**Spark Core - Task 3**

**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 top 3 hotels where people with children are interested but not booked in the end. Implement using scala or python. Create a separate application. Copy the application to the archive. Make screenshots of results: before and after execution.**

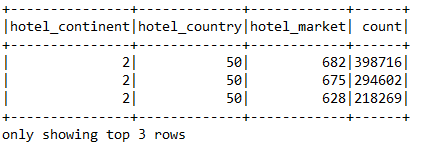
val df = fileData.filter("srch\_adults\_cnt>=1 AND srch\_children\_cnt >= 1 AND is\_booking==0").groupBy("hotel\_continent", "hotel\_country", "hotel\_market").count()

df.sort(desc("count")).show(3)

In the above query, I have applied filter on the basis (people and children) adult and children. So, at least one adult and one children should be present.

Then group by on the basis of hotel\_continent, hotel\_country, hotel\_market to calculate count. Then showing top 3 results.

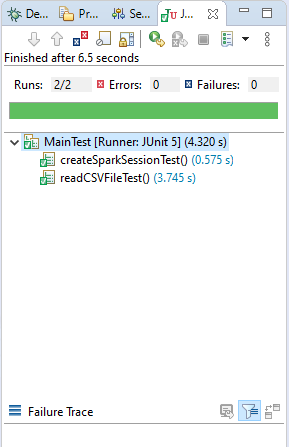
**Output:**



In the above o/p, you can see hotel identification composite keys: hotel\_continent, hotel\_country, hotel\_market. And on the basis of count top 3 values are display.

**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.