Pipeline Design Report

Author: The Vinh (Vin) Ha

I. Data Extraction and Organization

1. Connect to MongoDB and select database "Assignment1" and collection "Song"

2. Initialize an empty list for output

3. Retrieve documents from the collection

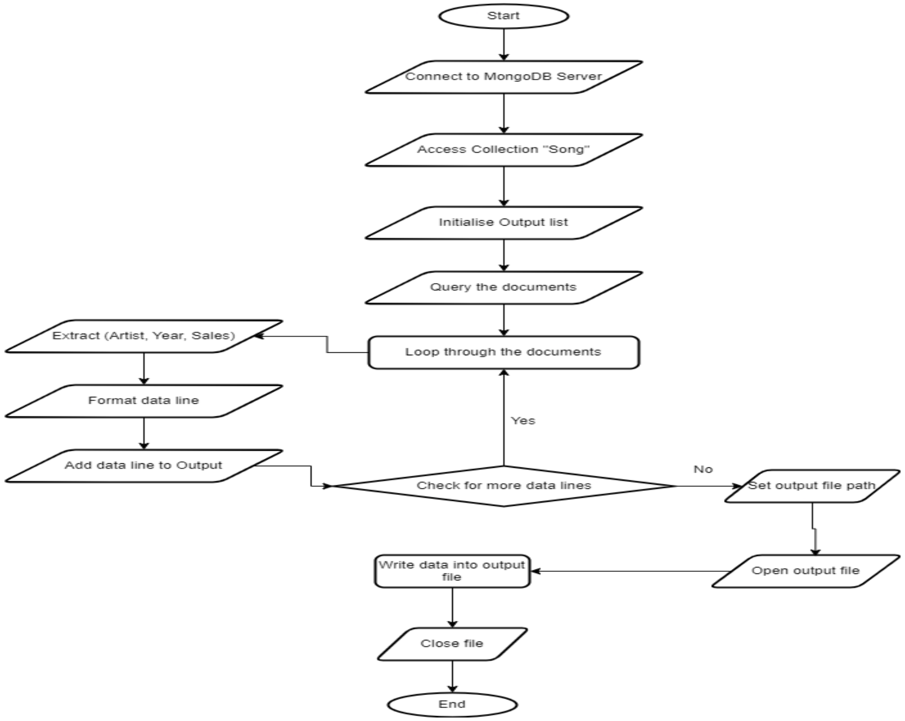
3.1. For each document:

3.1.1. Extract artist, year, and sales data

3.1.2. Concatenate them into a string triplet and append to the output list

4. Specify the output file path

5. Write output data to the output text file



II. Data Transformation and Loading

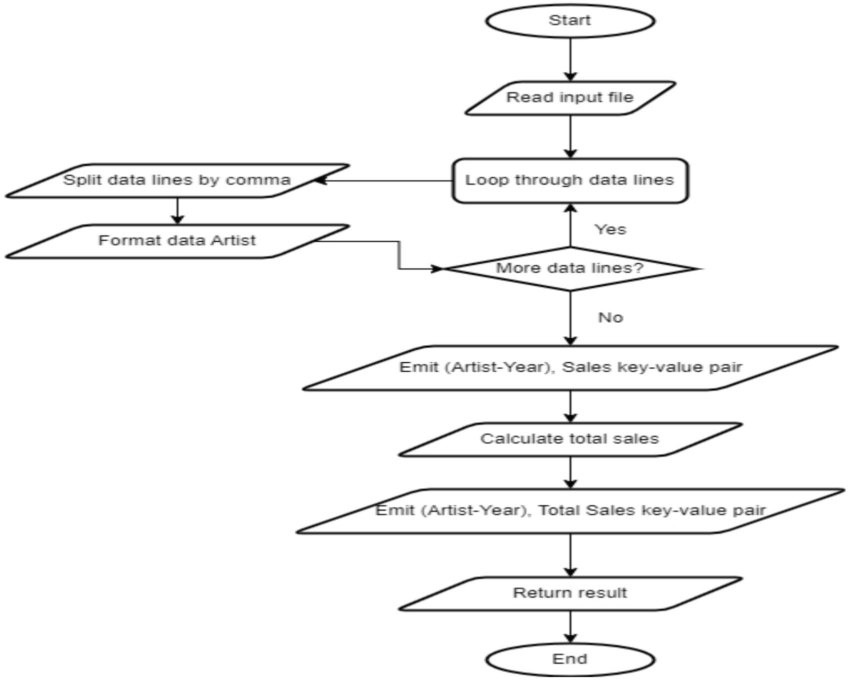
The Annual Total Sales Module:

1. Define a MapReduce job class named TotalSaleEachYearByArtist

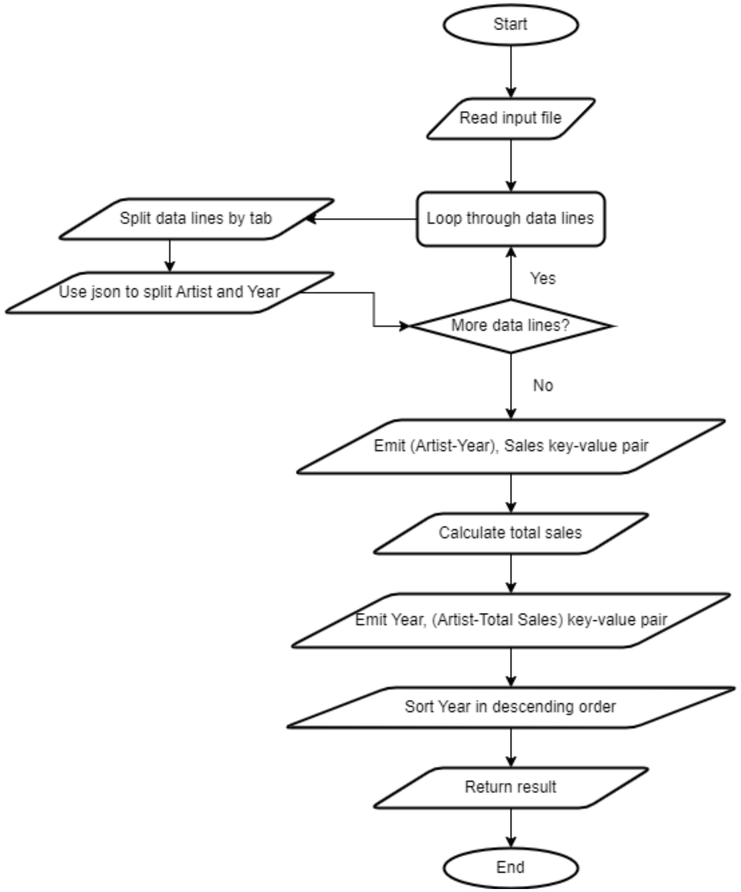
```
    1.1. Define mapper function:
        1.1.1. Split input line into artists, year, and sales
        1.1.2. Emit key-value pair: ((artists, year), sales)
    1.2. Define reducer function:
        1.2.1. Aggregate total sales for each artist-year pair
        1.2.2. Emit artist-year pair and total sales
    1.3. Define job steps:
        1.3.1. Set mapper and reducer functions
2. Execute the MapReduce job if this script is run as main
```

The Annual Top Sales Module:

```
1. Define a MapReduce job class named TopArtistEachYear
    1.1. Define mapper function:
        1.1.1. Split input line into data and sales
        1.1.2. Parse JSON data to extract artist and year
        1.1.3. Convert sales to float
        1.1.4. Emit key-value pair: ((year, artist), sales)
    1.2. Define reducer_sum_sales function:
        1.2.1. Aggregate total sales for each year-artist pair
```



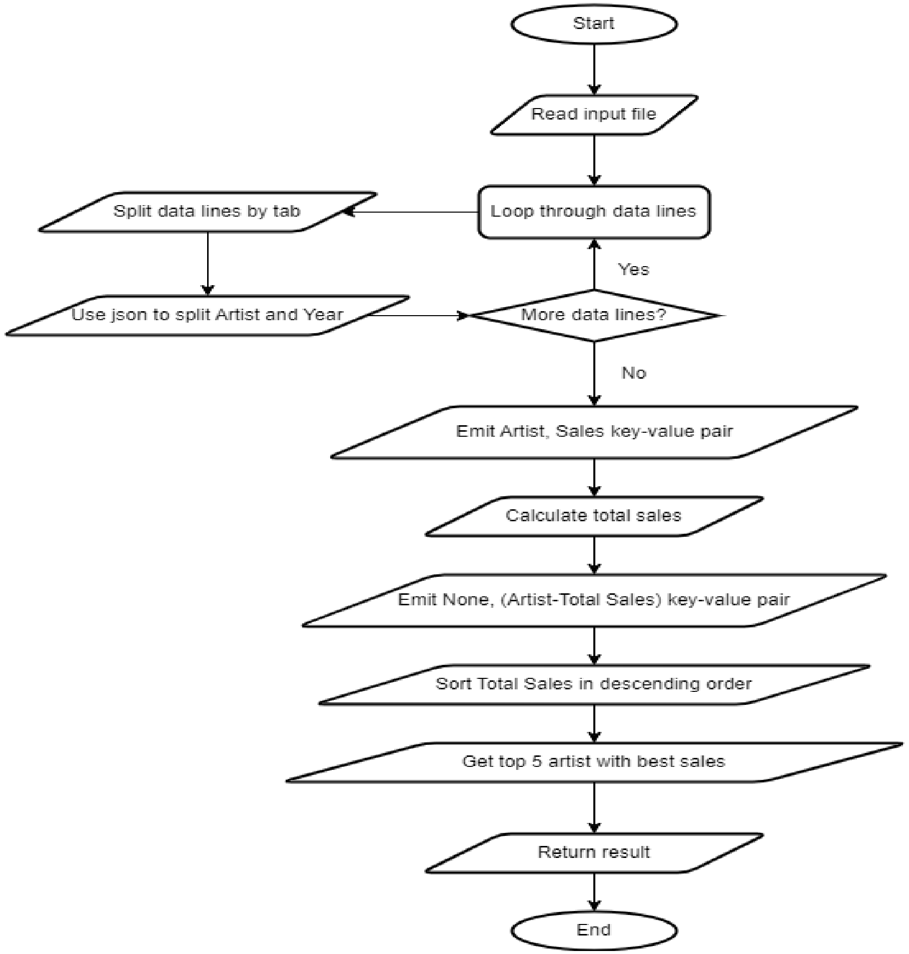
```
    1.2.2. Yield year and (artist, total_sales) as output
  1.3. Define reducer_best_sales function:
    1.3.1. Find the artist with the highest sales for each year
    1.3.2. Yield year and the artist with the highest sales
  1.4. Define mapper_prepare_for_sorting function:
    1.4.1. Create a sorting key for each year to ensure sorting in descending order
    1.4.2. Yield sorting_key and (year, artist_sales) as output
  1.5. Define reducer_final_output function:
    1.5.1. Iterate over values and yield year and artist_sales
  1.6. Define job steps:
    1.6.1. First step: Sum up sales for each year-artist pair
    1.6.2. Second step: Find the artist with the highest sales for each year
    1.6.3. Third step: Prepare for sorting and output
2. Execute the MapReduce job if this script is run as main
```


Start
Read input file
Loop through data lines
Split data lines by tab
Use json to split Artist and Year
Yes
More data lines?
No
Emit (Artist-Year), Sales key-value pair
Calculate total sales
Emit Year, (Artist-Total Sales) key-value pair
Sort Year in descending order
Return result
End

The Top 5 Best Sellers Module:

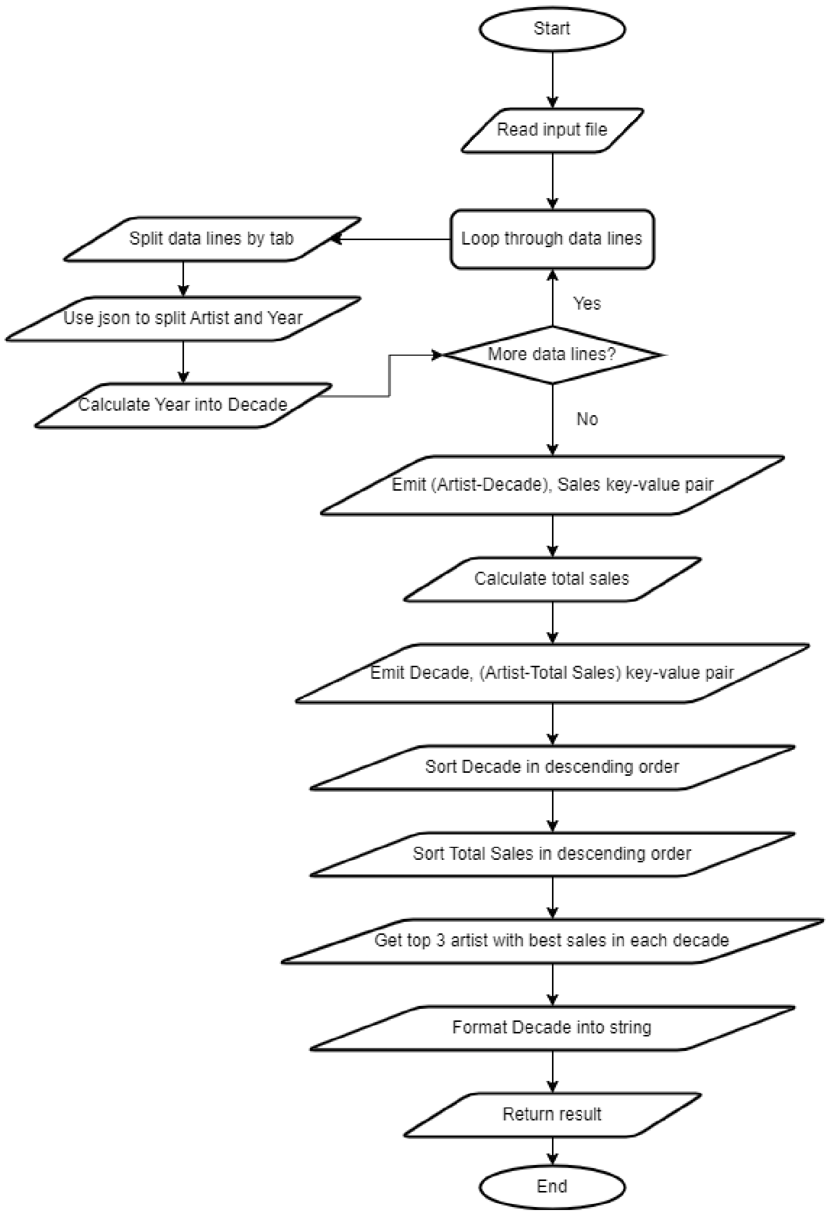
```
1. Define a MapReduce job class named TopSellingArtist
  1.1. Define mapper function:
    1.1.1. Split input line into data and sales
```

```
        1.1.2. Parse JSON data to extract artist
        1.1.3. Convert sales to float
        1.1.4. Emit artist and sales as key-value pair
    1.2. Define reducer_sum_sales function:
        1.2.1. Aggregate total sales for each artist
        1.2.2. Yield None as key and (artist, total_sales) as value
    1.3. Define reducer_top_5_artists function:
        1.3.1. Sort artist_sales by total sales in descending order
        1.3.2. Select top 5 artists with highest sales
        1.3.3. Yield artist and total sales for each of the top 5 artists
    1.4. Define job steps:
        1.4.1. First step: Sum up sales for each artist
        1.4.2. Second step: Find the top 5 selling artists
2. Execute the MapReduce job if this script is run as main
```


Start
Read input file
Loop through data lines
Split data lines by tab
Use json to split Artist and Year
More data lines?
Yes
No
Emit Artist, Sales key-value pair
Calculate total sales
Emit None, (Artist-Total Sales) key-value pair
Sort Total Sales in descending order
Get top 5 artist with best sales
Return result
End

The Best Sellers by Decades Module:

```
1. Define a MapReduce job class named TopSellingEachDecade
    1.1. Define mapper function:
        1.1.1. Split input line into data and sales
        1.1.2. Parse JSON data to extract artist and year
        1.1.3. Calculate the decade for the year
        1.1.4. Convert sales to float
        1.1.5. Emit key-value pair: ((decade, artist), sales)
    1.2. Define reducer_sum_sales function:
        1.2.1. Aggregate total sales for each decade-artist pair
        1.2.2. Yield decade and (artist, total_sales)
    1.3. Define reducer_sort_decades function:
        1.3.1. Sort decades and artist sales
    1.4. Define reducer_find_top_3_decade function:
        1.4.1. Sort decade-artist sales in descending order of decade
        1.4.2. For each decade, find the top 3 selling artists
        1.4.3. Yield decade range and the top 3 selling artists with total sales
    1.5. Define job steps:
        1.5.1. First step: Sum up sales for each decade-artist pair
        1.5.2. Second step: Sort decades and artist sales
        1.5.3. Third step: Find the top 3 selling artists for each decade
2. Execute the MapReduce job if this script is run as main
```


Start
Read input file
Loop through data lines
Split data lines by tab
Use json to split Artist and Year
Calculate Year into Decade
Yes
More data lines?
No
Emit (Artist-Decade), Sales key-value pair
Calculate total sales
Emit Decade, (Artist-Total Sales) key-value pair
Sort Decade in descending order
Sort Total Sales in descending order
Get top 3 artist with best sales in each decade
Format Decade into string
Return result
End