Mycart Documentation

Add jar /home/acadgild/ecommerce/hive-serdes-1.0-SNAPSHOT.jar

Products_info_raw table creation

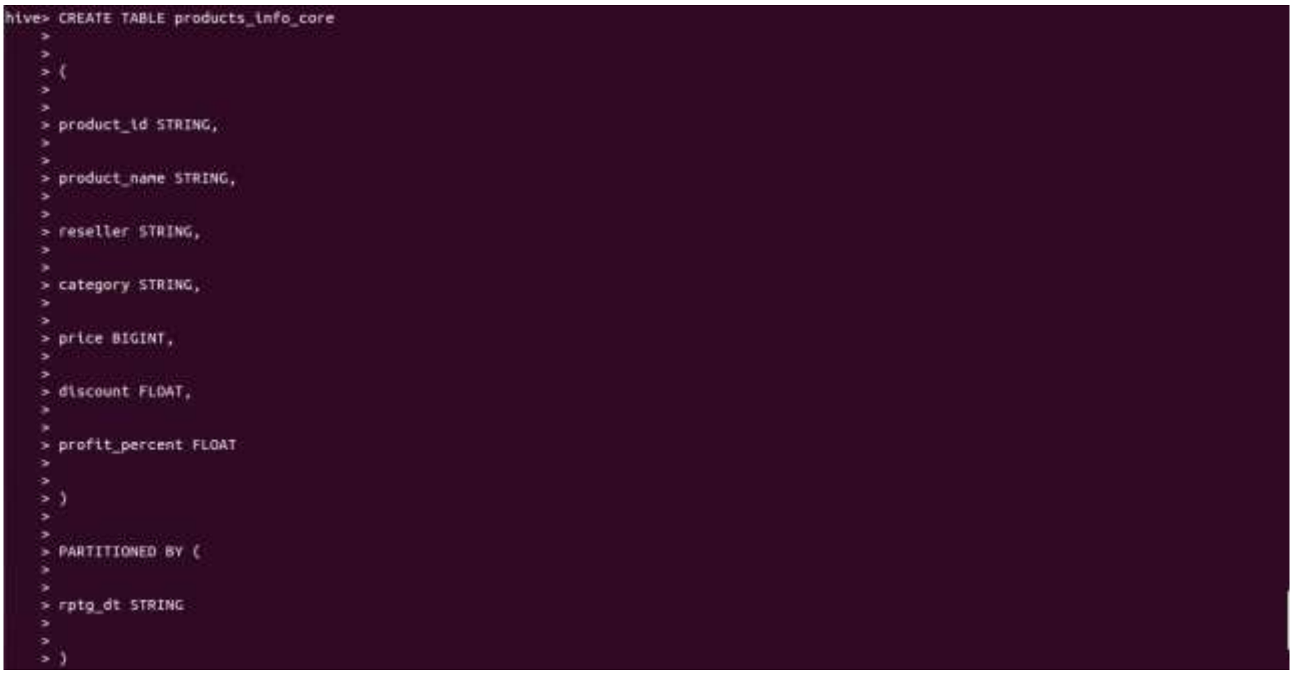
```
create table products_info_raw(id STRING, name STRING, reseller STRING, category STRING,
price DOUBLE, discount INT, profit_percent INT) ROW FORMAT SERDE
'com.cloudera.hive.serde.JSONSerDe';
```

Products_info_stg table creation

```
CREATE TABLE products_info_stg ( product_id STRING, product_name STRING, reseller
STRING, category STRING, price BIGINT, discount FLOAT, profit_percent FLOAT )
PARTITIONED BY ( rptg_dt STRING ) CLUSTERED BY ( product_id) INTO 8 BUCKETS
STORED AS ORC;
```

Creating products_info_core table

```
CREATE TABLE products_info_core

(

product_id STRING,

product_name STRING,

reseller STRING,

category STRING,

price BIGINT,

discount FLOAT,

profit_percent FLOAT

)

PARTITIONED BY (

rptg_dt STRING

)
CLUSTERED BY
(
product_id)
INTO 8 BUCKETS
STORED AS ORC;
```



```
hive> CREATE TABLE products_info_core
    >
    >
    > (
    >
    >
    > product_id STRING,
    >
    >
    > product_name STRING,
    >
    >
    > reseller STRING,
    >
    >
    > category STRING,
    >
    >
    > price BIGINT,
    >
    >
    > discount FLOAT,
    >
    >
    > profit_percent FLOAT
    >
    >
    > )
    >
    >
    > PARTITIONED BY (
    >
    >
    > rptg_dt STRING
    >
    >
    > )
```

Creating table products_info_excp

CREATE TABLE products_info_excp

(

product_id STRING,

product_name STRING,

reseller STRING,

category STRING,

price BIGINT,

discount FLOAT,

profit_percent FLOAT,

rule_failed STRING

)

PARTITIONED BY (

rptg_dt STRING

)
CLUSTERED BY (
product_id)
INTO 8 BUCKETS
STORED AS ORC;

User_activity_raw table creation

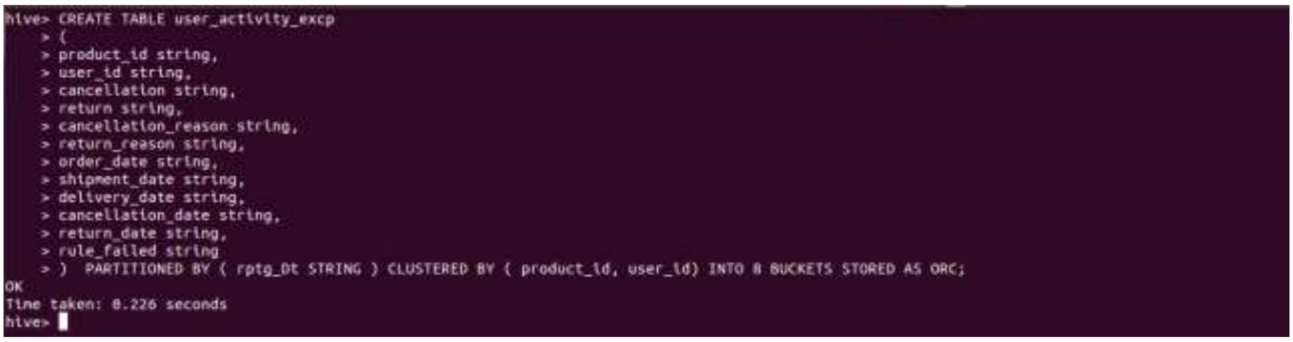
```
CREATE TABLE user_activity_raw (
product_id string,
user_id string,
cancellation string,
return string,
cancellation_reason string,
return_reason string,
order_date string,
shipment_date string,
delivery_date string,
cancellation_date string,
return_date string
)
ROW FORMAT SERDE
'com.cloudera.hive.serde.JSONSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat';
```

user_activity_stg table creation

```
CREATE TABLE user_activity_stg
(
product_id string,
user_id string,
cancellation string,
return string,
cancellation_reason string,
return_reason string,
order_date string,
shipment_date string,
delivery_date string,
cancellation_date string,
return_date string
)

PARTITIONED BY
(
rptg_Dt STRING
)
CLUSTERED BY (
product_id,
user_id)
INTO 8 BUCKETS
STORED AS ORC;
```

user_activity_excp table creation

CREATE TABLE user_activity_excp
(
product_id string,
user_id string,
cancellation string,
return string,
cancellation_reason string,
return_reason string,
order_date string,
shipment_date string,
delivery_date string,
cancellation_date string,
return_date string,
rule_failed string
) PARTITIONED BY (rptg_Dt STRING) CLUSTERED BY (product_id, user_id) INTO 8 BUCKETS STORED AS ORC;



```
hive> CREATE TABLE user_activity_excp
    > (
    > product_id string,
    > user_id string,
    > cancellation string,
    > return string,
    > cancellation_reason string,
    > return_reason string,
    > order_date string,
    > shipment_date string,
    > delivery_date string,
    > cancellation_date string,
    > return_date string,
    > rule_failed string
    > )  PARTITIONED BY ( rptg_Dt STRING ) CLUSTERED BY ( product_id, user_id) INTO 8 BUCKETS STORED AS ORC;
OK
Time taken: 0.226 seconds
hive> 
```

user_activity_core table creation

CREATE TABLE user_activity_core
(
product_id string,
user_id string,
cancellation string,
return string,
cancellation_reason string,
return_reason string,
order_date string,
shipment_date string,
delivery_date string,
cancellation_date string,
return_date string
)

PARTITIONED BY
(
rptg_Dt STRING
)
CLUSTERED BY (

```
product_id,
user_id)
INTO 8 BUCKETS
STORED AS ORC;
```

user_info_raw table creation

```
CREATE TABLE users_info_raw(

id string,
name string,
location struct<city:string,state:string>,
age INT,
category string
)

ROW FORMAT SERDE
'com.cloudera.hive.serde.JSONSerDe'

STORED AS INPUTFORMAT
'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat';
```

users_info_stg table creation

```
CREATE TABLE users_info_stg(

user_id string,
name string,
location struct<city:string,state:string>,
age bigint,
occupation string
)

PARTITIONED BY
(
rptg_Dt STRING
)
CLUSTERED BY (
user_id)
INTO 8 BUCKETS
STORED AS ORC;
```

users_info_excp table creation

```
CREATE TABLE users_info_excp(
user_id string,
name string,
location struct<city:string,state:string>,
age bigint,
occupation string,
rule_failed STRING
)
PARTITIONED BY
(
rptg_Dt STRING
)
CLUSTERED BY (
user_id)
INTO 8 BUCKETS
STORED AS ORC;
```

user_info_core table creation

```
CREATE TABLE users_info_core(
user_id string,
name string,
location struct<city:string,state:string>,
age bigint,
occupation string
)

PARTITIONED BY
(
rptg_Dt STRING
)
CLUSTERED BY (
user_id)
INTO 8 BUCKETS
STORED AS ORC;
```

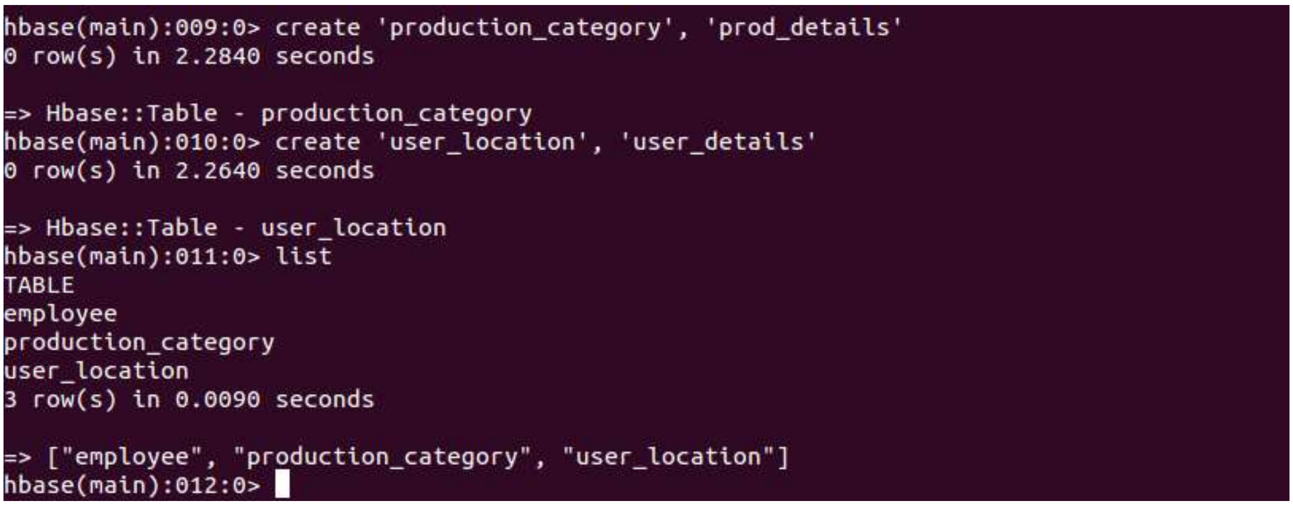
Hbase table creation

production_category hbase table creation

```
create 'production_category', 'prod_details'
```

user_location hbase table creation

```
create 'user_location', 'user_details'
```



```
hbase(main):009:0> create 'production_category', 'prod_details'
0 row(s) in 2.2840 seconds

=> Hbase::Table - production_category
hbase(main):010:0> create 'user_location', 'user_details'
0 row(s) in 2.2640 seconds

=> Hbase::Table - user_location
hbase(main):011:0> list
TABLE
employee
production_category
user_location
3 row(s) in 0.0090 seconds

=> ["employee", "production_category", "user_location"]
hbase(main):012:0> 
```

****
DATA INSERTION
****

Loading data to products_info_raw table

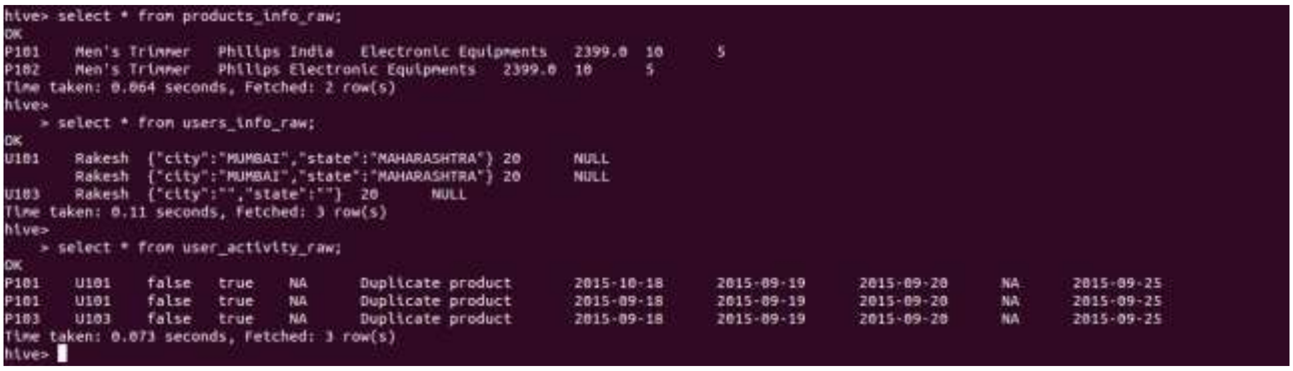
LOAD DATA LOCAL INPATH '/home/acadgild/ecommerce/data/product_info_merge.json'
INTO TABLE products_info_raw;

Loading data into users_info_raw table

LOAD DATA LOCAL INPATH '/home/acadgild/ecommerce/data/user_info_1.json'
INTO TABLE users_info_raw;

Loading data into user_activity_raw tbale
LOAD DATA LOCAL INPATH '/home/acadgild/ecommerce/data/user_activity_1.json'
INTO TABLE user_activity_raw;

Displaying the inserted data

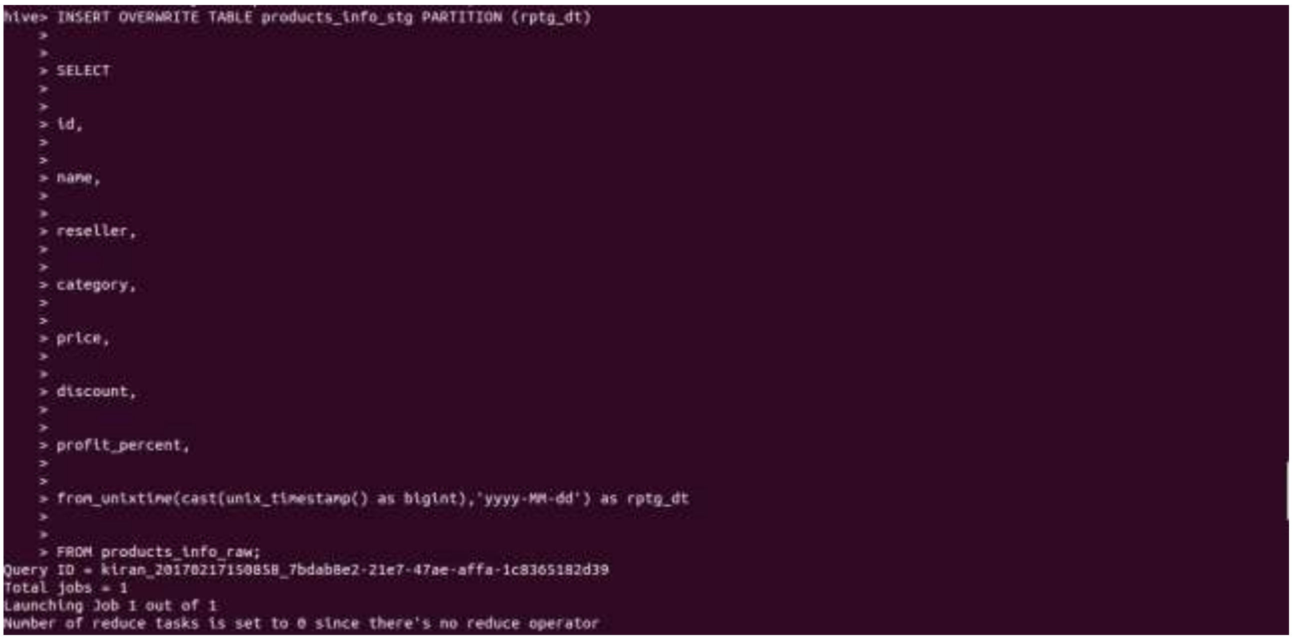


```
hive> select * from products_info_raw;
OK
P101    Men's Trimmer   Phillips India   Electronic Equipments   2399.0  10      5
P102    Men's Trimmer   Phillips Electronic Equipments   2399.0  10      5
Time taken: 0.064 seconds, Fetched: 2 row(s)
hive>
    > select * from users_info_raw;
OK
U101    Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL
        Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL
U103    Rakesh  {"city":"","state":""}  20      NULL
Time taken: 0.11 seconds, Fetched: 3 row(s)
hive>
    > select * from user_activity_raw;
OK
P101    U101    false   true    NA      Duplicate product       2015-10-18      2015-09-19      2015-09-20      NA      2015-09-25
P101    U101    false   true    NA      Duplicate product       2015-09-18      2015-09-19      2015-09-20      NA      2015-09-25
P103    U103    false   true    NA      Duplicate product       2015-09-18      2015-09-19      2015-09-20      NA      2015-09-25
Time taken: 0.073 seconds, Fetched: 3 row(s)
hive> 
```

Set the below property
set hive.exec.dynamic.partition.mode=nonstrict

Loading data into products_info_stg table

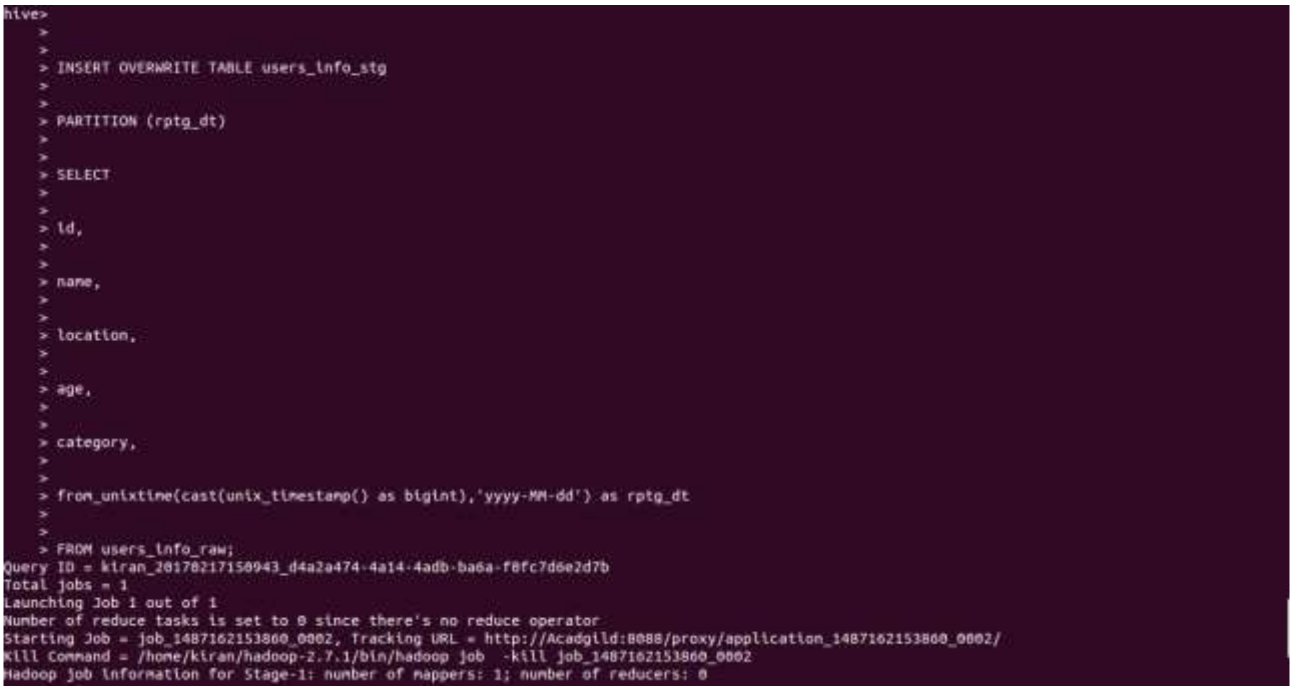
INSERT OVERWRITE TABLE products_info_stg PARTITION (rptg_dt) SELECT id, name, reseller, category, price, discount, profit_percent, from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt FROM products_info_raw;



```
hive> INSERT OVERWRITE TABLE products_info_stg PARTITION (rptg_dt)
    >
    >
    > SELECT
    >
    >
    > id,
    >
    >
    > name,
    >
    >
    > reseller,
    >
    >
    > category,
    >
    >
    > price,
    >
    >
    > discount,
    >
    >
    > profit_percent,
    >
    >
    > from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt
    >
    >
    > FROM products_info_raw;
Query ID = kiran_20170217150858_7bdab8e2-21e7-47ae-affa-1c8365182d39
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
```

Loading data into users_info_stg table

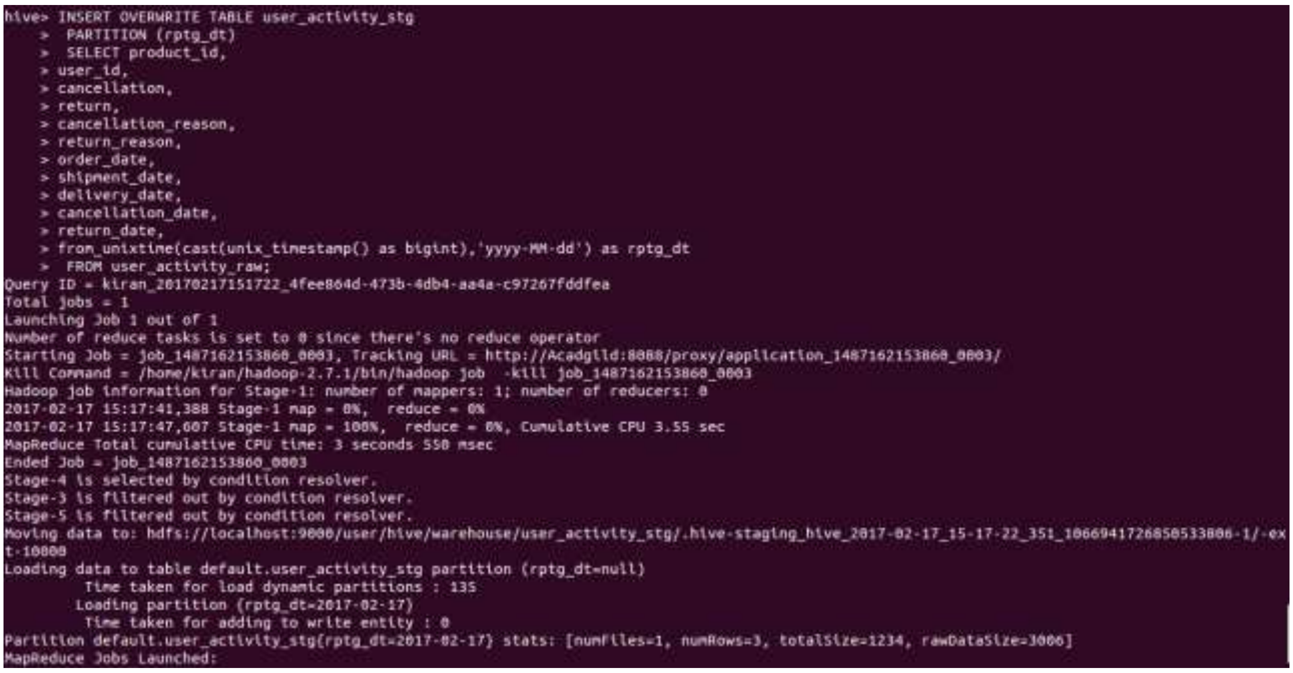
INSERT OVERWRITE TABLE users_info_stg PARTITION (rptg_dt) SELECT id,name, location, age, category, from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt FROM users_info_raw;



```
hive>
    >
    >
    > INSERT OVERWRITE TABLE users_info_stg
    >
    >
    > PARTITION (rptg_dt)
    >
    >
    > SELECT
    >
    >
    > id,
    >
    >
    > name,
    >
    >
    > location,
    >
    >
    > age,
    >
    >
    > category,
    >
    >
    > from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt
    >
    >
    > FROM users_info_raw;
Query ID = kiran_20170217150943_d4a2a474-4a14-4adb-ba6a-f0fc7d6e2d7b
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_1487162153860_0002, Tracking URL = http://Acadgild:8088/proxy/application_1487162153860_0002/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1487162153860_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
```

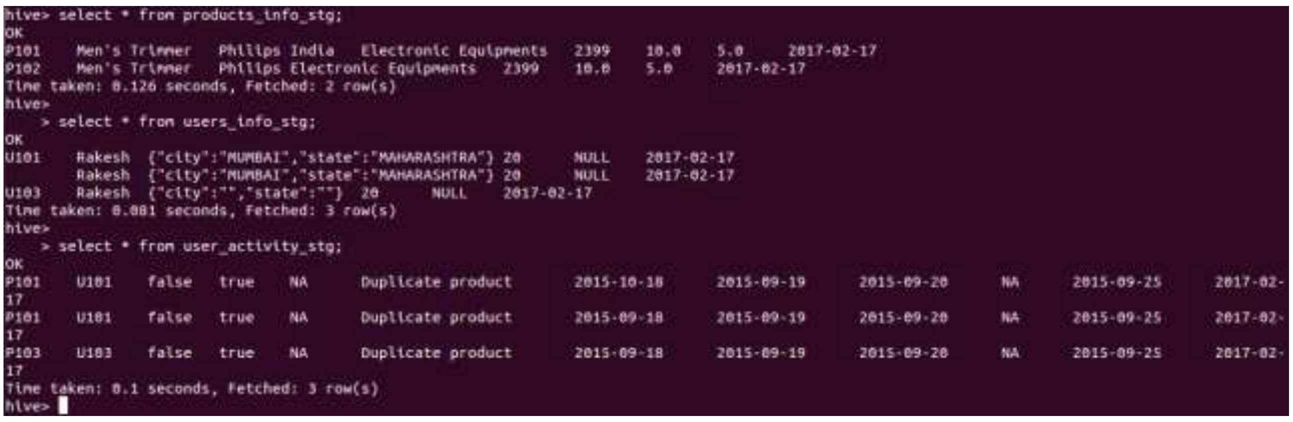
Loading data into user_activity_stg table

INSERT OVERWRITE TABLE user_activity_stg PARTITION (rptg_dt) SELECT product_id, user_id, cancellation, return, cancellation_reason, return_reason, order_date, shipment_date, delivery_date, cancellation_date, return_date, from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt FROM user_activity_raw;



```
hive> INSERT OVERWRITE TABLE user_activity_stg
    >  PARTITION (rptg_dt)
    >  SELECT product_id,
    > user_id,
    > cancellation,
    > return,
    > cancellation_reason,
    > return_reason,
    > order_date,
    > shipment_date,
    > delivery_date,
    > cancellation_date,
    > return_date,
    > from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') as rptg_dt
    >  FROM user_activity_raw;
Query ID = kiran_20170217151722_4fee864d-473b-4db4-aa4a-c97267fddfea
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_1487162153860_0003, Tracking URL = http://Acadgild:8088/proxy/application_1487162153860_0003/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1487162153860_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-02-17 15:17:41,388 Stage-1 map = 0%,  reduce = 0%
2017-02-17 15:17:47,607 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.55 sec
MapReduce Total cumulative CPU time: 3 seconds 550 msec
Ended Job = job_1487162153860_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: hdfs://localhost:9000/user/hive/warehouse/user_activity_stg/.hive-staging_hive_2017-02-17_15-17-22_351_1066941726850533806-1/-ex
t-10000
Loading data to table default.user_activity_stg partition (rptg_dt=null)
         Time taken for load dynamic partitions : 135
        Loading partition {rptg_dt=2017-02-17}
         Time taken for adding to write entity : 0
Partition default.user_activity_stg{rptg_dt=2017-02-17} stats: [numFiles=1, numRows=3, totalSize=1234, rawDataSize=3006]
MapReduce Jobs Launched:
```

Displaying the inserted data

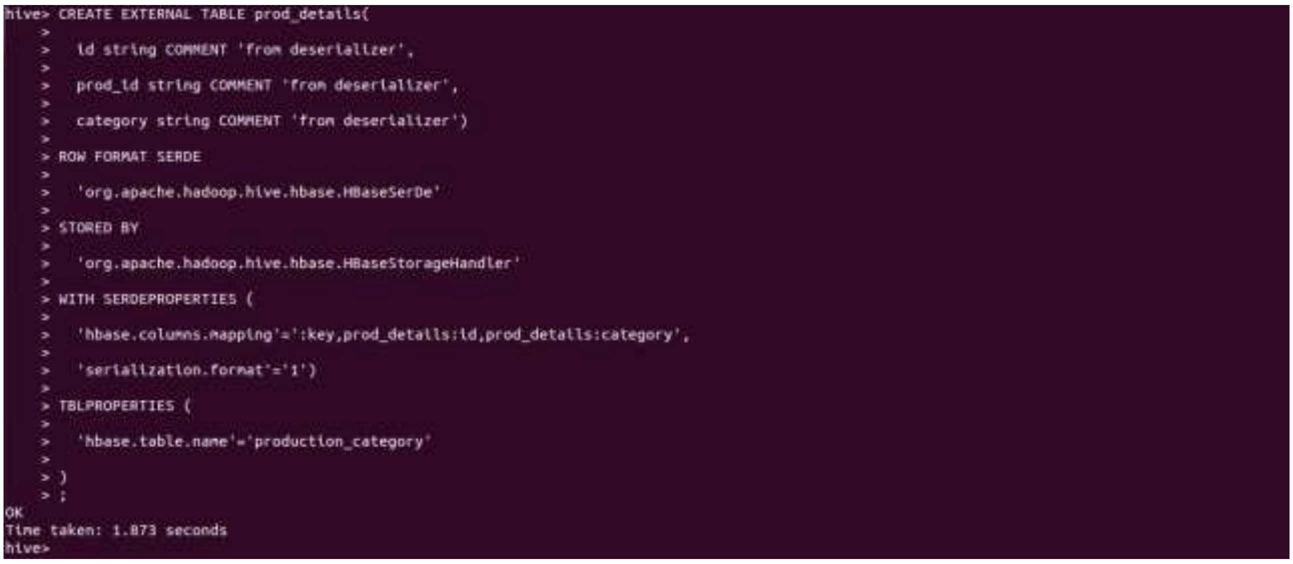


```
hive> select * from products_info_stg;
OK
P101    Men's Trimmer   Philips India   Electronic Equipments   2399    10.0    5.0     2017-02-17
P102    Men's Trimmer   Philips Electronic Equipments   2399    10.0    5.0     2017-02-17
Time taken: 0.126 seconds, Fetched: 2 row(s)
hive>
    > select * from users_info_stg;
OK
U101    Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL    2017-02-17
        Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL    2017-02-17
U103    Rakesh  {"city":"","state":""}  20      NULL    2017-02-17
Time taken: 0.081 seconds, Fetched: 3 row(s)
hive>
    > select * from user_activity_stg;
OK
P101    U101    false   true    NA      Duplicate product       2015-10-18      2015-09-19      2015-09-20      NA      2015-09-25      2017-02-
17
P101    U101    false   true    NA      Duplicate product       2015-09-18      2015-09-19      2015-09-20      NA      2015-09-25      2017-02-
17
P103    U103    false   true    NA      Duplicate product       2015-09-18      2015-09-19      2015-09-20      NA      2015-09-25      2017-02-
17
Time taken: 0.1 seconds, Fetched: 3 row(s)
hive>
```

creating prod_details table

```
CREATE EXTERNAL TABLE prod_details(
  id string COMMENT 'from deserializer',
  prod_id string COMMENT 'from deserializer',
  category string COMMENT 'from deserializer')
ROW FORMAT SERDE
  'org.apache.hadoop.hive.hbase.HBaseSerDe'
STORED BY
  'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
  'hbase.columns.mapping'=':key,prod_details:id,prod_details:category',
  'serialization.format'='1')
TBLPROPERTIES (
  'hbase.table.name'='production_category'
```

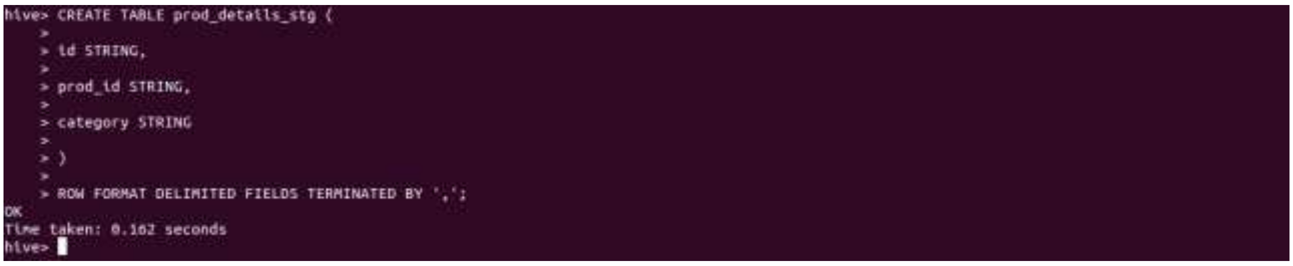
);



```
hive> CREATE EXTERNAL TABLE prod_details(
    >
    >   id string COMMENT 'from deserializer',
    >
    >   prod_id string COMMENT 'from deserializer',
    >
    >   category string COMMENT 'from deserializer')
    >
    > ROW FORMAT SERDE
    >
    >   'org.apache.hadoop.hive.hbase.HBaseSerDe'
    >
    > STORED BY
    >
    >   'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    > WITH SERDEPROPERTIES (
    >
    >   'hbase.columns.mapping'=':key,prod_details:id,prod_details:category',
    >
    >   'serialization.format'='1')
    >
    > TBLPROPERTIES (
    >
    >   'hbase.table.name'='production_category'
    >
    > )
    > ;
OK
Time taken: 1.873 seconds
hive>
```

Creating prod_details_stg table

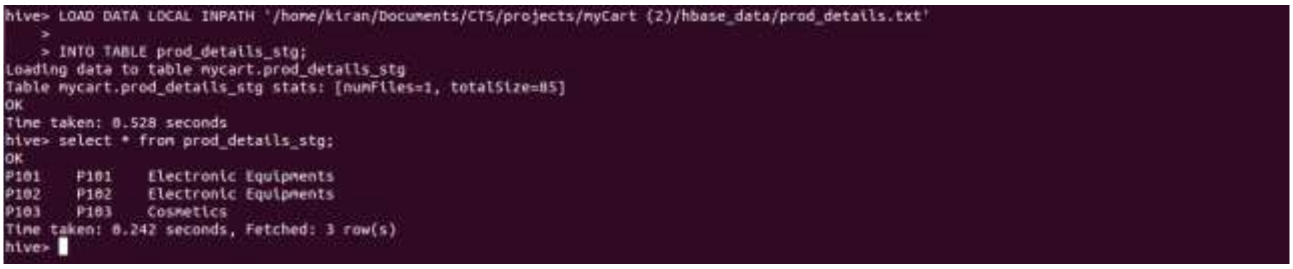
```
CREATE TABLE prod_details_stg (
id STRING,
prod_id STRING,
category STRING
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```



```
hive> CREATE TABLE prod_details_stg (
    >
    > id STRING,
    >
    > prod_id STRING,
    >
    > category STRING
    >
    > )
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.162 seconds
hive>
```

Inserting data into prod_details_stg table

```
LOAD DATA LOCAL INPATH '/home/acadgild/ecommerce/hbase_data/prod_details.txt'
INTO TABLE prod_details_stg;
```

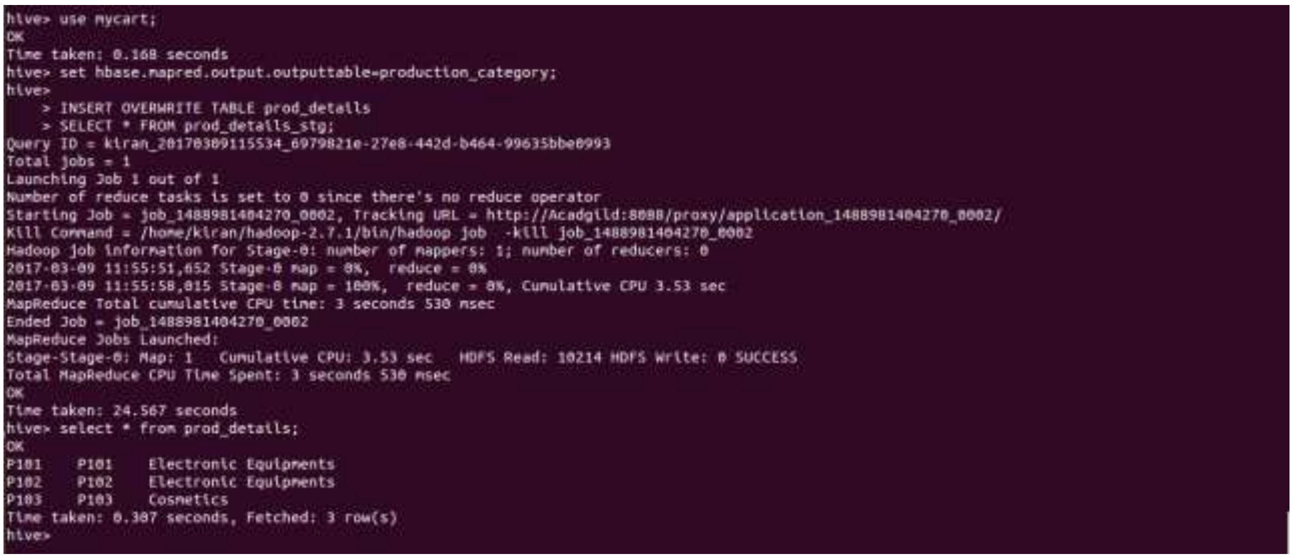


```
hive> LOAD DATA LOCAL INPATH '/home/kiran/Documents/CTS/projects/myCart (2)/hbase_data/prod_details.txt'
    >
    > INTO TABLE prod_details_stg;
Loading data to table mycart.prod_details_stg
Table mycart.prod_details_stg stats: [numFiles=1, totalSize=85]
OK
Time taken: 0.528 seconds
hive> select * from prod_details_stg;
OK
P101    P101    Electronic Equipments
P102    P102    Electronic Equipments
P103    P103    Cosmetics
Time taken: 0.242 seconds, Fetched: 3 row(s)
hive>
```

Inserting data into prod_details table

set hbase.mapred.output.outputtable=production_category;

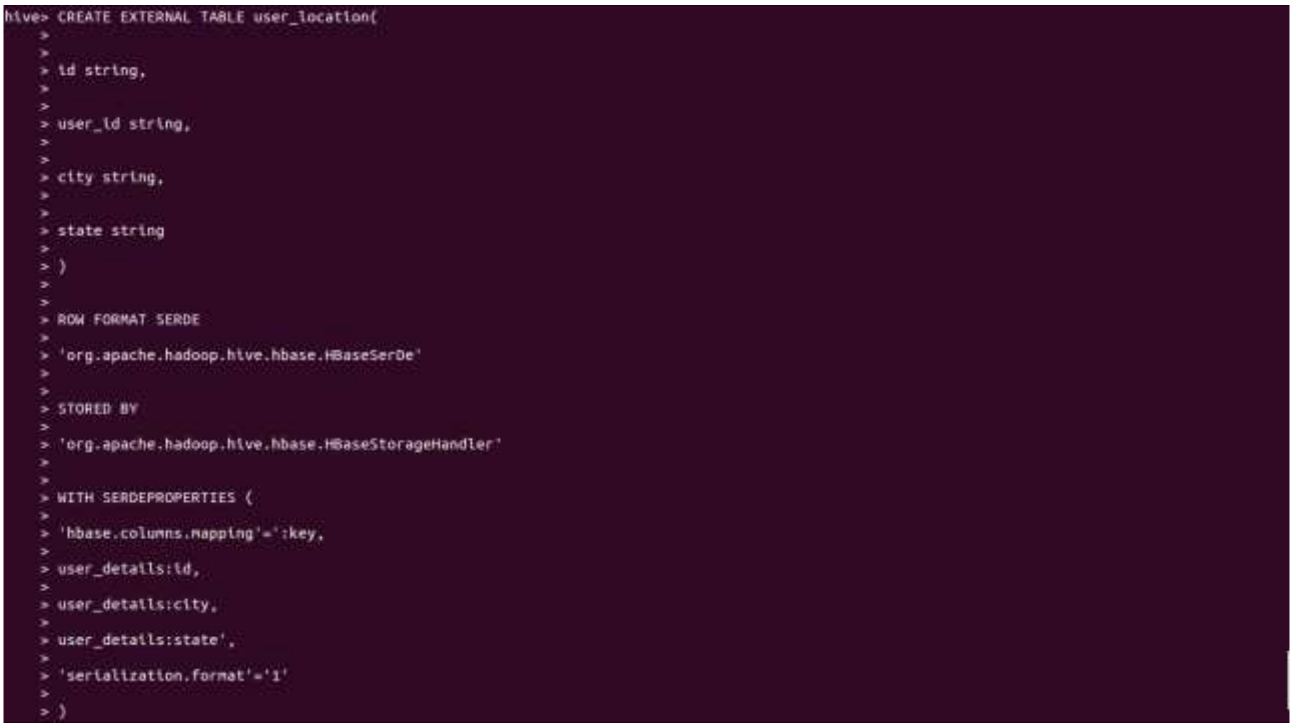
INSERT OVERWRITE TABLE prod_details
SELECT * FROM prod_details_stg;



```
hive> use mycart;
OK
Time taken: 0.168 seconds
hive> set hbase.mapred.output.outputtable=production_category;
hive>
    > INSERT OVERWRITE TABLE prod_details
    > SELECT * FROM prod_details_stg;
Query ID = kiran_20170309115534_6979821e-27e8-442d-b464-99635bbe0993
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_1488981404270_0002, Tracking URL = http://Acadgild:8088/proxy/application_1488981404270_0002/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1488981404270_0002
Hadoop job information for Stage-0: number of mappers: 1; number of reducers: 0
2017-03-09 11:55:51,652 Stage-0 map = 0%,  reduce = 0%
2017-03-09 11:55:58,015 Stage-0 map = 100%,  reduce = 0%, Cumulative CPU 3.53 sec
MapReduce Total cumulative CPU time: 3 seconds 530 msec
Ended Job = job_1488981404270_0002
MapReduce Jobs Launched:
Stage-Stage-0: Map: 1   Cumulative CPU: 3.53 sec   HDFS Read: 10214 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 530 msec
OK
Time taken: 24.567 seconds
hive> select * from prod_details;
OK
P101    P101    Electronic Equipments
P102    P102    Electronic Equipments
P103    P103    Cosmetics
Time taken: 0.387 seconds, Fetched: 3 row(s)
hive>
```

Creating user_location table

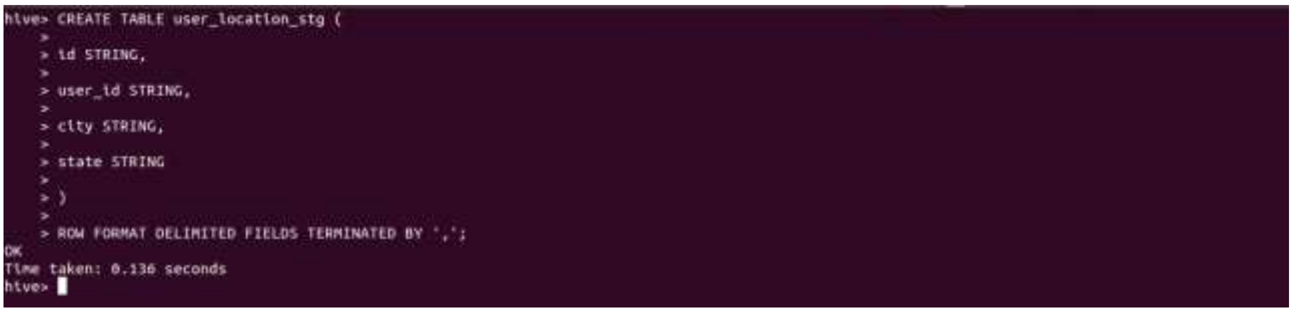
CREATE EXTERNAL TABLE user_location(id string, user_id string, city string, state string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.hbase.HBaseSerDe' STORED BY
'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES (
'hbase.columns.mapping'=':key,
user_details:id,
user_details:city,
user_details:state',
'serialization.format'='1'
) TBLPROPERTIES ('hbase.table.name'='user_location');



```
hive> CREATE EXTERNAL TABLE user_location(
    >
    >
    > id string,
    >
    >
    > user_id string,
    >
    >
    > city string,
    >
    >
    > state string
    >
    > )
    >
    >
    > ROW FORMAT SERDE
    >
    > 'org.apache.hadoop.hive.hbase.HBaseSerDe'
    >
    >
    > STORED BY
    >
    > 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    >
    >
    > WITH SERDEPROPERTIES (
    >
    > 'hbase.columns.mapping'=':key,
    >
    > user_details:id,
    >
    > user_details:city,
    >
    > user_details:state',
    >
    > 'serialization.format'='1'
    >
    > )
```

Creating user_location_stg table

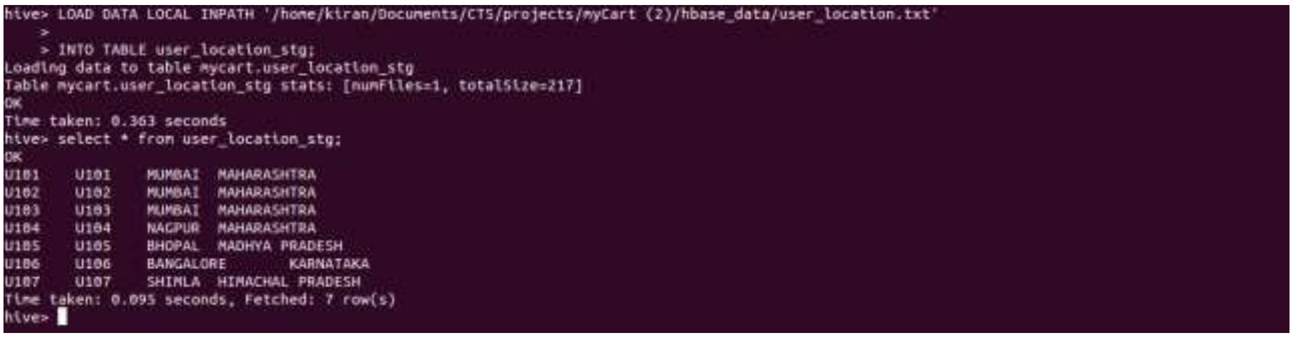
CREATE TABLE user_location_stg (
id STRING,
user_id STRING,
city STRING,
state STRING
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';



```
hive> CREATE TABLE user_location_stg (
    >
    > id STRING,
    >
    > user_id STRING,
    >
    > city STRING,
    >
    > state STRING
    >
    > )
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.136 seconds
hive> 
```

Inserting data into user_location_stg table

LOAD DATA LOCAL INPATH '/home/acadgild/ecommerce/hbase_data/user_location.txt'
INTO TABLE user_location_stg;

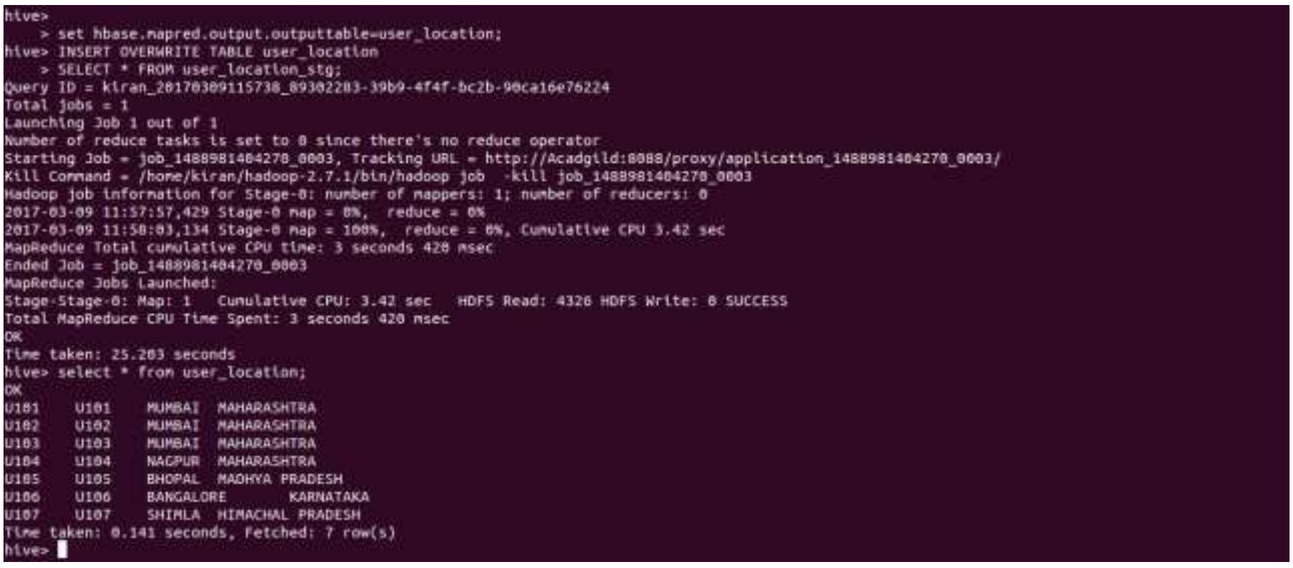


```
hive> LOAD DATA LOCAL INPATH '/home/kiran/Documents/CTS/projects/myCart (2)/hbase_data/user_location.txt'
    >
    > INTO TABLE user_location_stg;
Loading data to table mycart.user_location_stg
Table mycart.user_location_stg stats: [numFiles=1, totalSize=217]
OK
Time taken: 0.363 seconds
hive> select * from user_location_stg;
OK
U101    U101    MUMBAI  MAHARASHTRA
U102    U102    MUMBAI  MAHARASHTRA
U103    U103    MUMBAI  MAHARASHTRA
U104    U104    NAGPUR  MAHARASHTRA
U105    U105    BHOPAL  MADHYA PRADESH
U106    U106    BANGALORE       KARNATAKA
U107    U107    SHIMLA  HIMACHAL PRADESH
Time taken: 0.095 seconds, Fetched: 7 row(s)
hive>
```

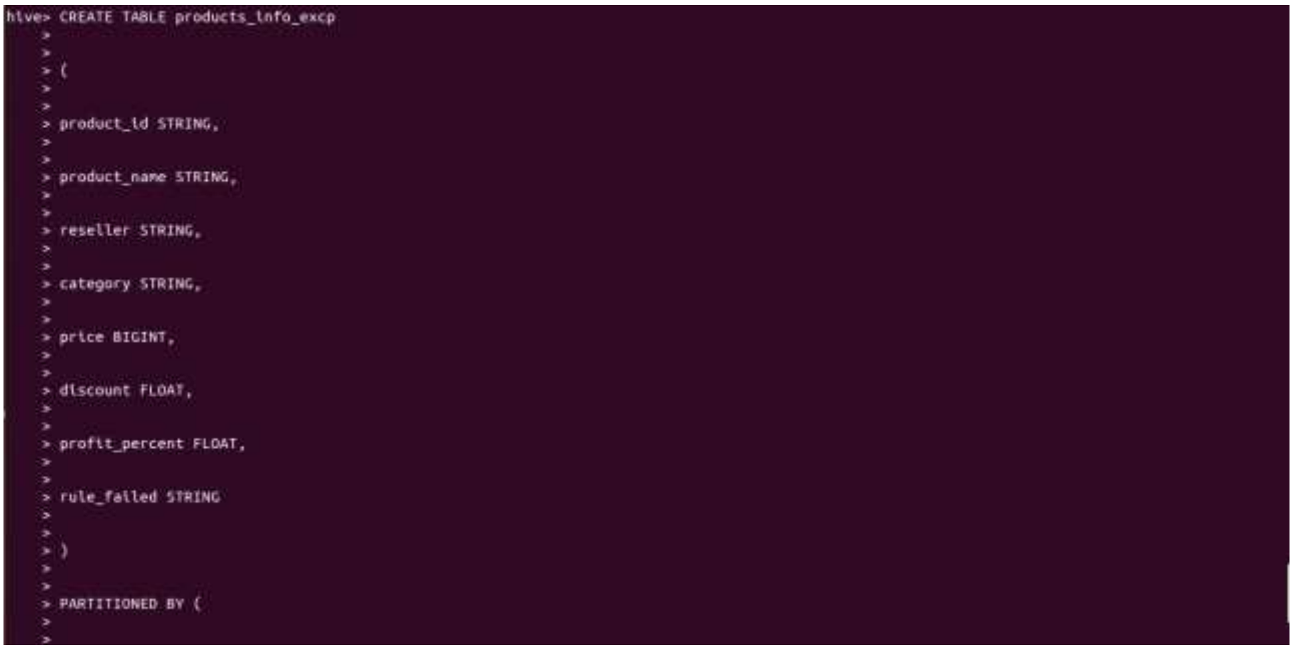
Inserting data into user_lcoation table

set hbase.mapred.output.outputtable=user_location;

INSERT OVERWRITE TABLE user_location
SELECT * FROM user_location_stg;



```
hive>
    > set hbase.mapred.output.outputtable=user_location;
hive> INSERT OVERWRITE TABLE user_location
    > SELECT * FROM user_location_stg;
Query ID = kiran_20170309115738_89302283-39b9-4f4f-bc2b-90ca16e76224
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_1488981404270_0003, Tracking URL = http://Acadgild:8088/proxy/application_1488981404270_0003/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1488981404270_0003
Hadoop job information for Stage-0: number of mappers: 1; number of reducers: 0
2017-03-09 11:57:57,429 Stage-0 map = 0%,  reduce = 0%
2017-03-09 11:58:03,134 Stage-0 map = 100%,  reduce = 0%, Cumulative CPU 3.42 sec
MapReduce Total cumulative CPU time: 3 seconds 420 msec
Ended Job = job_1488981404270_0003
MapReduce Jobs Launched:
Stage-Stage-0: Map: 1   Cumulative CPU: 3.42 sec   HDFS Read: 4326 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 420 msec
OK
Time taken: 25.203 seconds
hive> select * from user_location;
OK
U101    U101    MUMBAI  MAHARASHTRA
U102    U102    MUMBAI  MAHARASHTRA
U103    U103    MUMBAI  MAHARASHTRA
U104    U104    NAGPUR  MAHARASHTRA
U105    U105    BHOPAL  MADHYA PRADESH
U106    U106    BANGALORE       KARNATAKA
U107    U107    SHIMLA  HIMACHAL PRADESH
Time taken: 0.141 seconds, Fetched: 7 row(s)
hive>
```



```
hive> CREATE TABLE products_info_excp
    >
    >
    > (
    >
    >
    > product_id STRING,
    >
    >
    > product_name STRING,
    >
    >
    > reseller STRING,
    >
    >
    > category STRING,
    >
    >
    > price BIGINT,
    >
    >
    > discount FLOAT,
    >
    >
    > profit_percent FLOAT,
    >
    >
    > rule_failed STRING
    >
    >
    > )
    >
    >
    > PARTITIONED BY (
    >
    >
```

Inserting data into products_info_excp and products_info_core tables

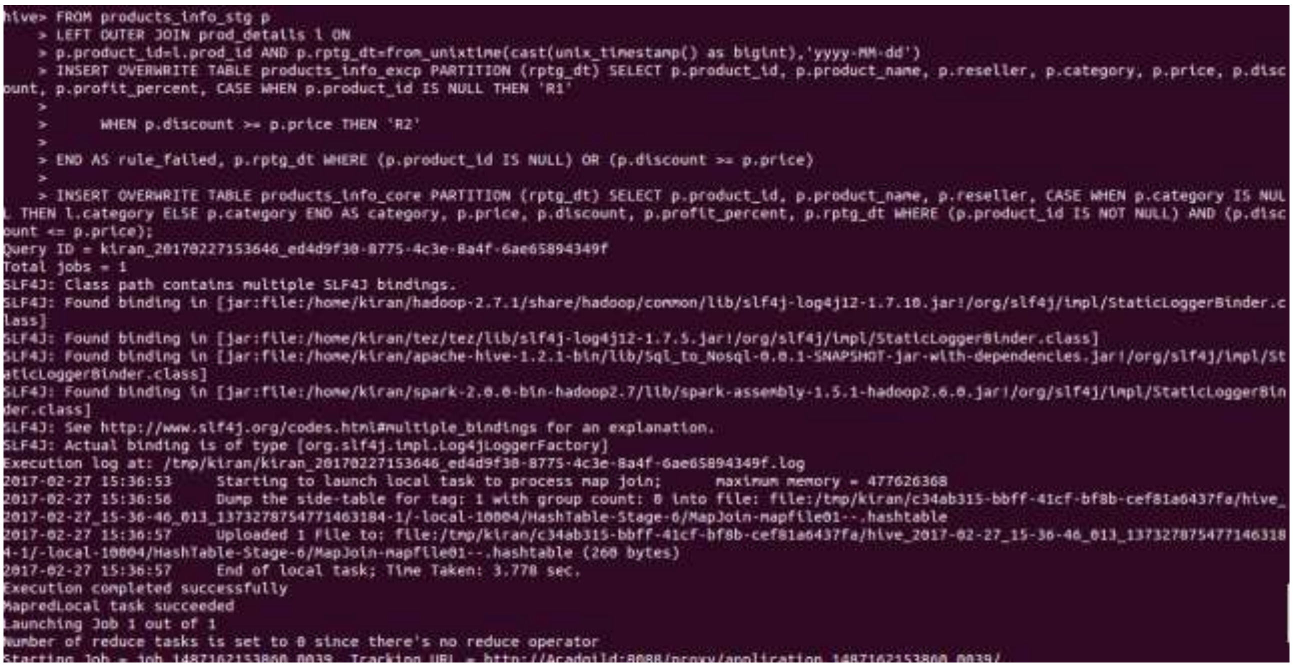
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.auto.convert.join=false;

FROM products_info_stg p
LEFT OUTER JOIN prod_details l ON
p.product_id=l.prod_id AND p.rptg_dt=from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd')
INSERT OVERWRITE TABLE products_info_excp PARTITION (rptg_dt) SELECT p.product_id, p.product_name, p.reseller, p.category, p.price, p.discount, p.profit_percent, CASE WHEN p.product_id IS NULL THEN 'R1'

WHEN p.discount >= p.price THEN 'R2'

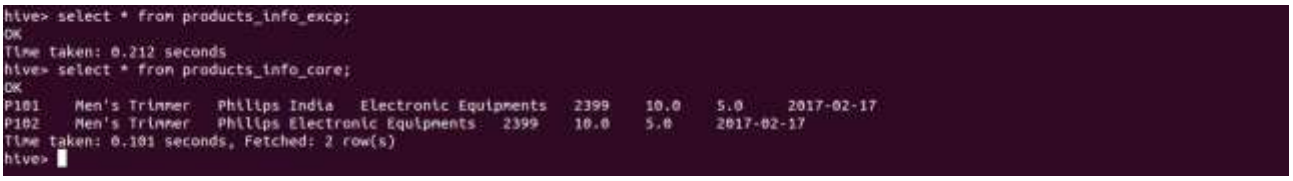
END AS rule_failed, p.rptg_dt WHERE (p.product_id IS NULL) OR (p.discount >= p.price)

INSERT OVERWRITE TABLE products_info_core PARTITION (rptg_dt) SELECT p.product_id, p.product_name, p.reseller, CASE WHEN p.category IS NULL THEN l.category ELSE p.category END AS category, p.price, p.discount, p.profit_percent, p.rptg_dt WHERE (p.product_id IS NOT NULL) AND (p.discount <= p.price);



```
hive> FROM products_info_stg p
    > LEFT OUTER JOIN prod_details l ON
    > p.product_id=l.prod_id AND p.rptg_dt=from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd')
    > INSERT OVERWRITE TABLE products_info_excp PARTITION (rptg_dt) SELECT p.product_id, p.product_name, p.reseller, p.category, p.price, p.disc
ount, p.profit_percent, CASE WHEN p.product_id IS NULL THEN 'R1'
    >
    >      WHEN p.discount >= p.price THEN 'R2'
    >
    > END AS rule_failed, p.rptg_dt WHERE (p.product_id IS NULL) OR (p.discount >= p.price)
    >
    > INSERT OVERWRITE TABLE products_info_core PARTITION (rptg_dt) SELECT p.product_id, p.product_name, p.reseller, CASE WHEN p.category IS NUL
L THEN l.category ELSE p.category END AS category, p.price, p.discount, p.profit_percent, p.rptg_dt WHERE (p.product_id IS NOT NULL) AND (p.disc
ount <= p.price);
Query ID = kiran_20170227153646_ed4d9f30-8775-4c3e-8a4f-6ae65894349f
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/Sql_to_Nosql-0.0.1-SNAPSHOT-jar-with-dependencies.jar!/org/slf4j/impl/St
aticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Execution log at: /tmp/kiran/kiran_20170227153646_ed4d9f30-8775-4c3e-8a4f-6ae65894349f.log
2017-02-27 15:36:53     Starting to launch local task to process map join;      maximum memory = 477626368
2017-02-27 15:36:56     Dump the side-table for tag: 1 with group count: 0 into file: file:/tmp/kiran/c34ab315-bbff-41cf-bf8b-cef81a6437fa/hive_
2017-02-27_15-36-46_013_1373278754771463184-1/-local-10004/HashTable-Stage-6/MapJoin-mapfile01--.hashtable
2017-02-27 15:36:57     Uploaded 1 File to: file:/tmp/kiran/c34ab315-bbff-41cf-bf8b-cef81a6437fa/hive_2017-02-27_15-36-46_013_137327875477146318
4-1/-local-10004/HashTable-Stage-6/MapJoin-mapfile01--.hashtable (260 bytes)
2017-02-27 15:36:57     End of local task; Time Taken: 3.778 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_1487162153860_0039, Tracking URL = http://Acadgild:8088/proxy/application_1487162153860_0039/
```

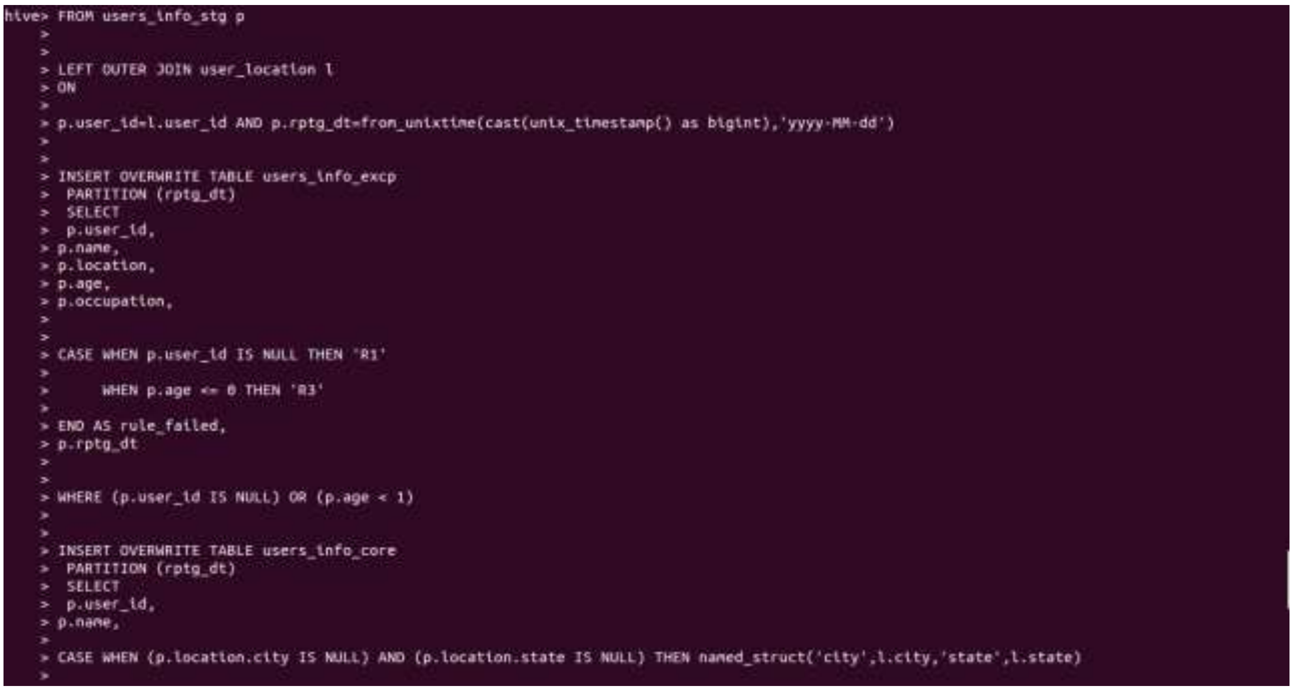
Displaying the results



```
hive> select * from products_info_excp;
OK
Time taken: 0.212 seconds
hive> select * from products_info_core;
OK
P101    Men's Trimmer   Philips India   Electronic Equipments   2399    10.0    5.0     2017-02-17
P102    Men's Trimmer   Philips Electronic Equipments   2399    10.0    5.0     2017-02-17
Time taken: 0.101 seconds, Fetched: 2 row(s)
hive> █
```

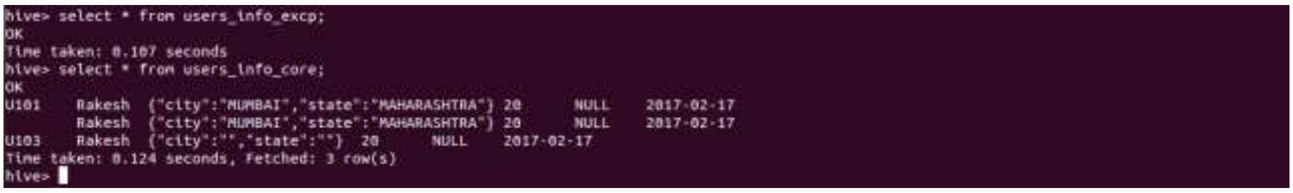
Inserting data into users_info_excp and users_info_core tables

```
FROM users_info_stg p
LEFT OUTER JOIN user_location l ON
p.user_id=l.user_id AND p.rptg_dt=from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd')
INSERT OVERWRITE TABLE users_info_excp PARTITION (rptg_dt) SELECT p.user_id, p.name, p.location, p.age, p.occupation,
CASE WHEN p.user_id IS NULL THEN 'R1'
    WHEN p.age <= 0 THEN 'R3'
END AS rule_failed, p.rptg_dt
WHERE (p.user_id IS NULL) OR (p.age < 1)
INSERT OVERWRITE TABLE users_info_core PARTITION (rptg_dt) SELECT p.user_id, p.name,
CASE WHEN (p.location.city IS NULL) AND (p.location.state IS NULL) THEN named_struct('city',l.city,'state',l.state)
    WHEN (p.location.city IS NULL) AND (p.location.state IS NOT NULL) THEN named_struct('city',l.city,'state',p.location.state)
    WHEN (p.location.city IS NOT NULL) AND (p.location.state IS NULL) THEN named_struct('city',p.location.city,'state',l.state) ELSE p.location END AS location, p.age, p.occupation, p.rptg_dt WHERE (p.user_id IS NOT NULL) AND (p.age >= 1);
```



```
hive> FROM users_info_stg p
    >
    >
    > LEFT OUTER JOIN user_location l
    > ON
    >
    > p.user_id=l.user_id AND p.rptg_dt=from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd')
    >
    >
    > INSERT OVERWRITE TABLE users_info_excp
    >  PARTITION (rptg_dt)
    >  SELECT
    >  p.user_id,
    > p.name,
    > p.location,
    > p.age,
    > p.occupation,
    >
    >
    > CASE WHEN p.user_id IS NULL THEN 'R1'
    >
    >      WHEN p.age <= 0 THEN 'R3'
    >
    > END AS rule_failed,
    > p.rptg_dt
    >
    >
    > WHERE (p.user_id IS NULL) OR (p.age < 1)
    >
    >
    > INSERT OVERWRITE TABLE users_info_core
    >  PARTITION (rptg_dt)
    >  SELECT
    >  p.user_id,
    > p.name,
    >
    > CASE WHEN (p.location.city IS NULL) AND (p.location.state IS NULL) THEN named_struct('city',l.city,'state',l.state)
    >
```

Displaying the data



```
hive> select * from users_info_excp;
OK
Time taken: 0.107 seconds
hive> select * from users_info_core;
OK
U101    Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL    2017-02-17
        Rakesh  {"city":"MUMBAI","state":"MAHARASHTRA"} 20      NULL    2017-02-17
U103    Rakesh  {"city":"","state":""}  20      NULL    2017-02-17
Time taken: 0.124 seconds, Fetched: 3 row(s)
hive> 
```

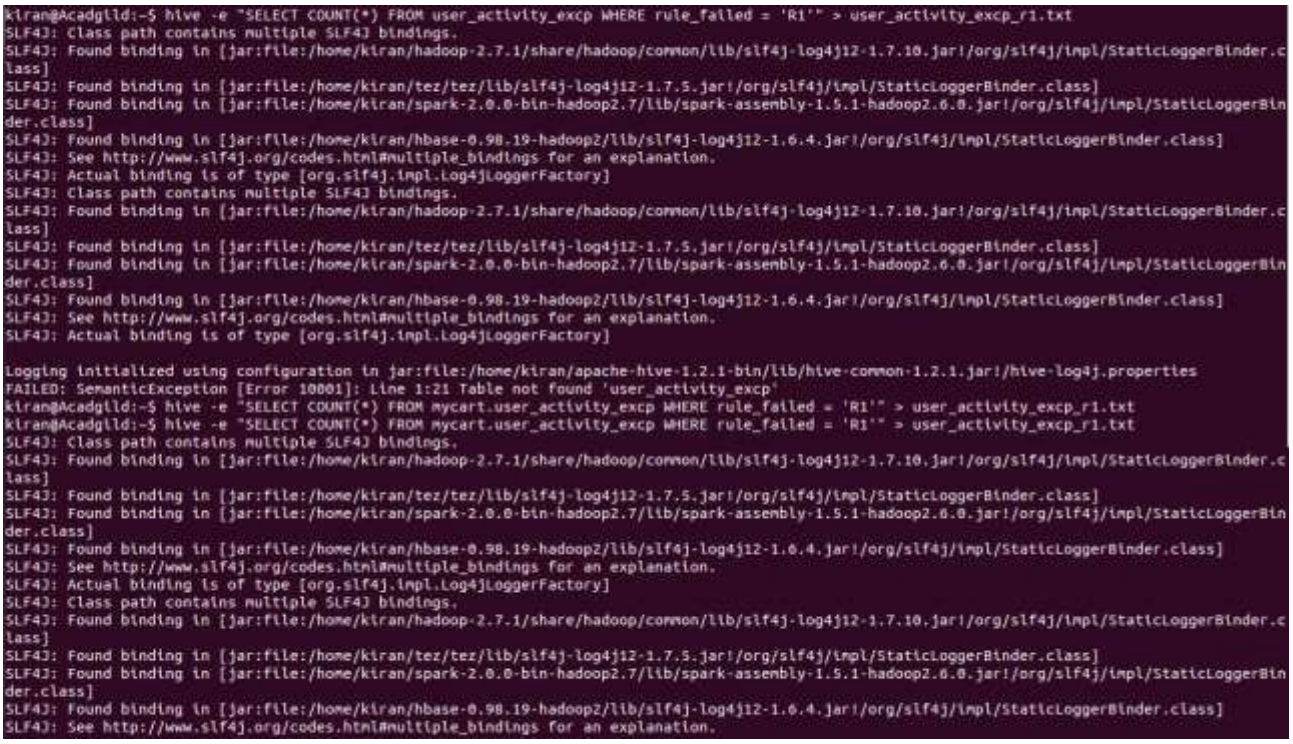
Inserting data into user_activity_excp and user_activity_core tables

FROM user_activity_stg p
LEFT OUTER JOIN user_location l ON p.user_id=l.user_id AND
p.rptg_dt=from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') LEFT OUTER JOIN prod_details pd
ON p.product_id=pd.id
INSERT OVERWRITE TABLE user_activity_excp PARTITION (rptg_dt) SELECT p.product_id, p.user_id,
p.cancellation, p.return, p.cancellation_reason, p.return_reason, p.order_date, p.shipment_date,
p.delivery_date, p.cancellation_date, p.return_date,
CASE WHEN (p.product_id IS NULL) OR (p.user_id IS NULL) THEN 'R1'
WHEN (p.order_date > p.shipment_date) THEN 'R2' ELSE 'NA' END AS rule_failed , p.rptg_dt
WHERE (p.user_id IS NULL) OR (p.product_id IS NULL) OR (p.order_date > p.shipment_date)
INSERT OVERWRITE TABLE user_activity_core PARTITION (rptg_dt) SELECT p.product_id, p.user_id,
p.cancellation, p.return, p.cancellation_reason, p.return_reason, p.order_date, p.shipment_date,
p.delivery_date, p.cancellation_date, p.return_date, p.rptg_dt
WHERE (p.user_id IS NOT NULL) AND (p.product_id IS NOT NULL) AND (p.order_date <=
p.shipment_date);

Data validation & Rules checking

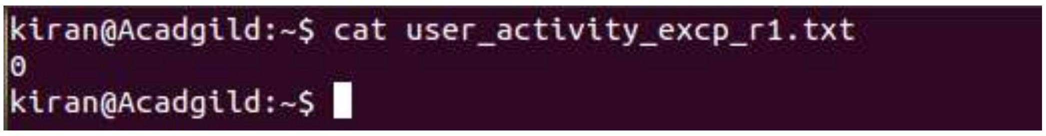
Rules checking on user_acitivity_excp table

1.
hive -e "SELECT COUNT(*) FROM ecom.user_activity_excp WHERE rule_failed = 'R1'" > user_activity_excp_r1.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM user_activity_excp WHERE rule_failed = 'R1'" > user_activity_excp_r1.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
FAILED: SemanticException [Error 10001]: Line 1:21 Table not found 'user_activity_excp'
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.user_activity_excp WHERE rule_failed = 'R1'" > user_activity_excp_r1.txt
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.user_activity_excp WHERE rule_failed = 'R1'" > user_activity_excp_r1.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
```

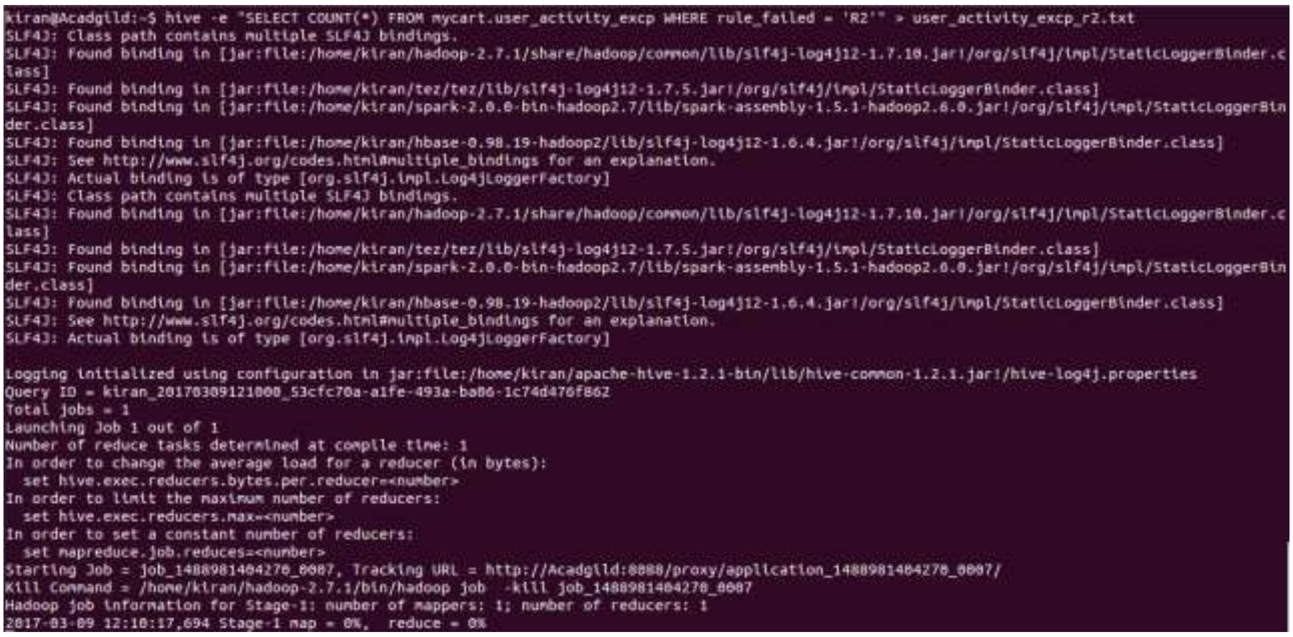
Checking the data in user_activity_excp_r1.txt file



```
kiran@Acadgild:~$ cat user_activity_excp_r1.txt
0
kiran@Acadgild:~$
```

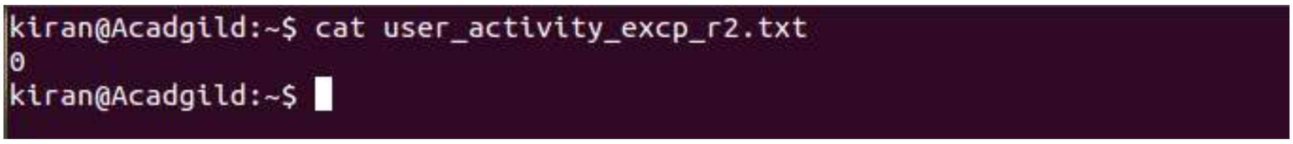
2.

hive -e "SELECT COUNT(*) FROM ecom.user_activity_excp WHERE rule_failed = 'R2'" > user_activity_excp_r2.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.user_activity_excp WHERE rule_failed = 'R2'" > user_activity_excp_r2.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309121000_53cfc70a-a1fe-493a-ba86-1c74d476f862
Total jobs = 1
Launching Job 1 out of 1
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_1488981404270_0007, Tracking URL = http://Acadgild:8088/proxy/application_1488981404270_0007/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1488981404270_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 12:10:17,694 Stage-1 map = 0%,  reduce = 0%
```

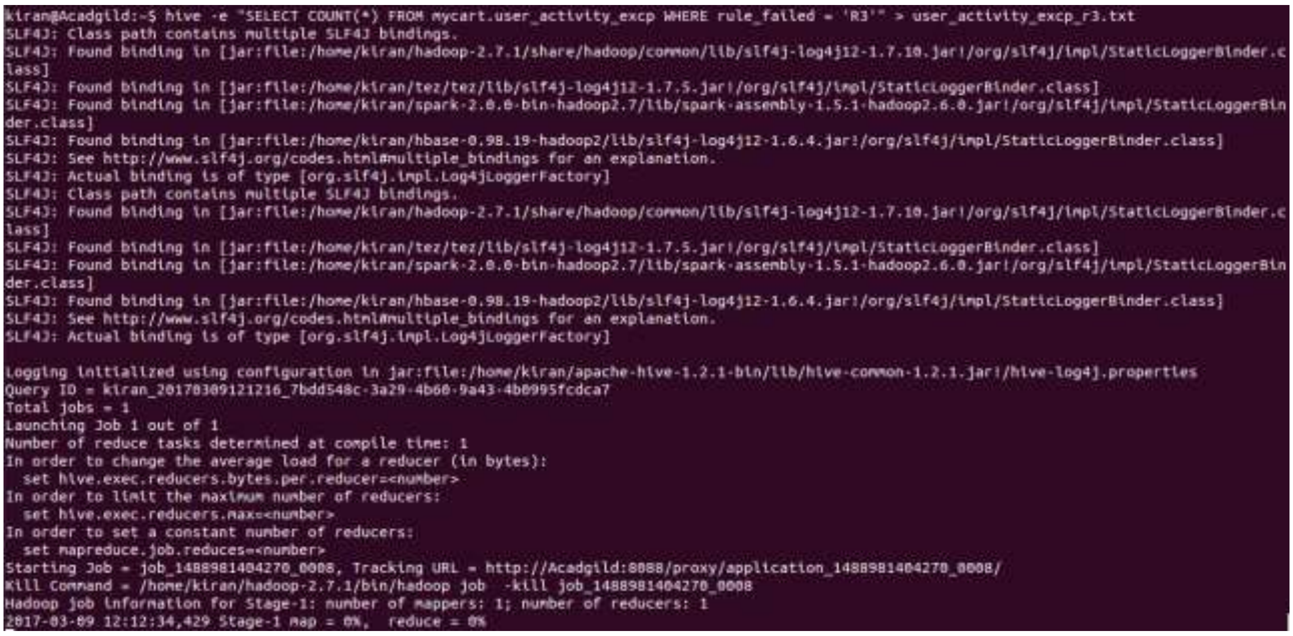
Checking the data in user_excp_r2.txt



```
kiran@Acadgild:~$ cat user_activity_excp_r2.txt
0
kiran@Acadgild:~$ 
```

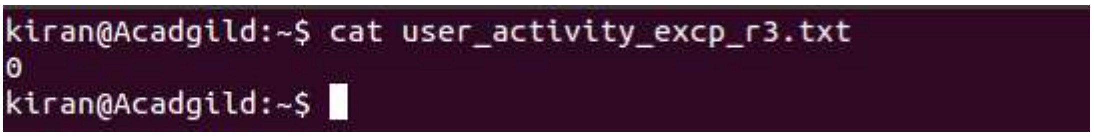
3.

hive -e "SELECT COUNT(*) FROM ecom.user_activity_excp WHERE rule_failed = 'R3'" > user_activity_excp_r3.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.user_activity_excp WHERE rule_failed = 'R3'" > user_activity_excp_r3.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309121216_7bdd548c-3a29-4b60-9a43-4b0995fcdca7
Total jobs = 1
Launching Job 1 out of 1
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_1488981404270_0008, Tracking URL = http://Acadgild:8088/proxy/application_1488981404270_0008/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1488981404270_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 12:12:34,429 Stage-1 map = 0%,  reduce = 0%
```

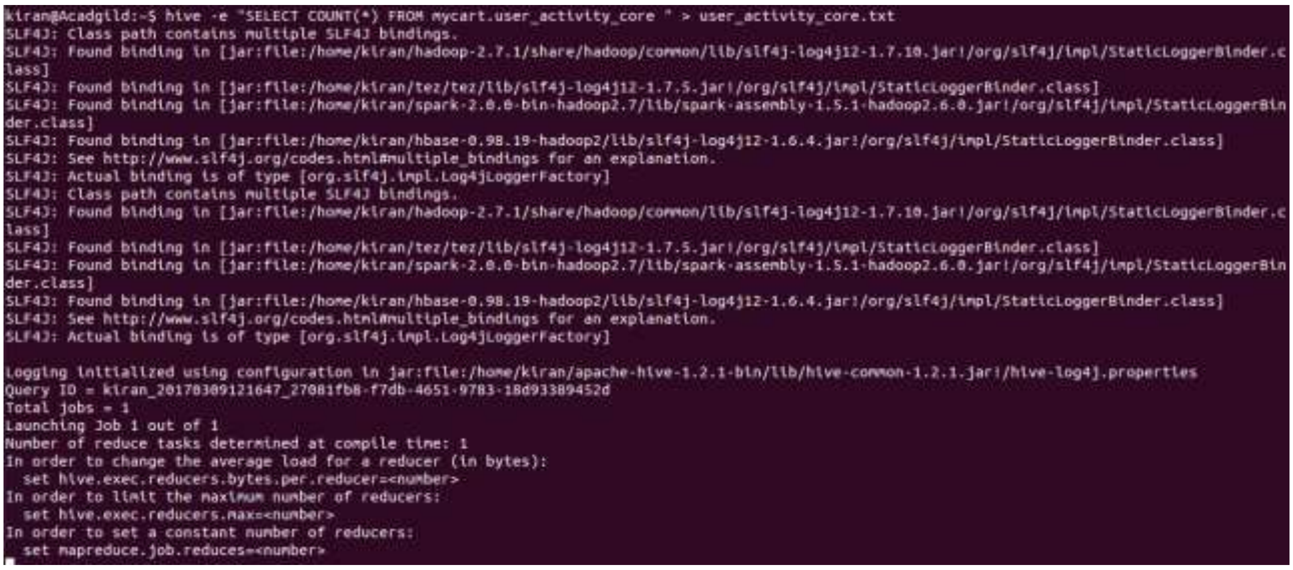
Checking the data in user_excp_r3.txt



```
kiran@Acadgild:~$ cat user_activity_excp_r3.txt
0
kiran@Acadgild:~$ 
```

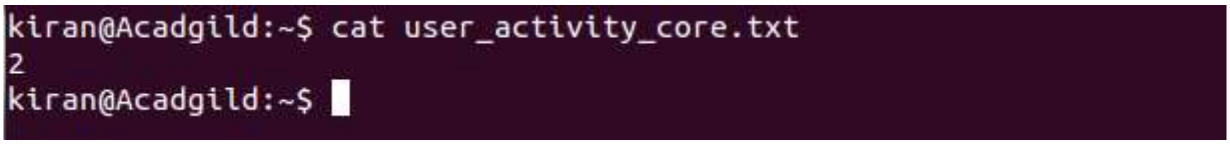
Checking the count of rows in user_activity_core table

hive -e "SELECT COUNT(*) FROM ecom.user_activity_core " > user_activity_core.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.user_activity_core " > user_activity_core.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309121647_27081fb8-f7db-4651-9783-18d93389452d
Total jobs = 1
Launching Job 1 out of 1
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>
```

Checking the data in user_activity_core

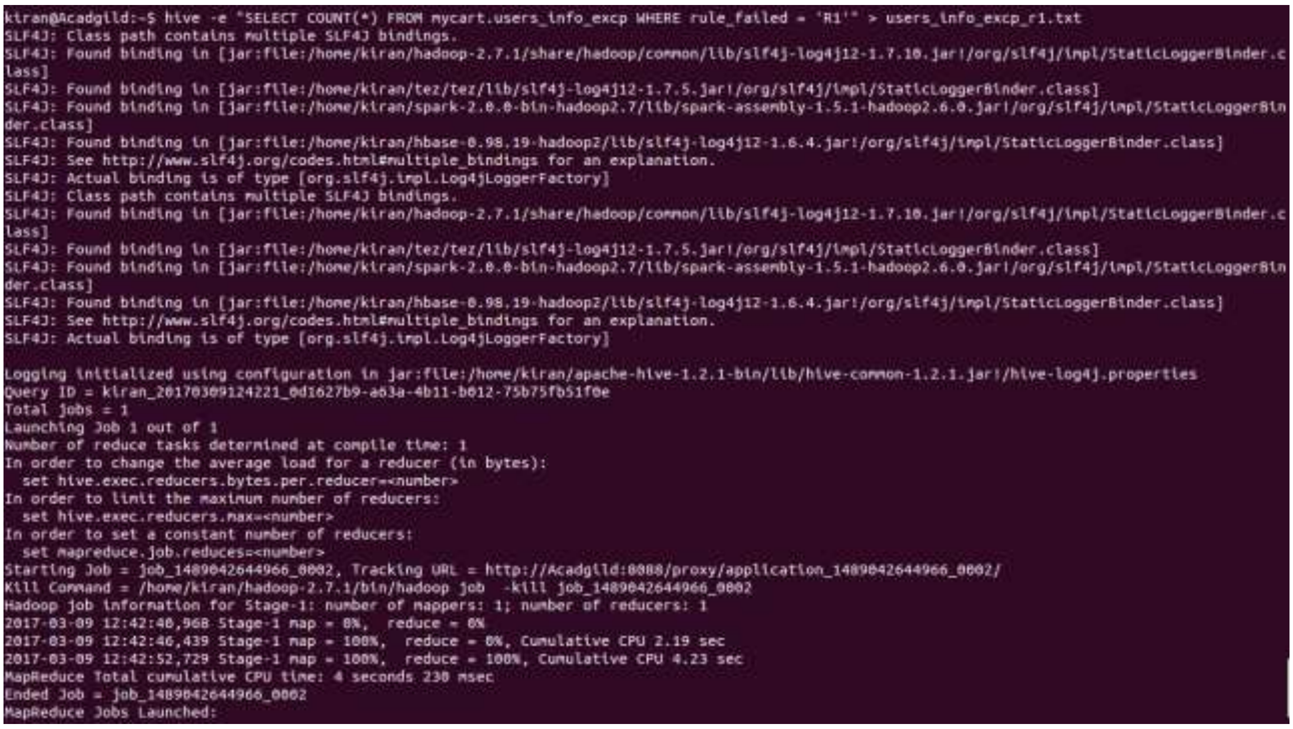


```
kiran@Acadgild:~$ cat user_activity_core.txt
2
kiran@Acadgild:~$ 
```

Rules checking on user_info table

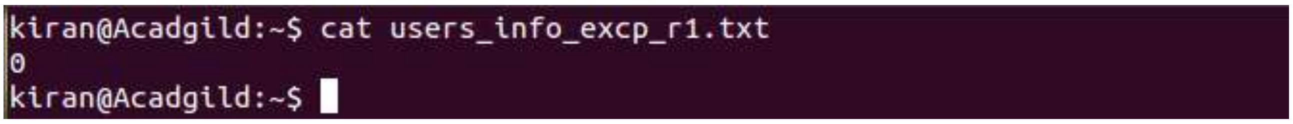
1.

hive -e "SELECT COUNT(*) FROM ecom.users_info_excp WHERE rule_failed = 'R1'" > users_info_excp_r1.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.users_info_excp WHERE rule_failed = 'R1'" > users_info_excp_r1.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309124221_0d1627b9-a63a-4b11-b012-75b75fb51f0e
Total jobs = 1
Launching Job 1 out of 1
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_1489042644966_0002, Tracking URL = http://Acadgild:8088/proxy/application_1489042644966_0002/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489042644966_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 12:42:40,968 Stage-1 map = 0%,  reduce = 0%
2017-03-09 12:42:46,439 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.19 sec
2017-03-09 12:42:52,729 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.23 sec
MapReduce Total cumulative CPU time: 4 seconds 230 msec
Ended Job = job_1489042644966_0002
MapReduce Jobs Launched:
```

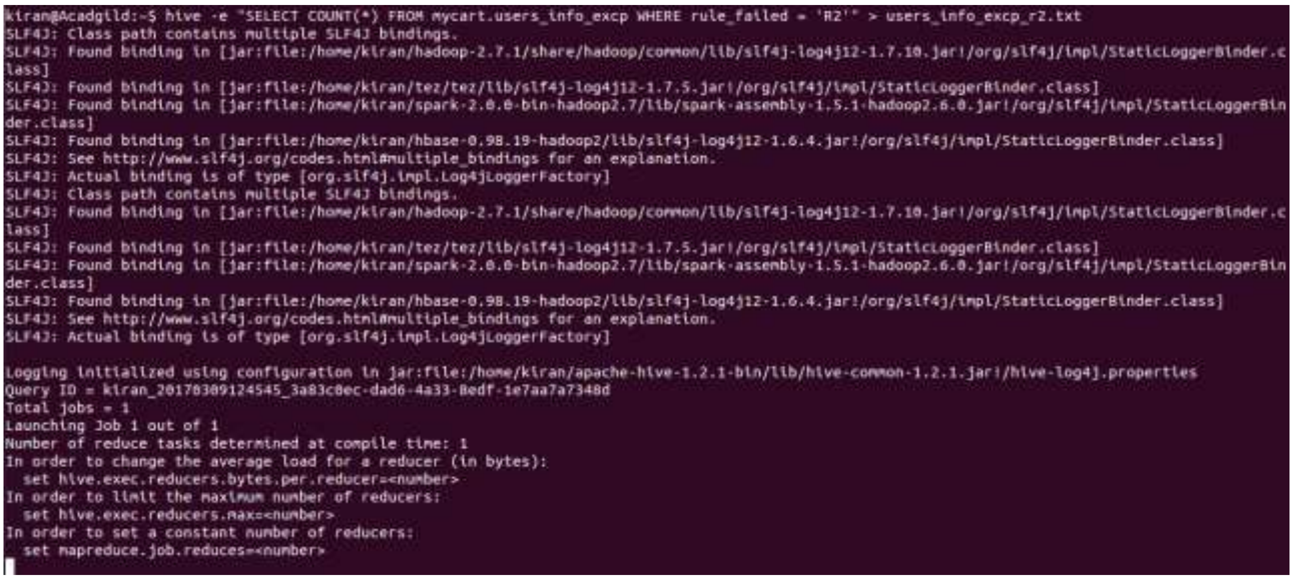
Checking the data in users_info_excp_r1.txt



```
kiran@Acadgild:~$ cat users_info_excp_r1.txt
0
kiran@Acadgild:~$ 
```

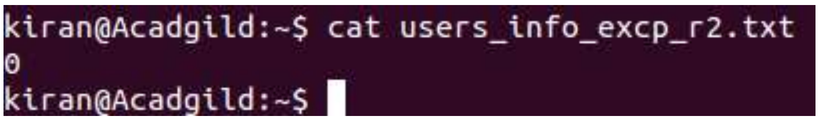
2.

hive -e "SELECT COUNT(*) FROM ecom.users_info_excp WHERE rule_failed = 'R2'" > users_info_excp_r2.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.users_info_excp WHERE rule_failed = 'R2'" > users_info_excp_r2.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309124545_3a83c8ec-dad6-4a33-8edf-1e7aa7a7348d
Total jobs = 1
Launching Job 1 out of 1
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>
```

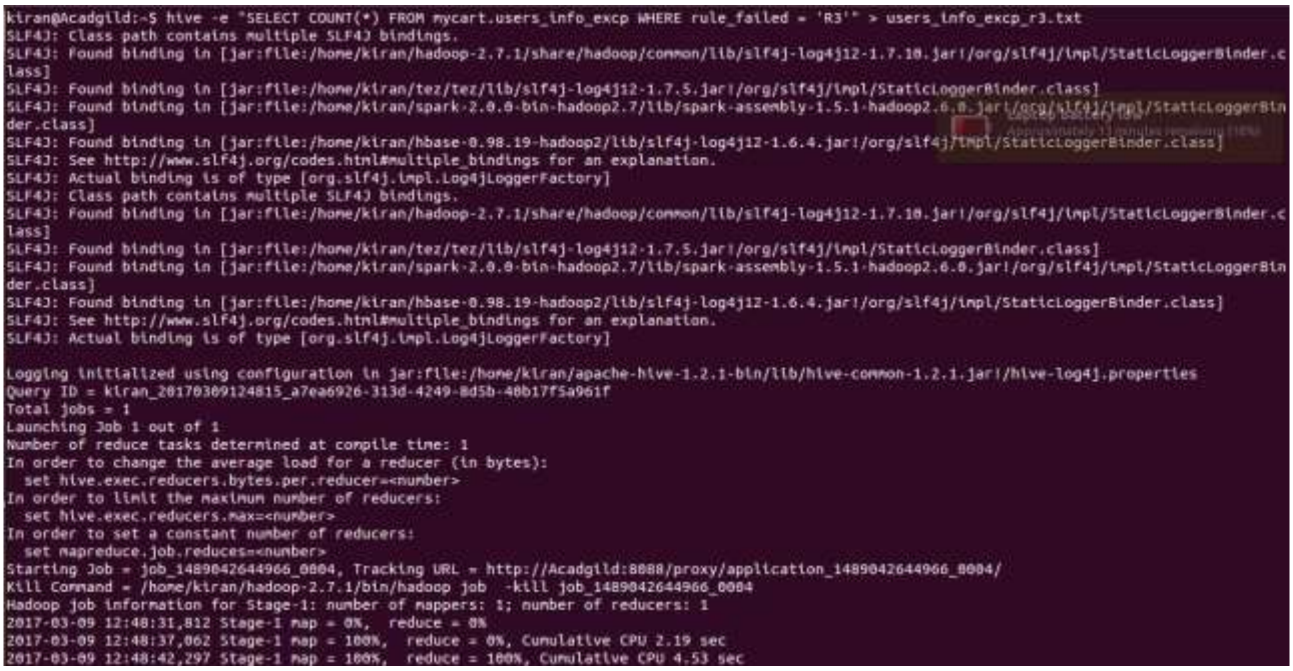
Checking the data in users_info_excp_r2.txt



```
kiran@Acadgild:~$ cat users_info_excp_r2.txt
0
kiran@Acadgild:~$ 
```

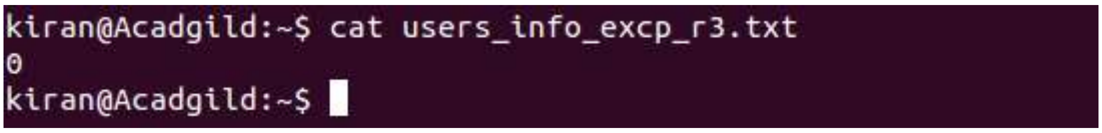
3.

hive -e "SELECT COUNT(*) FROM ecom.users_info_excp WHERE rule_failed = 'R3'" > users_info_excp_r3.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.users_info_excp WHERE rule_failed = 'R3'" > users_info_excp_r3.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309124815_a7ea6926-313d-4249-8d5b-40b17f5a961f
Total jobs = 1
Launching Job 1 out of 1
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_1489042644966_0004, Tracking URL = http://Acadgild:8088/proxy/application_1489042644966_0004/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489042644966_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 12:48:31,812 Stage-1 map = 0%,  reduce = 0%
2017-03-09 12:48:37,062 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.19 sec
2017-03-09 12:48:42,297 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.53 sec
```

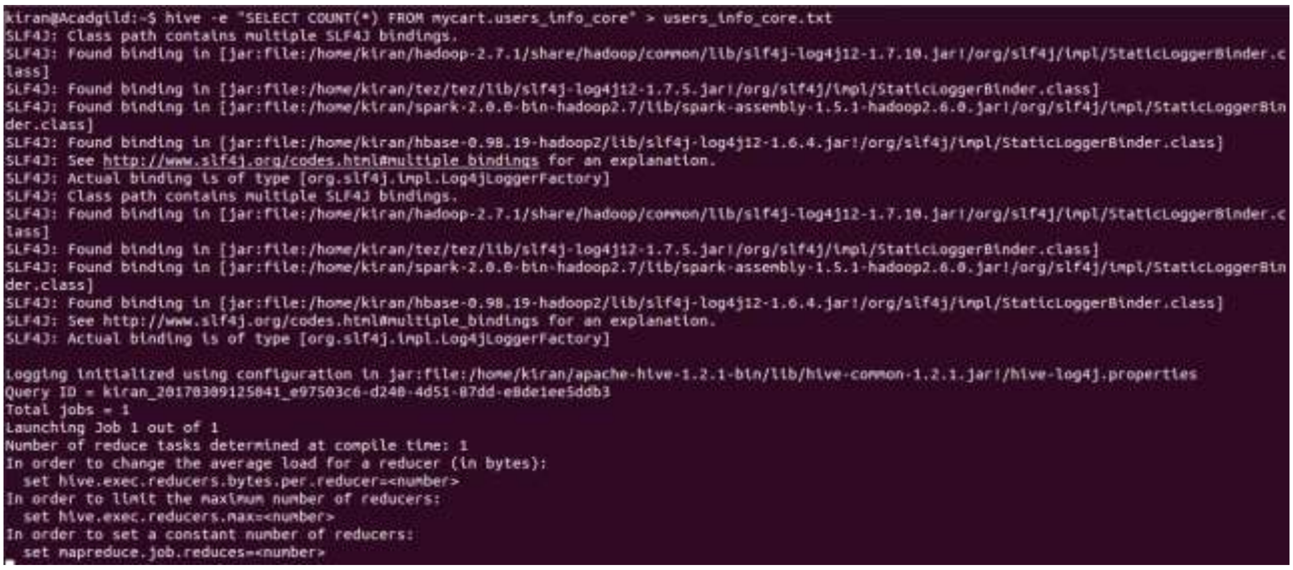
Checking the data in users_info_excp_r3.txt



```
kiran@Acadgild:~$ cat users_info_excp_r3.txt
0
kiran@Acadgild:~$ 
```

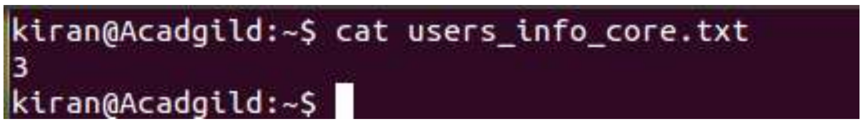
Checking the data in users_info_core table

hive -e "SELECT COUNT(*) FROM ecom.users_info_core" > users_info_core.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.users_info_core" > users_info_core.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309125841_e97503c6-d240-4d51-87dd-e8de1ee5ddb3
Total jobs = 1
Launching Job 1 out of 1
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>
```

Checking the data in users_info_core.txt

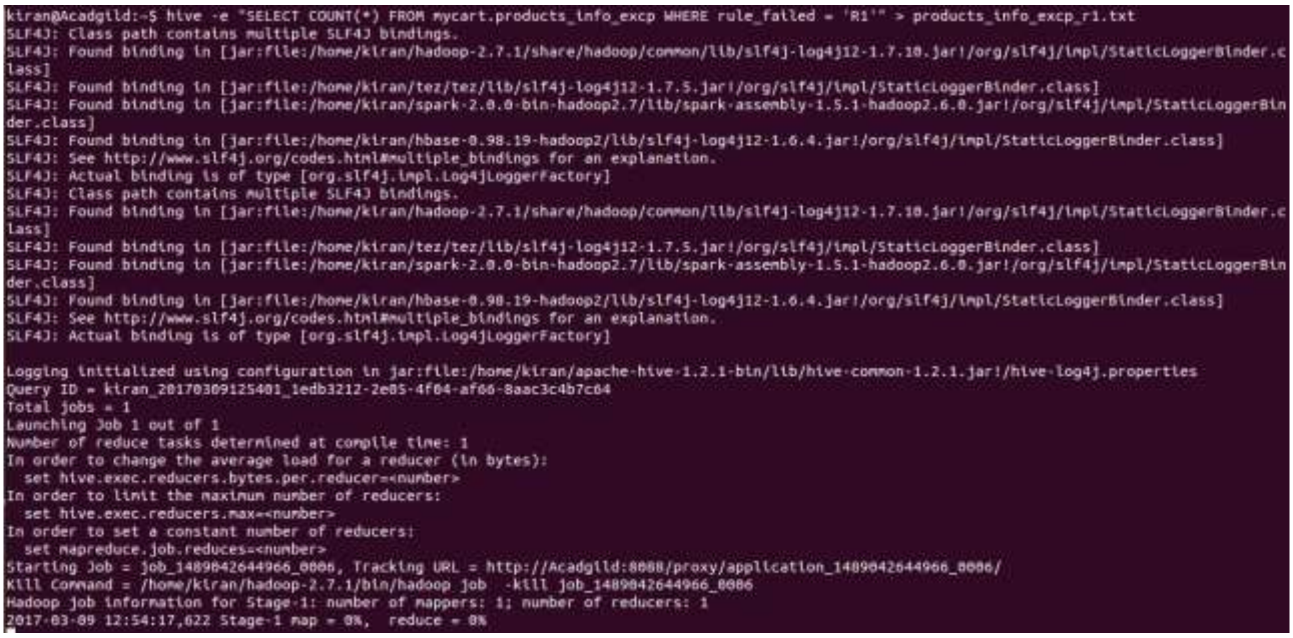


```
kiran@Acadgild:~$ cat users_info_core.txt
3
kiran@Acadgild:~$
```

Rules checking on products_info table

1.

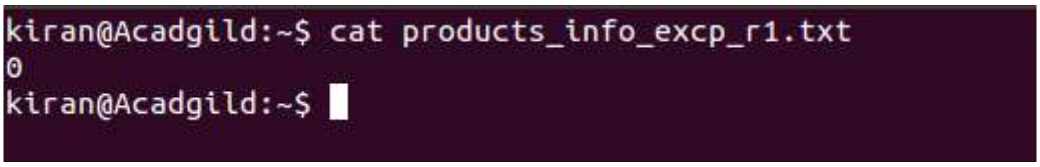
hive -e "SELECT COUNT(*) FROM ecom.products_info_excp WHERE rule_failed = 'R1'" > products_info_excp_r1.txt



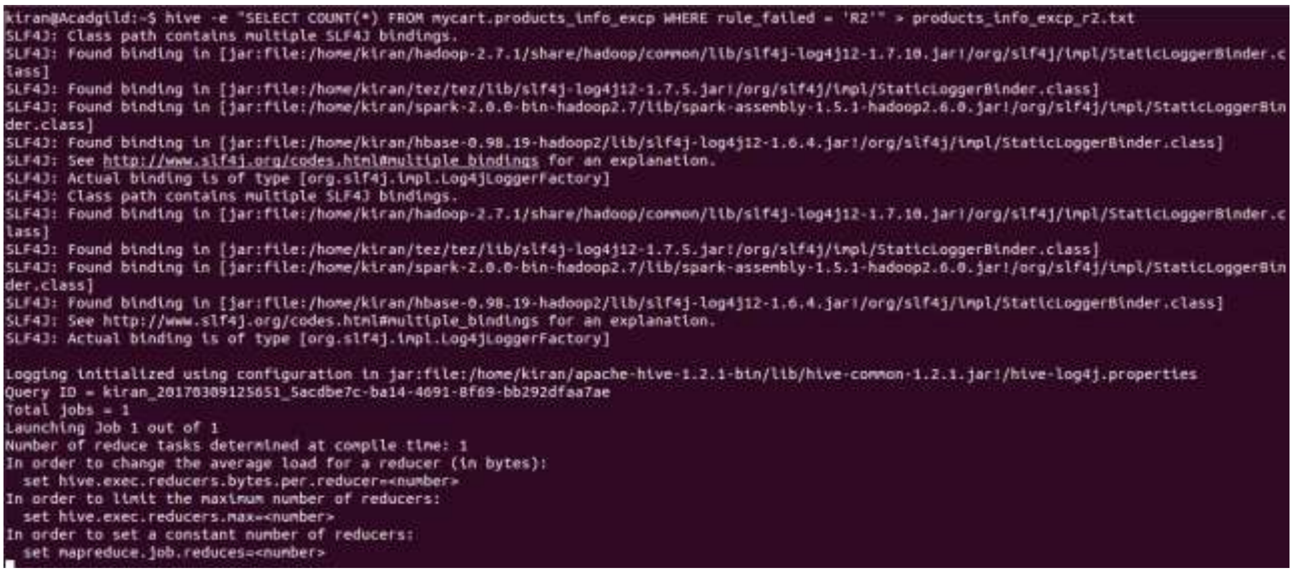
```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.products_info_excp WHERE rule_failed = 'R1'" > products_info_excp_r1.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309125401_1edb3212-2e05-4f04-af66-8aac3c4b7c64
Total jobs = 1
Launching Job 1 out of 1
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_1489042644966_0006, Tracking URL = http://Acadgild:8088/proxy/application_1489042644966_0006/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489042644966_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 12:54:17,622 Stage-1 map = 0%,  reduce = 0%
```

Checking the data in products_info_excp_r1

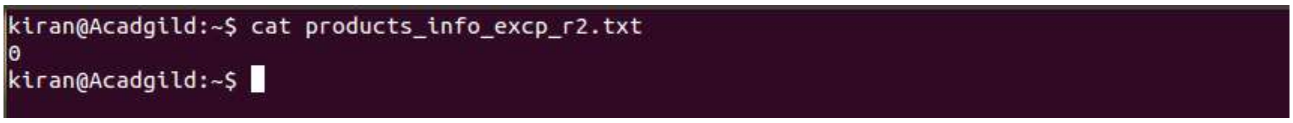
2.


```
kiran@Acadgild:~$ cat products_info_excp_r1.txt
0
kiran@Acadgild:~$ 
```

hive -e "SELECT COUNT(*) FROM ecom.products_info_excp WHERE rule_failed = 'R2'" > products_info_excp_r2.txt

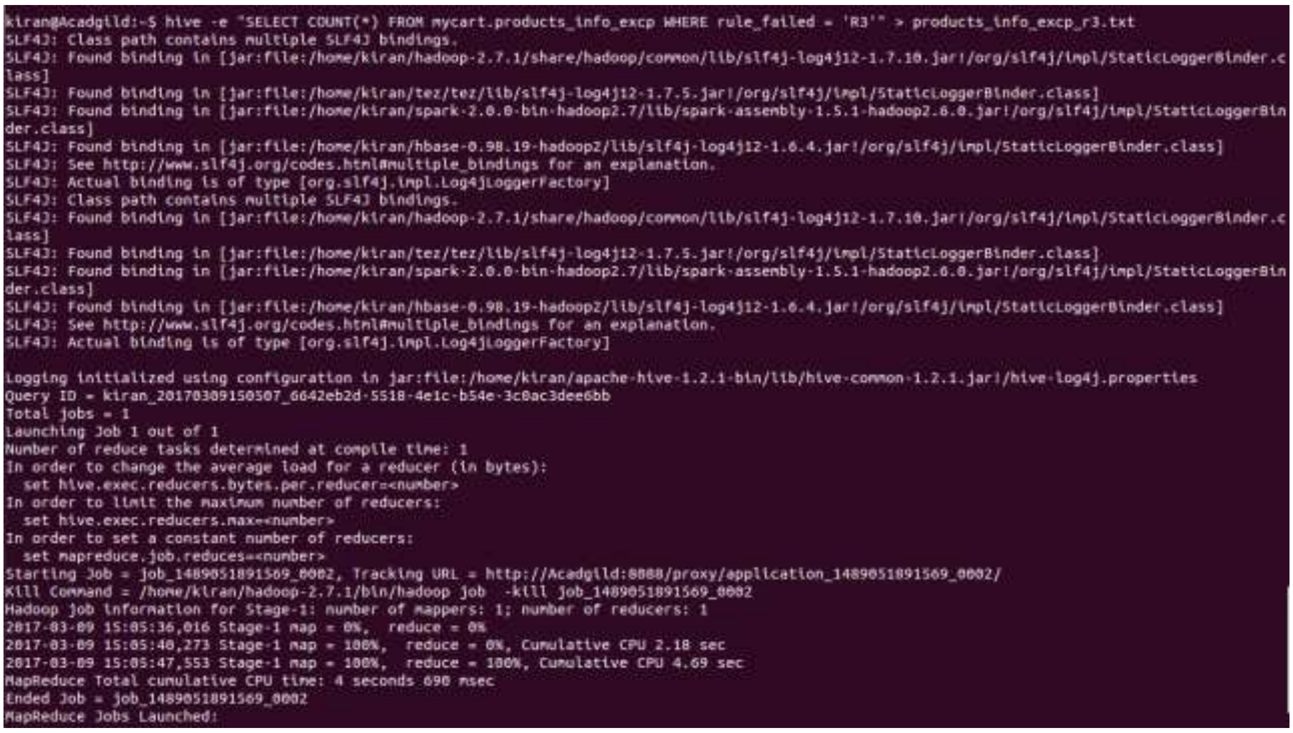

```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.products_info_excp WHERE rule_failed = 'R2'" > products_info_excp_r2.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309125651_5acdbe7c-ba14-4091-8f69-bb292dfaa7ae
Total jobs = 1
Launching Job 1 out of 1
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>
```

Checking the data in products_info_excp_r2.txt


```
kiran@Acadgild:~$ cat products_info_excp_r2.txt
0
kiran@Acadgild:~$ 
```

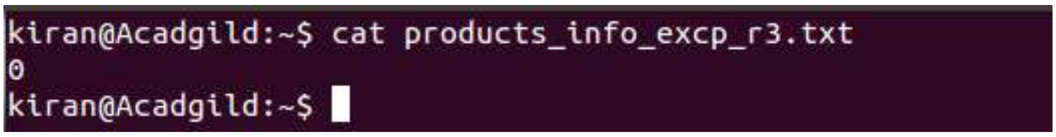
3.

hive -e "SELECT COUNT(*) FROM ecom.products_info_excp WHERE rule_failed = 'R3'" > products_info_excp_r3.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.products_info_excp WHERE rule_failed = 'R3'" > products_info_excp_r3.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309150507_6642eb2d-5518-4e1c-b54e-3c0ac3dee6bb
Total jobs = 1
Launching Job 1 out of 1
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_1489051891569_0002, Tracking URL = http://Acadgild:8088/proxy/application_1489051891569_0002/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489051891569_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 15:05:36,016 Stage-1 map = 0%,  reduce = 0%
2017-03-09 15:05:40,273 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.18 sec
2017-03-09 15:05:47,553 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.69 sec
MapReduce Total cumulative CPU time: 4 seconds 690 msec
Ended Job = job_1489051891569_0002
MapReduce Jobs Launched:
```

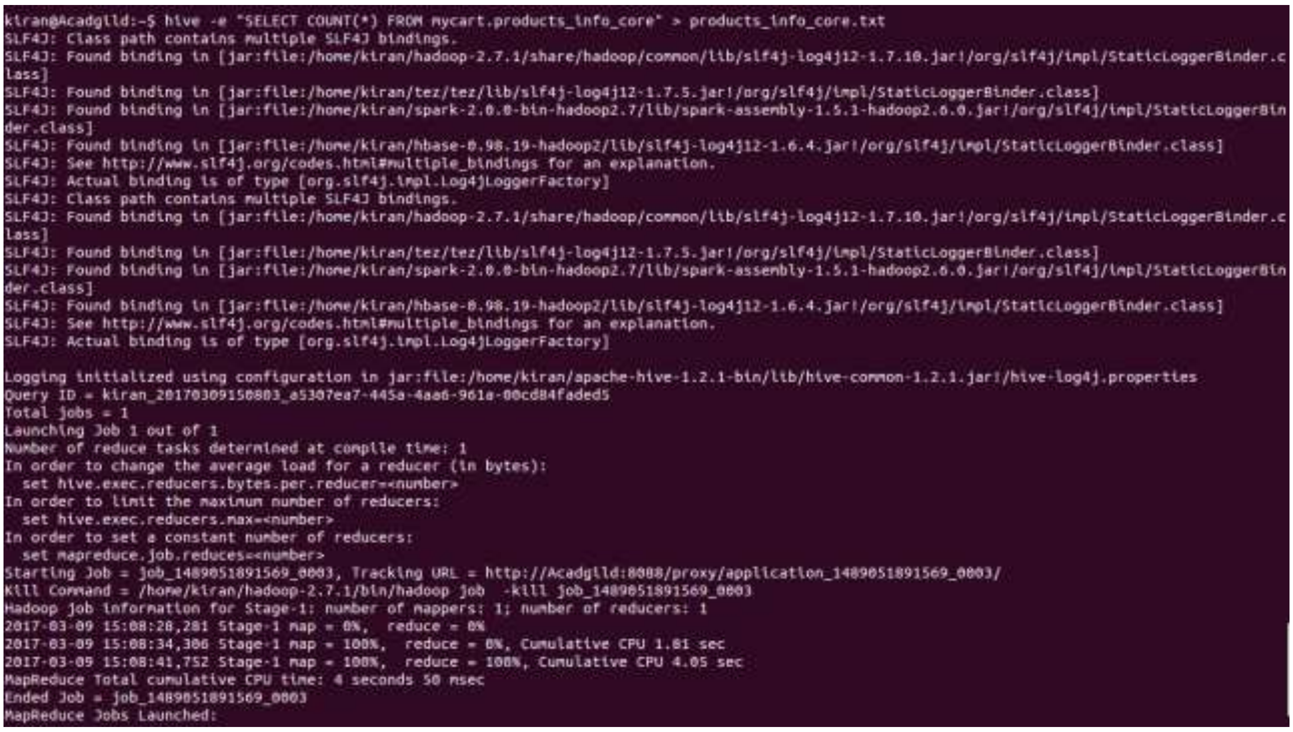
Checking the data in products_info_excp_r3.txt



```
kiran@Acadgild:~$ cat products_info_excp_r3.txt
0
kiran@Acadgild:~$ 
```

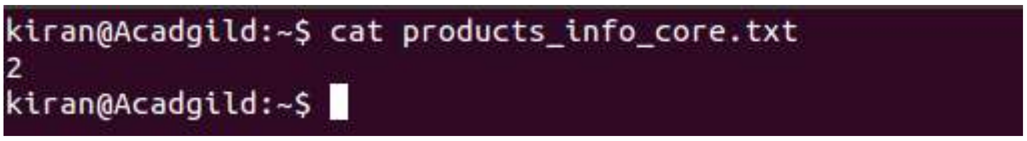
Checking the contents of products_info_core table

hive -e "SELECT COUNT(*) FROM ecom.products_info_core" > products_info_core.txt



```
kiran@Acadgild:~$ hive -e "SELECT COUNT(*) FROM mycart.products_info_core" > products_info_core.txt
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/kiran/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
Query ID = kiran_20170309150803_a5307ea7-445a-4aa6-961a-00cd84faded5
Total jobs = 1
Launching Job 1 out of 1
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_1489051891569_0003, Tracking URL = http://Acadgild:8088/proxy/application_1489051891569_0003/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489051891569_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-03-09 15:08:28,281 Stage-1 map = 0%,  reduce = 0%
2017-03-09 15:08:34,306 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.81 sec
2017-03-09 15:08:41,752 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.05 sec
MapReduce Total cumulative CPU time: 4 seconds 50 msec
Ended Job = job_1489051891569_0003
MapReduce Jobs Launched:
```

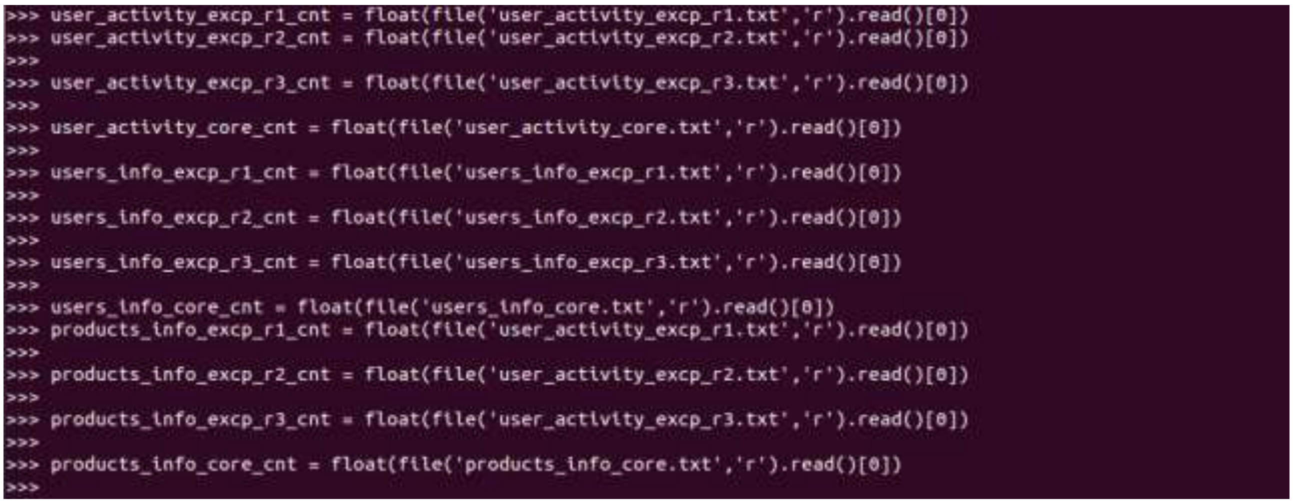
Checking the data in products_info_core.txt file



```
kiran@Acadgild:~$ cat products_info_core.txt
2
kiran@Acadgild:~$ 
```

Python code

```
user_activity_excp_r1_cnt = float(file('user_activity_excp_r1.txt','r').read()[0])
user_activity_excp_r2_cnt = float(file('user_activity_excp_r2.txt','r').read()[0])

user_activity_excp_r3_cnt = float(file('user_activity_excp_r3.txt','r').read()[0])
user_activity_core_cnt = float(file('user_activity_core.txt','r').read()[0])

users_info_excp_r1_cnt = float(file('users_info_excp_r1.txt','r').read()[0])
users_info_excp_r2_cnt = float(file('users_info_excp_r2.txt','r').read()[0])
users_info_excp_r3_cnt = float(file('users_info_excp_r3.txt','r').read()[0])
users_info_core_cnt = float(file('users_info_core.txt','r').read()[0])

products_info_excp_r1_cnt = float(file('user_activity_excp_r1.txt','r').read()[0])
products_info_excp_r2_cnt = float(file('user_activity_excp_r2.txt','r').read()[0])
products_info_excp_r3_cnt = float(file('user_activity_excp_r3.txt','r').read()[0])
products_info_core_cnt = float(file('products_info_core.txt','r').read()[0])
```



```
>>> user_activity_excp_r1_cnt = float(file('user_activity_excp_r1.txt','r').read()[0])
>>> user_activity_excp_r2_cnt = float(file('user_activity_excp_r2.txt','r').read()[0])
>>>
>>> user_activity_excp_r3_cnt = float(file('user_activity_excp_r3.txt','r').read()[0])
>>>
>>> user_activity_core_cnt = float(file('user_activity_core.txt','r').read()[0])
>>>
>>> users_info_excp_r1_cnt = float(file('users_info_excp_r1.txt','r').read()[0])
>>>
>>> users_info_excp_r2_cnt = float(file('users_info_excp_r2.txt','r').read()[0])
>>>
>>> users_info_excp_r3_cnt = float(file('users_info_excp_r3.txt','r').read()[0])
>>>
>>> users_info_core_cnt = float(file('users_info_core.txt','r').read()[0])
>>> products_info_excp_r1_cnt = float(file('user_activity_excp_r1.txt','r').read()[0])
>>>
>>> products_info_excp_r2_cnt = float(file('user_activity_excp_r2.txt','r').read()[0])
>>>
>>> products_info_excp_r3_cnt = float(file('user_activity_excp_r3.txt','r').read()[0])
>>>
>>> products_info_core_cnt = float(file('products_info_core.txt','r').read()[0])
>>>
```

```
threshold = file('rules_threshold.txt','r').read().strip().split(',')
r1_threshold, r2_threshold, r3_threshold = float(threshold[0])/100, float(threshold[1])/100,
float(threshold[2])/100

usr_activity_cnt = user_activity_excp_r1_cnt + user_activity_excp_r2_cnt +
user_activity_excp_r3_cnt + user_activity_core_cnt
users_info_cnt = users_info_excp_r1_cnt + users_info_excp_r2_cnt + users_info_excp_r3_cnt +
users_info_core_cnt
products_info_cnt = products_info_excp_r1_cnt + products_info_excp_r2_cnt +
products_info_excp_r3_cnt + products_info_core_cnt

if (user_activity_excp_r1_cnt/usr_activity_cnt > r1_threshold or
user_activity_excp_r2_cnt/usr_activity_cnt > r2_threshold or
user_activity_excp_r3_cnt/usr_activity_cnt > r3_threshold):
        print("User activity records are invalid")
elif (users_info_excp_r1_cnt/users_info_cnt > r1_threshold or
users_info_excp_r2_cnt/users_info_cnt > r2_threshold or users_info_excp_r3_cnt/users_info_cnt >
r3_threshold):
        print("User info records are invalid")
elif (products_info_excp_r1_cnt/products_info_core_cnt > r1_threshold or
products_info_excp_r2_cnt/products_info_core_cnt > r2_threshold or
products_info_excp_r3_cnt/products_info_core_cnt > r3_threshold):
        print("Products info records are invalid")
```

If the number of invalid records are more than the threshold the project should be stopped ideally.

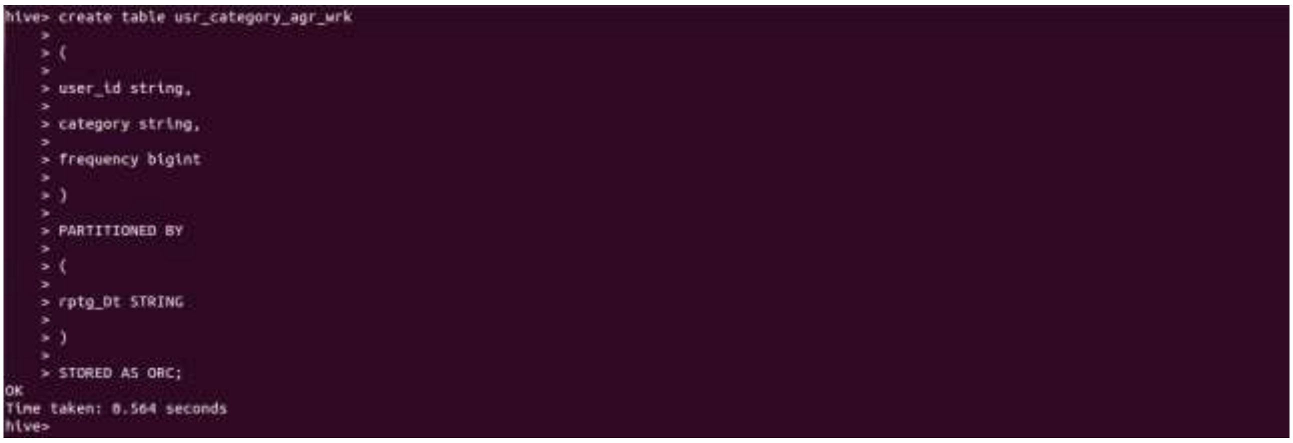
Data Analysis

Purchase Pattern Detection

1.What is the most purchased category for every user? Identify the users with maximum amount of valid purchase.

Creating table usr_category_agr_wrk

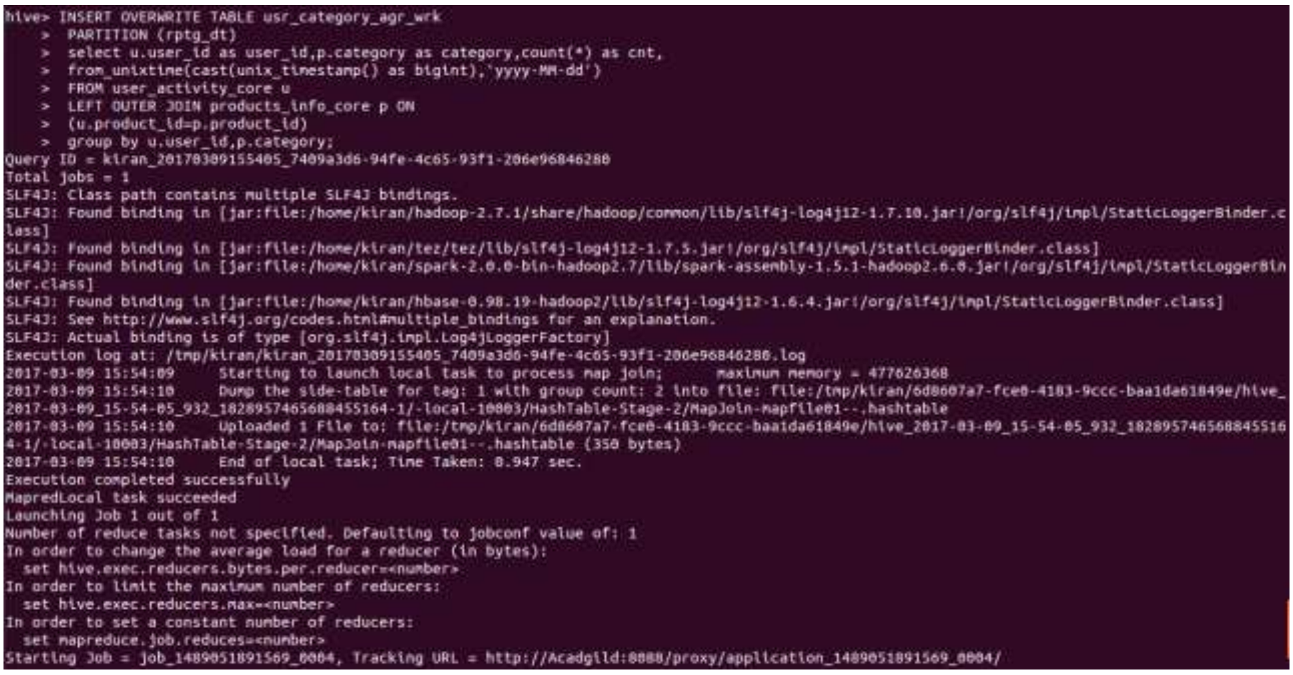
```
create table usr_category_agr_wrk
(
user_id string,
category string,
frequency bigint
)
PARTITIONED BY
(
rptg_Dt STRING
)
STORED AS ORC;
```



```
hive> create table usr_category_agr_wrk
    >
    > (
    >
    > user_id string,
    >
    > category string,
    >
    > frequency bigint
    >
    > )
    >
    > PARTITIONED BY
    >
    > (
    >
    > rptg_Dt STRING
    >
    > )
    >
    > STORED AS ORC;
OK
Time taken: 0.564 seconds
hive>
```

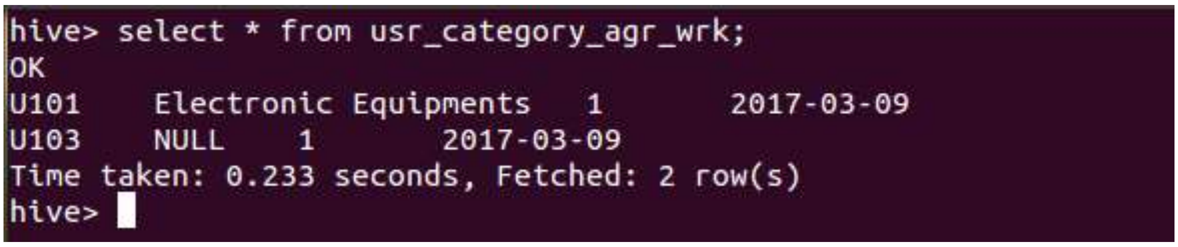
Query to find the frequency of most purchased category and inserting into the created table

INSERT OVERWRITE TABLE usr_category_agr_wrk PARTITION (rptg_dt) select u.user_id as user_id,p.category as category,count(*) as cnt, from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd') FROM user_activity_core u LEFT OUTER JOIN products_info_core p ON (u.product_id=p.product_id) group by u.user_id,p.category;



```
hive> INSERT OVERWRITE TABLE usr_category_agr_wrk
    >  PARTITION (rptg_dt)
    >  select u.user_id as user_id,p.category as category,count(*) as cnt,
    >  from_unixtime(cast(unix_timestamp() as bigint),'yyyy-MM-dd')
    >  FROM user_activity_core u
    >  LEFT OUTER JOIN products_info_core p ON
    >  (u.product_id=p.product_id)
    >  group by u.user_id,p.category;
Query ID = kiran_20170309155405_7409a3d6-94fe-4c65-93f1-206e96846280
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Execution log at: /tmp/kiran/kiran_20170309155405_7409a3d6-94fe-4c65-93f1-206e96846280.log
2017-03-09 15:54:09     Starting to launch local task to process map join;      maximum memory = 477626368
2017-03-09 15:54:10     Dump the side-table for tag: 1 with group count: 2 into file: file:/tmp/kiran/6d8607a7-fce0-4183-9ccc-baa1da61849e/hive_
2017-03-09_15-54-05_932_1828957465688455164-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2017-03-09 15:54:10     Uploaded 1 File to: file:/tmp/kiran/6d8607a7-fce0-4183-9ccc-baa1da61849e/hive_2017-03-09_15-54-05_932_182895746568845516
4-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (350 bytes)
2017-03-09 15:54:10     End of local task; Time Taken: 0.947 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Defaulting to jobconf value of: 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_1489051891569_0004, Tracking URL = http://Acadgild:8088/proxy/application_1489051891569_0004/
```

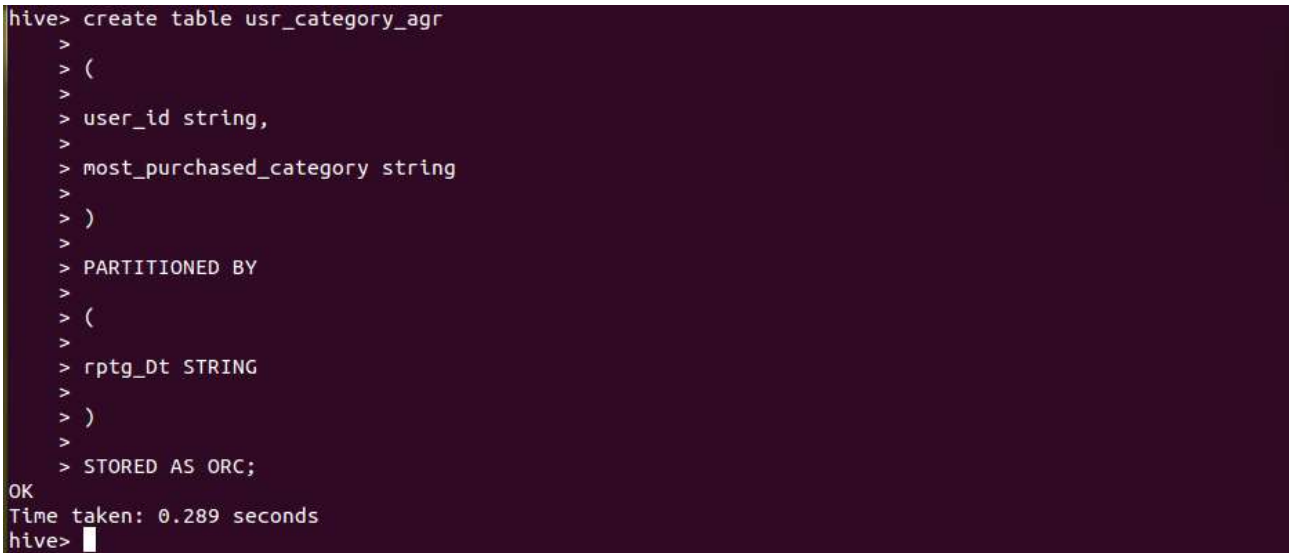
Checking the most purchased category frequency



```
hive> select * from usr_category_agr_wrk;
OK
U101    Electronic Equipments   1       2017-03-09
U103    NULL    1       2017-03-09
Time taken: 0.233 seconds, Fetched: 2 row(s)
hive>
```

Creating table usr_category_agr

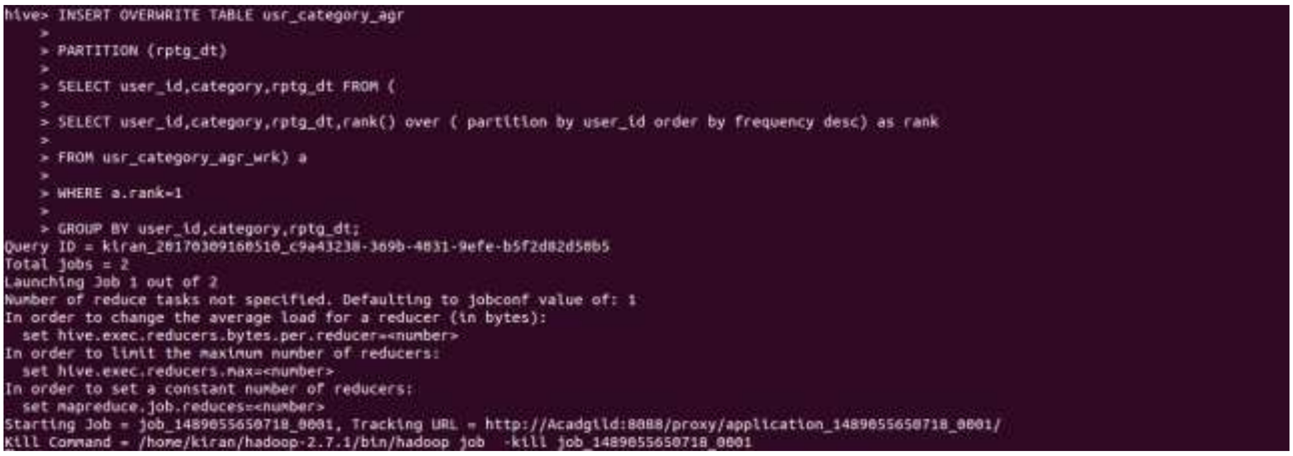
create table usr_category_agr
(
user_id string,
most_purchased_category string
)
PARTITIONED BY
(
rptg_Dt STRING
)
STORED AS ORC;



```
hive> create table usr_category_agr
    >
    > (
    >
    > user_id string,
    >
    > most_purchased_category string
    >
    > )
    >
    > PARTITIONED BY
    >
    > (
    >
    > rptg_Dt STRING
    >
    > )
    >
    > STORED AS ORC;
OK
Time taken: 0.289 seconds
hive>
```

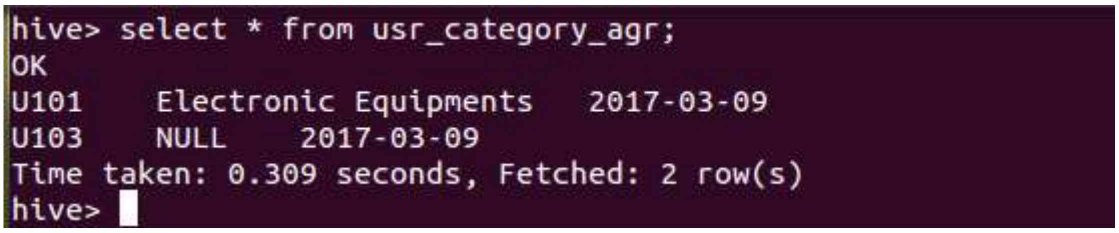
Query to find the most purchased category and inserting into the created table

INSERT OVERWRITE TABLE usr_category_agr
PARTITION (rptg_dt)
SELECT user_id,category,rptg_dt FROM (
SELECT user_id,category,rptg_dt,rank() over (partition by user_id order by frequency desc) as rank
FROM usr_category_agr_wrk) a
WHERE a.rank=1
GROUP BY user_id,category,rptg_dt;



```
hive> INSERT OVERWRITE TABLE usr_category_agr
    >
    > PARTITION (rptg_dt)
    >
    > SELECT user_id,category,rptg_dt FROM (
    >
    > SELECT user_id,category,rptg_dt,rank() over ( partition by user_id order by frequency desc) as rank
    >
    > FROM usr_category_agr_wrk) a
    >
    > WHERE a.rank=1
    >
    > GROUP BY user_id,category,rptg_dt;
Query ID = kiran_20170309160510_c9a43238-369b-4031-9efe-b5f2d82d58b5
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Defaulting to jobconf value of: 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_1489055650718_0001, Tracking URL = http://Acadgild:8088/proxy/application_1489055650718_0001/
Kill Command = /home/kiran/hadoop-2.7.1/bin/hadoop job  -kill job_1489055650718_0001
```

Checking the data

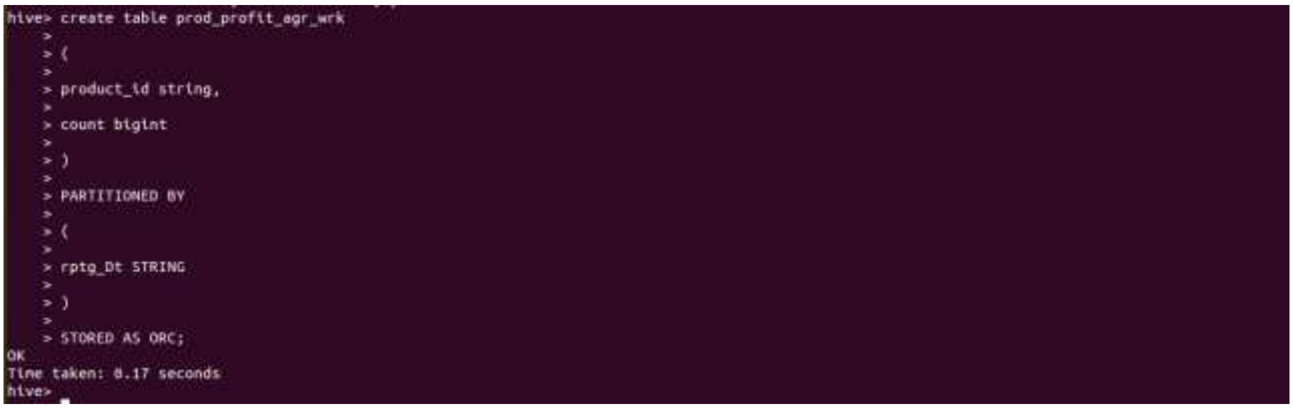


```
hive> select * from usr_category_agr;
OK
U101    Electronic Equipments   2017-03-09
U103    NULL    2017-03-09
Time taken: 0.309 seconds, Fetched: 2 row(s)
hive>
```

2.Which products are generating the maximum profit? (Profit = (price - discount) * profit_precentage)

Creating table prod_profit_agr_wrk

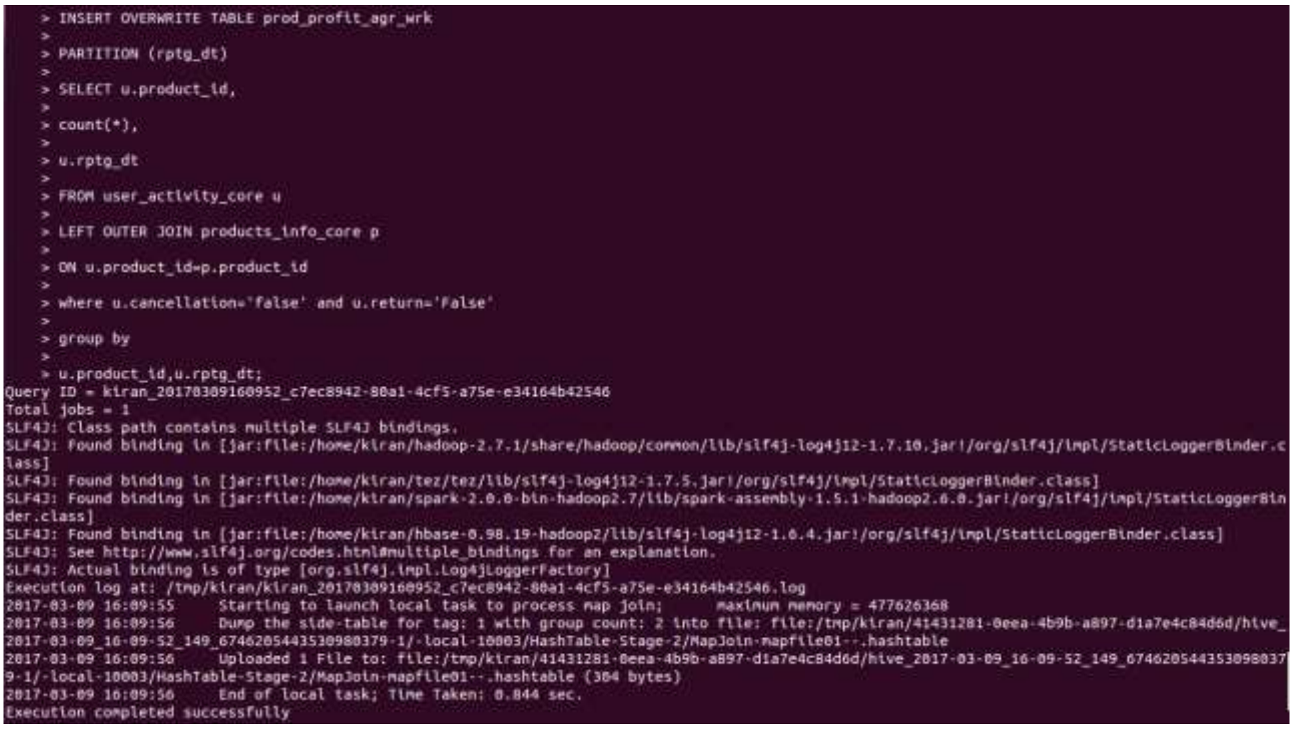
```
create table prod_profit_agr_wrk
(
product_id string,
count bigint
)
PARTITIONED BY
(
rptg_Dt STRING
)
STORED AS ORC;
```



```
hive> create table prod_profit_agr_wrk
    >
    > (
    >
    > product_id string,
    >
    > count bigint
    >
    > )
    >
    > PARTITIONED BY
    >
    > (
    >
    > rptg_Dt STRING
    >
    > )
    >
    > STORED AS ORC;
OK
Time taken: 0.17 seconds
hive>
```

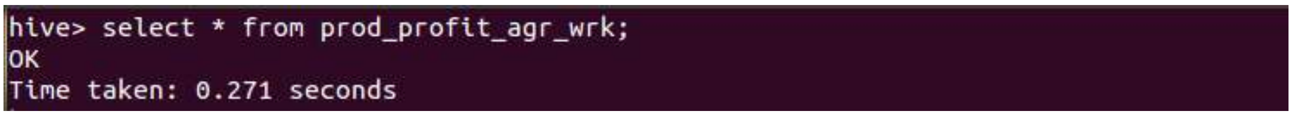
Inserting data into the table

```
INSERT OVERWRITE TABLE prod_profit_agr_wrk
PARTITION (rptg_dt)
SELECT u.product_id,
count(*),
u.rptg_dt
FROM user_activity_core u
LEFT OUTER JOIN products_info_core p
ON u.product_id=p.product_id
where u.cancellation='false' and u.return='False'
group by
u.product_id,u.rptg_dt;
```



```
    > INSERT OVERWRITE TABLE prod_profit_agr_wrk
    >
    > PARTITION (rptg_dt)
    >
    > SELECT u.product_id,
    >
    > count(*),
    >
    > u.rptg_dt
    >
    > FROM user_activity_core u
    >
    > LEFT OUTER JOIN products_info_core p
    >
    > ON u.product_id=p.product_id
    >
    > where u.cancellation='false' and u.return='False'
    >
    > group by
    >
    > u.product_id,u.rptg_dt;
Query ID = kiran_20170309160952_c7ec8942-80a1-4cf5-a75e-e34164b42546
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/kiran/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.c
lass]
SLF4J: Found binding in [jar:file:/home/kiran/tez/tez/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/kiran/spark-2.0.0-bin-hadoop2.7/lib/spark-assembly-1.5.1-hadoop2.6.0.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/home/kiran/hbase-0.98.19-hadoop2/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Execution log at: /tmp/kiran/kiran_20170309160952_c7ec8942-80a1-4cf5-a75e-e34164b42546.log
2017-03-09 16:09:55     Starting to launch local task to process map join;      maximum memory = 477626368
2017-03-09 16:09:56     Dump the side-table for tag: 1 with group count: 2 into file: file:/tmp/kiran/41431281-0eea-4b9b-a897-d1a7e4c84d6d/hive_
2017-03-09_16-09-52_149_6746205443530980379-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable
2017-03-09 16:09:56     Uploaded 1 File to: file:/tmp/kiran/41431281-0eea-4b9b-a897-d1a7e4c84d6d/hive_2017-03-09_16-09-52_149_674620544353098037
9-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile01--.hashtable (304 bytes)
2017-03-09 16:09:56     End of local task; Time Taken: 0.844 sec.
Execution completed successfully
```

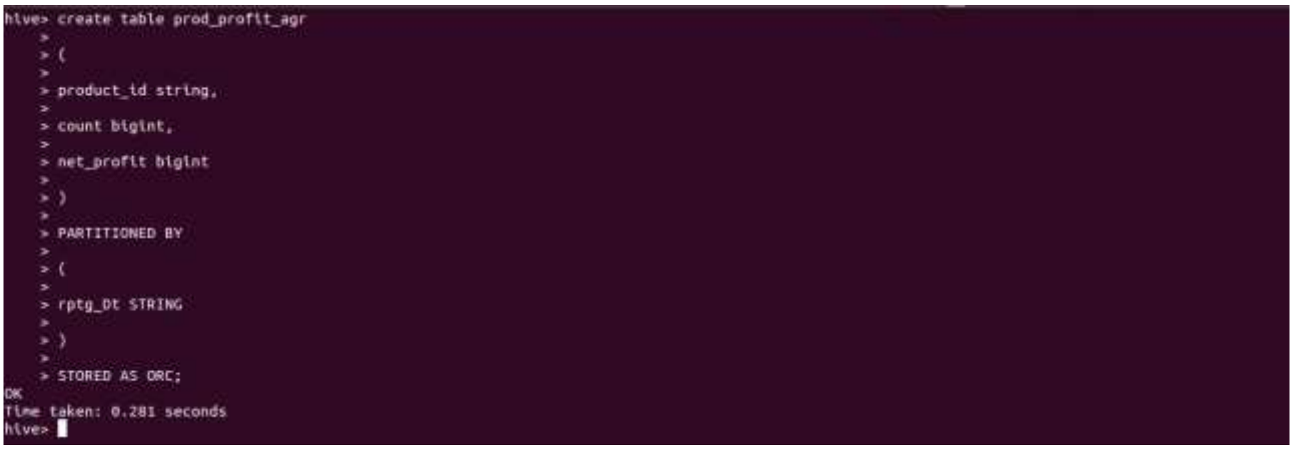
Checking the data



```
hive> select * from prod_profit_agr_wrk;
OK
Time taken: 0.271 seconds
```

Creating table prod_proft_agr

create table prod_profit_agr
(
product_id string,
count bigint,
net_profit bigint
)
PARTITIONED BY
(
rptg_Dt STRING
)
STORED AS ORC;



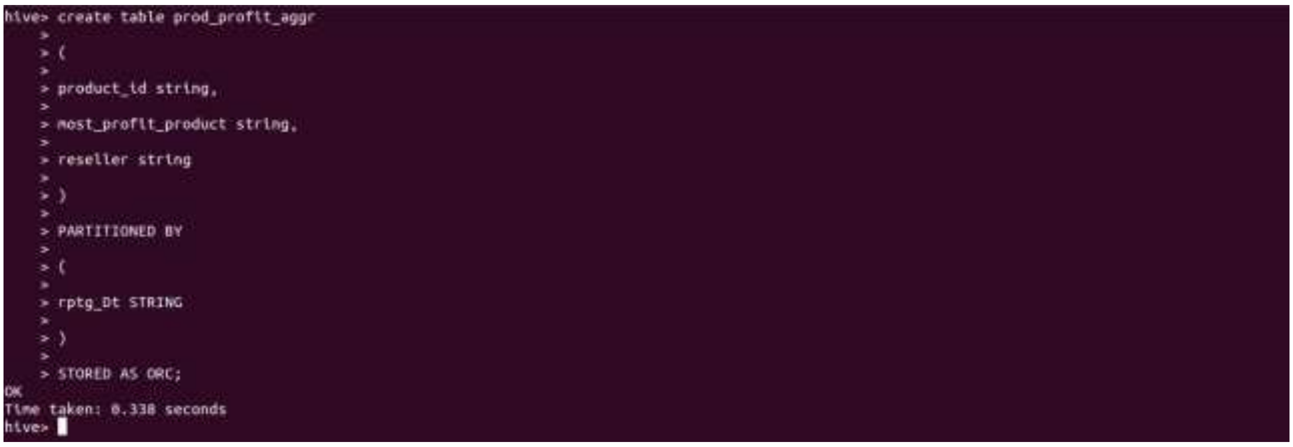
```
hive> create table prod_profit_agr
    >
    > (
    >
    > product_id string,
    >
    > count bigint,
    >
    > net_profit bigint
    >
    > )
    >
    > PARTITIONED BY
    >
    > (
    >
    > rptg_Dt STRING
    >
    > )
    >
    > STORED AS ORC;
OK
Time taken: 0.281 seconds
hive>
```

Inserting data into the table

INSERT OVERWRITE TABLE prod_profit_agr
PARTITION (rptg_dt)
SELECT u.product_id,
count,
count * (cast((price-cast(discount as bigint)) as bigint)* cast(profit_percent as bigint)/100) as net_profit,
u.rptg_dt
FROM prod_profit_agr_wrk u
LEFT OUTER JOIN products_info_core p
ON u.product_id=p.product_id
group by u.product_id,count,
count * (cast((price-cast(discount as bigint)) as bigint)* cast(profit_percent as bigint)/100),
u.rptg_dt;

3.Which resellers are generating the maximum profit?

create table prod_profit_aggr (product_id string, most_profit_product string, reseller string)
PARTITIONED BY (rptg_Dt STRING) STORED AS ORC;



```
hive> create table prod_profit_aggr
    >
    > (
    >
    > product_id string,
    >
    > most_profit_product string,
    >
    > reseller string
    >
    > )
    >
    > PARTITIONED BY
    >
    > (
    >
    > rptg_Dt STRING
    >
    > )
    >
    > STORED AS ORC;
OK
Time taken: 0.338 seconds
hive>
```

INSERT OVERWRITE TABLE prod_profit_aggr
PARTITION (rptg_dt)
SELECT product_id,most_profit_product,rptg_dt FROM (

```
SELECT p.product_id,p.net_profit as most_profit_product,pi.reseller,p.rptg_dt,rank() over (order
by net_profit desc) as rank
FROM prod_profit_agr p
LEFT OUTER JOIN products_info_core pi ON p.product_id=pi.product_id) a
WHERE a.rank=1
GROUP BY product_id,most_profit_product,rptg_dt;
```

4.Which is most sought after category corresponding to very occupation?

```
create table ocupation_category_aggr_wrk
(
user_id string,
occupation string,
category string,
count bigint
)
partitioned by
(
rptg_dt string)
stored as ORC;
```

Query

```
INSERT OVERWRITE TABLE ocupation_category_aggr_wrk
partition (rptg_dt)
select ua.user_id,u.occupation,p.category, count(*),ua.rptg_dt from
user_activity_core ua
LEFT OUTER JOIN users_info_core u
ON u.id=ua.user_id
LEFT OUTER JOIN products_info_core p
ON ua.product_id=p.product_id
group by u.occupation,p.category,ua.rptg_dt;
```

Table creating

```
create table ocupation_category_aggr
(
user_id string,
occupation string,
category string
)
partitioned by
(
rptg_dt string)
stored as ORC;
```

Query

```
INSERT OVERWRITE TABLE ocupation_category_aggr
partition (rptg_dt)
select user_id,occupation,category,rptg_dt from
```

```
(select occupation,category,rptg_dt, rank() over (partition by occupation order by count desc) as
rank
from ocupation_category_aggr_wrk )a
where a.rank=1;
```

Fraud detection:

1.Which user has performed most returns? What is the valid purchase made by those users?

```
create table fraud_detection_work1
(
user_id string,
return bigint
--valid_purchase bigint
)
partitioned by
(
rptg_dt string)
stored as ORC;
```

Query

```
INSERT OVERWRITE TABLE fraud_detection_work1
PARTITION (rptg_dt)

select user_id,count(*),rptg_dt from
user_activity_core u
where return='True'
group by user_id,rptg_dt;
```

Table

```
create table fraud_detection_work2
(
user_id string,
--return bigint
valid_purchase bigint
)
partitioned by
(
rptg_dt string)
stored as ORC;
```

Query

```
INSERT OVERWRITE TABLE fraud_detection_work2
PARTITION(rptg_dt)

select user_id,count(*),rptg_dt from
user_activity_core u
where return='False'
group by user_id,rptg_dt;
```

```
create table fraud_detection
(
user_id string,
return bigint,
valid_purchase bigint
)
partitioned by
(
rptg_dt string)
stored as ORC;
```

Query

```
INSERT OVERWRITE TABLE fraud_detection
PARTITION (rptg_dt)
select user_id,return,valid_purchase,rptg_dt from (
select w1.user_id as user_id,w1.return as return,w2.valid_purchase as valid_purchase,w1.rptg_dt as
rptg_dt,rank () over( order by w1.return desc) as rank
from fraud_detection_work1 w1
left outer join fraud_detection_work2 w2
ON w1.user_id=w2.user_id
)a
where a.rank=1;
```

2.Which location is getting most cancellation?

```
create table return_cancel_work
(
location struct<city:string,state:string>,
count bigint
)
PARTITIONED BY
(
rptg_dt string
)
STORED AS ORC;
```

Query

```
INSERT OVERWRITE TABLE return_cancel_work
PARTITION (rptg_dt)
select
CASE WHEN u.location IS NULL THEN named_struct('city','NA','state','NA')
ELSE u.location END AS location,
count(*),
CASE WHEN u.rptg_dt IS NULL THEN 'NA'
ELSE u.rptg_dt END AS rptg_dt from
user_activity_core ua
LEFT OUTER JOIN users_info_core u ON
ua.user_id=u.user_id
WHERE
ua.return='True'
```

```
group by u.location,u.rptg_dt ;
```

3.Which location is getting most returns?

Creating table

```
create table return_aggr
(
location struct<city:string,state:string>,
count bigint
)
PARTITIONED BY
(
rptg_dt string
)
STORED AS ORC;
```

Query

```
INSERT OVERWRITE TABLE return_aggr
PARTITION (rptg_dt)
select
location,count,rptg_dt from
(
select location,count,rptg_dt,rank() over (order by count desc) rank
FROM return_cancel_work
)a
WHERE a.rank=1
group by location,count,rptg_dt;
```

QUESTION - Final Hive table to generate most purchased category which fraud detectation is return value is true

```
create table most_valid_purch_ctgr
(
user_id string,
category string,
purchase bigint
)
PARTITIONED BY
(
month string
)
STORED AS ORC;
```

Query

```
INSERT OVERWRITE TABLE most_valid_purch_ctgr
PARTITION (month)
```

```
select
u.user_id,
u.category,
fr.valid_purchase,
from_unixtime(unix_timestamp(fr.rptg_dt),'MM-YYYY') as month
from
ocupation_category_aggr u
left outer join
fraud_detection fr
ON (u.user_id = fr.user_id)
where
fr.return='True';
```

Sqoop final hive table data to MySQL using multiexport
===

one way to do it by hcatlog.

```
sqoop export --connect jdbc:mysql://localhost/test --driver com.mysql.jdbc.Driver --username hive --password hive --table  mysql_table_export --hcatalog-table table_text --input-fields-terminated-by '|' --input-lines-terminated-by '#'
```

else use the hadoop directory directly with normal export command as below - use /"*" for recursive export of partitions

```
sqoop export --connect jdbc:mysql://localhost/db --username root --table employee  --export-dir /emp/emp_data/*
```