

# Large Scale Data Management

## Athens University of Economics and Business

### 2<sup>nd</sup> homework

Theodoros Anagnos, p3352323

April 1, 2024

MSc in Data Science (PT)

## Part I

```
1 import json
2 import asyncio
3 import random
4 import csv
5 from datetime import datetime
6 from aiokafka import AIOKafkaProducer
7 from faker import Faker
8
9 def load_spotify_tracks(file_path):
10     """Load Spotify track names from a CSV file."""
11     with open(file_path, newline='', encoding='utf-8') as csvfile:
12         return [row['name'] for row in csv.DictReader(csvfile)]
13
14 # Initialize the Faker library for generating fake names
15 fake_data_generator = Faker()
16
17 # Specify the Kafka topic for song streaming data
18 kafka_topic = 'song_stream'
19
20 async def stream_song_data():
21     """Asynchronously produce song streaming data to a Kafka topic.
22     """
23     producer = AIOKafkaProducer(
24         bootstrap_servers='localhost:29092',
25         value_serializer=lambda v: json.dumps(v).encode('utf-8')
26     )
27     await producer.start()
28
29     # Generate a list of random names, including a specific name
30     user_names = [fake_data_generator.name() for _ in range(15)] +
31     ["Theodoros Anagnos"]
32
33     while True:
34         for user_name in user_names:
35             # Select a random song from the list
36             song = random.choice(tracks)
37             # Generate a timestamp in the specified format
```

```

36         timestamp = datetime.now().strftime('%Y-%m-%dT%H:%M:%S'
37         )
38         record = {"user_name": user_name, "timestamp":
39         timestamp, "song": song}
40
41         # Send the data to the Kafka topic
42         await producer.send_and_wait(kafka_topic, record)
43         print(f"Sent record: {record}")
44
45         # Wait for 60 seconds before sending the next batch of
46         records
47         await asyncio.sleep(60)
48
49 # Load song data from the CSV file
50 tracks = load_spotify_tracks('spotify-songs.csv')
51
52 # Execute the asynchronous song streaming function
53 # Run the producer
54 loop = asyncio.get_event_loop()
55 loop.run_until_complete(stream_song_data())

```

## Part II

```

1 from pyspark.sql import SparkSession
2 from pyspark.sql.types import StructType, StructField, LongType,
3   IntegerType, FloatType, StringType, TimestampType
4 from pyspark.sql.functions import split, from_json, col
5
6 # Define Kafka message schema
7 kafkaSchema = StructType([
8     StructField("name", StringType(), False),
9     StructField("time", StringType(), False), # Adjusted to
10    StringType to match the produced format
11    StructField("song_name", StringType(), False)
12 ])
13
14 # Define schema for Spotify songs data
15 spotifySchema = StructType([
16     StructField("name", StringType(), False),
17     StructField("artists", StringType(), False),
18     StructField("duration_ms", LongType(), False),
19     StructField("album_name", StringType(), False),
20     StructField("album_release_date", StringType(), False),
21     StructField("danceability", FloatType(), False),
22     StructField("energy", FloatType(), False),
23     StructField("key", IntegerType(), False),
24     StructField("loudness", FloatType(), False),
25     StructField("mode", IntegerType(), False),
26     StructField("speechiness", FloatType(), False),
27     StructField("acousticness", FloatType(), False),
28     StructField("instrumentalness", FloatType(), False),
29     StructField("liveness", FloatType(), False),
30     StructField("valence", FloatType(), False),
31     StructField("tempo", FloatType(), False)
32 ])
33
34 # Initialize SparkSession
35 spark = SparkSession.builder.appName("SSKafka").config("spark.jars.
36   packages", "org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.0").
37   getOrCreate()

```

```

34 spark.sparkContext.setLogLevel("ERROR")
35
36 # Load Kafka stream
37 kafka_df = spark.readStream.format("kafka").option("kafka.bootstrap
.servers", "localhost:29092").option("subscribe", "song_stream"
).option("startingOffsets", "latest").load()
38
39 # Parse JSON message from Kafka
40 parsed_kafka_df = kafka_df.selectExpr("CAST(value AS STRING)").
select(from_json(col("value"), kafkaSchema).alias("data")).
select("data.*")
41
42 # Load Spotify songs data from CSV
43 spotify_df = spark.read.csv("spotify-songs.csv", header=True,
schema=spotifySchema)
44
45 # Join Kafka stream with Spotify songs data
46 joined_df = parsed_kafka_df.join(spotify_df, parsed_kafka_df.
song_name == spotify_df.name, "inner").drop(spotify_df.name)
47
48 # Define function to write to Cassandra
49 def writeToCassandra(write_df, batch_id):
50     write_df.write \
51         .format("org.apache.spark.sql.cassandra") \
52         .options(table="song_records", keyspace="spotify") \
53         .mode("append") \
54         .save()
55
56 # Write joined data to Cassandra
57 query = joined_df.writeStream \
58     .foreachBatch(writeToCassandra) \
59     .outputMode("append") \
60     .start()
61
62 # Await termination
63 query.awaitTermination()

```

```

1 SELECT * FROM song_records WHERE name = 'Theodoros Anagnos' LIMIT
50;
2
3 '''
4 This query filters the records by my username and the specified
time range, then calculates the
5 average danceability value for these records.
6 '''
7 SELECT song_name
8 FROM song_records
9 WHERE name = 'Theodoros Anagnos'
10     AND time >= '2024-04-01 06:00:00'
11     AND time < '2024-04-01 06:20:00';
12
13 '''
14 This query selects the song_name for all records that match my
username and fall within the specified hour.
15 '''
16 SELECT AVG(danceability) AS avg_danceability
17 FROM song_records
18 WHERE name = 'Theodoros Anagnos'
19     AND time >= '2024-04-01 06:00:00'
20     AND time < '2024-04-01 06:20:00';

```

vagrant@vagrant: ~														
Theodore Anagnos   2024-04-01 06:27:42.000000+0000   0.0271   Miss 2024 Stabekrussen (Hjemsnes46)   2023-09-28   Spurtalløen, LilMeys, LilRize, LilPussy, LilKing, LilEllis   0.663   150000   0.063   0.000201														
3   0.071   0.271   0   Miss 2024 Stabekrussen (Hjemsnes46)   5.220   150.2   0.012														
(16 rows)														
cqlsh:spotify> SELECT * FROM song_records LIMIT 50;														
name	time	acousticness	album_name	album_release_date	artists	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	song_name
Shannon French	2020-04-01 06:09:00.000000+0000	0.147	Fr & Love II	2022-11-03	Monolisa	0.795	1389729	0.252	1.9e-06	A	0.0000	-18.308	0	
Shannon French	2020-04-01 06:10:00.000000+0000	0.202	Feeling Myself (Roc Boyz Remix)	2022-08-10	2J, Roc Boyz	0.661	127230	0.556	0	A	0.0000	-7.423	0	
Shannon French	2020-04-01 06:11:00.000000+0000	0.231	HUPPER BOUNCI	2022-11-17	Paper Kawaii	0.601	92000	0.597	0.0117	A	0.0000	-11.501	0	
Shannon French	2020-04-01 06:12:00.000000+0000	0.263	We Should Be Talking	2021-11-01	Jessica Bari	0.627	176101	0.185	1.9e-05	A	0.0000	-8.301	0	
Shannon French	2020-04-01 06:13:00.000000+0000	0.207	Sm ve ELITE	2022-11-03	Reel	0.567	100000	0.398	0	A	0.0000	-26.505	1	
Shannon French	2020-04-01 06:14:00.000000+0000	0.408	M' Manc (con Gaelier & Sfera Ebbasta)	2020-04-11	Shablo, Gaelier, Sfera Ebbasta	0.763	180000	0.706	0	A	0.0000	-6.117	0	
Shannon French	2020-04-01 06:15:00.000000+0000	0.116	rd unaid' souan (JUST TO LET ME KNOW) - Single	2022-11-03	rd unaid' souan	0.625				A	0.0000			
2023-11-27   0.000			Paper Planes											
Shannon French	2020-04-01 06:16:00.000000+0000	0.812	EL AFTER DEL AFTER	2023-11-11	YSY A, Duki, ONKIRA, Yessan 曹山	0.406	188800	0.929	0	A	0.0000	-2.530	0	
Shannon French	2020-04-01 06:17:00.000000+0000	0.326		2022-11-26	Elizabet Taylor	0.509	149701	0.407	0	A	0.0000	-4.227	0	
Shannon French	2020-04-01 06:18:00.000000+0000	0.6653		2023-11-03	Eliza Wilson	0.407	180310	0.925	0	A	0.0000	-16.307	1	
Shannon French	2020-04-01 06:19:00.000000+0000	0.176	Omniplanet	2022-10-27	Cine	0.400	120000	0.403	0.0712	A	0.0000	-11.300	0	
Shannon French	2020-04-01 06:20:00.000000+0000	0.232	DELINQUENTE	2021-08-27	Baby Gang, Boko, Noad	0.7	279230	0.402	1.9e-06	A	0.0000	-11.870	1	
Shannon French	2020-04-01 06:21:00.000000+0000	0.205	Erro da Minha Vida (ao Vivo)	2022-12-00	Solange Almeida, Paul Fernandes	0.400	171000	0.400	0	A	0.0000			
2023-11-27   0.000			Erro da Minha Vida - Ao Vivo	0.203	163.010	0.400								
Shannon French	2020-04-01 06:22:00.000000+0000	0.702	P2 - THE BIG HEARTED BAG GUY	2022-10-10	A-Reece	0.706	12000	0.407	0.012	A	0.0000	-10.717	0	
Shannon French	2020-04-01 06:23:00.000000+0000	0.604	Never Lose Me	2022-11-10	Flo Milli	0.701	12000	0.407	0	A	0.0000	-7.807	1	
Shannon French	2020-04-01 06:24:00.000000+0000	0.581	Amor Desolado	2022-10-00	Bolero, many more, G.O.E	0.52	11000	0.408	0	A	0.0000	-6.400	1	
David Miller	2020-04-01 06:25:00.000000+0000	0.388	FW	2024-01-10	Cardenas del Cielo	0.528	201770	0.405	1.9e-06	A	0.0000	-9.22	0	
David Miller	2020-04-01 06:26:00.000000+0000	0.527	Sus Mejores Gafas	2004-01-01	La Fève	0.722	161117	0.407	2.0e-06	A	0.0000	-10.621	0	
David Miller	2020-04-01 06:27:00.000000+0000	0.178	2H	2022-12-22	La Fève	0.406	120000	0.40	0	A	0.0000	-10.400	1	
David Miller	2020-04-01 06:28:00.000000+0000	0.220	Aph Olan	2022-06-00	Cabi	0.706	161100	0.400	0	A	0.0000	-8.220	0	
David Miller	2020-04-01 06:29:00.000000+0000	0.151	nadie sabe lo que va a pasar mañana	2022-10-15	Bad Bunny	0.651	211700	0.700	0.010	A	0.0000	-4.02	0	
David Miller	2020-04-01 06:30:00.000000+0000	0.621	10 Exitos de Los Hermanos Flores	1990-01-01	Los Hermanos Flores	0.61	18000	0.566	0.0001	A	0.0000	-4.277	0	
David Miller	2020-04-01 06:31:00.000000+0000	0.435	Divane	1990-01-01	Vejar	0.721	218800	0.942	0	A	0.0000	-7.101	0	
David Miller	2020-04-01 06:32:00.000000+0000	0.460	CHUMES	2022-06-20	Paseo Plaza, Tito Double P	0.726	150371	0.402	0	A	0.0000	-4.100	0	
David Miller	2020-04-01 06:33:00.000000+0000	0.0882	GRX	2022-12-14	Lola Indigo, La Zoni	0.712	140020	0.42	0.070	A	0.0000	-4.507	1	
David Miller	2020-04-01 06:34:00.000000+0000	0.736	SALE EDOGE	2022-11-00	Lacini	0.536	201000	0.406	0	A	0.0000			
2023-11-27   0.000			CODE BARRE	0.111	79.101	0.51								
David Miller	2020-04-01 06:35:00.000000+0000	0.171	CODE HEART BEAT	2022-11-20	YOSAKOI	0.727	130000	0.717	1.1e-06	A	0.0000	-8.800	0	
David Miller	2020-04-01 06:36:00.000000+0000	0.0388	TUNNEL	2024-01-05	Sinea La Rue, PT Kings	0.706	140000	0.518	0	A	0.0000	-7.071	0	
David Miller	2020-04-01 06:37:00.000000+0000	0.460	SAFE	2022-10-20	Gibbo, Sepabo, Sotamar, Whony	0.601	200700	0.407	0	A	0.0000	-4.170	1	
David Miller	2020-04-01 06:38:00.000000+0000	0.0361	P2 - THE BIG HEARTED BAG GUY	2022-10-20	A-Reece, Jay Jody	0.407	188370	0.514	1.0e-05	A	0.0000	-7.011	1	
David Miller	2020-04-01 06:39:00.000000+0000	0.288	Para Bailar!	2020-01-12	Rich	0.706	130000	0.401	0.000700	A	0.0000	-8.707	0	
Henry Espaza	2020-04-01 06:40:00.000000+0000	0.292	Myraam	1990-01-21	The Latin Brothers, Jessica Martinez	0.710	200000	0.439	0	A	0.0000	-2.17	0	
Henry Espaza	2020-04-01 06:41:00.000000+0000	0.296	Desi Malabar	2015-06-00	Yo Yo Honey Singh	0.600	100000	0.505	0.00070	A	0.0000	-16.200	1	
Henry Espaza	2020-04-01 06:42:00.000000+0000	0.256	5 u n e r t i n e	2022-10-20	Projector Band	0.603	101700	0.516	7.10e-06	A	0.0000	-11.200	1	
Henry Espaza	2020-04-01 06:43:00.000000+0000	0.199	Fluid	2022-10-00	RoT, Wubbel	0.703	142200	0.578	1.12e-06	A	0.0000	-8.176	1	
Henry Espaza	2020-04-01 06:44:00.000000+0000	0.0011	3/3 Mob, 2	2022-11-31	V.M.O, Ediz, Tuf	0.526	102000	0.408	0	A	0.0000	-11.200	0	
Henry Espaza	2020-04-01 06:45:00.000000+0000	0.0115	Confart Zone	2022-12-20	Guy Sheetrix, Ofir Nalul	0.700	170000	0.600	0.00070	A	0.0000	-5.000	1	
Henry Espaza	2020-04-01 06:46:00.000000+0000	0.221	Kallian & Toplim	2022-11-17	Shony, AlPhe	0.707	111000	0.405	0.00012	A	0.0000	-7.000	1	
Henry Espaza	2020-04-01 06:47:00.000000+0000	0.422	Que e o Centor	2022-01-01	Efe Dezays, Ipirica	0.700	40025	0.702	0.00015	A	0.0000	-8.000	1	
Henry Espaza	2020-04-01 06:48:00.000000+0000	0.181	The Last Son	2000-01-01	Don Omar	0.600	100000	0.702	0.00012	A	0.0000	-8.000	1	
Henry Espaza	2020-04-01 06:49:00.000000+0000	0.310	Stop Giving Me Notice	2022-09-21	Morcasol, Zoi	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:50:00.000000+0000	0.504	Stop Giving Me Notice	2022-12-08	Lyrical Lemonade, Jack Harlow, Sawe	0.610	100000	0.707	2.7e-06	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:51:00.000000+0000	0.205	Mis Treinta Mejores Canciones (Parte 1)	2022-01-01	La Sonora De Tommy Ray	0.600	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:52:00.000000+0000	0.0016	After Hours	2022-01-01	The Weeknd	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:53:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:54:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:55:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:56:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:57:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:58:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Henry Espaza	2020-04-01 06:59:00.000000+0000	0.0016	M'anc (con Gaelier & Sfera Ebbasta)	2022-01-01	Shablo, Gaelier, Sfera Ebbasta	0.610	100000	0.707	0	A	0.0000	-8.000	0	
Theodore Anagnos	2020-04-01 06:00:00.000000+0000	0.111	Chet Cha Cha	2022-12-01	MADE, SHABD, Raia	0.700	101010	0.52	2.0e-06	A	0.0000	-9.100	0	
Theodore Anagnos	2020-04-01 06:01:00.000000+0000	0.406	A la Carte	2022-12-01	Immune, Dani Guelino, Rick	0.602	200000	0.713	0	A	0.0000	-4.00	1	
(16 rows)														
cqlsh:spotify>														
cqlsh:spotify>														
cqlsh:spotify>														

Figure 1: A sample of persisted lines (50) of the Cassandra table.

```
vagrant@vagrant: ~
cqlsh:spotify>
cqlsh:spotify> SELECT song_name
... FROM song_records
... WHERE name = 'Theodoros Anagnos'
... AND time >= '2024-04-01 06:00:00'
... AND time < '2024-04-01 06:20:00';

song_name
-----
À la Carte
Porselani
PACATE
BLIND SPOT
BAIXO
Jaga Jaga
Brilla
51 - Freestyle
Piękny Świat
On Ira
Mghayer

(11 rows)
cqlsh:spotify>
cqlsh:spotify>
cqlsh:spotify> SELECT AVG(danceability) AS avg_danceability
... FROM song_records
... WHERE name = 'Theodoros Anagnos'
... AND time >= '2024-04-01 06:00:00'
... AND time < '2024-04-01 06:20:00';

avg_danceability
-----
0.678545

(1 rows)
cqlsh:spotify>
cqlsh:spotify>
cqlsh:spotify> |
```

Figure 2: CQL queries with otutputs.

## 1 Introduction

This report outlines the Cassandra data model designed for storing streaming records processed by a Spark application. The Spark application integrates live data streams from Kafka, containing user listening events, with static Spotify song data. The processed records are then stored in Cassandra for real-time analytics and query purposes.

## 2 Data Sources

### 2.1 Kafka Streams

The Kafka stream provides real-time user listening events with the following schema:

- **name**: The name of the user (StringType)
- **time**: The timestamp of the event (StringType)
- **song\_name**: The name of the song being played (StringType)

### 2.2 Spotify Song Data

The static Spotify song dataset includes detailed metadata about each song, with the following schema:

- **name**: Song name (StringType)
- **artists**: Artists performing the song (StringType)
- **duration\_ms**: Duration of the song in milliseconds (LongType)
- **album\_name**: Name of the album (StringType)
- **album\_release\_date**: Release date of the album (StringType)
- **danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo**: Various musical features of the song (FloatType or IntegerType)

## 3 Cassandra Data Model

The Cassandra table *song\_records* in the keyspace *spotify* is designed to store the joined records from Kafka streams and Spotify song data. The table schema is designed as follows:

```
CREATE TABLE spotify.song_records (  
    name TEXT,  
    time TIMESTAMP,  
    song_name TEXT,  
    artist_names TEXT,  
    song_duration_ms BIGINT,  
    album_name TEXT,  
    album_release_date TEXT,  
    danceability FLOAT,  
    energy FLOAT,  
    ...  
    PRIMARY KEY ((name), time)  
);
```

**Primary Key Design:** The combination of *name* and *time* is used as the primary key to uniquely identify each listening event. This choice supports queries that retrieve all listening events for a user in a specific time range, optimising the table for user-centric analytics.

## 4 Conclusion

The Cassandra data model is optimised for efficient storage and retrieval of user listening events, combined with rich song metadata from Spotify. This model supports a wide range of queries necessary for real-time music streaming analytics, such as user listening habits, song popularity trends, and more.