Assignment 8.3:

Table of Contents

Problem Statement:

Link: https://acadgild.com/blog/transactions-in-hive/

Refer the above given link for transactions in Hive and implement the operations given in the blog using your own sample data set and send us the screenshot.

Steps:

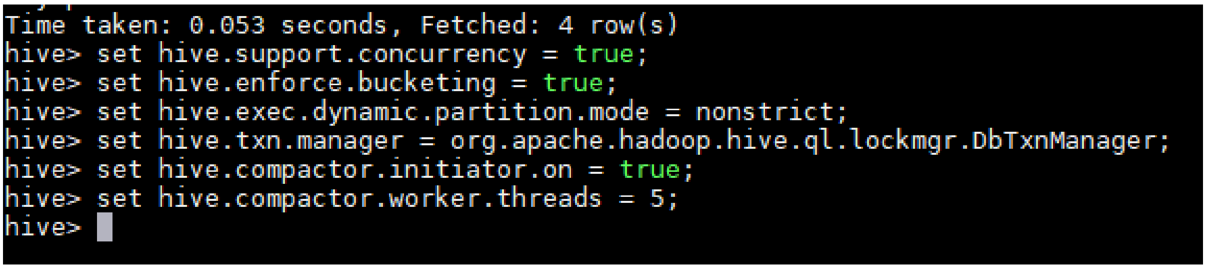
Transactions are provided at the row-level in Hive 0.14. The different row-level transactions available in Hive 0.14 are as follows:

- Insert
- Delete
- Update

There are numerous limitations with the present transactions available in Hive 0.14. ORC is the file format supported by Hive transaction. It is now essential to have ORC file format for performing transactions in Hive. The table needs to be bucketed in order to support transactions.

Setting Properties for Row-level Transactions:-

The below properties needs to be set appropriately in hive shell , order-wise to work with transactions in Hive:-



```
Time taken: 0.053 seconds, Fetched: 4 row(s)
hive> set hive.support.concurrency = true;
hive> set hive.enforce.bucketing = true;
hive> set hive.exec.dynamic.partition.mode = nonstrict;
hive> set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
hive> set hive.compactor.initiator.on = true;
hive> set hive.compactor.worker.threads = 5;
hive>
```

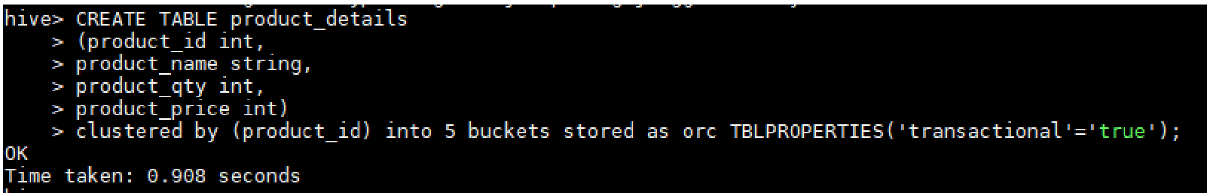
Create table in hive and insert data:

CREATE TABLE product_details

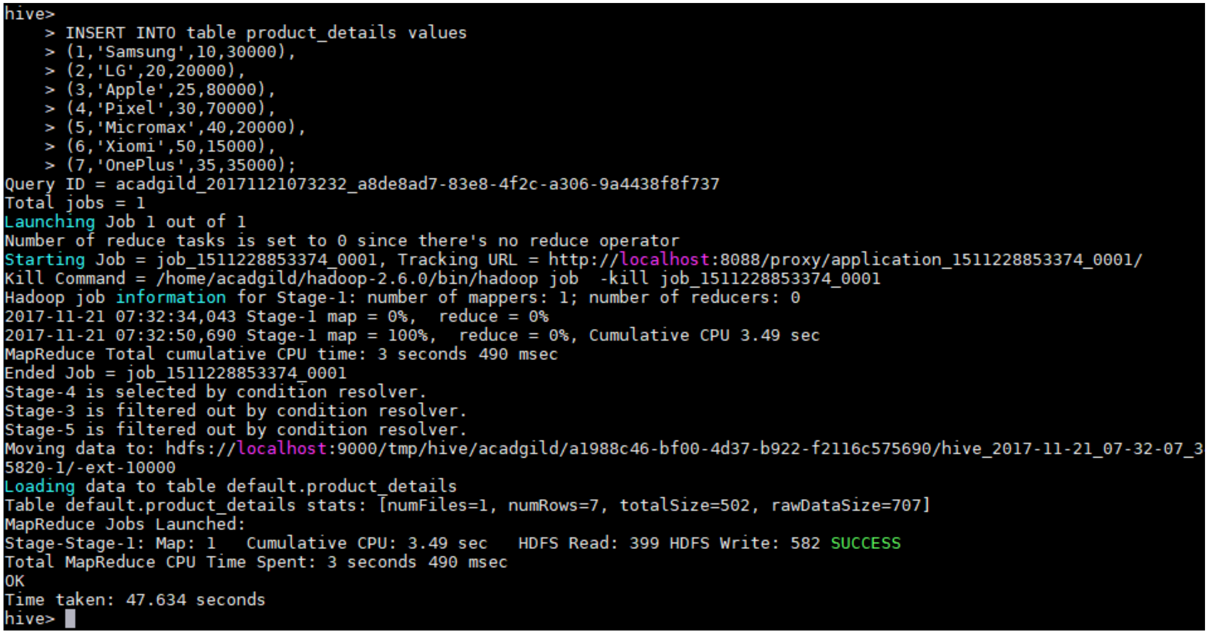
(product_id int,

product_name string,
product_qty int,
product_price int)
clustered by (product_id) into 5 buckets stored as orc
TBLPROPERTIES('transactional'='true');

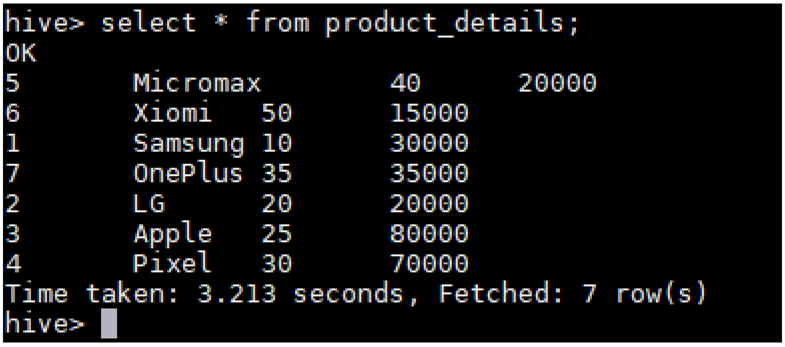
INSERT INTO table product_details values
(1,'Samsung',10,30000),
(2,'LG',20,20000),
(3,'Apple',25,80000),
(4,'Pixel',30,70000),
(5,'Micromax',40,20000),
(6,'Xiomi',50,15000),
(7,'OnePlus',35,35000);



```
hive> CREATE TABLE product_details
    > (product_id int,
    > product_name string,
    > product_qty int,
    > product_price int)
    > clustered by (product_id) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true');
OK
Time taken: 0.908 seconds
```



```
hive>
    > INSERT INTO table product_details values
    > (1,'Samsung',10,30000),
    > (2,'LG',20,20000),
    > (3,'Apple',25,80000),
    > (4,'Pixel',30,70000),
    > (5,'Micromax',40,20000),
    > (6,'Xiomi',50,15000),
    > (7,'OnePlus',35,35000);
Query ID = acadgild_20171121073232_a8de8ad7-83e8-4f2c-a306-9a4438f8f737
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1511228853374_0001, Tracking URL = http://localhost:8088/proxy/application_1511228853374_0001/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1511228853374_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-11-21 07:32:34,043 Stage-1 map = 0%,  reduce = 0%
2017-11-21 07:32:50,690 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.49 sec
MapReduce Total cumulative CPU time: 3 seconds 490 msec
Ended Job = job_1511228853374_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://localhost:9000/tmp/hive/acadgild/a1988c46-bf00-4d37-b922-f2116c575690/hive_2017-11-21_07-32-07_3
5820-1/-ext-10000
Loading data to table default.product_details
Table default.product_details stats: [numFiles=1, numRows=7, totalSize=502, rawDataSize=707]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.49 sec   HDFS Read: 399 HDFS Write: 582 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 490 msec
OK
Time taken: 47.634 seconds
hive>
```

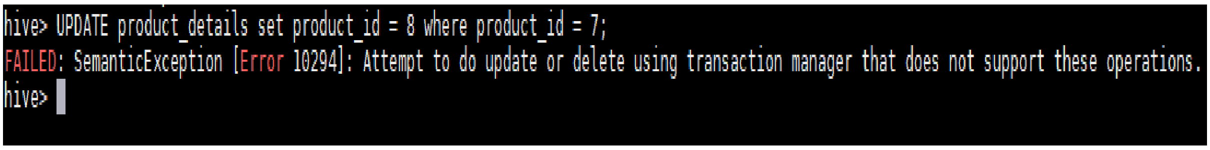


```
hive> select * from product_details;
OK
5       Micromax        40      20000
6       Xiomi   50      15000
1       Samsung 10      30000
7       OnePlus 35      35000
2       LG      20      20000
3       Apple   25      80000
4       Pixel   30      70000
Time taken: 3.213 seconds, Fetched: 7 row(s)
hive>
```

Updating the Data in Hive Table

UPDATE product_details set product_id = 8 where product_id = 7;

The above command is used to update a row in Hive table.



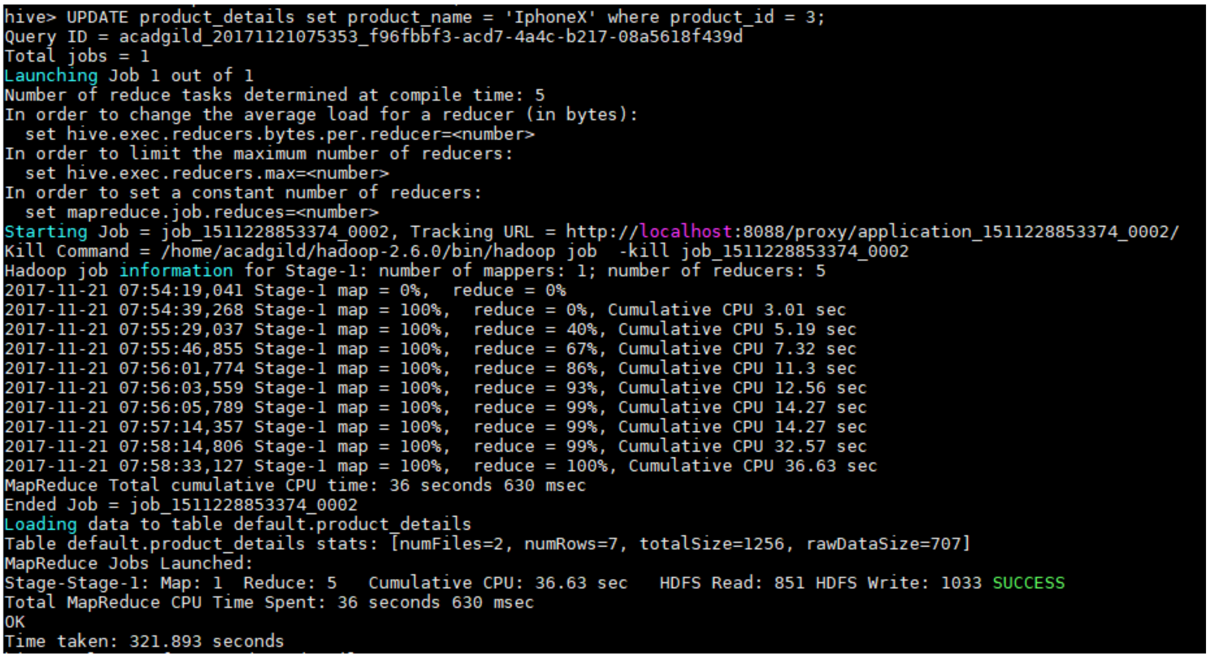
```
hive> UPDATE product_details set product_id = 8 where product_id = 7;
FAILED: SemanticException [Error 10294]: Attempt to do update or delete using transaction manager that does not support these operations.
hive>
```

From the above image, we can see that we have received an error message. This means that the Update command is not supported on the columns that are bucketed.

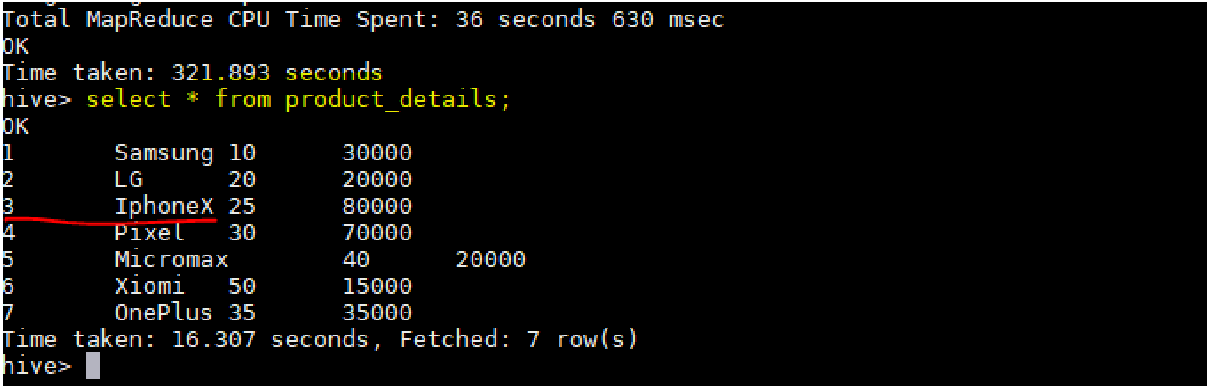
In this table, we have bucketed the ‘product_id’ column and performing the Update operation on the same column, so we have go the error

FAILED: SemanticException[Error 10302]: Updating values of bucketing columns is not supported. Column clg_id

Now let’s perform the update operation on Non bucketed column:
UPDATE product_details set product_name = 'IphoneX' where product_id = 3;



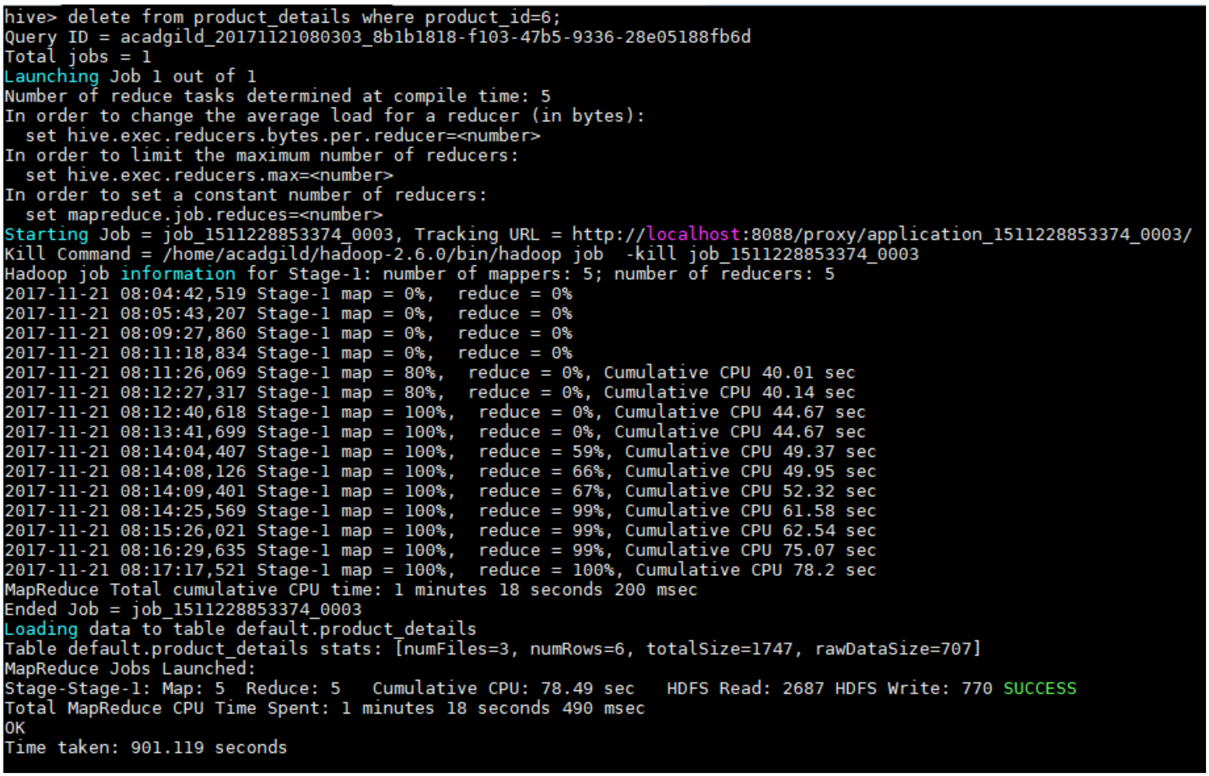
```
hive> UPDATE product_details set product_name = 'IphoneX' where product_id = 3;
Query ID = acadgild_20171121075353_f96fbbf3-acd7-4a4c-b217-08a5618f439d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1511228853374_0002, Tracking URL = http://localhost:8088/proxy/application_1511228853374_0002/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1511228853374_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 5
2017-11-21 07:54:19,041 Stage-1 map = 0%,  reduce = 0%
2017-11-21 07:54:39,268 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.01 sec
2017-11-21 07:55:29,037 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 5.19 sec
2017-11-21 07:55:46,855 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 7.32 sec
2017-11-21 07:56:01,774 Stage-1 map = 100%,  reduce = 86%, Cumulative CPU 11.3 sec
2017-11-21 07:56:03,559 Stage-1 map = 100%,  reduce = 93%, Cumulative CPU 12.56 sec
2017-11-21 07:56:05,789 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 14.27 sec
2017-11-21 07:57:14,357 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 14.27 sec
2017-11-21 07:58:14,806 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 32.57 sec
2017-11-21 07:58:33,127 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 36.63 sec
MapReduce Total cumulative CPU time: 36 seconds 630 msec
Ended Job = job_1511228853374_0002
Loading data to table default.product_details
Table default.product_details stats: [numFiles=2, numRows=7, totalSize=1256, rawDataSize=707]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 5   Cumulative CPU: 36.63 sec   HDFS Read: 851 HDFS Write: 1033 SUCCESS
Total MapReduce CPU Time Spent: 36 seconds 630 msec
OK
Time taken: 321.893 seconds
```



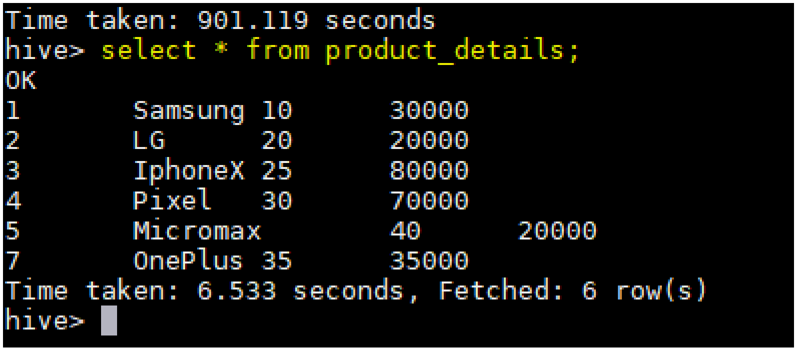
```
Total MapReduce CPU Time Spent: 36 seconds 630 msec
OK
Time taken: 321.893 seconds
hive> select * from product_details;
OK
1       Samsung 10      30000
2       LG      20      20000
3       IphoneX 25      80000
4       Pixel   30      70000
5       Micromax        40      20000
6       Xiomi   50      15000
7       OnePlus 35      35000
Time taken: 16.307 seconds, Fetched: 7 row(s)
hive>
```

Deleting a Row from Hive Table

delete from product_details where product_id=6;



```
hive> delete from product_details where product_id=6;
Query ID = acadgild_20171121080303_8b1b1818-f103-47b5-9336-28e05188fb6d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1511228853374_0003, Tracking URL = http://localhost:8088/proxy/application_1511228853374_0003/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1511228853374_0003
Hadoop job information for Stage-1: number of mappers: 5; number of reducers: 5
2017-11-21 08:04:42,519 Stage-1 map = 0%,  reduce = 0%
2017-11-21 08:05:43,207 Stage-1 map = 0%,  reduce = 0%
2017-11-21 08:09:27,860 Stage-1 map = 0%,  reduce = 0%
2017-11-21 08:11:18,834 Stage-1 map = 0%,  reduce = 0%
2017-11-21 08:11:26,069 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 40.01 sec
2017-11-21 08:12:27,317 Stage-1 map = 80%,  reduce = 0%, Cumulative CPU 40.14 sec
2017-11-21 08:12:40,618 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 44.67 sec
2017-11-21 08:13:41,699 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 44.67 sec
2017-11-21 08:14:04,407 Stage-1 map = 100%,  reduce = 59%, Cumulative CPU 49.37 sec
2017-11-21 08:14:08,126 Stage-1 map = 100%,  reduce = 66%, Cumulative CPU 49.95 sec
2017-11-21 08:14:09,401 Stage-1 map = 100%,  reduce = 67%, Cumulative CPU 52.32 sec
2017-11-21 08:14:25,569 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 61.58 sec
2017-11-21 08:15:26,021 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 62.54 sec
2017-11-21 08:16:29,635 Stage-1 map = 100%,  reduce = 99%, Cumulative CPU 75.07 sec
2017-11-21 08:17:17,521 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 78.2 sec
MapReduce Total cumulative CPU time: 1 minutes 18 seconds 200 msec
Ended Job = job_1511228853374_0003
Loading data to table default.product_details
Table default.product_details stats: [numFiles=3, numRows=6, totalSize=1747, rawDataSize=707]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 5  Reduce: 5   Cumulative CPU: 78.49 sec   HDFS Read: 2687 HDFS Write: 770 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 18 seconds 490 msec
OK
Time taken: 901.119 seconds
```



```
Time taken: 901.119 seconds
hive> select * from product_details;
OK
1       Samsung 10      30000
2       LG      20      20000
3       IphoneX 25      80000
4       Pixel   30      70000
5       Micromax        40      20000
7       OnePlus 35      35000
Time taken: 6.533 seconds, Fetched: 6 row(s)
hive>
```