**Spark Core - Task 2**

**SPARK:**

* It is a open source **cluster-computing framework** designed for speed & ease of use. It's well known for in memory performace.
* It's highly accessible for offering view api for scala, java, python, r & sql. It has integrated libraries for ML, SQl, streaming, etc.
* **100x faster** **in memory** than Mapreduce.
* **10x faster in disk** than Mapreduce.
* Spark does not have it's own distributed file system but can **use hdfs**.

**#. Find the most popular country where hotels are booked and searched from the same country. Implement using scala or python. Create a separate application. Copy the application to the archive. Make screenshots of results: before and after execution.**

val df = fileData.filter("user\_location\_country==hotel\_country AND is\_booking==1").groupBy("hotel\_country").count()

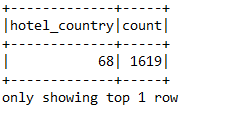
df.sort(desc("count")).show(1)

In the above query, I have applied filter on the basis of user location country and hotel country.

Then group by result on the basis of hotel country to calculate count. Then showing only one result.

Note: I have used desc in place of agg() to speed up the query. Because two agg() and 1 join() is needed to provide the output same output.

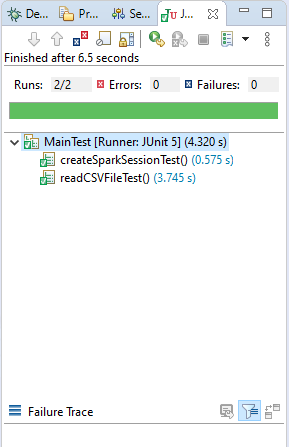
**Output:**



In the above o/p, you can see the most popular hotel country and maximum counts.

**Test Cases:**

I have also write test cases to verify train.csv schema and session variables.



**Mentee's primary skill is not software developer:**

To provide this feature application first ask to user to file path means file path is not hardcoded.

So, easily you can the o/p. Also, I have written the comments on the top of all files to which will help to understand project.