# Task 1

## Task 1.1

A diagram of a flowchart

Description automatically generated

1. Connect to MongoDB
2. Access the "song" collection within the "song" database.
3. Open a new text file for writing the extracted data.
4. Iterate over each document in the "song" collection.
5. Extract the "Artist", "Year", and "Sales" fields from each document.
6. Write the extracted data as triplets to the text file in the format "<Artist>,<Year>,<Sales>".
7. Close the text file.
8. Display a message indicating the completion of the process.

## Task 1.2

A diagram of a diagram

Description automatically generated with medium confidence

1. Read input line as data (in the format [artist, year, sales]) of the given file.
2. Extract artist, year, and sales from data.
3. Emit (artist, year) as key and sales as value.
4. Sum up the sales for each key (artist, year) in the reducer.
5. Emit key (artist, year) with total sales as the output.
6. Run the command using “python task1\_2.py task1\_1\_output.txt < task1\_2\_output.txt”

Where task1\_1\_output.txt is the input file and task1\_2\_output.txt is output file which will save all the results

# Task 2

## Task 2.1

A blue rectangles with white text

Description automatically generated

1. Read txt file
2. In the mapper function:
   1. Split the input line by tab to extract artist, year, and sales.
   2. Convert sales to float.
   3. Yield year as key and a tuple of artist and sales as value.
3. 5. In the reducer function:
   1. Find the artist with the maximum sales for each year using max() and lambda function.
   2. Yield None as key and a tuple of year and the artist with maximum sales as value.
4. In the sort\_by\_year function:
   1. Sort the data by year in descending order using sorted() with key=lambda x: x[0].
   2. Yield year as key and the corresponding data as value.
5. Run all the functions in sequence and result is stored in txt file task2\_1\_output.txt by running the script through “python task2\_1.py task1\_2\_output.txt < task2\_1\_output.txt”

## Task 2.2

A screenshot of a diagram

Description automatically generated

1. Read txt file
2. In the mapper function:
   1. Split the input line by tab to extract artist and sales.
   2. Convert sales to float.
   3. Yield artist as key and sales as value.
3. In the reducer function:
   1. Sum up the sales for each artist.
   2. Yield None as key and a tuple of artist and total sales as value.
4. In the sort\_by\_year function:
   1. Initialize a counter.
   2. Sort the data by total sales in descending order using sorted() with key=lambda x: x[1].
   3. Yield artist and total sales for the top 5 artists.
5. Run all the functions in sequence and result is stored in txt file task2\_2\_output.txt by running the script through “python task2\_2.py task1\_2\_output.txt < task2\_2\_output.txt”

## Task 2.3

A blue rectangular object with white text

Description automatically generated

1. Read txt file
2. In the mapper function:
   1. Split the input line by tab to extract artist, year, and sales.
   2. Convert sales to float.
   3. Determine the decade of the year.
   4. Yield (decade, artist) as key and sales as value.
3. In the reducer function:
   1. Sum up the sales for each artist within each decade.
   2. Yield decade as key and a tuple of total sales and artist as value.
4. In the sort\_by\_sales function:
   1. Sort the artists by total sales within each decade in descending order.
   2. Yield the top 3 selling artists for each decade.
5. In the sort\_by\_decade function:
   1. Sort the data by decade and total sales in descending order.
   2. Yield the top 3 selling artists with their total sales for each decade.
6. Run all the functions in sequence and result is stored in txt file task2\_3\_output.txt by running the script through “python task2\_3.py task1\_2\_output.txt < task2\_3\_output.txt”