**Spotify Music Platform Data Analysis Using SQL**

**-- Describe table**

DESC spotify;

**-- Show all data**

SELECT \* FROM spotify;

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

SELECT Track FROM spotify

WHERE Stream > 1000000000;

**2. List all albums along with their respective artists.**

SELECT DISTINCT Album, Artist

FROM spotify;

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

SELECT Track, SUM(Comments) AS Total\_Comments

FROM spotify

WHERE Licensed = 'True'

GROUP BY Track;

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

SELECT Album\_type, Track

FROM spotify

WHERE Album\_type = 'single';

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

SELECT Artist, COUNT(Track) AS Total\_Tracks

FROM spotify

GROUP BY Artist;

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

SELECT Album, AVG(Dancebility) AS Avg\_Danceability

FROM spotify

GROUP BY Album;

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

SELECT Track, MAX(Energy) AS Max\_Energy

FROM spotify

GROUP BY Track

ORDER BY Max\_Energy DESC

LIMIT 5;

**8. List all tracks along with their views and likes where official\_video = TRUE.**

SELECT Track, Views, Likes

FROM spotify

WHERE official\_video = 'TRUE';

**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;

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

SELECT Track

FROM spotify

WHERE most\_playedon = 'Spotify';

**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\_Views,

DENSE\_RANK() OVER (ORDER BY SUM(Views) DESC) AS Ranks

FROM spotify

GROUP BY Artist, Track

)

SELECT \*

FROM Ranking\_Artist

WHERE Ranks <= 3;

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

SELECT Track, Liveness

FROM spotify

WHERE Liveness > (SELECT AVG(Liveness) FROM spotify);

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

WITH Energy\_Difference 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 Difference

FROM Energy\_Difference;

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

SELECT Track, (Energy / Liveness) AS Energy\_Liveness\_Ratio

FROM spotify

WHERE (Energy / Liveness) > 1.2;

**15. Calculate the cumulative sum of likes, ordered by views using window functions.**

SELECT

Track,

SUM(Likes) OVER (ORDER BY Views DESC) AS Cumulative\_Likes,

Views

FROM spotify;

**Dataset Link:** **Dataset Link :**

**https://github.com/najirh/najirh-Spotify-Data-Analysis-using-SQL**