Project 1.1:

Checking list of files in Hadoop.

Hadoop fs –ls /hadoopdata.

```
hadoop fs -ls /hadoopdata
```

Created folder acadgild mini project .

Hadoop fs –ls /hadoopdata/aminiproject.

```
hadoop fs -ls /hadoopdata/agminiproject
```

Imported Crimes.csv file into agminiproject.

Hadoop fs –ls /hadoopdata/aminiproject/Crimes.csv

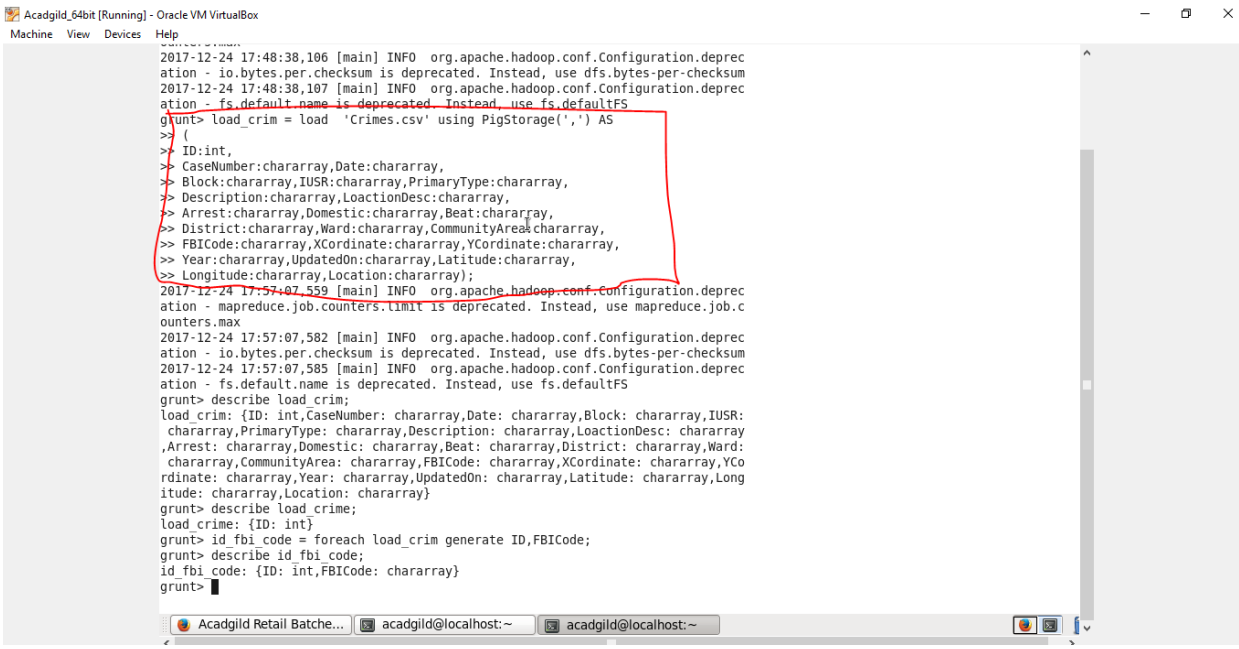
```
/hadoopdata/agminiproject/Crimes.csv
```

For showing data in Crimes.csv file

Hadoop fs –cat /hadoopdata/aminiproject/Crimes.csv

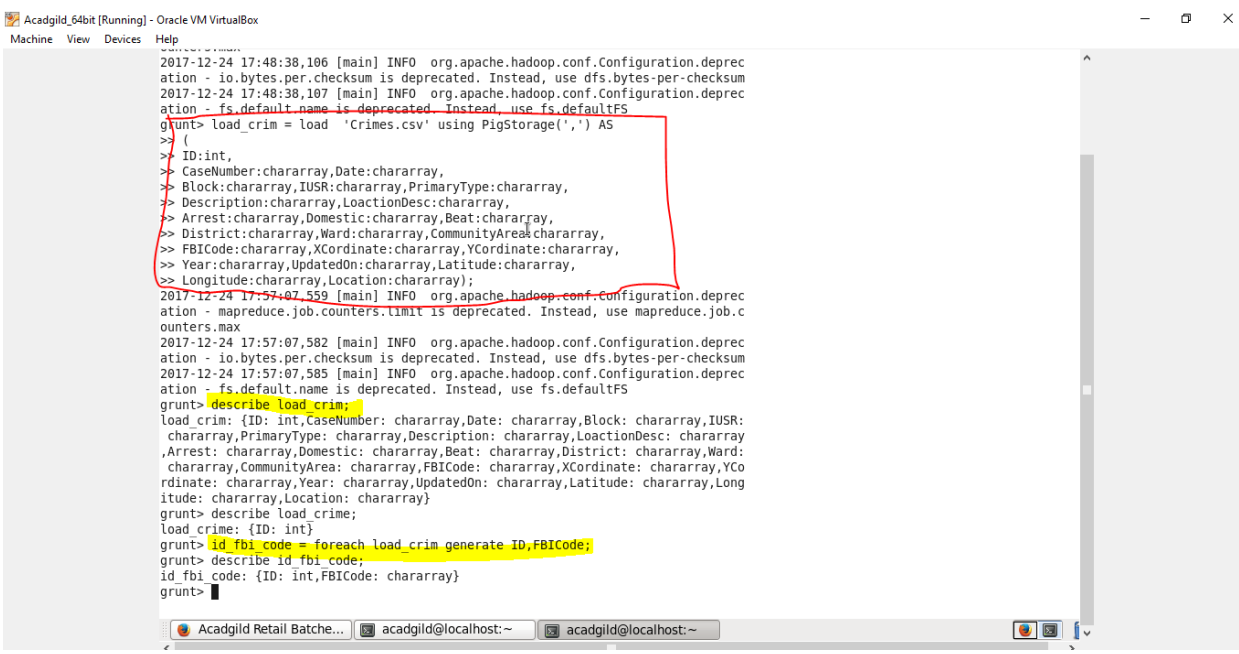
1. Write a MapReduce/Pig program to calculate the number of cases investigated under each FBI code

Using Pig :



```
2017-12-24 17:48:38,106 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-12-24 17:48:38,107 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> load_crim = load  'Crimes.csv' using PigStorage(',') AS
>> (
>> ID:int,
>> CaseNumber:chararray,Date:chararray,
>> Block:chararray,IUSR:chararray,PrimaryType:chararray,
>> Description:chararray,LoactionDesc:chararray,
>> Arrest:chararray,Domestic:chararray,Beat:chararray,
>> District:chararray,Ward:chararray,CommunityArea:chararray,
>> FBICode:chararray,XCordinate:chararray,YCordinate:chararray,
>> Year:chararray,UpdatedOn:chararray,Latitude:chararray,
>> Longitude:chararray,Location:chararray);
2017-12-24 17:57:07,559 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-12-24 17:57:07,582 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-12-24 17:57:07,585 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> describe load_crim;
load_crim: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR:
 chararray,PrimaryType: chararray,Description: chararray,LoactionDesc: chararray
,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward:
 chararray,CommunityArea: chararray,FBICode: chararray,XCordinate: chararray,YCo
rdinate: chararray,Year: chararray,UpdatedOn: chararray,Latitude: chararray,Long
itude: chararray,Location: chararray}
grunt> describe load_crime;
load_crime: {ID: int}
grunt> id_fbi_code = foreach load_crim generate ID,FBICode;
grunt> describe id_fbi_code;
id_fbi_code: {ID: int,FBICode: chararray}
grunt>
```

Loading in fbi code:



```
2017-12-24 17:48:38,106 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-12-24 17:48:38,107 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> load_crim = load  'Crimes.csv' using PigStorage(',') AS
>> (
>> ID:int,
>> CaseNumber:chararray,Date:chararray,
>> Block:chararray,IUSR:chararray,PrimaryType:chararray,
>> Description:chararray,LoactionDesc:chararray,
>> Arrest:chararray,Domestic:chararray,Beat:chararray,
>> District:chararray,Ward:chararray,CommunityArea:chararray,
>> FBICode:chararray,XCordinate:chararray,YCordinate:chararray,
>> Year:chararray,UpdatedOn:chararray,Latitude:chararray,
>> Longitude:chararray,Location:chararray);
2017-12-24 17:57:07,559 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapreduce.job.counters.limit is deprecated. Instead, use mapreduce.job.c
ounters.max
2017-12-24 17:57:07,582 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-12-24 17:57:07,585 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> describe load_crim;
load_crim: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR:
 chararray,PrimaryType: chararray,Description: chararray,LoactionDesc: chararray
,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward:
 chararray,CommunityArea: chararray,FBICode: chararray,XCordinate: chararray,YCo
rdinate: chararray,Year: chararray,UpdatedOn: chararray,Latitude: chararray,Long
itude: chararray,Location: chararray}
grunt> describe load_crime;
load_crime: {ID: int}
grunt> id_fbi_code = foreach load_crim generate ID,FBICode;
grunt> describe id_fbi_code;
id_fbi_code: {ID: int,FBICode: chararray}
grunt>
```

Grouping the fbi code by using fbi code:

```
grunt> group_fbi = group id_fbi_code by FBICode;
grunt> describe group_fbi;
group_fbi: {group: chararray,id_fbi_code: {(ID: int,FBICode: chararray)}}
grunt>
```

Acadgild Retail Batche... | acadgild@localhost:~ | acadgild@localhost:~

After dumping data above group_fbi:

Help

```
27,08B),(10116784,08B),(10116830,08B),(10116740,08B),(10116734,08B),(10121294,08B),(10116801,08B),(10120825,08B),(10116838,08
B),(9733429,08B),(10130137,08B),(10116706,08B),(10128777,08B),(9733340,08B),(10116630,08B),(10116748,08B),(10127891,08B),(101
16725,08B),(10116821,08B),(10116697,08B),(10127969,08B),(10130398,08B),(10117828,08B),(10127910,08B),(10116617,08B),(10116735
,08B),(10127925,08B),(10116809,08B),(10116728,08B),(10117829,08B),(10116551,08B),(10117015,08B),(10116660,08B),(10127927,08B)
,(10116640,08B),(10129631,08B),(10130435,08B),(10116807,08B),(10127930,08B),(10128865,08B),(10127970,08B),(10129367,08B),(101
27941,08B),(10128321,08B),(10127979,08B),(10118101,08B),(10116580,08B),(10116711,08B),(10116644,08B),(10116492,08B),(10127975
,08B),(10127977,08B),(10116598,08B),(10117446,08B),(10116510,08B),(10116460,08B),(10116973,08B),(10118934,08B),(10116364,08B)
,(10127994,08B),(10119507,08B),(10116494,08B),(10116548,08B),(10128320,08B),(10116233,08B),(10116412,08B),(10117054,08B),(101
28023,08B),(10116408,08B),(10116336,08B),(10116312,08B),(10116358,08B),(10117344,08B),(10128413,08B),(10116317,08B),(10116698
,08B),(10116163,08B),(10116117,08B),(10128152,08B),(10116625,08B),(10116432,08B),(10115889,08B),(10116051,08B),(10115742,08B)
,(10115772,08B),(10116156,08B),(10115738,08B),(10116453,08B),(10128486,08B),(10116283,08B),(10115705,08B),(10116144,08B),(101
15762,08B),(10128263,08B),(10115776,08B),(10117418,08B),(10115467,08B),(10115455,08B),(10115463,08B),(10128294,08B),(10128148
,08B),(10115452,08B),(10115711,08B),(10115456,08B),(10116392,08B),(10115775,08B),(10115418,08B),(10115389,08B),(10115427,08B)
,(10128157,08B),(10115419,08B),(10128437,08B),(10128524,08B),(10128348,08B),(10128672,08B),(10115399,08B),(10115421,08B),(101
15402,08B),(10115431,08B),(10128404,08B),(10115472,08B),(10130005,08B),(10115400,08B),(10115387,08B),(10115362,08B),(10116988
,08B),(10124825,08B),(10128803,08B),(10115324,08B),(10115349,08B),(10115388,08B),(10115415,08B),(10115359,08B),(10115382,08B)
,(10115367,08B),(10120980,08B),(10134541,08B),(10128671,08B),(10115417,08B),(10115405,08B),(10128619,08B),(10115962,08B),(101
15303,08B),(10128429,08B),(10115295,08B),(10115438,08B),(9733640,08B),(10128539,08B),(10115334,08B),(10128496,08B),(10124675,
08B),(10115252,08B),(10115294,08B),(10125652,08B),(10128778,08B),(10116482,08B),(10115178,08B),(10115319,08B),(10128566,08B),
(10128705,08B),(10115214,08B),(10115179,08B),(10115263,08B),(10115102,08B),(10115075,08B),(10115188,08B),(10115121,08B),(1011
6573,08B),(10128644,08B),(10128737,08B),(10128481,08B),(10115142,08B),(10129408,08B),(10115093,08B),(10115124,08B),(10128668,
08B),(10116508,08B),(10115129,08B),(10115168,08B),(10115036,08B),(10128592,08B),(10117264,08B),(10116681,08B),(10115019,08B),
(10117864,08B),(10114983,08B),(10114974,08B),(10114981,08B),(10117553,08B),(9733437,08B),(10114967,08B),(10114966,08B),(10114
896,08B),(10128960,08B),(10114888,08B),(10114909,08B),(10129800,08B),(10114965,08B),(10129405,08B),(10114747,08B),(10114786,0
8B),(10114645,08B),(10129660,08B),(10114638,08B),(10114900,08B),(10114772,08B),(10115151,08B),(10114591,08B),(10114632,08B),(
10117106,08B),(10129130,08B),(10129808,08B),(10114819,08B),(10116773,08B),(10116244,08B),(10114808,08B),(10114458,08B),(10123
546,08B),(10114434,08B),(10114473,08B),(10114461,08B),(10115030,08B),(10132342,08B),(10114765,08B),(10114477,08B),(10129434,0
8B),(10114378,08B),(10128763,08B),(9894527,08B),(10114752,08B),(10116258,08B),(10114276,08B),(10129814,08B),(10114173,08B),(1
0114661,08B),(10115061,08B),(10114507,08B),(10114306,08B),(10113959,08B),(10114052,08B),(10128843,08B),(10113914,08B),(101140
28,08B),(10114130,08B),(10113835,08B),(10113689,08B),(10129832,08B),(10113648,08B),(10113710,08B),(10115423,08B),(10113637,08
B),(10115633,08B),(10113624,08B),(10128900,08B),(10113647,08B),(10129106,08B),(10115390,08B),(10113654,08B),(10113607,08B),(1
0122215,08B),(10113599,08B),(10113668,08B),(9733506,08B),(10114261,08B),(10113636,08B),(10113585,08B),(10113600,08B),(1012895
1,08B),(10123902,08B),(10113691,08B)})
(1923,{(10181133,1923)})
(,{(9,)})
grunt>
```

```
grunt> describe count_fbicode;
count_fbicode: {group: chararray,long}
grunt> count_fbicode = foreach group_fbi generate group,COUNT(id_fbi_code.ID);
```

Acadgild Retail Batche... | acadgild@localhost:~ | acadgild@localhost:~

```
cess : 1
(1,172)
(2,362)
(3,266)
(4,154)
(5,197)
(6,198)
(7,138)
(8,301)
(9,192)
(02,1480)
(03,10552)
(05,14735)
(06,62826)
(07,10520)
(09,437)
(10,1708)
(11,13637)
(12,79)
(13,151)
(14,31244)
(15,3780)
(16,1949)
(17,1165)
(18,24989)
(19,590)
(20,1435)
(21,293)
(22,483)
```

Acadgild Retail Batche... | acadgild@localhost:~ | acadgild@localhost:~

```
(32,76)
(33,105)
(34,184)
(35,56)
(36,63)
(37,161)
(38,117)
(39,98)
(40,97)
(41,123)
(42,87)
(43,101)
(44,35)
(45,34)
(46,62)
(47,137)
(48,61)
(49,61)
(50,40)
(56,15)
(57,1)
(58,3)
(61,5)
(66,7)
(68,2)
(76,51)
(01A,533)
(01B,6)
(04A,4912)
(04B,7598)
(08A,13161)
(08B,44935)
(1923,1)
(,1)
grunt>
```

Acadgild Retail Batche... | acadgild@localhost:~ | acadgild@localhost:~

2. Write a MapReduce/Pig program to calculate the number of cases investigated under FBI code 32.

```
(9767782,32)
(9763207,32)
(9759854,32)
(9744293,32)
(9740583,32)
(9735627,32)
(9728843,32)
grunt> fil_fbi_code_32 = filter id_fbi_code by ($1 == '32');
```

Dumping data in

```
(9763207,32)
(9759854,32)
(9744293,32)
(9740583,32)
(9735627,32)
(9728843,32)
grunt> dump fil_fbi_code_32;
```

After dumping data:

```
(9984570,32)
(9934976,32)
(9923692,32)
(9927510,32)
(9934478,32)
(9909489,32)
(9895902,32)
(9893016,32)
(9874736,32)
(9867402,32)
(9867160,32)
(9865092,32)
(9864588,32)
(9864383,32)
(9880326,32)
(9857642,32)
(9862971,32)
(9841405,32)
(9833798,32)
(9825436,32)
(9829783,32)
(9811856,32)
(9809908,32)
(9809666,32)
(9917857,32)
(9791000,32)
(9772898,32)
(9773856,32)
(9767782,32)
(9763207,32)
(9759854,32)
(9744293,32)
(9740583,32)
(9735627,32)
(9728843,32)
grunt>
```

```
(9744293,32)
(9740583,32)
(9735627,32)
(9728843,32)
grunt> group_fbi_32 = group fil_fbi_code_32 ALL;
grunt> dump group_fbi_32;
```

For counting investigated FBI code under 32.

```
2017-12-24 19:18:05,472 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not g
enerate code.
2017-12-24 19:18:05,862 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-24 19:18:05,864 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(76)
grunt> count_fbi_32 = foreach group_fbi_32 generate COUNT(fil_fbi_code_32);
grunt> describe count_fbi_32;
count_fbi_32: {long}
grunt> dump count_fbi_32;
```

3. Write a MapReduce/Pig program to calculate the number of arrests in theft district wise.

Step - 1

```
grunt> describe load_crim;
load_crim: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR: chararray,PrimaryType: chararray,Description
: chararray,LoactionDesc: chararray,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward: chararray
,CommunityArea: chararray,FBICode: chararray,XCordinate: chararray,YCordinate: chararray,Year: chararray,UpdatedOn: chararray
,Latitude: chararray,Longitude: chararray,Location: chararray}
grunt>
```

Step – 2 : filter only THEFTS records.

```
grunt> fil_theft = filter load_crim by(chararray)$5 matches '.*THEFT.*';
grunt> describe fil_theft;
fil_theft: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR: chararray,PrimaryType: chararray,Description
: chararray,LoactionDesc: chararray,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward: chararray
,CommunityArea: chararray,FBICode: chararray,XCordinate: chararray,YCordinate: chararray,Year: chararray,UpdatedOn: chararray
,Latitude: chararray,Longitude: chararray,Location: chararray}
grunt>
```

Step – 3 : grouping with respect to district:

```
grunt> group_district = group fil_theft by $11;
grunt> describe group_district;
group_district: {group: chararray,fil_theft: {(ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR: chararray
,PrimaryType: chararray,Description: chararray,LoactionDesc: chararray,Arrest: chararray,Domestic: chararray,Beat: chararray,
District: chararray,Ward: chararray,CommunityArea: chararray,FBICode: chararray,XCordinate: chararray,YCordinate: chararray,Y
ear: chararray,UpdatedOn: chararray,Latitude: chararray,Longitude: chararray,Location: chararray)}}
grunt>
```

Step – 4 grouping each district wise:

```
grunt> each_district = foreach group_district generate group as DISTRICT, COUNT(fil_theft) as NUM_THEFTS;
grunt> describe each_disctrict;
2017-12-24 23:57:21,629 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1003: Unable to find an operator for alias each
_disctrict
Details at logfile: /home/acadgild/pig_1514116395689.log
grunt> describe each_district;
each_district: {DISTRICT: chararray,NUM_THEFTS: long}
grunt>
```

```
enerate code.
2017-12-24 23:21:29,080 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-24 23:21:29,087 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(001,6090)
(002,3110)
(003,2688)
(004,3399)
(005,2371)
(006,3816)
(007,2667)
(008,4750)
(009,3301)
(010,2337)
(011,2727)
(012,4583)
(014,3733)
(015,1911)
(016,2664)
(017,2650)
(018,5809)
(019,5016)
(020,1446)
(022,2367)
(024,2045)
(025,3865)
(031,1)
(true,8)
(false,2079)
```

4. Write a MapReduce/Pig program to calculate the number of arrests done between October 2014 and October 2015.

Filtering records in between 2014 and 2015;

```
load_crime: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR: chararray,PrimaryType: chararray,Descriptio
n: chararray,LocationDesc: chararray,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward: chararra
y,CommunityArea: chararray,FBICode: chararray,Xcordinate: chararray,Ycordinate: chararray,Year: chararray,UpdatedOn: chararra
y,Latitude: chararray,Longitude: chararray,Location: chararray}
grunt> col_date = foreach load_crime generate (chararray)$2 as DateCol,(chararray)$8 as Arrest;
grunt> fil_date_col = filter col_date by (DateCol is not null) and Arrest == 'TRUE';
grunt> date_sub = foreach fil_date_col generate ToDate(SUBSTRING(DateCol,0,19),'MM/dd/yyyy hh:mm:ss') as SubDt,Arrest;
grunt> date_wise_arrest = foreach date_sub generate GetMonth(SubDt) as Month,
>> GetYear(SubDt) as year,Arrest;
grunt> Total_Arrest = filter date_wise_arrest by (Month > 9 and year == 2014) or (Month<11 and year == 2015);
grunt> group_tot_arrest = group Total_Arrest ALL;
grunt> count_total_arrest = foreach group_tot_arrest generate COUNT(Total_Arrest.Arrest) as TotalArrest;
grunt>
```

Selecting columns:

```
grunt> describe load_crime;
load_crime: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR
: chararray,PrimaryType: chararray,Description: chararray,LocationDesc: chararra
y,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward
: chararray,CommunityArea: chararray,FBICode: chararray,Xcordinate: chararray,Yc
ordinate: chararray,Year: chararray,UpdatedOn: chararray,Latitude: chararray,Lon
gitude: chararray,Location: chararray}
grunt> describe load_crime;
load_crime: {ID: int,CaseNumber: chararray,Date: chararray,Block: chararray,IUSR: chararray,PrimaryType: chararray,Descriptio
n: chararray,LocationDesc: chararray,Arrest: chararray,Domestic: chararray,Beat: chararray,District: chararray,Ward: chararra
y,CommunityArea: chararray,FBICode: chararray,Xcordinate: chararray,Ycordinate: chararray,Year: chararray,UpdatedOn: chararra
y,Latitude: chararray,Longitude: chararray,Location: chararray}
grunt> col_date = foreach load_crime generate (chararray)$2 as DateCol,(chararray)$8 as Arrest;
grunt>
```

acadgild@localhost:~ | [acadgild@localhost:~... | Solved: org.apache.pi... | acadgild@localhost:~

Data dumped on col_date:

Applications Places System Mon Dec 25, 8:19 PM acadgil

acadgild@localhost:~

File Edit View Search Terminal Help

```
(08/03/2014 01:07:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,true)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 01:00:00 PM,false)
(08/03/2014 12:58:00 PM,true)
(08/03/2014 12:50:00 PM,false)
```

Filtering from above selected columns:

```
(08/03/2014 12:00:00 PM,false)
(08/03/2014 12:00:00 PM,false)
(08/03/2014 12:00:00 PM,false)
(,)
grunt> fil_date_col = FILTER col_date BY ((DateCol is not null) and (Arrest=='true'));
grunt> dump fil_date_col;
```

acadgild@localhost:~ | acadgild@localhost:~/... | Solved: org.apache.pi... | acadgild@localhost:~

After dumping :

```
(08/03/2014 12:40:00 PM,true)
(08/03/2014 12:40:00 PM,true)
(08/03/2014 12:35:00 PM,true)
(08/03/2014 12:35:00 PM,true)
(08/03/2014 12:30:00 PM,true)
(08/03/2014 12:27:00 PM,true)
(08/03/2014 12:26:00 PM,true)
(08/03/2014 12:19:00 PM,true)
(08/03/2014 12:18:00 PM,true)
(08/03/2014 12:10:00 PM,true)
(08/03/2014 12:07:00 PM,true)
grunt>
```

acadgild@localhost:~ | acadgild@localhost:~/... | Solved: org.apache.pi... | acadgild@localhost:~

```
(8,2014,true)
(8,2014,true)
(8,2014,true)
(8,2014,true)
grunt> Total_Arrest = filter date_wise_arrest by (Month > 9 and year == 2014) or (Month<11 and year == 2015);
grunt> dump Total_Arrest;
```

After dumping data:

```
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
grunt>
```

Grouping by using Arrest :

```
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
(10,2014,true)
grunt> group_tot_arrest = group Total_Arrest ALL;
grunt> count_total_arrest = foreach group_tot_arrest generate COUNT(Total_Arrest.Arrest) as TotalArrest;
```

After counting total arrest :

```
2017-12-25 21:25:01,123 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not g
enerate code.
2017-12-25 21:25:01,299 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-12-25 21:25:01,299 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to pro
cess : 1
(63173)
grunt>
```