CASE study 2

Finding the transactions done by the customer-

Creating the transaction table using the command and loading the data using the commands below-

```
CREATE TABLE TXNRECORDS (txnno INT, txndate STRING, custno INT, amount DOUBLE, category STRING,
product STRING, city STRING, state STRING, spendby STRING)
row format delimited fields terminated by ',';
```

```
LOAD DATA LOCAL INPATH '/home/acadgild/Desktop/Test hadoop/hive/txns.txt' into table TXNRECORDS;
```

Output

```
26      10-11-2011      4000009 31.58   Combat Sports   Wrestling       Orange  California      credit
27      09-29-2011      4000010 66.4    Games   Mahjong Fremont California      credit
28      05-12-2011      4000008 79.78   Team Sports     Cricket Lexington       Kentucky        credit
29      06-03-2011      4000001 126.9   Outdoor Recreation      Hunting Phoenix Arizona credit
30      03-14-2011      4000001 47.05   Water Sports    Swimming        Lincoln Nebraska        credit
31      11-28-2011      4000008 5.03    Games   Dice & Dice Sets        Los Angeles     California      credit
32      01-29-2011      4000008 20.13   Team Sports     Soccer  Springfield     Illinois        credit
33      06-15-2011      4000008 154.15  Outdoor Recreation      Lawn Games      Nashville       Tennessee       credit
34      05-06-2011      4000008 98.96   Team Sports     Indoor Volleyball       Atlanta Georgia credit
35      04-12-2011      4000008 185.26  Games   Board Games     Centennial      Colorado        credit
36      10-13-2011      4000007 35.66   Team Sports     Football        Saint Paul      Minnesota       credit
37      04-19-2011      4000007 20.2    Outdoor Recreation      Shooting Games  San Diego       California      credit
38      08-05-2011      4000007 150.6   Outdoor Recreation      Camping & Backpacking & Hiking  Hampton         Virgini
redit
39      03-12-2011      4000006 174.36  Outdoor Play Equipment  Swing Sets      Pittsburgh      Pennsylvania    credit
40      11-07-2011      4000005 165.1   Team Sports     Cheerleading    Reno    Nevada  credit
41      04-16-2011      4000004 28.11   Indoor Games    Bowling Westminster     Colorado        cash
42      09-10-2011      4000004 38.52   Outdoor Recreation      Tetherball      Denton  Texas   cash
43      04-22-2011      4000004 32.34   Water Sports    Water Polo      Las Vegas       Nevada  cash
44      09-11-2011      4000001 135.37  Water Sports    Surfing Seattle Washington      credit
45      11-27-2011      4000001 90.04   Exercise & Fitness      Abdominal Equipment     Honolulu        Hawaii  credit
46      05-27-2011      4000001 52.29   Gymnastics      Vaulting Horses Cleveland       Ohio    credit
47      10-23-2011      4000008 100.1   Outdoor Play Equipment  Swing Sets      Everett Washington      credit
48      09-27-2011      4000007 157.94  Exercise & Fitness      Exercise Bands  Philadelphia    Pennsylvania    credit
49      07-12-2011      4000010 144.59  Jumping Jumping Stilts  Cambridge       Massachusetts   credit
50      10-20-2011      4000010 55.93   Jumping Pogo Sticks     Everett Washington      credit
51      02-17-2011      4000002 32.65   Water Sports    Life Jackets    Columbus        Georgia cash
52      02-04-2011      4000005 44.82   Outdoor Play Equipment  Lawn Water Slides       Hampton         Virginia
53      06-12-2011      4000004 44.46   Water Sports    Scuba Diving & Snorkeling       Charleston      South Carolina
54      10-03-2011      4000007 154.87  Outdoor Recreation      Running Long Beach      California      credit
55      12-16-2011      4000006 106.11  Water Sports    Swimming        New York        New York        credit
56      06-21-2011      4000002 176.63  Outdoor Recreation      Geocaching      Boston  Massachusetts   credit
57      12-20-2011      4000003 178.2   Outdoor Recreation      Skating San Jose        California      credit
58      12-29-2011      4000002 194.86  Water Sports    Windsurfing     Oklahoma City   Oklahoma        credit
59      11-07-2011      4000001 21.43   Winter Sports   Snowboarding    Philadelphia    Pennsylvania    cash
```

Creating a new table called TRANSACTIONS_COUNT with 3 fields as show below-

```
hive> CREATE TABLE TRANSACTIONS_COUNT(
    > custid INT,
    > fname STRING,
    > count INT
    > )
    > row format delimited fields terminated by ',';
```

```
hive> CREATE TABLE TRANSACTIONS_COUNT(
    > custid INT,
    > fname STRING,
    > count INT
    > )
    > row format delimited fields terminated by ',';
OK
Time taken: 1.758 seconds
hive> show tables;
OK
buck_users
college
csv_table
mycustomer
mycustomer_ext
mycustomernew
olympics
olympics_swim
olympics_swimming
swim
transactions_count
txnrecords
txnrecords_seq
users
Time taken: 0.08 seconds, Fetched: 14 row(s)
hive> describe transactions_count;
OK
custid                  int
fname                   string
count                   int
Time taken: 0.193 seconds, Fetched: 3 row(s)
hive>
```

Writing a hive query to populate the table

Use the command below -

```
select a.fname, count(a.fname) from TRANSACTIONS_COUNT a join TXNRECORDS b on a.custid =b.custno
group by a.fname;
```

```
2018-09-19 07:43:40,881 Stage-2 map = 0%,  reduce = 0%
2018-09-19 07:43:52,426 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.33 sec
2018-09-19 07:44:03,797 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 5.89 sec
MapReduce Total cumulative CPU time: 5 seconds 890 msec
Ended Job = job_1537247019915_0027
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.89 sec   HDFS Read: 23244 HDFS Write: 309 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 890 msec
OK
Dolores 12
Elsie   12
Gretchen        10
Hazel   20
Karen   10
Kristina        16
Malcolm 12
Paige   12
Patrick 10
Sherri  6
```

INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.custid, a.fname, count(a.fname) from CUSTOMER a join TXNRECORDS b on a.custid =b.custno group by a.fname,a.custid;

```
4000001 Kristina        16
4000002 Paige   12
4000003 Sherri  6
4000004 Gretchen        10
4000005 Karen   10
4000006 Patrick 10
4000007 Elsie   12
4000008 Hazel   20
4000009 Malcolm 12
4000010 Dolores 12
```

Making TRANSACTION TABLE a Hbase compalint

use the command below-

**create table TRANSACTIONS_COUNT(custid int, fname String, count string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'with serdeproperties
("hbase.columns.mapping"=":key,customerdetails:fname, customerdetails:count")
tblproperties("hbase.table.name"="transactions_count");**

Loading the data in to the table using the command-

INSERT OVERWRITE TABLE TRANSACTIONS_COUNT select a.custid, a.fname, count(a.fname) from CUSTOMER a join TXNRECORDS b on a.custid =b.custno group by a.fname,a.custid;

Code -

```
        import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CaseStudyIUseCasesMoviesMapper extends
                      Mapper<LongWritable, Text, Text, Text> {


        public void map(LongWritable key, Text value, Context context)
                              throws IOException, InterruptedException {

                try {
            if (key.get() == 0 && value.toString().contains("movieId")){
                return;
            } else {
                String record = value.toString();
                        String[] parts = record.split(",");
                        context.write(new Text(parts[0]), new Text("movies\t" + parts[1]));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        }
}
```

```
import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CaseStudyIUseCasesRatingsMapper extends
                        Mapper<LongWritable, Text, Text, Text> {

         public void map(LongWritable key, Text value, Context context)
                                  throws IOException, InterruptedException {

                            try {
                    if (key.get() == 0 && value.toString().contains("userId")){
                        return;
                    } else {
                                String record = value.toString();
                                String[] parts = record.split(",");
                                context.write(new Text(parts[1]), new Text("ratings\t" + parts[2]));
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }

         }
}
```

```
import java.io.IOException;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;


public class CaseStudyIUseCasesReducer extends
                        Reducer<Text, Text, Text, Text> {

                public void reduce(Text key, Iterable<Text> values, Context context)
                                throws IOException, InterruptedException {
                        String titles = "";
                        double total = 0.0;
                        int count = 0;
                        System.out.println("Text Key    =>"+key.toString());
                        for (Text t : values) {
                                String parts[] = t.toString().split("\t");
                                System.out.println("Text values =>"+t.toString());
                                if (parts[0].equals("ratings")) {
                                        count++;
                                        String rating = parts[1].trim();
                                        System.out.println("Rating is =>"+rating);
                                        total += Double.parseDouble(rating);
                                } else if (parts[0].equals("movies")) {
                                        titles = parts[1];
                                }
                        }

                        double average = total / count;
                        String str = String.format("%d\t%f", count, average);
                    //String str = String.format("%d", count);

                        context.write(new Text(titles), new Text(str));
```

```
4000001,Kristina,Chung,55,Pilot
4000002,Paige,Chen,74,Teacher
4000003,Sherri,Melton,34,Firefighter
4000004,Gretchen,Hill,66,Computer hardware engineer
4000005,Karen,Puckett,74,Lawyer
4000006,Patrick,Song,42,Veterinarian
4000007,Elsie,Hamilton,43,Pilot
4000008,Hazel,Bender,63,Carpenter
4000009,Malcolm,Wagner,39,Artist
4000010,Dolores,McLaughlin,60,Writer
```