Assignment 03 – Big Data Management

Student Number: 3049429 **Student Name: Jawad Adil**

I have used this code to split files.

```
import time
csvfile = open('./assigment1/2021-10-31.csv', 'r').readlines()
# removing header
csvfile.pop(0)
filename = 1
counter = 0
for i in range(0,len(csvfile)):
    # 120/3 = 40, hence I'm stopping at 40 files
    if filename>40:
        break
    if i % 1000 == 0:
        open("./streaming/logs"+str(filename) + '.csv', 'w+').writelines(csvfile[counter:counter+1000])
        filename += 1
        counter+= 1000;
        time.sleep(3)
```

This code runs parallel to the streaming process. I>60 check is placed because I'm running streaming for 3 minutes and there are 180 seconds in 3 minutes. We are splitting the data after every 3 seconds and streaming is reading after 3 seconds as well. So, 180/3 = 60, this means we will have 60 files in 3 minutes. Without this check, the code will run until the whole main file is splited, and this is not necessary. Hence, this check will stop after 3 minutes with 60 files.

Code:

This 1st function is used to remove double quotes in the following code. Rest 4 functions are used to store data in Cassandra for respective questions.

```
def remove_quotation(x):
    return([xx.replace('"', '') for xx in x])
def save_package_count(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
        lambda row: Row(time=time, package=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
        .options(table="package_count", keyspace="streaming")\
        .save(mode="append")
    except:
        pass
def save_top_package(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
        lambda row: Row(time=time, package=row[0], count=row[1])))
        df = df.limit(1)
        df.write.format("org.apache.spark.sql.cassandra")\
        .options(table="top_package", keyspace="streaming")\
        .save(mode="append")
    except:
        pass
def save_country(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
        lambda row: Row(time=time, country=row[0], count=row[1])))
        df = df.limit(5)
        df.write.format("org.apache.spark.sql.cassandra")\
        .options(table="top_country", keyspace="streaming")\
        .save(mode="append")
    except:
        pass
def save_ggplot2_count(time, rdd):
    try:
        df = spark.createDataFrame(rdd.map(\
        lambda row: Row(time=time, ggplot2=row[0], count=row[1])))
        df.write.format("org.apache.spark.sql.cassandra")\
        .options(table="ggplot2_count", keyspace="streaming")\
        .save(mode="append")
    except:
        pass
```

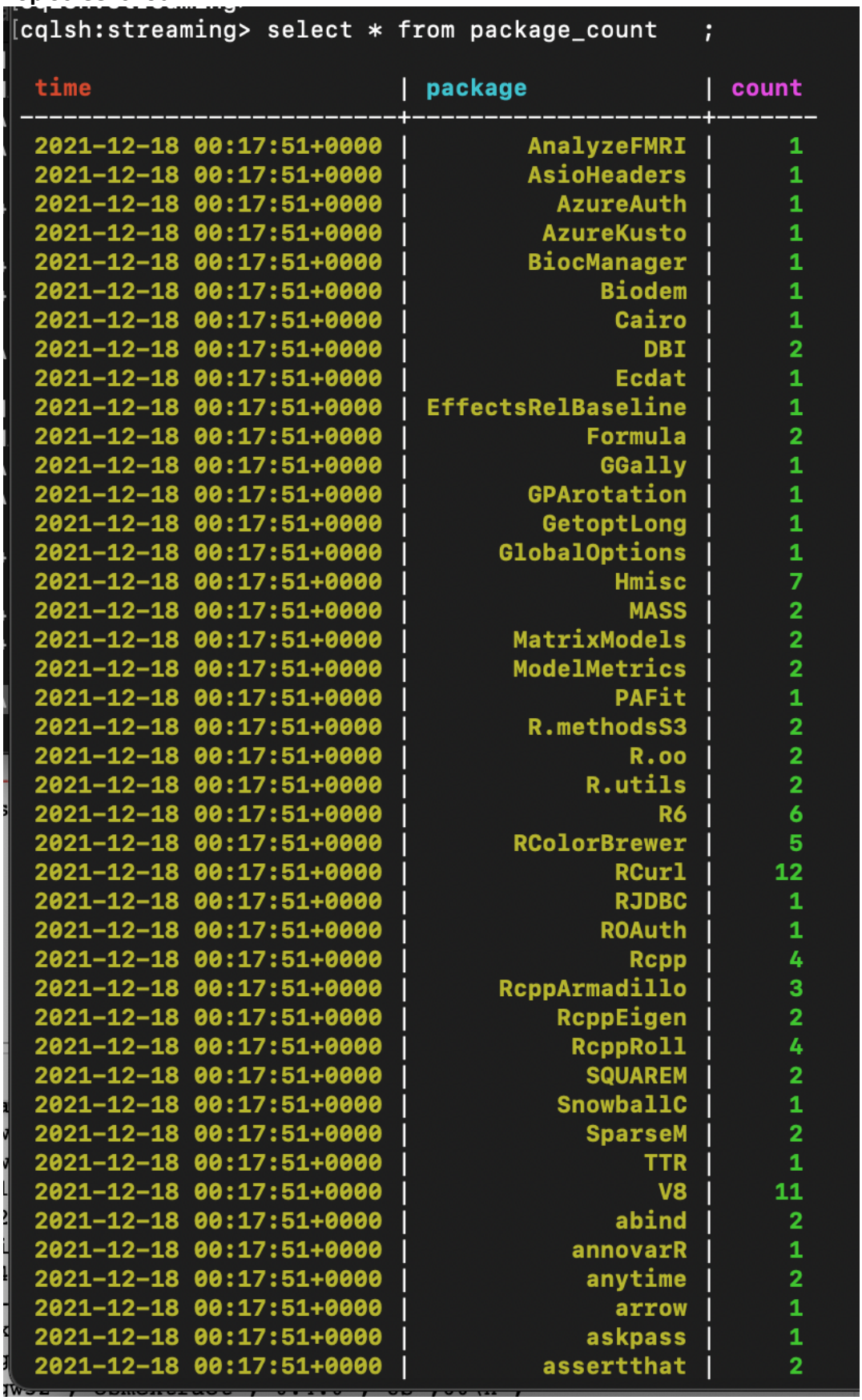
Following is the code for 4 questions together.

```
from pyspark.streaming import StreamingContext
from pyspark.sql import Row
import time
ssc = StreamingContext(sc, 3)
lines = ssc.textFileStream("file:///home/jawad/streaming/")
downloads_RDD = lines.map(lambda x: x.split(','))

downloads_RDD = downloads_RDD.map(remove_quotation)
# 1 - Package Count
package_RDD = downloads_RDD.map(lambda x: (x[6], 1))
package_RDD = package_RDD.reduceByKey(lambda a,b: a+b)

# 2 - highest downloaded package
package_RDD_reduced = package_RDD.transform(lambda package_RDD: package_RDD.sortBy(lambda a: a[1], ascending=False))

# 3 - Top 5 countries with number of downlaods
country_RDD = downloads_RDD.map(lambda x: (x[8], 1))
country_RDD = country_RDD.reduceByKey(lambda a,b: a+b)
country_RDD = country_RDD.transform(lambda downloads_RDD_reduced: downloads_RDD_reduced.sortBy(lambda x: x[1],
                                                                                               ascending=False))

# 4 - Number of downloads of ggplot2
ggplot2 = downloads_RDD.filter(lambda x: x[6]=="ggplot2")
ggplot2_mapped = ggplot2.map(lambda x: (x[6], 1))
ggplot2_reduced = ggplot2_mapped.reduceByKey(lambda a,b: a+b)



package_RDD.foreachRDD(save_package_count)
package_RDD_reduced.foreachRDD(save_top_package)
country_RDD.foreachRDD(save_country)
ggplot2_reduced.foreachRDD(save_ggplot2_count)
#package_RDD.pprint()
#package_RDD_reduced.pprint(1)
#country_RDD.pprint(5)
#ggplot2_reduced.pprint()
```

```
ssc.start()
time.sleep(122)
ssc.stop(stopSparkContext=False)
```

I've added 2 more seconds because the code to split the file is running in separate tab. I have to start that manually after starting this time which will make sure I get count from every file. I can't set it more than 122.999 because 123 may read 1 extra file as I'm performing everything after every 3 seconds.

Output:
Question#1:
Top Screenshot:



```
[cqlsh:streaming> select * from package_count   ;

 time                     | package             | count
--------------------------+---------------------+-------
 2021-12-18 00:17:51+0000 |         AnalyzeFMRI |     1
 2021-12-18 00:17:51+0000 |         AsioHeaders |     1
 2021-12-18 00:17:51+0000 |           AzureAuth |     1
 2021-12-18 00:17:51+0000 |          AzureKusto |     1
 2021-12-18 00:17:51+0000 |         BiocManager |     1
 2021-12-18 00:17:51+0000 |              Biodem |     1
 2021-12-18 00:17:51+0000 |               Cairo |     1
 2021-12-18 00:17:51+0000 |                 DBI |     2
 2021-12-18 00:17:51+0000 |               Ecdat |     1
 2021-12-18 00:17:51+0000 | EffectsRelBaseline |     1
 2021-12-18 00:17:51+0000 |             Formula |     2
 2021-12-18 00:17:51+0000 |              GGally |     1
 2021-12-18 00:17:51+0000 |         GPArotation |     1
 2021-12-18 00:17:51+0000 |          GetoptLong |     1
 2021-12-18 00:17:51+0000 |       GlobalOptions |     1
 2021-12-18 00:17:51+0000 |               Hmisc |     7
 2021-12-18 00:17:51+0000 |                MASS |     2
 2021-12-18 00:17:51+0000 |        MatrixModels |     2
 2021-12-18 00:17:51+0000 |        ModelMetrics |     2
 2021-12-18 00:17:51+0000 |               PAFit |     1
 2021-12-18 00:17:51+0000 |         R.methodsS3 |     2
 2021-12-18 00:17:51+0000 |                R.oo |     2
 2021-12-18 00:17:51+0000 |             R.utils |     2
 2021-12-18 00:17:51+0000 |                  R6 |     6
 2021-12-18 00:17:51+0000 |        RColorBrewer |     5
 2021-12-18 00:17:51+0000 |               RCurl |    12
 2021-12-18 00:17:51+0000 |               RJDBC |     1
 2021-12-18 00:17:51+0000 |              ROAuth |     1
 2021-12-18 00:17:51+0000 |                Rcpp |     4
 2021-12-18 00:17:51+0000 |       RcppArmadillo |     3
 2021-12-18 00:17:51+0000 |           RcppEigen |     2
 2021-12-18 00:17:51+0000 |            RcppRoll |     4
 2021-12-18 00:17:51+0000 |             SQUAREM |     2
 2021-12-18 00:17:51+0000 |            SnowballC |     1
 2021-12-18 00:17:51+0000 |             SparseM |     2
 2021-12-18 00:17:51+0000 |                 TTR |     1
 2021-12-18 00:17:51+0000 |                  V8 |    11
 2021-12-18 00:17:51+0000 |               abind |     2
 2021-12-18 00:17:51+0000 |            annovarR |     1
 2021-12-18 00:17:51+0000 |            anytime |     2
 2021-12-18 00:17:51+0000 |              arrow |     1
 2021-12-18 00:17:51+0000 |            askpass |     1
 2021-12-18 00:17:51+0000 |         assertthat |     2
```

End of results for question#1:

2021-12-18 00:16:48+0000	thief	1
2021-12-18 00:16:48+0000	tibble	4
2021-12-18 00:16:48+0000	tidygeocoder	1
2021-12-18 00:16:48+0000	tidyr	8
2021-12-18 00:16:48+0000	tidyselect	8
2021-12-18 00:16:48+0000	tidyverse	5
2021-12-18 00:16:48+0000	timeDate	4
2021-12-18 00:16:48+0000	tinytex	4
2021-12-18 00:16:48+0000	tm	1
2021-12-18 00:16:48+0000	tmvnsim	1
2021-12-18 00:16:48+0000	truncnorm	1
2021-12-18 00:16:48+0000	tseries	1
2021-12-18 00:16:48+0000	tweenr	1
2021-12-18 00:16:48+0000	tzdb	4
2021-12-18 00:16:48+0000	udunits2	1
2021-12-18 00:16:48+0000	units	5
2021-12-18 00:16:48+0000	usdata	2
2021-12-18 00:16:48+0000	usethis	2
2021-12-18 00:16:48+0000	utf8	9
2021-12-18 00:16:48+0000	uuid	6
2021-12-18 00:16:48+0000	vcd	1
2021-12-18 00:16:48+0000	vctrs	10
2021-12-18 00:16:48+0000	viridis	3
2021-12-18 00:16:48+0000	viridisLite	2
2021-12-18 00:16:48+0000	vroom	6
2021-12-18 00:16:48+0000	webshot	3
2021-12-18 00:16:48+0000	whisker	2
2021-12-18 00:16:48+0000	wikitaxa	1
2021-12-18 00:16:48+0000	withr	4
2021-12-18 00:16:48+0000	wordcloud	1
2021-12-18 00:16:48+0000	xfun	2
2021-12-18 00:16:48+0000	xlsxjars	1
2021-12-18 00:16:48+0000	xml2	4
2021-12-18 00:16:48+0000	xopen	1
2021-12-18 00:16:48+0000	xtable	2
2021-12-18 00:16:48+0000	xts	3
2021-12-18 00:16:48+0000	yaml	10
2021-12-18 00:16:48+0000	zeallot	2
2021-12-18 00:16:48+0000	zip	6
2021-12-18 00:16:48+0000	zoo	3

(14340 rows)

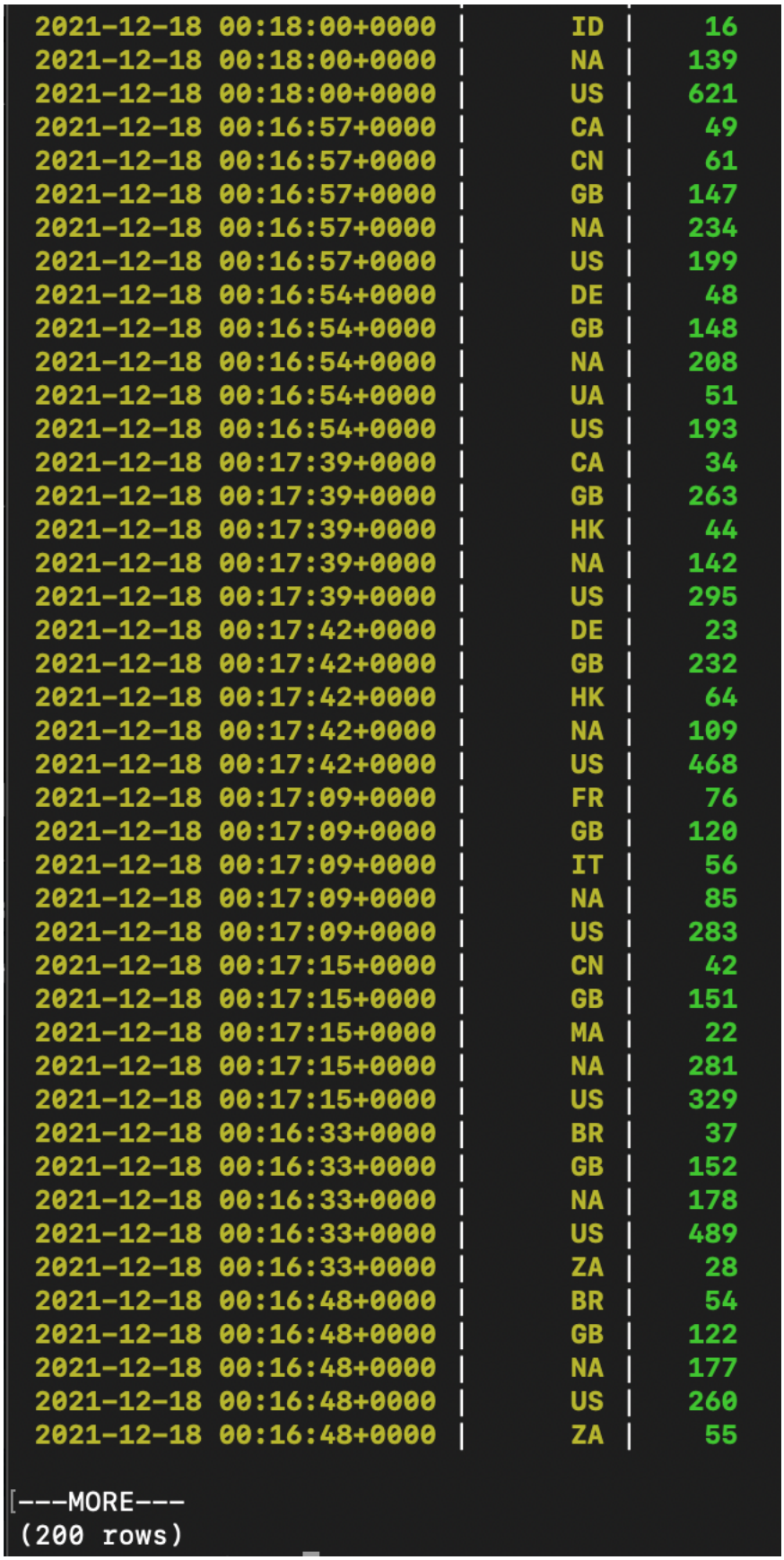
Question#2:

```
cqlsh:streaming> select * from top_package;

 time                            | package   | count
---------------------------------+-----------+-------
 2021-12-18 00:17:51+0000 | devtools |    27
 2021-12-18 00:16:36+0000 |  ggplot2 |    21
 2021-12-18 00:16:42+0000 | devtools |    23
 2021-12-18 00:17:57+0000 |  ggplot2 |    29
 2021-12-18 00:17:36+0000 |  ggplot2 |    30
 2021-12-18 00:16:12+0000 |  ggplot2 |    26
 2021-12-18 00:17:45+0000 |  ggplot2 |    30
 2021-12-18 00:16:09+0000 |  ggplot2 |    27
 2021-12-18 00:17:00+0000 |  ggplot2 |    20
 2021-12-18 00:17:12+0000 |  ggplot2 |    25
 2021-12-18 00:16:27+0000 |  ggplot2 |    17
 2021-12-18 00:17:24+0000 |  ggplot2 |    20
 2021-12-18 00:17:27+0000 |  ggplot2 |    27
 2021-12-18 00:17:18+0000 |  ggplot2 |    19
 2021-12-18 00:16:21+0000 |    rlang |    24
 2021-12-18 00:17:03+0000 |  ggplot2 |    25
 2021-12-18 00:17:54+0000 |  ggplot2 |    31
 2021-12-18 00:17:21+0000 |  ggplot2 |    17
 2021-12-18 00:17:30+0000 | devtools |    20
 2021-12-18 00:16:18+0000 | devtools |    23
 2021-12-18 00:16:45+0000 |  ggplot2 |    17
 2021-12-18 00:16:51+0000 |  ggplot2 |    28
 2021-12-18 00:17:48+0000 |  ggplot2 |    22
```

```
 2021-12-18 00:16:30+0000 |  ggplot2 |    16
 2021-12-18 00:18:06+0000 |  ggplot2 |    15
 2021-12-18 00:16:15+0000 |  ggplot2 |    30
 2021-12-18 00:17:06+0000 |       sf |    19
 2021-12-18 00:16:24+0000 |  ggplot2 |    24
 2021-12-18 00:17:33+0000 | devtools |    32
 2021-12-18 00:18:03+0000 |  ggplot2 |    17
 2021-12-18 00:16:39+0000 |  ggplot2 |    30
 2021-12-18 00:18:00+0000 |  ggplot2 |    27
 2021-12-18 00:16:57+0000 |    broom |    21
 2021-12-18 00:16:54+0000 |  ggplot2 |    22
 2021-12-18 00:17:39+0000 |  ggplot2 |    31
 2021-12-18 00:17:42+0000 |  ggplot2 |    35
 2021-12-18 00:17:09+0000 |    rlang |    19
 2021-12-18 00:17:15+0000 |  ggplot2 |    17
 2021-12-18 00:16:33+0000 |  ggplot2 |    23
 2021-12-18 00:16:48+0000 |  ggplot2 |    23

(40 rows)
cqlsh:streaming>
```

Question#3:

```
cqlsh:streaming> select * from top_country;

 time                            | country | count
---------------------------------+---------+-------
 2021-12-18 00:17:51+0000 |      AR |    32
 2021-12-18 00:17:51+0000 |      GB |   238
 2021-12-18 00:17:51+0000 |      JP |    42
 2021-12-18 00:17:51+0000 |      NA |    81
 2021-12-18 00:17:51+0000 |      US |   454
 2021-12-18 00:16:36+0000 |      BR |    48
 2021-12-18 00:16:36+0000 |      GB |   165
 2021-12-18 00:16:36+0000 |      NA |   104
 2021-12-18 00:16:36+0000 |      US |   476
 2021-12-18 00:16:36+0000 |      ZA |    48
 2021-12-18 00:16:42+0000 |      BE |    51
 2021-12-18 00:16:42+0000 |      GB |   176
 2021-12-18 00:16:42+0000 |      HK |    37
 2021-12-18 00:16:42+0000 |      NA |   117
 2021-12-18 00:16:42+0000 |      US |   429
 2021-12-18 00:17:57+0000 |      CA |    49
 2021-12-18 00:17:57+0000 |      GB |   209
 2021-12-18 00:17:57+0000 |      IT |    19
 2021-12-18 00:17:57+0000 |      NA |   147
 2021-12-18 00:17:57+0000 |      US |   448
 2021-12-18 00:17:36+0000 |      AU |   122
 2021-12-18 00:17:36+0000 |      ES |    46
 2021-12-18 00:17:36+0000 |      GB |   228
 2021-12-18 00:17:36+0000 |      NA |   103
 2021-12-18 00:17:36+0000 |      US |   286
 2021-12-18 00:16:12+0000 |      AE |    26
 2021-12-18 00:16:12+0000 |      GB |   195
 2021-12-18 00:16:12+0000 |      NA |   135
 2021-12-18 00:16:12+0000 |      PL |    27
 2021-12-18 00:16:12+0000 |      US |   457
 2021-12-18 00:17:45+0000 |      CH |    27
 2021-12-18 00:17:45+0000 |      DK |    37
 2021-12-18 00:17:45+0000 |      GB |   194
 2021-12-18 00:17:45+0000 |      NA |   158
 2021-12-18 00:17:45+0000 |      US |   425
 2021-12-18 00:16:09+0000 |      BR |    23
 2021-12-18 00:16:09+0000 |      ES |    36
 2021-12-18 00:16:09+0000 |      GB |   203
 2021-12-18 00:16:09+0000 |      NA |   128
 2021-12-18 00:16:09+0000 |      US |   457
 2021-12-18 00:17:00+0000 |      CA |    58
 2021-12-18 00:17:00+0000 |      GB |   114
 2021-12-18 00:17:00+0000 |      NA |   213
 2021-12-18 00:17:00+0000 |      NL |    49
```



2021-12-18 00:18:00+0000	ID	16
2021-12-18 00:18:00+0000	NA	139
2021-12-18 00:18:00+0000	US	621
2021-12-18 00:16:57+0000	CA	49
2021-12-18 00:16:57+0000	CN	61
2021-12-18 00:16:57+0000	GB	147
2021-12-18 00:16:57+0000	NA	234
2021-12-18 00:16:57+0000	US	199
2021-12-18 00:16:54+0000	DE	48
2021-12-18 00:16:54+0000	GB	148
2021-12-18 00:16:54+0000	NA	208
2021-12-18 00:16:54+0000	UA	51
2021-12-18 00:16:54+0000	US	193
2021-12-18 00:17:39+0000	CA	34
2021-12-18 00:17:39+0000	GB	263
2021-12-18 00:17:39+0000	HK	44
2021-12-18 00:17:39+0000	NA	142
2021-12-18 00:17:39+0000	US	295
2021-12-18 00:17:42+0000	DE	23
2021-12-18 00:17:42+0000	GB	232
2021-12-18 00:17:42+0000	HK	64
2021-12-18 00:17:42+0000	NA	109
2021-12-18 00:17:42+0000	US	468
2021-12-18 00:17:09+0000	FR	76
2021-12-18 00:17:09+0000	GB	120
2021-12-18 00:17:09+0000	IT	56
2021-12-18 00:17:09+0000	NA	85
2021-12-18 00:17:09+0000	US	283
2021-12-18 00:17:15+0000	CN	42
2021-12-18 00:17:15+0000	GB	151
2021-12-18 00:17:15+0000	MA	22
2021-12-18 00:17:15+0000	NA	281
2021-12-18 00:17:15+0000	US	329
2021-12-18 00:16:33+0000	BR	37
2021-12-18 00:16:33+0000	GB	152
2021-12-18 00:16:33+0000	NA	178
2021-12-18 00:16:33+0000	US	489
2021-12-18 00:16:33+0000	ZA	28
2021-12-18 00:16:48+0000	BR	54
2021-12-18 00:16:48+0000	GB	122
2021-12-18 00:16:48+0000	NA	177
2021-12-18 00:16:48+0000	US	260
2021-12-18 00:16:48+0000	ZA	55

[---MORE---
(200 rows)

I’m adding screenshot from the last of output to show the total number of rows as 200. Top 5* 40 files = 200 records.

Question#4:

```
cqlsh:streaming> select * from ggplot2_count ;

 time                            | ggplot2 | count
---------------------------------+---------+-------
 2021-12-18 00:17:51+0000 | ggplot2 |    25
 2021-12-18 00:16:36+0000 | ggplot2 |    21
 2021-12-18 00:16:42+0000 | ggplot2 |    18
 2021-12-18 00:17:57+0000 | ggplot2 |    29
 2021-12-18 00:17:36+0000 | ggplot2 |    30
 2021-12-18 00:16:12+0000 | ggplot2 |    26
 2021-12-18 00:17:45+0000 | ggplot2 |    30
 2021-12-18 00:16:09+0000 | ggplot2 |    27
 2021-12-18 00:17:00+0000 | ggplot2 |    20
 2021-12-18 00:17:12+0000 | ggplot2 |    25
 2021-12-18 00:16:27+0000 | ggplot2 |    17
 2021-12-18 00:17:24+0000 | ggplot2 |    20
 2021-12-18 00:17:27+0000 | ggplot2 |    27
 2021-12-18 00:17:18+0000 | ggplot2 |    19
 2021-12-18 00:16:21+0000 | ggplot2 |    21
 2021-12-18 00:17:03+0000 | ggplot2 |    25
 2021-12-18 00:17:54+0000 | ggplot2 |    31
 2021-12-18 00:17:21+0000 | ggplot2 |    17
 2021-12-18 00:17:30+0000 | ggplot2 |    19
 2021-12-18 00:16:18+0000 | ggplot2 |    13
 2021-12-18 00:16:45+0000 | ggplot2 |    17
 2021-12-18 00:16:51+0000 | ggplot2 |    28
```

```
 2021-12-18 00:17:48+0000 | ggplot2 |    22
 2021-12-18 00:16:30+0000 | ggplot2 |    16
 2021-12-18 00:18:06+0000 | ggplot2 |    15
 2021-12-18 00:16:15+0000 | ggplot2 |    30
 2021-12-18 00:17:06+0000 | ggplot2 |    19
 2021-12-18 00:16:24+0000 | ggplot2 |    24
 2021-12-18 00:17:33+0000 | ggplot2 |    32
 2021-12-18 00:18:03+0000 | ggplot2 |    17
 2021-12-18 00:16:39+0000 | ggplot2 |    30
 2021-12-18 00:18:00+0000 | ggplot2 |    27
 2021-12-18 00:16:57+0000 | ggplot2 |    15
 2021-12-18 00:16:54+0000 | ggplot2 |    22
 2021-12-18 00:17:39+0000 | ggplot2 |    31
 2021-12-18 00:17:42+0000 | ggplot2 |    35
 2021-12-18 00:17:09+0000 | ggplot2 |    15
 2021-12-18 00:17:15+0000 | ggplot2 |    17
 2021-12-18 00:16:33+0000 | ggplot2 |    23
 2021-12-18 00:16:48+0000 | ggplot2 |    23

(40 rows)
```

40 files = 40 records for question 4.