**Kafka-MongoDB Doc**

This file details and describes all the attached files for this Kafka-MongoDB project

**Objective:**

Developed a Python-based application that integrates Kafka and MongoDB to process logistics data. The application involved a Kafka producer and consumer, data serialization/deserialization with Avro, and data ingestion into MongoDB.

**Tools Used:**

1. Python3
2. Confluent Kafka
3. MongoDB Atlas
4. MongoDB Compass

**Files Attached:**

1. delivery\_trip\_truck\_data.csv – The csv raw data used to push to the kafka topic
2. logistics\_data\_producer.py – Python producer script
3. logistics\_data\_consumer.py – Python consumer script

**Process and File Descriptions:**

**Step 1**

Created a kafka topic called ‘logistics\_data’ with 6 partitions and I made sure to save the API keys for the producer. I also created an appropriate schema value and key to prepare the kafka topic for data ingestion/retrieval looking at the delivery\_trip\_truck\_data.csv file. I especially made sure to handle the ‘Nan’ values by replacing them with the string ‘unknown value’ if the field is string type.

**Step 2**

I created a producer script called “logistics\_data\_producer.py” that produces the data to the afore- mentioned Kafka topic. The script also serializes the data into Avro format and uses GPSProvider as the key. The below image shows the producer fetching data. It also sends out a message saying that the record value has been successfully produced in a particular partition.

A screen shot of a computer screen

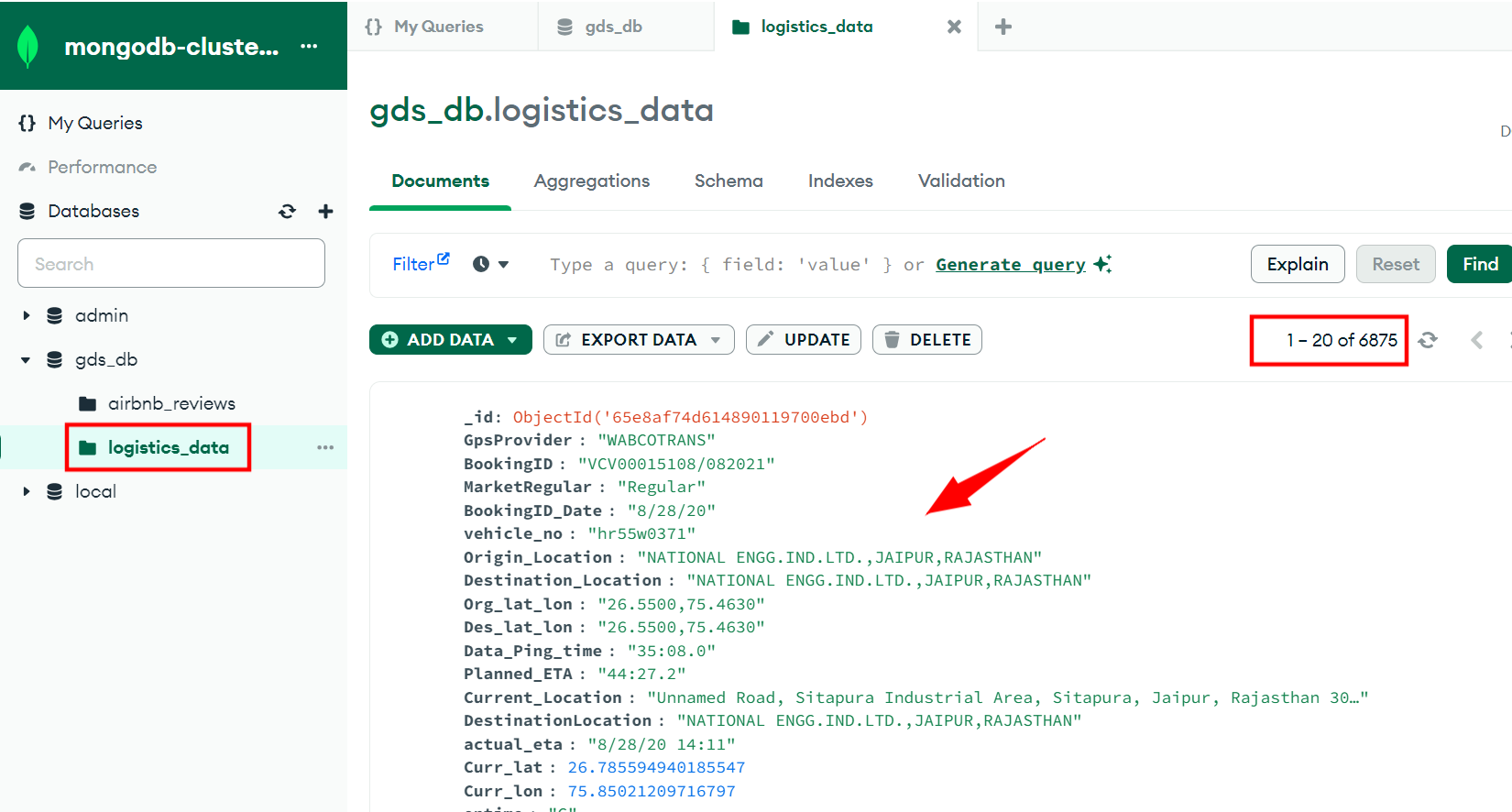
Description automatically generated

A screen shot of a computer screen

Description automatically generated

**Step 3**

Created a mongodb database called ‘gds\_db’ and created an empty collection called ‘logistics\_data’ so that data can be stored once a consumer script can be run.



**Step 4**

I then created a consumer script called “logistics\_data\_consumer.py” that deserializes the avro data back into a python object. I then implemented data validation checks in the code to make sure that it accounts for null values and correct data types checks. Before pushing the data into the logistics\_data collection that was created in the gdb\_db mongodb database, I make sure that there are no duplicate records pushed when the consumer runs.

**Step 5**

We can also check the data using Mongodb Compass:

