# **SQL ASSIGNMENT 5- SPOTIFY**

**CREATING SPOTIFY DATA TABLE**

**A screenshot of a computer code

Description automatically generated**

**1. Check the entire dataset**

TOTAL ROWS: 2300

TOTAL COLUMNS: 23

SELECT \* FROM SPOTIFY;

A screenshot of a computer

Description automatically generated

**2. Number of songs on Spotify for each artist**

SELECT ARTIST\_NAME, COUNT(TRACK\_ID) AS TOT\_NO\_OF\_SONGS

FROM SPOTIFY

GROUP BY ARTIST\_NAME

ORDER BY COUNT(TRACK\_ID) DESC;

A white sheet of paper with lines

Description automatically generated

**3. Top 10 songs based on popularity**

SELECT TOP 10 TRACK\_POPULARITY, TRACK\_NAME

FROM SPOTIFY

ORDER BY TRACK\_POPULARITY DESC;

A screenshot of a music list

Description automatically generated

**4. Total number of songs on spotify based on year**

SELECT YEAR, COUNT(TRACK\_ID)

FROM SPOTIFY

GROUP BY YEAR

ORDER BY YEAR;

A screenshot of a computer

Description automatically generated

**5. Top song for each year (2000-2022) based on popularity**

SELECT YEAR,MAX(TRACK\_NAME) AS TOP\_SONG, MAX(TRACK\_POPULARITY) AS SONG\_POPULARITY

FROM SPOTIFY

GROUP BY YEAR

ORDER BY YEAR;

A screenshot of a computer

Description automatically generated

**6. Analysis based on Tempo :**

**tempo > 121.08 -> 'Above Average Tempo'**

**tempo = 121.08 -> 'Average Tempo'**

**tempo < 121.08 -> 'Below Average Tempo'**

SELECT TRACK\_NAME, ROUND(TEMPO),

CASE

WHEN TEMPO > 121.08 THEN 'Above Average Tempo'

WHEN TEMPO = 121.08 THEN 'Average Tempo'

ELSE 'Below Average Tempo'

END AS TEMPO\_SCALE

FROM SPOTIFY;

A white lined paper with black text

Description automatically generated

**7. Songs with Highest Tempo**

SELECT TRACK\_NAME, MAX(TEMPO) AS HIGHEST\_TEMPO

FROM SPOTIFY

GROUP BY 1

ORDER BY MAX(TEMPO) DESC;

A screenshot of a computer

Description automatically generated

**8. Number of Songs for different Tempo Range : track\_name, energy**

**Modern\_Music -> tempo BETWEEN 60.00 AND 100.00**

**Classical\_Music -> tempo BETWEEN 100.001 AND 120.00**

**Dance\_Music -> tempo BETWEEN 120.001 AND 150.01**

**HighTempo\_Music -> tempo > 150.01**

SELECT TRACK\_NAME, ROUND(ENERGY) AS ENERGY, ROUND(TEMPO,2) AS TEMPO,

CASE

WHEN tempo BETWEEN 60.00 AND 100.00 THEN 'Modern\_Music'

WHEN tempo BETWEEN 100.001 AND 120.00 THEN 'Classical\_Music'

WHEN tempo BETWEEN 120.001 AND 150.01 THEN 'Dance\_Music'

WHEN tempo > 150.01 THEN 'HighTempo\_Music'

END AS TEMPO\_RATING

FROM SPOTIFY;

A screenshot of a data

Description automatically generated

**9. Energy Analysis : TOP 10 track\_name, danceability, track\_popularity**

**energy > 0.64 -> 'Above Average Energy**

**energy = 0.64 -> 'Average Energy’**

**energy < 0.64 -> 'Below Average Energy’**

**energy BETWEEN 0.1 AND 0.3 -> 'Calm Music'**

**energy BETWEEN 0.3 AND 0.6 -> 'Moderate Music'**

**Energy >0.6 -> ‘Energetic Music**

CREATE OR REPLACE VIEW ENERGY\_RANGE\_ANALYSIS AS

SELECT TOP 10 TRACK\_NAME, DANCEABILITY, TRACK\_POPULARITY, ENERGY,

CASE

WHEN energy > 0.64 THEN 'Above Average Energy'

WHEN energy = 0.64 THEN 'Average Energy'

WHEN energy < 0.64 THEN 'Below Average Energy'

WHEN energy BETWEEN 0.1 AND 0.3 THEN 'Calm Music'

WHEN energy BETWEEN 0.3 AND 0.6 THEN 'Moderate Music'

WHEN Energy >0.6 AND 0.6 THEN 'Energetic Music'

END AS ENERGY\_ANALYSIS

FROM SPOTIFY

ORDER BY 3 DESC;

A screenshot of a white sheet with black text

Description automatically generated

**10. Number of Songs for different energy ranges(above)**

SELECT COUNT(\*) AS SONGS\_DIFF\_ENERGY\_RANGE,

CASE

WHEN energy > 0.64 THEN 'Above Average Energy'

WHEN energy = 0.64 THEN 'Average Energy'

WHEN energy < 0.64 THEN 'Below Average Energy'

WHEN energy BETWEEN 0.1 AND 0.3 THEN 'Calm Music'

WHEN energy BETWEEN 0.3 AND 0.6 THEN 'Moderate Music'

WHEN Energy >0.6 AND 0.6 THEN 'Energetic Music'

END AS ENERGY\_ANALYSIS

FROM SPOTIFY

GROUP BY 2;

A screenshot of a cell phone

Description automatically generated

11. Danceability Analysis: Top 20 track name, danceability danceability BETWEEN 0.69 AND 0.79 -> 'Low Danceability' (danceability BETWEEN 0.49 AND 0.68) OR (danceability BETWEEN 0.79 AND 0.89) -> 'Moderate Danceability' (danceability BETWEEN 0.39 AND 0.49) OR (danceability BETWEEN 0.89 AND 0.99) -> 'High Danceability' danceability < 0.39 OR danceability > 0.99 -> 'Can’t Dance on this one

A screenshot of a music list

Description automatically generated

**12. Number of Songs for different danceability ranges(above)**

SELECT count(\*) AS SONG\_DIFF\_DANCEABILITY,

CASE

WHEN DANCEABILITY BETWEEN 0.69 AND 0.79 THEN 'Low Danceability'

WHEN (DANCEABILITY BETWEEN 0.49 AND 0.68) OR (DANCEABILITY BETWEEN 0.79 AND

0.89) THEN 'Moderate Danceability'

WHEN (DANCEABILITY BETWEEN 0.39 AND 0.49) OR (DANCEABILITY BETWEEN 0.89 AND

0.99) THEN 'High Danceability'

WHEN DANCEABILITY < 0.39 OR DANCEABILITY > 0.99 THEN 'Cant Dance on this one'

END AS DANCEABILITY\_ANALYSIS

FROM SPOTIFY

GROUP BY 2

ORDER BY 1 DESC;

A screenshot of a dance performance

Description automatically generated

**13. Loudness Analysis : Top 20 track\_name, loudness,**

**loudness BETWEEN -23.00 AND -15.00 ->'Low Loudness'**

**loudness BETWEEN -14.99 AND -6.00 -> 'Below Average Loudness'**

**loudness BETWEEN -5.99 AND -2.90 -> 'Above Average Loudness'**

**loudness BETWEEN -2.89 AND -1.00 -> 'Peak Loudness'**

SELECT TOP 20 TRACK\_NAME AS TOP\_20\_SONGS ,LOUDNESS,

CASE

WHEN LOUDNESS BETWEEN -23.00 AND -15.00 THEN 'Low Loudness'

WHEN LOUDNESS BETWEEN -14.99 AND -6.00 THEN 'Below Average Loudness'

WHEN LOUDNESS BETWEEN -5.99 AND -2.90 THEN 'Above Average Loudness'

WHEN LOUDNESS BETWEEN -2.89 AND -1.00 THEN 'Peak Loudness'

ELSE 'OTHERS'

END AS LOUDNESS\_ANALYSIS

FROM SPOTIFY

ORDER BY 2 DESC;

A screenshot of a music list

Description automatically generated

**14. Number of Songs for different loudness ranges(above)**

SELECT COUNT(\*) AS SONG\_DIFF\_LOUDNESS,

CASE

WHEN LOUDNESS BETWEEN -23.00 AND -15.00 THEN 'Low Loudness'

WHEN LOUDNESS BETWEEN -14.99 AND -6.00 THEN 'Below Average Loudness'

WHEN LOUDNESS BETWEEN -5.99 AND -2.90 THEN 'Above Average Loudness'

WHEN LOUDNESS BETWEEN -2.89 AND -1.00 THEN 'Peak Loudness'

ELSE 'OTHERS'

END AS LOUDNESS\_ANALYSIS

FROM SPOTIFY

GROUP BY 2

ORDER BY 1 DESC;

A screenshot of a white table

Description automatically generated

**15. Valence Analysis : Top 20 track\_name, valence, track\_popularity,**

**valence > 0.535 -> Above Avg Valence**

**valence = 0.535 -> Avg Valence**

**valence < 0.535 -> Below Average'**

SELECT TOP 20 TRACK\_NAME AS TOP\_20\_SONG,VALENCE,TRACK\_POPULARITY,

CASE

WHEN VALENCE > 0 AND VALENCE < 0.535 THEN 'Below Average'

WHEN VALENCE = 0.535 THEN 'Avg Valence'

WHEN VALENCE > 0.535 THEN 'Above Avg Valence'

END AS VALENCE\_ANALYSIS

FROM SPOTIFY

ORDER BY 3 DESC;

A screenshot of a music chart

Description automatically generated

**16. Number of Songs for different valence ranges(above)**

SELECT COUNT(\*) AS SONG\_DIFF\_VALENCE,

CASE

WHEN VALENCE > 0 AND VALENCE < 0.535 THEN 'Below Average'

WHEN VALENCE = 0.535 THEN 'Avg Valence'

WHEN VALENCE > 0.535 THEN 'Above Avg Valence'

END AS VALENCE\_ANALYSIS

FROM SPOTIFY

GROUP BY 2

ORDER BY 1 DESC;

A screenshot of a computer

Description automatically generated

**17. Speechiness Analsis : Top 20 track\_name, speechiness, tempo,**

**speechiness > 0.081-> Above Avg Speechiness**

**speechiness = 0.081-> Avg Speechiness**

**speechiness < 0.081-> Below Speechiness**

SELECT TOP 20 TRACK\_NAME AS TOP\_20\_SONG,SPEECHINESS,TEMPO,

CASE

WHEN SPEECHINESS > 0 AND SPEECHINESS < 0.081 THEN 'Below Speechiness'

WHEN SPEECHINESS = 0.081 THEN 'Avg Speechiness'

WHEN SPEECHINESS > 0.081 THEN 'Above Avg Speechiness'

END AS SPEECHINESS\_ANALYSIS

FROM SPOTIFY

ORDER BY 2 DESC;

A screenshot of a computer

Description automatically generated