Exercise 7.2 – Spark REPL

The purpose of this exercise is to practice using Spark REPL to solve simply queries.

In this exercise, you will:

- Understand how to start the Spark REPL client
- Gain familiarity with `spark.sql()` and the dataset object

Part 1: Start the Spark REPL

Starting the Spark SQL client is simple.

1. Open a terminal window.
2. Type `dse spark`
3. After a brief delay, the *spark repl* command line should appear as follows:

```
Creating a new Spark Session
Spark context Web UI available at http://10.0.2.15:4040
Spark Context available as 'sc' (master = dse://?, app id = app-
   20190312162210-0001).
Spark Session available as 'spark'.
Spark SqlContext (Deprecated use Spark Session instead) available as
   'sqlContext'
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.2.2.5
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java
   1.8.0_161)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Note that in the block it says; “Spark Session available as ‘spark’”. Basically, you have started a client application with a session variable called ‘spark’. We will use this variable in a moment to submit a query.

Part 2: Use the Spark REPL

Let’s practice using the Spark REPL.

1. Type the following Scala code into the command line and press enter:

```
val results = spark.sql("SELECT * FROM killrvideo_spark.users LIMIT
   10")
```

2. Here you used the spark session object to create a dataset called “results”. The data that the dataset delivers is based on the `SELECT` statement you submitted.
3. Now verify that you got the following result:

```
results: org.apache.spark.sql.DataFrame = [userid: string,
   created_date: timestamp ... 3 more fields]
```

4. From this result you can see that the dataset has been generated. Execute a command that will populate the dataframe and display the results:
5. Type `results.show()` and press enter. You should see the following results:

```
+--------------------+-------------------+--------------------+---------+---------+
|              userid|       created_date|               email|firstname| lastname|
+--------------------+-------------------+--------------------+---------+---------+
|6d98290e-39ec-44e...|2013-06-25 00:00:00|widdendencr@googl...|  Whitney| Iddenden|
|4cd3bf7f-52cb-470...|2012-02-09 00:00:00|htargettpc@squido...|  Harwell|  Targett|
|b73c452a-cd06-4aa...|2014-03-12 00:00:00| kgallehawkjy@ed.gov|    Karyl|Gallehawk|
|3d2711d6-2843-4a5...|2014-10-15 00:00:00| rleopardg@weibo.com|  Reynold|  Leopard|
|3577ed76-6081-43e...|2010-04-07 00:00:00|lmcelreeh5@siteme...| Leonidas|  McElree|
|1baa092f-9610-412...|2010-06-04 00:00:00|tfurmengerax@jiat...|   Tobias|Furmenger|
|7608a5a0-65b3-4e3...|2016-01-05 00:00:00|  ywarrepg@yandex.ru|     Yuri|    Warre|
|fd0ec920-2829-47f...|2013-06-02 00:00:00|rjirieckr6@bibleg...|   Rossie|  Jirieck|
|e198a897-869d-4fa...|2015-01-27 00:00:00| fleemingcr@narod.ru|  Filbert|  Leeming|
|c0590cee-0db2-4fe...|2013-08-13 00:00:00|dmulvaneybp@youtu.be|   Darbie| Mulvaney|
+--------------------+-------------------+--------------------+---------+---------+
```

Part 3: Writing the Queries

Submit the queries you wrote in the last exercise using `spark.sql()` and compare the results to those you got in the previous exercise.

Note that sometimes the Scala REPL struggles with interpreting multi-line statements. If you have this issue, type `:paste` at the command line and then then hit `Enter`. This puts the REPL into paste mode. From there, paste your multi-line expression at the command prompt. Then, put your cursor on the next line and hit `CTRL-D`. Your statement will then be interpreted.