Kafka Assignment

1. Python script to insert records into MySQL

import datetime

import random

import mysql.connector

# MySQL connection details

mysql\_config = {

'host': 'localhost',

'user': 'your\_username',

'password': 'your\_password',

'database': 'your\_database'

}

def insert\_dummy\_record():

cnx = mysql.connector.connect(\*\*mysql\_config)

cursor = cnx.cursor()

now = datetime.datetime.now()

created\_at = now.strftime('%Y-%m-%d %H:%M:%S')

updated\_at = created\_at

# Generate dummy data

id = random.randint(1, 100)

data = f'Data {id}'

# Prepare the SQL statement

sql = "INSERT INTO my\_table (id, data, created\_at, updated\_at) VALUES (%s, %s, %s, %s)"

values = (id, data, created\_at, updated\_at)

# Execute the SQL statement

cursor.execute(sql, values)

cnx.commit()

cursor.close()

cnx.close()

# Insert dummy records in an infinite loop

while True:

insert\_dummy\_record()

2. Setup Confluent Kafka You'll need to install and set up Confluent Kafka on your system. You can refer to the official Confluent documentation for instructions specific to your operating system.

kafka-topics --create --topic my\_topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

3.create JSON schema on Schema Registry To create a JSON schema on the Schema Registry, you can use the Confluent REST API or a language-specific library like **confluent-kafka-python**

**4.** **from confluent\_kafka import avro**

**from confluent\_kafka.avro import AvroProducer**

**# Schema Registry configuration**

**schema\_registry\_config = {**

**'url': 'http://localhost:8081'**

**}**

**# Load the schema**

**schema\_str = """**

**{**

**"type": "record",**

**"name": "MyRecord",**

**"fields": [**

**{"name": "id", "type": "int"},**

**{"name": "data", "type": "string"},**

**{"name": "created\_at", "type": "string"},**

**{"name": "updated\_at", "type": "string"}**

**]**

**}**

**"""**

**# Create an AvroProducer instance**

**avro\_producer = AvroProducer({**

**'bootstrap.servers': 'localhost:9092',**

**'schema.registry.url': schema\_registry\_config['url']**

**}, default\_value\_schema=avro.loads(schema\_str))**

**# No need to explicitly register the schema, it will be registered automatically when sending messages**

**# Example usage:**

**record = {**

**"id": 1,**

**"data": "Sample data",**

**"created\_at": "2023-06-05 12:00:00",**

**"updated\_at": "2023-06-05 12:00:00"**

**}**

**# Produce the message with the Avro schema**

**avro\_producer.produce(topic='my\_topic', value=record)**

**# Flush the producer to ensure all messages are sent**

**avro\_producer.flush()**

**5.** **Producer code to read data from MySQL and publish to Kafka.**

import datetime

import mysql.connector

from confluent\_kafka import avro

from confluent\_kafka.avro import AvroProducer

# MySQL connection details

mysql\_config = {

'host': 'localhost',

'user': 'your\_username',

'password': 'your\_password',

'database': 'your\_database'

}

# Schema Registry configuration

schema\_registry\_config = {

'url': 'http://localhost:8081'

}

# Load the schema

schema\_str = """

{

"type": "record",

"name": "MyRecord",

"fields": [

{"name": "id", "type": "int"},

{"name": "data", "type": "string"},

{"name": "created\_at", "type": "string"},

{"name": "updated\_at", "type": "string"}

]

}

"""

# Create an AvroProducer instance

avro\_producer = AvroProducer({

'bootstrap.servers': 'localhost:9092',

'schema.registry.url': schema\_registry\_config['url']

}, default\_value\_schema=avro.loads(schema\_str))

def read\_records\_from\_mysql(start\_date):

cnx = mysql.connector.connect(\*\*mysql\_config)

cursor = cnx.cursor()

query = f"SELECT \* FROM my\_table WHERE created\_at >= '{start\_date}'"

cursor.execute(query)

records = cursor.fetchall()

cursor.close()

cnx.close()

return records

def publish\_records\_to\_kafka(records):

for record in records:

avro\_producer.produce(topic='my\_topic', value={

"id": record[0],

"data": record[1],

"created\_at": record[2],

"updated\_at": record[3]

})

avro\_producer.flush()

# Start reading records from MySQL and publishing to Kafka in an infinite loop

start\_date = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

while True:

records = read\_records\_from\_mysql(start\_date)

publish\_records\_to\_kafka(records)

start\_date = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

6. Consumer group to consume data from Kafka topic

from confluent\_kafka import avro

from confluent\_kafka.avro import AvroConsumer

from cassandra.cluster import Cluster

# Cassandra connection details

cassandra\_config = {

'contact\_points': ['localhost'],

'port': 9042,

'keyspace': 'your\_keyspace'

}

# Schema Registry configuration

schema\_registry\_config = {

'url': 'http://localhost:8081'

}

# Load the Avro schema

schema\_str = """

{

"type": "record",

"name": "MyRecord",

"fields": [

{"name": "id", "type": "int"},

{"name": "data", "type": "string"},

{"name": "created\_at", "type": "string"},

{"name": "updated\_at", "type": "string"}

]

}

"""

# Create a Cassandra session

cluster = Cluster([cassandra\_config['contact\_points']], port=cassandra\_config['port'])

session = cluster.connect(cassandra\_config['keyspace'])