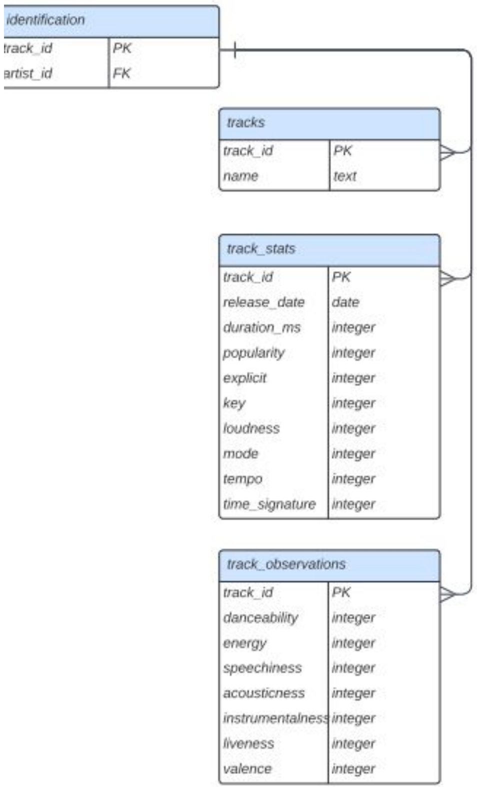
artists	
artist_id	PK
name	text

artist_stats	
artist_id	PK
followers	integer
popularity	integer

artist_genres	
artist_id	PK
genre	text

- Genre data was fully separated from the rest of artist data to prevent redundancy
- Statistics were separated from artist name to eliminate potential transitive dependencies

Our Schema (Breakdown)



identification	
track_id	PK
artist_id	FK

tracks	
track_id	PK
name	text

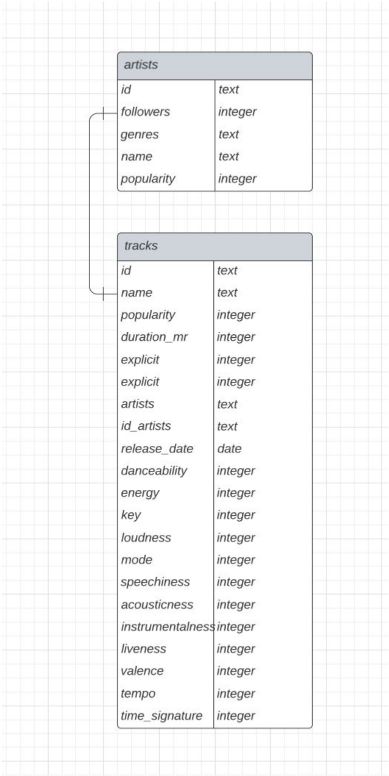
track_stats	
track_id	PK
release_date	date
duration_ms	integer
popularity	integer
explicit	integer
key	integer
loudness	integer
mode	integer
tempo	integer
time_signature	integer

track_observations	
track_id	PK
danceability	integer
energy	integer
speechiness	integer
acousticness	integer
instrumentalness	integer
liveness	integer
valence	integer

- Statistics were separated from artist name to eliminate potential transitive dependencies
- Observational and factual track statistics may be transitively dependent, so they were also separated

Challenges

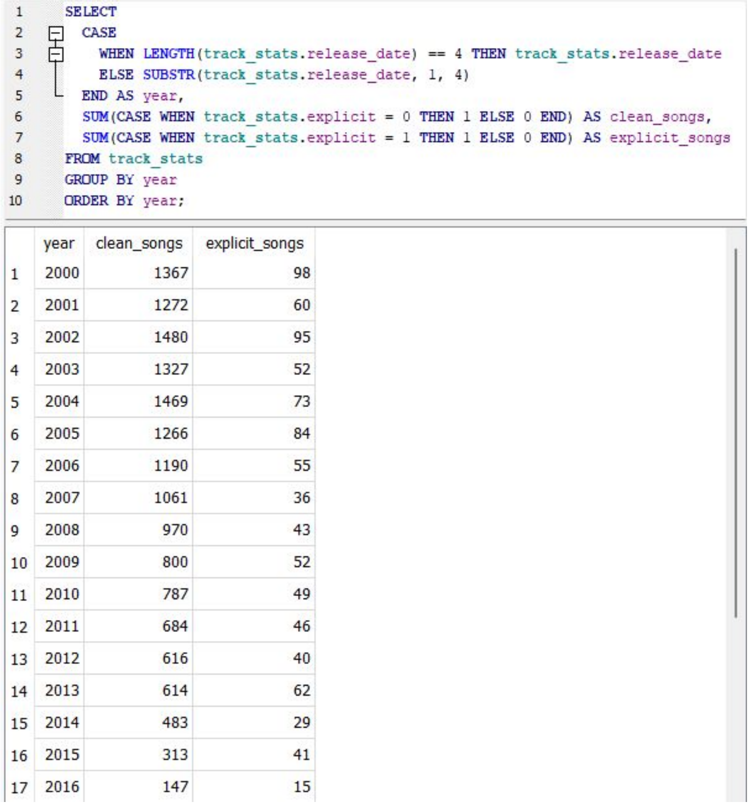
- Genres and artists
    - Separate those two columns to reach 1NF
- Creating a schema helped us visualize



artists	
id	text
followers	integer
genres	text
name	text
popularity	integer

tracks	
id	text
name	text
popularity	integer
duration_mr	integer
explicit	integer
explicit	integer
artists	text
id_artists	text
release_date	date
danceability	integer
energy	integer
key	integer
loudness	integer
mode	integer
speechiness	integer
acousticness	integer
instrumentalness	integer
liveness	integer
valence	integer
tempo	integer
time_signature	integer

Query 1a

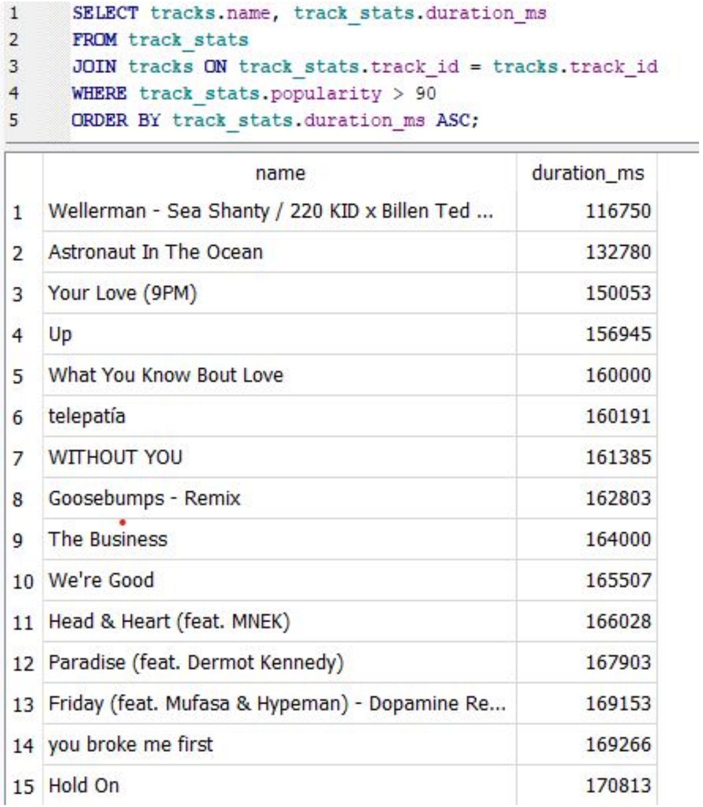


```sql
SELECT
  CASE
    WHEN LENGTH(track_stats.release_date) == 4 THEN track_stats.release_date
    ELSE SUBSTR(track_stats.release_date, 1, 4)
  END AS year,
  SUM(CASE WHEN track_stats.explicit = 0 THEN 1 ELSE 0 END) AS clean_songs,
  SUM(CASE WHEN track_stats.explicit = 1 THEN 1 ELSE 0 END) AS explicit_songs
FROM track_stats
GROUP BY year
ORDER BY year;
```

	year	clean_songs	explicit_songs
1	2000	1367	98
2	2001	1272	60
3	2002	1480	95
4	2003	1327	52
5	2004	1469	73
6	2005	1266	84
7	2006	1190	55
8	2007	1061	36
9	2008	970	43
10	2009	800	52
11	2010	787	49
12	2011	684	46
13	2012	616	40
14	2013	614	62
15	2014	483	29
16	2015	313	41
17	2016	147	15

- Comparing the number of clean and explicit songs released each year (since 2000)
- Sorted by year
- Can be used to find trends in song content and new norms

Query 1b

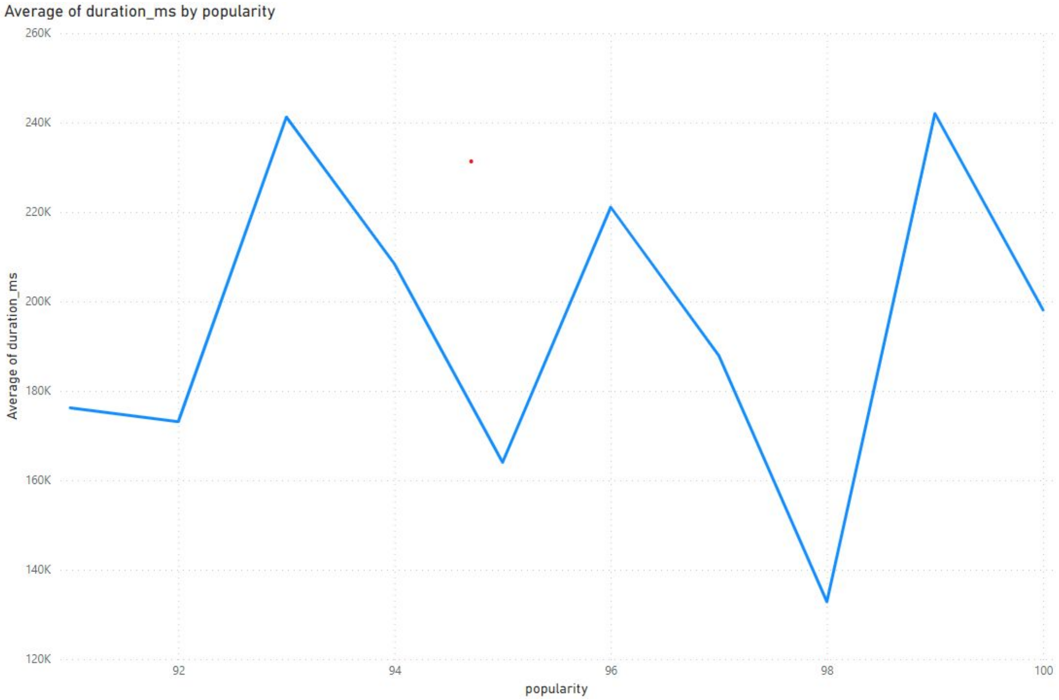


```
SELECT tracks.name, track_stats.duration_ms
FROM track_stats
JOIN tracks ON track_stats.track_id = tracks.track_id
WHERE track_stats.popularity > 90
ORDER BY track_stats.duration_ms ASC;
```

	name	duration_ms
1	Wellerman - Sea Shanty / 220 KID x Billen Ted ...	116750
2	Astronaut In The Ocean	132780
3	Your Love (9PM)	150053
4	Up	156945
5	What You Know Bout Love	160000
6	telepatía	160191
7	WITHOUT YOU	161385
8	Goosebumps - Remix	162803
9	The Business	164000
10	We're Good	165507
11	Head & Heart (feat. MNEK)	166028
12	Paradise (feat. Dermot Kennedy)	167903
13	Friday (feat. Mufasa & Hypeman) - Dopamine Re...	169153
14	you broke me first	169266
15	Hold On	170813

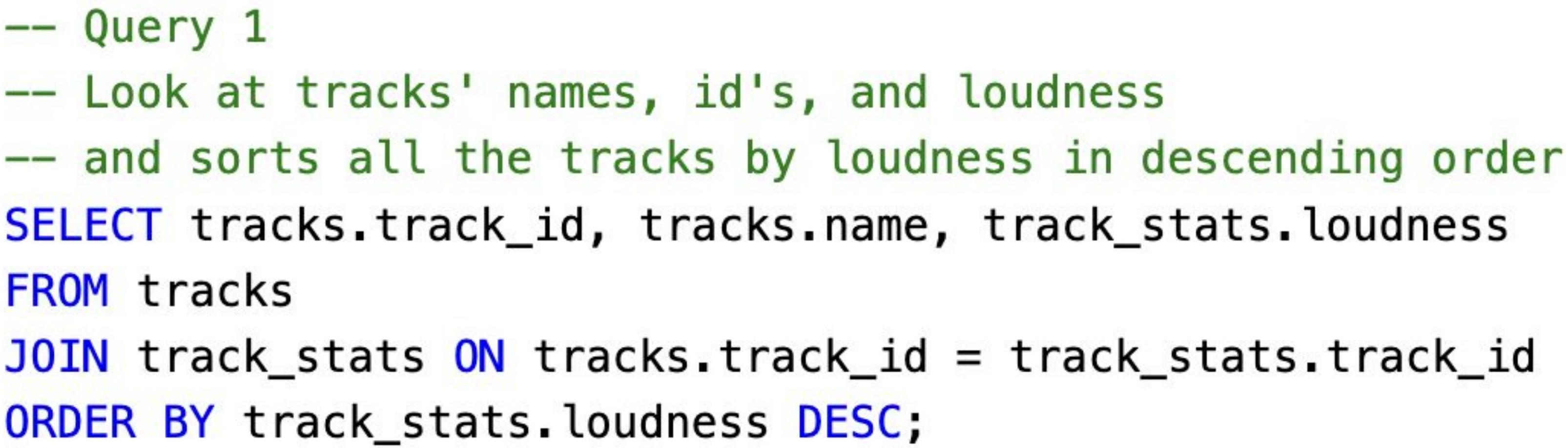
- Finding the ideal song length
- Finds tracks with popularity between 90-100, their name, and duration
- Sorted from shortest length to longest

Visualization 1



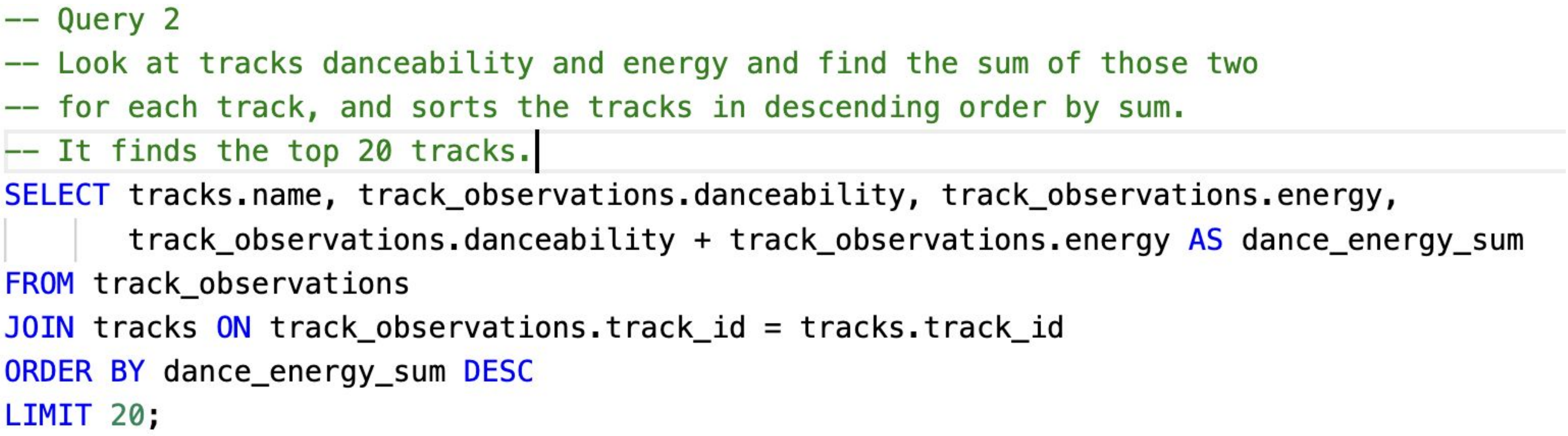
- Finding ideal song length amongst most popular tracks
- Average song length amongst songs with a popularity score of over 90/100

Query 2a



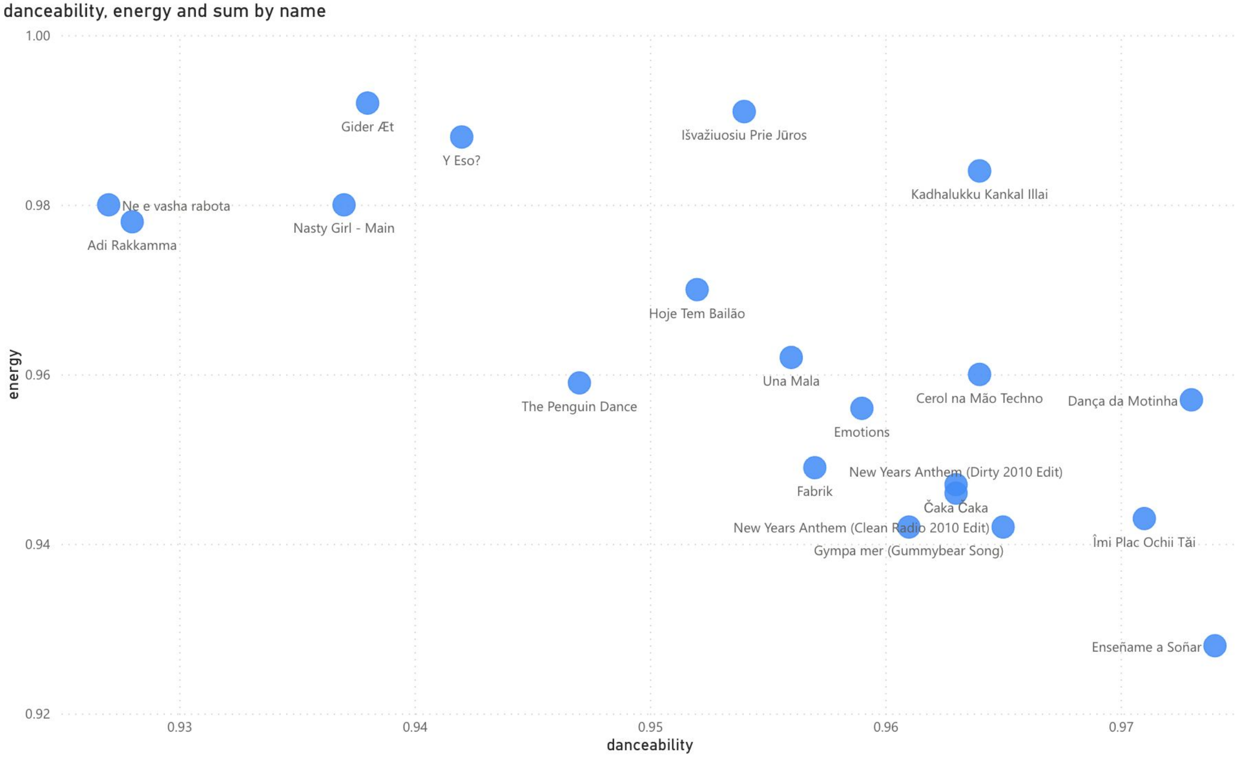
```
-- Query 1
-- Look at tracks' names, id's, and loudness
-- and sorts all the tracks by loudness in descending order
SELECT tracks.track_id, tracks.name, track_stats.loudness
FROM tracks
JOIN track_stats ON tracks.track_id = track_stats.track_id
ORDER BY track_stats.loudness DESC;
```

Query 2b

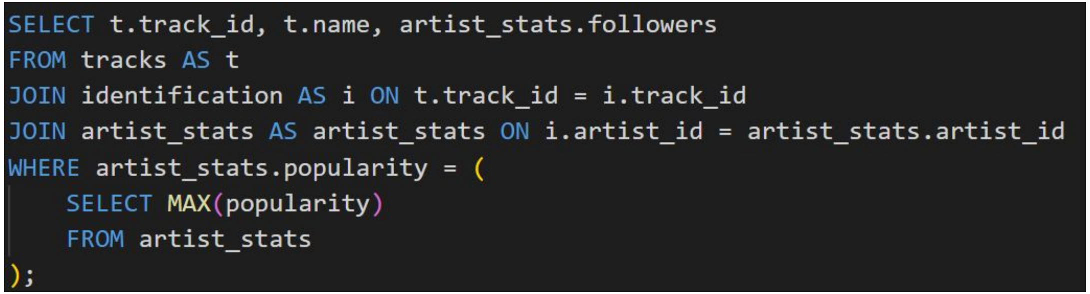


```
-- Query 2
-- Look at tracks danceability and energy and find the sum of those two
-- for each track, and sorts the tracks in descending order by sum.
-- It finds the top 20 tracks.
SELECT tracks.name, track_observations.danceability, track_observations.energy,
       track_observations.danceability + track_observations.energy AS dance_energy_sum
FROM track_observations
JOIN tracks ON track_observations.track_id = tracks.track_id
ORDER BY dance_energy_sum DESC
LIMIT 20;
```

Visualization 2



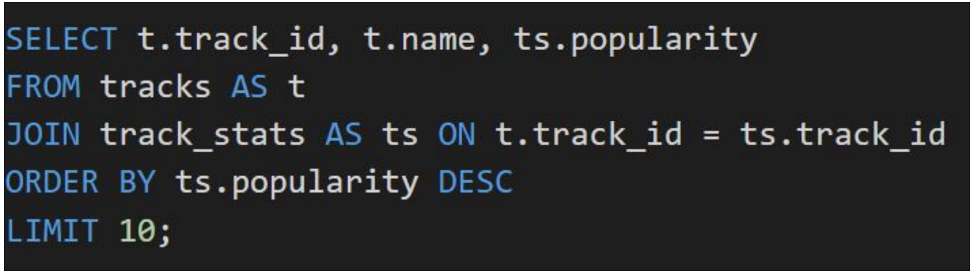
Query 3a



```sql
SELECT t.track_id, t.name, artist_stats.followers
FROM tracks AS t
JOIN identification AS i ON t.track_id = i.track_id
JOIN artist_stats AS artist_stats ON i.artist_id = artist_stats.artist_id
WHERE artist_stats.popularity = (
    SELECT MAX(popularity)
    FROM artist_stats
);
```

- Finding the correlated track id, name, and followers from artists with the maximum popularity number

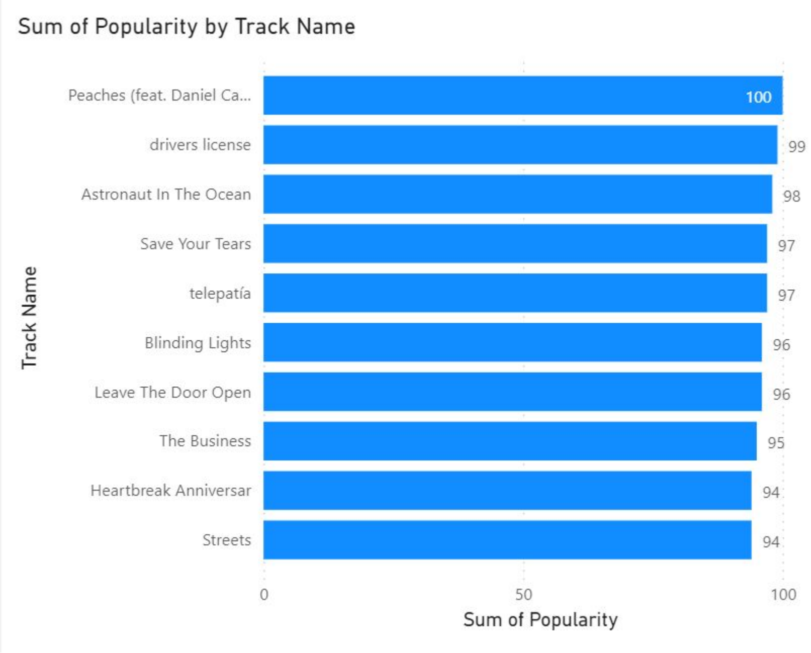
Query 3b



```sql
SELECT t.track_id, t.name, ts.popularity
FROM tracks AS t
JOIN track_stats AS ts ON t.track_id = ts.track_id
ORDER BY ts.popularity DESC
LIMIT 10;
```

- Finding the top 10 tracks with the highest popularity numbers

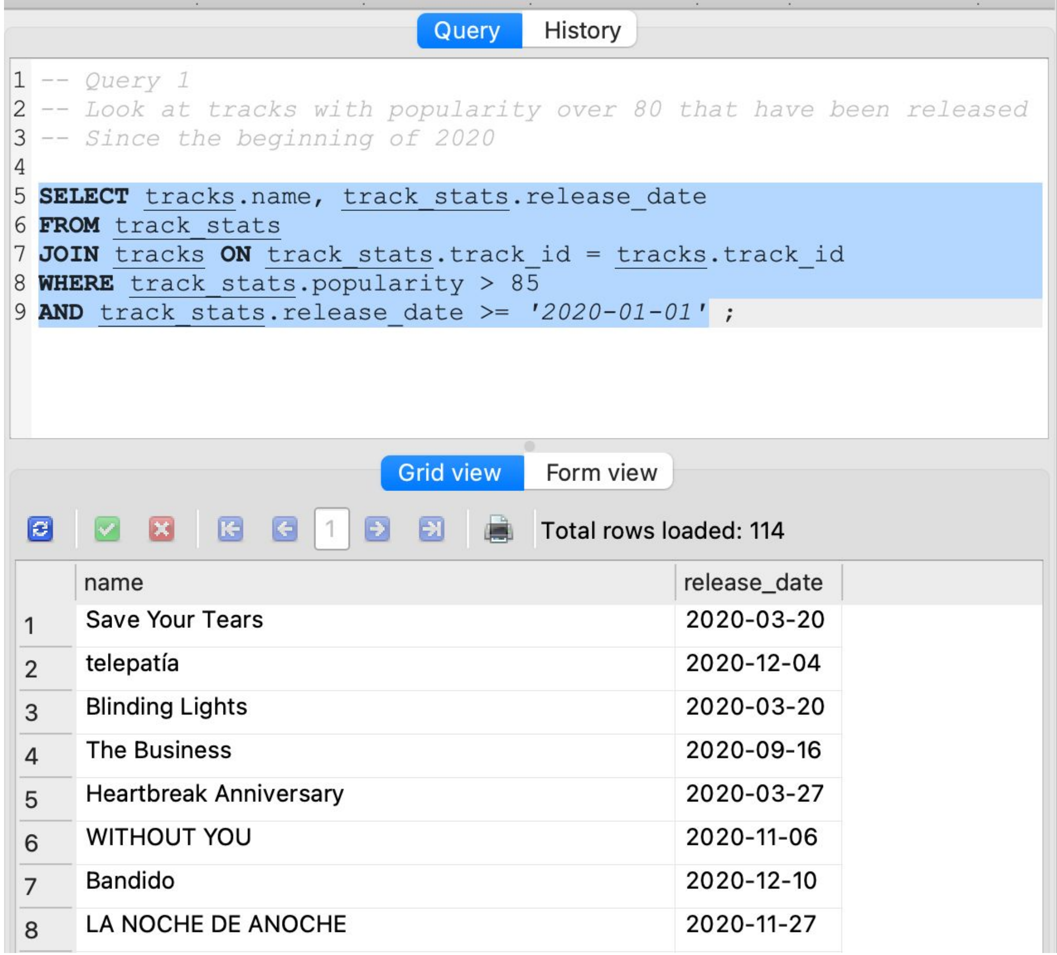
Visualization 3



Sum of Popularity by Track Name

Query 4a

- Look at the most popular songs from 2020 and their release date
- Helpful for music producers looking at recent trends



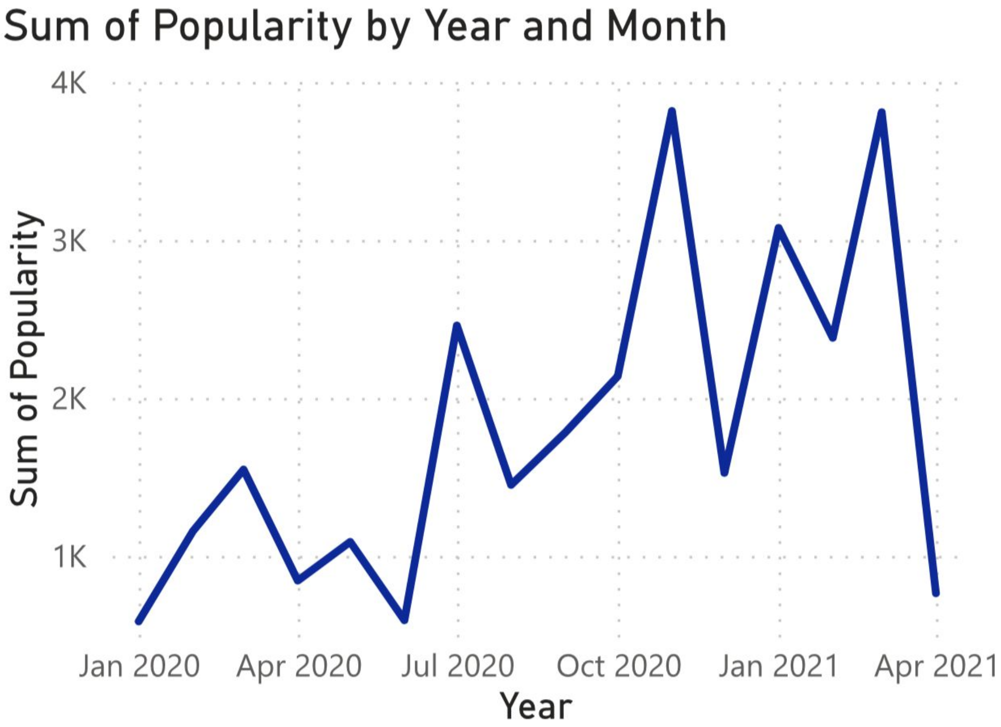
```
-- Query 1
-- Look at tracks with popularity over 80 that have been released
-- Since the beginning of 2020

SELECT tracks.name, track_stats.release_date
FROM track_stats
JOIN tracks ON track_stats.track_id = tracks.track_id
WHERE track_stats.popularity > 85
AND track_stats.release_date >= '2020-01-01' ;
```

Total rows loaded: 114

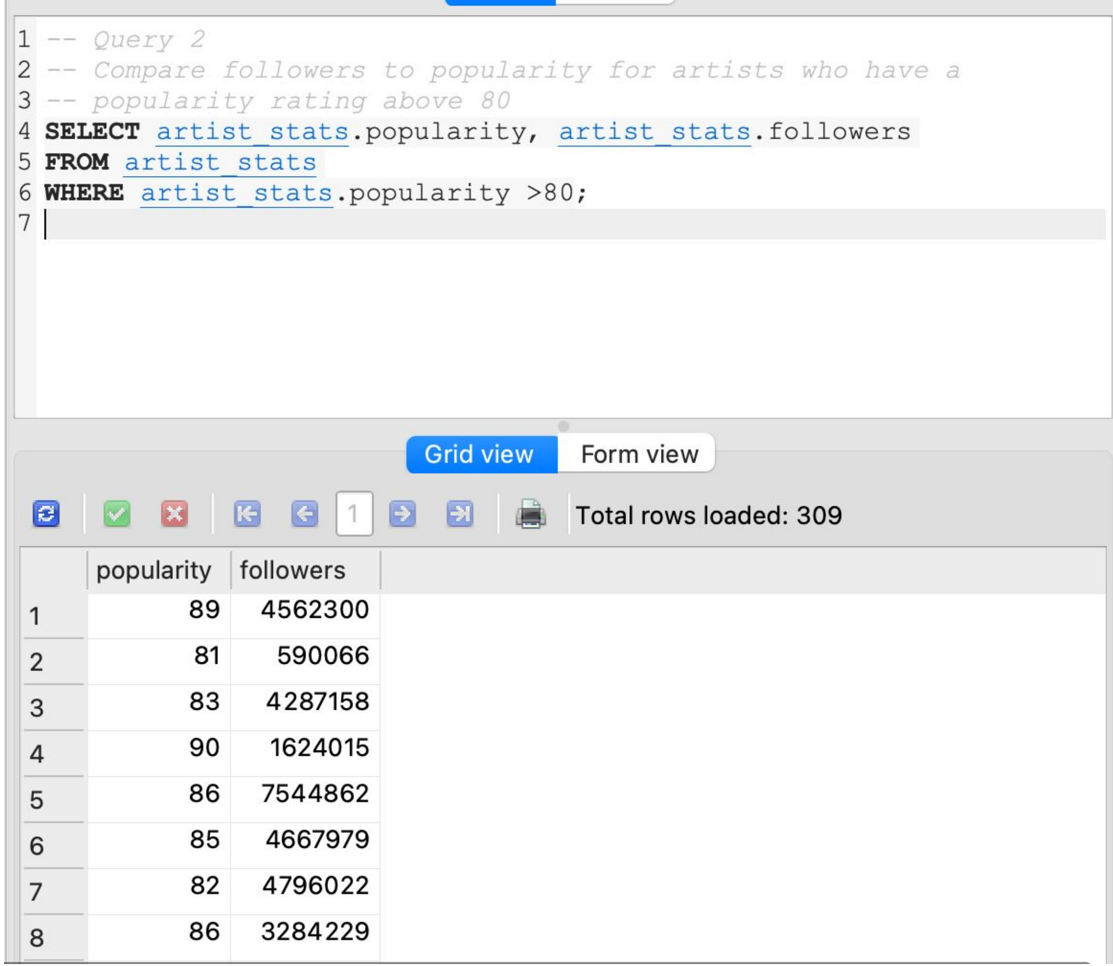
	name	release_date
1	Save Your Tears	2020-03-20
2	telepatía	2020-12-04
3	Blinding Lights	2020-03-20
4	The Business	2020-09-16
5	Heartbreak Anniversary	2020-03-27
6	WITHOUT YOU	2020-11-06
7	Bandido	2020-12-10
8	LA NOCHE DE ANOCHE	2020-11-27

Visualization 4A



Query 4b

- Look at popularity and followers on Spotify to look for a correlation
- Helps music producers look for artists who are trendings but less well known

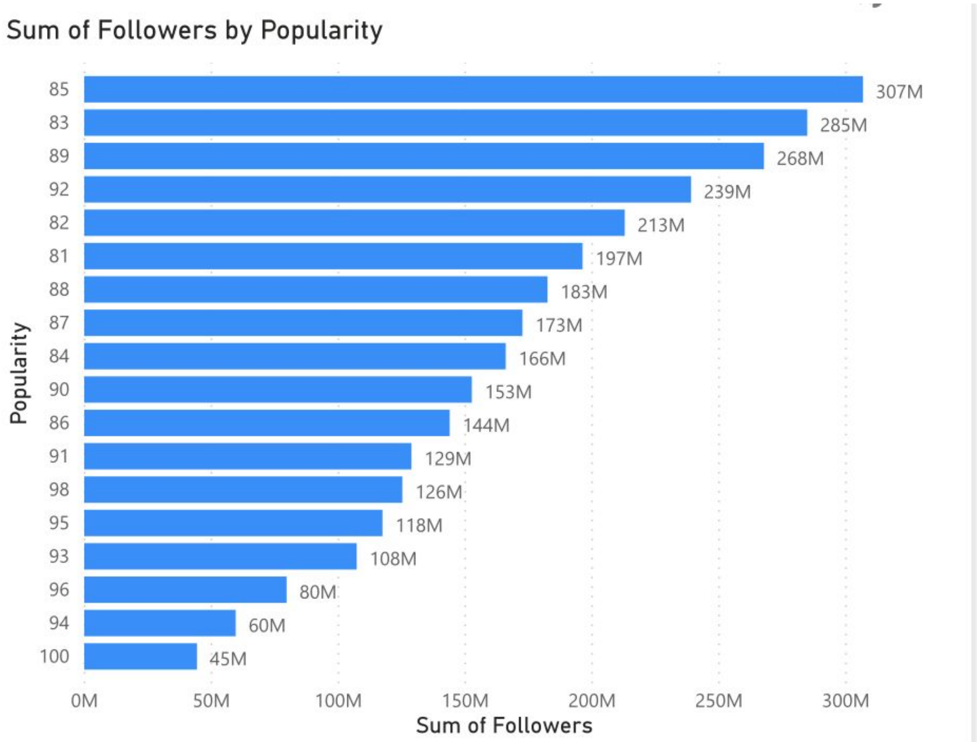


```
-- Query 2
-- Compare followers to popularity for artists who have a
-- popularity rating above 80
SELECT artist_stats.popularity, artist_stats.followers
FROM artist_stats
WHERE artist_stats.popularity >80;
```

Total rows loaded: 309

	popularity	followers
1	89	4562300
2	81	590066
3	83	4287158
4	90	1624015
5	86	7544862
6	85	4667979
7	82	4796022
8	86	3284229

Visualization 4B


Sum of Followers by Popularity
307M
285M
268M
239M
213M
197M
183M
173M
166M
153M
144M
129M
126M
118M
108M
80M
60M
45M
0M
50M
100M
150M
200M
250M
300M
Popularity
Sum of Followers