```
--Niyo final Projects - Spotify EDA

--Retrieving from the first DataSet, top 10,000
SELECT * FROM `niyo-sql-project.Spotify.Analysis`
LIMIT 10;
-- Also shows top 10 streamed songs

-- Retrieving from second dataset, 2022
SELECT * FROM `niyo-sql-project.Spotify.Short`;

--Merging the two Datasets, looking only at rows with position in top 100
SELECT Position, Artist_Name, Song_Name, Peak_Position, weeks_on_chart,
Total_Streams, Top_10__xTimes_, danceability, energy, key, loudness, mode,
speechiness, acousticness, instrumentalness, liveness, tempo, duration_ms
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-
project.Spotify.Analysis`.Song_Name
WHERE Position < 100
ORDER BY Position ASC;
-- Shows 27 results

-- Looking for Twice Song(s) in Merged Data set
SELECT Position, Artist_Name, Song_Name, Peak_Position, weeks_on_chart,
Total_Streams, Top_10__xTimes_, danceability, energy, key, loudness, mode,
speechiness, acousticness, instrumentalness, liveness, tempo, duration_ms
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-
project.Spotify.Analysis`.Song_Name
WHERE Artist_Name = 'TWICE ';

-- Retrieving from second dataset, 2022
SELECT * FROM `niyo-sql-project.Spotify.Short`;

--Merging the two Datasets, looking only at rows with position in top 100
SELECT Position, Artist_Name, Song_Name, Peak_Position, weeks_on_chart,
Total_Streams, Top_10__xTimes_, danceability, energy, key, loudness, mode,
speechiness, acousticness, instrumentalness, liveness, tempo, duration_ms
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-
project.Spotify.Analysis`.Song_Name
WHERE Position < 11
ORDER BY Position ASC;

---- Effect of Days on position
SELECT Position, Artist_Name, Song_Name, Days
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
```

```sql
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-project.Spotify.Analysis`.Song_Name
ORDER BY Days DESC
Limit 10;
-- Can't see much of a correlation between duration and position. Would need visual

-- Does the number of days the song has been released have any impact on number of streams
SELECT Position, Artist_Name, Song_Name, Days, Total_Streams
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-project.Spotify.Analysis`.Song_Name
WHERE Position = 2 OR Position = 53
ORDER BY Days Desc;
-- Can see that although the 2nd position has around double the streams, it has been released for around double the amount of time, which may be a contributing factor to its high streams and therefore position

-- Categorise Total_Streams to see distribution (make 9 Buckets)
SELECT COUNT(Total_Streams),
CASE
WHEN Total_Streams < 100000000 THEN 'Less than 100'
WHEN Total_Streams > 800000000 THEN 'More than 800'
WHEN Total_Streams > 700000000 THEN 'Between 700 and 800'
WHEN Total_Streams > 600000000 THEN 'Between 600 and 700'
WHEN Total_Streams > 500000000 THEN 'Between 500 and 600'
WHEN Total_Streams > 400000000 THEN 'Between 400 and 500'
WHEN Total_Streams > 300000000 THEN 'Between 300 and 400'
WHEN Total_Streams > 200000000 THEN 'Between 200 and 100'
WHEN Total_Streams > 100000000 THEN 'Between 100 and 200'
END AS TotalStreamsInMillions
FROM `niyo-sql-project.Spotify.Short`
INNER JOIN `niyo-sql-project.Spotify.Analysis`
ON `niyo-sql-project.Spotify.Short`.track_name = `niyo-sql-project.Spotify.Analysis`.Song_Name
GROUP BY TotalStreamsInMillions
ORDER BY COUNT(Total_Streams) DESC;
-- Majority of the songs have less than 100 million streams
-- Hints that dataset may need to be split for clearer visuals.
-- Is not necessary as will majorly looking at top 10 songs (<'between 600 and 700 Million')
```