

BDA Lab 12

(Spark, MongoDB, mogodb-spark-connector, RDD's, CSV, Transformations)

The goal of today's lab is to load csv data in spark RDD's apply some transformations and save the transformed data into MongoDB Database.

1. Get an overview from: <https://www.mongodb.com/blog/post/getting-started-with-mongodb-pyspark-and-jupyter-notebook>
2. Install pyspark
3. Install **MongoDB Spark Connector** from: <https://github.com/mongodb/mongo-spark#downloading>
4. Build the connector as mentioned in github documentation
5. Run mongo db
6. Create a document "YourName" (database in MySQL term)
7. Create a collection "Rollnumber" (table in MySQL term)
8. At this point it's empty
9. Now create a Jupyter notebook
10. Follow the same process as shown in lab demo

Debugging:

If memory issue faced during build process:

<https://stackoverflow.com/questions/54682907/on-scala-project-getting-error-gc-overhead-limit-exceeded-when-running-sbt-test>

Requirements:

```
21/04/18 18:46:49 INFO ContextHandler: Started o.s.j.s.ServletContextHandler@6371f0aa{/SQL,null,AVAILABLE,@Spark}
21/04/18 18:46:49 INFO ContextHandler: Started o.s.j.s.ServletContextHandler@358c1b7d{/SQL/json,null,AVAILABLE,@Spark}
21/04/18 18:46:49 INFO ContextHandler: Started o.s.j.s.ServletContextHandler@2f1e6e58{/SQL/execution,null,AVAILABLE,@Spark}
21/04/18 18:46:49 INFO ContextHandler: Started o.s.j.s.ServletContextHandler@1cbf6927{/SQL/execution/json,null,AVAILABLE,@Spark}
21/04/18 18:46:49 INFO ContextHandler: Started o.s.j.s.ServletContextHandler@6802e9ac{/static/sql,null,AVAILABLE,@Spark}
21/04/18 18:46:49 INFO MongoRelation: requiredColumns: _id, arrayInt, binary, boolean, code, codeWithScope, date, dbPointer, document, double, int32, int64, maxKey, minKey, null, objectId, oldBinary, regex, string, symbol, timestamp, undefined, filters:
21/04/18 18:46:49 INFO AbstractConnector: Stopped Spark865548568(HTTP/1.1,[http://1.1]:0.0.0.0:4842)
[info] Test run started
[info] Test com.mongodb.spark.sql.MongoDataFrameTest.shouldRoundTripAllBsonTypes started
[info] Test run finished: 0 failed, 0 ignored, 1 total, 0.64s
21/04/18 18:46:50 INFO MapFunctionsSpec: Ended Test: 'rowToDocument should handle null values in maps if MapType.valueContainsNull'
[info] MapFunctionsSpec:
[info] documentToRow
[info] - should convert a Document into a Row with the given schema
[info] - should not prune the schema when given a document with missing values
[info] - should prune the schema when limited by passed required columns
[info] - should ignore any extra data in the document that is not included in the schema
[info] - should handle nested schemas
[info] - should handle schemas containing maps
[info] - should throw an exception when passed maps without string keys
[info] rowToDocument
[info] - should convert a Row into a Document
[info] - should handle nested schemas
[info] - should handle nested schemas within nested arrays
[info] - should handle converting mongo types to bson correctly
[info] - should handle converting complex structs with mongo types to bson correctly
[info] - should handle mixed numerics based on the schema
[info] - should throw a MongoTypeConversionException when casting to an invalid DataType
[info] - should handle null values if field.nullable
[info] - should handle null values in arrays if ArrayType.containsNull
[info] - should handle null values in maps if MapType.valueContainsNull
[info] ScalaTest
[info] Run completed in 1 minute, 29 seconds.
[info] Total number of tests run: 196
[info] Suites: completed 25, aborted 0
[info] Tests: succeeded 196, failed 0, canceled 7, ignored 0, pending 0
[info] All tests passed.
[info] Passed: Total 234, Failed 0, Errors 0, Passed 234, Canceled 7
[success] Total time: 1917 s, completed Apr 18, 2021 6:46:51 PM
[info] Aggregating coverage from subprojects...
[info] Found 0 subproject report files []
[info] No subproject data to aggregate, skipping reports
[success] Total time: 0 s, completed Apr 18, 2021 6:46:51 PM
[info] Waiting for measurement data to sync...
[info] Reading coverage instrumentation [/Users/saadnaeem/Downloads/mongo-spark-master/target/scale-2.12/coverage-data/scoverage.coverage.xml]
[info] Reading coverage measurements...
[info] Generating coverage reports...
[info] Written Cobertura report [/Users/saadnaeem/Downloads/mongo-spark-master/target/scale-2.12/coverage-report/cobertura.xml]
[info] Written XML coverage report [/Users/saadnaeem/Downloads/mongo-spark-master/target/scale-2.12/coverage-report/scoverage.xml]
[info] Written HTML coverage report [/Users/saadnaeem/Downloads/mongo-spark-master/target/scale-2.12/coverage-report/index.html]
[info] Statement coverage: 88.82%
[info] Branch coverage: 76.42%
[info] Coverage reports completed
[info] All done. Coverage was [88.82%]
[success] Total time: 3 s, completed Apr 18, 2021 6:46:54 PM
21/04/18 18:46:54 INFO MongoClientCache: Closing MongoClient: [localhost:27017]
SaadNaeem@Macbook-Pro ~/Downloads/mongo-spark-master <master>
```

1: Must show that the build was successful with terminal username

```
In [1]: 1 from pyspark.sql import SparkSession
        2 from pyspark.sql import functions as F
        3 import os
```

```
In [2]: 1 Saad_Spark_Context = SparkSession \
        2     .builder \
        3     .appName("SaadNaeem") \
        4     .config("spark.mongodb.input.uri", "mongodb://127.0.0.1:27017/SaadNaeem.i19-1207") \
        5     .config("spark.mongodb.output.uri", "mongodb://127.0.0.1:27017/SaadNaeem.i19-1207") \
        6     .config('spark.jars.packages', 'org.mongodb.spark:mongo-spark-connector_2.12:3.0.0') \
        7     .getOrCreate()
```

```
In [3]: 1 Saad_Spark_Context
```

```
Out[3]: SparkSession - in-memory
SparkContext
```

```
Spark UI
Version
v3.1.1
Master
local[*]
AppName
SaadNaeem
```

2: Proof that your connection was successful using spark-mongodb-connector and the AppName = "yourName"

```

In [22]: 1 # Load Back
          2 new_df = Saad_Spark_Context.read.format("mongo").load()

In [23]: 1 new_df.head(5)

Out[23]: [Row(_id=Row(oid='607c468ec428fc462ebaa3a2'), fname='Musa', lname='DC', name='Musa, DC'),
          Row(_id=Row(oid='607c468ec428fc462ebaa3a3'), fname='Ahmed', lname='Wadood', name='Ahmed, Wadood'),
          Row(_id=Row(oid='607c468ec428fc462ebaa3a4'), fname='Ali', lname='Kamal', name='Ali, Kamal'),
          Row(_id=Row(oid='607c468ec428fc462ebaa3a5'), fname='Shayan', lname='Mansoor', name='Shayan, Mansoor'),
          Row(_id=Row(oid='607c468ec428fc462ebaa3a6'), fname='Mahreen', lname='Athar', name='Mahreen, Athar')]

In [25]: 1 # Good to stop SparkSession at the end of the application
          2 Saad_Spark_Context.stop()

```

3: From Notebook show that you are able to successfully read the data from mongo db using spark context

```

> show dbs
LabDatabase  0.001GB
SaadNaeem   0.000GB
admin        0.000GB
config       0.000GB
local        0.000GB
pyspark      0.000GB
> use SaadNaeem
switched to db SaadNaeem
> db["i19-1207"].find()
{ "_id" : ObjectId("607c468ec428fc462ebaa3a2"), "lname" : "DC", "fname" : "Musa", "name" : "Musa, DC" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a3"), "lname" : "Wadood", "fname" : "Ahmed", "name" : "Ahmed, Wadood" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a4"), "lname" : "Kamal", "fname" : "Ali", "name" : "Ali, Kamal" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a5"), "lname" : "Mansoor", "fname" : "Shayan", "name" : "Shayan, Mansoor" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a6"), "lname" : "Athar", "fname" : "Mahreen", "name" : "Mahreen, Athar" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a7"), "lname" : "Rahool", "fname" : "Saifullah", "name" : "Saifullah, Rahool" }
{ "_id" : ObjectId("607c468ec428fc462ebaa3a8"), "lname" : "Ahmad", "fname" : "Arsal", "name" : "Arsal, Ahmad" }
> db.dropDatabase()
{ "dropped" : "SaadNaeem", "ok" : 1 }
>

```

4: Finally, go to the terminal and run all the commands in sequence which also shows that spark Correctly transformed values and loaded them in the database