**Spark Core - Task 1**

**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 most popular hotels between couples. (treat hotel as composite key of continent, country and market). Implement using scala or python. Create a separate application. Copy the application to the archive. Make screenshots of results: before and after execution.**

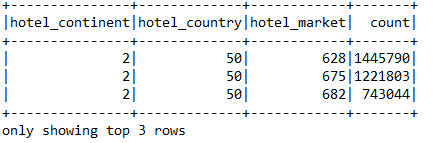
val df = fileData.filter("srch\_adults\_cnt>=2").groupBy("hotel\_continent", "hotel\_country", "hotel\_market").count()

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

In the above query, I have assumed that the adults count is greater than or equal to 2 are couples.

I have filtered data on the basis of couples and then group by hotel location. On the the basis of count I have displayed top 3 hotels.

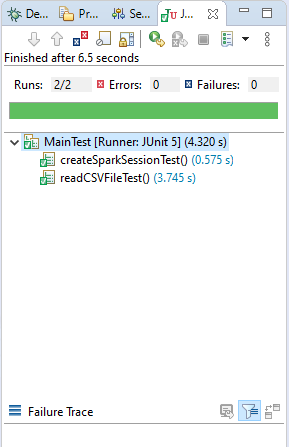
**Output:**



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.