readme.md

> Note : these steps are for after having installed hive with a schema metadata manager like derby of nosql

Details :

- Name : P K Navin Shrinivas
- SRN : PES2UG20CS237
- SECTION : D

Steps :

- First up we need to create the schema :

```
create table netflix(
    show_id String,
    type String,
    title String,
    director String,
    country String,
    release_year int,
    primary key (show_id) disable novalidate
        );
```

- Next up download the csv file like so :

```
wget https://raw.githubusercontent.com/Cloud-Computing-Big-Data/UE20CS322-H2/m
```

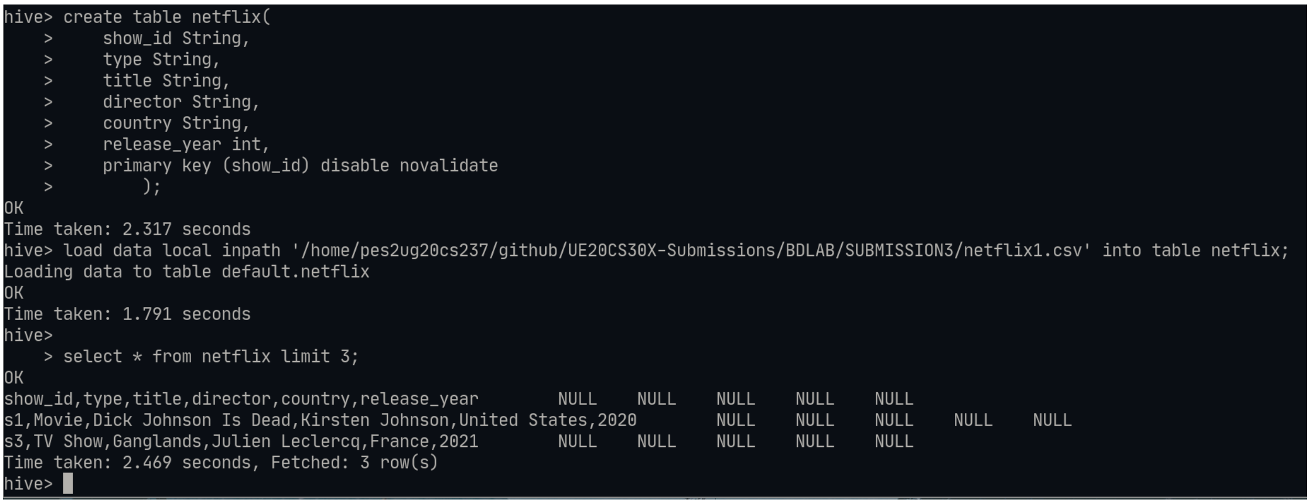
- Lets load it into hive! (Make sure to use absolute paths in linux)

```
load data local inpath '/home/pes2ug20cs237/github/UE20CS30X-Submissions/BDLAB
```

> Note : replace the path of file with what fits your system

- Lets see if the data is loaded using :

```
select * from netflix limit 3;
```

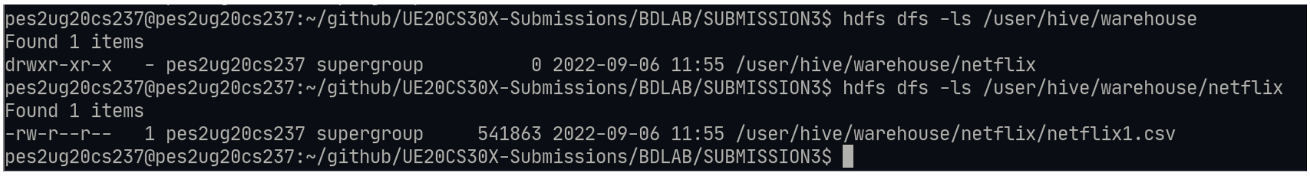


```
hive> create table netflix(
    >     show_id String,
    >     type String,
    >     title String,
    >     director String,
    >     country String,
    >     release_year int,
    >     primary key (show_id) disable novalidate
    >         );
OK
Time taken: 2.317 seconds
hive> load data local inpath '/home/pes2ug20cs237/github/UE20CS30X-Submissions/BDLAB/SUBMISSION3/netflix1.csv' into table netflix;
Loading data to table default.netflix
OK
Time taken: 1.791 seconds
hive>
    > select * from netflix limit 3;
OK
show_id,type,title,director,country,release_year        NULL    NULL    NULL    NULL    NULL
s1,Movie,Dick Johnson Is Dead,Kirsten Johnson,United States,2020        NULL    NULL    NULL    NULL    NULL
s3,TV Show,Ganglands,Julien Leclercq,France,2021        NULL    NULL    NULL    NULL    NULL
Time taken: 2.469 seconds, Fetched: 3 row(s)
hive> 
```

> Note : For jobs like SELECT, FILTER, LIMIT....Hive does not run map red job, instead uses FETCH of HDFS!

- By default all data warehouse data from hive are stored in /user/hive/warehouse, we can check this by doing the following hdfs commands :

```
hdfs dfs -ls /user/hive/warehouse
```



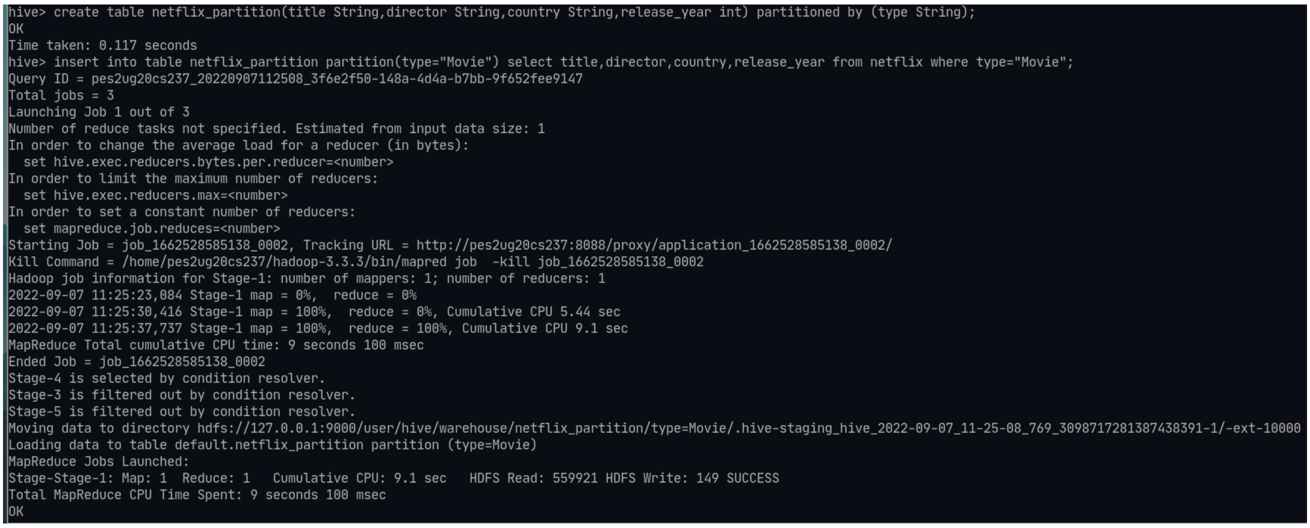
```
pes2ug20cs237@pes2ug20cs237:~/github/UE20CS30X-Submissions/BDLAB/SUBMISSION3$ hdfs dfs -ls /user/hive/warehouse
Found 1 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-06 11:55 /user/hive/warehouse/netflix
pes2ug20cs237@pes2ug20cs237:~/github/UE20CS30X-Submissions/BDLAB/SUBMISSION3$ hdfs dfs -ls /user/hive/warehouse/netflix
Found 1 items
-rw-r--r--   1 pes2ug20cs237 supergroup     541863 2022-09-06 11:55 /user/hive/warehouse/netflix/netflix1.csv
pes2ug20cs237@pes2ug20cs237:~/github/UE20CS30X-Submissions/BDLAB/SUBMISSION3$ 
```

> Note : this shows us the fact that a new folder is created for every schema we have!

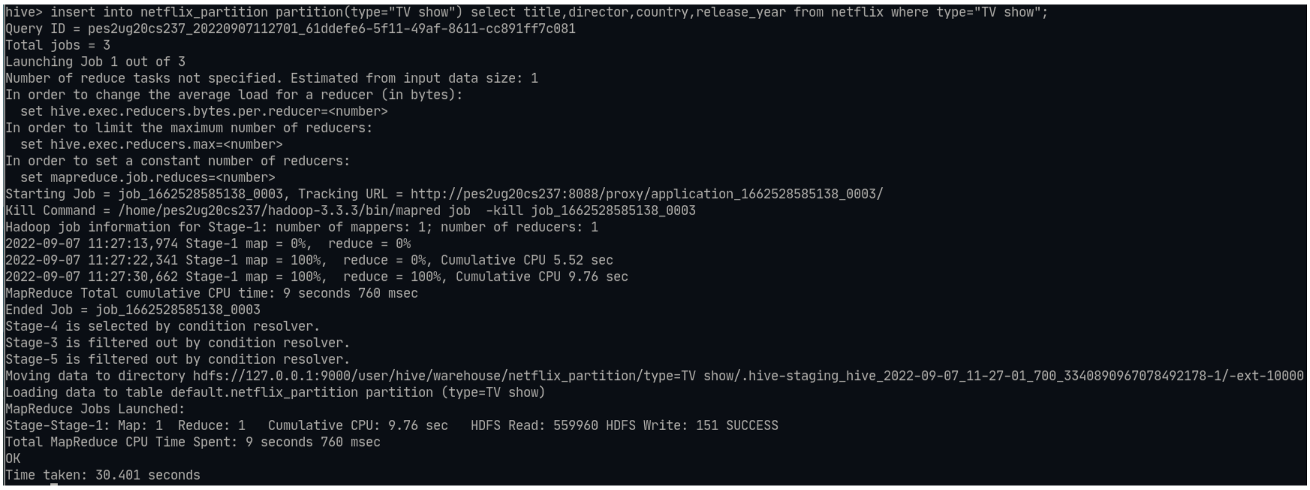
- Partitioning data : Hive organizes tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns such as type,country etc. Using partition, it is easy to query a portion of the data. For example, a table named Employee contains employee data such as id, name, dept, and yoj (i.e., year of joining). Suppose you need to retrieve the details of all employees who joined in 2012. A query searches the whole table for the required information. However, if you partition the employee data with the year and store it in a separate file, it reduces the query processing time.

```
hive >set hive.exec.dynamic.partition=True;
hive > set hive.exec.dynamic.partition.mode=nonstrict;
hive > create table netflix_partition(
    title String,
    director String,
    country String,
    release_year int
        ) partitioned by (type String);
```

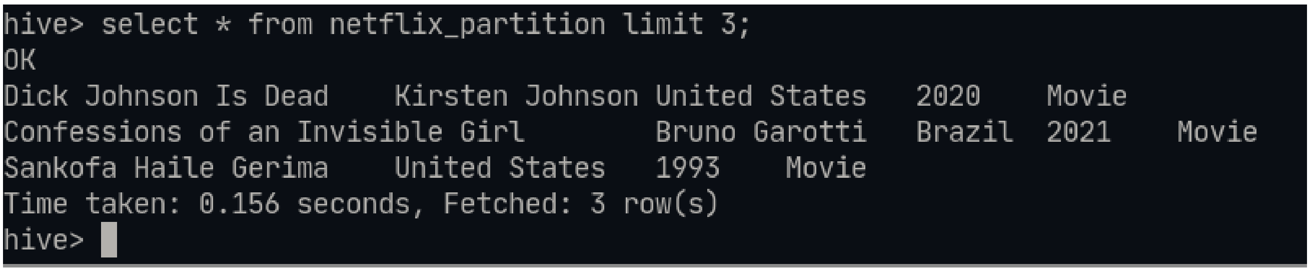
```
hive > insert into table netflix_partition partition(type="Movie") select titl
hive > insert into table netflix_partition partition(type="TV Show") select ti
hive > select * from netflix_partition limit 3;
```



```
hive> create table netflix_partition(title String,director String,country String,release_year int) partitioned by (type String);
OK
Time taken: 0.117 seconds
hive> insert into table netflix_partition partition(type="Movie") select title,director,country,release_year from netflix where type="Movie";
Query ID = pes2ug20cs237_20220907112508_3f6e2f50-148a-4d4a-b7bb-9f652fee9147
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
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_1662528585138_0002, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0002/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-07 11:25:23,084 Stage-1 map = 0%,  reduce = 0%
2022-09-07 11:25:30,416 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.44 sec
2022-09-07 11:25:37,737 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.1 sec
MapReduce Total cumulative CPU time: 9 seconds 100 msec
Ended Job = job_1662528585138_0002
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 directory hdfs://127.0.0.1:9000/user/hive/warehouse/netflix_partition/type=Movie/.hive-staging_hive_2022-09-07_11-25-08_769_3098717281387438391-1/-ext-10000
Loading data to table default.netflix_partition partition (type=Movie)
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.1 sec   HDFS Read: 559921 HDFS Write: 149 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 100 msec
OK
```

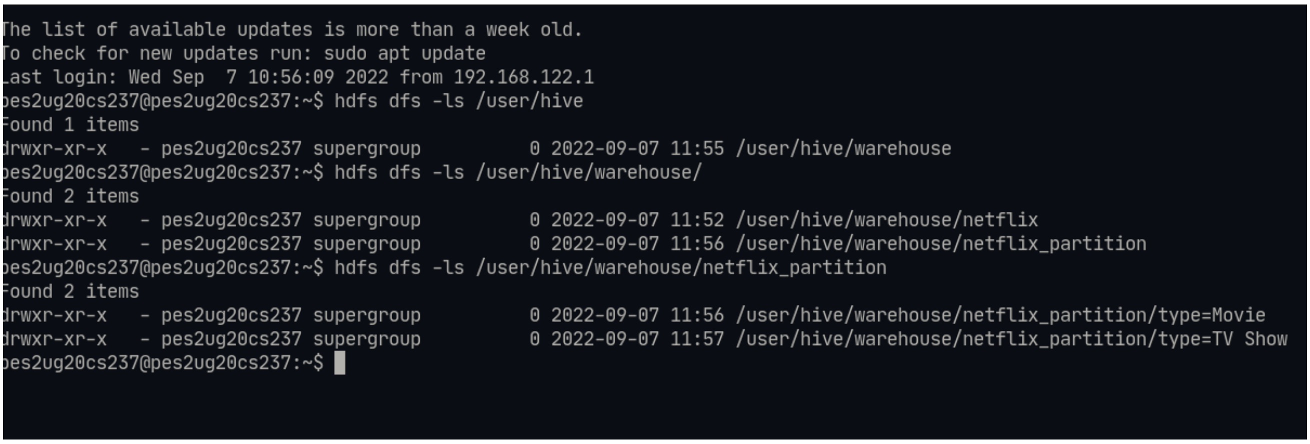


```
hive> insert into netflix_partition partition(type="TV show") select title,director,country,release_year from netflix where type="TV show";
Query ID = pes2ug20cs237_20220907112701_61ddefe6-5f11-49af-8611-cc891ff7c081
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
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_1662528585138_0003, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0003/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-07 11:27:13,974 Stage-1 map = 0%,  reduce = 0%
2022-09-07 11:27:22,341 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.52 sec
2022-09-07 11:27:30,662 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.76 sec
MapReduce Total cumulative CPU time: 9 seconds 760 msec
Ended Job = job_1662528585138_0003
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 directory hdfs://127.0.0.1:9000/user/hive/warehouse/netflix_partition/type=TV show/.hive-staging_hive_2022-09-07_11-27-01_700_3340890967078492178-1/-ext-10000
Loading data to table default.netflix_partition partition (type=TV show)
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.76 sec   HDFS Read: 559960 HDFS Write: 151 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 760 msec
OK
Time taken: 30.401 seconds
```



```
hive> select * from netflix_partition limit 3;
OK
Dick Johnson Is Dead    Kirsten Johnson United States   2020    Movie
Confessions of an Invisible Girl        Bruno Garotti   Brazil  2021    Movie
Sankofa Haile Gerima    United States   1993    Movie
Time taken: 0.156 seconds, Fetched: 3 row(s)
hive>
```

We can also see these partitions in the dfs :



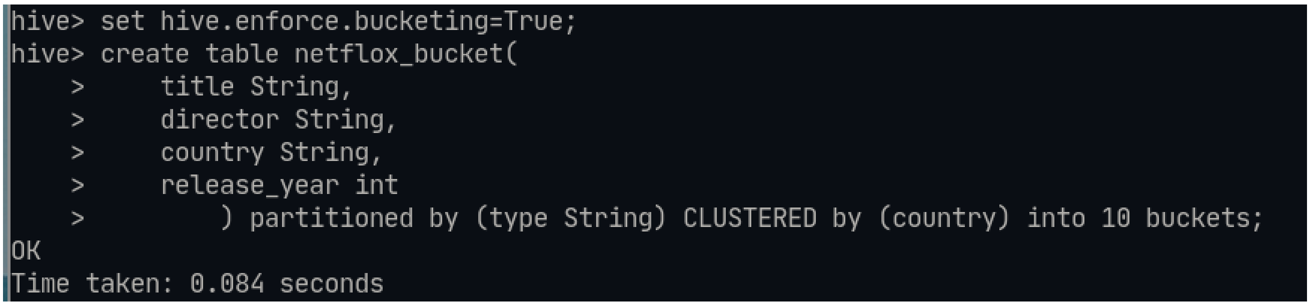
```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Wed Sep  7 10:56:09 2022 from 192.168.122.1
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive
Found 1 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:55 /user/hive/warehouse
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/
Found 2 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:52 /user/hive/warehouse/netflix
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:56 /user/hive/warehouse/netflix_partition
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/netflix_partition
Found 2 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:56 /user/hive/warehouse/netflix_partition/type=Movie
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:57 /user/hive/warehouse/netflix_partition/type=TV Show
pes2ug20cs237@pes2ug20cs237:~$
```

- Bucketing : Tables or partitions are sub-divided into buckets, to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

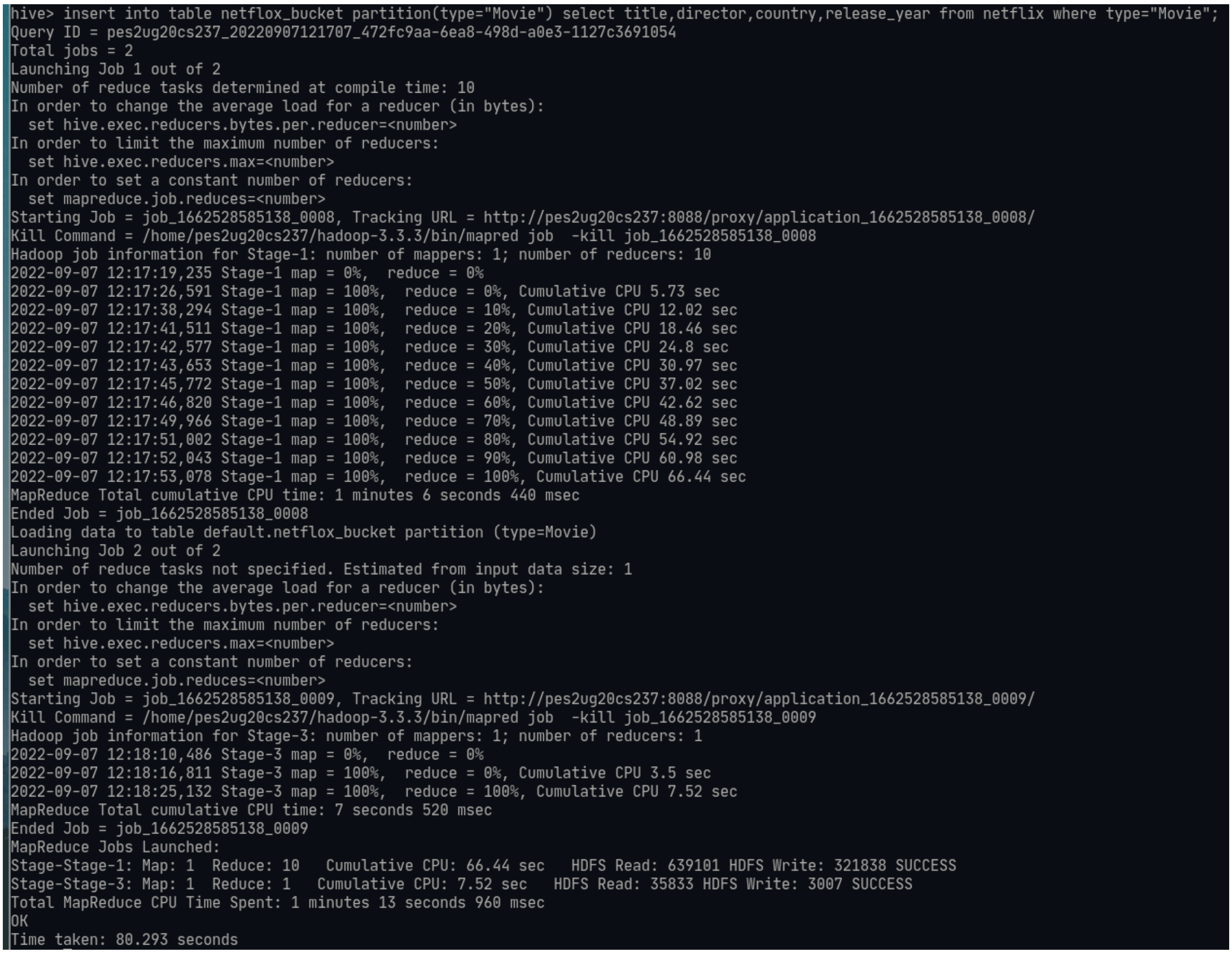
```
hive > set hive.enforce.bucketing=True;
hive > create table netflix_bucket(
    title String,
    director String,
    country String,
    release_year int
        ) partitioned by (type String) CLUSTERED by (country) into 10 buckets;
```

> Note : into 10 buckets indicative that the hash function is %10

```
hive > insert into table netflix_bucket partition(type="Movie") select title,d
```

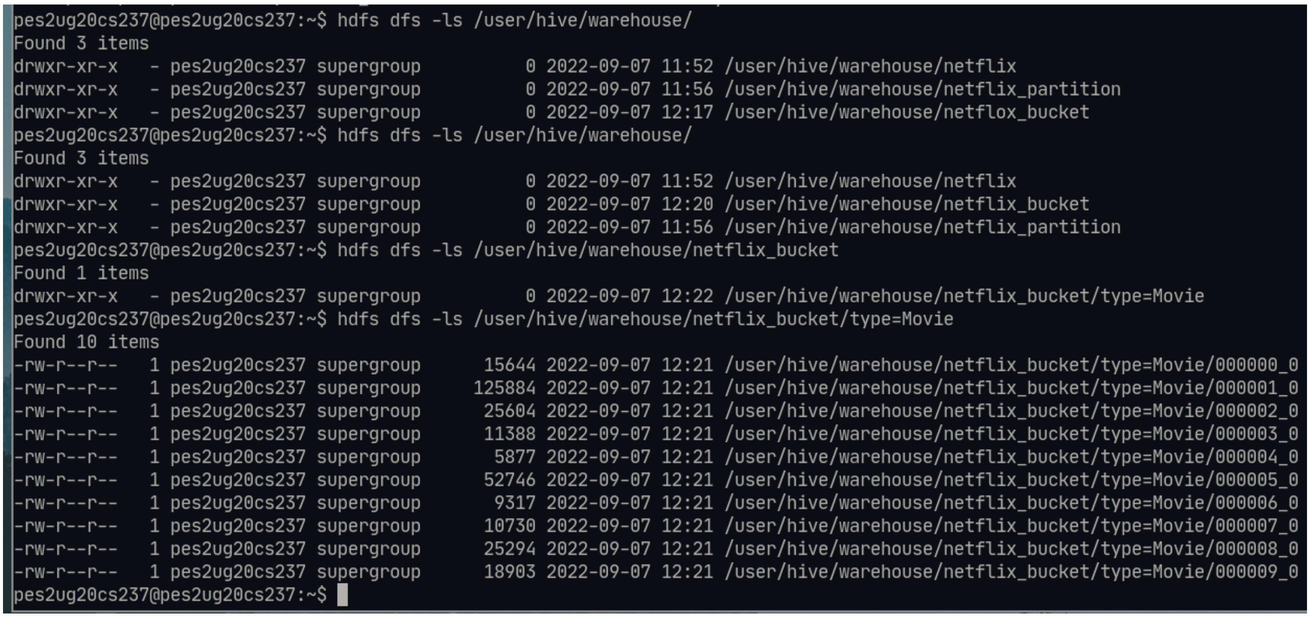


```
hive> set hive.enforce.bucketing=True;
hive> create table netflox_bucket(
    >     title String,
    >     director String,
    >     country String,
    >     release_year int
    >         ) partitioned by (type String) CLUSTERED by (country) into 10 buckets;
OK
Time taken: 0.084 seconds
```



```
hive> insert into table netflox_bucket partition(type="Movie") select title,director,country,release_year from netflix where type="Movie";
Query ID = pes2ug20cs237_20220907121707_472fc9aa-6ea8-498d-a0e3-1127c3691054
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 10
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_1662528585138_0008, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0008/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 10
2022-09-07 12:17:19,235 Stage-1 map = 0%,  reduce = 0%
2022-09-07 12:17:26,591 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.73 sec
2022-09-07 12:17:38,294 Stage-1 map = 100%,  reduce = 10%, Cumulative CPU 12.02 sec
2022-09-07 12:17:41,511 Stage-1 map = 100%,  reduce = 20%, Cumulative CPU 18.46 sec
2022-09-07 12:17:42,577 Stage-1 map = 100%,  reduce = 30%, Cumulative CPU 24.8 sec
2022-09-07 12:17:43,653 Stage-1 map = 100%,  reduce = 40%, Cumulative CPU 30.97 sec
2022-09-07 12:17:45,772 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 37.02 sec
2022-09-07 12:17:46,820 Stage-1 map = 100%,  reduce = 60%, Cumulative CPU 42.62 sec
2022-09-07 12:17:49,966 Stage-1 map = 100%,  reduce = 70%, Cumulative CPU 48.89 sec
2022-09-07 12:17:51,002 Stage-1 map = 100%,  reduce = 80%, Cumulative CPU 54.92 sec
2022-09-07 12:17:52,043 Stage-1 map = 100%,  reduce = 90%, Cumulative CPU 60.98 sec
2022-09-07 12:17:53,078 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 66.44 sec
MapReduce Total cumulative CPU time: 1 minutes 6 seconds 440 msec
Ended Job = job_1662528585138_0008
Loading data to table default.netflox_bucket partition (type=Movie)
Launching Job 2 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
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_1662528585138_0009, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0009/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0009
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2022-09-07 12:18:10,486 Stage-3 map = 0%,  reduce = 0%
2022-09-07 12:18:16,811 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 3.5 sec
2022-09-07 12:18:25,132 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 7.52 sec
MapReduce Total cumulative CPU time: 7 seconds 520 msec
Ended Job = job_1662528585138_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 10   Cumulative CPU: 66.44 sec   HDFS Read: 639101 HDFS Write: 321838 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 7.52 sec   HDFS Read: 35833 HDFS Write: 3007 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 13 seconds 960 msec
OK
Time taken: 80.293 seconds
```

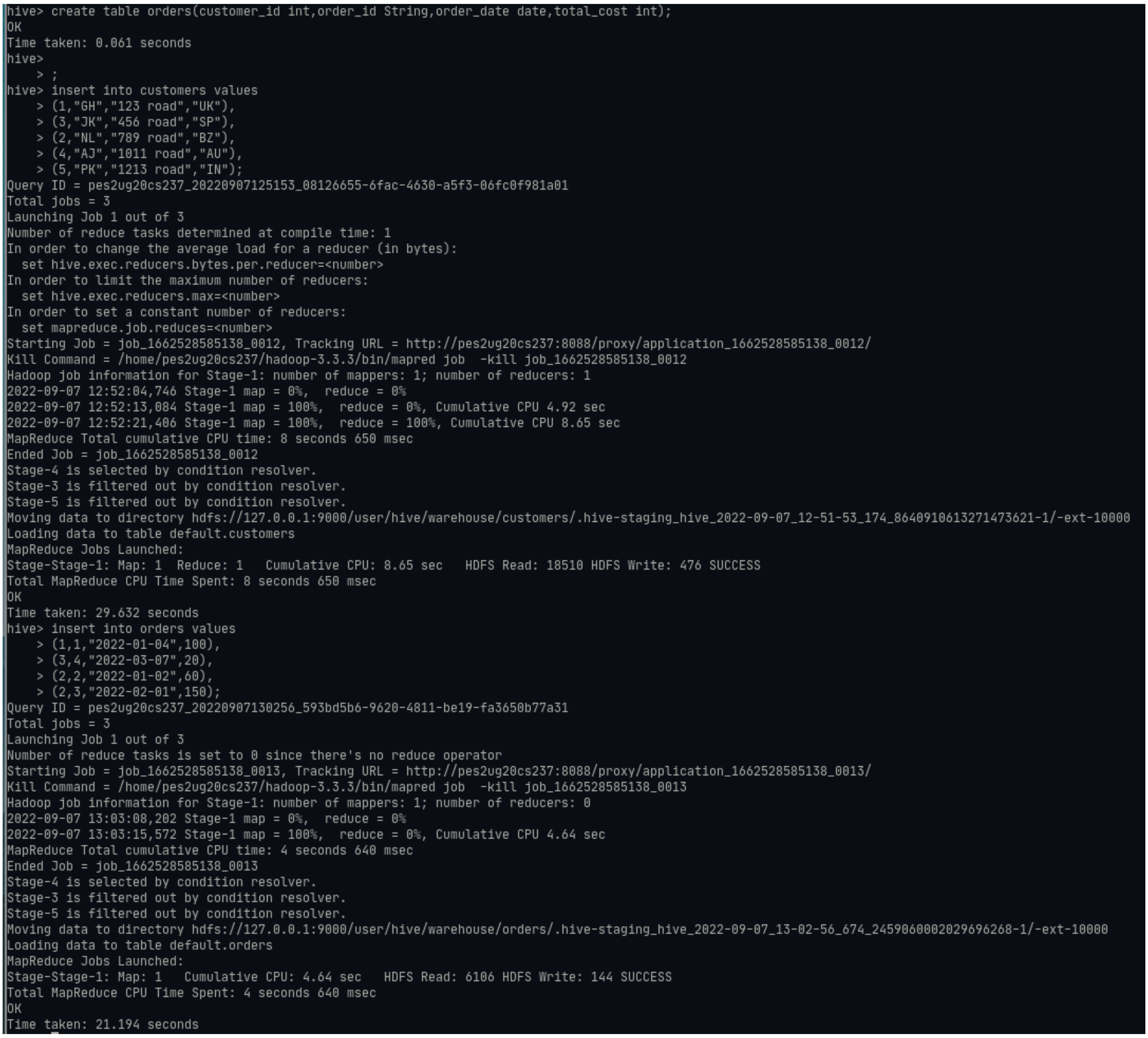
We can see these buckets with partitions in the dfs too :



```
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/
Found 3 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:52 /user/hive/warehouse/netflix
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:56 /user/hive/warehouse/netflix_partition
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 12:17 /user/hive/warehouse/netflox_bucket
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/
Found 3 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:52 /user/hive/warehouse/netflix
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 12:20 /user/hive/warehouse/netflix_bucket
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 11:56 /user/hive/warehouse/netflix_partition
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/netflix_bucket
Found 1 items
drwxr-xr-x   - pes2ug20cs237 supergroup          0 2022-09-07 12:22 /user/hive/warehouse/netflix_bucket/type=Movie
pes2ug20cs237@pes2ug20cs237:~$ hdfs dfs -ls /user/hive/warehouse/netflix_bucket/type=Movie
Found 10 items
-rw-r--r--   1 pes2ug20cs237 supergroup      15644 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000000_0
-rw-r--r--   1 pes2ug20cs237 supergroup     125884 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000001_0
-rw-r--r--   1 pes2ug20cs237 supergroup      25604 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000002_0
-rw-r--r--   1 pes2ug20cs237 supergroup      11388 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000003_0
-rw-r--r--   1 pes2ug20cs237 supergroup       5877 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000004_0
-rw-r--r--   1 pes2ug20cs237 supergroup      52746 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000005_0
-rw-r--r--   1 pes2ug20cs237 supergroup       9317 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000006_0
-rw-r--r--   1 pes2ug20cs237 supergroup      10730 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000007_0
-rw-r--r--   1 pes2ug20cs237 supergroup      25294 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000008_0
-rw-r--r--   1 pes2ug20cs237 supergroup      18903 2022-09-07 12:21 /user/hive/warehouse/netflix_bucket/type=Movie/000009_0
pes2ug20cs237@pes2ug20cs237:~$ 
```

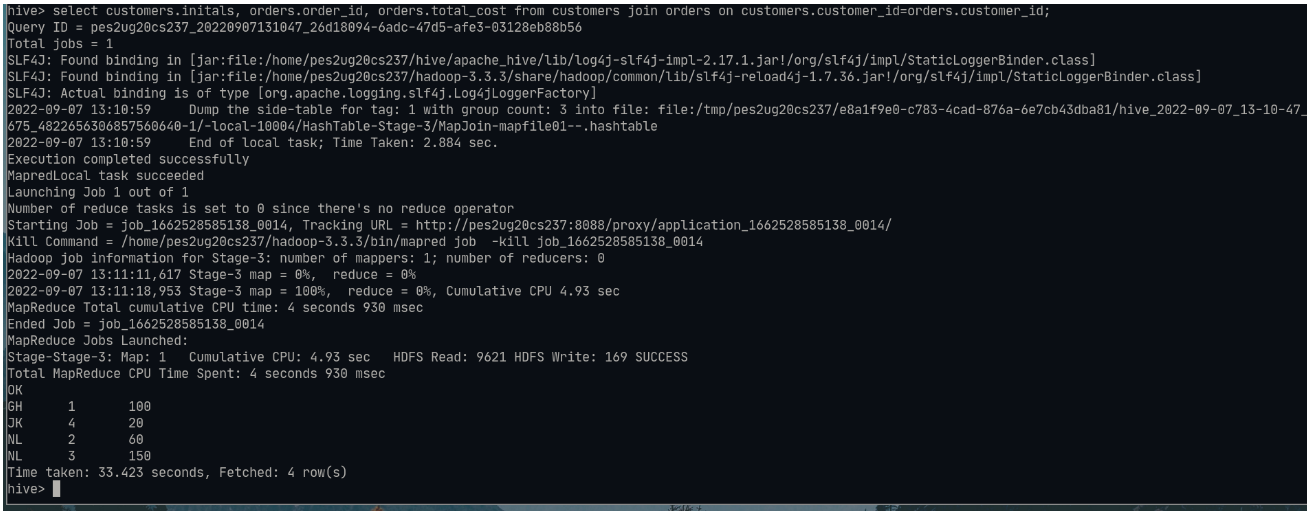
- Map joins and normal joins : First up, create two table and load the data.

```
hive > create table customers(customer_id int,initals String,street String,cou
hive > create table orders(customer_id int,order_id String,order_date date,tot
hive > insert into customers values
(1,"GH","123 road","UK"),
(3,"JK","456 road","SP"),
(2,"NL","789 road","BZ"),
(4,"AJ","1011 road","AU"),
(5,"PK","1213 road","IN");
hive > insert into orders values
(1,1,"2022-01-04",100),
(3,4,"2022-03-07",20),
(2,2,"2022-01-02",60),
(2,3,"2022-02-01",150);
```



```
hive> create table orders(customer_id int,order_id String,order_date date,total_cost int);
OK
Time taken: 0.061 seconds
hive>
    > ;
hive> insert into customers values
    > (1,"GH","123 road","UK"),
    > (3,"JK","456 road","SP"),
    > (2,"NL","789 road","BZ"),
    > (4,"AJ","1011 road","AU"),
    > (5,"PK","1213 road","IN");
Query ID = pes2ug20cs237_20220907125153_08126655-6fac-4630-a5f3-06fc0f981a01
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
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_1662528585138_0012, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0012/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-07 12:52:04,746 Stage-1 map = 0%,  reduce = 0%
2022-09-07 12:52:13,084 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.92 sec
2022-09-07 12:52:21,406 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.65 sec
MapReduce Total cumulative CPU time: 8 seconds 650 msec
Ended Job = job_1662528585138_0012
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 directory hdfs://127.0.0.1:9000/user/hive/warehouse/customers/.hive-staging_hive_2022-09-07_12-51-53_174_8640910613271473621-1/-ext-10000
Loading data to table default.customers
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 8.65 sec   HDFS Read: 18510 HDFS Write: 476 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 650 msec
OK
Time taken: 29.632 seconds
hive> insert into orders values
    > (1,1,"2022-01-04",100),
    > (3,4,"2022-03-07",20),
    > (2,2,"2022-01-02",60),
    > (2,3,"2022-02-01",150);
Query ID = pes2ug20cs237_20220907130256_593bd5b6-9620-4811-be19-fa3650b77a31
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1662528585138_0013, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0013/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-09-07 13:03:08,202 Stage-1 map = 0%,  reduce = 0%
2022-09-07 13:03:15,572 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.64 sec
MapReduce Total cumulative CPU time: 4 seconds 640 msec
Ended Job = job_1662528585138_0013
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 directory hdfs://127.0.0.1:9000/user/hive/warehouse/orders/.hive-staging_hive_2022-09-07_13-02-56_674_2459060002029696268-1/-ext-10000
Loading data to table default.orders
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 4.64 sec   HDFS Read: 6106 HDFS Write: 144 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 640 msec
OK
Time taken: 21.194 seconds
```

Doing a normal join on these two tables :

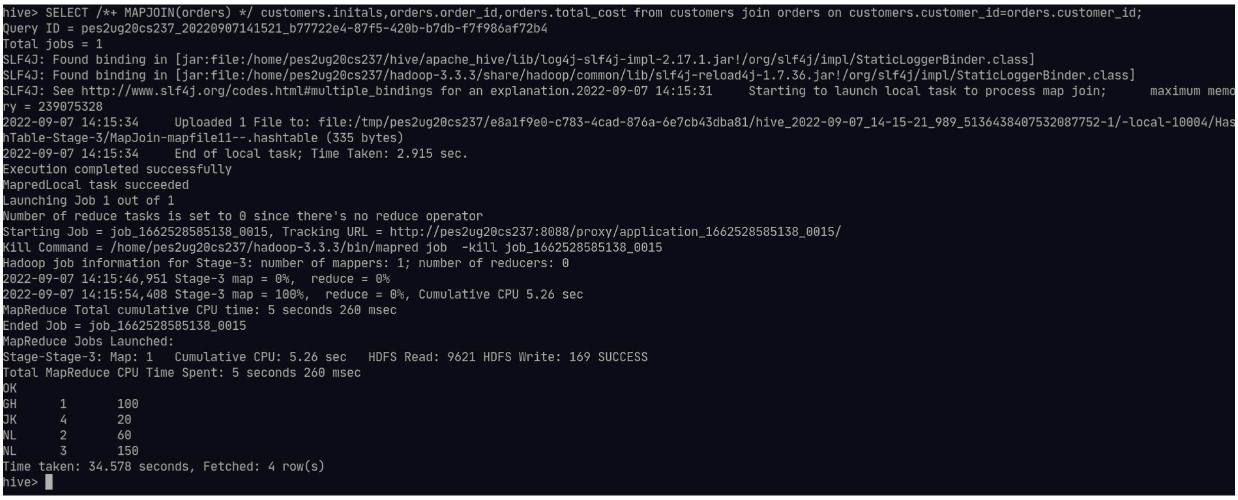


```
hive> select customers.initals, orders.order_id, orders.total_cost from customers join orders on customers.customer_id=orders.customer_id;
Query ID = pes2ug20cs237_20220907131047_26d18094-6adc-47d5-afe3-03128eb88b56
Total jobs = 1
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hive/apache_hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hadoop-3.3.3/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2022-09-07 13:10:59     Dump the side-table for tag: 1 with group count: 3 into file: file:/tmp/pes2ug20cs237/e8a1f9e0-c783-4cad-876a-6e7cb43dba81/hive_2022-09-07_13-10-47_
675_4822656306857560640-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2022-09-07 13:10:59     End of local task; Time Taken: 2.884 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1662528585138_0014, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0014/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0014
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2022-09-07 13:11:11,617 Stage-3 map = 0%,  reduce = 0%
2022-09-07 13:11:18,953 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 4.93 sec
MapReduce Total cumulative CPU time: 4 seconds 930 msec
Ended Job = job_1662528585138_0014
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 4.93 sec   HDFS Read: 9621 HDFS Write: 169 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 930 msec
OK
GH      1       100
JK      4       20
NL      2       60
NL      3       150
Time taken: 33.423 seconds, Fetched: 4 row(s)
hive> 
```

What is map join : A table can be loaded into the memory completely within a mapper without using the Map/Reducer process. It reads the data from the smaller table and stores it in an in-memory hash table and then serializes it to a hash memory file, thus substantially reducing the time. It is also known as Map Side Join in Hive. Basically, it involves performing joins between 2 tables by using only the Map phase and skipping the Reduce phase. A time decrease in your queries' computation can be observed if they regularly use a small table joins. Map-side join helps in minimizing the cost that is incurred for sorting and merging in the shuffle and reduce stages. Map-side join also helps in improving the performance of the task by decreasing the time to finish the task.

```
SELECT /*+ MAPJOIN(orders) */ customers.initals,orders.order_id,orders.total_c
```

> Note : in the above MAPJOIN(orders) loads orders into memory and stores it in a hash map.



```
hive> SELECT /*+ MAPJOIN(orders) */ customers.initals,orders.order_id,orders.total_cost from customers join orders on customers.customer_id=orders.customer_id;
Query ID = pes2ug20cs237_20220907141521_b77722e4-87f5-420b-b7db-f7f986af72b4
Total jobs = 1
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hive/apache_hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hadoop-3.3.3/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.2022-09-07 14:15:31     Starting to launch local task to process map join;      maximum memo
ry = 239075328
2022-09-07 14:15:34     Uploaded 1 File to: file:/tmp/pes2ug20cs237/e8a1f9e0-c783-4cad-876a-6e7cb43dba81/hive_2022-09-07_14-15-21_989_5136438407532087752-1/-local-10004/Has
hTable-Stage-3/MapJoin-mapfile11--.hashtable (335 bytes)
2022-09-07 14:15:34     End of local task; Time Taken: 2.915 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1662528585138_0015, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0015/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0015
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2022-09-07 14:15:46,951 Stage-3 map = 0%,  reduce = 0%
2022-09-07 14:15:54,408 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 5.26 sec
MapReduce Total cumulative CPU time: 5 seconds 260 msec
Ended Job = job_1662528585138_0015
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 5.26 sec   HDFS Read: 9621 HDFS Write: 169 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 260 msec
OK
GH      1       100
JK      4       20
NL      2       60
NL      3       150
Time taken: 34.578 seconds, Fetched: 4 row(s)
hive> 
```

- Transacitions : Update, Delete and Modify These are transactional commansd and hence need particular configs in hive :

```
SET hive.support.concurrency=true;
SET hive.txn.manager=org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;
SET hive.compactor.initiator.on=true;
SET hive.compactor.worker.threads=1;
```

Let's now create a new table with transactional properties :

```
create table transaction_table(
    name String
        ) STORED AS ORC TBLPROPERTIES ('transactional' = 'true');
```

```
- ORC : The Optimized Row Columnar (ORC) file format provides a highly efficie
```

```
 insert into transaction_table VALUES
    ("navin1"),
    ("navin2"),
    ("navin3");
```

```
- Updating
```

```
UPDATE transaction_table
SET name="not_navin"
WHERE name="navin3";
```

> WHERE clause is optional, if not present it updates all the records. - Deleting records :

```
DELETE FROM transaction_table
WHERE name="not_navin";
```

> WHERE clause is optional, if not present it deletes all the recods.

Excersise :

- creating table :

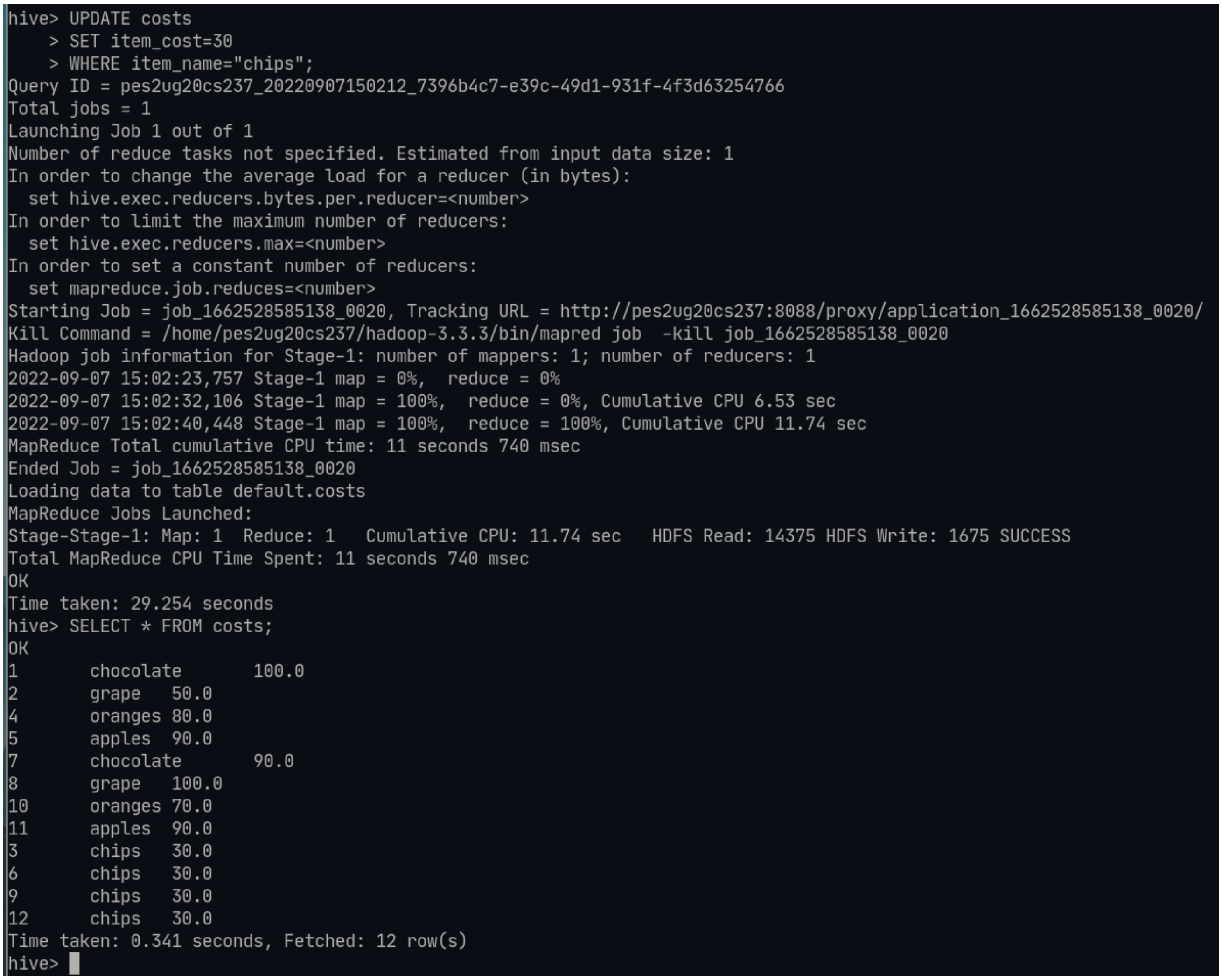
```
CREATE TABLE costs(
    id int,
    item_name String,
    item_cost double,
    primary key (id) disable novalidate
        ) STORED AS ORC TBLPROPERTIES('transactional' = 'true');
```

- Inserting records :

```
INSERT INTO costs VALUES
(1,"chocolate",100),
(2,"grape", 50),
(3,"chips", 10),
(4,"oranges", 80),
(5,"apples", 90),
(6,"chips", 20),
(7,"chocolate", 90),
(8,"grape", 100),
(9,"chips", 40),
(10,"oranges", 70),
(11,"apples", 90),
(12,"chips", 20);
```

- Updating the cost of chips to 30 :

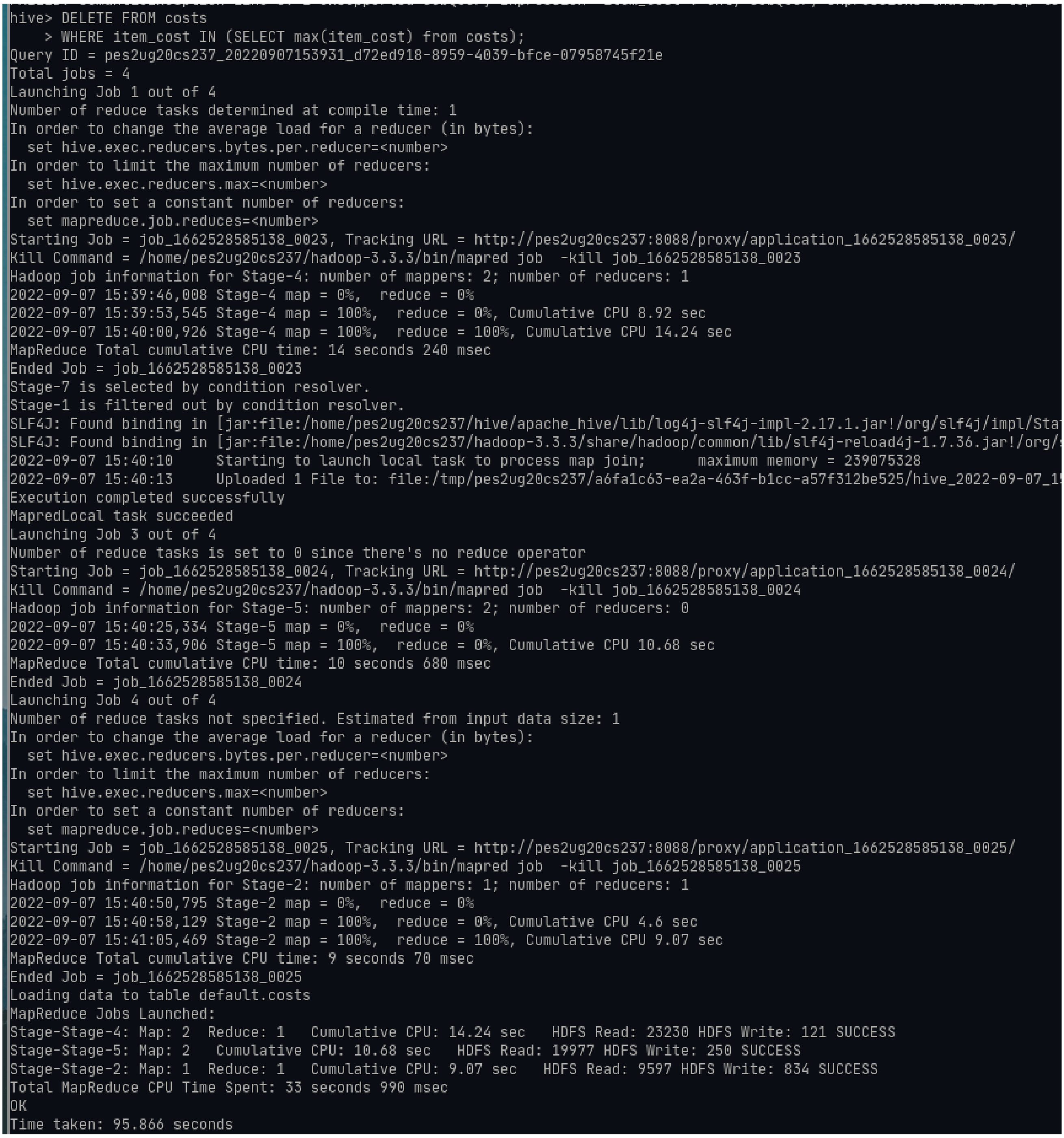
```
UPDATE costs
SET item_cost=30
WHERE item_name="chips";
```



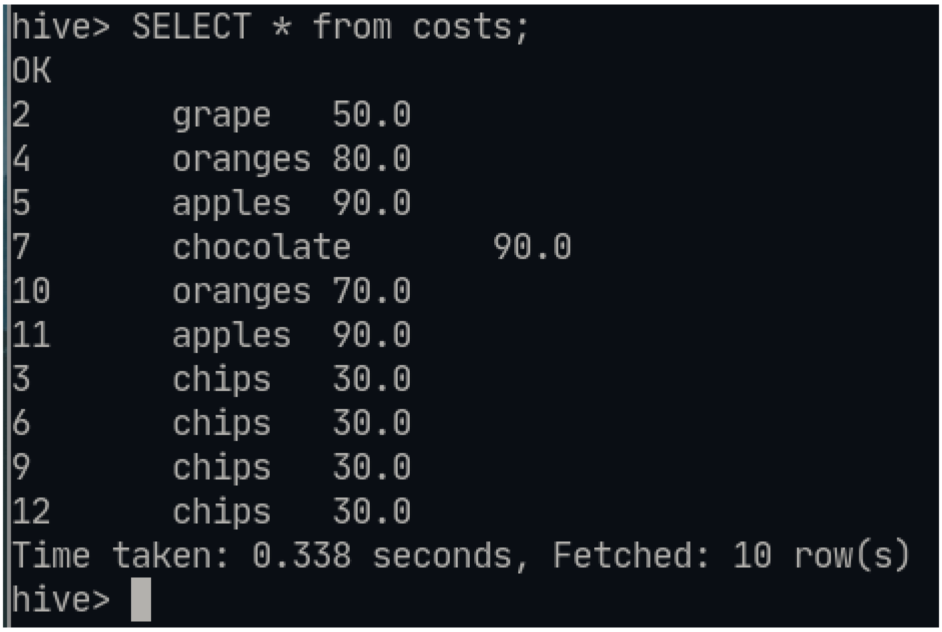
```
hive> UPDATE costs
    > SET item_cost=30
    > WHERE item_name="chips";
Query ID = pes2ug20cs237_20220907150212_7396b4c7-e39c-49d1-931f-4f3d63254766
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load 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_1662528585138_0020, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0020/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0020
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-09-07 15:02:23,757 Stage-1 map = 0%,  reduce = 0%
2022-09-07 15:02:32,106 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.53 sec
2022-09-07 15:02:40,448 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.74 sec
MapReduce Total cumulative CPU time: 11 seconds 740 msec
Ended Job = job_1662528585138_0020
Loading data to table default.costs
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.74 sec   HDFS Read: 14375 HDFS Write: 1675 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 740 msec
OK
Time taken: 29.254 seconds
hive> SELECT * FROM costs;
OK
1       chocolate       100.0
2       grape   50.0
4       oranges 80.0
5       apples  90.0
7       chocolate       90.0
8       grape   100.0
10      oranges 70.0
11      apples  90.0
3       chips   30.0
6       chips   30.0
9       chips   30.0
12      chips   30.0
Time taken: 0.341 seconds, Fetched: 12 row(s)
hive> 
```

- Deleting records with max item_cost :

```
DELETE FROM costs
WHERE item_cost IN (SELECT max(item_cost) from costs);
```



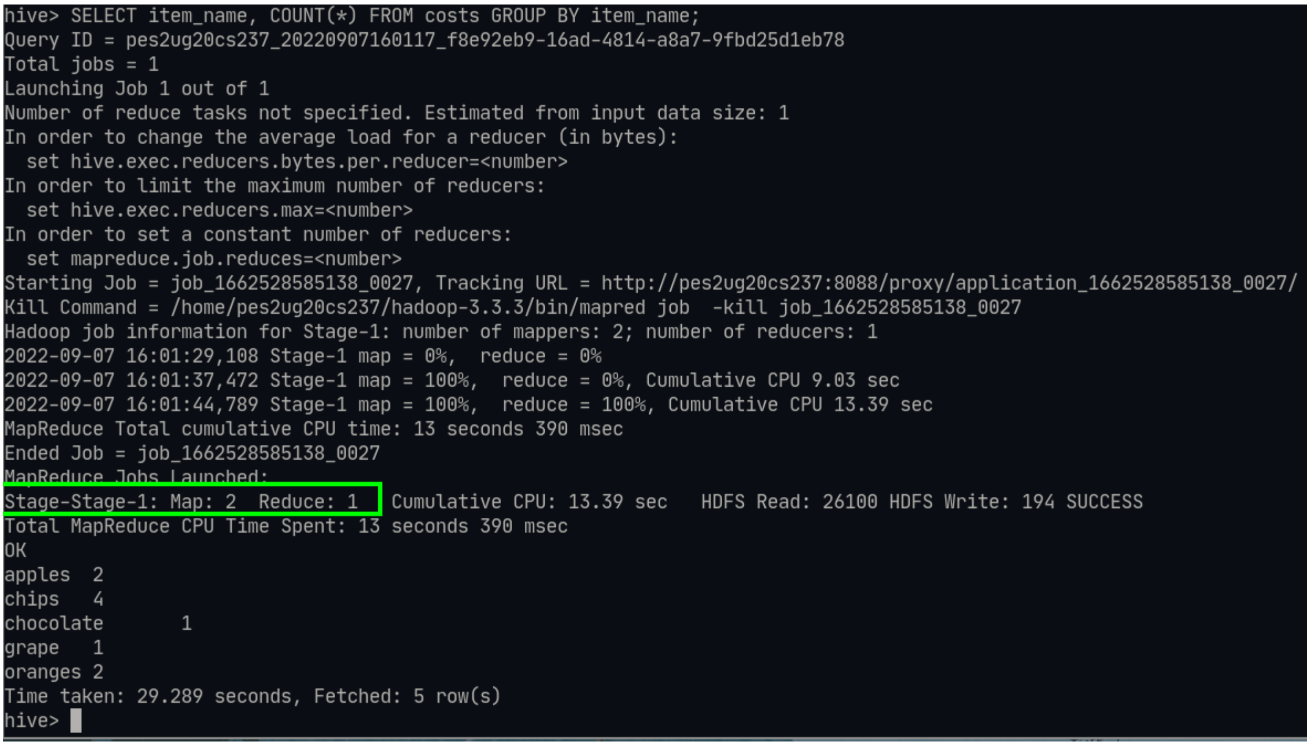
```
hive> DELETE FROM costs
    > WHERE item_cost IN (SELECT max(item_cost) from costs);
Query ID = pes2ug20cs237_20220907153931_d72ed918-8959-4039-bfce-07958745f21e
Total jobs = 4
Launching Job 1 out of 4
Number of reduce tasks determined at compile time: 1
In order to change the average load for a 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_1662528585138_0023, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0023/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0023
Hadoop job information for Stage-4: number of mappers: 2; number of reducers: 1
2022-09-07 15:39:46,008 Stage-4 map = 0%,  reduce = 0%
2022-09-07 15:39:53,545 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 8.92 sec
2022-09-07 15:40:00,926 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 14.24 sec
MapReduce Total cumulative CPU time: 14 seconds 240 msec
Ended Job = job_1662528585138_0023
Stage-7 is selected by condition resolver.
Stage-1 is filtered out by condition resolver.
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hive/apache_hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/Sta
SLF4J: Found binding in [jar:file:/home/pes2ug20cs237/hadoop-3.3.3/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/
2022-09-07 15:40:10     Starting to launch local task to process map join;      maximum memory = 239075328
2022-09-07 15:40:13     Uploaded 1 File to: file:/tmp/pes2ug20cs237/a6fa1c63-ea2a-463f-b1cc-a57f312be525/hive_2022-09-07_1
Execution completed successfully
MapredLocal task succeeded
Launching Job 3 out of 4
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1662528585138_0024, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0024/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0024
Hadoop job information for Stage-5: number of mappers: 2; number of reducers: 0
2022-09-07 15:40:25,334 Stage-5 map = 0%,  reduce = 0%
2022-09-07 15:40:33,906 Stage-5 map = 100%,  reduce = 0%, Cumulative CPU 10.68 sec
MapReduce Total cumulative CPU time: 10 seconds 680 msec
Ended Job = job_1662528585138_0024
Launching Job 4 out of 4
Number of reduce tasks not specified. Estimated from input data size: 1
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_1662528585138_0025, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0025/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0025
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-09-07 15:40:50,795 Stage-2 map = 0%,  reduce = 0%
2022-09-07 15:40:58,129 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.6 sec
2022-09-07 15:41:05,469 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 9.07 sec
MapReduce Total cumulative CPU time: 9 seconds 70 msec
Ended Job = job_1662528585138_0025
Loading data to table default.costs
MapReduce Jobs Launched:
Stage-Stage-4: Map: 2  Reduce: 1   Cumulative CPU: 14.24 sec   HDFS Read: 23230 HDFS Write: 121 SUCCESS
Stage-Stage-5: Map: 2   Cumulative CPU: 10.68 sec   HDFS Read: 19977 HDFS Write: 250 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.07 sec   HDFS Read: 9597 HDFS Write: 834 SUCCESS
Total MapReduce CPU Time Spent: 33 seconds 990 msec
OK
Time taken: 95.866 seconds
```



```
hive> SELECT * from costs;
OK
2       grape   50.0
4       oranges 80.0
5       apples  90.0
7       chocolate       90.0
10      oranges 70.0
11      apples  90.0
3       chips   30.0
6       chips   30.0
9       chips   30.0
12      chips   30.0
Time taken: 0.338 seconds, Fetched: 10 row(s)
hive>
```

- Query to find total number of each item

```
SELECT item_name, COUNT(*) FROM costs GROUP BY item_name;
```



```
hive> SELECT item_name, COUNT(*) FROM costs GROUP BY item_name;
Query ID = pes2ug20cs237_20220907160117_f8e92eb9-16ad-4814-a8a7-9fbd25d1eb78
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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_1662528585138_0027, Tracking URL = http://pes2ug20cs237:8088/proxy/application_1662528585138_0027/
Kill Command = /home/pes2ug20cs237/hadoop-3.3.3/bin/mapred job  -kill job_1662528585138_0027
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-09-07 16:01:29,108 Stage-1 map = 0%,  reduce = 0%
2022-09-07 16:01:37,472 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 9.03 sec
2022-09-07 16:01:44,789 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.39 sec
MapReduce Total cumulative CPU time: 13 seconds 390 msec
Ended Job = job_1662528585138_0027
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 13.39 sec   HDFS Read: 26100 HDFS Write: 194 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 390 msec
OK
apples  2
chips   4
chocolate       1
grape   1
oranges 2
Time taken: 29.289 seconds, Fetched: 5 row(s)
hive>
```