# 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.

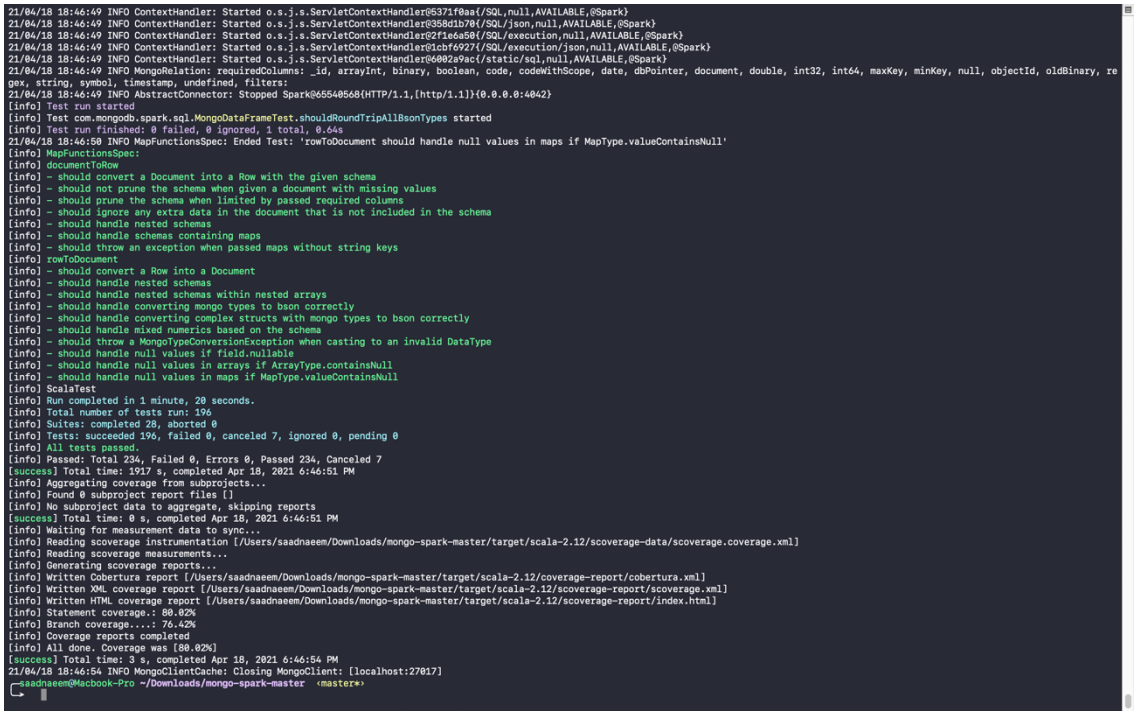
1. Get an overview from: https://www.mongodb.com/blog/post/getting-started-with-mongodb-pyspark-and-jupyter-notebook
2. Install pyspark
3. Install **MongoDB Spark Connector** from: https://github.com/mongodb/mongo-spark#downloading
4. Build the connector as mentioned in github documentation
5. Run mongo 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-tes

# Requirements:




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

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

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

In [3]:
```
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*