# ASSIGNMENT_5 - SPOTIFY DATA ANALYSIS

## Introduction

Spotify is a digital music streaming service that provides users access to over 82 million songs, podcasts and audio books. The app was developed by Daniel Ek and Martin Lorenzton in 2006. This app has become a family name over the years and boasts over 457 million subscribers as of 2022, rivaling SoundCloud and Apple Music.

Spotify measures the popularity of its' artists based on their monthly listeners and number of streams they receive on songs produced. These streams are then multipled by (0.003) and paid to artists as "Royalties", it is a modernized system of monetizing digital sales from traditional album sales (100 streams = 1 album). Ed Sheeran was Spotify's most streamed artist in 2019, however, the rank placements change rapidly depending on album relases, EP's, mixtapes and so forth!

Spotify is a perfect dataset to measure the popularity of songs against various music elements, across a large set of songs throughout the decades. This analysis can be used to demonstrate how peoples music tastes have been translated throughout the past two decades!

I will be creating an exploratory analysis by creating data visualizations and conducting statistical analyses to investigate the relationship between the use of non-traditional musical elements and the popularity of Spotify hits from 2000 to 2019.

### Track Metadata

| Column | Description |
|---|---|
| Track_Name | Song title |
| Artist_Name | Song Artist |
| Artist_Genre | Song Genre Category |
| Year | Song Billboard chart entry year |

### Audio Numerical Quantitive Data

| Column | Description |
|---|---|
| Loudness | How loud a song is (db) |
| Duration_MS | How long the song is (seconds) |
| Tempo | How fast a song is (bpm) |

### Audio Qualitative Data

| Column | Description |
|---|---|
| Energy | How energetic the song is |
| Dance_Ability | How easy it is to dance to |
| Valence | How positive the mood of the song is |
| Acousticness | How acoustic sounding the song is |
| Speechiness | How much of a song is spoken word |
| Track_Popularity | How popular a song is (as of time of data collection) |

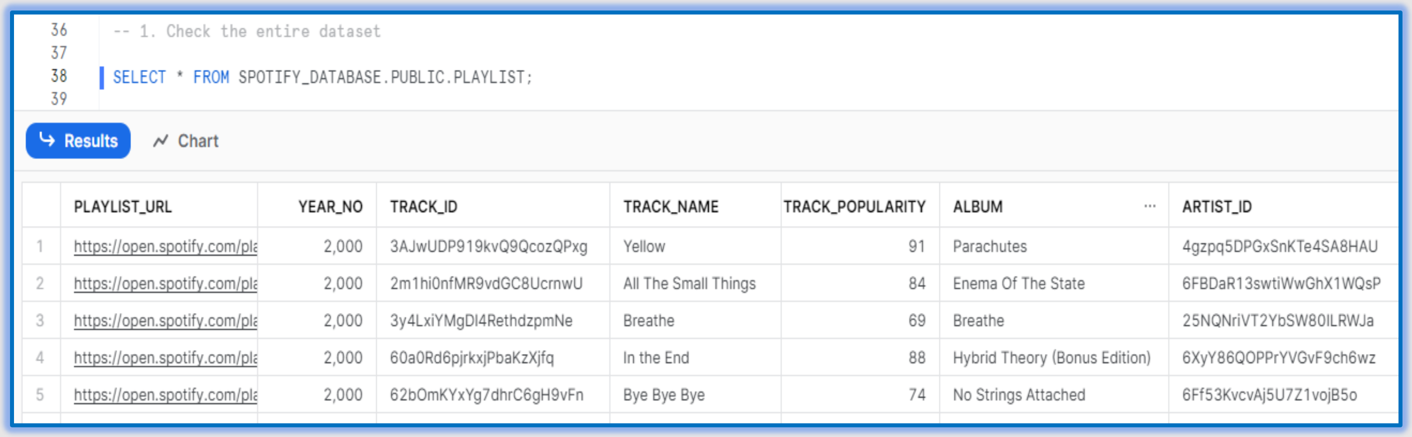
**Table Structure:**

Create a table named PLAYLIST with the following structure:

```
CREATE OR REPLACE TABLE PLAYLIST
(
 PLAYLIST_URL VARCHAR(100),
 YEAR_NO INT,
 TRACK_ID VARCHAR(50),
 TRACK_NAME VARCHAR(100),
 TRACK_POPULARITY INT,
 ALBUM VARCHAR(100),
 ARTIST_ID VARCHAR(30) ,
 ARTIST_NAME VARCHAR(50),
 ARTIST_GENRES VARCHAR(200),
 ARTIST_POPULARITY INT,
 DANCE_ABILITY DECIMAL(5,3),
 ENERGY DECIMAL(6,4),
 KEY_ID TINYINT,
 LOUDNESS DECIMAL(6,4),
 MODE_BIT TINYINT,
 SPEECHINESS DECIMAL(6,4),
 ACOUSTICNESS DECIMAL(10,8),
 INSTRUMENTALNESS DECIMAL(15,10),
 LIVENESS DECIMAL(6,4),
 VALENCE DECIMAL(6,4),
 TEMPO DECIMAL(7,4),
 DURATION_MS INT,
 TIME_SIGNATURE TINYINT,
 PRIMARY KEY (TRACK_ID, ARTIST_ID)
);
```

## Task :

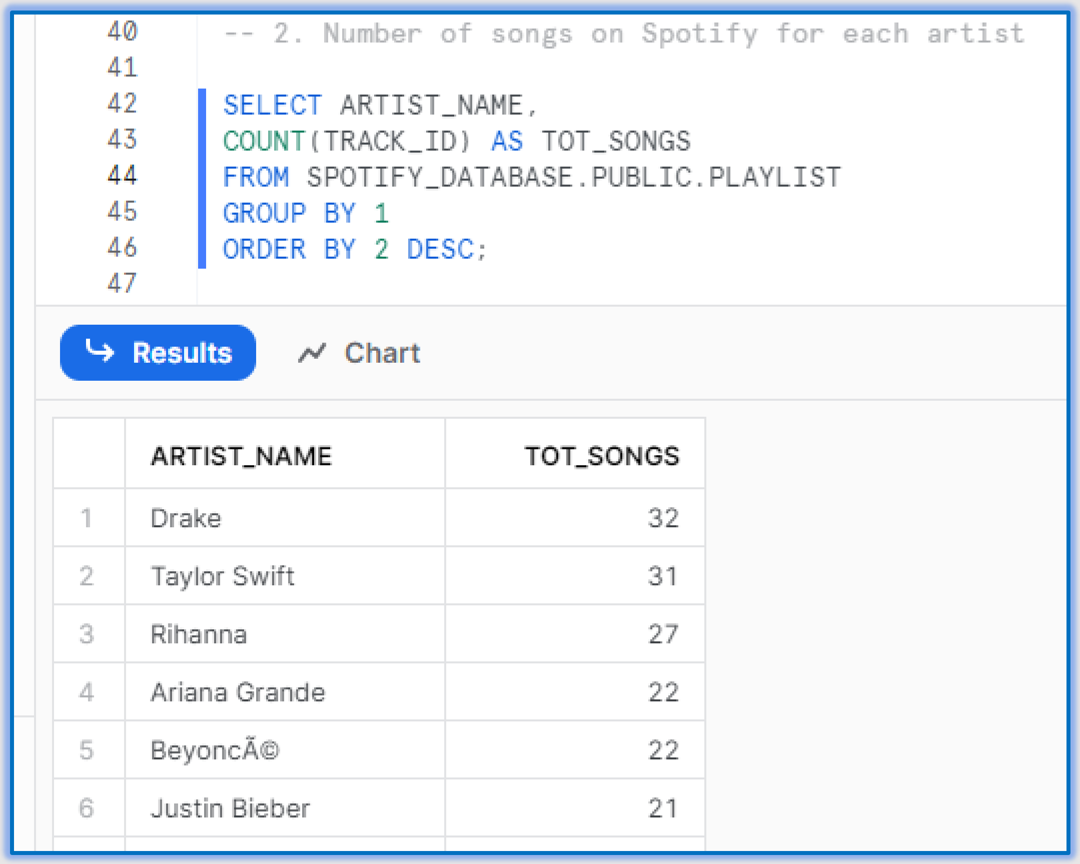
1. **Check the entire dataset**
   SELECT * FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST;



```
-- 1. Check the entire dataset

SELECT * FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST;
```

| | PLAYLIST_URL | YEAR_NO | TRACK_ID | TRACK_NAME | TRACK_POPULARITY | ALBUM | ARTIST_ID |
|---|---|---|---|---|---|---|---|
| 1 | https://open.spotify.com/pla | 2,000 | 3AJwUDP919kvQ9QcozQPxg | Yellow | 91 | Parachutes | 4gzpq5DPGxSnKTe4SA8HAU |
| 2 | https://open.spotify.com/pla | 2,000 | 2m1hi0nfMR9vdGC8UcrnwU | All The Small Things | 84 | Enema Of The State | 6FBDaR13swtiWwGhX1WQsP |
| 3 | https://open.spotify.com/pla | 2,000 | 3y4LxiYMgDl4RethdzpmNe | Breathe | 69 | Breathe | 25NQNriVT2YbSW80ILRWJa |
| 4 | https://open.spotify.com/pla | 2,000 | 60a0Rd6pjrkxjPbaKzXjfq | In the End | 88 | Hybrid Theory (Bonus Edition) | 6XyY86QOPPrYVGvF9ch6wz |
| 5 | https://open.spotify.com/pla | 2,000 | 62bOmKYxYg7dhrC6gH9vFn | Bye Bye Bye | 74 | No Strings Attached | 6Ff53KvcvAj5U7Z1vojB5o |

2. **Number of songs on Spotify for each artist**
   SELECT ARTIST_NAME,
   COUNT(TRACK_ID) AS TOT_SONGS
   FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
   GROUP BY 1
   ORDER BY 2 DESC;

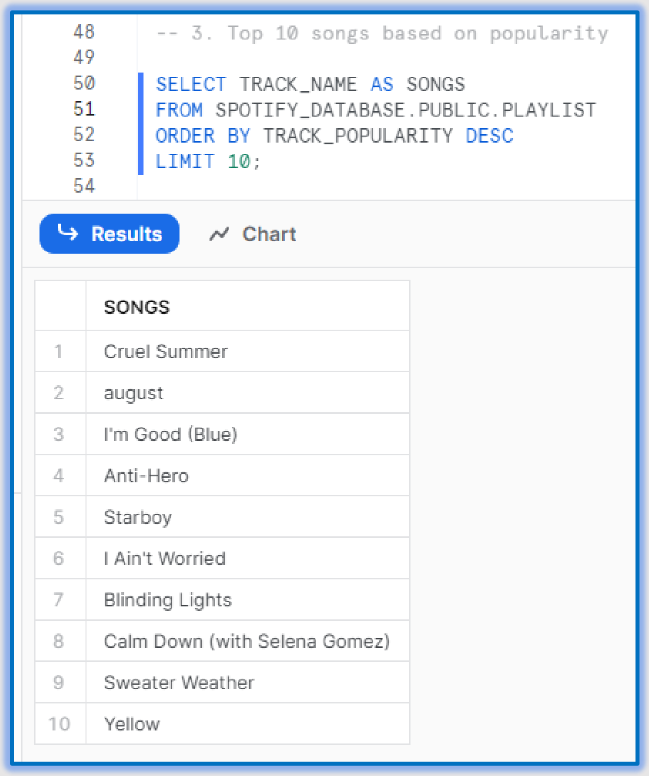


```
-- 2. Number of songs on Spotify for each artist

SELECT ARTIST_NAME,
COUNT(TRACK_ID) AS TOT_SONGS
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
GROUP BY 1
ORDER BY 2 DESC;
```

| | ARTIST_NAME | TOT_SONGS |
|---|---|---|
| 1 | Drake | 32 |
| 2 | Taylor Swift | 31 |
| 3 | Rihanna | 27 |
| 4 | Ariana Grande | 22 |
| 5 | BeyoncÃ© | 22 |
| 6 | Justin Bieber | 21 |

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

```
SELECT TRACK_NAME AS SONGS
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
ORDER BY TRACK_POPULARITY DESC
LIMIT 10;
```

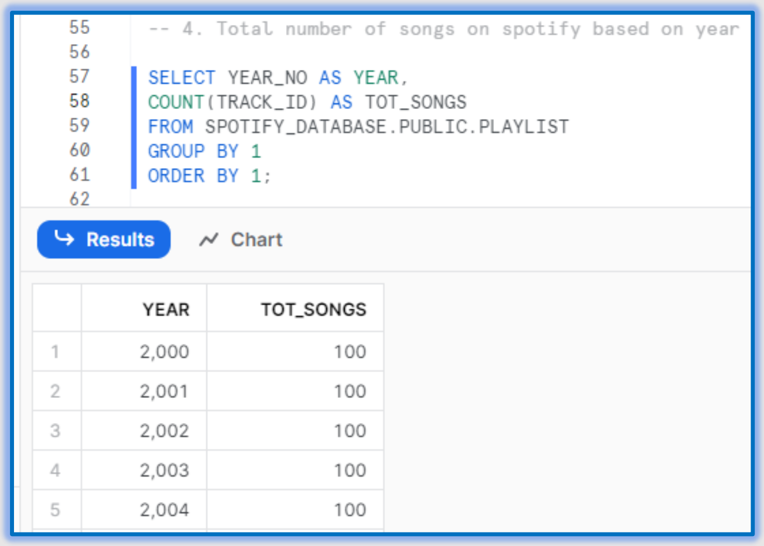


```
-- 3. Top 10 songs based on popularity

SELECT TRACK_NAME AS SONGS
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
ORDER BY TRACK_POPULARITY DESC
LIMIT 10;
```

| | SONGS |
|---|---|
| 1 | Cruel Summer |
| 2 | august |
| 3 | I'm Good (Blue) |
| 4 | Anti-Hero |
| 5 | Starboy |
| 6 | I Ain't Worried |
| 7 | Blinding Lights |
| 8 | Calm Down (with Selena Gomez) |
| 9 | Sweater Weather |
| 10 | Yellow |

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

```
SELECT YEAR_NO AS YEAR,
COUNT(TRACK_ID) AS TOT_SONGS
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
GROUP BY 1
ORDER BY 1;
```

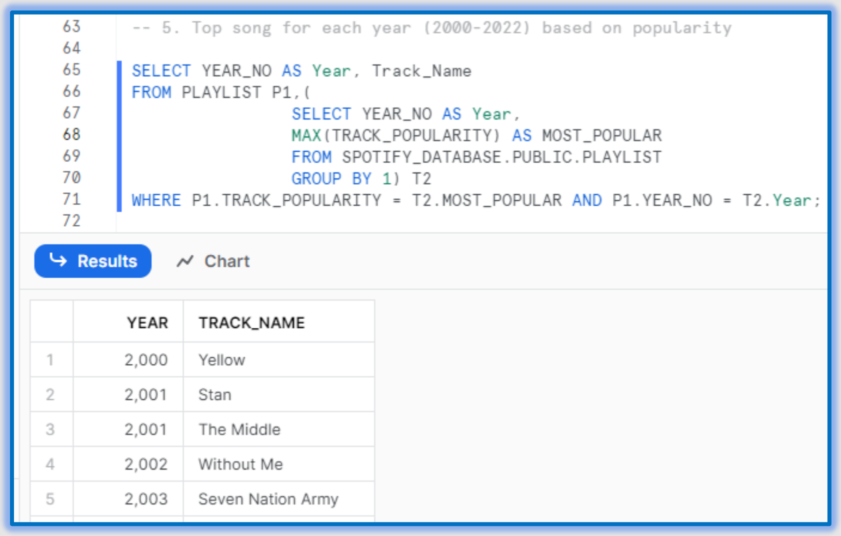


```
-- 4. Total number of songs on spotify based on year

SELECT YEAR_NO AS YEAR,
COUNT(TRACK_ID) AS TOT_SONGS
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
GROUP BY 1
ORDER BY 1;
```

| | YEAR | TOT_SONGS |
|---|---|---|
| 1 | 2,000 | 100 |
| 2 | 2,001 | 100 |
| 3 | 2,002 | 100 |
| 4 | 2,003 | 100 |
| 5 | 2,004 | 100 |

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

```
SELECT YEAR_NO AS Year, Track_Name
FROM PLAYLIST P1,(
            SELECT YEAR_NO AS Year,
            MAX(TRACK_POPULARITY) AS MOST_POPULAR
            FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
            GROUP BY 1) T2
WHERE P1.TRACK_POPULARITY = T2.MOST_POPULAR AND P1.YEAR_NO = T2.Year;
```



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

SELECT YEAR_NO AS Year, Track_Name
FROM PLAYLIST P1,(
                SELECT YEAR_NO AS Year,
                MAX(TRACK_POPULARITY) AS MOST_POPULAR
                FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
                GROUP BY 1) T2
WHERE P1.TRACK_POPULARITY = T2.MOST_POPULAR AND P1.YEAR_NO = T2.Year;
```

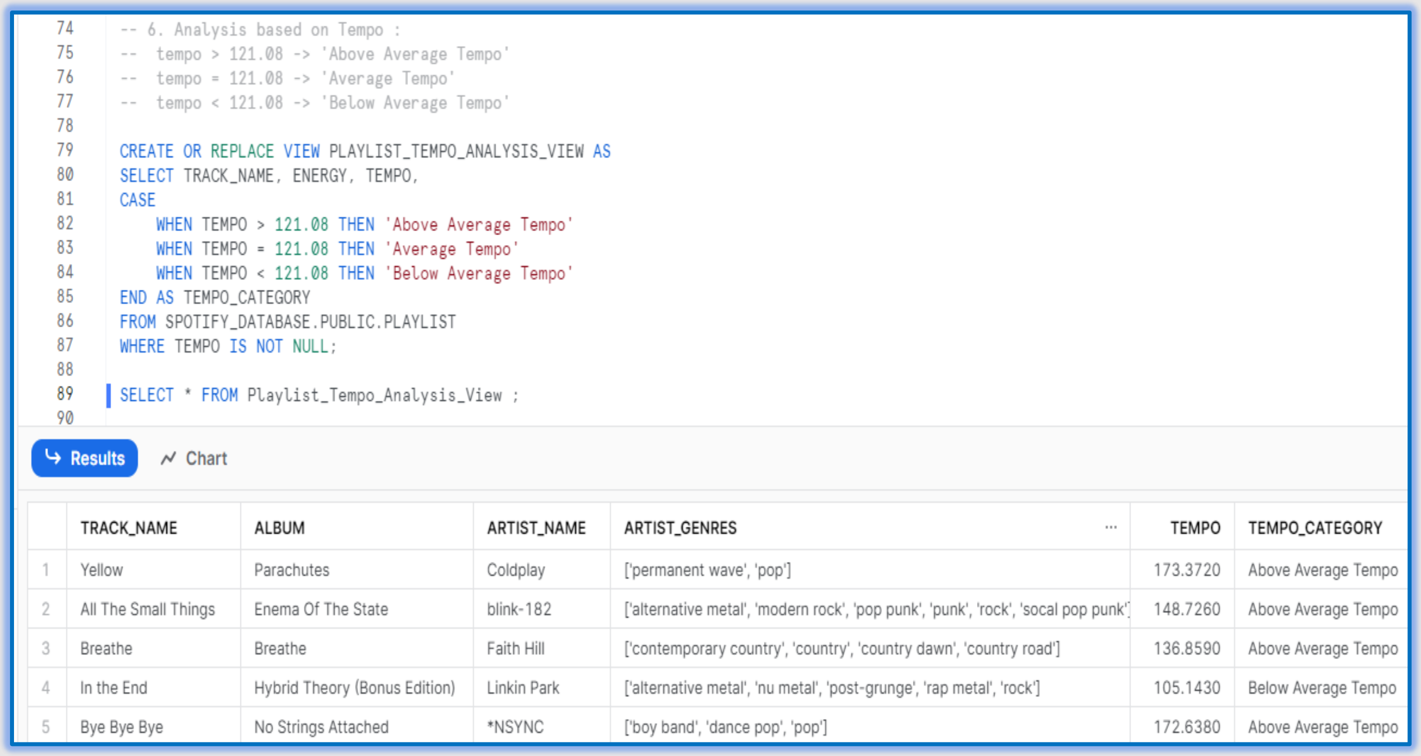
|   | YEAR | TRACK_NAME |
|---|---|---|
| 1 | 2,000 | Yellow |
| 2 | 2,001 | Stan |
| 3 | 2,001 | The Middle |
| 4 | 2,002 | Without Me |
| 5 | 2,003 | Seven Nation Army |

6. **Analysis based on Tempo :**

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

**Note:**
**I have created a View here so that I can use this view to answer other queries related to this analysis.**

```
CREATE OR REPLACE VIEW PLAYLIST_TEMPO_ANALYSIS_VIEW AS
SELECT TRACK_NAME, ENERGY, TEMPO,
CASE
   WHEN TEMPO > 121.08 THEN 'Above Average Tempo'
   WHEN TEMPO = 121.08 THEN 'Average Tempo'
   WHEN TEMPO < 121.08 THEN 'Below Average Tempo'
END AS TEMPO_CATEGORY
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE TEMPO IS NOT NULL;

SELECT * FROM Playlist_Tempo_Analysis_View ;
```

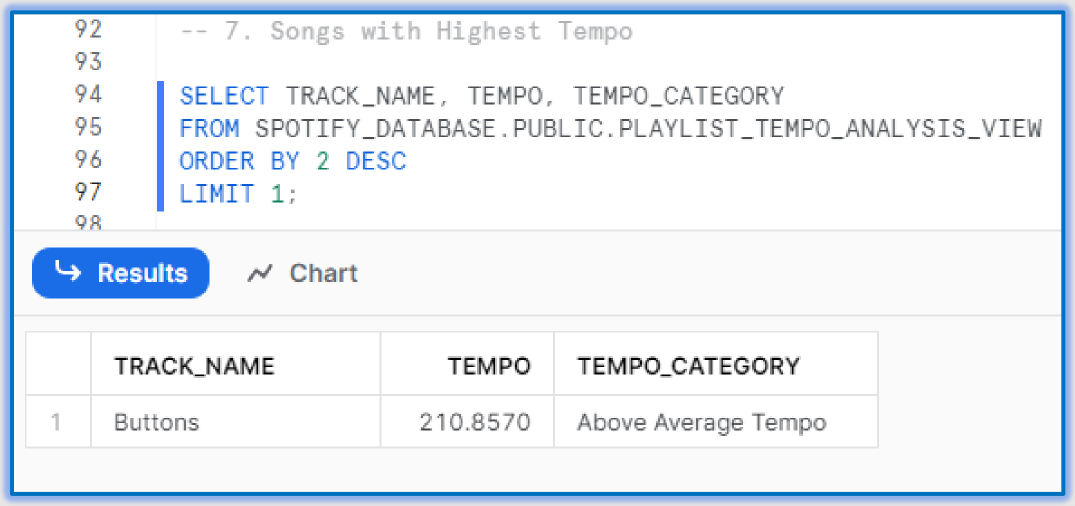


```
-- 6. Analysis based on Tempo :
--  tempo > 121.08 -> 'Above Average Tempo'
--  tempo = 121.08 -> 'Average Tempo'
--  tempo < 121.08 -> 'Below Average Tempo'

CREATE OR REPLACE VIEW PLAYLIST_TEMPO_ANALYSIS_VIEW AS
SELECT TRACK_NAME, ENERGY, TEMPO,
CASE
    WHEN TEMPO > 121.08 THEN 'Above Average Tempo'
    WHEN TEMPO = 121.08 THEN 'Average Tempo'
    WHEN TEMPO < 121.08 THEN 'Below Average Tempo'
END AS TEMPO_CATEGORY
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE TEMPO IS NOT NULL;

SELECT * FROM Playlist_Tempo_Analysis_View ;
```

| | TRACK_NAME | ALBUM | ARTIST_NAME | ARTIST_GENRES | TEMPO | TEMPO_CATEGORY |
|---|---|---|---|---|---|---|
| 1 | Yellow | Parachutes | Coldplay | ['permanent wave', 'pop'] | 173.3720 | Above Average Tempo |
| 2 | All The Small Things | Enema Of The State | blink-182 | ['alternative metal', 'modern rock', 'pop punk', 'punk', 'rock', 'socal pop punk'] | 148.7260 | Above Average Tempo |
| 3 | Breathe | Breathe | Faith Hill | ['contemporary country', 'country', 'country dawn', 'country road'] | 136.8590 | Above Average Tempo |
| 4 | In the End | Hybrid Theory (Bonus Edition) | Linkin Park | ['alternative metal', 'nu metal', 'post-grunge', 'rap metal', 'rock'] | 105.1430 | Below Average Tempo |
| 5 | Bye Bye Bye | No Strings Attached | *NSYNC | ['boy band', 'dance pop', 'pop'] | 172.6380 | Above Average Tempo |

## 7. Songs with Highest Tempo

SELECT TRACK_NAME, TEMPO, TEMPO_CATEGORY
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST_TEMPO_ANALYSIS_VIEW
ORDER BY 2 DESC
LIMIT 1;



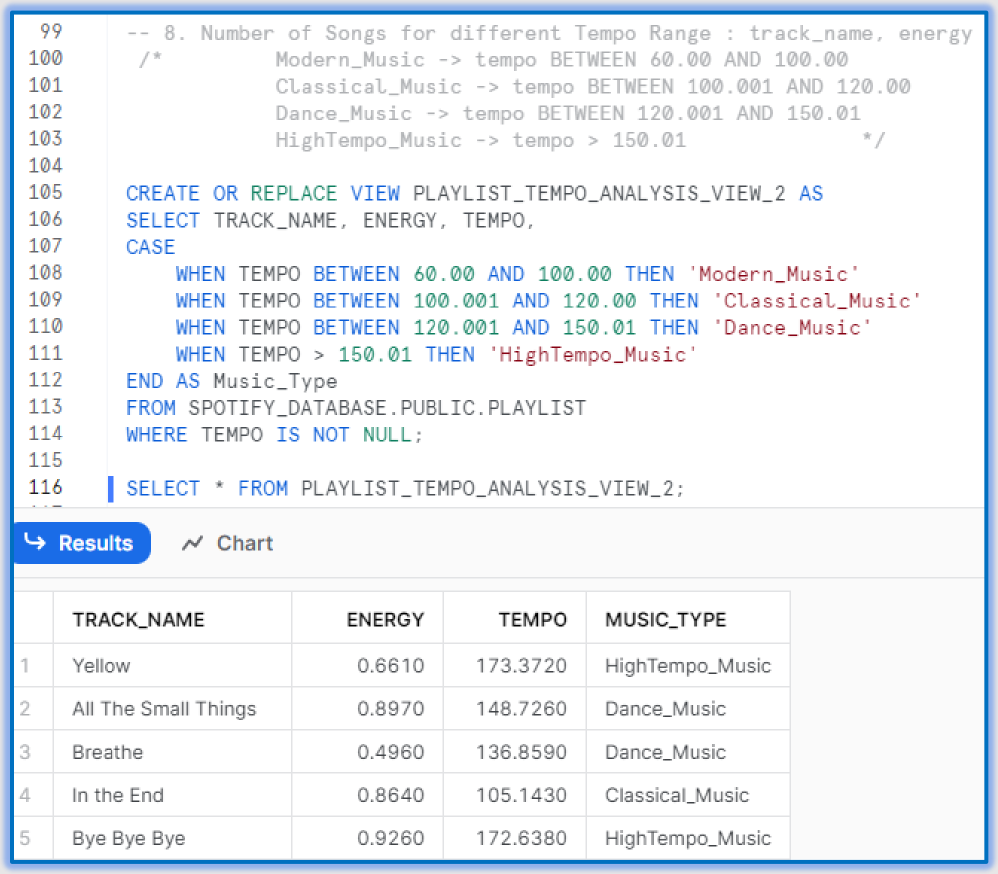
```
-- 7. Songs with Highest Tempo

SELECT TRACK_NAME, TEMPO, TEMPO_CATEGORY
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST_TEMPO_ANALYSIS_VIEW
ORDER BY 2 DESC
LIMIT 1;
```

| | TRACK_NAME | TEMPO | TEMPO_CATEGORY |
|---|---|---|---|
| 1 | Buttons | 210.8570 | Above Average Tempo |

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**

**Note:**
**I have created a View here so that I can use this view to answer other queries related to this analysis.**

```
CREATE OR REPLACE VIEW PLAYLIST_TEMPO_ANALYSIS_VIEW_2 AS
SELECT TRACK_NAME, ENERGY, 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 Music_Type
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE TEMPO IS NOT NULL;

SELECT * FROM PLAYLIST_TEMPO_ANALYSIS_VIEW;
```

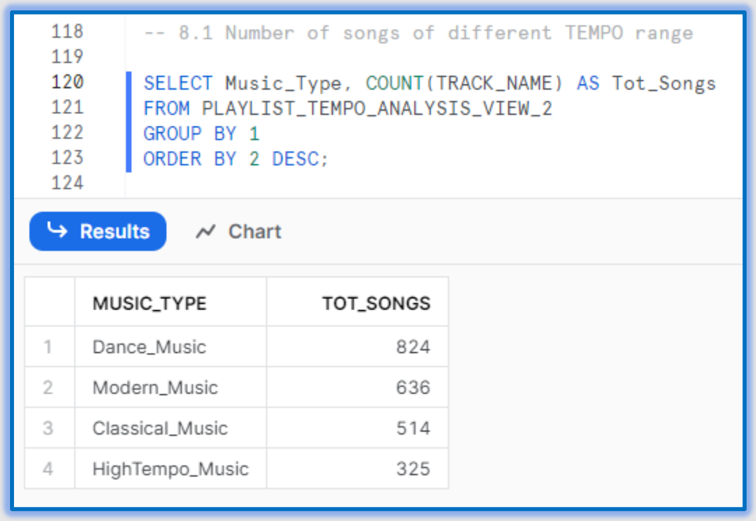


```
-- 8. Number of Songs for different Tempo Range : track_name, energy
 /*          Modern_Music -> tempo BETWEEN 60.00 AND 100.00
             Classical_Music -> tempo BETWEEN 100.001 AND 120.00
             Dance_Music -> tempo BETWEEN 120.001 AND 150.01
             HighTempo_Music -> tempo > 150.01            */

CREATE OR REPLACE VIEW PLAYLIST_TEMPO_ANALYSIS_VIEW_2 AS
SELECT TRACK_NAME, ENERGY, 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 Music_Type
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE TEMPO IS NOT NULL;

SELECT * FROM PLAYLIST_TEMPO_ANALYSIS_VIEW_2;
```

| | TRACK_NAME | ENERGY | TEMPO | MUSIC_TYPE |
|---|---|---|---|---|
| 1 | Yellow | 0.6610 | 173.3720 | HighTempo_Music |
| 2 | All The Small Things | 0.8970 | 148.7260 | Dance_Music |
| 3 | Breathe | 0.4960 | 136.8590 | Dance_Music |
| 4 | In the End | 0.8640 | 105.1430 | Classical_Music |
| 5 | Bye Bye Bye | 0.9260 | 172.6380 | HighTempo_Music |

**8.1 Number of songs of different TEMPO range**

```
SELECT Music_Type, COUNT(TRACK_NAME) AS Tot_Songs
FROM PLAYLIST_TEMPO_ANALYSIS_VIEW_2
GROUP BY 1
ORDER BY 2 DESC;
```



```
-- 8.1 Number of songs of different TEMPO range

SELECT Music_Type, COUNT(TRACK_NAME) AS Tot_Songs
FROM PLAYLIST_TEMPO_ANALYSIS_VIEW_2
GROUP BY 1
ORDER BY 2 DESC;
```

| | MUSIC_TYPE | TOT_SONGS |
|---|---|---|
| 1 | Dance_Music | 824 |
| 2 | Modern_Music | 636 |
| 3 | Classical_Music | 514 |
| 4 | HighTempo_Music | 325 |

**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'**

**Note:**

- **In this question, there is given 2 types of range on Energy so I have divided this question in 2 parts.**
- **I have created a View here so that I can use this view to answer other queries related to this analysis.**

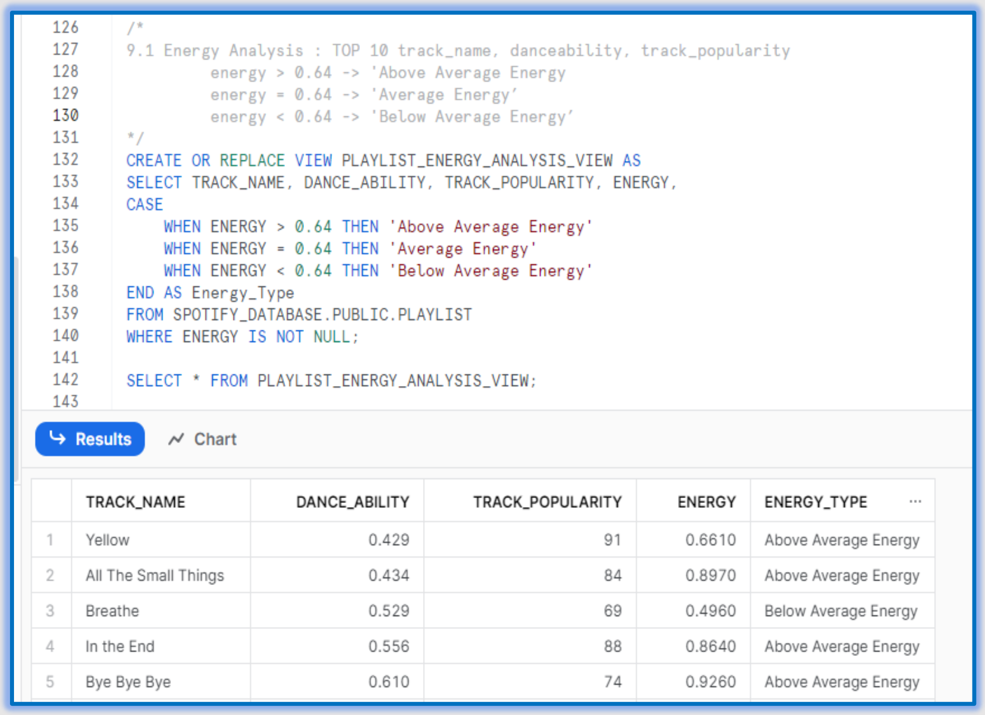
**9.1 Energy Analysis : TOP 10 track_name, danceability, track_popularity**

**energy > 0.64 -> 'Above Average Energy**
**energy = 0.64 -> 'Average Energy'**
**energy < 0.64 -> 'Below Average Energy'**

```
CREATE OR REPLACE VIEW PLAYLIST_ENERGY_ANALYSIS_VIEW AS
SELECT TRACK_NAME, DANCE_ABILITY, 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'
END AS Energy_Type
```

FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ENERGY IS NOT NULL;

SELECT * FROM PLAYLIST_ENERGY_ANALYSIS_VIEW;

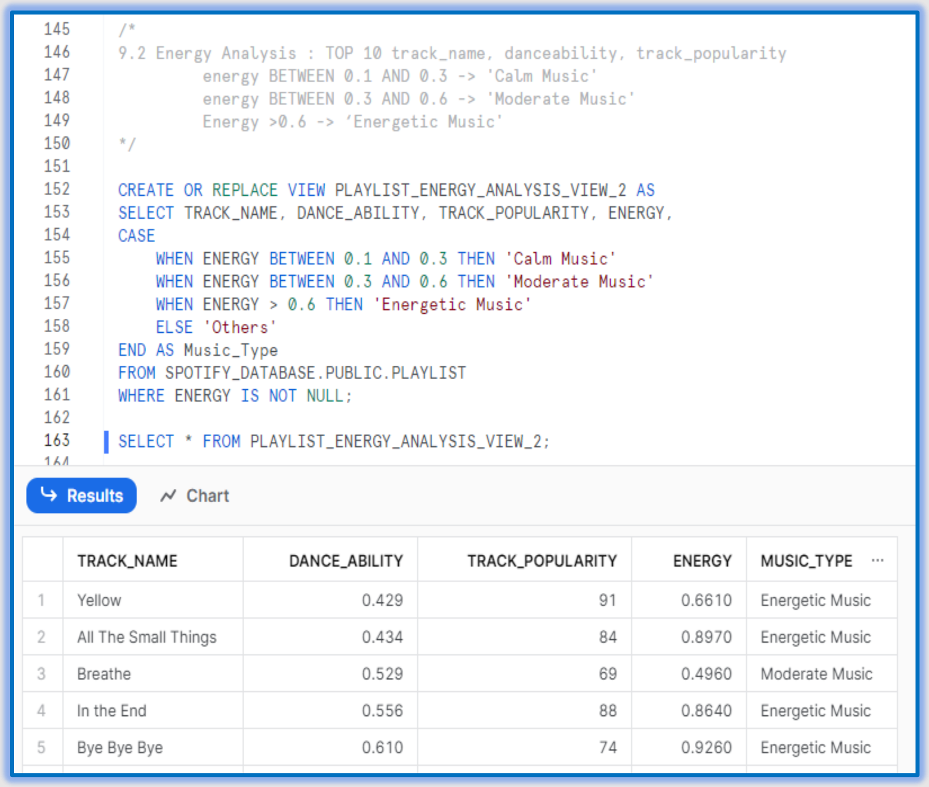


```
/*
9.1 Energy Analysis : TOP 10 track_name, danceability, track_popularity
        energy > 0.64 -> 'Above Average Energy
        energy = 0.64 -> 'Average Energy'
        energy < 0.64 -> 'Below Average Energy'
*/
CREATE OR REPLACE VIEW PLAYLIST_ENERGY_ANALYSIS_VIEW AS
SELECT TRACK_NAME, DANCE_ABILITY, 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'
END AS Energy_Type
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ENERGY IS NOT NULL;

SELECT * FROM PLAYLIST_ENERGY_ANALYSIS_VIEW;
```

| | TRACK_NAME | DANCE_ABILITY | TRACK_POPULARITY | ENERGY | ENERGY_TYPE |
|---|---|---|---|---|---|
| 1 | Yellow | 0.429 | 91 | 0.6610 | Above Average Energy |
| 2 | All The Small Things | 0.434 | 84 | 0.8970 | Above Average Energy |
| 3 | Breathe | 0.529 | 69 | 0.4960 | Below Average Energy |
| 4 | In the End | 0.556 | 88 | 0.8640 | Above Average Energy |
| 5 | Bye Bye Bye | 0.610 | 74 | 0.9260 | Above Average Energy |

**9.2 Energy Analysis : TOP 10 track_name, danceability, track_popularity**
**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 PLAYLIST_ENERGY_ANALYSIS_VIEW_2 AS
SELECT TRACK_NAME, DANCE_ABILITY, TRACK_POPULARITY, ENERGY,
CASE
   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 THEN 'Energetic Music'
   ELSE 'Others'
END AS Music_Type
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ENERGY IS NOT NULL;

SELECT * FROM PLAYLIST_ENERGY_ANALYSIS_VIEW_2;
```



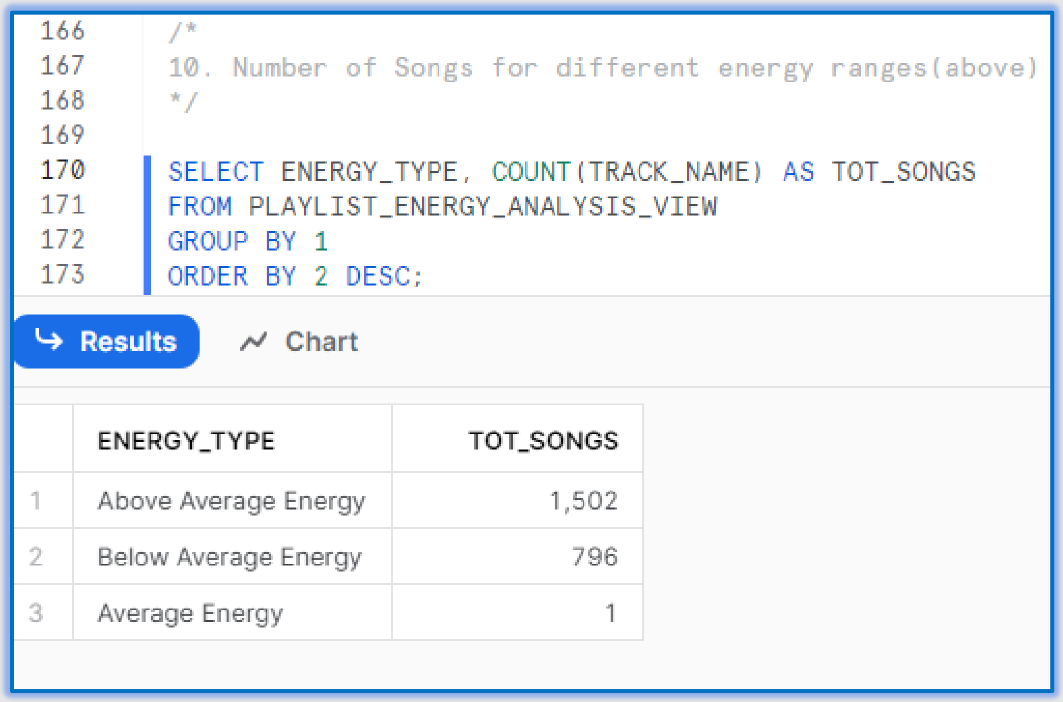
```
/*
9.2 Energy Analysis : TOP 10 track_name, danceability, track_popularity
        energy BETWEEN 0.1 AND 0.3 -> 'Calm Music'
        energy BETWEEN 0.3 AND 0.6 -> 'Moderate Music'
        Energy >0.6 -> 'Energetic Music'
*/

CREATE OR REPLACE VIEW PLAYLIST_ENERGY_ANALYSIS_VIEW_2 AS
SELECT TRACK_NAME, DANCE_ABILITY, TRACK_POPULARITY, ENERGY,
CASE
    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 THEN 'Energetic Music'
    ELSE 'Others'
END AS Music_Type
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ENERGY IS NOT NULL;

SELECT * FROM PLAYLIST_ENERGY_ANALYSIS_VIEW_2;
```

|  | TRACK_NAME | DANCE_ABILITY | TRACK_POPULARITY | ENERGY | MUSIC_TYPE |
|---|---|---|---|---|---|
| 1 | Yellow | 0.429 | 91 | 0.6610 | Energetic Music |
| 2 | All The Small Things | 0.434 | 84 | 0.8970 | Energetic Music |
| 3 | Breathe | 0.529 | 69 | 0.4960 | Moderate Music |
| 4 | In the End | 0.556 | 88 | 0.8640 | Energetic Music |
| 5 | Bye Bye Bye | 0.610 | 74 | 0.9260 | Energetic Music |

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

**FOR 9.1**

SELECT ENERGY_TYPE, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_ENERGY_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;

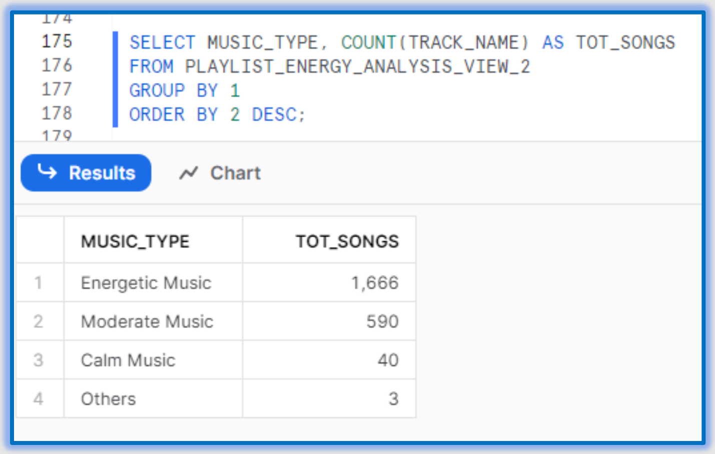


```
/*
10. Number of Songs for different energy ranges(above)
*/

SELECT ENERGY_TYPE, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_ENERGY_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```

|  | ENERGY_TYPE | TOT_SONGS |
|---|---|---|
| 1 | Above Average Energy | 1,502 |
| 2 | Below Average Energy | 796 |
| 3 | Average Energy | 1 |

**FOR 9.2**

```
SELECT MUSIC_TYPE, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_ENERGY_ANALYSIS_VIEW_2
GROUP BY 1
ORDER BY 2 DESC;
```



```
SELECT MUSIC_TYPE, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_ENERGY_ANALYSIS_VIEW_2
GROUP BY 1
ORDER BY 2 DESC;
```

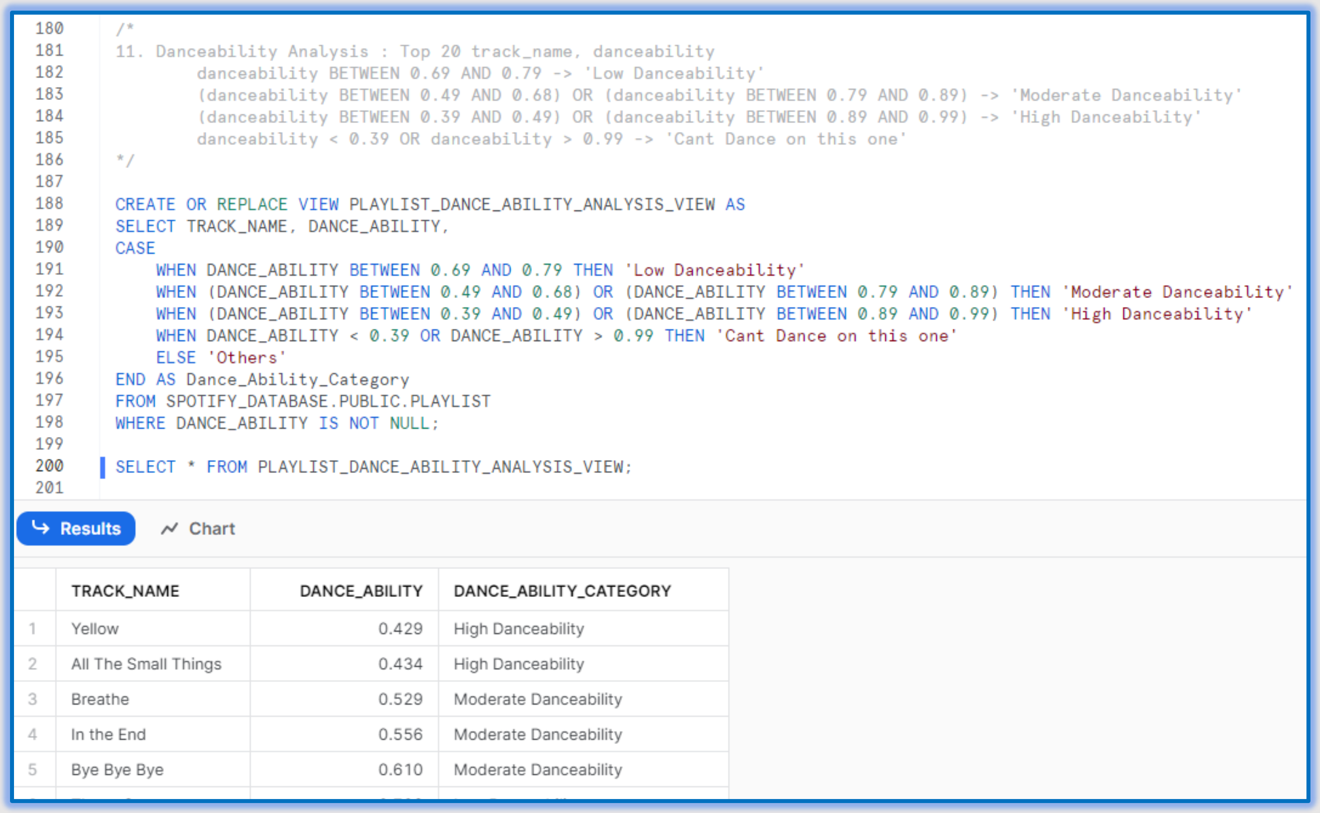
| | MUSIC_TYPE | TOT_SONGS |
|---|---|---|
| 1 | Energetic Music | 1,666 |
| 2 | Moderate Music | 590 |
| 3 | Calm Music | 40 |
| 4 | Others | 3 |

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 -> 'Cant Dance on this one'**

**Note:**
**I have created a View here so that I can use this view to answer other queries related to this analysis.**

```
CREATE OR REPLACE VIEW PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW AS
SELECT TRACK_NAME, DANCE_ABILITY,
CASE
   WHEN DANCE_ABILITY BETWEEN 0.69 AND 0.79 THEN 'Low Danceability'
   WHEN (DANCE_ABILITY BETWEEN 0.49 AND 0.68) OR (DANCE_ABILITY BETWEEN 0.79 AND 0.89) THEN 'Moderate Danceability'
   WHEN (DANCE_ABILITY BETWEEN 0.39 AND 0.49) OR (DANCE_ABILITY BETWEEN 0.89 AND 0.99) THEN 'High Danceability'
   WHEN DANCE_ABILITY < 0.39 OR DANCE_ABILITY > 0.99 THEN 'Cant Dance on this one'
   ELSE 'Others'
END AS Dance_Ability_Category
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE DANCE_ABILITY IS NOT NULL;
```

SELECT * FROM PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW;



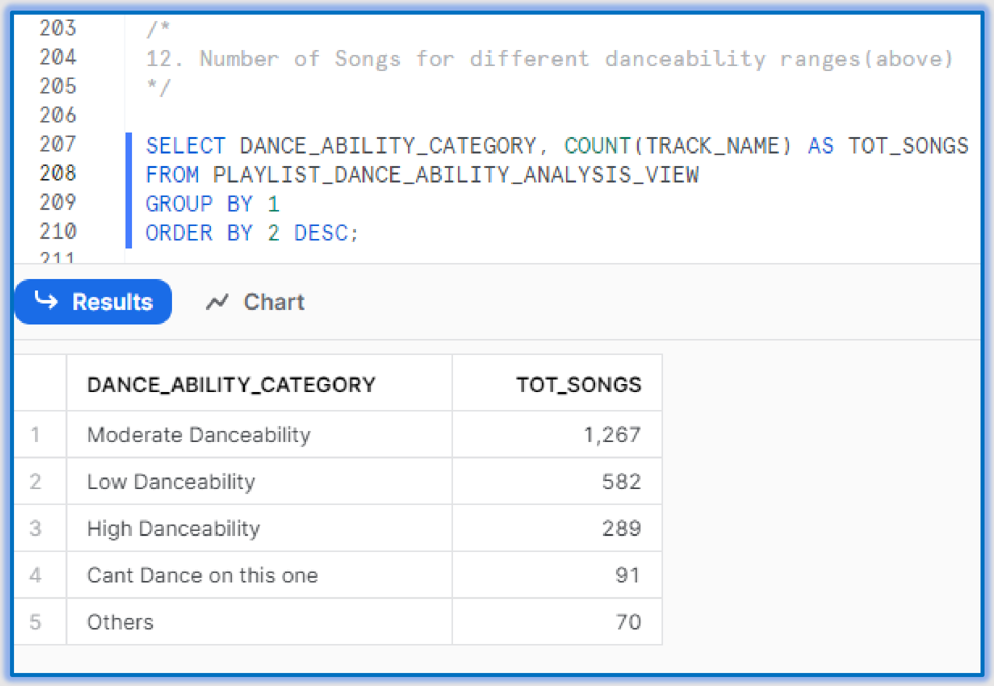
```
/*
11. Danceability Analysis : Top 20 track_name, danceability
        danceability BETWEEN 0.69 AND 0.79 -> 'Low Danceability'
        (danceability BETWEEN 0.49 AND 0.68) OR (danceability BETWEEN 0.79 AND 0.89) -> 'Moderate Danceability'
        (danceability BETWEEN 0.39 AND 0.49) OR (danceability BETWEEN 0.89 AND 0.99) -> 'High Danceability'
        danceability < 0.39 OR danceability > 0.99 -> 'Cant Dance on this one'
*/

CREATE OR REPLACE VIEW PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW AS
SELECT TRACK_NAME, DANCE_ABILITY,
CASE
    WHEN DANCE_ABILITY BETWEEN 0.69 AND 0.79 THEN 'Low Danceability'
    WHEN (DANCE_ABILITY BETWEEN 0.49 AND 0.68) OR (DANCE_ABILITY BETWEEN 0.79 AND 0.89) THEN 'Moderate Danceability'
    WHEN (DANCE_ABILITY BETWEEN 0.39 AND 0.49) OR (DANCE_ABILITY BETWEEN 0.89 AND 0.99) THEN 'High Danceability'
    WHEN DANCE_ABILITY < 0.39 OR DANCE_ABILITY > 0.99 THEN 'Cant Dance on this one'
    ELSE 'Others'
END AS Dance_Ability_Category
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE DANCE_ABILITY IS NOT NULL;

SELECT * FROM PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW;
```

Results Chart

| | TRACK_NAME | DANCE_ABILITY | DANCE_ABILITY_CATEGORY |
|---|---|---|---|
| 1 | Yellow | 0.429 | High Danceability |
| 2 | All The Small Things | 0.434 | High Danceability |
| 3 | Breathe | 0.529 | Moderate Danceability |
| 4 | In the End | 0.556 | Moderate Danceability |
| 5 | Bye Bye Bye | 0.610 | Moderate Danceability |

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

SELECT DANCE_ABILITY_CATEGORY, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;



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

SELECT DANCE_ABILITY_CATEGORY, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_DANCE_ABILITY_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```

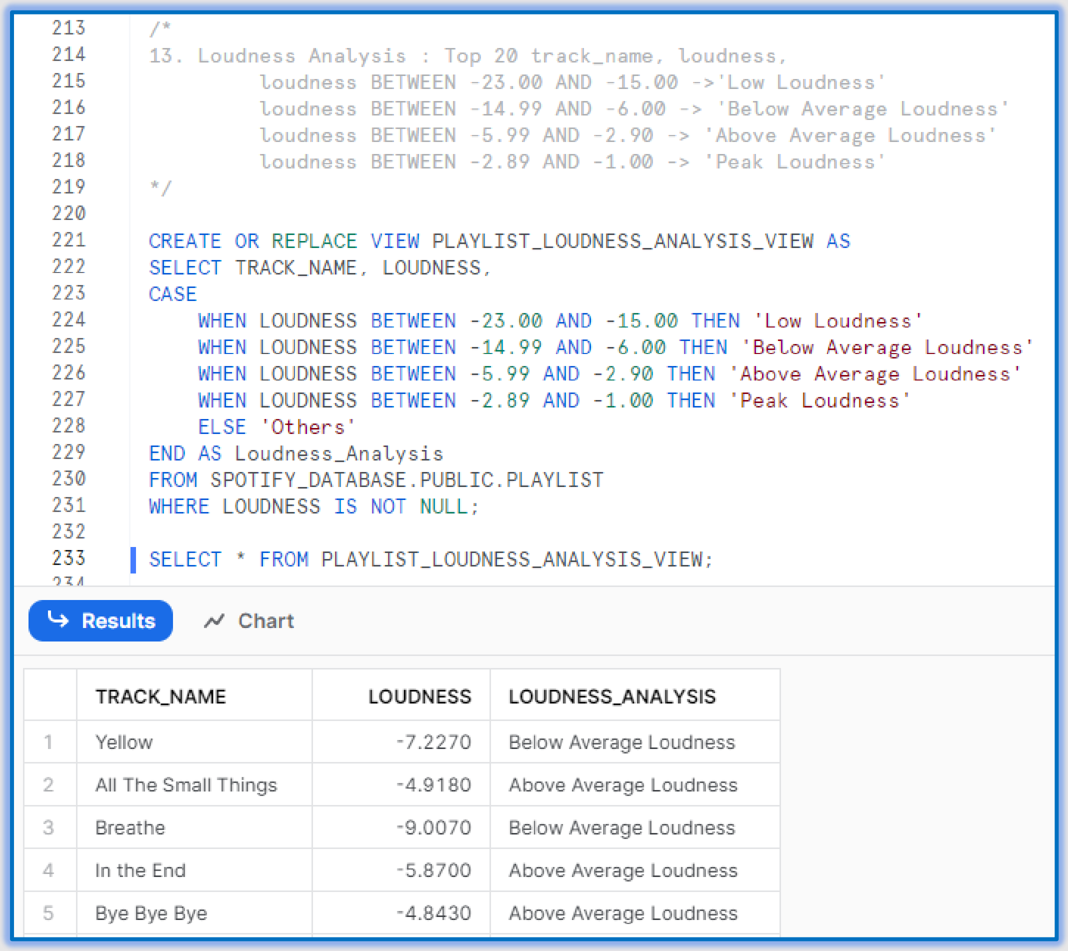
Results Chart

| | DANCE_ABILITY_CATEGORY | TOT_SONGS |
|---|---|---|
| 1 | Moderate Danceability | 1,267 |
| 2 | Low Danceability | 582 |
| 3 | High Danceability | 289 |
| 4 | Cant Dance on this one | 91 |
| 5 | Others | 70 |

**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'**

**Note:**
**I have created a View here so that I can use this view to answer other queries related to this analysis.**

```
CREATE OR REPLACE VIEW PLAYLIST_LOUDNESS_ANALYSIS_VIEW AS
SELECT TRACK_NAME, 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_DATABASE.PUBLIC.PLAYLIST
WHERE LOUDNESS IS NOT NULL;

SELECT * FROM PLAYLIST_LOUDNESS_ANALYSIS_VIEW;
```

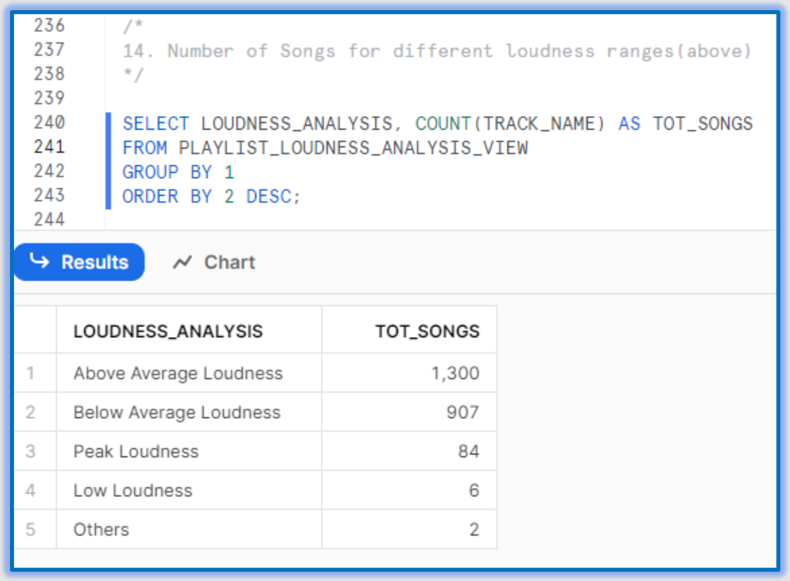


```
/*
13. Loudness Analysis : Top 20 track_name, loudness,
         loudness BETWEEN -23.00 AND -15.00 ->'Low Loudness'
         loudness BETWEEN -14.99 AND -6.00 -> 'Below Average Loudness'
         loudness BETWEEN -5.99 AND -2.90 -> 'Above Average Loudness'
         loudness BETWEEN -2.89 AND -1.00 -> 'Peak Loudness'
*/

CREATE OR REPLACE VIEW PLAYLIST_LOUDNESS_ANALYSIS_VIEW AS
SELECT TRACK_NAME, 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_DATABASE.PUBLIC.PLAYLIST
WHERE LOUDNESS IS NOT NULL;

SELECT * FROM PLAYLIST_LOUDNESS_ANALYSIS_VIEW;
```

| | TRACK_NAME | LOUDNESS | LOUDNESS_ANALYSIS |
|---|---|---|---|
| 1 | Yellow | -7.2270 | Below Average Loudness |
| 2 | All The Small Things | -4.9180 | Above Average Loudness |
| 3 | Breathe | -9.0070 | Below Average Loudness |
| 4 | In the End | -5.8700 | Above Average Loudness |
| 5 | Bye Bye Bye | -4.8430 | Above Average Loudness |

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

```
SELECT LOUDNESS_ANALYSIS, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_LOUDNESS_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```



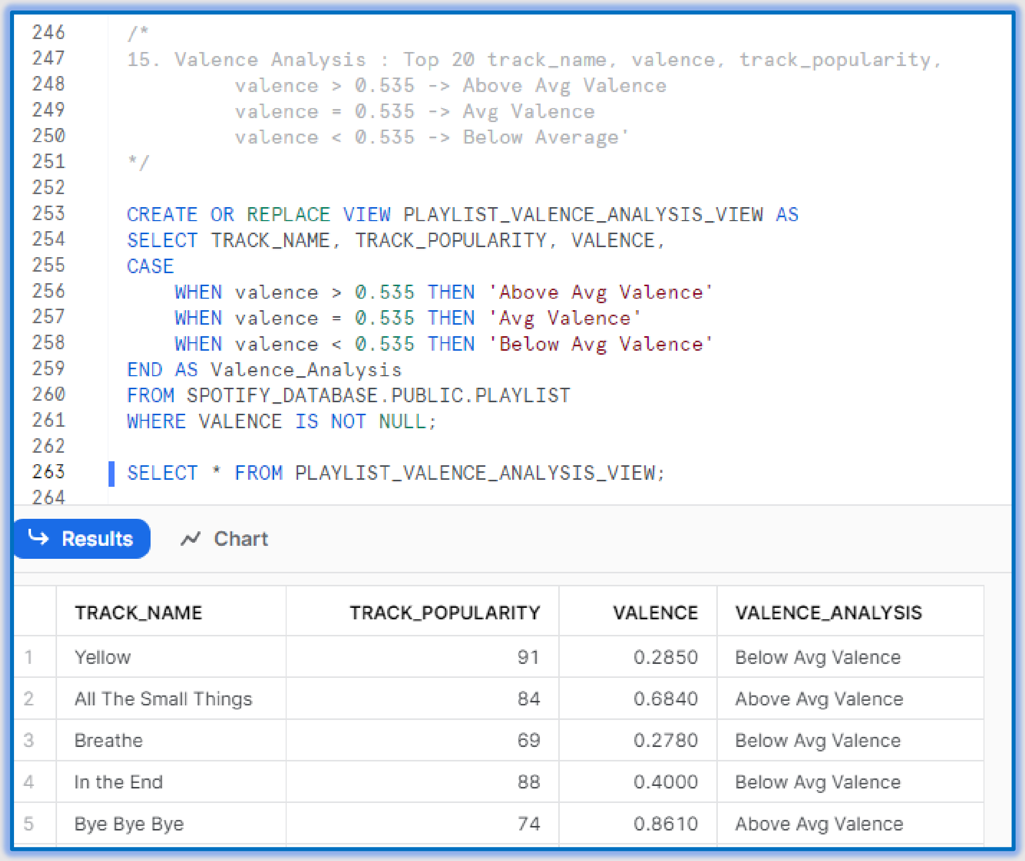
```
/*
14. Number of Songs for different loudness ranges(above)
*/

SELECT LOUDNESS_ANALYSIS, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_LOUDNESS_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```

|  | LOUDNESS_ANALYSIS | TOT_SONGS |
|---|---|---|
| 1 | Above Average Loudness | 1,300 |
| 2 | Below Average Loudness | 907 |
| 3 | Peak Loudness | 84 |
| 4 | Low Loudness | 6 |
| 5 | Others | 2 |

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'**

**Note:**
**I have created a View here so that I can use this view to answer other queries related to this analysis.**

```
CREATE OR REPLACE VIEW PLAYLIST_VALENCE_ANALYSIS_VIEW AS
SELECT TRACK_NAME, TRACK_POPULARITY, VALENCE,
CASE
   WHEN valence > 0.535 THEN 'Above Avg Valence'
   WHEN valence = 0.535 THEN 'Avg Valence'
   WHEN valence < 0.535 THEN 'Below Avg Valence'
END AS Valence_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE VALENCE IS NOT NULL;

SELECT * FROM PLAYLIST_VALENCE_ANALYSIS_VIEW;
```



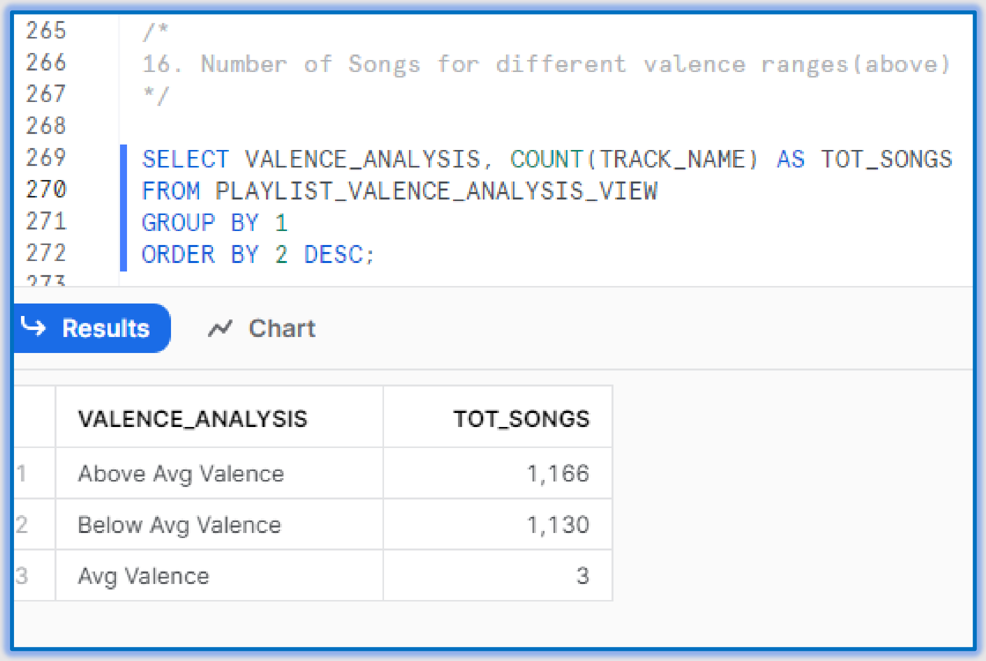
```
/*
15. Valence Analysis : Top 20 track_name, valence, track_popularity,
          valence > 0.535 -> Above Avg Valence
          valence = 0.535 -> Avg Valence
          valence < 0.535 -> Below Average'
*/

CREATE OR REPLACE VIEW PLAYLIST_VALENCE_ANALYSIS_VIEW AS
SELECT TRACK_NAME, TRACK_POPULARITY, VALENCE,
CASE
    WHEN valence > 0.535 THEN 'Above Avg Valence'
    WHEN valence = 0.535 THEN 'Avg Valence'
    WHEN valence < 0.535 THEN 'Below Avg Valence'
END AS Valence_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE VALENCE IS NOT NULL;

SELECT * FROM PLAYLIST_VALENCE_ANALYSIS_VIEW;
```

Results   Chart

| | TRACK_NAME | TRACK_POPULARITY | VALENCE | VALENCE_ANALYSIS |
|---|---|---|---|---|
| 1 | Yellow | 91 | 0.2850 | Below Avg Valence |
| 2 | All The Small Things | 84 | 0.6840 | Above Avg Valence |
| 3 | Breathe | 69 | 0.2780 | Below Avg Valence |
| 4 | In the End | 88 | 0.4000 | Below Avg Valence |
| 5 | Bye Bye Bye | 74 | 0.8610 | Above Avg Valence |

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

```
SELECT VALENCE_ANALYSIS, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_VALENCE_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```



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

SELECT VALENCE_ANALYSIS, COUNT(TRACK_NAME) AS TOT_SONGS
FROM PLAYLIST_VALENCE_ANALYSIS_VIEW
GROUP BY 1
ORDER BY 2 DESC;
```

Results   Chart

| | VALENCE_ANALYSIS | TOT_SONGS |
|---|---|---|
| 1 | Above Avg Valence | 1,166 |
| 2 | Below Avg Valence | 1,130 |
| 3 | Avg Valence | 3 |

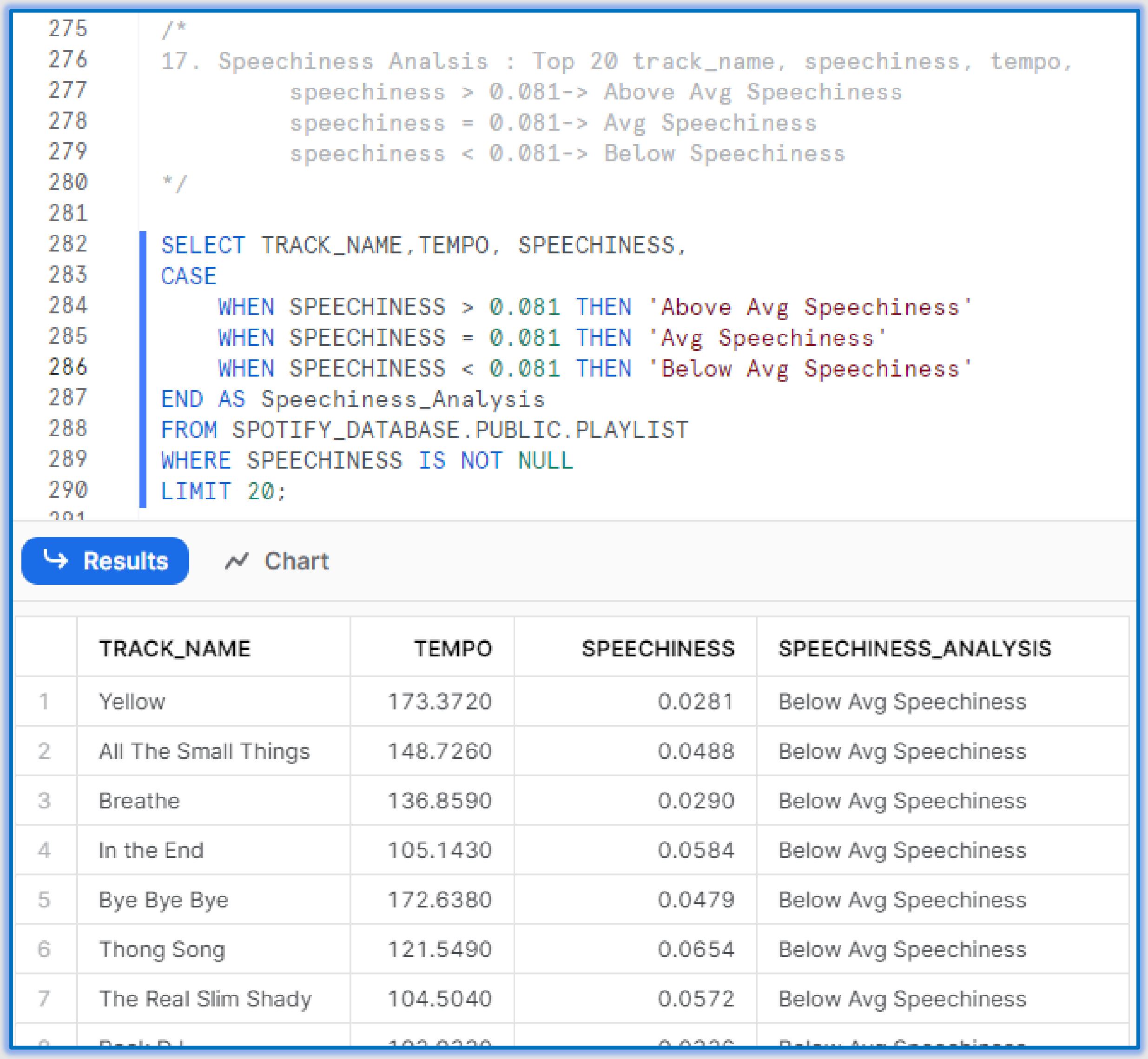
**17. Speechiness Analsis : Top 20 track_name, speechiness, tempo,**

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

**speechiness = 0.081-> Avg Speechiness**

**speechiness < 0.081-> Below Speechiness**

```
SELECT TRACK_NAME,TEMPO, SPEECHINESS,
CASE
   WHEN SPEECHINESS > 0.081 THEN 'Above Avg Speechiness'
   WHEN SPEECHINESS = 0.081 THEN 'Avg Speechiness'
   WHEN SPEECHINESS < 0.081 THEN 'Below Avg Speechiness'
END AS Speechiness_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE SPEECHINESS IS NOT NULL
LIMIT 20;
```

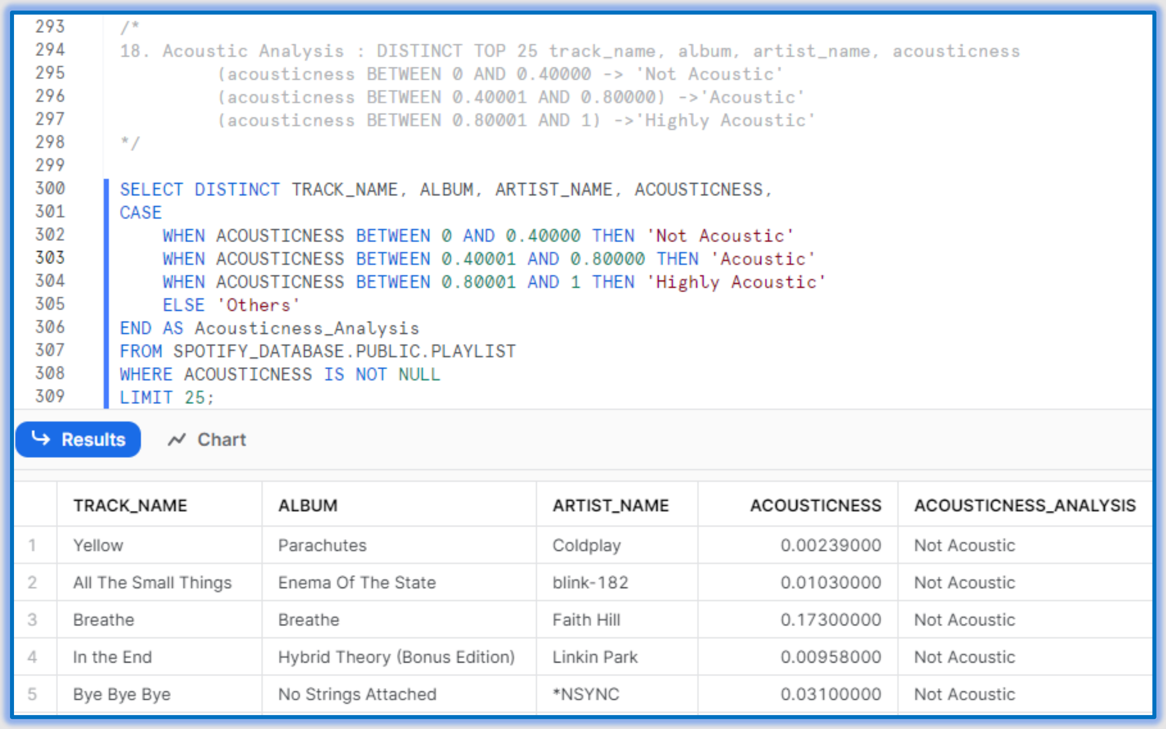


```
/*
17. Speechiness Analsis : Top 20 track_name, speechiness, tempo,
        speechiness > 0.081-> Above Avg Speechiness
        speechiness = 0.081-> Avg Speechiness
        speechiness < 0.081-> Below Speechiness
*/

SELECT TRACK_NAME,TEMPO, SPEECHINESS,
CASE
    WHEN SPEECHINESS > 0.081 THEN 'Above Avg Speechiness'
    WHEN SPEECHINESS = 0.081 THEN 'Avg Speechiness'
    WHEN SPEECHINESS < 0.081 THEN 'Below Avg Speechiness'
END AS Speechiness_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE SPEECHINESS IS NOT NULL
LIMIT 20;
```

| | TRACK_NAME | TEMPO | SPEECHINESS | SPEECHINESS_ANALYSIS |
|---|---|---|---|---|
| 1 | Yellow | 173.3720 | 0.0281 | Below Avg Speechiness |
| 2 | All The Small Things | 148.7260 | 0.0488 | Below Avg Speechiness |
| 3 | Breathe | 136.8590 | 0.0290 | Below Avg Speechiness |
| 4 | In the End | 105.1430 | 0.0584 | Below Avg Speechiness |
| 5 | Bye Bye Bye | 172.6380 | 0.0479 | Below Avg Speechiness |
| 6 | Thong Song | 121.5490 | 0.0654 | Below Avg Speechiness |
| 7 | The Real Slim Shady | 104.5040 | 0.0572 | Below Avg Speechiness |

**18. Acoustic Analysis : DISTINCT TOP 25 track_name, album, artist_name, acousticness**
**(acousticness BETWEEN 0 AND 0.40000 -> 'Not Acoustic'**
**(acousticness BETWEEN 0.40001 AND 0.80000) ->'Acoustic'**
**(acousticness BETWEEN 0.80001 AND 1) ->'Highly Acoustic'**

```
SELECT DISTINCT TRACK_NAME, ALBUM, ARTIST_NAME, ACOUSTICNESS,
CASE
   WHEN ACOUSTICNESS BETWEEN 0 AND 0.40000 THEN 'Not Acoustic'
   WHEN ACOUSTICNESS BETWEEN 0.40001 AND 0.80000 THEN 'Acoustic'
   WHEN ACOUSTICNESS BETWEEN 0.80001 AND 1 THEN 'Highly Acoustic'
   ELSE 'Others'
END AS Acousticness_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ACOUSTICNESS IS NOT NULL
LIMIT 25;
```



```
/*
18. Acoustic Analysis : DISTINCT TOP 25 track_name, album, artist_name, acousticness
        (acousticness BETWEEN 0 AND 0.40000 -> 'Not Acoustic'
        (acousticness BETWEEN 0.40001 AND 0.80000) ->'Acoustic'
        (acousticness BETWEEN 0.80001 AND 1) ->'Highly Acoustic'
*/

SELECT DISTINCT TRACK_NAME, ALBUM, ARTIST_NAME, ACOUSTICNESS,
CASE
    WHEN ACOUSTICNESS BETWEEN 0 AND 0.40000 THEN 'Not Acoustic'
    WHEN ACOUSTICNESS BETWEEN 0.40001 AND 0.80000 THEN 'Acoustic'
    WHEN ACOUSTICNESS BETWEEN 0.80001 AND 1 THEN 'Highly Acoustic'
    ELSE 'Others'
END AS Acousticness_Analysis
FROM SPOTIFY_DATABASE.PUBLIC.PLAYLIST
WHERE ACOUSTICNESS IS NOT NULL
LIMIT 25;
```

| | TRACK_NAME | ALBUM | ARTIST_NAME | ACOUSTICNESS | ACOUSTICNESS_ANALYSIS |
|---|---|---|---|---|---|
| 1 | Yellow | Parachutes | Coldplay | 0.00239000 | Not Acoustic |
| 2 | All The Small Things | Enema Of The State | blink-182 | 0.01030000 | Not Acoustic |
| 3 | Breathe | Breathe | Faith Hill | 0.17300000 | Not Acoustic |
| 4 | In the End | Hybrid Theory (Bonus Edition) | Linkin Park | 0.00958000 | Not Acoustic |
| 5 | Bye Bye Bye | No Strings Attached | *NSYNC | 0.03100000 | Not Acoustic |

**************** THANK YOU ****************