# IDS 561 Analytics for Big Data

## HOMEWORK 3

a.  Use Hadoop fs -ls command (terminal command, not Hive command) to check what you have in the directory of Table 4. You should see something like the following screenshot (you can see the filter folders only).
    We use the command, **hadoop fs -ls /user/hive/warehouse/hivehomework.db/table4**



b.  Use the same command to check filter=Normal folder under table 4. Save the screenshot in the PDF file. You should see something like the following
    We use the command,
    **hadoop fs -ls /user/hive/warehouse/hivehomework.db/table4/filter=Normal**



c.  Two screenshots of the results from step 5 and 6.
    We execute the following queries to arrive at the results,

SELECT filter, count(filter) as Count FROM hiveHomeWork.table1 WHERE filter IN ('Clarendon','Gingham','Reyes') GROUP BY filter ORDER BY Count DESC;

**Time taken: 69.108 Seconds running from Table 1**

```
MapReduce Total cumulative CPU time: 2 seconds 320 msec
Ended Job = job_1507685636278_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 2   Cumulative CPU: 11.47 sec    HDFS Read: 360713933 HDFS Write: 277
SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 2.32 sec    HDFS Read: 5494 HDFS Write: 43 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 790 msec
OK
Clarendon       114401
Gingham 41575
Reyes   26897
Time taken: 69.108 seconds, Fetched: 3 row(s)
hive> ▉
```

SELECT filter, count(filter) as Count FROM hiveHomeWork.table4 WHERE filter IN ('Clarendon','Gingham','Reyes') GROUP BY filter ORDER BY Count DESC;

**Time taken: 51.068 Seconds running from Table 4**

```
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.89 sec    HDFS Read: 11948802 HDFS Write: 181 SU
CCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 2.2 sec   HDFS Read: 5128 HDFS Write: 43 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 90 msec
OK
Clarendon       114401
Gingham 41575
Reyes   26897
Time taken: 51.068 seconds, Fetched: 3 row(s)
hive> ▉
```

**Table 4 (51.068 seconds)** with Partitioning and Bucketing yields a comparatively better result compared to **Table 1 (69.108 seconds)** as it has a comparatively Shorter Execution Time.