UNIVERSIDADE DE COIMBRA

Faculdade de Ciências e Tecnologias,
Departamento de Engenharia Informática,

Mestrado em Engenharia e Ciência de Dados

Advanced Infraestructures for Data Science

Project #2

João Pino, 2020210945

Luís Seco, 2020211927

Assignment 1 - Kafka

Setup Zookeper and Kafka Broker

- zkServer.sh start
- cd
- cd kafka-3.9.0-src/
- bin/kafka-server-start.sh config/server.properties

Exercise 1

Terminal 1

- source venv/bin/activate
- python3 consumer.py

```
(venv) joaopino@din-org39 task1-topic %
python3 consumer.py
Listening to task1-topic...
Received: {'sensor_id': 4, 'temperature': 28.07}
Received: {'sensor_id': 8, 'temperature': 27.93}
Received: {'sensor_id': 4, 'temperature': 31.77}
Received: {'sensor_id': 9, 'temperature': 34.05}
```

Terminal 2

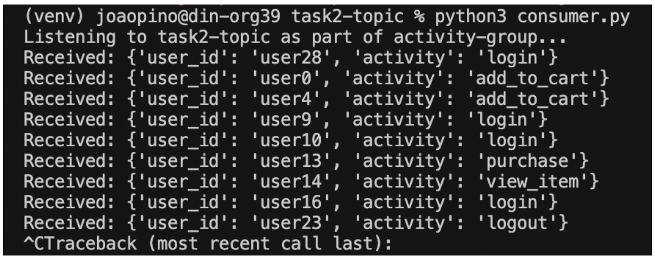
- source venv/bin/activate
- python3 producer.py

```
(venv) joaopino@din-org39 task1-topic %
python3 producer.py
Sent: {'sensor_id': 4, 'temperature': 28.07}
Sent: {'sensor_id': 8, 'temperature': 27.93}
Sent: {'sensor_id': 4, 'temperature': 31.77}
Sent: {'sensor_id': 9, 'temperature': 34.05}
```

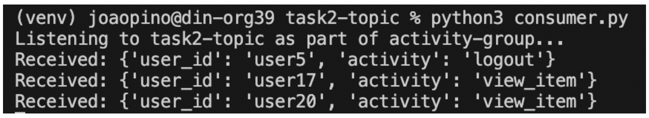
Exercise 2

- Create a topic-2 with 3 partitions:

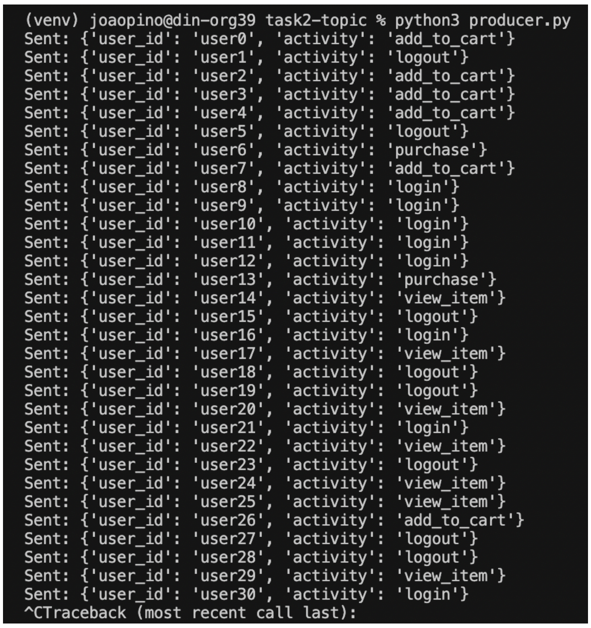
```
bin/kafka-topics.sh --create \
--topic task2-topic \
--bootstrap-server localhost:9092 \
--partitions 3 \
--replication-factor
```



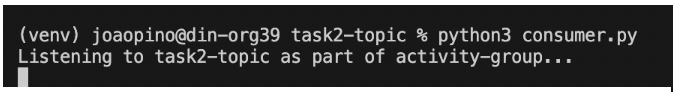
```
(venv) joaopino@din-org39 task2-topic % python3 consumer.py
Listening to task2-topic as part of activity-group...
Received: {'user_id': 'user28', 'activity': 'login'}
Received: {'user_id': 'user0', 'activity': 'add_to_cart'}
Received: {'user_id': 'user4', 'activity': 'add_to_cart'}
Received: {'user_id': 'user9', 'activity': 'login'}
Received: {'user_id': 'user10', 'activity': 'login'}
Received: {'user_id': 'user13', 'activity': 'purchase'}
Received: {'user_id': 'user14', 'activity': 'view_item'}
Received: {'user_id': 'user16', 'activity': 'login'}
Received: {'user_id': 'user23', 'activity': 'logout'}
^CTraceback (most recent call last):
```



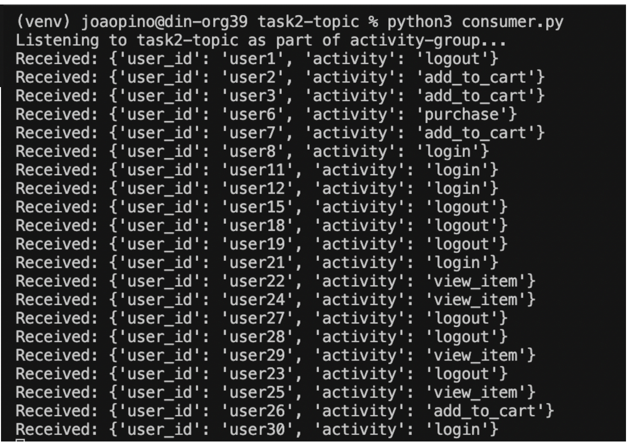
```
(venv) joaopino@din-org39 task2-topic % python3 consumer.py
Listening to task2-topic as part of activity-group...
Received: {'user_id': 'user5', 'activity': 'logout'}
Received: {'user_id': 'user17', 'activity': 'view_item'}
Received: {'user_id': 'user20', 'activity': 'view_item'}
```



```
(venv) joaopino@din-org39 task2-topic % python3 producer.py
Sent: {'user_id': 'user0', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user1', 'activity': 'logout'}
Sent: {'user_id': 'user2', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user3', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user4', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user5', 'activity': 'logout'}
Sent: {'user_id': 'user6', 'activity': 'purchase'}
Sent: {'user_id': 'user7', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user8', 'activity': 'login'}
Sent: {'user_id': 'user9', 'activity': 'login'}
Sent: {'user_id': 'user10', 'activity': 'login'}
Sent: {'user_id': 'user11', 'activity': 'login'}
Sent: {'user_id': 'user12', 'activity': 'login'}
Sent: {'user_id': 'user13', 'activity': 'purchase'}
Sent: {'user_id': 'user14', 'activity': 'view_item'}
Sent: {'user_id': 'user15', 'activity': 'logout'}
Sent: {'user_id': 'user16', 'activity': 'login'}
Sent: {'user_id': 'user17', 'activity': 'view_item'}
Sent: {'user_id': 'user18', 'activity': 'logout'}
Sent: {'user_id': 'user19', 'activity': 'logout'}
Sent: {'user_id': 'user20', 'activity': 'view_item'}
Sent: {'user_id': 'user21', 'activity': 'login'}
Sent: {'user_id': 'user22', 'activity': 'view_item'}
Sent: {'user_id': 'user23', 'activity': 'logout'}
Sent: {'user_id': 'user24', 'activity': 'view_item'}
Sent: {'user_id': 'user25', 'activity': 'view_item'}
Sent: {'user_id': 'user26', 'activity': 'add_to_cart'}
Sent: {'user_id': 'user27', 'activity': 'logout'}
Sent: {'user_id': 'user28', 'activity': 'logout'}
Sent: {'user_id': 'user29', 'activity': 'view_item'}
Sent: {'user_id': 'user30', 'activity': 'login'}
^CTraceback (most recent call last):
```



```
(venv) joaopino@din-org39 task2-topic % python3 consumer.py
Listening to task2-topic as part of activity-group...
```



```
(venv) joaopino@din-org39 task2-topic % python3 consumer.py
Listening to task2-topic as part of activity-group...
Received: {'user_id': 'user1', 'activity': 'logout'}
Received: {'user_id': 'user2', 'activity': 'add_to_cart'}
Received: {'user_id': 'user3', 'activity': 'add_to_cart'}
Received: {'user_id': 'user6', 'activity': 'purchase'}
Received: {'user_id': 'user7', 'activity': 'add_to_cart'}
Received: {'user_id': 'user8', 'activity': 'login'}
Received: {'user_id': 'user11', 'activity': 'login'}
Received: {'user_id': 'user12', 'activity': 'login'}
Received: {'user_id': 'user15', 'activity': 'logout'}
Received: {'user_id': 'user18', 'activity': 'logout'}
Received: {'user_id': 'user19', 'activity': 'logout'}
Received: {'user_id': 'user21', 'activity': 'login'}
Received: {'user_id': 'user22', 'activity': 'view_item'}
Received: {'user_id': 'user24', 'activity': 'view_item'}
Received: {'user_id': 'user27', 'activity': 'logout'}
Received: {'user_id': 'user28', 'activity': 'logout'}
Received: {'user_id': 'user29', 'activity': 'view_item'}
Received: {'user_id': 'user23', 'activity': 'logout'}
Received: {'user_id': 'user25', 'activity': 'view_item'}
Received: {'user_id': 'user26', 'activity': 'add_to_cart'}
Received: {'user_id': 'user30', 'activity': 'login'}
```

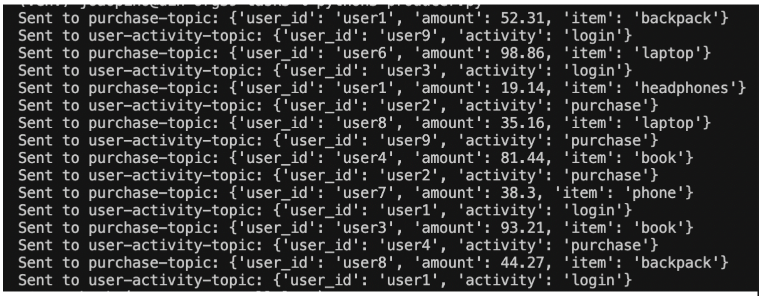
In this project, as we can see, 3 of the consumers are receiving information, one at a time, while 1 of them is not receiving any.
That is due to the topic only having three partitions, so only 3 consumerscan actively listen at a time.

We also analysed that in the case of failure in one of the consumers, the fourth one that was actively listening starts receiving information.

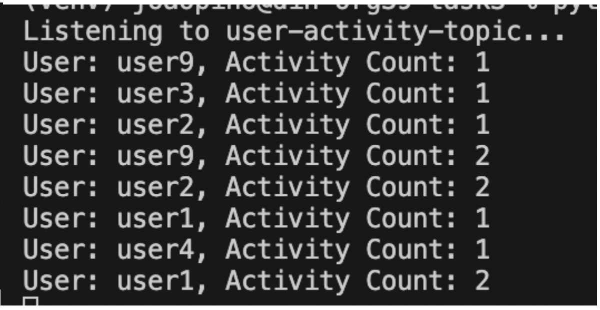
Exercise 3

Create Purchase topic
bin/kafka-topics.sh --create \
--topic purchase-topic \
--bootstrap-server localhost:9092 \
--partitions 3 \
--replication-factor 1

Create user-activity-topic
bin/kafka-topics.sh --create \
--topic user-activity-topic \
--bootstrap-server localhost:9092 \
--partitions 3 \
--replication-factor

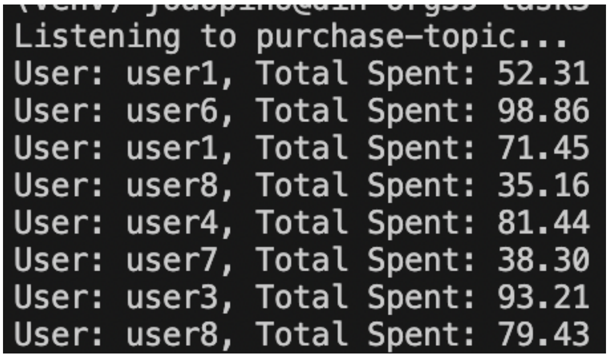


```
Sent to purchase-topic: {'user_id': 'user1', 'amount': 52.31, 'item': 'backpack'}
Sent to user-activity-topic: {'user_id': 'user9', 'activity': 'login'}
Sent to purchase-topic: {'user_id': 'user6', 'amount': 98.86, 'item': 'laptop'}
Sent to user-activity-topic: {'user_id': 'user3', 'activity': 'login'}
Sent to purchase-topic: {'user_id': 'user1', 'amount': 19.14, 'item': 'headphones'}
Sent to user-activity-topic: {'user_id': 'user2', 'activity': 'purchase'}
Sent to purchase-topic: {'user_id': 'user8', 'amount': 35.16, 'item': 'laptop'}
Sent to user-activity-topic: {'user_id': 'user9', 'activity': 'purchase'}
Sent to purchase-topic: {'user_id': 'user4', 'amount': 81.44, 'item': 'book'}
Sent to user-activity-topic: {'user_id': 'user2', 'activity': 'purchase'}
Sent to purchase-topic: {'user_id': 'user7', 'amount': 38.3, 'item': 'phone'}
Sent to user-activity-topic: {'user_id': 'user1', 'activity': 'login'}
Sent to purchase-topic: {'user_id': 'user3', 'amount': 93.21, 'item': 'book'}
Sent to user-activity-topic: {'user_id': 'user4', 'activity': 'purchase'}
Sent to purchase-topic: {'user_id': 'user8', 'amount': 44.27, 'item': 'backpack'}
Sent to user-activity-topic: {'user_id': 'user1', 'activity': 'login'}
```



```
Listening to user-activity-topic...
User: user9, Activity Count: 1
User: user3, Activity Count: 1
User: user2, Activity Count: 1
User: user9, Activity Count: 2
User: user2, Activity Count: 2
User: user1, Activity Count: 1
User: user4, Activity Count: 1
User: user1, Activity Count: 2
```

With this exercise we evaluated the ability of an producer to feed information into various topics and the advantage of local processment.



```
Listening to purchase-topic...
User: user1, Total Spent: 52.31
User: user6, Total Spent: 98.86
User: user1, Total Spent: 71.45
User: user8, Total Spent: 35.16
User: user4, Total Spent: 81.44
User: user7, Total Spent: 38.30
User: user3, Total Spent: 93.21
User: user8, Total Spent: 79.43
```

Assignment 2 - Hadoop Map Reduce

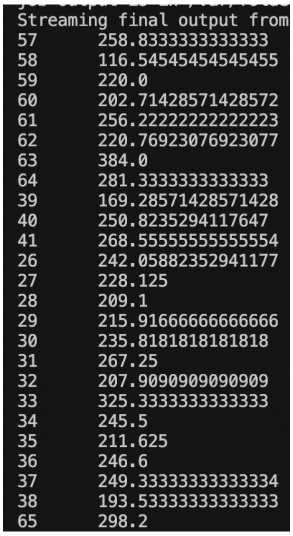
For this assignment we had to create a python environment with python 3.10 for it be compatible to MrJob package.

- python3.10 -m venv python3_10_venv
- source python3_10_venv/bin/activate
- pip install mrjob

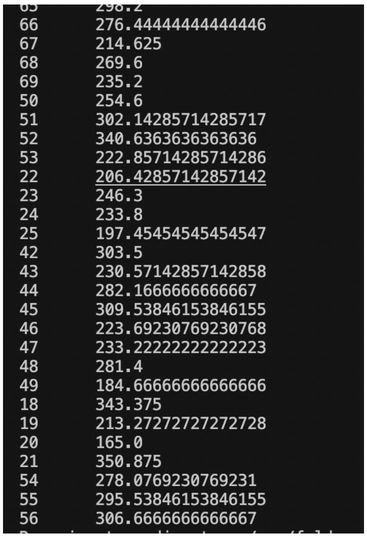
Exercise 1

- python3 MapReduce/Task_1/task1.py MapReduce/Task_1/fakefriends.csv

- Read the CSV
- Map all entries in key value form: (age: number of friends)
- Reducer to Computes the total number of friends and calculates the average number of friends for each age.
- Converts age to an integer and yields it alongside the average.



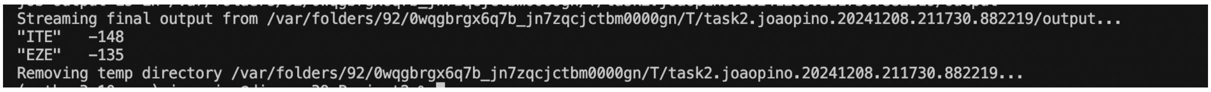
```
Streaming final output from
57      258.8333333333333
58      116.54545454545455
59      220.0
60      202.71428571428572
61      256.22222222222223
62      220.76923076923077
63      384.0
64      281.3333333333333
39      169.28571428571428
40      250.8235294117647
41      268.55555555555554
26      242.05882352941177
27      228.125
28      209.1
29      215.91666666666666
30      235.8181818181818
31      267.25
32      207.9090909090909
33      325.3333333333333
34      245.5
35      211.625
36      246.6
37      249.33333333333334
38      193.53333333333333
65      298.2
```



```
66      276.44444444444446
67      214.625
68      269.6
69      235.2
50      254.6
51      302.14285714285717
52      340.6363636363636
53      222.85714285714286
22      206.42857142857142
23      246.3
24      233.8
25      197.45454545454547
42      303.5
43      230.57142857142858
44      282.1666666666667
45      309.53846153846155
46      223.69230769230768
47      233.22222222222223
48      281.4
49      184.66666666666666
18      343.375
19      213.27272727272728
20      165.0
21      350.875
54      278.0769230769231
55      295.53846153846155
56      306.6666666666667
```

Exercise 2

- python3 MapReduce/Task_2/task2.py MapReduce/Task_2/1800.csv



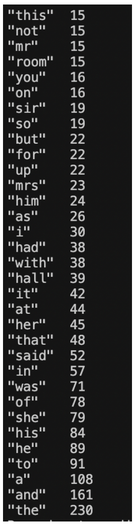
```
Streaming final output from /var/folders/92/0wqgbrgx6q7b_jn7zqcjctbm0000gn/T/task2.joaopino.20241208.211730.882219/output...
"ITE"   -148
"EZE"   -135
Removing temp directory /var/folders/92/0wqgbrgx6q7b_jn7zqcjctbm0000gn/T/task2.joaopino.20241208.211730.882219...
```

- Mapper: Extracts the capital name and minimum temperature (`TMIN`) as key-value pairs (capital, temperature).
- Reducer: Finds and yields the minimum temperature for each capital.

Exercise 3

- python3 MapReduce/Task_3/task3.py MapReduce/Task_3/Book

- Mapper 1: Extracts words from lines (removing punctuation and converting to lowercase) and emits (word, 1).
- Reducer 1: Counts occurrences of each word and emits (word, total_count).
- Mapper 2: Prepares words for sorting by emitting (None, (count, word)).
- Reducer 2: Sorts words by count in descending order and emits (word, count).

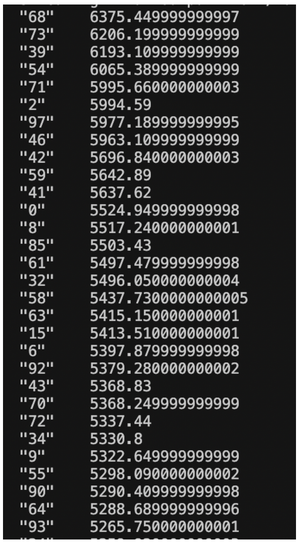


```
"this"  15
"not"   15
"mr"    15
"room"  15
"you"   16
"on"    16
"sir"   19
"so"    19
"but"   22
"for"   22
"up"    22
"mrs"   23
"him"   24
"as"    26
"i"     30
"had"   38
"with"  38
"hall"  39
"it"    42
"at"    44
"her"   45
"that"  48
"said"  52
"in"    57
"was"   71
"of"    78
"she"   79
"his"   84
"he"    89
"to"    91
"a"     108
"and"   161
"the"   230
```

Exercise 4

- python3 MapReduce/Task_4/task4.py MapReduce/Task_4/customer-orders.csv

- Mapper 1: Extracts customer IDs and their spending amounts, emitting (customer_id, amount).
- Reducer 1: Sums up the spending for each customer, emitting (customer_id, total_amount).
- Mapper 2: Prepares data for sorting by emitting (None, (total_amount, customer_id)).
- Reducer 2: Sorts customers by total spending in descending order and emits (customer_id, total_amount).



```
"68"    6375.449999999997
"73"    6206.199999999999
"39"    6193.109999999999
"54"    6065.389999999999
"71"    5995.660000000003
"2"     5994.59
"97"    5977.189999999995
"46"    5963.109999999999
"42"    5696.840000000003
"59"    5642.89
"41"    5637.62
"0"     5524.949999999998
"8"     5517.240000000001
"85"    5503.43
"61"    5497.479999999998
"32"    5496.050000000004
"58"    5437.7300000000005
"63"    5415.150000000001
"15"    5413.510000000001
"6"     5397.879999999998
"92"    5379.280000000002
"43"    5368.83
"70"    5368.249999999999
"72"    5337.44
"34"    5330.8
"9"     5322.649999999999
"55"    5298.090000000002
"90"    5290.409999999998
"64"    5288.689999999996
"93"    5265.750000000001
```

Assignment 3 - Spark RDD

In this assignment we used the python3.13 env we created in Assignment 1 and added PySpark.

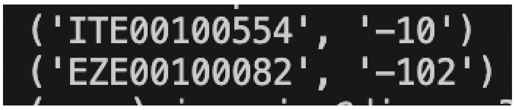
- source venv/bin/activate
- pip install pyspark
- cd RDD_Aproach

In this assignment we didn’t add the commands run because it was only necessary to run the python script.

In instances where a new RDD is created thanks to transformations we will only referencing the “Transformation”

Exercise 1

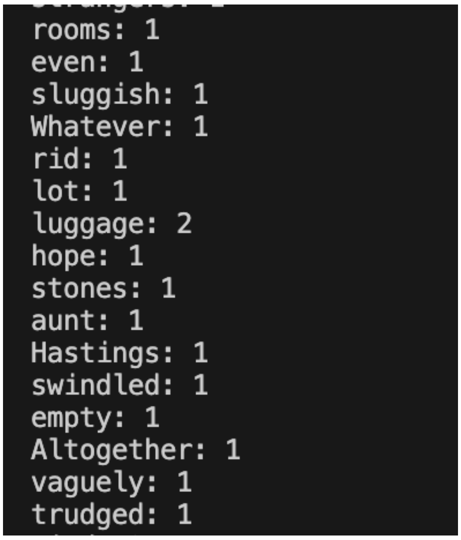
- Read file and create RDD
- Create new RDD from the previous by transforming RDD into a map of tuples: (stationID, entryType, temperature)
- Transform RDD to filter only records where entryType is "TMIN".
- Transform RDD to to only have tuples (stationID, temperature)
- Reduce using the key stationID by calculating the min in the temperature
- Collects the results and prints



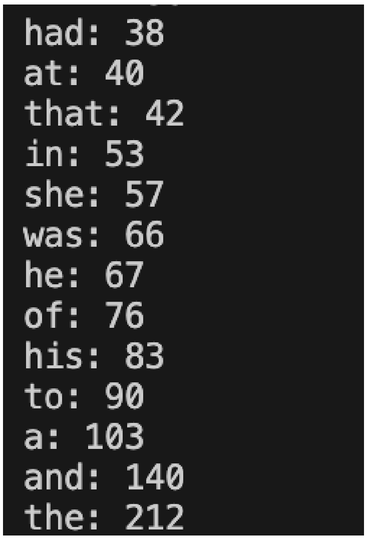
```
('ITE00100554', '-10')
('EZE00100082', '-102')
```

Exercise 2 and 3

- Read file and create RDD
- Transform using flatmap to create more entries based on an split (making the new RDD have all the words)
- Transform to filter RDD to only have words using function isalpha
- Transform RDD to have key-value tupples (word, 1)
- Reduce the RDD based on the key word and summing the values (getting the occurrences of the word)
- [Exercise 3] Transform RDD to be sorted by the key occurrences.
- Collects and prints results



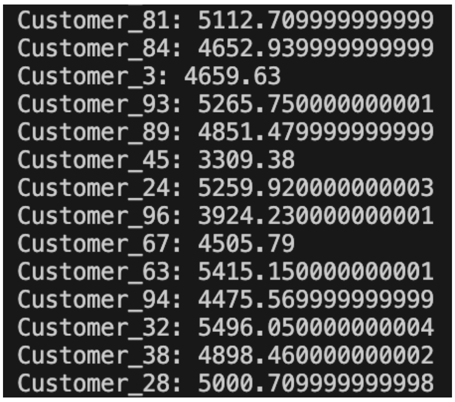
```
rooms: 1
even: 1
sluggish: 1
Whatever: 1
rid: 1
lot: 1
luggage: 2
hope: 1
stones: 1
aunt: 1
Hastings: 1
swindled: 1
empty: 1
Altogether: 1
vaguely: 1
trudged: 1
```



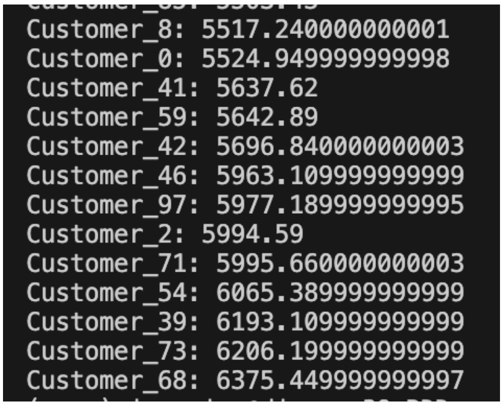
```
had: 38
at: 40
that: 42
in: 53
she: 57
was: 66
he: 67
of: 76
his: 83
to: 90
a: 103
and: 140
the: 212
```

Exercise 4 and 5

- Read the file and create the RDD
- Transform the RDD to have key-value pair of (customerID, amount)
- Reduce the RDD by the key customerID summing the value amount
- [Exercice 5] Sort the RDD by the amount
- Collect from the RDD and print the output



```
Customer_81: 5112.709999999999
Customer_84: 4652.939999999999
Customer_3: 4659.63
Customer_93: 5265.750000000001
Customer_89: 4851.479999999999
Customer_45: 3309.38
Customer_24: 5259.920000000003
Customer_96: 3924.230000000001
Customer_67: 4505.79
Customer_63: 5415.150000000001
Customer_94: 4475.569999999999
Customer_32: 5496.050000000004
Customer_38: 4898.460000000002
Customer_28: 5000.709999999998
```



```
Customer_8: 5517.240000000001
Customer_0: 5524.949999999998
Customer_41: 5637.62
Customer_59: 5642.89
Customer_42: 5696.840000000003
Customer_46: 5963.109999999999
Customer_97: 5977.189999999995
Customer_2: 5994.59
Customer_71: 5995.660000000003
Customer_54: 6065.389999999999
Customer_39: 6193.109999999999
Customer_73: 6206.199999999999
Customer_68: 6375.449999999997
```

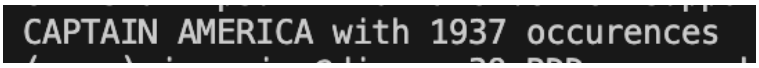
Exercise 6 and 7

RDD 1: Marvel Graph

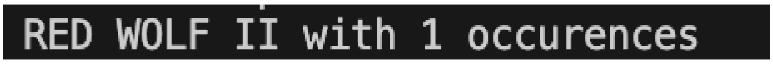
- Read the file and create the RDD
- Transform the RDD by using split to get all the mentions of a superhero
- Transform the RDD by filtering empty words “”
- Transform the RDD creating an map of (word,1)
- Reducing the RDD by the value superheroID and summing the occurrences
- [Exercise 6] Transforming the RDD by filtering from the key occurrence (Descending)
- [Exercise 7] Transforming the RDD by filtering from the key occurrence (Ascending)
- Taking the number 1 spot value
- [After the transformations in RDD 2]
- Collecting the RDD and print the output

RDD 2: Marvel Names

- Read the file and create the RDD
- Transforming the RDD by mapping (id, names) and removing the “ characters from the name
- [After all the transformations on the RDD 1…]
- Transforming the RDD by filtering to only display the value from the superheroID obtained through the collection in RDD1
- Collecting the RDD and print the output



```
CAPTAIN AMERICA with 1937 occurences
```



```
RED WOLF II with 1 occurences
```

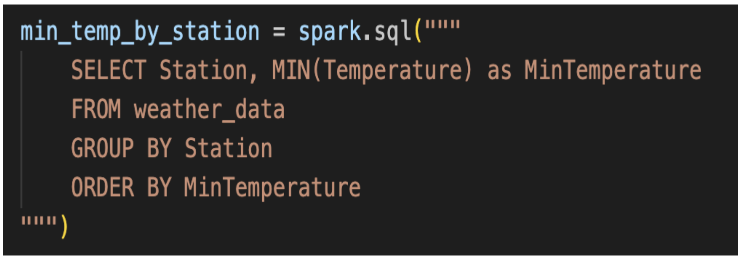
Assignment 4 - Spark SQL

This assignment was very similar to assignment 3 but instead of using RDD we used SQL queries. We created our environment using the following steps:

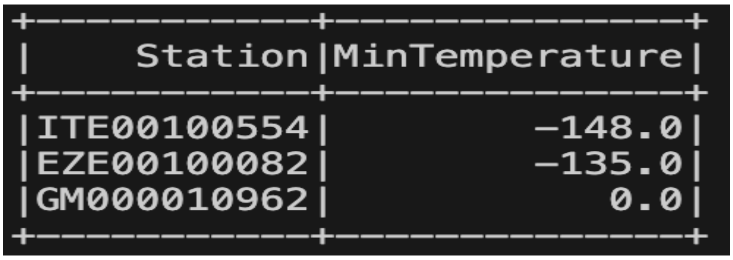
- source venv/bin/activate
- cd Spark_SQL

Exercise 1

- Create the Spark session and read the csv file containing the weather data
- Create a function called *mapper* that returns the data of each row split in four in order to have each field separated and then proceed with the mapping
- Create a Spark Dataframe after the mapping
- Use spark SQL to create a query that selects the minimum temperature for each station
- Show on the terminal and then stop Spark



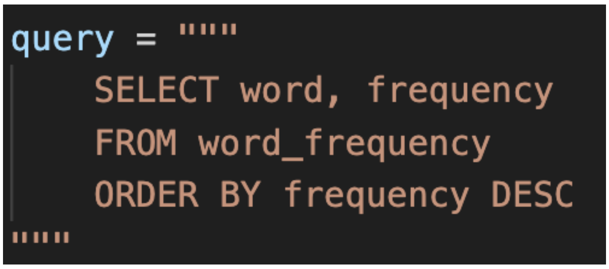
```
min_temp_by_station = spark.sql("""
    SELECT Station, MIN(Temperature) as MinTemperature
    FROM weather_data
    GROUP BY Station
    ORDER BY MinTemperature
""")
```



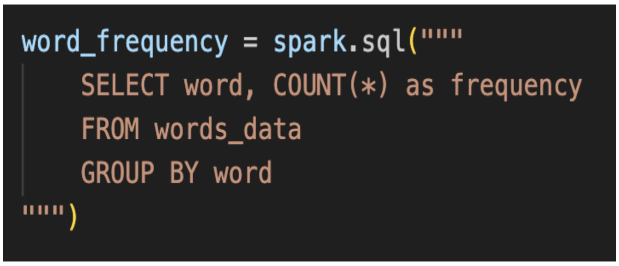
```
+-----------+--------------+
|    Station|MinTemperature|
+-----------+--------------+
|ITE00100554|        -148.0|
|EZE00100082|        -135.0|
|GM000010962|           0.0|
+-----------+--------------+
```

Exercise 2 and 3

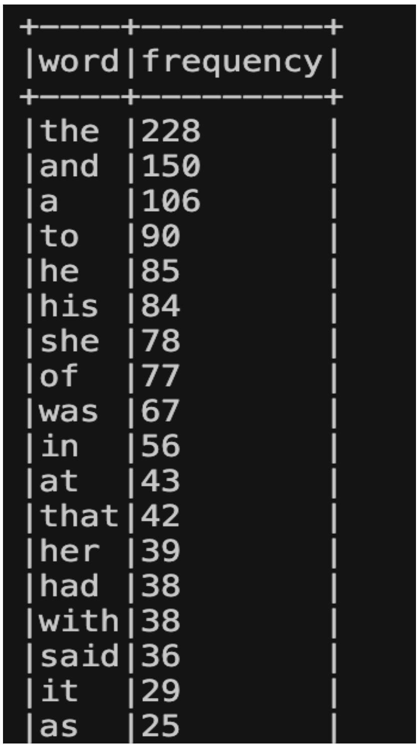
- Create the Spark session and read the file containing the words data
- Create a function called *mapper* that returns the data of each row, in an array, split word by word in lowercase
- Create a Spark Dataframe of the words in the data book after the mapping
- Use spark SQL to create a query that counts the frequency of each word
- In exercise 2 show the frequency of one hundred words without taking account the most frequent words
- In exercise 3 do the same process but this time printing the most frequent words
- Stop Spark



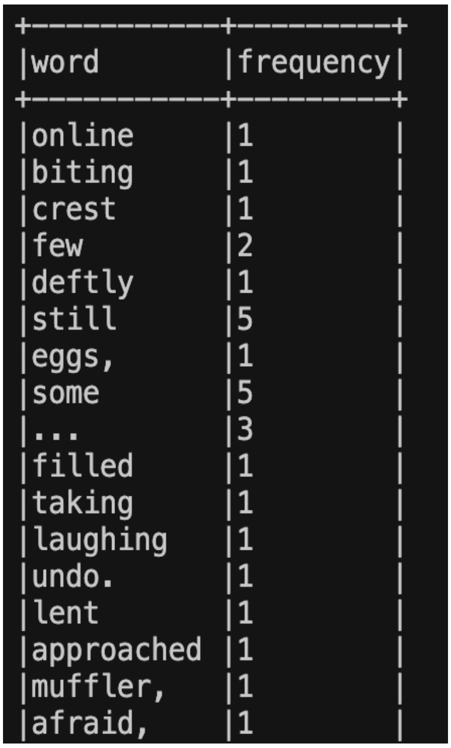
```
query = """
    SELECT word, frequency
    FROM word_frequency
    ORDER BY frequency DESC
"""
```



```
word_frequency = spark.sql("""
    SELECT word, COUNT(*) as frequency
    FROM words_data
    GROUP BY word
""")
```



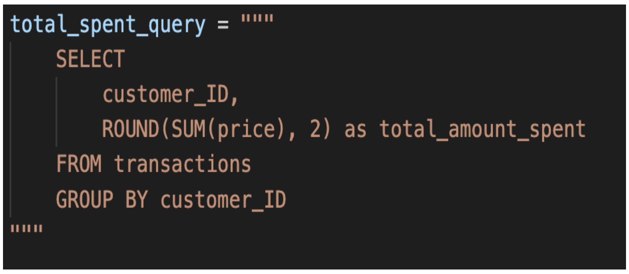
```
+----+---------+
|word|frequency|
+----+---------+
|the |228      |
|and |150      |
|a   |106      |
|to  |90       |
|he  |85       |
|his |84       |
|she |78       |
|of  |77       |
|was |67       |
|in  |56       |
|at  |43       |
|that|42       |
|her |39       |
|had |38       |
|with|38       |
|said|36       |
|it  |29       |
|as  |25       |
```



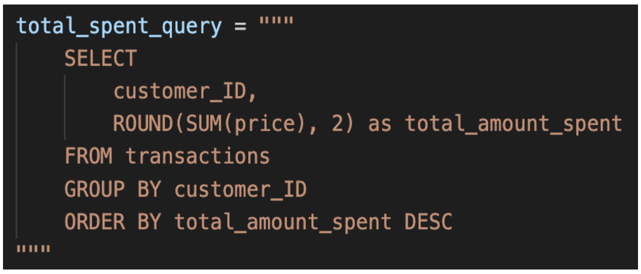
```
+-----------+---------+
|word       |frequency|
+-----------+---------+
|online     |1        |
|biting     |1        |
|crest      |1        |
|few        |2        |
|deftly     |1        |
|still      |5        |
|eggs,      |1        |
|some       |5        |
|...        |3        |
|filled     |1        |
|taking     |1        |
|laughing   |1        |
|undo.      |1        |
|lent       |1        |
|approached |1        |
|muffler,   |1        |
|afraid,    |1        |
```

Exercise 4 and 5

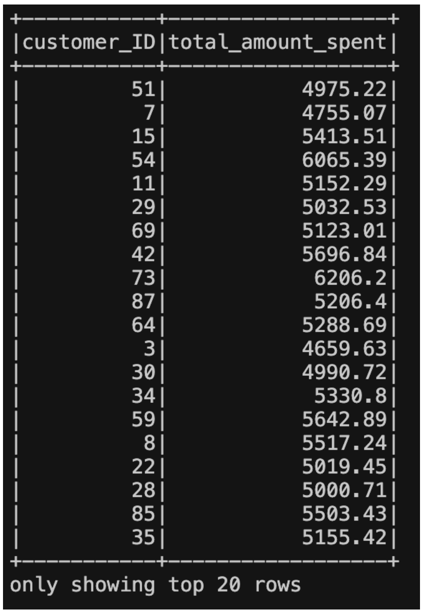
- Create the Spark session and read the file containing the consumer data
- Create a function called *mapper* that returns the data of each row in three in order to have each field separated and then proceed with the mapping
- Create a Spark Dataframe of the consumers information retrieved in the last step
- Use spark SQL to create a query that counts the amount spent by each consumer
- In exercise 4 show the amount spent of each consumer without taking account the users that spent the most
- In exercise 5 do the same process but this time printing the top spenders
- Stop Spark



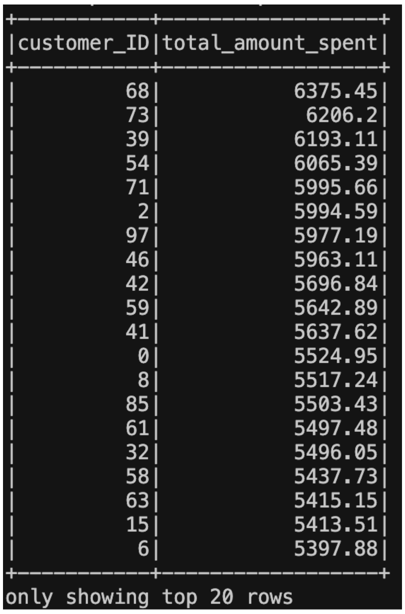
```
total_spent_query = """
    SELECT
        customer_ID,
        ROUND(SUM(price), 2) as total_amount_spent
    FROM transactions
    GROUP BY customer_ID
"""
```



```
total_spent_query = """
    SELECT
        customer_ID,
        ROUND(SUM(price), 2) as total_amount_spent
    FROM transactions
    GROUP BY customer_ID
    ORDER BY total_amount_spent DESC
"""
```



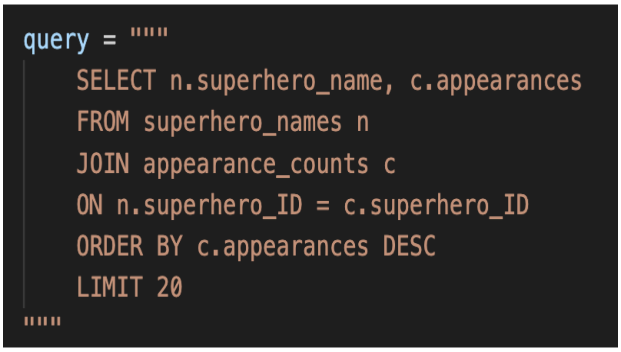
```
+-----------+------------------+
|customer_ID|total_amount_spent|
+-----------+------------------+
|         51|           4975.22|
|          7|           4755.07|
|         15|           5413.51|
|         54|           6065.39|
|         11|           5152.29|
|         29|           5032.53|
|         69|           5123.01|
|         42|           5696.84|
|         73|            6206.2|
|         87|            5206.4|
|         64|           5288.69|
|          3|           4659.63|
|         30|           4990.72|
|         34|            5330.8|
|         59|           5642.89|
|          8|           5517.24|
|         22|           5019.45|
|         28|           5000.71|
|         85|           5503.43|
|         35|           5155.42|
+-----------+------------------+
only showing top 20 rows
```



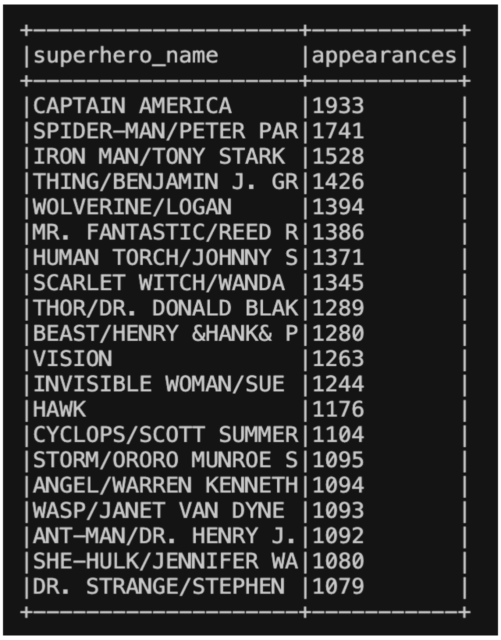
```
+-----------+------------------+
|customer_ID|total_amount_spent|
+-----------+------------------+
|         68|           6375.45|
|         73|            6206.2|
|         39|           6193.11|
|         54|           6065.39|
|         71|           5995.66|
|          2|           5994.59|
|         97|           5977.19|
|         46|           5963.11|
|         42|           5696.84|
|         59|           5642.89|
|         41|           5637.62|
|          0|           5524.95|
|          8|           5517.24|
|         85|           5503.43|
|         61|           5497.48|
|         32|           5496.05|
|         58|           5437.73|
|         63|           5415.15|
|         15|           5413.51|
|          6|           5397.88|
+-----------+------------------+
only showing top 20 rows
```

Exercise 6 and 7

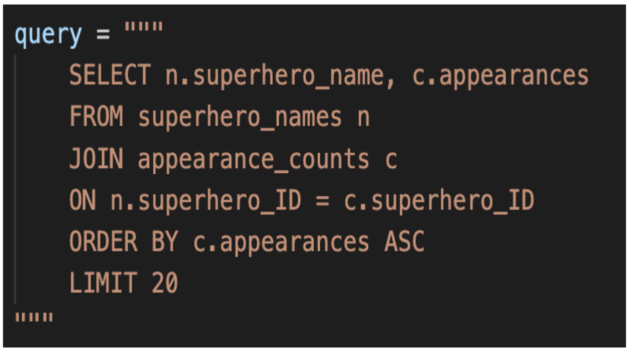
- Create the Spark session and read the file containing the data from Marvel superheroes appearances
- Create a variable that counts the appearances of each superhero ID
- Read the file containing the correspondence of each ID to each superhero name
- Create a Spark Dataframe containing the ID and the number of appearances of that ID
- Create a Spark Dataframe of the correspondence of each ID and superhero name
- Use spark SQL to create a query that uses both dataframes in order to obtain the correspondence between the number of appearances and the name of the superhero
- In exercise 6 show the top 20 most popular superheroes
- In exercise 7 do the same process but this time showing the less 20 popular superheroes
- Stop Spark



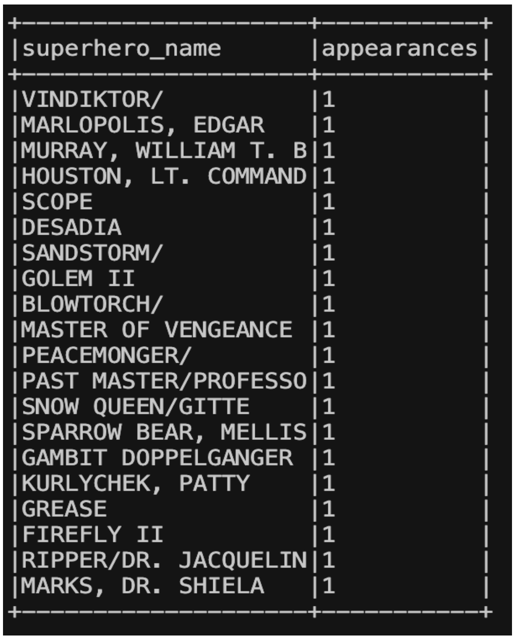
```
query = """
    SELECT n.superhero_name, c.appearances
    FROM superhero_names n
    JOIN appearance_counts c
    ON n.superhero_ID = c.superhero_ID
    ORDER BY c.appearances DESC
    LIMIT 20
"""
```



```
+--------------------+-----------+
|superhero_name      |appearances|
+--------------------+-----------+
|CAPTAIN AMERICA     |1933       |
|SPIDER-MAN/PETER PAR|1741       |
|IRON MAN/TONY STARK |1528       |
|THING/BENJAMIN J. GR|1426       |
|WOLVERINE/LOGAN     |1394       |
|MR. FANTASTIC/REED R|1386       |
|HUMAN TORCH/JOHNNY S|1371       |
|SCARLET WITCH/WANDA |1345       |
|THOR/DR. DONALD BLAK|1289       |
|BEAST/HENRY &HANK& P|1280       |
|VISION              |1263       |
|INVISIBLE WOMAN/SUE |1244       |
|HAWK                |1176       |
|CYCLOPS/SCOTT SUMMER|1104       |
|STORM/ORORO MUNROE S|1095       |
|ANGEL/WARREN KENNETH|1094       |
|WASP/JANET VAN DYNE |1093       |
|ANT-MAN/DR. HENRY J.|1092       |
|SHE-HULK/JENNIFER WA|1080       |
|DR. STRANGE/STEPHEN |1079       |
+--------------------+-----------+
```



```
query = """
    SELECT n.superhero_name, c.appearances
    FROM superhero_names n
    JOIN appearance_counts c
    ON n.superhero_ID = c.superhero_ID
    ORDER BY c.appearances ASC
    LIMIT 20
"""
```



```
+--------------------+-----------+
|superhero_name      |appearances|
+--------------------+-----------+
|VINDIKTOR/          |1          |
|MARLOPOLIS, EDGAR   |1          |
|MURRAY, WILLIAM T. B|1          |
|HOUSTON, LT. COMMAND|1          |
|SCOPE               |1          |
|DESADIA             |1          |
|SANDSTORM/          |1          |
|GOLEM II            |1          |
|BLOWTORCH/          |1          |
|MASTER OF VENGEANCE |1          |
|PEACEMONGER/        |1          |
|PAST MASTER/PROFESSO|1          |
|SNOW QUEEN/GITTE    |1          |
|SPARROW BEAR, MELLIS|1          |
|GAMBIT DOPPELGANGER |1          |
|KURLYCHEK, PATTY    |1          |
|GREASE              |1          |
|FIREFLY II          |1          |
|RIPPER/DR. JACQUELIN|1          |
|MARKS, DR. SHIELA   |1          |
+--------------------+-----------+
```