Question 2

a)

dataframe-a.py

```python
from pyspark.sql import SparkSession
import json
import sys

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()


from pyspark import SparkContext

country_file = sys.argv[1]


sc = spark.sparkContext

country_df = spark.read.json(country_file)

"""##### query-a"""

answer = country_df.select(country_df['Name']).where(country_df['Continent']=="North
America").show()
```

```
+--------------------+
|                Name|
+--------------------+
|               Aruba|
|            Anguilla|
|Netherlands Antilles|
| Antigua and Barbuda|
|             Bahamas|
|              Belize|
|             Bermuda|
|            Barbados|
|              Canada|
|          Costa Rica|
|                Cuba|
|      Cayman Islands|
|            Dominica|
|  Dominican Republic|
|          Guadeloupe|
|             Grenada|
|           Greenland|
|           Guatemala|
|            Honduras|
|               Haiti|
+--------------------+
only showing top 20 rows
```

b)

dataframe-b.py

```python
from pyspark.sql import SparkSession
import json
import sys

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()


from pyspark import SparkContext

print("First enter country file, then enter city file\n")

country_file = sys.argv[1]                          # country file
city_file = sys.argv[2]                             # city file


sc = spark.sparkContext

country_df = spark.read.json(country_file)

city_df = spark.read.json(city_file)

query_b_df = country_df.join(city_df, country_df.Capital == city_df.ID).select(country_df.Name,city_df.Name).show()
```

```python
query_b_df = country_df.join(city_df, country_df.Capital == city_df.ID).select(country_df.Name,city_df.Name).show()
```

```
+--------------------+----------------+
|                Name|            Name|
+--------------------+----------------+
|         Afghanistan|           Kabul|
|         Netherlands|       Amsterdam|
|Netherlands Antilles|      Willemstad|
|             Albania|          Tirana|
|             Algeria|           Alger|
|      American Samoa|        Fagatogo|
|             Andorra|Andorra la Vella|
|              Angola|          Luanda|
|            Anguilla|      The Valley|
| Antigua and Barbuda|   Saint JohnÂ´s|
|United Arab Emirates|       Abu Dhabi|
|           Argentina|    Buenos Aires|
|             Armenia|         Yerevan|
|               Aruba|      Oranjestad|
|           Australia|        Canberra|
|          Azerbaijan|            Baku|
|             Bahamas|          Nassau|
|             Bahrain|       al-Manama|
|          Bangladesh|           Dhaka|
|            Barbados|      Bridgetown|
+--------------------+----------------+
only showing top 20 rows
```

c)

dataframe-c.py

```python
from pyspark.sql import SparkSession
import json
import sys

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()


from pyspark import SparkContext

country_file = sys.argv[1]              # country file

country_df = spark.read.json(country_file)


sc = spark.sparkContext


answer_c = country_df.select(country_df['Continent']).distinct().show()
```

```python
answer_c = country_df.select(country_df['Continent']).distinct().show()
```

```
+-------------+
|    Continent|
+-------------+
|       Europe|
|       Africa|
|North America|
|   Antarctica|
|South America|
|      Oceania|
|         Asia|
+-------------+
```

d)

dataframe-d.py

```python
from pyspark.sql import SparkSession
import json
import sys

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()


from pyspark import SparkContext

language_file = sys.argv[1]             # language file


cl_df = spark.read.json(language_file)

#cl_df.show()

answerd_d = cl_df.select(cl_df['Language']).filter(cl_df['CountryCode']=='CAN').show()

print(cl_df.select(cl_df['Language']).filter(cl_df['CountryCode']=='CAN').show())
```

```python
answerd_d = cl_df.select(cl_df['Language']).filter(cl_df['CountryCode']=='CAN').show()
```

```
+----------------+
|        Language|
+----------------+
|         Chinese|
|           Dutch|
|         English|
|Eskimo Languages|
|          French|
|          German|
|         Italian|
|          Polish|
|      Portuguese|
|         Punjabi|
|         Spanish|
|       Ukrainian|
+----------------+
```

e)

dataframe-e.py

```
from pyspark.sql import SparkSession
import json
import sys

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()


from pyspark import SparkContext
import pyspark.sql.functions as fc


print("Enter in format <python script> <country file> <city file>")

country_df = spark.read.json(country_file)

#country_df.select(country_df['Continent']).show()

answer_e = country_df.select(country_df['Continent'], country_df['LifeExpectancy']) \
            .groupBy(country_df['Continent']) \
            .agg(fc.avg(country_df['LifeExpectancy']).alias('avg_le'),fc.count('*').alias(
'count'))

final_ans = answer_e[answer_e['count'] >= 20].orderBy(fc.desc('count')).limit(1)

final_ans = final_ans.select(final_ans['Continent'],final_ans['avg_le']).show()
```

```
answer_e = country_df.select(country_df['Continent'], country_df['LifeExpectancy']) \
            .groupBy(country_df['Continent']) \
            .agg(fc.avg(country_df['LifeExpectancy']).alias('avg_le'),fc.count('*').alias('count'))
```

```
final_ans = answer_e[answer_e['count'] >= 20].orderBy(fc.desc('count')).limit(1)
```

```
final_ans = final_ans.select(final_ans['Continent'],final_ans['avg_le']).show()
```

```
+---------+----------------+
|Continent|          avg_le|
+---------+----------------+
|   Africa|51.6655172413793|
+---------+----------------+
```

Rdd_a)

rdd-a.py

```
from pyspark.sql import SparkSession

spark = SparkSession \
    .builder \
    .appName("HW5") \
    .getOrCreate()

from pyspark import SparkContext
import sys

sc = spark.sparkContext

country_file = sys.argv[1]

country_df = spark.read.json(country_file)

rdd_a = country_df.rdd.map(list)

print(rdd_a.map(lambda x: (x[11],x[3])).filter(lambda x: x[1] == "North
America").map(lambda x: x[0]).collect())
```

```
rdd_a.map(lambda x: (x[11],x[3])).filter(lambda x: x[1] == "North America").map(lambda x: x[0]).collect()
```

```
['Aruba',
 'Anguilla',
 'Netherlands Antilles',
 'Antigua and Barbuda',
 'Bahamas',
 'Belize',
 'Bermuda',
 'Barbados',
 'Canada',
 'Costa Rica',
 'Cuba',
 'Cayman Islands',
 'Dominica',
 'Dominican Republic',
 'Guadeloupe',
 'Grenada',
 'Greenland',
 'Guatemala',
 'Honduras',
 'Haiti',
 'Jamaica',
 'Saint Kitts and Nevis',
 'Saint Lucia',
 'Mexico',
 'Montserrat',
 'Martinique',
 'Nicaragua',
 'Panama',
 'Puerto Rico',
 'El Salvador',
```

Rdd_b)

38% Sat 9:12 PM Yash Naik

rdd-b.py

```
country_file = sys.argv[1]
city_file = sys.argv[2]

country_df = spark.read.json(country_file)

city_df = spark.read.json(city_file)


rdd_country = country_df.rdd.map(list)
rdd_city = city_df.rdd.map(list)

#rdd_country.take(1)

a = rdd_country.map(lambda x: (x[11],x[0]))
#a.collect()

#type(a)

#rdd_city.first()

b = rdd_city.map(lambda x: (x[3],x[2]))
#b.collect()

#type(b)

new_rdd = a.join(b)
rdd_query_b_answer = new_rdd.map(lambda x: (x[0],x[1][0])).collect()

print(rdd_query_b_answer)
```

```
new_rdd = a.join(b)
rdd_query_b_answer = new_rdd.map(lambda x: (x[0],x[1][0])).collect()
```

```
rdd_query_b_answer
```

```
[('Gibraltar', 915),
 ('Kuwait', 2429),
 ('Mexico', 2515),
 ('Armenia', 126),
 ('Djibouti', 585),
 ('Macao', 2454),
 ('Singapore', 3208),
 ('San Marino', 3171)]
```