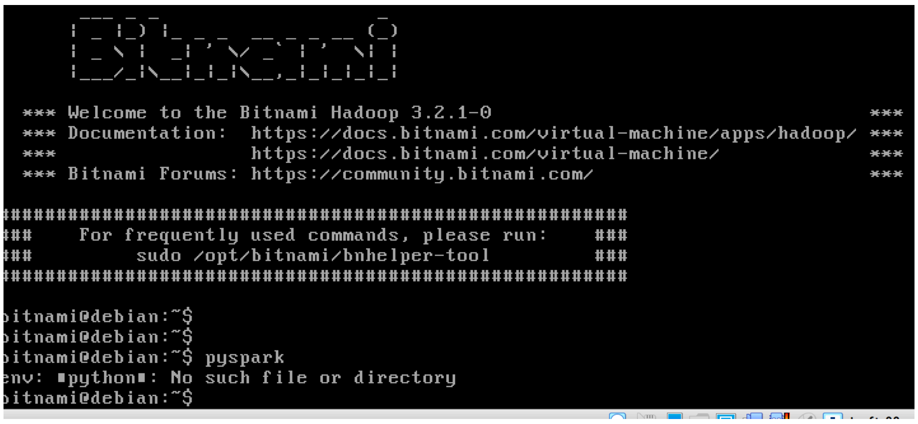
WEEK 8 ASSIGNMENT 1

Large-Scale Data Storage Systems – DATA-5400 | Spring 2020

Christina Morgenstern

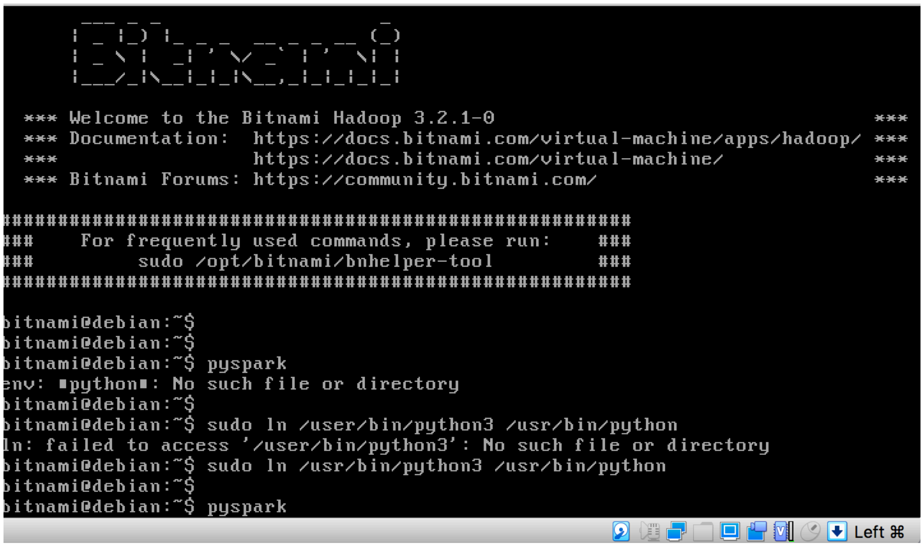
For this assignment, I am using Apache Spark on my Bitnami Hadoop Virtual Machine. For running the queries, I will be using Python, as I am most familiar with this programming language.

Start Bitnami and check for availability of the package `pyspark` which comes preinstalled with Bitnami. However, by default, `pyspark` is not found.



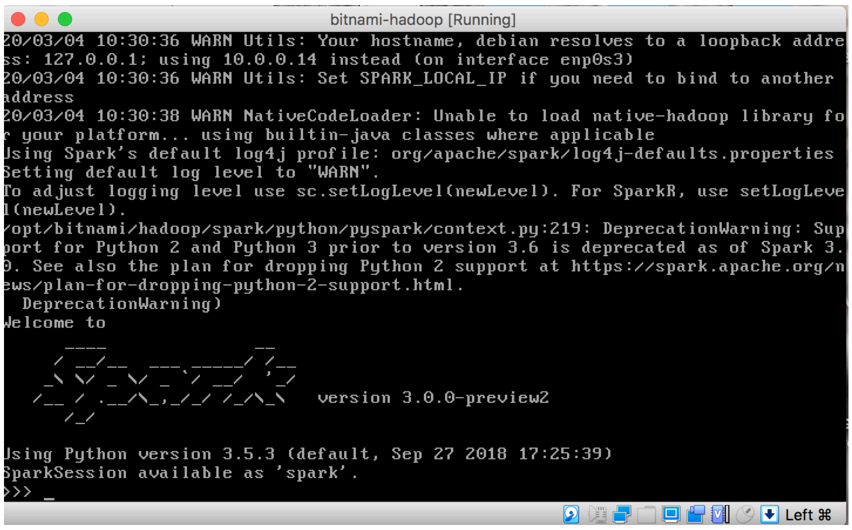
```
*** Welcome to the Bitnami Hadoop 3.2.1-0                                  ***
*** Documentation:  https://docs.bitnami.com/virtual-machine/apps/hadoop/ ***
***                 https://docs.bitnami.com/virtual-machine/             ***
*** Bitnami Forums: https://community.bitnami.com/                        ***

#######################################################
###    For frequently used commands, please run:    ###
###         sudo /opt/bitnami/bnhelper-tool         ###
#######################################################

bitnami@debian:~$
bitnami@debian:~$
bitnami@debian:~$ pyspark
env: ▪python▪: No such file or directory
bitnami@debian:~$
```

I need to create a soft-link to Python using the following command:
`sudo ln /usr/bin/python3 /usr/bin/python`



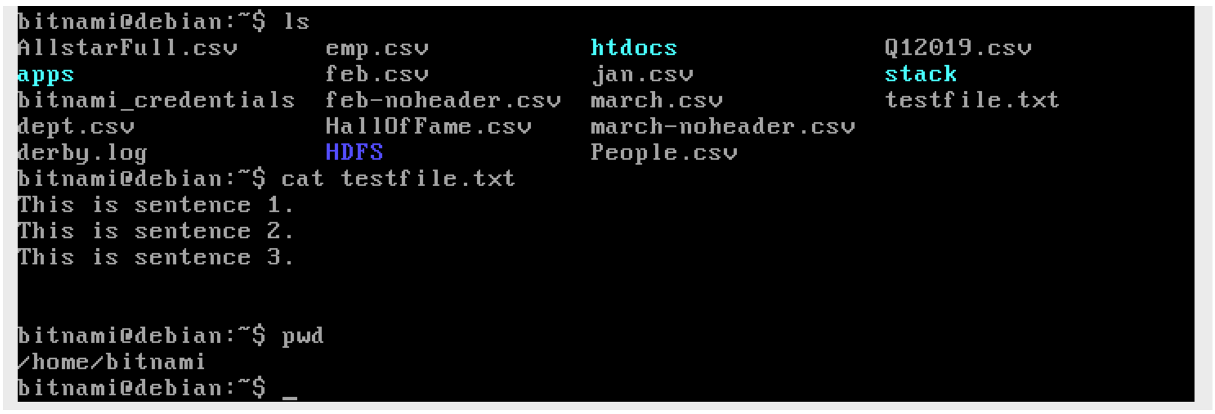
```
*** Welcome to the Bitnami Hadoop 3.2.1-0                                  ***
*** Documentation:  https://docs.bitnami.com/virtual-machine/apps/hadoop/ ***
***                 https://docs.bitnami.com/virtual-machine/             ***
*** Bitnami Forums: https://community.bitnami.com/                        ***

#######################################################
###    For frequently used commands, please run:    ###
###         sudo /opt/bitnami/bnhelper-tool         ###
#######################################################

bitnami@debian:~$
bitnami@debian:~$
bitnami@debian:~$ pyspark
env: ▪python▪: No such file or directory
bitnami@debian:~$
bitnami@debian:~$ sudo ln /user/bin/python3 /usr/bin/python
ln: failed to access '/user/bin/python3': No such file or directory
bitnami@debian:~$ sudo ln /usr/bin/python3 /usr/bin/python
bitnami@debian:~$
bitnami@debian:~$ pyspark
```

Now `pyspark` is available for use with Spark. Entering the command `pyspark` starts Spark using Python version 3.5.3.

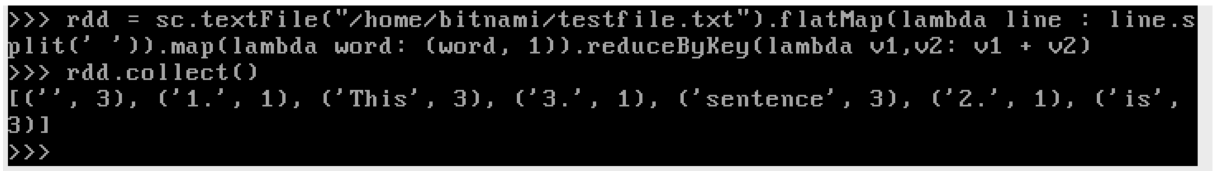

```
20/03/04 10:30:36 WARN Utils: Your hostname, debian resolves to a loopback addre
ss: 127.0.0.1; using 10.0.0.14 instead (on interface enp0s3)
20/03/04 10:30:36 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
20/03/04 10:30:38 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
Jsing Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
/opt/bitnami/hadoop/spark/python/pyspark/context.py:219: DeprecationWarning: Sup
port for Python 2 and Python 3 prior to version 3.6 is deprecated as of Spark 3.
0. See also the plan for dropping Python 2 support at https://spark.apache.org/n
ews/plan-for-dropping-python-2-support.html.
  DeprecationWarning)
Jelcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.0-preview2
      /_/

Jsing Python version 3.5.3 (default, Sep 27 2018 17:25:39)
SparkSession available as 'spark'.
>>> _
```

Part 1) Word count.

Exit Spark with command `exit()` and list files and directories on local VM in Bitnami.

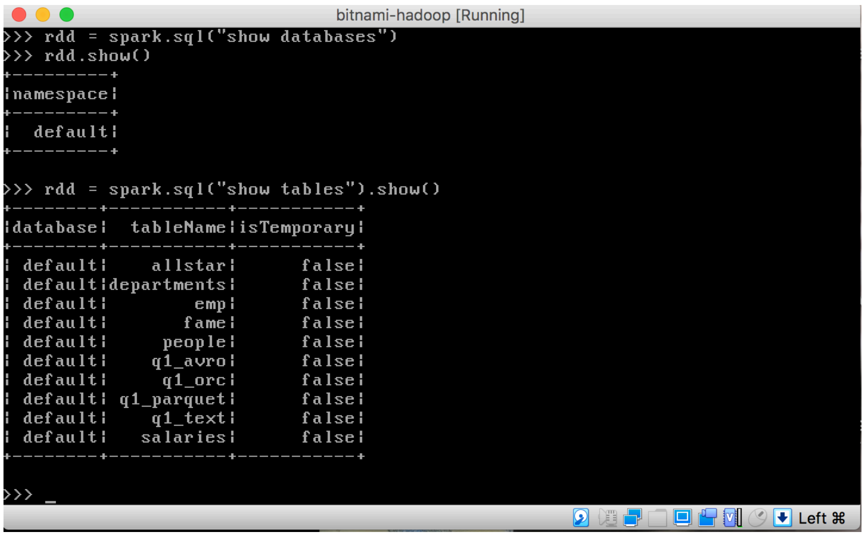

```
bitnami@debian:~$ ls
AllstarFull.csv      emp.csv            htdocs              Q12019.csv
apps                 feb.csv            jan.csv             stack
bitnami_credentials  feb-noheader.csv   march.csv           testfile.txt
dept.csv             HallOfFame.csv     march-noheader.csv
derby.log            HDFS               People.csv
bitnami@debian:~$ cat testfile.txt
This is sentence 1.
This is sentence 2.
This is sentence 3.


bitnami@debian:~$ pwd
/home/bitnami
bitnami@debian:~$ _
```
The text file `testfile.txt` previously created contains three sentences. This file will be used for the word count in Spark.

Start `pyspark` again and run word count command:


```
>>> rdd = sc.textFile("/home/bitnami/testfile.txt").flatMap(lambda line : line.s
plit(' ')).map(lambda word: (word, 1)).reduceByKey(lambda v1,v2: v1 + v2)
>>> rdd.collect()
[('', 3), ('1.', 1), ('This', 3), ('3.', 1), ('sentence', 3), ('2.', 1), ('is',
3)]
>>>
```
The first transformation retrieves the testfile.txt from the local directory.
The flatMap function splits the sentences into individual tokens based on spaces.
The Map function creates a count list for each word.
The reduceByKey function adds up similar words.

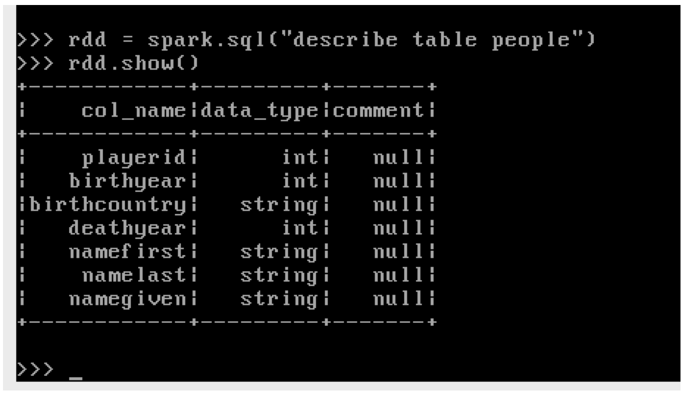
The results are as expected, the strings 'This', 'sentence' and 'is' occur each three times, the numbers '1', '2' and '3' each once. I guess (' ',3) means the period at the end of the sentences.

Part 2) Using Spark to run SQL queries on the Baseball dataset.
To run SQL commands in Spark, a variable e.g. `rdd` is created and assigned to `spark.sql` followed by the SQL command in brackets and double quotes. These SQL commands are the same as previously encountered in Hive.
`rdd = spark.sql("show tables").show()` lists all stored tables. This includes tables, that were created in earlier assignments.



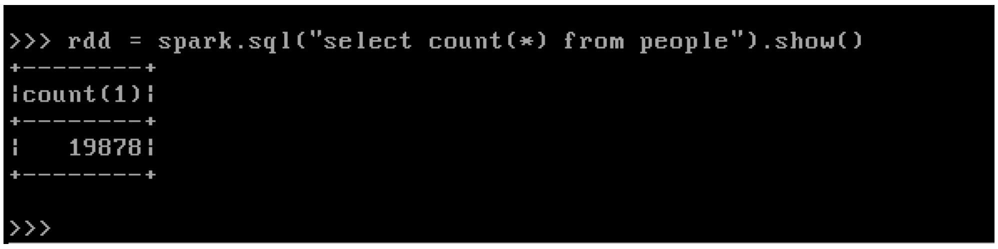
```
>>> rdd = spark.sql("show databases")
>>> rdd.show()
+---------+
|namespace|
+---------+
|  default|
+---------+

>>> rdd = spark.sql("show tables").show()
+--------+-----------+-----------+
|database|  tableName|isTemporary|
+--------+-----------+-----------+
| default|    allstar|      false|
| default|departments|      false|
| default|        emp|      false|
| default|       fame|      false|
| default|     people|      false|
| default|    q1_avro|      false|
| default|     q1_orc|      false|
| default| q1_parquet|      false|
| default|    q1_text|      false|
| default|   salaries|      false|
+--------+-----------+-----------+

>>> _
```

For the following queries, the people table is used. Use the `describe table` command to remind myself of the column names.



```
>>> rdd = spark.sql("describe table people")
>>> rdd.show()
+------------+---------+-------+
|    col_name|data_type|comment|
+------------+---------+-------+
|    playerid|      int|   null|
|   birthyear|      int|   null|
|birthcountry|   string|   null|
|   deathyear|      int|   null|
|   namefirst|   string|   null|
|    namelast|   string|   null|
|   namegiven|   string|   null|
+------------+---------+-------+

>>> _
```

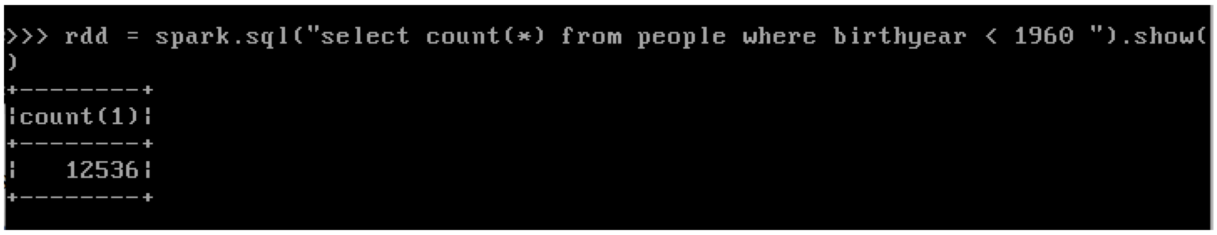
From the baseball players table, called people, the number of entries is retrieved as follows:



```
>>> rdd = spark.sql("select count(*) from people").show()
+--------+
|count(1)|
+--------+
|   19878|
+--------+

>>>
```

The total number of baseball players is 19.878. This result confirms the previous result. Although this query took slightly longer than a Hive query.

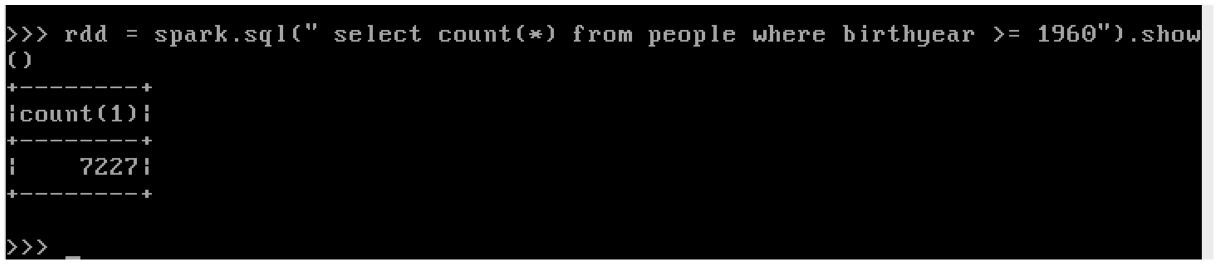
To get the number of players born before 1960, the following query was run:



```
>>> rdd = spark.sql("select count(*) from people where birthyear < 1960 ").show(
)
+--------+
|count(1)|
+--------+
|   12536|
+--------+
```

12.536 players were born before 1960.

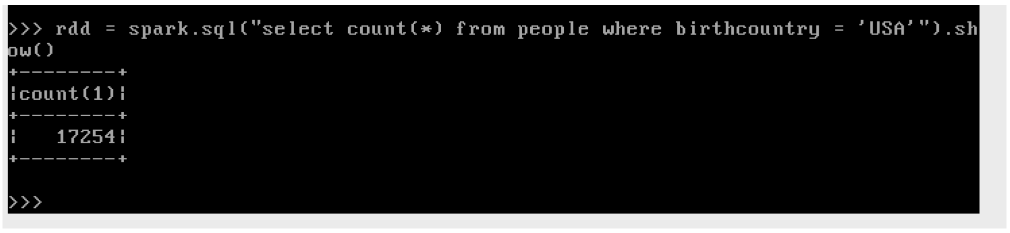
To get the number of players born in or after 1960, the following query was run:



```
>>> rdd = spark.sql(" select count(*) from people where birthyear >= 1960").show
()
+--------+
|count(1)|
+--------+
|    7227|
+--------+

>>> _
```

7.227 players were born in or after 1960.

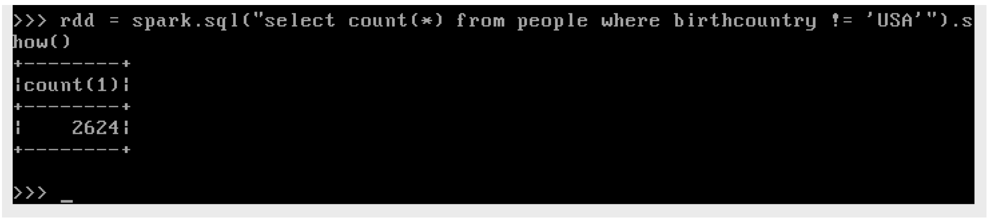
To get the number of players born in the USA use the following command. Note that for string comparison single quotes are used.



```
>>> rdd = spark.sql("select count(*) from people where birthcountry = 'USA'").sh
ow()
+--------+
|count(1)|
+--------+
|   17254|
+--------+

>>>
```

17.254 players were born in the USA.

To get the number of players born outside the USA use the following command.



```
>>> rdd = spark.sql("select count(*) from people where birthcountry != 'USA'").s
how()
+--------+
|count(1)|
+--------+
|    2624|
+--------+

>>> _
```

2.624 players were born outside the US.