Delivery Logistics Data Ingestion Kafka-MongoDB

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

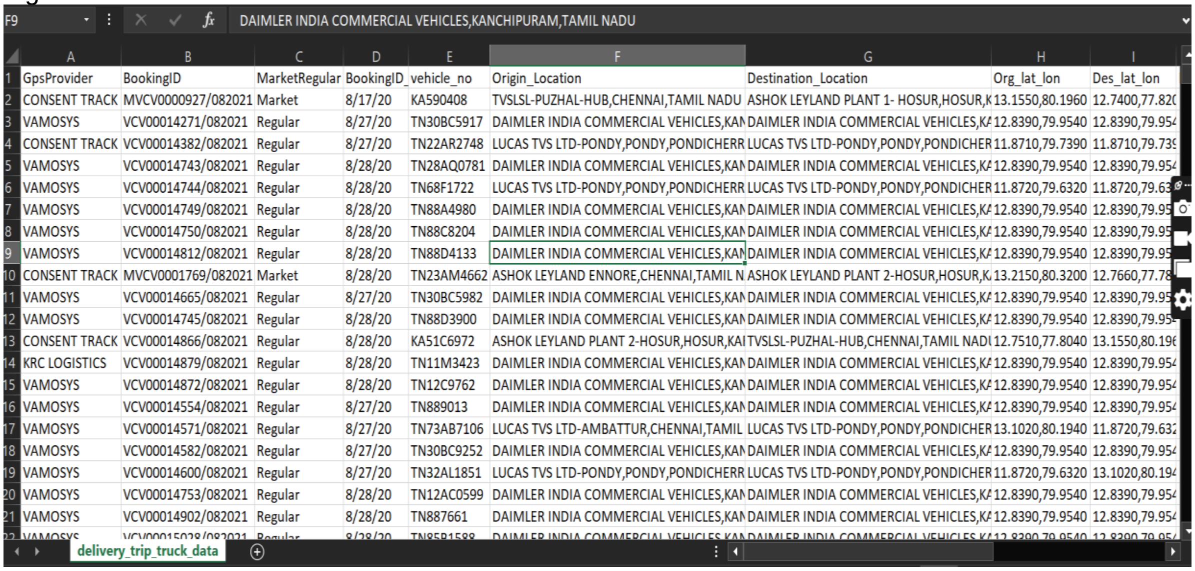
Files Attached:

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:

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.

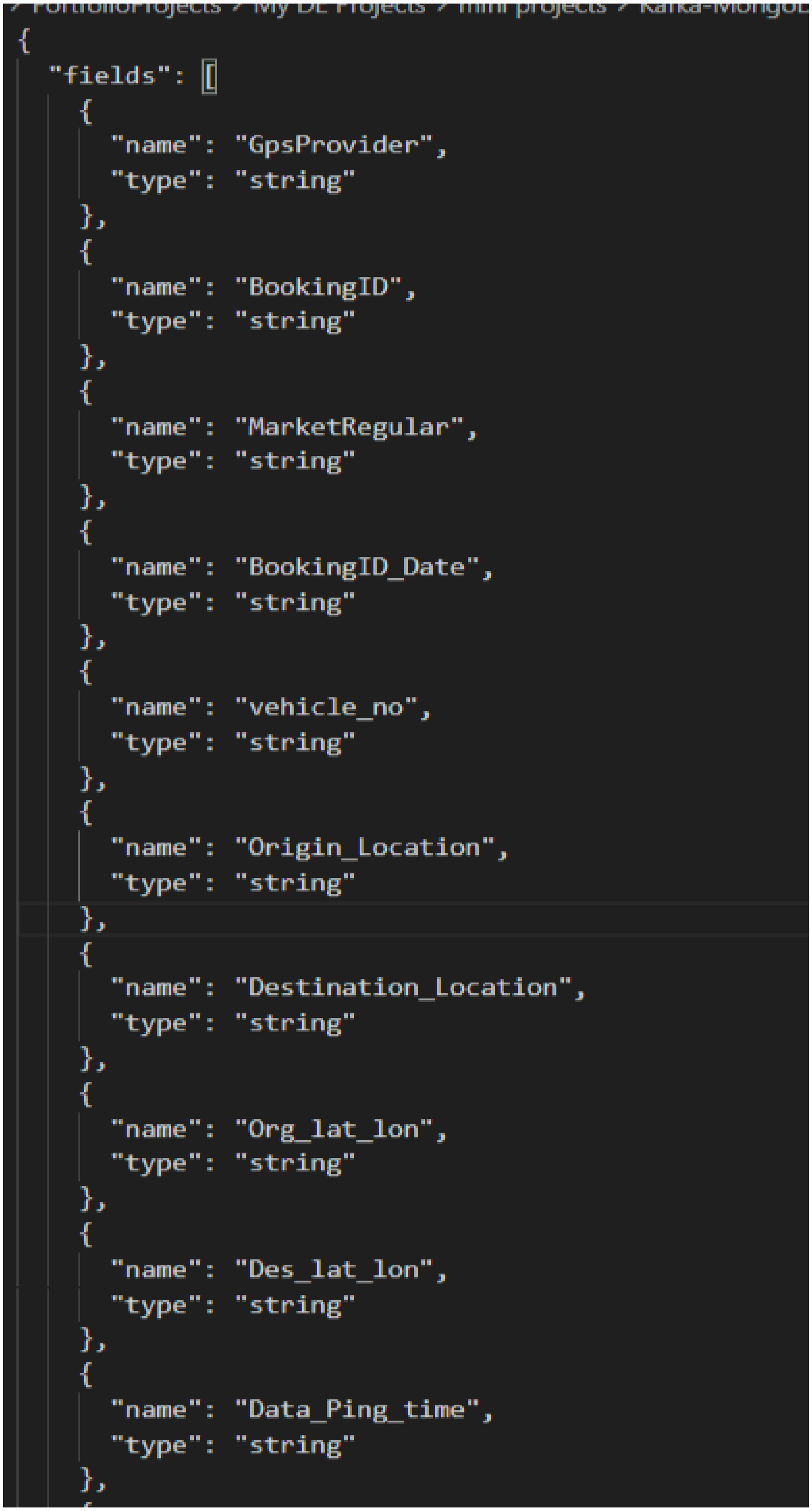
logistics csv:



F9 | DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU

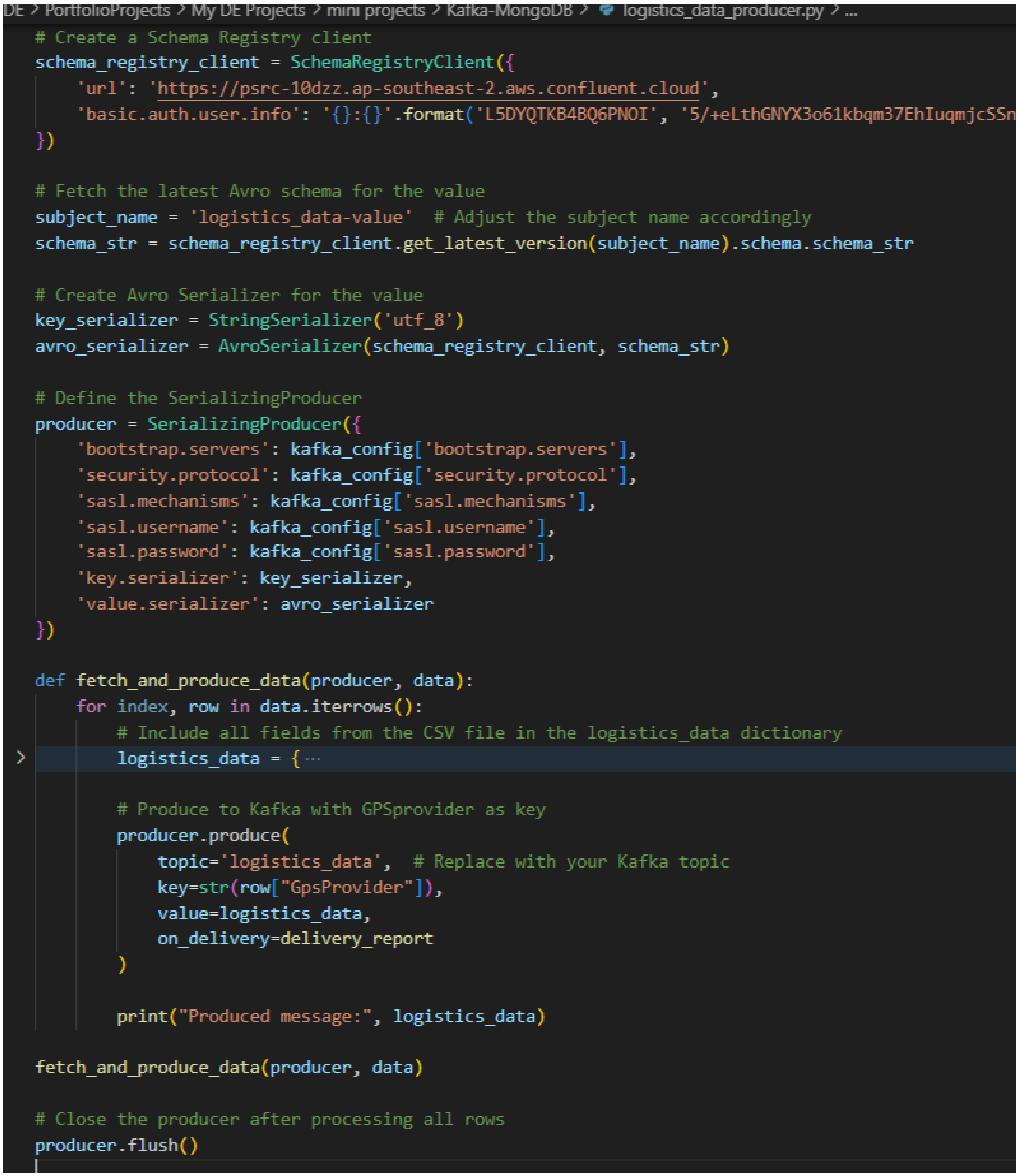
	A	B	C	D	E	F	G	H	I
1	GpsProvider	BookingID	MarketRegular	BookingID_	vehicle_no	Origin_Location	Destination_Location	Org_lat_lon	Des_lat_lon
2	CONSENT TRACK	MVCV0000927/082021	Market	8/17/20	KA590408	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	ASHOK LEYLAND PLANT 1- HOSUR,HOSUR,K	13.1550,80.1960	12.7400,77.820
3	VAMOSYS	VCV00014271/082021	Regular	8/27/20	TN30BC5917	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
4	CONSENT TRACK	VCV00014382/082021	Regular	8/27/20	TN22AR2748	LUCAS TVS LTD-PONDY,PONDY,PONDICHERR	LUCAS TVS LTD-PONDY,PONDY,PONDICHER	11.8710,79.7390	11.8710,79.739
5	VAMOSYS	VCV00014743/082021	Regular	8/28/20	TN28AQ0781	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
6	VAMOSYS	VCV00014744/082021	Regular	8/28/20	TN68F1722	LUCAS TVS LTD-PONDY,PONDY,PONDICHERR	LUCAS TVS LTD-PONDY,PONDY,PONDICHER	11.8720,79.6320	11.8720,79.63
7	VAMOSYS	VCV00014749/082021	Regular	8/28/20	TN88A4980	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.95
8	VAMOSYS	VCV00014750/082021	Regular	8/28/20	TN88C8204	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.95
9	VAMOSYS	VCV00014812/082021	Regular	8/28/20	TN88D4133	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.95
10	CONSENT TRACK	MVCV0001769/082021	Market	8/28/20	TN23AM4662	ASHOK LEYLAND ENNORE,CHENNAI,TAMIL N	ASHOK LEYLAND PLANT 2-HOSUR,HOSUR,K	13.2150,80.3200	12.7660,77.78
11	VAMOSYS	VCV00014665/082021	Regular	8/27/20	TN30BC5982	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.95
12	VAMOSYS	VCV00014745/082021	Regular	8/28/20	TN88D3900	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
13	CONSENT TRACK	VCV00014866/082021	Regular	8/28/20	KA51C6972	ASHOK LEYLAND PLANT 2-HOSUR,HOSUR,KAI	TVSLSL-PUZHAL-HUB,CHENNAI,TAMIL NADU	12.7510,77.8040	13.1550,80.196
14	KRC LOGISTICS	VCV00014879/082021	Regular	8/28/20	TN11M3423	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
15	VAMOSYS	VCV00014872/082021	Regular	8/28/20	TN12C9762	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
16	VAMOSYS	VCV00014554/082021	Regular	8/27/20	TN889013	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
17	VAMOSYS	VCV00014571/082021	Regular	8/27/20	TN73AB7106	LUCAS TVS LTD-AMBATTUR,CHENNAI,TAMIL	LUCAS TVS LTD-PONDY,PONDY,PONDICHER	13.1020,80.1940	11.8720,79.632
18	VAMOSYS	VCV00014582/082021	Regular	8/27/20	TN30BC9252	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
19	VAMOSYS	VCV00014600/082021	Regular	8/27/20	TN32AL1851	LUCAS TVS LTD-PONDY,PONDY,PONDICHERR	LUCAS TVS LTD-PONDY,PONDY,PONDICHER	11.8720,79.6320	13.1020,80.194
20	VAMOSYS	VCV00014753/082021	Regular	8/28/20	TN12AC0599	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
21	VAMOSYS	VCV00014902/082021	Regular	8/28/20	TN887661	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954
22	VAMOSYS	VCV00015028/082021	Regular	8/28/20	TN85B1588	DAIMLER INDIA COMMERCIAL VEHICLES,KAN	DAIMLER INDIA COMMERCIAL VEHICLES,KA	12.8390,79.9540	12.8390,79.954

avro schema serializer format:


> PortfolioProjects > My DL Projects > mini projects > Kafka-MongoD

```
{
  "fields": [
    {
      "name": "GpsProvider",
      "type": "string"
    },
    {
      "name": "BookingID",
      "type": "string"
    },
    {
      "name": "MarketRegular",
      "type": "string"
    },
    {
      "name": "BookingID_Date",
      "type": "string"
    },
    {
      "name": "vehicle_no",
      "type": "string"
    },
    {
      "name": "Origin_Location",
      "type": "string"
    },
    {
      "name": "Destination_Location",
      "type": "string"
    },
    {
      "name": "Org_lat_lon",
      "type": "string"
    },
    {
      "name": "Des_lat_lon",
      "type": "string"
    },
    {
      "name": "Data_Ping_time",
      "type": "string"
    },
    {
```

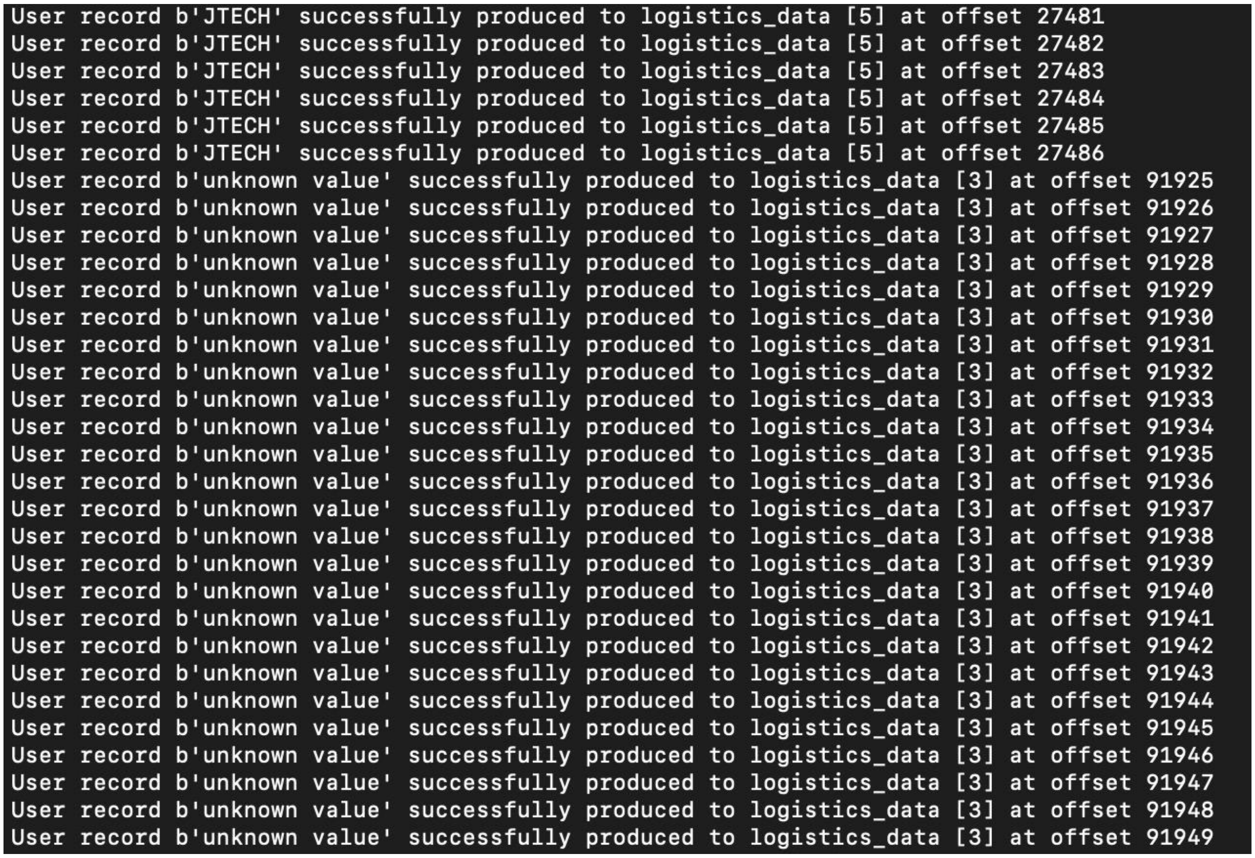
2. I created a producer script called “logistics_data_producer.py” that produces the data to the afore- mentioned Kafka topic. The script also serializes the data into Avro format and uses GPSProvider as the key.



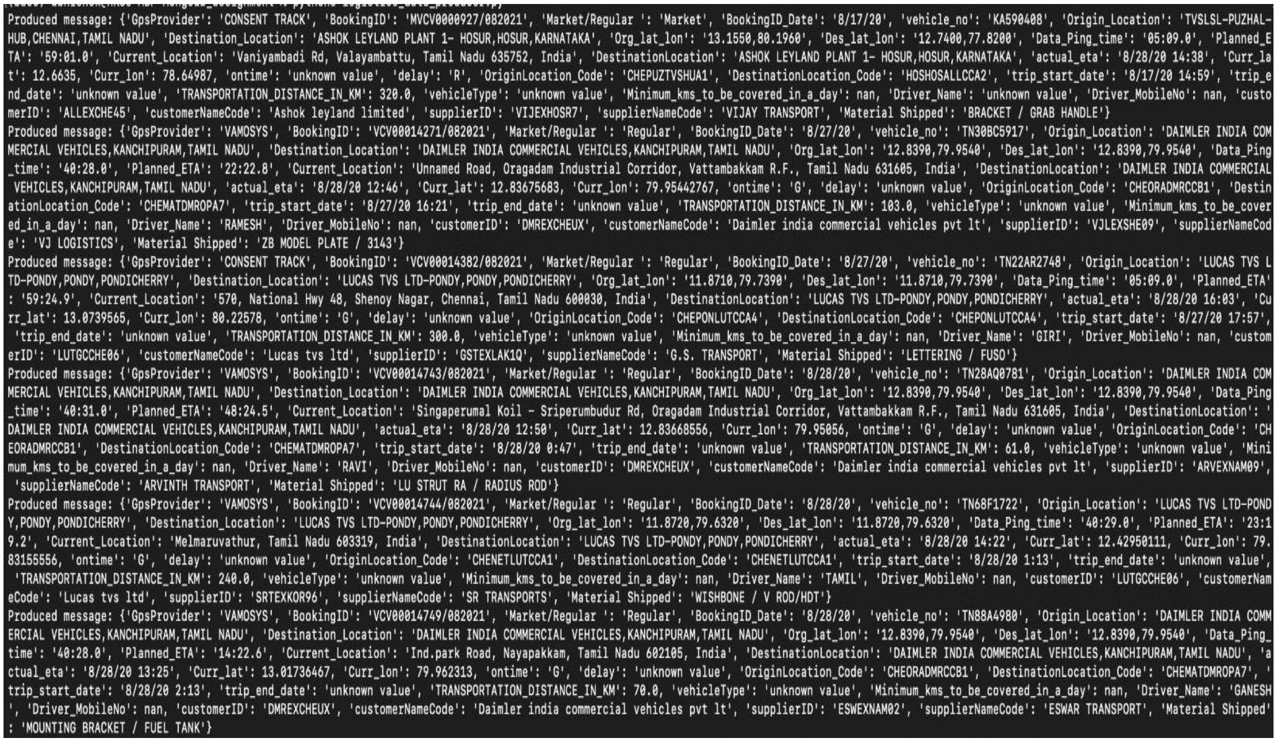
DE > PortfolioProjects > My DE Projects > mini projects > Kafka-MongoDB > logistics_data_producer.py > ...

```python
# Create a Schema Registry client
schema_registry_client = SchemaRegistryClient({
    'url': 'https://psrc-10dzz.ap-southeast-2.aws.confluent.cloud',
    'basic.auth.user.info': '{}:{}'.format('L5DYQTKB4BQ6PNOI', '5/+eLthGNYX3o61kbqm37EhIuqmjcSSn
})

# Fetch the latest Avro schema for the value
subject_name = 'logistics_data-value'  # Adjust the subject name accordingly
schema_str = schema_registry_client.get_latest_version(subject_name).schema.schema_str

# Create Avro Serializer for the value
key_serializer = StringSerializer('utf_8')
avro_serializer = AvroSerializer(schema_registry_client, schema_str)

# Define the SerializingProducer
producer = SerializingProducer({
    'bootstrap.servers': kafka_config['bootstrap.servers'],
    'security.protocol': kafka_config['security.protocol'],
    'sasl.mechanisms': kafka_config['sasl.mechanisms'],
    'sasl.username': kafka_config['sasl.username'],
    'sasl.password': kafka_config['sasl.password'],
    'key.serializer': key_serializer,
    'value.serializer': avro_serializer
})

def fetch_and_produce_data(producer, data):
    for index, row in data.iterrows():
        # Include all fields from the CSV file in the logistics_data dictionary
        logistics_data = {···

        # Produce to Kafka with GPSprovider as key
        producer.produce(
            topic='logistics_data',  # Replace with your Kafka topic
            key=str(row["GpsProvider"]),
            value=logistics_data,
            on_delivery=delivery_report
        )

        print("Produced message:", logistics_data)

fetch_and_produce_data(producer, data)

# Close the producer after processing all rows
producer.flush()
```

3. 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.

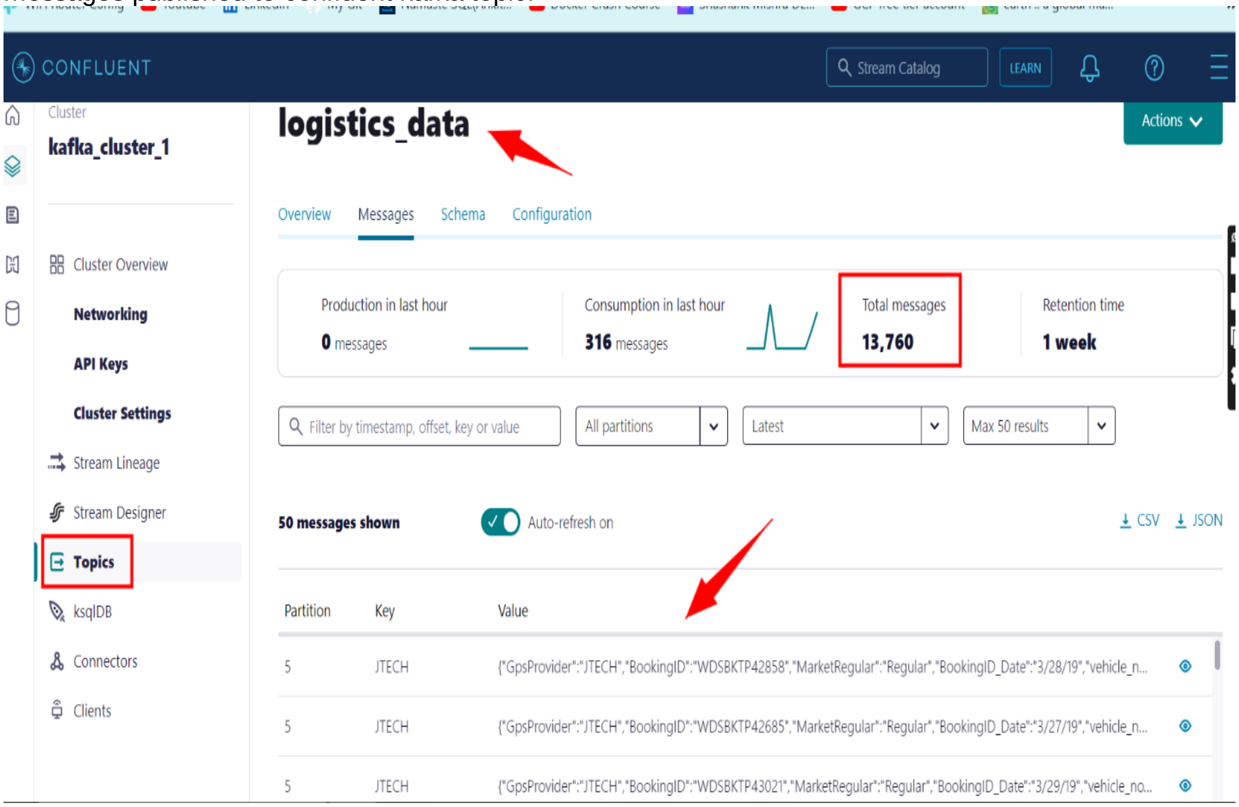


```
User record b'JTECH' successfully produced to logistics_data [5] at offset 27481
User record b'JTECH' successfully produced to logistics_data [5] at offset 27482
User record b'JTECH' successfully produced to logistics_data [5] at offset 27483
User record b'JTECH' successfully produced to logistics_data [5] at offset 27484
User record b'JTECH' successfully produced to logistics_data [5] at offset 27485
User record b'JTECH' successfully produced to logistics_data [5] at offset 27486
User record b'unknown value' successfully produced to logistics_data [3] at offset 91925
User record b'unknown value' successfully produced to logistics_data [3] at offset 91926
User record b'unknown value' successfully produced to logistics_data [3] at offset 91927
User record b'unknown value' successfully produced to logistics_data [3] at offset 91928
User record b'unknown value' successfully produced to logistics_data [3] at offset 91929
User record b'unknown value' successfully produced to logistics_data [3] at offset 91930
User record b'unknown value' successfully produced to logistics_data [3] at offset 91931
User record b'unknown value' successfully produced to logistics_data [3] at offset 91932
User record b'unknown value' successfully produced to logistics_data [3] at offset 91933
User record b'unknown value' successfully produced to logistics_data [3] at offset 91934
User record b'unknown value' successfully produced to logistics_data [3] at offset 91935
User record b'unknown value' successfully produced to logistics_data [3] at offset 91936
User record b'unknown value' successfully produced to logistics_data [3] at offset 91937
User record b'unknown value' successfully produced to logistics_data [3] at offset 91938
User record b'unknown value' successfully produced to logistics_data [3] at offset 91939
User record b'unknown value' successfully produced to logistics_data [3] at offset 91940
User record b'unknown value' successfully produced to logistics_data [3] at offset 91941
User record b'unknown value' successfully produced to logistics_data [3] at offset 91942
User record b'unknown value' successfully produced to logistics_data [3] at offset 91943
User record b'unknown value' successfully produced to logistics_data [3] at offset 91944
User record b'unknown value' successfully produced to logistics_data [3] at offset 91945
User record b'unknown value' successfully produced to logistics_data [3] at offset 91946
User record b'unknown value' successfully produced to logistics_data [3] at offset 91947
User record b'unknown value' successfully produced to logistics_data [3] at offset 91948
User record b'unknown value' successfully produced to logistics_data [3] at offset 91949
```



```
Produced message: {'GpsProvider': 'CONSENT TRACK', 'BookingID': 'MVCV0000927/082021', 'Market/Regular ': 'Market', 'BookingID_Date': '8/17/20', 'vehicle_no': 'KA590408', 'Origin_Location': 'TVSLSL-PUZHAL-
HUB,CHENNAI,TAMIL NADU', 'Destination_Location': 'ASHOK LEYLAND PLANT 1- HOSUR,HOSUR,KARNATAKA', 'Org_lat_lon': '13.1550,80.1960', 'Des_lat_lon': '12.7400,77.8200', 'Data_Ping_time': '05:09.0', 'Planned_E
TA': '59:01.0', 'Current_Location': 'Vaniyambadi Rd, Valayambattu, Tamil Nadu 635752, India', 'DestinationLocation': 'ASHOK LEYLAND PLANT 1- HOSUR,HOSUR,KARNATAKA', 'actual_eta': '8/28/20 14:38', 'Curr_la
t': 12.6635, 'Curr_lon': 78.64987, 'ontime': 'unknown value', 'delay': 'R', 'OriginLocation_Code': 'CHEPUZTVSHUA1', 'DestinationLocation_Code': 'HOSHOSALLCCA2', 'trip_start_date': '8/17/20 14:59', 'trip_e
nd_date': 'unknown value', 'TRANSPORTATION_DISTANCE_IN_KM': 320.0, 'vehicleType': 'unknown value', 'Minimum_kms_to_be_covered_in_a_day': nan, 'Driver_Name': 'unknown value', 'Driver_MobileNo': nan, 'custo
merID': 'ALLEXCHE45', 'customerNameCode': 'Ashok leyland limited', 'supplierID': 'VIJEXHOSR7', 'supplierNameCode': 'VIJAY TRANSPORT', 'Material Shipped': 'BRACKET / GRAB HANDLE'}
Produced message: {'GpsProvider': 'VAMOSYS', 'BookingID': 'VCV00014271/082021', 'Market/Regular ': 'Regular', 'BookingID_Date': '8/27/20', 'vehicle_no': 'TN30BC5917', 'Origin_Location': 'DAIMLER INDIA COM
MERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Destination_Location': 'DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Org_lat_lon': '12.8390,79.9540', 'Des_lat_lon': '12.8390,79.9540', 'Data_Ping
_time': '40:28.0', 'Planned_ETA': '22:22.8', 'Current_Location': 'Unnamed Road, Oragadam Industrial Corridor, Vattambakkam R.F., Tamil Nadu 631605, India', 'DestinationLocation': 'DAIMLER INDIA COMMERCIAL
 VEHICLES,KANCHIPURAM,TAMIL NADU', 'actual_eta': '8/28/20 12:46', 'Curr_lat': 12.83675683, 'Curr_lon': 79.95442767, 'ontime': 'G', 'delay': 'unknown value', 'OriginLocation_Code': 'CHEORADMRCCB1', 'Destin
ationLocation_Code': 'CHEMATDMROPA7', 'trip_start_date': '8/27/20 16:21', 'trip_end_date': 'unknown value', 'TRANSPORTATION_DISTANCE_IN_KM': 103.0, 'vehicleType': 'unknown value', 'Minimum_kms_to_be_cover
ed_in_a_day': nan, 'Driver_Name': 'RAMESH', 'Driver_MobileNo': nan, 'customerID': 'DMREXCHEUX', 'customerNameCode': 'Daimler india commercial vehicles pvt lt', 'supplierID': 'VJLEXSHE09', 'supplierNameCod
e': 'VJ LOGISTICS', 'Material Shipped': 'ZB MODEL PLATE / 3143'}
Produced message: {'GpsProvider': 'CONSENT TRACK', 'BookingID': 'VCV00014382/082021', 'Market/Regular ': 'Regular', 'BookingID_Date': '8/27/20', 'vehicle_no': 'TN22AR2748', 'Origin_Location': 'LUCAS TVS L
TD-PONDY,PONDY,PONDICHERRY', 'Destination_Location': 'LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY', 'Org_lat_lon': '11.8710,79.7390', 'Des_lat_lon': '11.8710,79.7390', 'Data_Ping_time': '05:09.0', 'Planned_ETA'
: '59:24.9', 'Current_Location': '570, National Hwy 48, Shenoy Nagar, Chennai, Tamil Nadu 600030, India', 'DestinationLocation': 'LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY', 'actual_eta': '8/28/20 16:03', 'Cu
rr_lat': 13.0739565, 'Curr_lon': 80.22578, 'ontime': 'G', 'delay': 'unknown value', 'OriginLocation_Code': 'CHEPONLUTCCA4', 'DestinationLocation_Code': 'CHEPONLUTCCA4', 'trip_start_date': '8/27/20 17:57',
 'trip_end_date': 'unknown value', 'TRANSPORTATION_DISTANCE_IN_KM': 300.0, 'vehicleType': 'unknown value', 'Minimum_kms_to_be_covered_in_a_day': nan, 'Driver_Name': 'GIRI', 'Driver_MobileNo': nan, 'custom
erID': 'LUTGCCHE06', 'customerNameCode': 'Lucas tvs ltd', 'supplierID': 'GSTEXLAK1Q', 'supplierNameCode': 'G.S. TRANSPORT', 'Material Shipped': 'LETTERING / FUSO'}
Produced message: {'GpsProvider': 'VAMOSYS', 'BookingID': 'VCV00014743/082021', 'Market/Regular ': 'Regular', 'BookingID_Date': '8/28/20', 'vehicle_no': 'TN28AQ0781', 'Origin_Location': 'DAIMLER INDIA COM
MERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Destination_Location': 'DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Org_lat_lon': '12.8390,79.9540', 'Des_lat_lon': '12.8390,79.9540', 'Data_Ping
_time': '40:31.0', 'Planned_ETA': '48:24.5', 'Current_Location': 'Singaperumal Koil - Sriperumbudur Rd, Oragadam Industrial Corridor, Vattambakkam R.F., Tamil Nadu 631605, India', 'DestinationLocation': '
DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'actual_eta': '8/28/20 12:50', 'Curr_lat': 12.83668556, 'Curr_lon': 79.95056, 'ontime': 'G', 'delay': 'unknown value', 'OriginLocation_Code': 'CH
EORADMRCCB1', 'DestinationLocation_Code': 'CHEMATDMROPA7', 'trip_start_date': '8/28/20 0:47', 'trip_end_date': 'unknown value', 'TRANSPORTATION_DISTANCE_IN_KM': 61.0, 'vehicleType': 'unknown value', 'Mini
mum_kms_to_be_covered_in_a_day': nan, 'Driver_Name': 'RAVI', 'Driver_MobileNo': nan, 'customerID': 'DMREXCHEUX', 'customerNameCode': 'Daimler india commercial vehicles pvt lt', 'supplierID': 'ARVEXNAM09',
 'supplierNameCode': 'ARVINTH TRANSPORT', 'Material Shipped': 'LU STRUT RA / RADIUS ROD'}
Produced message: {'GpsProvider': 'VAMOSYS', 'BookingID': 'VCV00014744/082021', 'Market/Regular ': 'Regular', 'BookingID_Date': '8/28/20', 'vehicle_no': 'TN68F1722', 'Origin_Location': 'LUCAS TVS LTD-POND
Y,PONDY,PONDICHERRY', 'Destination_Location': 'LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY', 'Org_lat_lon': '11.8720,79.6320', 'Des_lat_lon': '11.8720,79.6320', 'Data_Ping_time': '40:29.0', 'Planned_ETA': '23:1
9.2', 'Current_Location': 'Melmaruvathur, Tamil Nadu 603319, India', 'DestinationLocation': 'LUCAS TVS LTD-PONDY,PONDY,PONDICHERRY', 'actual_eta': '8/28/20 14:22', 'Curr_lat': 12.42950111, 'Curr_lon': 79.
83155556, 'ontime': 'G', 'delay': 'unknown value', 'OriginLocation_Code': 'CHENETLUTCCA1', 'DestinationLocation_Code': 'CHENETLUTCCA1', 'trip_start_date': '8/28/20 1:13', 'trip_end_date': 'unknown value',
 'TRANSPORTATION_DISTANCE_IN_KM': 240.0, 'vehicleType': 'unknown value', 'Minimum_kms_to_be_covered_in_a_day': nan, 'Driver_Name': 'TAMIL', 'Driver_MobileNo': nan, 'customerID': 'LUTGCCHE06', 'customerNam
eCode': 'Lucas tvs ltd', 'supplierID': 'SRTEXKOR96', 'supplierNameCode': 'SR TRANSPORTS', 'Material Shipped': 'WISHBONE / V ROD/HDT'}
Produced message: {'GpsProvider': 'VAMOSYS', 'BookingID': 'VCV00014749/082021', 'Market/Regular ': 'Regular', 'BookingID_Date': '8/28/20', 'vehicle_no': 'TN88A4980', 'Origin_Location': 'DAIMLER INDIA COMM
ERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Destination_Location': 'DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'Org_lat_lon': '12.8390,79.9540', 'Des_lat_lon': '12.8390,79.9540', 'Data_Ping_
time': '40:28.0', 'Planned_ETA': '14:22.6', 'Current_Location': 'Ind.park Road, Nayapakkam, Tamil Nadu 602105, India', 'DestinationLocation': 'DAIMLER INDIA COMMERCIAL VEHICLES,KANCHIPURAM,TAMIL NADU', 'a
ctual_eta': '8/28/20 13:25', 'Curr_lat': 13.01736467, 'Curr_lon': 79.962313, 'ontime': 'G', 'delay': 'unknown value', 'OriginLocation_Code': 'CHEORADMRCCB1', 'DestinationLocation_Code': 'CHEMATDMROPA7', '
trip_start_date': '8/28/20 2:13', 'trip_end_date': 'unknown value', 'TRANSPORTATION_DISTANCE_IN_KM': 70.0, 'vehicleType': 'unknown value', 'Minimum_kms_to_be_covered_in_a_day': nan, 'Driver_Name': 'GANESH
', 'Driver_MobileNo': nan, 'customerID': 'DMREXCHEUX', 'customerNameCode': 'Daimler india commercial vehicles pvt lt', 'supplierID': 'ESWEXNAM02', 'supplierNameCode': 'ESWAR TRANSPORT', 'Material Shipped'
: 'MOUNTING BRACKET / FUEL TANK'}
```

messages published to confluent kafka topic:



logistics_data

Production in last hour: 0 messages

Consumption in last hour: 316 messages

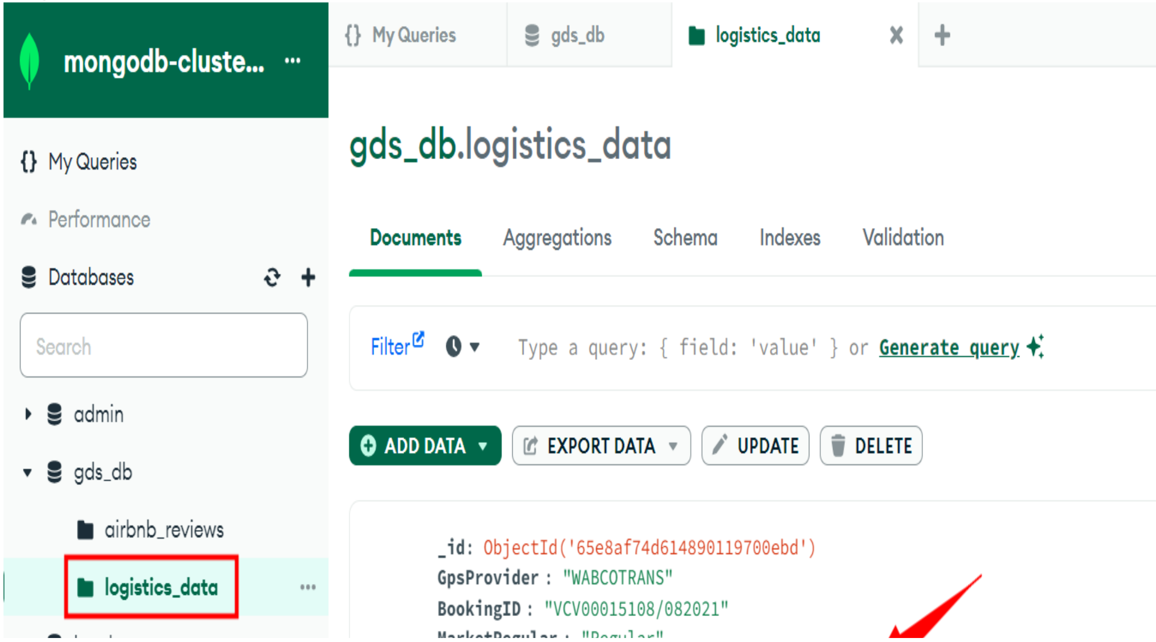
Total messages: 13,760

Retention time: 1 week

50 messages shown

Partition	Key	Value
5	JTECH	{"GpsProvider":"JTECH","BookingID":"WDSBKTP42858","MarketRegular":"Regular","BookingID_Date":"3/28/19","vehicle_n...
5	JTECH	{"GpsProvider":"JTECH","BookingID":"WDSBKTP42685","MarketRegular":"Regular","BookingID_Date":"3/27/19","vehicle_n...
5	JTECH	{"GpsProvider":"JTECH","BookingID":"WDSBKTP43021","MarketRegular":"Regular","BookingID_Date":"3/29/19","vehicle_no...

4. 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.

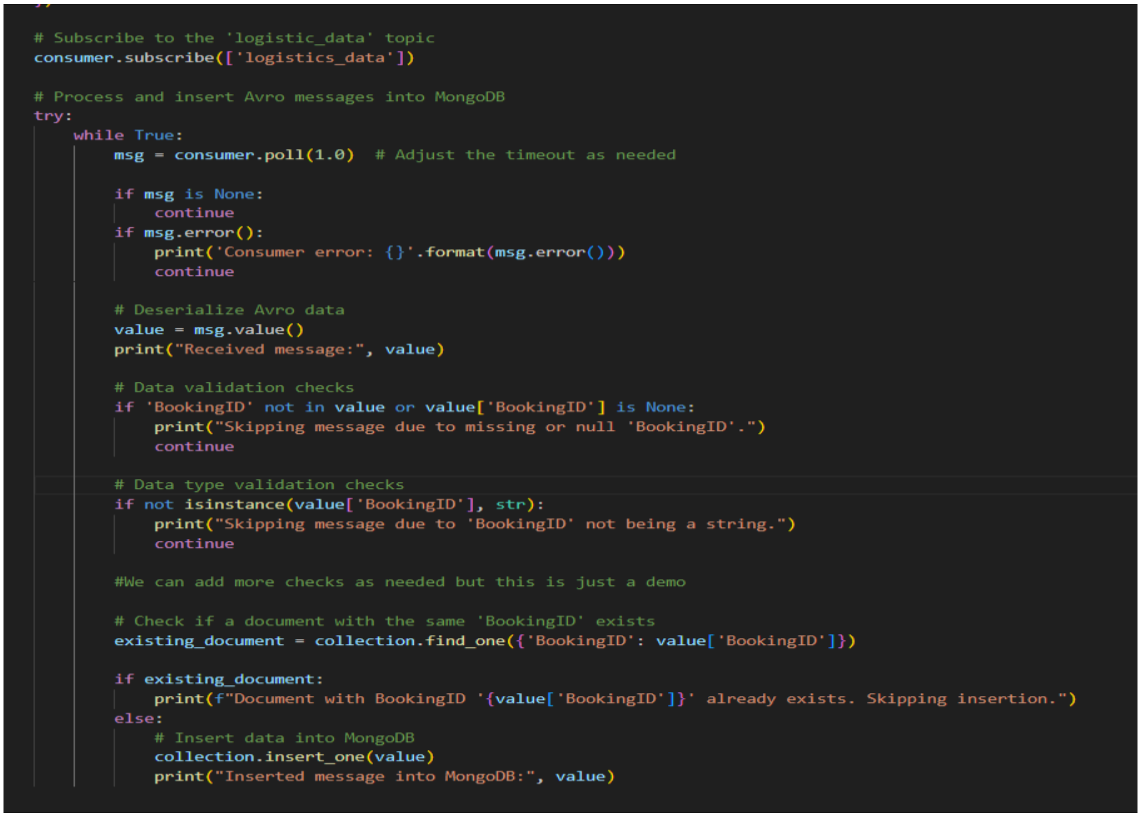


gds_db.logistics_data

```
_id: ObjectId('65e8af74d614890119700ebd')
GpsProvider : "WABCOTRANS"
BookingID : "VCV00015108/082021"
```

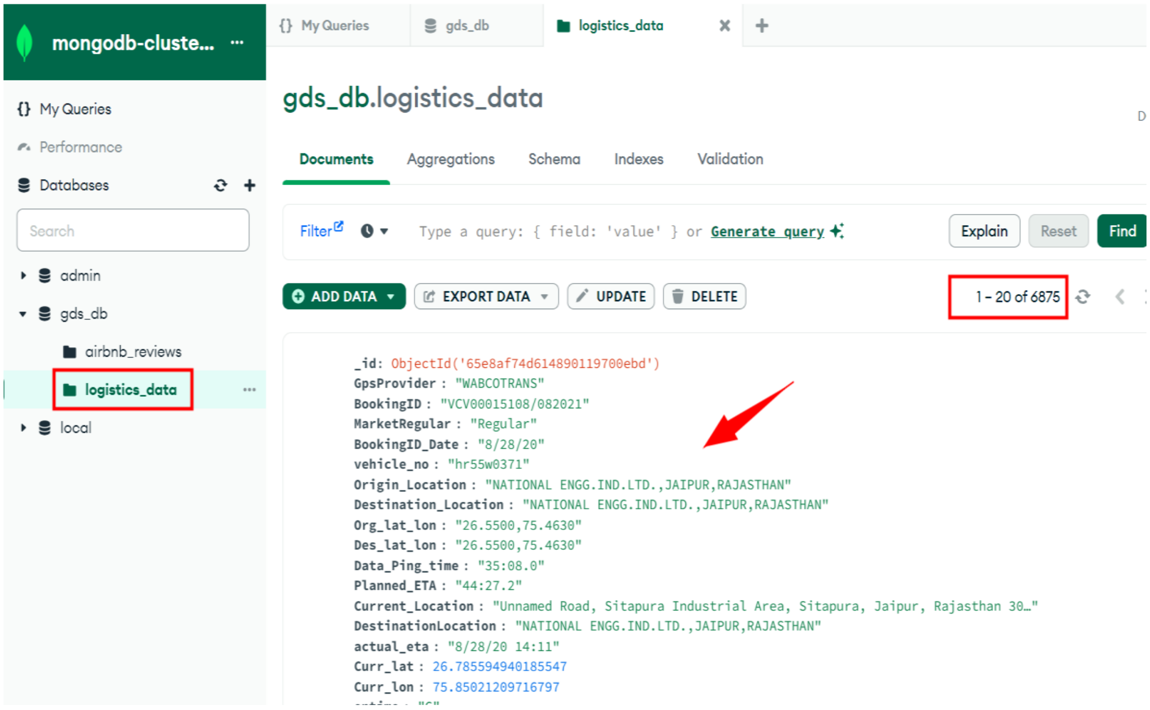
5. then created a consumer script called “logistics_data_consumer.py” that deserializes the avro data back into a python object. I then implemented data validation checks in the code to make sure that it accounts for null values and correct data types checks.

Before pushing the data into the logistics_data collection that was created in the gdb_db mongodb database, I made sure that there are no duplicate records pushed when the consumer runs.



```python
# Subscribe to the 'logistic_data' topic
consumer.subscribe(['logistics_data'])

# Process and insert Avro messages into MongoDB
try:
    while True:
        msg = consumer.poll(1.0)  # Adjust the timeout as needed

        if msg is None:
            continue
        if msg.error():
            print('Consumer error: {}'.format(msg.error()))
            continue

        # Deserialize Avro data
        value = msg.value()
        print("Received message:", value)

        # Data validation checks
        if 'BookingID' not in value or value['BookingID'] is None:
            print("Skipping message due to missing or null 'BookingID'.")
            continue

        # Data type validation checks
        if not isinstance(value['BookingID'], str):
            print("Skipping message due to 'BookingID' not being a string.")
            continue

        #We can add more checks as needed but this is just a demo

        # Check if a document with the same 'BookingID' exists
        existing_document = collection.find_one({'BookingID': value['BookingID']})

        if existing_document:
            print(f"Document with BookingID '{value['BookingID']}' already exists. Skipping insertion.")
        else:
            # Insert data into MongoDB
            collection.insert_one(value)
            print("Inserted message into MongoDB:", value)
```

6. We can also check the data using Mongodb Compass:



```
_id: ObjectId('65e8af74d614890119700ebd')
GpsProvider : "WABCOTRANS"
BookingID : "VCV00015108/082021"
MarketRegular : "Regular"
BookingID_Date : "8/28/20"
vehicle_no : "hr55w0371"
Origin_Location : "NATIONAL ENGG.IND.LTD.,JAIPUR,RAJASTHAN"
Destination_Location : "NATIONAL ENGG.IND.LTD.,JAIPUR,RAJASTHAN"
Org_lat_lon : "26.5500,75.4630"
Des_lat_lon : "26.5500,75.4630"
Data_Ping_time : "35:08.0"
Planned_ETA : "44:27.2"
Current_Location : "Unnamed Road, Sitapura Industrial Area, Sitapura, Jaipur, Rajasthan 30…"
DestinationLocation : "NATIONAL ENGG.IND.LTD.,JAIPUR,RAJASTHAN"
actual_eta : "8/28/20 14:11"
Curr_lat : 26.785594940185547
Curr_lon : 75.85021209716797
```