Solution for week 2

Initially read the fixed json file having the artists and their recommended artists. Then read the csv fie containing the information regarding artists.

Here, notice that even though artists.csv has multiple column we are only selecting id and name. This will help pyspark optimize the read operation by only reading the required columns.

Join the two dataframes on id column, the join should be inner join

```
>>> from pyspark.sql import functions as F
>>> recommended_artists = spark.read.json("fixed_da.json")
>>> artists = spark.read.csv("artists.csv", header=True).select("id","name")
>>> joined = recommended_artists.join(artists, on="id", how="inner")
>>> joined.printSchema()
root
 |-- id: string (nullable = true)
 |-- related_ids: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- name: string (nullable = true)

>>>
```

Let us change the names to avoid naming conflicts. Explode the related ids into single rows. Then join with the artist df again to get the name of the related/recommended artists.

```
>>> joined = joined.withColumnRenamed("name", "artist_name")
>>> artists = artists.withColumnRenamed("id", "related_id").withColumnRenamed("name","related_artist_name")
>>> joined = joined.withColumn("related_id", F.explode("related_ids")).drop("related_ids")
>>> joined.printSchema()
root
 |-- id: string (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- related_id: string (nullable = true)

>>> joined = joined.join(artists, "related_id", "inner")
>>> joined.printSchema()
root
 |-- related_id: string (nullable = true)
 |-- id: string (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- related_artist_name: string (nullable = true)
```

Finally, aggregate the data again on id and artist name to collect all the related artists to that particular artists

```
>>> final = joined.groupBy("id", "artist_name").agg(F.collect_set("related_artist_name").alias("related_artists"))
>>> final.printSchema()
root
 |-- id: string (nullable = true)
 |-- artist_name: string (nullable = true)
 |-- related_artists: array (nullable = false)
 |    |-- element: string (containsNull = false)

>>> final.show(2)
+--------------------+--------------+--------------------+
|                  id|   artist_name|     related_artists|
+--------------------+--------------+--------------------+
|0001wHqxbF2YYRQxG...|  Motion Drive|[Yotopia, Protoni...|
|0006sHOabJ20USMjG...|The Art Company|[Maryla Rodowicz,...|
+--------------------+--------------+--------------------+
only showing top 2 rows
```

Finally, filter using artist_name and display the related artists. Make sure you keep truncate to false so you can view all the recommendations.

```
final.filter(final.artist_name == 'Nepathya').select("artist_name", "related_artists").show(truncate=False)
+-----------+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
|artist_name|related_artists

                                                                           |
+-----------+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
|Nepathya   |[Prem Dhoj Pradhan, Deepak Bajracharya, Andazification, Mongolian Heart, Arun Thapa, Cobweb, Deep Shres
 Prashant Tamang, Sanjay Shrestha, Tara Devi, The Uglyz, Narayan Gopal, Ram Krishna Dhakal, Sanjeev Singh, Anil Sing
974 Ad, Adrian Pradhan, Yogeshwor Amatya, Raju Lama, Astha Tamang-Maskey]|
+-----------+------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------+
```