USA Crime Analysis

The dataset contains attributes related to crimes taking place in various areas like type of crime, FBI code related to that criminal case, arrest frequency, location of crime etc.

Dataset Description:

ID,Case Number,Date,Block,IUCR,Primary Type,Description,Location

Description,Arrest,Domestic,Beat,District,Ward,Community Area,FBICode,X Coordinate,Y Coordinate,Year,Updated On,Latitude,Longitude,Location

Problem Statement

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

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

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

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

1)

usecase_1.pig (~/sayantan/Project1) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

usecase_1.pig

```
REGISTER '/home/cloudera/sayantan/Project1/piggybank.jar';
A = load '/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',');
B = FOREACH A GENERATE (Chararray)$0 AS id,(Chararray)$1 AS case_number,(Chararray)$13 AS FBI_code;
C = GROUP B BY FBI_code;
D = FOREACH C GENERATE GROUP,COUNT(B.FBI_code);
dump D;
```

Command: exec usecase_1.pig

```
2017-11-13 13:13:26,781 [main] INFO  org.apache.pig.backend.hadoop.executic
(0,2)
(1,3931)
(2,3472)
(3,4055)
(4,2001)
(5,1623)
(6,6064)
(7,4062)
(8,9664)
(9,287)
(10,1394)
(11,1289)
(12,495)
(13,894)
(14,2723)
(15,3763)
(16,3189)
(17,1794)
(18,625)
(19,5376)
(20,1870)
(21,2561)
(22,5304)
(23,9313)
(24,7513)
(25,19879)
(26,6403)
(27,5933)
(28,8808)
(29,9178)
(30,4852)
(31,2777)
```

```
(43,10229)
(44,6757)
(45,1600)
(46,5721)
(47,423)
(48,1671)
(49,7598)
(50,1247)
(51,2268)
(52,1520)
(53,4496)
(54,1381)
(55,588)
(56,2021)
(57,1104)
(58,3076)
(59,1179)
(60,1799)
(61,5507)
(62,1100)
(63,2656)
(64,1046)
(65,2285)
(66,6956)
(67,8208)
(68,7877)
(69,7295)
(70,2688)
(71,8454)
(72,1116)
(73,3475)
(74,672)
(75,2345)
```

2)

```
grunt> A = load '/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',')
>> ;
grunt> B = FOREACH A GENERATE (Chararray)$0 AS id,(Chararray)$1 AS case_number,(Chararray)$13 AS FBI_code;
```

```
grunt> C = filter B by FBI_code=='32';
grunt> D = group C by FBI_code;
grunt> E = foreach D generate group,COUNT(C.FBI_code);
grunt> dump E;
```

```
Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local437251389_0002 A,B,C,D,E      GROUP_BY,COMBINER      file:/tmp/temp-817356212/tmp300668053,

Input(s):
Successfully read records from: "/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv"

Output(s):
Successfully stored records in: "file:/tmp/temp-817356212/tmp300668053"

Job DAG:
job_local437251389_0002


2017-11-13 13:29:56,850 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-11-13 13:29:56,851 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-11-13 13:29:56,851 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-11-13 13:29:56,851 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-11-13 13:29:56,852 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-11-13 13:29:56,898 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-11-13 13:29:56,898 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(32,7987)
grunt>
```

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

```
grunt> A = load '/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',')
>> ;
grunt> B = FOREACH A GENERATE (Chararray)$0 AS id,(Chararray)$1 AS case_number,(Chararray)$8 AS arrest,(Chararray)$11 AS district;
grunt> C = filter B by arrest=='true';
grunt> D = group C by district;
grunt> E = foreach D generate group,COUNT(C.district);
grunt> dump E;
```

```
2017-11-13 18:15:59,390 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-11-13 18:15:59,391 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-11-13 18:15:59,391 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-11-13 18:15:59,391 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-11-13 18:15:59,392 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-11-13 18:15:59,558 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-11-13 18:15:59,558 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(001,3156)
(002,2060)
(003,3682)
(004,5086)
(005,3745)
(006,4820)
(007,5495)
(008,4667)
(009,3830)
(010,4094)
(011,9615)
(012,2448)
(014,1663)
(015,5458)
(016,2014)
(017,1419)
(018,2473)
(019,2471)
(020,1123)
(022,2124)
(024,1712)
(025,5175)
grunt>
```

4)

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

We have to first filter out the date according to the problem statement

```
grunt> A = load '/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',');
grunt> B = FOREACH A GENERATE (Chararray)$2 AS date,(chararray)$8 AS arrest;
grunt> S_Date = FOREACH B Generate (date), SUBSTRING(date,6,10) AS Year,arrest;
grunt> dump;
```

```
(03/04/2015 12:50:00 AM,2015,true)
(03/04/2015 12:49:00 AM,2015,false)
(03/04/2015 12:45:00 AM,2015,false)
(03/04/2015 12:40:00 AM,2015,false)
(03/04/2015 12:40:00 AM,2015,true)
(03/04/2015 12:30:00 AM,2015,true)
(03/04/2015 12:20:00 AM,2015,false)
(03/04/2015 12:19:00 AM,2015,false)
(03/04/2015 12:15:00 AM,2015,true)
(03/04/2015 12:15:00 AM,2015,false)
(03/04/2015 12:15:00 AM,2015,true)
(03/04/2015 12:10:00 AM,2015,false)
(03/04/2015 12:07:00 AM,2015,true)
(03/04/2015 12:05:00 AM,2015,false)
(03/04/2015 12:03:00 AM,2015,true)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:01:00 AM,2015,false)
(03/04/2015 12:00:00 AM,2015,false)
(03/04/2015 12:00:00 AM,2015,false)
(03/04/2015 12:00:00 AM,2015,false)
(03/04/2015 12:00:00 AM,2015,false)
```

```
grunt> S_Date = FOREACH B GENERATE (date
grunt> S_Fil = GROUP S_Date by Year;
grunt> dump;
```

```
se),(09/10/2015 09:50:00 PM,2015,true),(09/10/2015 09:51:00 PM,2015,false),(09/10/2015 09:54:00 PM,2015,true),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 P
M,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/201
5 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false
),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 PM,2015,false),(09/10/2015 10:00:00 P
M,2015,false),(09/10/2015 10:02:00 PM,2015,true),(09/10/2015 10:02:00 PM,2015,true),(09/10/2015 10:08:00 PM,2015,false),(09/10/2015 10:10:00 PM,2015,true),(09/10/2015 1
0:10:00 PM,2015,false),(09/10/2015 10:19:00 PM,2015,true),(09/10/2015 10:20:00 PM,2015,true),(09/10/2015 10:20:00 PM,2015,true),(09/10/2015 10:23:00 PM,2015,true),(09/1
0/2015 10:25:00 PM,2015,false),(09/10/2015 10:28:00 PM,2015,true),(09/10/2015 10:30:00 PM,2015,false),(09/10/2015 10:30:00 PM,2015,false),(09/10/2015 10:30:00 PM,2015,f
alse),(09/10/2015 10:32:00 PM,2015,true),(09/10/2015 10:35:00 PM,2015,false),(09/10/2015 10:36:00 PM,2015,false),(09/10/2015 10:40:00 PM,2015,true),(09/10/2015 10:40:00
 PM,2015,false),(09/10/2015 10:42:00 PM,2015,true),(09/10/2015 10:45:00 PM,2015,false),(09/10/2015 10:45:00 PM,2015,true),(09/10/2015 10:45:00 PM,2015,true),(09/10/2015
 10:48:00 PM,2015,true),(09/10/2015 10:50:00 PM,2015,false),(09/10/2015 10:50:00 PM,2015,false),(09/10/2015 10:54:00 PM,2015,true),(09/10/2015 10:56:00 PM,2015,true),(0
9/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,2015,true),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,201
5,false),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:00:00 PM,2015,false),(09/10/2015 11:
00:00 PM,2015,false),(09/10/2015 11:03:00 PM,2015,true),(09/10/2015 11:05:00 PM,2015,true),(09/10/2015 11:10:00 PM,2015,false),(09/10/2015 11:20:00 PM,2015,false),(09/1
0/2015 11:23:00 PM,2015,false),(09/10/2015 11:23:00 PM,2015,false),(09/10/2015 11:30:00 PM,2015,false),(09/10/2015 11:30:00 PM,2015,false),(09/10/2015 11:30:00 PM,2015,
false),(09/10/2015 11:30:00 PM,2015,false),(09/10/2015 11:30:00 PM,2015,false),(09/10/2015 11:35:00 PM,2015,false),(09/10/2015 11:45:00 PM,2015,true),(09/10/2015 11:50:
00 PM,2015,false),(09/10/2015 11:55:00 PM,2015,true),(09/10/2015 11:56:00 PM,2015,true)})
(,{(,,)})
grunt>
```

Now to filter out the final result of arrests between oct 2014 and oct 2014

```
grunt> final_result_date = Filter S_Date by (Year=='2014' or Year=='2015');
grunt> final_result_arrest = Filter final_result_date by (arrest=='true');
grunt> dump final_result_arrest;
```

```
(03/04/2015 10:27:00 AM,2015,true)
(03/04/2015 10:10:00 AM,2015,true)
(03/04/2015 09:50:00 AM,2015,true)
(03/04/2015 09:49:00 AM,2015,true)
(03/04/2015 09:45:00 AM,2015,true)
(03/04/2015 09:26:00 AM,2015,true)
(03/04/2015 09:13:00 AM,2015,true)
(03/04/2015 09:04:00 AM,2015,true)
(03/04/2015 09:00:00 AM,2015,true)
(03/04/2015 08:59:00 AM,2015,true)
(03/04/2015 08:30:00 AM,2015,true)
(03/04/2015 08:30:00 AM,2015,true)
(03/04/2015 08:30:00 AM,2015,true)
(03/04/2015 08:15:00 AM,2015,true)
(03/04/2015 08:10:00 AM,2015,true)
(03/04/2015 04:30:00 AM,2015,true)
(03/04/2015 03:50:00 AM,2015,true)
(03/04/2015 02:40:00 AM,2015,true)
(03/04/2015 02:30:00 AM,2015,true)
(03/04/2015 02:30:00 AM,2015,true)
(03/04/2015 02:23:00 AM,2015,true)
(03/04/2015 02:01:00 AM,2015,true)
(03/04/2015 01:30:00 AM,2015,true)
(03/04/2015 12:51:00 AM,2015,true)
(03/04/2015 12:50:00 AM,2015,true)
(03/04/2015 12:40:00 AM,2015,true)
(03/04/2015 12:30:00 AM,2015,true)
(03/04/2015 12:15:00 AM,2015,true)
(03/04/2015 12:15:00 AM,2015,true)
(03/04/2015 12:07:00 AM,2015,true)
(03/04/2015 12:03:00 AM,2015,true)
grunt>
```

Now to find the count of arrests done between those dates

```
(03/04/2015 12:15:00 AM,2015,true)
(03/04/2015 12:07:00 AM,2015,true)
(03/04/2015 12:03:00 AM,2015,true)
grunt> final_sorted_result = group final_result_arrest by Year;
grunt> final_result_count = foreach final_sorted_result generate group,COUNT(final_result_arrest.Year);
grunt> dump final_result_count;
```

```
Success!

Job Stats (time in seconds):
JobId   Alias   Feature Outputs
job_local1399738137_0011        A,B,S_Date,final_result_arrest,final_result_count,final_result_date,final_sorted_result GROUP_BY,COMBINER       file:/tmp/temp-81735
/tmp2124349174,

Input(s):
Successfully read records from: "/home/cloudera/sayantan/Project1/Crimes_-_2001_to_present.csv"

Output(s):
Successfully stored records in: "file:/tmp/temp-817356212/tmp2124349174"

Job DAG:
job_local1399738137_0011


2017-11-13 21:03:23,611 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2017-11-13 21:03:23,611 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2017-11-13 21:03:23,611 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2017-11-13 21:03:23,612 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2017-11-13 21:03:23,612 [main] WARN  org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been initialized
2017-11-13 21:03:23,729 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2017-11-13 21:03:23,729 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(2014,31155)
(2015,47175)
grunt>
```