# Structure Break Documentation

SQL SETUP:

* Install MySQL from the official website using the [link](https://dev.mysql.com/downloads/mysql/) given.
* For GUI install the corresponding version of the [workbench](https://dev.mysql.com/downloads/workbench/).

Data

* The original data has the following fields :
  + Date : In the form of ‘yyyy-mm-dd’.
  + vader\_stopwords
  + vader\_no\_stopwords
  + textblob\_stopwords
  + textblob\_no\_stopwords
  + finbert\_sentiment\_score
* The modified dataset includes the date in the form of ‘yyyy-mm-dd 23:59:59’ which shows the financial data updated into the database at the end of the day.
* The code for the conversion of data is given below :
  + CODE :

import pandas as pd

# Load the CSV file into a pandas DataFrame

df = pd.read\_csv('/content/daily\_sentiment\_score.csv')

# Replace the path to your actual CSV file

# Append ' 23:59:59' to the date column to make it compatible with the DATETIME format and ensure data is entered at the end of the day

df['date'] = df['date'].apply(lambda x: x + ' 23:59:59')

# Convert the 'date' column from text (string) to datetime format

df['date'] = pd.to\_datetime(df['date'], format='%Y-%m-%d %H:%M:%S')

# Display the updated DataFrame

print(df)

# Optional: Save the updated DataFrame back to a new CSV

df.to\_csv('/content/daily\_sentiment\_score\_analysis.csv', index=False)

Data Aggregation and Frequency Conversion

* The original data that is in the daily format is converted into weekly, monthly, and yearly aggregation for deriving better insights.
* The code for the data aggregation is given below as well in the code section of the repository :

import pandas as pd

# Load the data

file\_path = '/content/daily\_sentiment\_score\_analysis.csv'

daily\_data = pd.read\_csv(file\_path)

# Convert the 'date' column to DateTime format for resampling

daily\_data['date'] = pd.to\_datetime(daily\_data['date'])

# Set the date column as the index

daily\_data.set\_index('date', inplace=True)

# Resample data to weekly, monthly, and yearly frequencies and calculate mean

weekly\_data = daily\_data.resample('W').mean()

monthly\_data = daily\_data.resample('M').mean()

yearly\_data = daily\_data.resample('Y').mean()

# Save the weekly, monthly, and yearly data to CSV files

weekly\_data.to\_csv('weekly\_sentiment\_score\_analysis.csv')

monthly\_data.to\_csv('monthly\_sentiment\_score\_analysis.csv')

yearly\_data.to\_csv('yearly\_sentiment\_score\_analysis.csv')