

Analyzing Spotify Data with SQL

Introduction:

This project focuses on analyzing Spotify's music data using SQL queries to uncover insights into track performance, artist engagement, and user preferences. The queries cover various aspects, such as identifying top tracks based on views and streams, calculating cumulative sums of likes, and exploring metrics like energy-to-liveness ratios. By applying SQL techniques like aggregation, window functions, and subqueries, this project aims to provide a deeper understanding of Spotify's catalog and user behaviour.

Some Exploratory Data Analysis

```
CREATE TABLE spotify (  
  artist VARCHAR(300),  
  track VARCHAR(300),  
  album VARCHAR(300),  
  album_type VARCHAR(100),  
  danceability FLOAT,  
  energy FLOAT,  
  loudness FLOAT,  
  speechiness FLOAT,  
  acousticness FLOAT,  
  instrumentalness FLOAT,  
  liveness FLOAT,  
  valence FLOAT,  
  tempo FLOAT,  
  duration_min FLOAT,  
  title VARCHAR(300),  
  channel VARCHAR(300),  
  views FLOAT,  
  likes BIGINT,  
  comments BIGINT,  
  licensed BOOLEAN,  
  official_video BOOLEAN,  
  stream BIGINT,  
  energy_liveness FLOAT,  
  most_played_on VARCHAR(100)  
);
```

```
SELECT * FROM spotify
```

```
SELECT COUNT(*) FROM spotify;
```

```
SELECT COUNT(DISTINCT artist) FROM spotify;
```

```
SELECT COUNT(DISTINCT album) FROM spotify;
```

```
SELECT DISTINCT album_type FROM spotify;
```

```
SELECT MAX(duration_min) FROM spotify;  
SELECT MIN(duration_min) FROM spotify;
```

```
SELECT * FROM spotify  
WHERE duration_min = 0;
```

```
DELETE FROM spotify  
WHERE duration_min = 0;
```

```
SELECT * FROM spotify  
WHERE duration_min = 0;
```

```
SELECT DISTINCT channel FROM spotify;
```

Key Insights from Spotify Using SQL Queries

1. Retrieve the names of all tracks that have more than 1 billion streams.

```
SELECT * FROM spotify
WHERE stream > 1000000000;
```

		channel character varying (300)	views double precision	likes bigint	comments bigint	licensed boolean	official_video boolean	stream bigint	energy_liveness double precision	most_play character
1		Gorillaz	693555221	6220896	169907	true	true	1040234854	1.150081566	Spotify
2	PGRAD[E]	Red Hot Chili Peppers	1018811259	4394471	121452	true	true	1055738398	6.039370079	Spotify
3		Red Hot Chili Peppers	246687714	1213572	32761	true	true	1061750522	2.446808511	Spotify
4		50CentVEVO	1682616458	10481678	296745	true	true	1041736808	9.612817089	Youtube
5		Coldplay	832532409	4600933	118296	true	true	1483622308	2.824786325	Spotify
6		Coldplay	789581468	4370461	261790	true	true	1325575712	5.660550459	Spotify
7		ChainsmokersVEVO	2118018686	10282499	270444	true	true	2030825694	3.87195122	Youtube
8		Coldplay	1082588109	5532787	124357	true	true	1465980600	4.018181818	Spotify
9		Coldplay	786404613	3741300	79974	true	true	1021782403	3.229665072	Spotify
10		Coldplay	1828242112	13515772	377666	true	true	1049654852	2.132307692	Youtube

Total rows: 385 of 385 Query complete 00:00:00.170 Ln 59, Col 26

2. List all albums along with their respective artists.

```
SELECT
  DISTINCT album, artist
FROM spotify
ORDER BY 1;
```

	album character varying (300)	artist character varying (300)
1	'Justments	Bill Withers
2	'N Sync UK Version	*NSYNC
3	'The Sounds of Nightwish Reborn: Early Demos for "Dark Passion Play" and B-Sides'	Nightwish
4	- TRAGEDY +	\$NOT
5	!Volare! The Very Best of the Gipsy Kings	Gipsy Kings
6	"Awaken, My Love!"	Childish Gambino
7	"Heroes" (2017 Remaster)	David Bowie
8	"Let's Rock"	The Black Keys
9	"Let Go" Di Paolo	Dina Rae

Total rows: 1000 of 14178 Query complete 00:00:00.173 Ln 61, Col 1

3. Get the total number of comments for tracks where `licensed = TRUE`.

```
SELECT SUM(comments) AS total_comments
FROM spotify
WHERE licensed = TRUE;
```

	total_comments numeric
1	497015695

4. Find all tracks that belong to the album type `single`.

```
SELECT * FROM spotify
WHERE album_type = 'single';
```

	album_type	danceability
ng (300)	character varying (100)	double precision
1 t. Tame Impala and Bootie Brown)	single	0.695
2 t. Tame Impala and Bootie Brown) [Dom Dolla Remix]	single	0.716
3 (feat. Thundercat)	single	0.741
4 emix)	single	0.545
5	single	0.386
6 cide	single	0.425
7 t. Pharrell Williams & Nile Rodgers) [Radio Edit]	single	0.794
8 e: MTV Ultimate Mash-Ups Presents Collision Course	single	0.687

5. Count the total number of tracks by each artist.

```
SELECT artist, COUNT(*) AS total_tracks
FROM spotify
GROUP BY artist
ORDER BY total_tracks DESC;
```

	artist	total_tracks
	character varying (300)	bigint
1	Marisela	10
2	Ray Charles	10
3	Joey Bada\$\$	10
4	La Mosca Tse-Tse	10
5	TheFatRat	10
6	Leo Dan	10
7	Tyga	10
8	Grupo Marca Registrada	10
9	Tyler, The Creator	10

6. Calculate the average danceability of tracks in each album.

```
SELECT album, AVG(danceability) AS average_danceability
FROM spotify
GROUP BY album
ORDER BY average_danceability DESC;
```

	album	average_danceability
	character varying (300)	double precision
1	Funky Friday	0.975
2	FOR CERTAIN (Deluxe)	0.975
3	The House Is Burning [homies begged]	0.971
4	Quality Control: Control The Streets Volume 2	0.97
5	N9NA	0.97
6	Aka Entre el Humo	0.967
7	Peru (Remix)	0.966
8	Peru (feat. Medusa)	0.964

7. Find the top 5 tracks with the highest energy values.

```
SELECT track, energy
FROM spotify
ORDER BY energy DESC
LIMIT 5;
```

	track character varying (300) 🔒	energy double precision 🔒
1	Rain and Thunderstorm, Pt. 6	1
2	Rain and Thunderstorm, Pt. 7	1
3	Rain and Thunderstorm, Pt. 4	1
4	Rain and Thunderstorm, Pt. 3	1
5	Rain and Thunderstorm, Pt. 33	1

8. List all tracks along with their views and likes where `official_video = TRUE`.

```
SELECT track,
SUM(views) AS total_views,
SUM(likes) AS total_likes
FROM spotify
WHERE official_video = TRUE
GROUP BY track
ORDER BY total_views DESC
LIMIT 5;
```

	track character varying (300) 🔒	total_views double precision 🔒	total_likes numeric 🔒
1	Despacito	16159296273	101577278
2	See You Again (feat. Charlie Puth)	11547595554	80295292
3	Shape of You	5908398479	31047780
4	Calma - Remix	5322011392	25649519
5	This Is What You Came For	5252059812	21207312

9. For each album, calculate the total views of all associated tracks.

```
SELECT album, track, SUM(views) AS total_views
FROM spotify
GROUP BY album, track
ORDER BY total_views DESC;
```

	album character varying (300)	track character varying (300)	total_views double precision
1	VIDA	Despacito	16159296273
2	See You Again (feat. Charlie Puth)	See You Again (feat. Charlie Puth)	11547595554
3	Peace Is The Mission (Extended)	Lean On	9974504694
4	÷ (Deluxe)	Shape of You	5908398479
5	MUNAY	Calma - Remix	5322011392
6	This Is What You Came For	This Is What You Came For	5252059812
7	Swalla (feat. Nicki Minaj & Ty Dolla \$ign)	Swalla (feat. Nicki Minaj & Ty Dolla \$ign)	5162403752
8	CoComelon Kids Hits, Vol. 1	Wheels on the Bus	4898831101
9	Taki Taki (with Selena Gomez, Ozuna & Cardi B)	Taki Taki (with Selena Gomez, Ozuna & Cardi B)	4828403435

10. Retrieve the track names that have been streamed on Spotify more than YouTube.

```
SELECT * FROM
(SELECT track,
COALESCE(SUM(CASE WHEN most_played_on = 'Youtube' THEN stream END), 0)
AS streamed_on_youtube,
COALESCE(SUM(CASE WHEN most_played_on = 'Spotify' THEN stream END), 0) AS
streamed_on_spotify
FROM spotify
GROUP BY 1
) AS t1
WHERE
streamed_on_spotify > streamed_on_youtube
AND
streamed_on_youtube <> 0;
```

	track character varying (300)	streamed_on_youtube numeric	streamed_on_spotify numeric
1	Usted	30059201	137916795
2	21 Hungarian Dances, WoO 1: Hungarian Dance No. 5 in G Minor. Allegro (Orch. Schmeli...	39575743	79151486
3	Mientes Tan Bien	6915455	224299945
4	Have You Ever Seen The Rain	61903001	975300588
5	Dream A Little Dream Of Me	157256901	495674374
6	When I Grow Up	231236307	260959663
7	Me Hace Tanto Bien	56694580	187498268
8	What You Want (feat. Tyla)	12000000	85450215

Total rows: 155 of 155 Query complete 00:00:00.106 Ln 44, Col 1

11. Find the top 3 most-viewed tracks for each artist using window functions.

```

WITH ranking_artist
AS
(SELECT artist, track,
        SUM(views) as total_view,
        DENSE_RANK() OVER(PARTITION BY artist ORDER BY SUM(views)
DESC) as rank
FROM spotify
GROUP BY 1,2
ORDER BY 1, 3 DESC
)
SELECT * FROM ranking_artist
WHERE rank <=3;

```

	artist character varying (300)	track character varying (300)	total_view double precision	rank bigint
1	\$NOT	Tell Em	41100657	1
2	\$NOT	Like Me (feat. iann dior)	15803517	2
3	\$NOT	Mean	13563870	3
4	\$uicideboy\$	Paris	175156959	1
5	\$uicideboy\$	For the Last Time	91771038	2
6	\$uicideboy\$	Kill Yourself (Part III)	31674988	3
7	(G)I-DLE	POP/STARS	541864471	1
8	(G)I-DLE	TOMBOY	204797335	2
9	(G)I-DLE	Oh my god	191587399	3
10	*NSYNC	Bye Bye Bye	315874581	1
11	*NSYNC	This I Promise You	252377383	2
Total rows: 1000 of 6808 Query complete 00:00:00.212 Ln 57, Col 1				

12. Write a query to find tracks where the liveness score is above the average.

```

SELECT track, artist, liveness
FROM spotify
WHERE liveness > (SELECT AVG(liveness) FROM spotify);

```

	track character varying (300)	artist character varying (300)	liveness double precision
1	Feel Good Inc.	Gorillaz	0.613
2	DARE	Gorillaz	0.298
3	New Gold (feat. Tame Impala and Bootie Brown) - Dom Dolla Remix	Gorillaz	0.325
4	Cracker Island (feat. Thundercat)	Gorillaz	0.325
5	Dirty Harry	Gorillaz	0.672
6	Dani California	Red Hot Chili Peppers	0.346
7	Candy Shop	50 Cent	0.38
8	Just A Lil Bit	50 Cent	0.315
9	Disco Inferno	50 Cent	0.279

13. Use a `WITH` clause to calculate the difference between the highest and lowest energy values for tracks in each album.

```
WITH energy_stats AS (
  SELECT
    album,
    MAX(energy) AS max_energy,
    MIN(energy) AS min_energy
  FROM spotify
  GROUP BY album
)
SELECT
  album,
  max_energy,
  min_energy,
  (max_energy - min_energy) AS energy_difference
FROM energy_stats
ORDER BY energy_difference DESC;
```

	album character varying (300)	max_energy double precision	min_energy double precision	energy_difference double precision
1	White Noise	0.908	0.00125	0.9067500000000001
2	Spotify Singles - Holiday	0.904	0.068	0.8360000000000001
3	Spotify Singles	0.894	0.0708	0.8232
4	UNDERTALE Soundtrack	0.978	0.162	0.816
5	Making Mirrors	0.9	0.0891	0.8109000000000001
6	Everytime We Touch (Premium Edition)	0.978	0.173	0.8049999999999999
7	If I Can Dream: Elvis Presley with the Royal Philharmonic Orchestra	0.936	0.149	0.787
8	Fallen	0.943	0.178	0.7649999999999999
9	CeeLo's Magic Moment	0.953	0.191	0.762
10	Greatest Hits	0.885	0.144	0.741

14. Find tracks where the energy-to-liveness ratio is greater than 1.2.

```
SELECT track, energy, liveness, (energy / liveness) AS energy_to_liveness_ratio
FROM spotify
WHERE (energy / liveness) > 1.2;
```

	track character varying (300)	energy double precision	liveness double precision	energy_to_liveness_ratio double precision
1	Rhinestone Eyes	0.703	0.0463	15.183585313174945
2	New Gold (feat. Tame Impala and Bootie Brown)	0.923	0.116	7.956896551724138
3	On Melancholy Hill	0.739	0.064	11.546875
4	Clint Eastwood	0.694	0.0698	9.94269340974212
5	DARE	0.891	0.298	2.9899328859060406
6	New Gold (feat. Tame Impala and Bootie Brown) - Dom Dolla Remix	0.897	0.325	2.76
7	She's My Collar (feat. Kali Uchis)	0.815	0.112	7.2767857142857135
8	Cracker Island (feat. Thundercat)	0.913	0.325	2.809230769230769
9	Dirty Harry	0.877	0.672	1.3050595238095237
10	California	0.767	0.127	6.039370078740157

15. Calculate the cumulative sum of likes for tracks ordered by the number of views, using window functions.

SELECT track,

SUM(likes) OVER (ORDER BY views DESC) AS cumulative_sum

FROM spotify

ORDER BY cumulative_sum DESC;

	track character varying (300) 🔒	cumulative_sum numeric 🔒
1	Paartha Mudhal	13344544007
2	1, 2 Many	13344544007
3	Neon Moon	13344544007
4	los Disicipulos	13344544007
5	Pa' Que Te Lo Gozen	13344544007
6	Quitate Tu Pa Ponerme Yo	13344544007
7	Bonetrousle	13344544007
8	Vamo Tomar Uma	13344544007
9	Nem Vá	13344544007
10	S de Saudade	13344544007
11	1, 2 Many	13344544007

Conclusion:

This project demonstrates how SQL can be used to analyze large datasets from music streaming platforms like Spotify. By exploring key metrics and trends, the project offers valuable insights into track and album performance, which can help guide decisions for artists, labels, and streaming platforms. The use of advanced SQL techniques allows for efficient analysis of complex data, making it easier to uncover meaningful patterns and optimize strategies in the music industry.