Assignment 1 - Cleaning and Querying with Pig/Hive

GitHub

All the code for the cleaning, queries and visualisations can be found in GitHub repository here

1 Data Loading, Cleaning and Saving

For my dataset, I have chosen to use the spotify tracks dataset which can be found here: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset. This dataset consists of information about 114,000 tracks and various features such as the track_name, artists, album_name, popularity, duration etc…

Loading the dataset

Initially I had tried to load the data using the built-in function “*PigStorage(‘,’)”* however, I had noticed that certain rows were being mis-aligned due to the comma delimiter being present in some album or track names. One example would be the record found at index 21 where the album name was “*Cover Sessions, Vol. 3*". To work around this, I used the “org.apache.pig.piggybank.storage.CSVLoader()” which correctly aligned the rows as the values with commas were enclosed in double quotes.

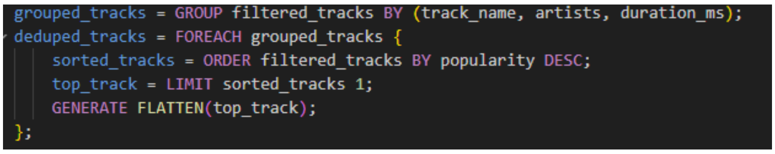
Cleaning the dataset

Header Removal

The first step in cleaning the dataset was to remove the header. Pig would treat this as just any-other row which would cause issues as the data in this row isn’t representative of actual tracks and their features. This was a simple fix as all I had to do was filterrows where the “*track_id != ‘track_id*”. This just removed the first row as track_ids are always integer values.

Deduplication of songs

An aspect of the dataset I noticed was that the same song can appear multiple times in different rows as it could be part of different albums at the same time (such as re-releases, top 50 albums, Christmas albums etc…). I didn’t want to include duplicates in my analysis as it could skew the results if a song has been re-released multiple times. To do this I grouped tracks by having the same track name, artists and duration. If all of these values were the same, I could be confident that these were in fact the exact same song and then I proceeded to sort these songs by popularity and took the song with the largest value. This ensured that the most popular version of the song was included in my analysis. Over 30,000 duplicate tracks were removed.

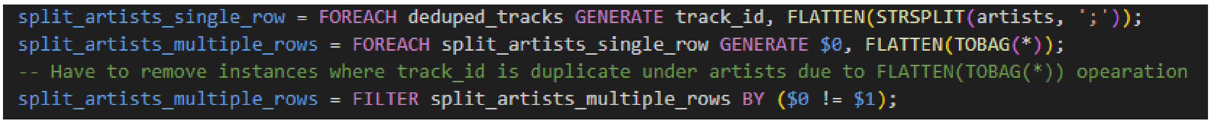


```
grouped_tracks = GROUP filtered_tracks BY (track_name, artists, duration_ms);
deduped_tracks = FOREACH grouped_tracks {
    sorted_tracks = ORDER filtered_tracks BY popularity DESC;
    top_track = LIMIT sorted_tracks 1;
    GENERATE FLATTEN(top_track);
};
```

Figure 1: Deduplication code

Splitting Artists

Originally all of the artists were listed in a single field with ‘;’ acting as a delimiter between them. For my analysis I consider individual artists not necessarily groups of them, if they collaborated on a song. To achieve this, I split the field and flattened out all of the results into separate rows with the track_id and artist.



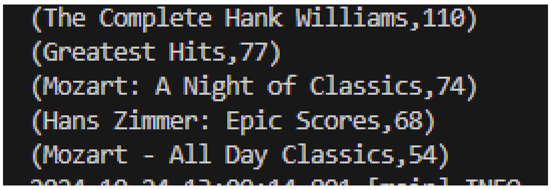
```
split_artists_single_row = FOREACH deduped_tracks GENERATE track_id, FLATTEN(STRSPLIT(artists, ';'));
split_artists_multiple_rows = FOREACH split_artists_single_row GENERATE $0, FLATTEN(TOBAG(*));
-- Have to remove instances where track_id is duplicate under artists due to FLATTEN(TOBAG(*)) opearation
split_artists_multiple_rows = FILTER split_artists_multiple_rows BY ($0 != $1);
```

Figure 2: Artist splitting code

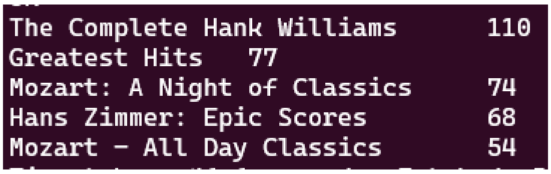
2 Simple Queries

What are the 5 albums with the most number of songs?

This query required a group by, count function, order and limit imposed in order to get the desired result:



```
(The Complete Hank Williams,110)
(Greatest Hits,77)
(Mozart: A Night of Classics,74)
(Hans Zimmer: Epic Scores,68)
(Mozart - All Day Classics,54)
```



```
The Complete Hank Williams      110
Greatest Hits   77
Mozart: A Night of Classics     74
Hans Zimmer: Epic Scores        68
Mozart - All Day Classics       54
```

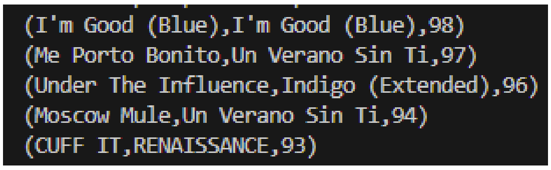
Figure 3: Query 1 Pig Output

Figure 4: Query 1 Hive Output

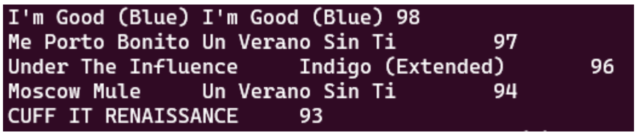
Looking at the results, it’s unsurprising that the largest albums are typically compilation albums where all of an artist's songs are compiled into a single album or a collection of hits. The results match above and it can be seen that the album “*The Complete Hank Williams”* has the most songs at 110 songs and the drop in total tracks is pretty severe with the 5th album being “*Mozart - All Day Classics*” with less than half at 54 songs.

Which are the top 5 most popular explicit tracks and list the track_name, artist(s) and popularity score

This query required a selection of columns, where clause checking if a song is explicit, ordering by popularity and limiting the results.



```
(I'm Good (Blue),I'm Good (Blue),98)
(Me Porto Bonito,Un Verano Sin Ti,97)
(Under The Influence,Indigo (Extended),96)
(Moscow Mule,Un Verano Sin Ti,94)
(CUFF IT,RENAISSANCE,93)
```



```
I'm Good (Blue) I'm Good (Blue) 98
Me Porto Bonito Un Verano Sin Ti        97
Under The Influence     Indigo (Extended)       96
Moscow Mule     Un Verano Sin Ti        94
CUFF IT RENAISSANCE     93
```

Figure 5: Query 2 Pig Output

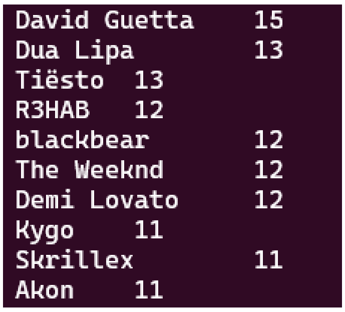
Figure 6: Query 2 Hive Output

Above, we can see that the results match and can see that even though the songs were explicit, they held very high *“popularity”* scores. The second and fourth songs were both part of the *“Un Verano Sin Ti ”* album indicating that the album was popular at the time of data collection.

3 Complex Queries

Which artists have the most unique genres in their song catalogue? (Complex Join)

Using the split artists and tracks, I can now perform a join in order to see how many tracks an individual artist has been a part off. I will leverage this to see which artists have produced songs in the most unique genres.



```
David Guetta    15
Dua Lipa        13
Tiësto  13
R3HAB   12
blackbear       12
The Weeknd      12
Demi Lovato     12
Kygo    11
Skrillex        11
Akon    11
```

Figure 7: Top 10 Artists with most unique genres

Above in the results we can see that the artist with the most unique genres is David Guetta at just 15. The data has 114 unique genres which shows that no artist comes close to making songs across most genres. We can also see that David Guetta is a bit of an outlier as Dua Lipa/Tiesto are the closest with 13 and it drops to 11 within the top 10.

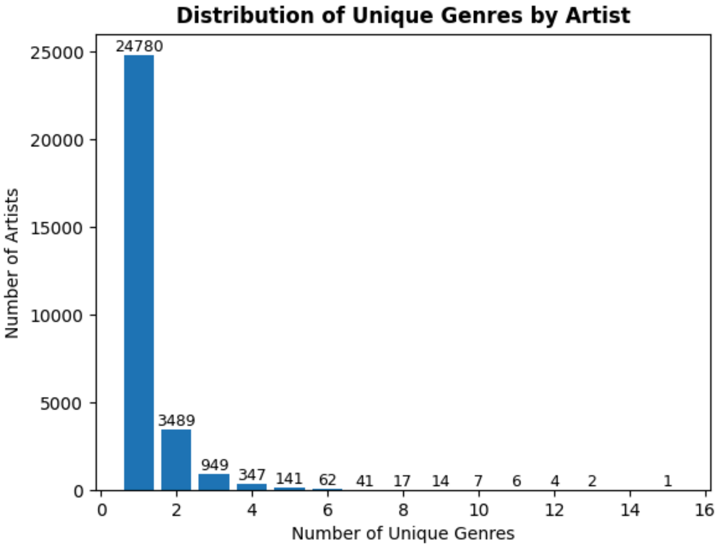
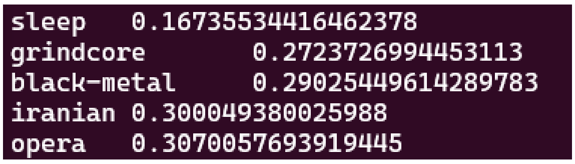


Figure 8: Distribution of Unique Genres by Artists

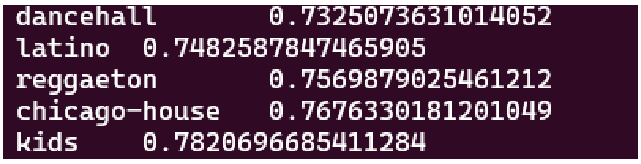
Taking all of the artists into consideration, we can see that the **vast majority** of artists only publish songs in one genre at 24,780. There is a huge fall-off with artists that published music in two genres at just 3,489 and it continues to get smaller. This shows that artists tend to specialise in a specific genre and don't diversify their music too much.

Which genres have the highest/lowest average danceability across all of their songs? (Aggregate Function)

Using the avg() function and a group by genre, we can determine which genres are on average more/less danceable than others.



```
sleep   0.16735534416462378
grindcore       0.2723726994453113
black-metal     0.29025449614289783
iranian 0.300049380025988
opera   0.3070057693919445
```



```
dancehall       0.7325073631014052
latino  0.7482587847465905
reggaeton       0.7569879025461212
chicago-house   0.7676330181201049
kids    0.7820696685411284
```

Figure 9: Top 5 lowest danceability genres

Figure 10: Top 5 highest danceability genres

Looking at the above results we can see that in the lowest danceability genres, unsurprisingly we have examples like sleep and opera which people typically don't dance to and sleep is by far the least danceable with a score of just 0.167. On the other hand, the highest scores in general are seen with genres like kids and dancehall with scores around 0.7-0.8 which is also unsurprising as most music made for kids or dancing is danceable.

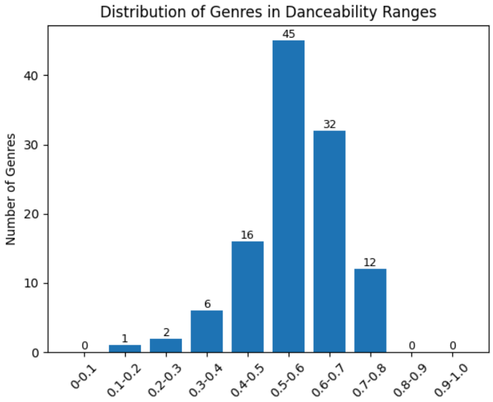
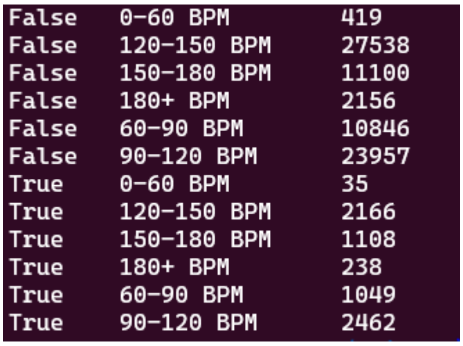


Figure 11: Distribution of Genres in Danceability ranges

As seen above in the figure, most genres are average in danceability between 0.5-0.7 with very little exceeding 0.7. However, we can see there are some genres reaching very low values between 0.1-0.3 showing that there are some genres that are not danceable at all but also there aren't any genres that are extremely danceable across all tracks.

In a sample of explicit and non-explicit tracks, is there a difference between the tempo of songs? (Sampling)

For the last query, I used a subquery and sampled just on 10% of the data. I created tempo ranges to see if there was a difference between the proportions of tempo between explicit and non-explicit tracks.



```
False   0-60 BPM        419
False   120-150 BPM     27538
False   150-180 BPM     11100
False   180+ BPM        2156
False   60-90 BPM       10846
False   90-120 BPM      23957
True    0-60 BPM        35
True    120-150 BPM     2166
True    150-180 BPM     1108
True    180+ BPM        238
True    60-90 BPM       1049
True    90-120 BPM      2462
```

Figure 12: Results of query

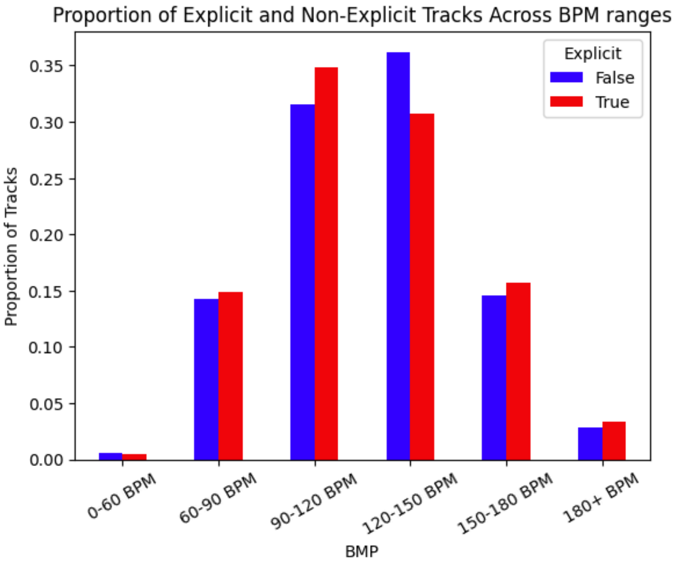


Figure 13: Distribution of tempo in explicit and non-explicit tracks

Interestingly, there isn’t a large difference between the distribution of explicit and non-explicit music. Both have similar ranges with non explicit tracks having a slightly higher proportion between 120-150 BPM. While there are slightly more explicit tracks proportionally in the 150 BPM+ (Beats Per Minute) range, the difference isn’t massive and it’s not necessarily true that explicit music tends to be more intense in terms of BPM like one might expect.