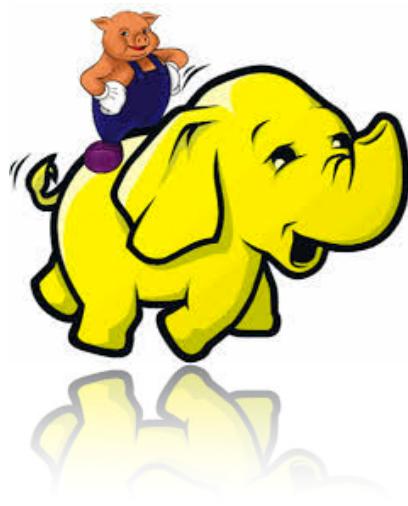


PIG SCRIPT ASSIGNMENT

BIG DATA ANALYTICS

Performed on Macintosh with Mojave OS 2019



Submitted To: Prof. Aditya Bharadwaj
Big Data Analytics

Submitted By: SID - 16103104
Name - Lovedeep Singh
BTech- CSE 4th Year

Ensure that all files are present in hdfs and run [pig](#) command to invoke grunt shell

```
sbm — java -Dproc_jar -Djava.net.preferIPv4Stack=true -Xmx1000m -Dpig.log.dir=/usr/local/Cellar/pig/0.17.0/libexec/logs -Dpig.log.fi...
2019-10-02 00:05:40,626 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2019-10-02 00:05:40,626 [main] INFO org.apache.pig.Main - Logging error messages to: /usr/local/Cellar/hadoop/3.1.2/libexec/sbin/pig_1569954940624.log
2019-10-02 00:05:40,657 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /Users/lovedeepsingh/.pigbootup not found
2019-10-02 00:05:40,938 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-10-02 00:05:40,969 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2019-10-02 00:05:40,969 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
2019-10-02 00:05:41,611 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2019-10-02 00:05:41,611 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:9010
2019-10-02 00:05:41,630 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-1e839120-2f9a-4609-93d8-4839c4048f6f
2019-10-02 00:05:41,630 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt> (//transaction_date:chararray,Product:chararray,Price:chararray,Payment_Type:chararray,Name:chararray,City:chararray,State:chararray,Country:chararray,Account_Created:chararray,Last_Login:chararray,Latitude:chararray,Longitude:chararray)
```

Problem 1: For the SalesJan2009.csv dataset answer the following queries:

Q1. How to load file in HDFS.

Ans : Use the following command to load the file in HDFS

```
salesTable = LOAD '/SalesJan2009.csv' USING PigStorage(',') AS  
(Transaction_date:chararray,Product:chararray,Price:chararray,Payment_Type:chararray,  
Name:chararray,City:chararray,State:chararray,Country:chararray,Account_Created:chararray,  
Last_Login:chararray,Latitude:chararray,Longitude:chararray);
```

```
grunt> salesTable = LOAD '/user/lovedeepsingh/SalesJan2009.csv' USING PigStorage  
(',') AS (Transaction_date:chararray,Product:chararray,Price:chararray,Payment_T  
ype:chararray,Name:chararray,City:chararray,State:chararray,Country:chararray,Ac  
count_Created:chararray,Last_Login:chararray,Latitude:chararray,Longitude:charar  
ray);  
grunt>
```

Q2. To group data by field Country

Ans: Use the following command

GroupByCountry = GROUP salesTable BY Country;

```
grunt> GroupByCountry = GROUP salesTable BY Country;  
grunt> █
```

Q3. To dump the data for city == New York

*data = FILTER salesTable by City == 'New York';
DUMP data;*

```
grunt> data = Filter salesTable BY City == 'New York';  
[  
grunt> DUMP data; █
```

```
TOTAL BYTES WRITTEN : 00010700  
Spillable Memory Manager spill count : 0  
Total bags proactively spilled: 0  
Total records proactively spilled: 0  
  
Job DAG:  
job_local162015339_0011
```

```
2019-10-02 00:41:59,938 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!  
2019-10-02 00:41:59,940 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!  
2019-10-02 00:41:59,942 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!  
2019-10-02 00:41:59,944 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!  
2019-10-02 00:41:59,945 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized  
2019-10-02 00:41:59,949 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1  
2019-10-02 00:41:59,949 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1  
grunt> █
```

Output is empty since there was no entry with City as New York.

Q4. To store the result in final output folder.

Ans Run the following command

STORE GroupByCountry INTO 'pig_output_sales' USING PigStorage('\t');

```
grunt> STORE GroupByCountry INTO 'pig_output_sales' USING PigStorage('\t');
```

```
Output(s):
Successfully stored 58 records (150387264 bytes) in: "hdfs://localhost:9000/user/lovedeepsingh/pig_output_sales"
```

Counters:

```
Total records written : 58
Total bytes written : 150387264
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

Job DAG:

```
job_local1844036798_0013
```

```
2019-10-02 00:55:49,340 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-10-02 00:55:49,343 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-10-02 00:55:49,344 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2019-10-02 00:55:49,347 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

```
grunt> ls
hdfs://localhost:9000/user/lovedeepsingh/CSE.csv<r 3> 3937
hdfs://localhost:9000/user/lovedeepsingh/SalesJan2009.csv<r 3> 123637
hdfs://localhost:9000/user/lovedeepsingh/abc.pages<r 3> 78317
hdfs://localhost:9000/user/lovedeepsingh/abc.rtf<r 3> 439
hdfs://localhost:9000/user/lovedeepsingh/abc.txt<r 3> 78317
hdfs://localhost:9000/user/lovedeepsingh/input.csv<r 3> 689413344
hdfs://localhost:9000/user/lovedeepsingh/output1 <dir>
hdfs://localhost:9000/user/lovedeepsingh/output2 <dir>
hdfs://localhost:9000/user/lovedeepsingh/output3 <dir>
hdfs://localhost:9000/user/lovedeepsingh/pig_output_sales <dir>
grunt>
```

```
hdfs dfs -cat pig_output_sales/part-r-00000
```

```
(base) Lovedeeps-MacBook-Pro:sbin lovedeepsingh$ hdfs dfs -cat pig_output_sales/part-r-00000
```

```
000/-2.828081,1/1//04 0:54,PrimusCafe,1200,Diners,Patricia and Martin,Buntingford,England,United Kingdom,1/12/08 7:00,2/19/09 23:30,52.4666667,-1.9166667),(1/26/09 8:58,Product1,1200,Diners,Miss,London,England,United Kingdom,1/26/09 8:46,1/26/09 11:14,51.52721,0.14559),(1/5/09 5:33,Product1,1200,Mastercard,Kim,London,England,United Kingdom,12/8/05 13:22,1/27/09 4:49,51.52721,0.14559),(1/14/09 1:28,Product1,1200,Mastercard,Sabine,Milton Keynes,England,United Kingdom,6/24/06 15:10,2/7/09 0:26,52.0333333,-0.7),(1/5/09 15:23,Product1,1200,Visa,maggie,Stockton,England,United Kingdom,6/12/08 10:59,2/7/09 11:08,52.25,-2.7),(1/25/09 8:48,Product1,1200,Mastercard,Malinda,Oxford,England,United Kingdom,4/5/08 13:20,2/21/09 1:36,51.75,-1.25)}  
Dominican Republic {(1/20/09 2:17,Product1,1200,Diners,Sarah,Santo Domingo City,Distrito Nacional,Dominican Republic,11/9/08 1:26,2/22/09 11:21,18.4666667,-69.9)}  
United Arab Emirates {(1/25/09 2:46,Product2,3600,Visa,Family,Dubai,Dubayy,United Arab Emirates,1/8/09 1:19,2/8/09 2:06,25.2522222,55.28),(1/14/09 11:19,Product1,1200,Visa,Jennifer,Jumeira,Dubayy,United Arab Emirates,1/14/09 10:44,1/14/09 21:26,25.2097222,55.2477778),(1/10/09 7:55,Product1,1200,Visa,Laura,Jumeira,Dubayy,United Arab Emirates,1/7/09 9:47,2/1/09 10:18,25.2097222,55.2477778),(1/17/09 20:46,Product2,3600,Visa,Michelle,Dubai,Dubayy,United Arab Emirates,4/13/08 2:36,2/8/09 2:12,25.2522222,55.28),(1/20/09 20:09,Product1,1200,Visa,Jeannette,Dubai,Dubayy,United Arab Emirates,1/18/09 9:58,2/23/09 7:01,25.2522222,55.28),(1/17/09 9:49,Product1,1200,Diners,Kathrin,Abu Dhabi,Abu Zaby,United Arab Emirates,2/27/08 20:02,1/30/09 8:45,24.4666667,54.3666667)}  
(base) Lovedeeps-MacBook-Pro:sbin lovedeepsingh$
```

Problem 2. For the movie dataset answer the following queries:

Before proceeding further, we need to load files appropriately, we use the following commands

```
hdfs dfs -copyFromLocal u.data /
```

```
(base) Lovedeeps-MacBook-Pro:fwdbigdataanalyticpigplatform lovedeepsingh$ hdfs dfs -copyFromLocal u.data/
```

```
hdfs dfs -copyFromLocal u.item /
```

```
(base) Lovedeeps-MacBook-Pro:fwdbigdataanalyticpigplatform lovedeepsingh$ hdfs dfs -copyFromLocal u.item/
```

```
hdfs dfs -ls
```

```
(base) Lovedeeps-MacBook-Pro:fwdbigdataanalyticpigplatform lovedeepsingh$ hdfs d  
fs -ls
```

```
(base) Lovedeeps-MacBook-Pro:fwdbigdataanalyticpigplatform lovedeepsingh$ hdfs d  
fs -ls  
2019-10-02 14:33:07,844 WARN util.NativeCodeLoader: Unable to load native-hadoop  
library for your platform... using builtin-java classes where applicable  
Found 12 items  
-rw-r--r-- 3 lovedeepsingh supergroup 3937 2019-09-27 19:49 CSE.csv  
-rw-r--r-- 3 lovedeepsingh supergroup 123637 2019-09-27 11:30 SalesJan2009  
.csv  
-rw-r--r-- 3 lovedeepsingh supergroup 78317 2019-09-27 20:39 abc.pages  
-rw-r--r-- 3 lovedeepsingh supergroup 439 2019-09-27 20:48 abc.rtf  
-rw-r--r-- 3 lovedeepsingh supergroup 78317 2019-09-27 20:41 abc.txt  
-rw-r--r-- 3 lovedeepsingh supergroup 689413344 2019-09-12 15:00 input.csv  
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:10 output1  
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:10 output2  
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:11 output3  
drwxr-xr-x - lovedeepsingh supergroup 0 2019-10-02 00:55 pig_output_s  
ales  
-rw-r--r-- 3 lovedeepsingh supergroup 2079229 2019-10-02 14:28 u.data  
-rw-r--r-- 3 lovedeepsingh supergroup 236344 2019-10-02 14:30 u.item
```

pig

```
ratings = LOAD '/user/lovedeepsingh/u.data' AS (userID:int, movieID:int, rating:int,  
ratingTime:int);
```

```
(base) Lovedeeps-MacBook-Pro:fwdbigdataanalyticpigplatform lovedeepsingh$ pig
```

```
grunt> ratings = LOAD '/user/lovedeepsingh/u.data' AS (userID:int, movieID:int,  
rating:int, ratingTime:int);
```

```
grunt> DUMP ratings;
```

```
(113,975,5,875936424)
(943,391,2,888640291)
(864,685,4,888891900)
(750,323,3,879445877)
(279,64,1,875308510)
(646,750,3,888528902)
(654,370,2,887863914)
(617,582,4,883789294)
(913,690,3,880824288)
(660,229,2,891406212)
(421,498,4,892241344)
(495,1091,4,888637503)
(806,421,4,882388897)
(676,538,4,892685437)
(721,262,3,877137285)
(913,209,2,881367150)
(378,78,3,880056976)
(880,476,3,880175444)
(716,204,5,879795543)
(276,1090,1,874795795)
(13,225,2,882399156)
(12,203,3,879959583)
```

Q1. Find the movie with avg rating >4 from u.data dataset.

```
ratingsByMovie = GROUP ratings BY movieID;
```

```
DUMP ratingsByMovie;
```

```
grunt> ratingsByMovie = GROUP ratings BY movieID;
```

```
grunt> DUMP ratings;
```

```
(1661,{{(751,1661,1,889299429)})}
(1662,{{(762,1662,1,878719324),(782,1662,4,891500110)})}
(1663,{{(782,1663,2,891499700)})}
(1664,{{(870,1664,4,890057322),(880,1664,4,892958799),(839,1664,1,875752902),(782,1664,4,891499699)})}
(1665,{{(782,1665,2,891500194)})}
(1666,{{(782,1666,2,891500194)})}
(1667,{{(782,1667,3,891500110)})}
(1668,{{(782,1668,3,891500067)})}
(1669,{{(782,1669,2,891500150)})}
(1670,{{(782,1670,3,891497793)})}
(1671,{{(787,1671,1,888980193)})}
(1672,{{(828,1672,2,891037722),(896,1672,2,887159554)})}
(1673,{{(835,1673,3,891034023)})}
(1674,{{(840,1674,4,891211682)})}
(1675,{{(851,1675,3,884222085)})}
(1676,{{(851,1676,2,875731674)})}
(1677,{{(854,1677,3,882814368)})}
(1678,{{(863,1678,1,889289570)})}
(1679,{{(863,1679,3,889289491)})}
(1680,{{(863,1680,2,889289570)})}
(1681,{{(896,1681,3,887160722)})}
(1682,{{(916,1682,3,880845755)}})
```

```
avgRatings = FOREACH ratingsByMovie GENERATE group AS movieID,  
AVG(ratings.rating) AS avgRating;  
DUMP avgRatings;
```

```
grunt> avgRatings = FOREACH ratingsByMovie GENERATE group AS movieID, AVG(ratings.rating) AS avgRating;
```

```
grunt> DUMP avgRatings;
```

```
(1660,2.0)  
(1661,1.0)  
(1662,2.5)  
(1663,2.0)  
(1664,3.25)  
(1665,2.0)  
(1666,2.0)  
(1667,3.0)  
(1668,3.0)  
(1669,2.0)  
(1670,3.0)  
(1671,1.0)  
(1672,2.0)  
(1673,3.0)  
(1674,4.0)  
(1675,3.0)  
(1676,2.0)  
(1677,3.0)  
(1678,1.0)  
(1679,3.0)  
(1680,2.0)  
(1681,3.0)  
(1682,3.0)
```

```
fiveStarMovies = FILTER avgRatings BY avgRating > 4.0;
```

```
grunt> fiveStarMovies = FILTER avgRatings BY avgRating > 4.0;
```

```
DUMP fiveStarMovies;
```

```
grunt> DUMP fiveStarMovies;
```

```
(1125,4.25)
(1142,4.0454545454546)
(1169,4.1)
(1189,5.0)
(1191,4.3333333333333)
(1194,4.064516129032258)
(1201,5.0)
(1203,4.0476190476190474)
(1251,4.125)
(1293,5.0)
(1367,4.2)
(1396,4.2)
(1398,4.5)
(1449,4.625)
(1467,5.0)
(1500,5.0)
(1524,4.25)
(1536,5.0)
(1594,4.5)
(1599,5.0)
(1639,4.3333333333333)
(1642,4.5)
(1653,5.0)
```

Q2. Find the oldest 5-star movies from u.data and u.item datasets.

Ans. Run the following queries

```
metadata = LOAD '/user/u.item' USING PigStorage('|')AS (movieID:int,
movieTitle:chararray, releaseDate:chararray, videoRelease:chararray,
imdbLink:chararray);
```

DUMP metadata;

```
grunt> metadata = LOAD '/user/lovedeepsingh/u.item' USING PigStorage('')AS (mo
vieID:int, movieTitle:chararray, releaseDate:chararray, videoRelease:chararray,
imdbLink:chararray);
grunt> DUMP metadata;
```

```
(1671,Further Gesture, A (1996),20-Feb-1998,,http://us.imdb.com/M/title-exact?Further+Gesture%2C+A+(1996))
(1672,Kika (1993),01-Jan-1993,,http://us.imdb.com/M/title-exact?Kika%20(1993))
(1673,Mirage (1995),01-Jan-1995,,http://us.imdb.com/M/title-exact?Mirage%20(1995))
)
(1674,Mamma Roma (1962),01-Jan-1962,,http://us.imdb.com/M/title-exact?Mamma%20Roma%20(1962))
(1675,Sunchaser, The (1996),25-Oct-1996,,http://us.imdb.com/M/title-exact?Sunchaser,%20The%20(1996))
(1676,War at Home, The (1996),01-Jan-1996,,http://us.imdb.com/M/title-exact?War%20at%20Home%2C%20The%20%281996%29)
(1677,Sweet Nothing (1995),20-Sep-1996,,http://us.imdb.com/M/title-exact?Sweet%20Nothing%20(1995))
(1678,Mat' i syn (1997),06-Feb-1998,,http://us.imdb.com/M/title-exact?Mat%27+i+syn+(1997))
(1679,B. Monkey (1998),06-Feb-1998,,http://us.imdb.com/M/title-exact?B%2E+Monkey+(1998))
(1680,Sliding Doors (1998),01-Jan-1998,,http://us.imdb.com/Title?Sliding+Doors+(1998))
(1681,You So Crazy (1994),01-Jan-1994,,http://us.imdb.com/M/title-exact?You%20So%20Crazy%20(1994))
(1682,Scream of Stone (Schrei aus Stein) (1991),08-Mar-1996,,http://us.imdb.com/M/title-exact?Schrei%20aus%20Stein%20(1991))
```

```
nameLookup = FOREACH metadata GENERATE movieID, movieTitle,
ToUnixTimeToDate(releaseDate, 'dd-MMM-yyyy')) AS releaseTime;
```

```
grunt> DUMP nameLookup;
```

```
(1660,Small Faces (1995),839529000)
(1661,New Age, The (1994),757362600)
(1662,Rough Magic (1995),864930600)
(1663,Nothing Personal (1995),862338600)
(1664,8 Heads in a Duffel Bag (1997),861301800)
(1665,Brother's Kiss, A (1997),861906600)
(1666,Ripe (1996),862511400)
(1667,Next Step, The (1995),866140200)
(1668,Wedding Bell Blues (1996),866140200)
(1669,MURDER and murder (1996),866745000)
(1670,Tainted (1998),886271400)
(1671,Further Gesture, A (1996),887913000)
(1672,Kika (1993),725826600)
(1673,Mirage (1995),788898600)
(1674,Mamma Roma (1962),-252480600)
(1675,Sunchaser, The (1996),846181800)
(1676,War at Home, The (1996),820434600)
(1677,Sweet Nothing (1995),843157800)
(1678,Mat' i syn (1997),886703400)
(1679,B. Monkey (1998),886703400)
(1680,Sliding Doors (1998),883593000)
(1681,You So Crazy (1994),757362600)
(1682,Scream of Stone (Schrei aus Stein) (1991),826223400)
```

```
DESCRIBE fiveStarMovies;
DESCRIBE nameLookup;
fiveStarsWithData = JOIN fiveStarMovies BY movieID, nameLookup      BY
movieID;
DESCRIBE fiveStarsWithData;
DUMP fiveStarsWithData;
```

```
[grunt> DESCRIBE fiveStarMovies;
fiveStarMovies: {movieID: int,avgRating: double}
grunt> ]
```

```
[grunt> DESCRIBE nameLookup;
nameLookup: {movieID: int,movieTitle: chararray,releaseTime: long}
grunt> ]
```

```
[grunt> fiveStarsWithData = JOIN fiveStarMovies BY movieID, nameLookup BY movieID
;]
```

```
grunt> DESCRIBE fiveStarsWithData;
fiveStarsWithData: {fiveStarMovies::movieID: int,fiveStarMovies::avgRating: dou
ble,nameLookup::movieID: int,nameLookup::movieTitle: chararray,nameLookup::releas
eTime: long}
```

```
grunt> DUMP fiveStarsWithData;]
```

```
(1125,4.25,1125,Innocents, The (1961),-284016600)
(1142,4.045454545454546,1142,When We Were Kings (1996),855858600)
(1169,4.1,1169,Fresh (1994),757362600)
(1189,5.0,1189,Prefontaine (1997),854044200)
(1191,4.33333333333333,1191,Letter From Death Row, A (1998),886271400)
(1194,4.064516129032258,1194,Once Were Warriors (1994),757362600)
(1201,5.0,1201,Marlene Dietrich: Shadow and Light (1996) ,828383400)
(1203,4.0476190476190474,1203,Top Hat (1935),-110455800)
(1251,4.125,1251,A Chef in Love (1996),861906600)
(1293,5.0,1293,Star Kid (1997),884889000)
(1367,4.2,1367,Faust (1994),757362600)
(1396,4.2,1396,Stonewall (1995),838319400)
(1398,4.5,1398,Anna (1996),847823400)
(1449,4.625,1449,Pather Panchali (1955),827433000)
(1467,5.0,1467,Saint of Fort Washington, The (1993),725826600)
(1500,5.0,1500,Santa with Muscles (1996),847391400)
(1524,4.25,1524,Kaspar Hauser (1993),834085800)
(1536,5.0,1536,Aiqing wansui (1994),837973800)
(1594,4.5,1594,Everest (1998),889468200)
(1599,5.0,1599,Someone Else's America (1995),831666600)
(1639,4.33333333333333,1639,Bitter Sugar (Azucar Amargo) (1996),848601000)
(1642,4.5,1642,Some Mother's Son (1996),851625000)
(1653,5.0,1653,Entertaining Angels: The Dorothy Day Story (1996),843762600)
```

```
oldestFiveStarMovies = ORDER fiveStarsWithData BY  
nameLookup::releaseTime;  
DUMP oldestFiveStarMovies;
```

```
grunt> oldestFiveStarMovies = ORDER fiveStarsWithData BY nameLookup::releaseTime  
;
```

```
grunt> DUMP oldestFiveStarMovies;
```

```
(223,4.198529411764706,223,Sling Blade (1996),848601000)  
(1639,4.33333333333333,1639,Bitter Sugar (Azucar Amargo) (1996),848601000)  
(320,4.05,320,Paradise Lost: The Child Murders at Robin Hood Hills (1996),849810  
600)  
(1642,4.5,1642,Some Mother's Son (1996),851625000)  
(313,4.2457142857142856,313,Titanic (1997),852057000)  
(272,4.2626262626263,272,Good Will Hunting (1997),852057000)  
(302,4.161616161616162,302,L.A. Confidential (1997),852057000)  
(1039,4.01111111111111,1039,Hamlet (1996),854044200)  
(1189,5.0,1189,Prefontaine (1997),854044200)  
(1007,4.127659574468085,1007,Waiting for Guffman (1996),854649000)  
(1142,4.045454545454546,1142,When We Were Kings (1996),855858600)  
(100,4.155511811023622,100,Fargo (1996),855858600)  
(181,4.007889546351085,181,Return of the Jedi (1983),858277800)  
(515,4.203980099502488,515,Boot, Das (1981),860092200)  
(1251,4.125,1251,A Chef in Love (1996),861906600)  
(251,4.260869565217392,251,Shall We Dance? (1996),868559400)  
(316,4.196428571428571,316,As Good As It Gets (1997),882815400)  
(1293,5.0,1293,Star Kid (1997),884889000)  
(1191,4.33333333333333,1191,Letter From Death Row, A (1998),886271400)  
(1594,4.5,1594,Everest (1998),889468200)  
(315,4.1,315,Apt Pupil (1998),909081000)
```

Q3. Find the oldest 3-star movies from u.data and u.item datasets.

Ans. Use the following queries

```
threeStarMovies = FILTER avgRatings BY avgRating > 2.0;
```

```
grunt> threeStarMovies = FILTER avgRatings BY avgRating > 2.0;
```

```
DUMP threeStarMovies;
```

```
grunt> DUMP threeStarMovies;
```

```
(1643,3.75)
(1645,4.0)
(1646,3.5)
(1647,3.0)
(1649,3.0)
(1650,4.0)
(1651,4.0)
(1653,5.0)
(1656,3.5)
(1657,3.0)
(1658,3.0)
(1662,2.5)
(1664,3.25)
(1667,3.0)
(1668,3.0)
(1670,3.0)
(1673,3.0)
(1674,4.0)
(1675,3.0)
(1677,3.0)
(1679,3.0)
(1681,3.0)
(1682,3.0)
```

DESCRIBE threeStarMovies;

DESCRIBE nameLookup;

threeStarsWithData = JOIN f=threeStarMovies BY movieID, nameLookup BY movieID;

DESCRIBE threeStarsWithData;

DUMP threeStarsWithData;

```
grunt> DESCRIBE threeStarMovies;
threeStarMovies: {movieID: int,avgRating: double}
```

```
grunt> DESCRIBE nameLookup;
nameLookup: {movieID: int,movieTitle: chararray,releaseTime: long}
grunt>
```

```
grunt> threeStarsWithData = JOIN threeStarMovies BY movieID, nameLookup BY movie
ID;
```

```
grunt> DESCRIBE threeStarsWithData;
threeStarsWithData: {threeStarMovies::movieID: int,threeStarMovies::avgRating: double,nameLookup::movieID: int,nameLookup::movieTitle: chararray,nameLookup::releaseTime: long}
```

```
grunt> DUMP threeStarsWithData;
```

```
(1645,4.0,1645,Butcher Boy, The (1998),883593000)
(1646,3.5,1646,Men With Guns (1997),889122600)
(1647,3.0,1647,Hana-bi (1997),890332200)
(1649,3.0,1649,Big One, The (1997),890937000)
(1650,4.0,1650,Butcher Boy, The (1998),883593000)
(1651,4.0,1651,Spanish Prisoner, The (1997),890937000)
(1653,5.0,1653,Entertaining Angels: The Dorothy Day Story (1996),843762600)
(1656,3.5,1656,Little City (1998),887913000)
(1657,3.0,1657,Target (1995),825445800)
(1658,3.0,1658,Substance of Fire, The (1996),849810600)
(1662,2.5,1662,Rough Magic (1995),864930600)
(1664,3.25,1664,8 Heads in a Duffel Bag (1997),861301800)
(1667,3.0,1667,Next Step, The (1995),866140200)
(1668,3.0,1668,Wedding Bell Blues (1996),866140200)
(1670,3.0,1670,Tainted (1998),886271400)
(1673,3.0,1673,Mirage (1995),788898600)
(1674,4.0,1674,Mamma Roma (1962),-252480600)
(1675,3.0,1675,Sunchaser, The (1996),846181800)
(1677,3.0,1677,Sweet Nothing (1995),843157800)
(1679,3.0,1679,B. Monkey (1998),886703400)
(1681,3.0,1681,You So Crazy (1994),757362600)
(1682,3.0,1682,Scream of Stone (Schrei aus Stein) (1991),826223400)
```

*oldestthreeStarMovies = ORDER threeStarsWithData BY
nameLookup::releaseTime;*

DUMP oldestthreeStarMovies;

```
grunt> oldestthreeStarMovies = ORDER threeStarsWithData BY nameLookup::releaseTi
me;
```

```
grunt> DUMP oldestthreeStarMovies;
```

```
(1613,4.0,1613,Tokyo Fist (1995),887135400)
(354,3.48611111111111,354,Wedding Singer, The (1998),887308200)
(355,2.926829268292683,355,Sphere (1998),887308200)
(1294,3.142857142857143,1294,Ayn Rand: A Sense of Life (1997),887308200)
(1313,3.142857142857143,1313,Palmetto (1998),887913000)
(1656,3.5,1656,Little City (1998),887913000)
(1646,3.5,1646,Men With Guns (1997),889122600)
(913,2.5,913,Love and Death on Long Island (1997),889468200)
(1594,4.5,1594,Everest (1998),889468200)
(912,3.0,912,U.S. Marshalls (1998),889468200)
(914,2.727272727272727,914,Wild Things (1998),889813800)
(1106,3.0,1106,Newton Boys, The (1998),889813800)
(1483,3.4166666666666665,1483,Man in the Iron Mask, The (1998),890073000)
(1647,3.0,1647,Hana-bi (1997),890332200)
(1602,3.333333333333335,1602,Price Above Rubies, A (1998),890332200)
(915,3.923076923076923,915,Primary Colors (1998),890332200)
(1525,4.0,1525,Object of My Affection, The (1998),890332200)
(1649,3.0,1649,Big One, The (1997),890937000)
(917,3.4285714285714284,917,Mercury Rising (1998),890937000)
(1651,4.0,1651,Spanish Prisoner, The (1997),890937000)
(916,3.1666666666666665,916,Lost in Space (1998),890937000)
(918,3.0,918,City of Angels (1998),891541800)
(315,4.1,315,Apt Pupil (1998),909081000)
```

Q4. Display name of all movies in uppercase.

Ans. Use the following queries.

```
uprcase = FOREACH metadata GENERATE UPPER(movieTitle);
```

```
grunt> uprcase = FOREACH metadata GENERATE UPPER(movieTitle);
```

```
DUMP uprcase;
```

```
grunt> DUMP uprcase;
```

(ENTERTAINING ANGELS: THE DOROTHY DAY STORY (1996))
(CHAIRMAN OF THE BOARD (1998))
(FAVOR, THE (1994))
(LITTLE CITY (1998))
(TARGET (1995))
(SUBSTANCE OF FIRE, THE (1996))
(GETTING AWAY WITH MURDER (1996))
(SMALL FACES (1995))
(NEW AGE, THE (1994))
(ROUGH MAGIC (1995))
(NOTHING PERSONAL (1995))
(8 HEADS IN A DUFFEL BAG (1997))
(BROTHER'S KISS, A (1997))
(RIPE (1996))
(NEXT STEP, THE (1995))
(WEDDING BELL BLUES (1996))
(MURDER AND MURDER (1996))
(TAINTED (1998))
(FURTHER GESTURE, A (1996))
(KIKA (1993))
(MIRAGE (1995))
(MAMMA ROMA (1962))
(SUNCHASER, THE (1996))
(WAR AT HOME, THE (1996))

Problem 3. For the student data analysis dataset answer the following queries:
CSE.csv, Dept.csv

Run the following commands before proceeding

*hdfs dfs -copyFromLocal CSE.csv
hdfs dfs -copyFromLocal Dept.csv*

```
(base) Lovedeeps-MacBook-Pro:Downloads lovedeepsingh$ hdfs dfs -copyFromLocal CS E.csv
```

```
(base) Lovedeeps-MacBook-Pro:Downloads lovedeepsingh$ hdfs dfs -copyFromLocal De pt.csv
```

hdfs dfs -ls

```
(base) Lovedeeps-MacBook-Pro:Downloads lovedeepsingh$ hdfs dfs -ls
2019-10-02 15:56:42,123 WARN util.NativeCodeLoader: Unable to load native-hadoop
library for your platform... using builtin-java classes where applicable
Found 13 items
-rw-r--r-- 3 lovedeepsingh supergroup 3937 2019-09-27 19:49 CSE.csv
-rw-r--r-- 3 lovedeepsingh supergroup 4579 2019-10-02 15:53 Dept.csv
-rw-r--r-- 3 lovedeepsingh supergroup 123637 2019-09-27 11:30 SalesJan2009
.csv
-rw-r--r-- 3 lovedeepsingh supergroup 78317 2019-09-27 20:39 abc.pages
-rw-r--r-- 3 lovedeepsingh supergroup 439 2019-09-27 20:48 abc.rtf
-rw-r--r-- 3 lovedeepsingh supergroup 78317 2019-09-27 20:41 abc.txt
-rw-r--r-- 3 lovedeepsingh supergroup 689413344 2019-09-12 15:00 input.csv
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:10 output1
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:10 output2
drwxr-xr-x - lovedeepsingh supergroup 0 2019-09-12 15:11 output3
drwxr-xr-x - lovedeepsingh supergroup 0 2019-10-02 00:55 pig_output_s
ales
-rw-r--r-- 3 lovedeepsingh supergroup 2079229 2019-10-02 14:28 u.data
-rw-r--r-- 3 lovedeepsingh supergroup 236344 2019-10-02 14:30 u.item
```

Now run pig to invoke grunt shell

pig

```
(base) Lovedeeps-MacBook-Pro:Downloads lovedeepsingh$ pig
```

```
2019-10-02 16:08:24,954 [main] INFO org.apache.pig.Main - Apache Pig version 0.
17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2019-10-02 16:08:24,954 [main] INFO org.apache.pig.Main - Logging error messages
to: /Users/lovedeepsingh/Downloads/pig_1570012704952.log
2019-10-02 16:08:24,981 [main] INFO org.apache.pig.impl.util.Utils - Default bo
otup file /Users/lovedeepsingh/.pigbootup not found
2019-10-02 16:08:25,288 [main] WARN org.apache.hadoop.util.NativeCodeLoader - U
nable to load native-hadoop library for your platform... using builtin-java clas
ses where applicable
2019-10-02 16:08:25,320 [main] INFO org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.add
ress
2019-10-02 16:08:25,320 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
2019-10-02 16:08:25,903 [main] INFO org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.add
ress
2019-10-02 16:08:25,903 [main] INFO org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to map-reduce job tracker at: localhost:9010
2019-10-02 16:08:25,924 [main] INFO org.apache.pig.PigServer - Pig Script ID fo
r the session: PIG-default-e9ca5e90-dd6f-4972-b03f-88d59adfc0d9
2019-10-02 16:08:25,924 [main] WARN org.apache.pig.PigServer - ATS is disabled
since yarn.timeline-service.enabled set to false
grunt>
```

Q1. To group the tuples of students base on city.

Run the following queries .

```
data = LOAD '/user/lovedeepsingh/CSE.csv' USING PigStorage(',') AS  
(Name:chararray, RollNO:int, Marks:int, City:chararray);
```

```
[grunt> data = LOAD '/user/lovedeepsingh/CSE.csv' USING PigStorage(',') AS (Name:  
chararray, RollNO:int, Marks:int, City:chararray);
```

DUMP data;

```
[grunt> DUMP data;
```

```
(Shivam Jand ,16103100,153,Gujrat,,,)  
(Abhishek Garg,16103101,154,Gujrat,,,)  
(Sahil Bhardwaj,16103102,155,Pune,,,)  
(Pragun Aggarwal,16103103,156,Pune,,,)  
(Lovedeep Singh,16103104,157,Pune,,,)  
( Kainaat ,16103105,158,Pune,,,)  
(Bhalinder Singh Munier,16103106,159,Pune,,,)  
(Neeti Jethlia,16103107,160,Pune,,,)  
(Shagundeepr Singh,16103108,161,Pune,,,)  
(Muskan Sharma,16103109,162,Pune,,,)  
(Shubhkarmandeep Kaur,16103110,163,Pune,,,)  
(Kunal Gupta,16103111,164,Pune,,,)  
(Tanishq Mittal,16103112,165,Pune,,,)  
(Pranita,16103113,166,Pune,,,)  
(Simran Bajaj,16103115,167,Pune,,,)  
(Vardaan Bajaj,16103116,168,Pune,,,)  
(Harsimran Singh,16103117,169,Pune,,,)  
(Kanishk Gautam,16103118,170,Pune,,,)  
(Sanyam Mehta,16103119,171,Pune,,,)  
(Gagandeep Chugh,16103120,172,Pune,,,)  
(Karan Chabra,16103121,173,Pune,,,)  
(Sanyam Dhawan,16103122,174,Pune,,,)  
(Guneet Bajaj,16103123,175,Pune,,,)
```

GroupByCity = GROUP data BY City;

```
[grunt> GroupByCity = GROUP data BY City;
```

DUMP GroupByCity;

```

une,,),(Kushal Dudeja,16103047,103,Pune,,),(Abhinav Chawla,16103046,102,Pune,,),
,(Parikh Oberoi,16103045,101,Pune,,),(Vaibhav Goel,16103044,100,Pune,,),(Pri
tish Gupta,16103043,99,Panjab,,),(Nirmaljot Singh,16103042,98,Panjab,,),(Madha
v Pruthi,16103041,97,Panjab,,),(Chetna Gupta,16103040,96,Panjab,,),(Kartik Kau
shal,16103039,95,Panjab,,),(Shivam Sehgal,16103037,94,Panjab,,),(Mayank Saggar
,16103036,93,Panjab,,),(Krittika Chhabra,16103035,92,Panjab,,),(Arshdeep Singh
Sachdeva,16103034,91,Panjab,,),(Priyanshu Sharma,16103033,90,Panjab,,),(Balji
nder Singh,16103032,89,Panjab,,),(Janamdeep Singh,16103031,88,Himachal,,),(Har
shima Singla,16103030,87,Himachal,,),(Leevansha,16103029,86,Himachal,,),(Aakan
kasha Sharma,16103028,85,Himachal,,),(Ritesh Goyal,16103027,84,Himachal,,),(Dh
anwant Kaur,16103026,83,Himachal,,),(Rohan Dhadwal,16103025,82,Himachal,,),(Ar
unav Sharma,16103024,81,Chandigarh,,),(Pradhuman Goyal,16103023,80,Chandigarh,,,
),(Nikita Gupta,16103022,79,Chandigarh,,),(Mankeerat Singh ,16103021,78,Chandi
garh,,),(Ishneet Kaur,16103020,77,Chandigarh,,),(Muskan Garg,16103019,76,Chand
igarh,,),(Pranjali,16103018,75,Chandigarh,,),(Deepak Tiwari,16103017,74,Chandi
garh,,),(Gaurav Ahuja,16103016,73,Chandigarh,,),(Abhinav Agarwal,16103015,72,N
ew Delhi,,),(Shounak Chowdhury ,16103014,71>New Delhi,,),(Abhinav Singla,16103
013,70,Chandigarh,,),(Sparsh Garg,16103012,69,Chandigarh,,),(Sarthak Vaswani,1
6103011,68>New Delhi,,),(Pradeep Parajuli,16103010,67>New Delhi,,),(Ramesh Cha
ndra Pokharel,16103008,66>New Delhi,,),(Nikhat Parveen,16103007,65>New Delhi,,,
),(Sanjana Koul,16103006,64>New Delhi,,),(Rajesh Jha,16103004,63,Hyderabad,,),
(Prayag Panta,16103003,62,Hyderabad,,),(Sidharth Bhandari,16103001,61,Hyderabad
,,),(Name,RollNo,Marks,City,,))}
```

Q2. To display the name of all students in upper case.

Ans. Run the following queries.

```

uprcase = FOREACH data GENERATE UPPER(Name);
DUMP uprcase;
```

```

-----.
grunt> uprcase = FOREACH data GENERATE UPPER(Name);
```

```

grunt> DUMP uprcase;
```

```

(ABHISHEK GARG)
(SAHIL BHARDWAJ)
(PRAGUN AGGARWAL)
(LOVEDEEP SINGH)
( KAINAAT )
(BHALINDER SINGH MUNIER)
(NEETI JETHLIA)
(SHAGUNDEEP SINGH)
(MUSKAN SHARMA)
(SHUBHKARMANDEEP KAUR)
(KUNAL GUPTA)
(TANISHQ MITTAL)
(PRANITA)
(SIMRAN BAJAJ)
(VARDAAN BAJAJ)
(HARSIMRAN SINGH)
(KANISHK GAUTAM)
(SANIVAM MEHTA)
```

Q3. To find the tuples of those students where the marks is greater than equal to 70.

Ans. Run the following queries.

greaterThanSeventy = FILTER data BY Marks > 70;

DUMP greaterThanSeventy;

```
grunt> greaterThanSeventy = FILTER data BY Marks > 70;
grunt> DUMP greaterThanSeventy;
```

```
(Sahil Bhardwaj,16103102,155,Pune)
(Pragun Aggarwal,16103103,156,Pune)
(Lovedeep Singh,16103104,157,Pune)
(Kainaat ,16103105,158,Pune)
(Bhalinder Singh Munier,16103106,159,Pune)
(Neeti Jethlia,16103107,160,Pune)
(Shagundeep Singh,16103108,161,Pune)
(Muskan Sharma,16103109,162,Pune)
(Shubhkarmandeep Kaur,16103110,163,Pune)
(Kunal Gupta,16103111,164,Pune)
(Tanishq Mittal,16103112,165,Pune)
(Pranita,16103113,166,Pune)
(Simran Bajaj,16103115,167,Pune)
(Vardaan Bajaj,16103116,168,Pune)
(Harsimran Singh,16103117,169,Pune)
```

Q4. To get details of students who belongs to New Delhi only.

Ans. Run the following queries.

NewDelhiOnly = FILTER data BY City == 'New Delhi';

DUMP NewDelhiOnly;

```
grunt> NewDelhiOnly = FILTER data BY City == 'New Delhi';
grunt> DUMP NewDelhiOnly;
```

```
(Sanjana Koul,16103006,64,New Delhi)
(Nikhat Parveen,16103007,65,New Delhi)
(Ramesh Chandra Pokharel,16103008,66,New Delhi)
(Pradeep Parajuli,16103010,67,New Delhi)
(Sarthak Vaswani,16103011,68,New Delhi)
(Shounak Chowdhury ,16103014,71,New Delhi)
(Abhinav Agarwal,16103015,72,New Delhi)
```

Q5. To display the results where the city name Chandigarh is replaced with shorter form CHD and New Delhi with NDLS.

Ans. Run the following queries.

```
replaceDataChd = FOREACH data GENERATE  
Name,RollNO,Marks,REPLACE(City, 'Chandigarh', 'CHD') AS City;  
replaceData = FOREACH replaceDataChd GENERATE  
Name,RollNO,Marks,REPLACE(City, 'New Delhi', 'NDLS') AS City;  
DUMP replaceData;
```

```
grunt> replaceDataChd = FOREACH data GENERATE Name,RollNO,Marks, REPLACE(City, '  
Chandigarh', 'CHD') AS City;
```

```
grunt> replaceData = FOREACH replaceDataChd GENERATE Name,RollNO,Marks,REPLACE(C  
ity, 'New Delhi', 'NDLS') AS City;
```

```
grunt> DUMP replaceData;
```

```
(Ramesh Chandra Pokharel,16103008,66,NDLS)  
(Pradeep Parajuli,16103010,67,NDLS)  
(Sarthak Vaswani,16103011,68,NDLS)  
(Sparsh Garg,16103012,69,CHD)  
(Abhinav Singla,16103013,70,CHD)  
(Shounak Chowdhury ,16103014,71,NDLS)  
(Abhinav Agarwal,16103015,72,NDLS)  
(Gaurav Ahuja,16103016,73,CHD)  
(Deepak Tiwari,16103017,74,CHD)  
(Pranjali,16103018,75,CHD)  
(Muskan Garg,16103019,76,CHD)  
(Ishneet Kaur,16103020,77,CHD)  
(Mankeerat Singh ,16103021,78,CHD)  
(Nikita Gupta,16103022,79,CHD)  
(Pradhuman Goyal,16103023,80,CHD)  
(Arunav Sharma,16103024,81,CHD)  
(Rohan Dhadwal,16103025,82,Himachal)  
(Dhanwant Kaur,16103026,83,Himachal)  
(Ritesh Goyal,16103027,84,Himachal)  
(Aakanksha Sharma,16103028,85,Himachal)
```

Q6. To calculate average marks for each student.

Ans. Run the following queries.

GroupByRollNO = GROUP data BY RollNO;

```
grunt> GroupByRollNO = GROUP data BY RollNO;
```

DUMP GroupByRollNO;

```
grunt> DUMP GroupByRollNO;
```

```
(16103101,{{Abhishek Garg,16103101,154,Gujrat}})  
(16103102,{{Sahil Bhardwaj,16103102,155,Pune}})  
(16103103,{{Pragun Aggarwal,16103103,156,Pune}})  
(16103104,{{Lovedeep Singh,16103104,157,Pune}})  
(16103105,{{Kainaat ,16103105,158,Pune}})  
(16103106,{{Bhalinder Singh Munier,16103106,159,Pune}})  
(16103107,{{Neeti Jethlia,16103107,160,Pune}})  
(16103108,{{Shagundeepraj Singh,16103108,161,Pune}})  
(16103109,{{Muskan Sharma,16103109,162,Pune}})  
(16103110,{{Shubhkarmandeep Kaur,16103110,163,Pune}})  
(16103111,{{Kunal Gupta,16103111,164,Pune}})  
(16103112,{{Tanishq Mittal,16103112,165,Pune}})  
(16103113,{{Pranita,16103113,166,Pune}})  
(16103115,{{Simran Bajaj,16103115,167,Pune}})  
(16103116,{{Vardaan Bajaj,16103116,168,Pune}})  
(16103117,{{Harsimran Singh,16103117,169,Pune}})  
(16103118,{{Kanishk Gautam,16103118,170,Pune}})  
(16103119,{{Sanyam Mehta,16103119,171,Pune}})  
(16103120,{{Gagandeep Chugh,16103120,172,Pune}})  
(16103121,{{Karan Chabra,16103121,173,Pune}})  
(16103122,{{Sanyam Dhawan,16103122,174,Pune}})  
(16103123,{{Guneet Bajaj,16103123,175,Pune}})  
(,{(Name,,,City)})
```

avgMarks = FOREACH GroupByRollNO GENERATE group, AVG(data.Marks) AS avgmarks;

```
grunt> avgMarks = FOREACH GroupByRollNO GENERATE group, AVG(data.Marks) AS avgmarks;
```

DUMP avgMarks;

```
grunt> DUMP avgMarks;
```

```
(16103101,154.0)
(16103102,155.0)
(16103103,156.0)
(16103104,157.0)
(16103105,158.0)
(16103106,159.0)
(16103107,160.0)
(16103108,161.0)
(16103109,162.0)
(16103110,163.0)
(16103111,164.0)
(16103112,165.0)
(16103113,166.0)
(16103115,167.0)
(16103116,168.0)
(16103117,169.0)
(16103118,170.0)
(16103119,171.0)
(16103120,172.0)
(16103121,173.0)
```

Q7. To merge contents of two relation CSE.csv and Dept.csv.

Ans. Run the following queries .

```
deptData = LOAD 'user/lovedeepsingh/Dept.csv' USING PigStorage(',') AS
(RollNO:int, Department:chararray, City:chararray, College:chararray);
```

DUMP deptData;

```
grunt> deptData = LOAD '/user/lovedeepsingh/Dept.csv' USING PigStorage(',') AS (
RollNO:int, Department:chararray, City:chararray, College:chararray);
```

```
grunt> DUMP deptData;
```

```
(16103101,Electronics,Gujrat,PEC)
(16103102,Electronics,Pune,PEC)
(16103103,Computer Science,Pune,PEC)
(16103104,Computer Science,Pune,PEC)
(16103105,Computer Science,Pune,PEC)
(16103106,Computer Science,Pune,PEC)
(16103107,Computer Science,Pune,PEC)
(16103108,Computer Science,Pune,PEC)
(16103109,Computer Science,Pune,PEC)
(16103110,Computer Science,Pune,PEC)
(16103111,Computer Science,Pune,PEC)
(16103112,Computer Science,Pune,PEC)
(16103113,Computer Science,Pune,PEC)
(16103115,Computer Science,Pune,PEC)
(16103116,Mechanical,Pune,PEC)
(16103117,Mechanical,Pune,PEC)
(16103118,Mechanical,Pune,PEC)
(16103119,Mechanical,Pune,PEC)
(16103120,Mechanical,Pune,PEC)
(16103121,Civil,Pune,PEC)
(16103122,Civil,Pune,PEC)
(16103123,Civil,Pune,PEC)
```

mergeTable = JOIN data BY RollNO, deptData BY RollNO;

```
grunt> mergeTable = JOIN data BY RollNO, deptData BY RollNO;
```

DESCRIBE mergeTable;

```
grunt> DESCRIBE mergeTable;
mergeTable: {data::Name: chararray,data::RollNO: int,data::Marks: int,data::City
: chararray,deptData::RollNO: int,deptData::Department: chararray,deptData::City
: chararray,deptData::College: chararray}
```

DUMP mergeTable;

```
grunt> DUMP mergeTable;
```

```
(Shivam Jand ,16103100,153,Gujrat,16103100,Electronics,Gujrat,PEC)
(Abhisek Garg,16103101,154,Gujrat,16103101,Electronics,Gujrat,PEC)
(Sahil Bhardwaj,16103102,155,Pune,16103102,Electronics,Pune,PEC)
(Pragun Aggarwal,16103103,156,Pune,16103103,Computer Science,Pune,PEC)
(Lovedeep Singh,16103104,157,Pune,16103104,Computer Science,Pune,PEC)
(Kainaat ,16103105,158,Pune,16103105,Computer Science,Pune,PEC)
(Bhalinder Singh Munier,16103106,159,Pune,16103106,Computer Science,Pune,PEC)
(Neeti Jethlia,16103107,160,Pune,16103107,Computer Science,Pune,PEC)
(Shagundeepr Singh,16103108,161,Pune,16103108,Computer Science,Pune,PEC)
(Muskan Sharma,16103109,162,Pune,16103109,Computer Science,Pune,PEC)
(Shubhkarmandeep Kaur,16103110,163,Pune,16103110,Computer Science,Pune,PEC)
(Kunal Gupta,16103111,164,Pune,16103111,Computer Science,Pune,PEC)
(Tanishq Mittal,16103112,165,Pune,16103112,Computer Science,Pune,PEC)
(Pranita,16103113,166,Pune,16103113,Computer Science,Pune,PEC)
(Simran Bajaj,16103115,167,Pune,16103115,Computer Science,Pune,PEC)
(Vardaan Bajaj,16103116,168,Pune,16103116,Mechanical,Pune,PEC)
(Harsimran Singh,16103117,169,Pune,16103117,Mechanical,Pune,PEC)
(Kanishk Gautam,16103118,170,Pune,16103118,Mechanical,Pune,PEC)
(Sanyam Mehta,16103119,171,Pune,16103119,Mechanical,Pune,PEC)
(Gagandeep Chugh,16103120,172,Pune,16103120,Mechanical,Pune,PEC)
(Karan Chabra,16103121,173,Pune,16103121,Civil,Pune,PEC)
(Sanyam Dhawan,16103122,174,Pune,16103122,Civil,Pune,PEC)
(Guneet Bajaj,16103123,175,Pune,16103123,Civil,Pune,PEC)
```

Q8. To display the details of students who belongs to computer Science only.

Ans. Run the following queries.

CSonly = FILTER mergeTable BY Department == 'Computer Science';

```
grunt> CSonly = FILTER mergeTable BY Department == 'Computer Science';
```

DUMP CSonly;

```
grunt> DUMP CSonly;
```

```
(Prabhjot Kaur,16103079,134,Chennai,16103079,Computer Science,Chennai,PEC)
(Smily Rana,16103080,135,Chennai,16103080,Computer Science,Chennai,PEC)
(Akshat Kasana,16103081,136,Chennai,16103081,Computer Science,Chennai,PEC)
(Shubam Bharti,16103082,137,Chennai,16103082,Computer Science,Chennai,PEC)
(Priyam,16103083,138,Chennai,16103083,Computer Science,Chennai,PEC)
(Riyanka Mondal,16103084,139,Chennai,16103084,Computer Science,Chennai,PEC)
(Gurvir singh,16103085,140,Chennai,16103085,Computer Science,Chennai,PEC)
(Anil Kumar,16103086,141,Chennai,16103086,Computer Science,Chennai,PEC)
(Divya Sehjal,16103087,142,Chennai,16103087,Computer Science,Chennai,PEC)
(Srishti,16103089,143,Chennai,16103089,Computer Science,Chennai,PEC)
(Pragun Aggarwal,16103103,156,Pune,16103103,Computer Science,Pune,PEC)
(Lovedeep Singh,16103104,157,Pune,16103104,Computer Science,Pune,PEC)
(Kainaat ,16103105,158,Pune,16103105,Computer Science,Pune,PEC)
(Bhalinder Singh Munier,16103106,159,Pune,16103106,Computer Science,Pune,PEC)
```

Q9. To display the details of students who does not belongs to computer science.

Ans. Run the following queries.

NotCS = FILTER mergeTable BY Department != 'Computer Science';

```
|> grunt> NotCS = FILTER mergeTable BY Department != 'Computer Science';|
```

DUMP NotCS;

```
|> grunt> DUMP NotCS;|
```

```
(Janamdeep Singh,16103031,88,Himachal,16103031,Electrical ,Himachal,IIT Delhi)
(Baljinder Singh,16103032,89,Panjab,16103032,Electrical ,Panjab,IIT Delhi)
(Priyanshu Sharma,16103033,90,Panjab,16103033,Electrical ,Panjab,IIT Delhi)
(Prabhnoor Kaur,16103090,144,Chennai,16103090,Electronics,Chennai,PEC)
(Noordeep Kaur,16103091,145,Chennai,16103091,Electronics,Chennai,PEC)
(Abhinav Thakur,16103093,146,Chennai,16103093,Electronics,Chennai,PEC)
(Kanika Sharma,16103094,147,Himachal,16103094,Electronics,Himachal,PEC)
(Akshant Sarmal,16103095,148,Chennai,16103095,Electronics,Chennai,PEC)
(Dhruv Guleria,16103096,149,Gujrat,16103096,Electronics,Gujrat,PEC)
(Abhishek,16103097,150,Gujrat,16103097,Electronics,Gujrat,PEC)
(Kshitij Porwal,16103098,151,Gujrat,16103098,Electronics,Gujrat,PEC)
(Raghav Sharma,16103099,152,Gujrat,16103099,Electronics,Gujrat,PEC)
(Shivam Jand ,16103100,153,Gujrat,16103100,Electronics,Gujrat,PEC)
(Abhishek Garg,16103101,154,Gujrat,16103101,Electronics,Gujrat,PEC)
(Sahil Bhardwaj,16103102,155,Pune,16103102,Electronics,Pune,PEC)
(Vardaan Bajaj,16103116,168,Pune,16103116,Mechanical,Pune,PEC)
(Harsimran Singh,16103117,169,Pune,16103117,Mechanical,Pune,PEC)
(Kanishk Gautam,16103118,170,Pune,16103118,Mechanical,Pune,PEC)
(Sanyam Mehta,16103119,171,Pune,16103119,Mechanical,Pune,PEC)
(Gagandeep Chugh,16103120,172,Pune,16103120,Mechanical,Pune,PEC)
(Karan Chabra,16103121,173,Pune,16103121,Civil,Pune,PEC)
(Sanyam Dhawan,16103122,174,Pune,16103122,Civil,Pune,PEC)
(Guneet Bajaj,16103123,175,Pune,16103123,Civil,Pune,PEC)
```

Q10. To display the details of students who belongs to PEC only.

PEConly = FILTER mergeTable BY College == 'PEC';

```
|> grunt> PEConly = FILTER mergeTable BY College == 'PEC';|
```

DUMP PEConly;

```
grunt> DUMP PEConly;
```

```
(Shivam Jand ,16103100,153,Gujrat,16103100,Electronics,Gujrat,PEC)
(Abhisek Garg,16103101,154,Gujrat,16103101,Electronics,Gujrat,PEC)
(Sahil Bhardwaj,16103102,155,Pune,16103102,Electronics,Pune,PEC)
(Pragun Aggarwal,16103103,156,Pune,16103103,Computer Science,Pune,PEC)
(Lovedeep Singh,16103104,157,Pune,16103104,Computer Science,Pune,PEC)
(Kainaat ,16103105,158,Pune,16103105,Computer Science,Pune,PEC)
(Bhalinder Singh Munier,16103106,159,Pune,16103106,Computer Science,Pune,PEC)
(Neeti Jethlia,16103107,160,Pune,16103107,Computer Science,Pune,PEC)
(Shagundeepr Singh,16103108,161,Pune,16103108,Computer Science,Pune,PEC)
(Muskan Sharma,16103109,162,Pune,16103109,Computer Science,Pune,PEC)
(Shubhkarmandeep Kaur,16103110,163,Pune,16103110,Computer Science,Pune,PEC)
(Kunal Gupta,16103111,164,Pune,16103111,Computer Science,Pune,PEC)
(Tanishq Mittal,16103112,165,Pune,16103112,Computer Science,Pune,PEC)
(Pranita,16103113,166,Pune,16103113,Computer Science,Pune,PEC)
(Simran Bajaj,16103115,167,Pune,16103115,Computer Science,Pune,PEC)
(Vardaan Bajaj,16103116,168,Pune,16103116,Mechanical,Pune,PEC)
(Harsimran Singh,16103117,169,Pune,16103117,Mechanical,Pune,PEC)
(Kanishk Gautam,16103118,170,Pune,16103118,Mechanical,Pune,PEC)
(Sanyam Mehta,16103119,171,Pune,16103119,Mechanical,Pune,PEC)
(Gagandeep Chugh,16103120,172,Pune,16103120,Mechanical,Pune,PEC)
(Karan Chabra,16103121,173,Pune,16103121,Civil,Pune,PEC)
(Sanyam Dhawan,16103122,174,Pune,16103122,Civil,Pune,PEC)
(Guneet Bajaj,16103123,175,Pune,16103123,Civil,Pune,PEC)
```

Q11. To display the details of students who does not belong to IIT Roorkee.

Ans. Run the following queries.

NotIITRoorkee = FILTER mergeTable BY College != 'IIT Roorkee';

```
grunt> NotIITRoorkee = FILTER mergeTable BY College != 'IIT Roorkee';
```

DUMP NotIITRoorkee;

```
grunt> DUMP NotIITRoorkee;
```

(Shivam Jand ,16103100,153,Gujrat,16103100,Electronics,Gujrat,PEC)
(Abhishek Garg,16103101,154,Gujrat,16103101,Electronics,Gujrat,PEC)
(Sahil Bhardwaj,16103102,155,Pune,16103102,Electronics,Pune,PEC)
(Pragun Aggarwal,16103103,156,Pune,16103103,Computer Science,Pune,PEC)
(Lovedeep Singh,16103104,157,Pune,16103104,Computer Science,Pune,PEC)
(Kainaat ,16103105,158,Pune,16103105,Computer Science,Pune,PEC)
(Bhalinder Singh Munier,16103106,159,Pune,16103106,Computer Science,Pune,PEC)
(Neeti Jethlia,16103107,160,Pune,16103107,Computer Science,Pune,PEC)
(Shagundeepr Singh,16103108,161,Pune,16103108,Computer Science,Pune,PEC)
(Muskan Sharma,16103109,162,Pune,16103109,Computer Science,Pune,PEC)
(Shubhkarmandeep Kaur,16103110,163,Pune,16103110,Computer Science,Pune,PEC)
(Kunal Gupta,16103111,164,Pune,16103111,Computer Science,Pune,PEC)
(Tanishq Mittal,16103112,165,Pune,16103112,Computer Science,Pune,PEC)
(Pranita,16103113,166,Pune,16103113,Computer Science,Pune,PEC)
(Simran Bajaj,16103115,167,Pune,16103115,Computer Science,Pune,PEC)
(Vardaan Bajaj,16103116,168,Pune,16103116,Mechanical,Pune,PEC)
(Harsimran Singh,16103117,169,Pune,16103117,Mechanical,Pune,PEC)
(Kanishk Gautam,16103118,170,Pune,16103118,Mechanical,Pune,PEC)
(Sanyam Mehta,16103119,171,Pune,16103119,Mechanical,Pune,PEC)
(Gagandeep Chugh,16103120,172,Pune,16103120,Mechanical,Pune,PEC)
(Karan Chabra,16103121,173,Pune,16103121,Civil,Pune,PEC)
(Sanyam Dhawan,16103122,174,Pune,16103122,Civil,Pune,PEC)
(Guneet Bajaj,16103123,175,Pune,16103123,Civil,Pune,PEC)

END OF ASSIGNMENT