**Assignment 4 – MongoDB & PyMongo**

Use Pymongo & Mongodb database to create a document of Flights Delay. Here is the data description.

**Dataset:** Flights\_Delay.csv

Dataset Description:

ID: Rows ID

YEAR: 2015

MONTH: 1-12

DAY: 1-31

DAY\_OF\_WEEK: 1 (Monday) - 7 (Sunday)

AIRLINE: Airline CODE

FLIGHT\_NUMBER: Flight Number

TAIL\_NUMBER: Flight’s tail number

ORIGIN\_AIRPORT: Origin IATA airport code

DESTINATION\_AIRPORT: Destination IATA airport code

SCHEDULED\_DEPARTURE: Actual departure time (local, hhmm)

DEPARTURE\_TIME: Scheduled departure time (local, hhmm)

DEPARTURE\_DELAY: Departure delay, in minutes

TAXI\_OUT: Taxi out time in minutes

WHEELS\_OFF:

SCHEDULED\_TIME: Scheduled arrival time (local, hhmm)

ELAPSED\_TIME: in Minutes

AIR\_TIME: in Minutes

DISTANCE: in Miles

WHEELS\_ON:

TAXI\_IN: Taxi in time, in minutes

SCHEDULED\_ARRIVAL: Scheduled arrival time (local, hhmm)

ARRIVAL\_TIME: Actual arrival time (local, hhmm)

ARRIVAL\_DELAY: Arrival delay, in minutes

DIVERTED: 1 = yes, 0 = no

CANCELLED: 1 = yes, 0 = no

1. Create collections “flights” inside database “airline\_delayDB”
2. How would you insert this entire dataset into a MongoDB collection named flights? Describe the structure of each document.
3. Write a MongoDB command to insert a single flight record from the dataset.

Write mongo queries to show following analysis

1. Write a MongoDB query to find all flights that were delayed by more than 60 minutes.
2. How would you query all flights that were cancelled (CANCELLED flag set to 1) and return only the AIRLINE, ORIGIN\_AIRPORT, and CANCELLATION\_REASON fields?
3. Using MongoDB’s aggregation framework, how would you calculate the average arrival delay (ARRIVAL\_DELAY) for each airline? [Create a suitable plot using matplotlib/seaborn]
4. Days of months with respect to average arrival delays. [Create a suitable plot using matplotlib/seaborn]
5. Write a MongoDB aggregation pipeline to find the top 10 airports with the highest average total delay (DEPARTURE\_DELAY + ARRIVAL DELAY).
6. Explain how you would create an index on the ORIGIN\_AIRPORT and DESTINATION\_AIRPORT fields to optimize queries filtering by these fields.
7. Arrange weekdays with respect to the average arrival delays caused. [Create a suitable plot using matplotlib/seaborn]
8. Arrange Days of month as per cancellations done in descending order. [Create a suitable plot using matplotlib/seaborn]
9. Find the busiest airports with respect to day of week. Represent it by using suitable plot.
10. **Find top 10 Airlines of US. Represent it by using suitable plot.**
11. Finding airlines that make the maximum, minimum number of cancellations.
12. Find and show airlines names in descending that make the most number of diversions made. [Create a suitable plot using matplotlib/seaborn]
13. Finding days of month that see the most number of diversion and delays.
14. Write a MongoDB query to find the flights with the shortest and longest AIR\_TIME. Return the flightNumber, airline, and AIR\_TIME.
15. Finding all diverted Route from a source to destination Airport & which route is the most diverted route.
16. Write a MongoDB aggregation pipeline to calculate the all aggregated values for departure delay (DEPARTURE\_DELAY) and arrival delay (ARRIVAL\_DELAY) for each airline, excluding flights that were either cancelled or diverted.
17. Write a MongoDB query to find all flights that were delayed due to WEATHER\_DELAY but were not cancelled or diverted. Include the flightNumber, airline, originAirport, and destinationAirport in the results.
18. Write a MongoDB query to find all flights that were delayed both at departure (DEPARTURE\_DELAY) and arrival (ARRIVAL\_DELAY). Return the count of such Flights which are delayed.
19. Write a MongoDB query to calculate the frequency of flight takeoffs and landings within defined time intervals (e.g., every hour) throughout the day. Generate a Suitable Plot.
20. Write a MongoDB query to calculate the frequency of flight takeoffs and landings within defined week of day. Generate a Suitable Plot.
21. Write a MongoDB query to find all flights that departed between 6 AM and 12 PM (noon) local time, regardless of the date. Return the flightNumber, airline, and departureTime. Generate a Bar Plot using Time (x-axis) and Frequency (y-axis).
22. **When is the best time of day/day of week/time of a year to fly with minimum delays?**
23. Create a partitioning table “flights\_partition” using partitioned by schema “**Airports”**
24. Write a MongoDB Map-Reduce function to calculate the total delay time for each airline, including all types of delays (e.g., AIR\_SYSTEM\_DELAY, SECURITY\_DELAY, etc.).
25. Explain how you could use MongoDB’s Map-Reduce feature to find the most common origin-destination airport pairs with the longest delays.