Large Scale Data Management
MSc Data Science (Part-Time)
Programming Project 2
(Kafka - Spark - Cassandra)

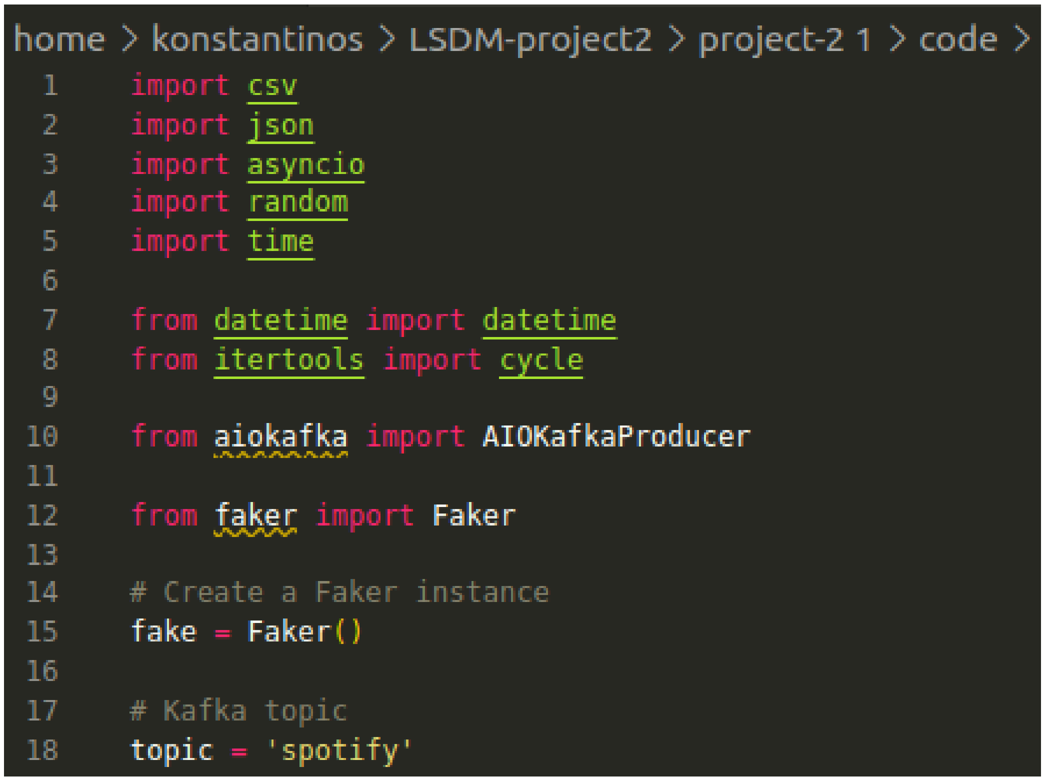
Konstantinos Kostis (p3352311)

Part I

Below you can find images depicting the developed script for the first part and a screenshot of the messages produced into kafka. Monitoring the messages produced into kafka was done via the excellent library provided by Magnus Edenhill, namely kafkacat.

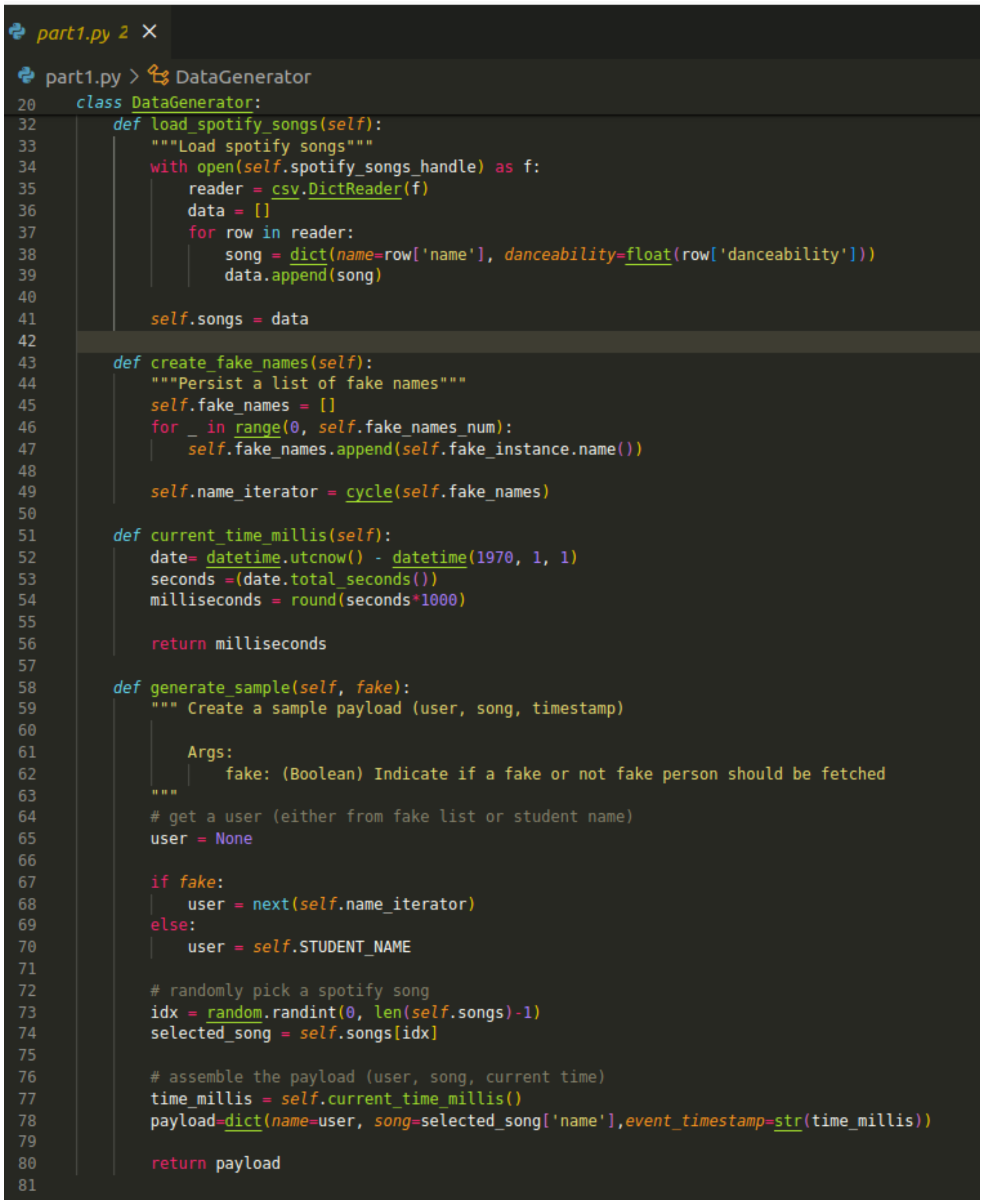
You can read more about kafkacat at `https://github.com/edenhill/kcat`

Kakfa producer script

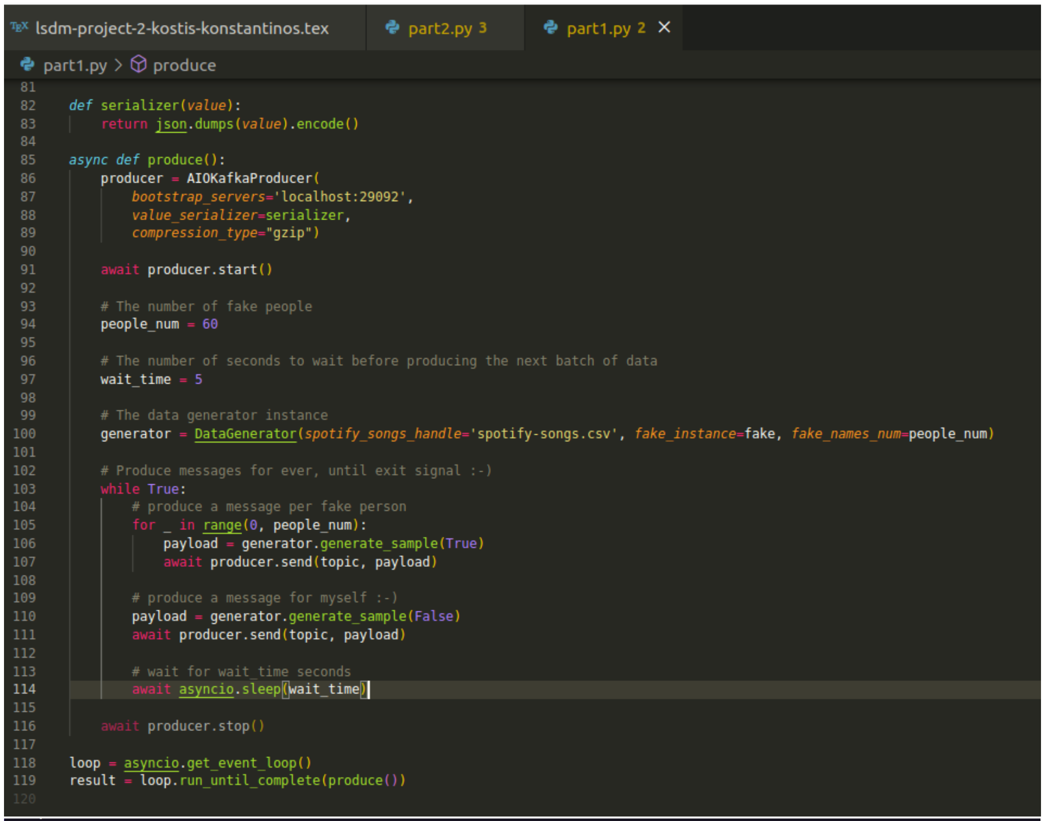


home > konstantinos > LSDM-project2 > project-2 1 > code >

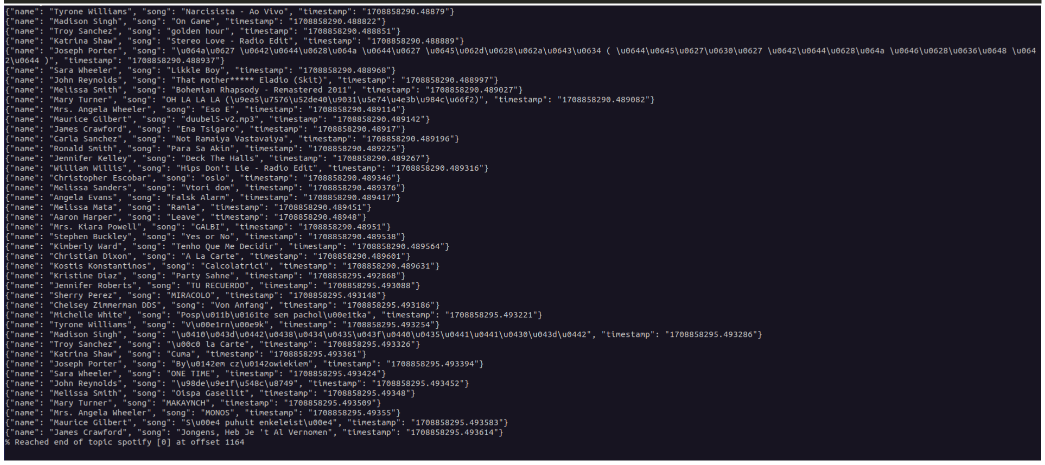
```python
import csv
import json
import asyncio
import random
import time

from datetime import datetime
from itertools import cycle

from aiokafka import AIOKafkaProducer

from faker import Faker

# Create a Faker instance
fake = Faker()

# Kafka topic
topic = 'spotify'
```



```python
class DataGenerator:
    def load_spotify_songs(self):
        """Load spotify songs"""
        with open(self.spotify_songs_handle) as f:
            reader = csv.DictReader(f)
            data = []
            for row in reader:
                song = dict(name=row['name'], danceability=float(row['danceability']))
                data.append(song)

        self.songs = data

    def create_fake_names(self):
        """Persist a list of fake names"""
        self.fake_names = []
        for _ in range(0, self.fake_names_num):
            self.fake_names.append(self.fake_instance.name())

        self.name_iterator = cycle(self.fake_names)

    def current_time_millis(self):
        date= datetime.utcnow() - datetime(1970, 1, 1)
        seconds =(date.total_seconds())
        milliseconds = round(seconds*1000)

        return milliseconds

    def generate_sample(self, fake):
        """ Create a sample payload (user, song, timestamp)

            Args:
                fake: (Boolean) Indicate if a fake or not fake person should be fetched
        """
        # get a user (either from fake list or student name)
        user = None

        if fake:
            user = next(self.name_iterator)
        else:
            user = self.STUDENT_NAME

        # randomly pick a spotify song
        idx = random.randint(0, len(self.songs)-1)
        selected_song = self.songs[idx]

        # assemble the payload (user, song, current time)
        time_millis = self.current_time_millis()
        payload=dict(name=user, song=selected_song['name'],event_timestamp=str(time_millis))

        return payload

```



```python

def serializer(value):
    return json.dumps(value).encode()

async def produce():
    producer = AIOKafkaProducer(
        bootstrap_servers='localhost:29092',
        value_serializer=serializer,
        compression_type="gzip")

    await producer.start()

    # The number of fake people
    people_num = 60

    # The number of seconds to wait before producing the next batch of data
    wait_time = 5

    # The data generator instance
    generator = DataGenerator(spotify_songs_handle='spotify-songs.csv', fake_instance=fake, fake_names_num=people_num)

    # Produce messages for ever, until exit signal :-)
    while True:
        # produce a message per fake person
        for _ in range(0, people_num):
            payload = generator.generate_sample(True)
            await producer.send(topic, payload)

        # produce a message for myself :-)
        payload = generator.generate_sample(False)
        await producer.send(topic, payload)

        # wait for wait_time seconds
        await asyncio.sleep(wait_time)

    await producer.stop()

loop = asyncio.get_event_loop()
result = loop.run_until_complete(produce())

```

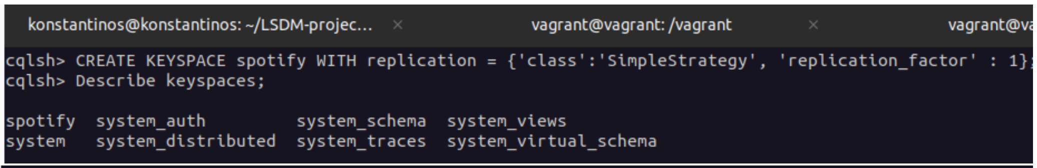


```
{"name": "Tyrone Williams", "song": "Narcisista - Ao Vivo", "timestamp": "1708858290.48879"}
{"name": "Madison Singh", "song": "On Game", "timestamp": "1708858290.488822"}
{"name": "Troy Sanchez", "song": "golden hour", "timestamp": "1708858290.488851"}
{"name": "Katrina Shaw", "song": "Stereo Love - Radio Edit", "timestamp": "1708858290.488889"}
{"name": "Joseph Porter", "song": "\u064a\u0627 \u0642\u0644\u0628\u064a \u0644\u0627 \u0645\u062d\u0628\u062a\u0643\u0634 ( \u0644\u0645\u0627\u0630\u0627 \u0642\u0644\u0628\u064a \u0646\u0628\u0636\u0648 \u064
2\u0644 )", "timestamp": "1708858290.488937"}
{"name": "Sara Wheeler", "song": "Likkle Boy", "timestamp": "1708858290.488968"}
{"name": "John Reynolds", "song": "That mother***** Eladio (Skit)", "timestamp": "1708858290.488997"}
{"name": "Melissa Smith", "song": "Bohemian Rhapsody - Remastered 2011", "timestamp": "1708858290.489027"}
{"name": "Mary Turner", "song": "OH LA LA LA (\u9ea5\u7576\u52de40\u9031\u5e74\u4e3b\u984c\u66f2)", "timestamp": "1708858290.489082"}
{"name": "Mrs. Angela Wheeler", "song": "Eso E", "timestamp": "1708858290.489114"}
{"name": "Maurice Gilbert", "song": "duubel5-v2.mp3", "timestamp": "1708858290.489142"}
{"name": "James Crawford", "song": "Ena Tsigaro", "timestamp": "1708858290.48917"}
{"name": "Carla Sanchez", "song": "Not Ramaiya Vastavaiya", "timestamp": "1708858290.489196"}
{"name": "Ronald Smith", "song": "Para Sa Akin", "timestamp": "1708858290.489225"}
{"name": "Jennifer Kelley", "song": "Deck The Halls", "timestamp": "1708858290.489267"}
{"name": "William Willis", "song": "Hips Don't Lie - Radio Edit", "timestamp": "1708858290.489316"}
{"name": "Christopher Escobar", "song": "oslo", "timestamp": "1708858290.489346"}
{"name": "Melissa Sanders", "song": "Vtori dom", "timestamp": "1708858290.489376"}
{"name": "Angela Evans", "song": "Falsk Alarm", "timestamp": "1708858290.489417"}
{"name": "Melissa Mata", "song": "Ramla", "timestamp": "1708858290.489451"}
{"name": "Aaron Harper", "song": "Leave", "timestamp": "1708858290.48948"}
{"name": "Mrs. Kiara Powell", "song": "GALBI", "timestamp": "1708858290.48951"}
{"name": "Stephen Buckley", "song": "Yes or No", "timestamp": "1708858290.489538"}
{"name": "Kimberly Ward", "song": "Tenho Que Me Decidir", "timestamp": "1708858290.489564"}
{"name": "Christian Dixon", "song": "A La Carte", "timestamp": "1708858290.489601"}
{"name": "Kostis Konstantinos", "song": "Calcolatrici", "timestamp": "1708858290.489631"}
{"name": "Kristine Diaz", "song": "Party Sahne", "timestamp": "1708858295.492868"}
{"name": "Jennifer Roberts", "song": "TU RECUERDO", "timestamp": "1708858295.493088"}
{"name": "Sherry Perez", "song": "MIRACOLO", "timestamp": "1708858295.493148"}
{"name": "Chelsey Zimmerman DDS", "song": "Von Anfang", "timestamp": "1708858295.493186"}
{"name": "Michelle White", "song": "Posp\u011b\u0161te sem pachol\u00e1tka", "timestamp": "1708858295.493221"}
{"name": "Tyrone Williams", "song": "V\u00e1rn\u00e9k", "timestamp": "1708858295.493254"}
{"name": "Madison Singh", "song": "\u0410\u043d\u0442\u0438\u0434\u0435\u043f\u0440\u0435\u0441\u0441\u0430\u043d\u0442", "timestamp": "1708858295.493286"}
{"name": "Troy Sanchez", "song": "\u00c0 la Carte", "timestamp": "1708858295.493326"}
{"name": "Katrina Shaw", "song": "Cuma", "timestamp": "1708858295.493361"}
{"name": "Joseph Porter", "song": "By\u0142em cz\u0142owiekiem", "timestamp": "1708858295.493394"}
{"name": "Sara Wheeler", "song": "ONE TIME", "timestamp": "1708858295.493424"}
{"name": "John Reynolds", "song": "\u98de\u9e1f\u548c\u8749", "timestamp": "1708858295.493452"}
{"name": "Melissa Smith", "song": "Oispa Gasellit", "timestamp": "1708858295.49348"}
{"name": "Mary Turner", "song": "MAKAYNCH", "timestamp": "1708858295.493509"}
{"name": "Mrs. Angela Wheeler", "song": "MONOS", "timestamp": "1708858295.49355"}
{"name": "Maurice Gilbert", "song": "S\u00e4 puhuit enkeleist\u00e4", "timestamp": "1708858295.493583"}
{"name": "James Crawford", "song": "Jongens, Heb Je 't Al Vernomen", "timestamp": "1708858295.493614"}
% Reached end of topic spotify [0] at offset 1164
```

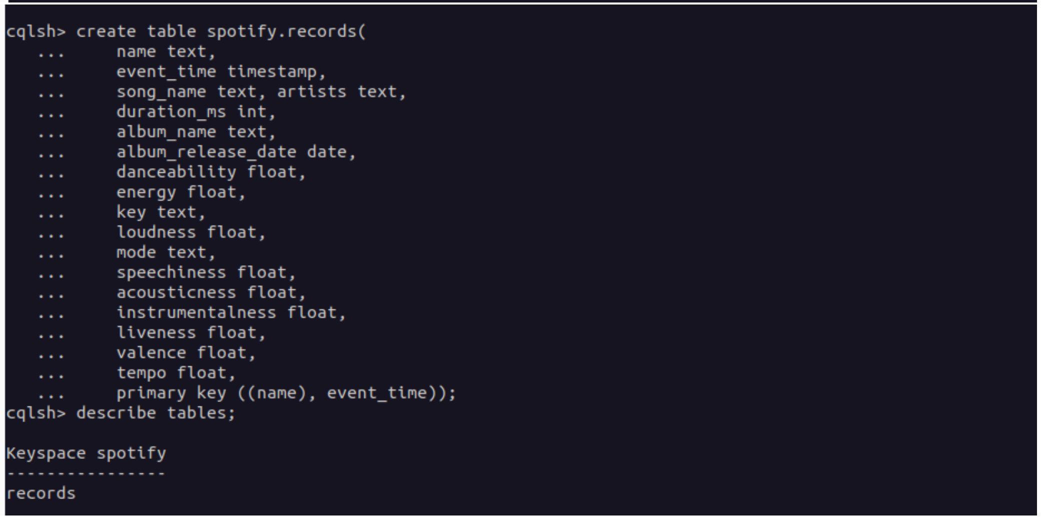
The source code of the script for the first part can be found under ”code/part1.py”

Part II

Lets start with cassandra's data modeling. The following screenshots depict the commands used in order to create the data model.


konstantinos@konstantinos: ~/LSDM-projec...
vagrant@vagrant: /vagrant
vagrant@va

```
cqlsh> CREATE KEYSPACE spotify WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 1};
cqlsh> Describe keyspaces;

spotify  system_auth         system_schema  system_views
system   system_distributed  system_traces  system_virtual_schema
```



```
cqlsh> create table spotify.records(
   ...     name text,
   ...     event_time timestamp,
   ...     song_name text, artists text,
   ...     duration_ms int,
   ...     album_name text,
   ...     album_release_date date,
   ...     danceability float,
   ...     energy float,
   ...     key text,
   ...     loudness float,
   ...     mode text,
   ...     speechiness float,
   ...     acousticness float,
   ...     instrumentalness float,
   ...     liveness float,
   ...     valence float,
   ...     tempo float,
   ...     primary key ((name), event_time));
cqlsh> describe tables;

Keyspace spotify
---------------
records
```

You can also find these commands in the file "code/scripts-commands-and-cassandra-schema.txt"

Briefly, the spotify.records table includes the name of the user (fake, and myself), the time of the event, and all the metadata of the song the user listened to.

The most important part of this model is its primary key. In order to be able to perform aggregations for a particular person and hour the following key is employed: **primary key ((name), event_time)**

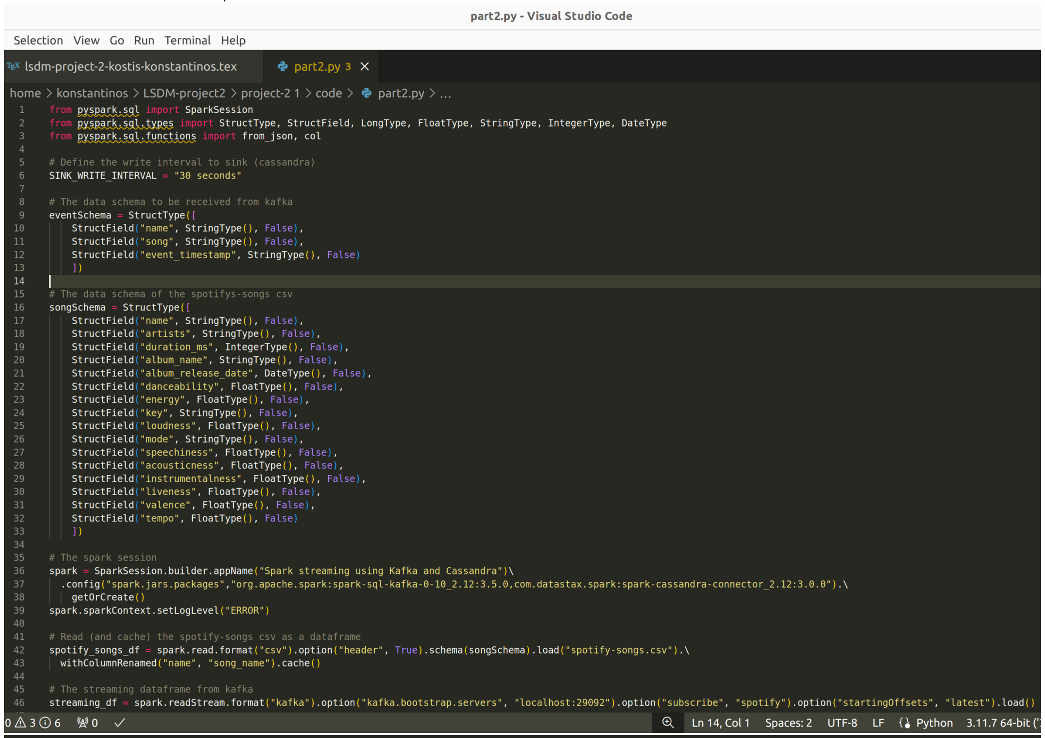
The name of the user is the partition key, and the rest of the primary key which is the event_time forms the clustering key. With this primary key in place, performing queries on a specific person in a given range of time is done in an optimal way.

With this setup we achieve fast access on the person (fast access on the partition) along with uniqueness on the pair (name, event_time). In general the partition key (name) is responsible for data distribution accross nodes and the clustering key (event_time) is responsible for data sorting within the partition.

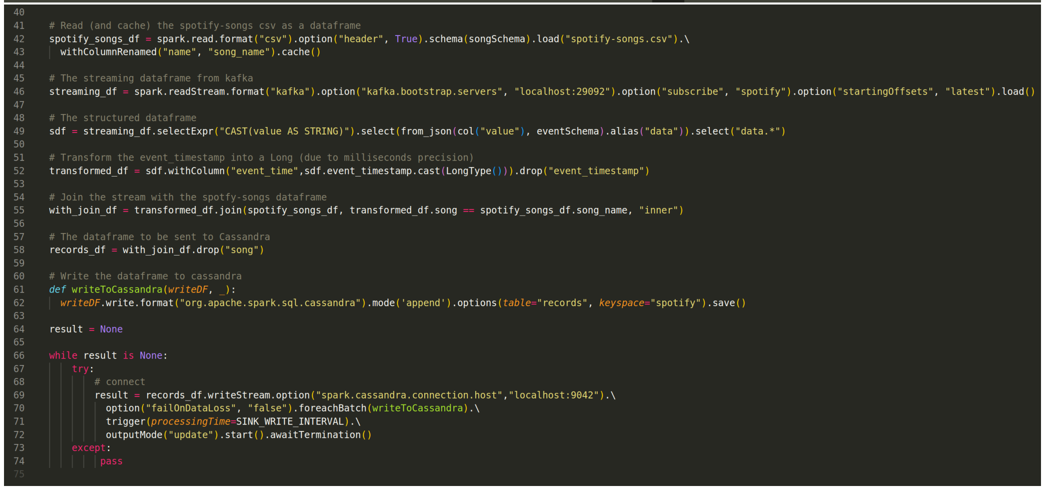
Needless to say that event_time is configured to be of type timestamp in order to provide millsecond query precision if needed.

Spark job script

The script for the second part can be found in the images below, and of course at the "code/part2.py".


part2.py - Visual Studio Code
Selection View Go Run Terminal Help
lsdm-project-2-kostis-konstantinos.tex
part2.py 3
home > konstantinos > LSDM-project2 > project-2 1 > code > part2.py > ...
0 ⚠ 3 ⓘ 6 0 ✓
Ln 14, Col 1 Spaces: 2 UTF-8 LF Python 3.11.7 64-bit

```python
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, LongType, FloatType, StringType, IntegerType, DateType
from pyspark.sql.functions import from_json, col

# Define the write interval to sink (cassandra)
SINK_WRITE_INTERVAL = "30 seconds"

# The data schema to be received from kafka
eventSchema = StructType([
    StructField("name", StringType(), False),
    StructField("song", StringType(), False),
    StructField("event_timestamp", StringType(), False)
    ])

# The data schema of the spotifys-songs csv
songSchema = StructType([
    StructField("name", StringType(), False),
    StructField("artists", StringType(), False),
    StructField("duration_ms", IntegerType(), False),
    StructField("album_name", StringType(), False),
    StructField("album_release_date", DateType(), False),
    StructField("danceability", FloatType(), False),
    StructField("energy", FloatType(), False),
    StructField("key", StringType(), False),
    StructField("loudness", FloatType(), False),
    StructField("mode", StringType(), False),
    StructField("speechiness", FloatType(), False),
    StructField("acousticness", FloatType(), False),
    StructField("instrumentalness", FloatType(), False),
    StructField("liveness", FloatType(), False),
    StructField("valence", FloatType(), False),
    StructField("tempo", FloatType(), False)
    ])

# The spark session
spark = SparkSession.builder.appName("Spark streaming using Kafka and Cassandra")\
  .config("spark.jars.packages","org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.0,com.datastax.spark:spark-cassandra-connector_2.12:3.0.0").\
    getOrCreate()
spark.sparkContext.setLogLevel("ERROR")

# Read (and cache) the spotify-songs csv as a dataframe
spotify_songs_df = spark.read.format("csv").option("header", True).schema(songSchema).load("spotify-songs.csv").\
  withColumnRenamed("name", "song_name").cache()

# The streaming dataframe from kafka
streaming_df = spark.readStream.format("kafka").option("kafka.bootstrap.servers", "localhost:29092").option("subscribe", "spotify").option("startingOffsets", "latest").load()
```

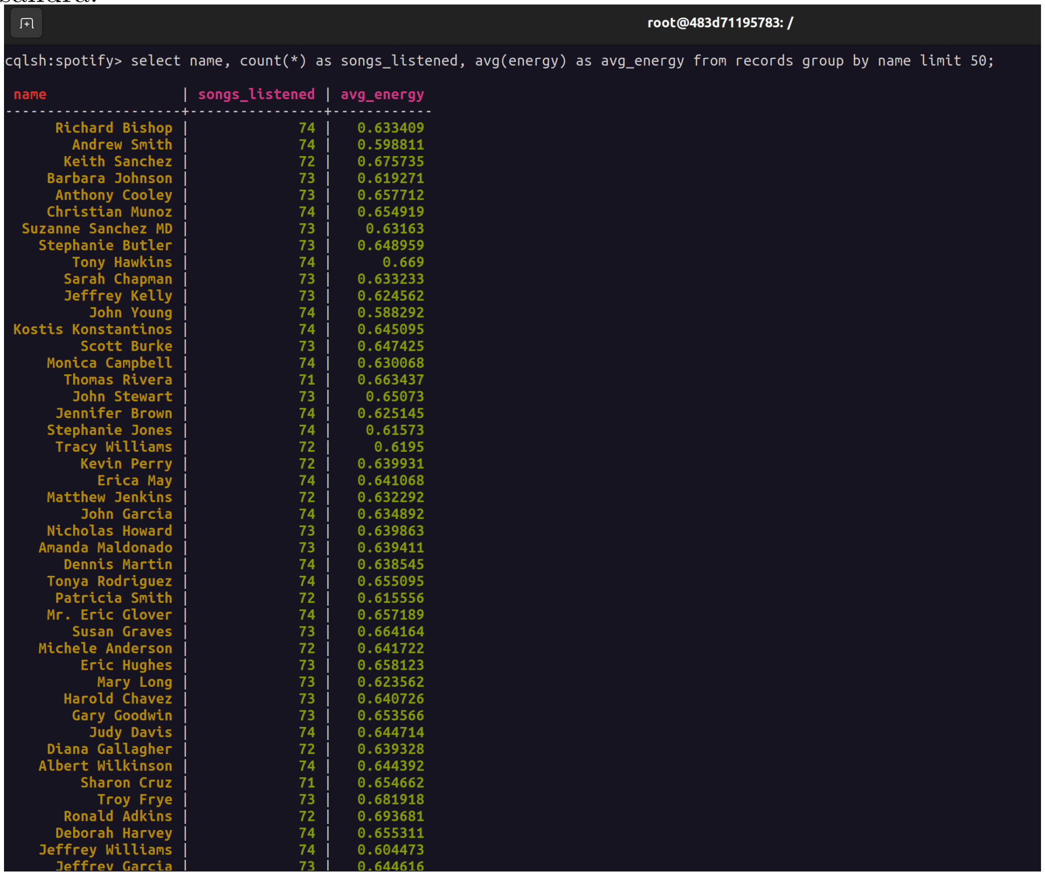


```python

# Read (and cache) the spotify-songs csv as a dataframe
spotify_songs_df = spark.read.format("csv").option("header", True).schema(songSchema).load("spotify-songs.csv").\
  withColumnRenamed("name", "song_name").cache()

# The streaming dataframe from kafka
streaming_df = spark.readStream.format("kafka").option("kafka.bootstrap.servers", "localhost:29092").option("subscribe", "spotify").option("startingOffsets", "latest").load()
```

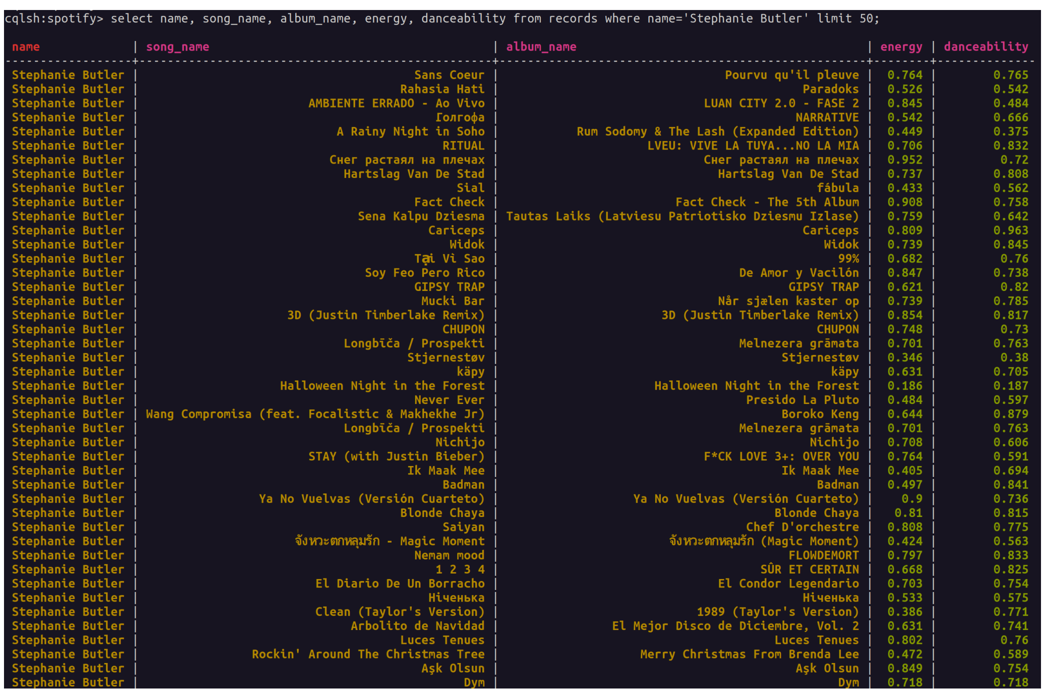
```python

# The structured dataframe
sdf = streaming_df.selectExpr("CAST(value AS STRING)").select(from_json(col("value"), eventSchema).alias("data")).select("data.*")

# Transform the event_timestamp into a Long (due to milliseconds precision)
transformed_df = sdf.withColumn("event_time",sdf.event_timestamp.cast(LongType())).drop("event_timestamp")

# Join the stream with the spotfy-songs dataframe
with_join_df = transformed_df.join(spotify_songs_df, transformed_df.song == spotify_songs_df.song_name, "inner")

# The dataframe to be sent to Cassandra
records_df = with_join_df.drop("song")

# Write the dataframe to cassandra
def writeToCassandra(writeDF, _):
  writeDF.write.format("org.apache.spark.sql.cassandra").mode('append').options(table="records", keyspace="spotify").save()

result = None

while result is None:
    try:
        # connect
        result = records_df.writeStream.option("spark.cassandra.connection.host","localhost:9042").\
          option("failOnDataLoss", "false").foreachBatch(writeToCassandra).\
          trigger(processingTime=SINK_WRITE_INTERVAL).\
          outputMode("update").start().awaitTermination()
    except:
          pass

```

Regarding samples of about 50 records

The following screenshots verify presence of data for different people, in cassandra.



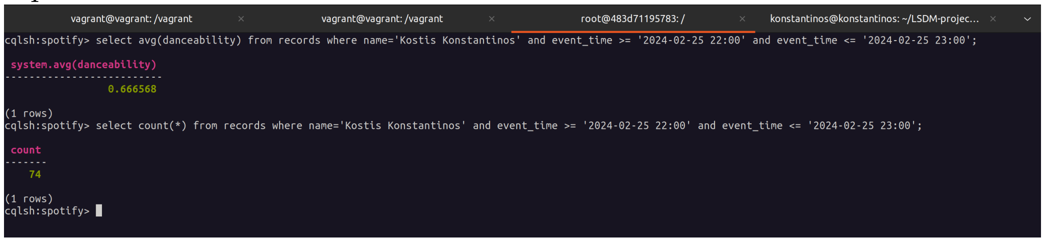
```
cqlsh:spotify> select name, count(*) as songs_listened, avg(energy) as avg_energy from records group by name limit 50;

 name                 | songs_listened | avg_energy
----------------------+----------------+------------
       Richard Bishop |             74 |   0.633409
         Andrew Smith |             74 |   0.598811
        Keith Sanchez |             72 |   0.675735
      Barbara Johnson |             73 |   0.619271
       Anthony Cooley |             73 |   0.657712
      Christian Munoz |             74 |   0.654919
   Suzanne Sanchez MD |             73 |    0.63163
    Stephanie Butler |             73 |   0.648959
         Tony Hawkins |             74 |      0.669
        Sarah Chapman |             73 |   0.633233
        Jeffrey Kelly |             73 |   0.624562
           John Young |             74 |   0.588292
 Kostis Konstantinos |             74 |   0.645095
          Scott Burke |             73 |   0.647425
      Monica Campbell |             74 |   0.630068
        Thomas Rivera |             71 |   0.663437
         John Stewart |             73 |    0.65073
       Jennifer Brown |             74 |   0.625145
      Stephanie Jones |             74 |    0.61573
       Tracy Williams |             72 |     0.6195
          Kevin Perry |             72 |   0.639931
            Erica May |             74 |   0.641068
      Matthew Jenkins |             72 |   0.632292
          John Garcia |             74 |   0.634892
      Nicholas Howard |             73 |   0.639863
     Amanda Maldonado |             73 |   0.639411
        Dennis Martin |             74 |   0.638545
      Tonya Rodriguez |             74 |   0.655095
       Patricia Smith |             72 |   0.615556
      Mr. Eric Glover |             74 |   0.657189
          Susan Graves |             73 |   0.664164
     Michele Anderson |             72 |   0.641722
          Eric Hughes |             73 |   0.658123
            Mary Long |             73 |   0.623562
        Harold Chavez |             73 |   0.640726
         Gary Goodwin |             73 |   0.653566
           Judy Davis |             74 |   0.644714
      Diana Gallagher |             72 |   0.639328
     Albert Wilkinson |             74 |   0.644392
          Sharon Cruz |             71 |   0.654662
            Troy Frye |             73 |   0.681918
        Ronald Adkins |             72 |   0.693681
       Deborah Harvey |             74 |   0.655311
     Jeffrey Williams |             74 |   0.604473
       Jeffrey Garcia |             73 |   0.644616
```



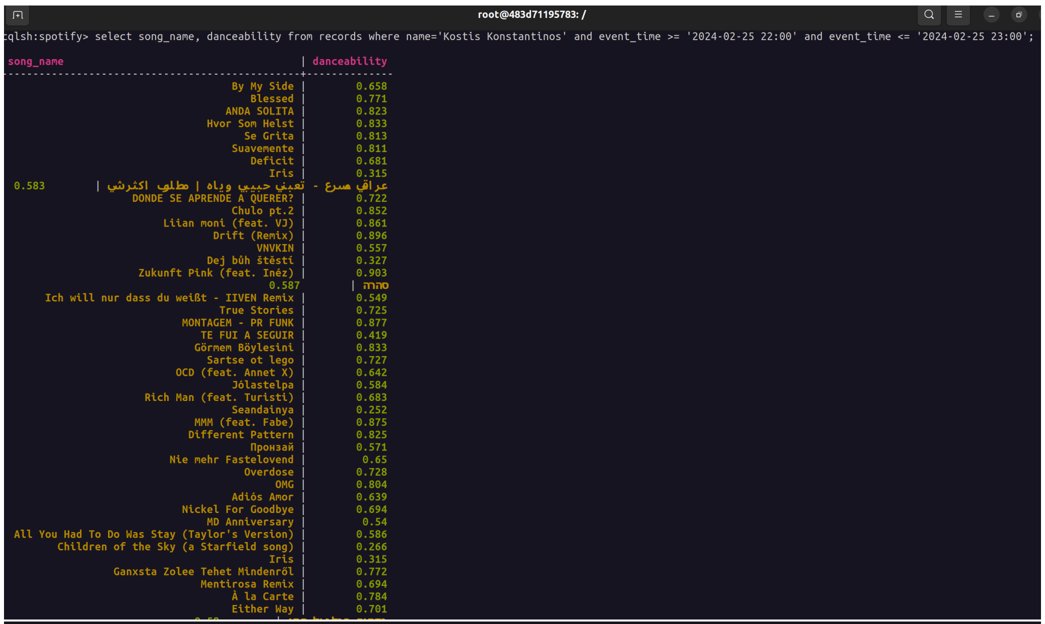
```
cqlsh:spotify> select name, song_name, album_name, energy, danceability from records where name='Stephanie Butler' limit 50;

 name             | song_name                                         | album_name                                          | energy | danceability
------------------+---------------------------------------------------+-----------------------------------------------------+--------+--------------
 Stephanie Butler |                                       Sans Coeur |                                 Pourvu qu'il pleuve |  0.764 |        0.765
 Stephanie Butler |                                     Rahasia Hati |                                            Paradoks |  0.526 |        0.542
 Stephanie Butler |                        AMBIENTE ERRADO - Ao Vivo |                              LUAN CITY 2.0 - FASE 2 |  0.845 |        0.484
 Stephanie Butler |                                          Голгофа |                                           NARRATIVE |  0.542 |        0.666
 Stephanie Butler |                            A Rainy Night in Soho |            Rum Sodomy & The Lash (Expanded Edition) |  0.449 |        0.375
 Stephanie Butler |                                           RITUAL |                      LVEU: VIVE LA TUYA...NO LA MIA |  0.706 |        0.832
 Stephanie Butler |                           Снег растаял на плечах |                              Снег растаял на плечах |  0.952 |         0.72
 Stephanie Butler |                             Hartslag Van De Stad |                                Hartslag Van De Stad |  0.737 |        0.808
 Stephanie Butler |                                             Sial |                                              fábula |  0.433 |        0.562
 Stephanie Butler |                                       Fact Check |                          Fact Check - The 5th Album |  0.908 |        0.758
 Stephanie Butler |                               Sena Kalpu Dziesma | Tautas Laiks (Latviesu Patriotisko Dziesmu Izlase) |  0.759 |        0.642
 Stephanie Butler |                                         Cariceps |                                            Cariceps |  0.809 |        0.963
 Stephanie Butler |                                            Widok |                                               Widok |  0.739 |        0.845
 Stephanie Butler |                                       Tại Vì Sao |                                                 99% |  0.682 |         0.76
 Stephanie Butler |                                Soy Feo Pero Rico |                                   De Amor y Vacilón |  0.847 |        0.738
 Stephanie Butler |                                       GIPSY TRAP |                                          GIPSY TRAP |  0.621 |         0.82
 Stephanie Butler |                                        Mucki Bar |                                Når sjælen kaster op |  0.739 |        0.785
 Stephanie Butler |                     3D (Justin Timberlake Remix) |                        3D (Justin Timberlake Remix) |  0.854 |        0.817
 Stephanie Butler |                                           CHUPON |                                              CHUPON |  0.748 |         0.73
 Stephanie Butler |                             Longbīča / Prospekti |                                   Melnezera grāmata |  0.701 |        0.763
 Stephanie Butler |                                      Stjernestøv |                                         Stjernestøv |  0.346 |         0.38
 Stephanie Butler |                                             käpy |                                                käpy |  0.631 |        0.705
 Stephanie Butler |                    Halloween Night in the Forest |                       Halloween Night in the Forest |  0.186 |        0.187
 Stephanie Butler |                                       Never Ever |                                    Presido La Pluto |  0.484 |        0.597
 Stephanie Butler | Wang Compromisa (feat. Focalistic & Makhekhe Jr) |                                         Boroko Keng |  0.644 |        0.879
 Stephanie Butler |                             Longbīča / Prospekti |                                   Melnezera grāmata |  0.701 |        0.763
 Stephanie Butler |                                          Nichijo |                                             Nichijo |  0.708 |        0.606
 Stephanie Butler |                        STAY (with Justin Bieber) |                              F*CK LOVE 3+: OVER YOU |  0.764 |        0.591
 Stephanie Butler |                                      Ik Maak Mee |                                         Ik Maak Mee |  0.405 |        0.694
 Stephanie Butler |                                           Badman |                                              Badman |  0.497 |        0.841
 Stephanie Butler |                 Ya No Vuelvas (Versión Cuarteto) |                    Ya No Vuelvas (Versión Cuarteto) |    0.9 |        0.736
 Stephanie Butler |                                     Blonde Chaya |                                        Blonde Chaya |   0.81 |        0.815
 Stephanie Butler |                                           Saiyan |                                    Chef D'orchestre |  0.808 |        0.775
 Stephanie Butler |                   จังหวะตกหลุมรัก - Magic Moment |                      จังหวะตกหลุมรัก (Magic Moment) |  0.424 |        0.563
 Stephanie Butler |                                       Nemam mood |                                          FLOWDEMORT |  0.797 |        0.833
 Stephanie Butler |                                          1 2 3 4 |                                      SÛR ET CERTAIN |  0.668 |        0.825
 Stephanie Butler |                         El Diario De Un Borracho |                                El Condor Legendario |  0.703 |        0.754
 Stephanie Butler |                                         Ніченька |                                            Ніченька |  0.533 |        0.575
 Stephanie Butler |                         Clean (Taylor's Version) |                             1989 (Taylor's Version) |  0.386 |        0.771
 Stephanie Butler |                              Arbolito de Navidad |                 El Mejor Disco de Diciembre, Vol. 2 |  0.631 |        0.741
 Stephanie Butler |                                     Luces Tenues |                                        Luces Tenues |  0.802 |         0.76
 Stephanie Butler |                Rockin' Around The Christmas Tree |                     Merry Christmas From Brenda Lee |  0.472 |        0.589
 Stephanie Butler |                                        Aşk Olsun |                                           Aşk Olsun |  0.849 |        0.754
 Stephanie Butler |                                              Dym |                                                 Dym |  0.718 |        0.718
```

Queries on my name

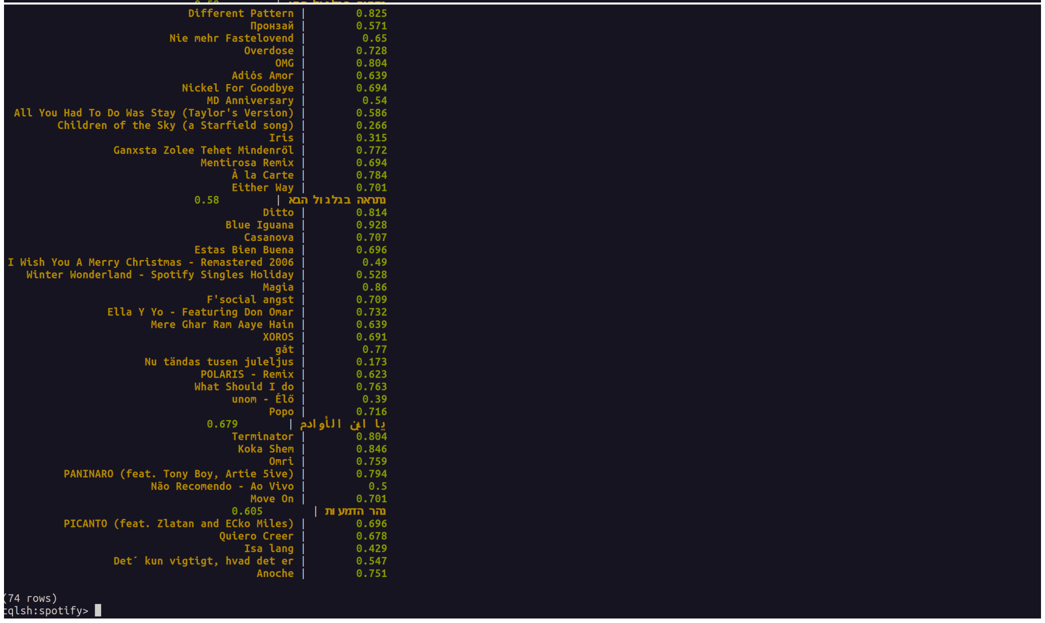
Below you can find screenshots of the queries (and results) for my name, as requested in the exercise.


vagrant@vagrant: /vagrant
vagrant@vagrant: /vagrant
root@483d71195783: /
konstantinos@konstantinos: ~/LSDM-projec...

```
cqlsh:spotify> select avg(danceability) from records where name='Kostis Konstantinos' and event_time >= '2024-02-25 22:00' and event_time <= '2024-02-25 23:00';

 system.avg(danceability)
--------------------------
                 0.666568

(1 rows)
cqlsh:spotify> select count(*) from records where name='Kostis Konstantinos' and event_time >= '2024-02-25 22:00' and event_time <= '2024-02-25 23:00';

 count
-------
    74

(1 rows)
cqlsh:spotify> 
```



```
root@483d71195783: /
cqlsh:spotify> select song_name, danceability from records where name='Kostis Konstantinos' and event_time >= '2024-02-25 22:00' and event_time <= '2024-02-25 23:00';

 song_name                                       | danceability
-------------------------------------------------+--------------
                                      By My Side |        0.658
                                         Blessed |        0.771
                                     ANDA SOLITA |        0.823
                                  Hvor Som Helst |        0.833
                                        Se Grita |        0.813
                                      Suavemente |        0.811
                                         Deficit |        0.681
                                            Iris |        0.315
 0.583        | مطلوب اكثرشي | تعبني حبيبي وياه - عراقي مسرع
                           DONDE SE APRENDE A QUERER? |        0.722
                                     Chulo pt.2 |        0.852
                             Liian moni (feat. VJ) |        0.861
                                   Drift (Remix) |        0.896
                                          VNVKIN |        0.557
                                  Dej bůh štěstí |        0.327
                         Zukunft Pink (feat. Inéz) |        0.903
                                           0.587        | סהרה
      Ich will nur dass du weißt - IIVEN Remix |        0.549
                                    True Stories |        0.725
                              MONTAGEM - PR FUNK |        0.877
                                 TE FUI A SEGUIR |        0.419
                                Görmem Böylesini |        0.833
                                  Sartse ot lego |        0.727
                              OCD (feat. Annet X) |        0.642
                                      Jólastelpa |        0.584
                         Rich Man (feat. Turisti) |        0.683
                                      Seandainya |        0.252
                                MMM (feat. Fabe) |        0.875
                               Different Pattern |        0.825
                                         Пронзай |        0.571
                              Nie mehr Fastelovend |         0.65
                                        Overdose |        0.728
                                             OMG |        0.804
                                      Adiós Amor |        0.639
                                Nickel For Goodbye |        0.694
                                  MD Anniversary |         0.54
 All You Had To Do Was Stay (Taylor's Version) |        0.586
      Children of the Sky (a Starfield song) |        0.266
                                            Iris |        0.315
                      Ganxsta Zolee Tehet Mindenről |        0.772
                                  Mentirosa Remix |        0.694
                                     À la Carte |        0.784
                                      Either Way |        0.701
```



```
                               Different Pattern |        0.825
                                         Пронзай |        0.571
                              Nie mehr Fastelovend |         0.65
                                        Overdose |        0.728
                                             OMG |        0.804
                                      Adiós Amor |        0.639
                                Nickel For Goodbye |        0.694
                                  MD Anniversary |         0.54
 All You Had To Do Was Stay (Taylor's Version) |        0.586
      Children of the Sky (a Starfield song) |        0.266
                                            Iris |        0.315
                      Ganxsta Zolee Tehet Mindenről |        0.772
                                  Mentirosa Remix |        0.694
                                     À la Carte |        0.784
                                      Either Way |        0.701
                              0.58        | נתראה בגלגול הבא
                                           Ditto |        0.814
                                     Blue Iguana |        0.928
                                        Casanova |        0.707
                                Estas Bien Buena |        0.696
 I Wish You A Merry Christmas - Remastered 2006 |         0.49
   Winter Wonderland - Spotify Singles Holiday |        0.528
                                           Magia |         0.86
                                  F'social angst |        0.709
                    Ella Y Yo - Featuring Don Omar |        0.732
                             Mere Ghar Ram Aaye Hain |        0.639
                                           XOROS |        0.691
                                             gát |         0.77
                            Nu tändas tusen juleljus |        0.173
                                 POLARIS - Remix |        0.623
                                  What Should I do |        0.763
                                     unom - Élő |         0.39
                                            Popo |        0.716
                                0.679        | يا ابن الأوادم
                                      Terminator |        0.804
                                       Koka Shem |        0.846
                                            Omri |        0.759
        PANINARO (feat. Tony Boy, Artie 5ive) |        0.794
                         Não Recomendo - Ao Vivo |          0.5
                                         Move On |        0.701
                                    0.605        | נהר הדמעות
        PICANTO (feat. Zlatan and ECko Miles) |        0.696
                                    Quiero Creer |        0.678
                                        Isa lang |        0.429
                      Det´ kun vigtigt, hvad det er |        0.547
                                          Anoche |        0.751

(74 rows)
cqlsh:spotify> 
```