Dataset Details

Dataset - link

- **ATM-Dataset**

  This dataset contains 2.5 Million transactions (2017-2019) for 105 ATMs in Denmark (Spar Nord Bank)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	year	month	day	weekday	hour	atm_statu	atm_id	atm_manufacturer	atm_location	atm_streetname	atm_street_numb	atm_zipco	atm_lat	atm_lon	currency	card_type	service	message_	message_text
	2017	January	1	Sunday	0	Active	1	NCR	NÃ¦stved	Farimagsvej	8	4700	55.23342	11.76327	DKK	MasterCard	Withdrawal		
	2017	January	1	Sunday	0	Inactive	2	NCR	Vejgaard	Hadsundvej	20	9000	57.04284	9.950013	DKK	MasterCard	Withdrawal		
	2017	January	1	Sunday	0	Inactive	2	NCR	Vejgaard	Hadsundvej	20	9000	57.04284	9.950013	DKK	VISA	Withdrawal		
	2017	January	1	Sunday	0	Inactive	3	NCR	Ikast	RÃ¥dhusstrÃ¦det	12	7430	56.13858	9.153843	DKK	VISA	Withdrawal		
	2017	January	1	Sunday	0	Active	4	NCR	Svogerslev	BrÃ¸nsager	1	4000	55.63356	12.01786	DKK	MasterCard	Withdrawal		
	2017	January	1	Sunday	0	Active	5	NCR	Nibe	Torvet	1	9240	56.98254	9.638719	DKK	MasterCard	Withdrawal		
	2017	January	1	Sunday	0	Active	6	NCR	Fredericia	SjÃ¦llandsgade	33	7000	55.56437	9.757486	DKK	MasterCard	Withdrawal		
	2017	January	1	Sunday	0	Active	7	Diebold Nixdorf	Hjallerup	Hjallerup Centret	18	9320	57.16761	10.14766	DKK	Mastercard - on-	Withdrawal		
0	2017	January	1	Sunday	0	Active	8	NCR	GlyngÃ¸re	FÃ¦rgevej	1	7870	56.76233	8.866716	DKK	MasterCard	Withdrawal		
1	2017	January	1	Sunday	0	Active	9	Diebold Nixdorf	Hadsund	Storegade	12	9560	56.71571	10.11378	DKK	VISA	Withdrawal		
2	2017	January	1	Sunday	0	Active	10	NCR	NÃ¸rresundby	Torvet	6	9400	57.05868	9.922473	DKK	Dankort	Withdrawal		
3	2017	January	1	Sunday	0	Active	11	NCR	Sauersvej	Fridtjof Nansens V	2	9210	57.02324	9.940357	DKK	Visa Dankort	Withdrawal		
4	2017	January	1	Sunday	0	Inactive	2	NCR	Vejgaard	Hadsundvej	20	9000	57.04284	9.950013	DKK	Mastercard - on-	Withdrawal		
5	2017	January	1	Sunday	0	Inactive	12	NCR	Ã˜sterÃ¥ Duus	Ã˜sterÃ¥	12	9000	57.04852	9.92193	DKK	Mastercard - on-	Withdrawal		
6	2017	January	1	Sunday	0	Active	13	NCR	SÃ¦by	Vestergade	3	9300	57.33425	10.51505	DKK	Mastercard - on-	Withdrawal		
7	2017	January	1	Sunday	0	Inactive	14	NCR	HÃ¸rning	NÃ¸rrealle	12	8362	56.08609	10.03735	DKK	Visa Dankort - or	Withdrawal		
8	2017	January	1	Sunday	0	Active	15	NCR	Vestre	Kastetvej	36	9000	57.05345	9.905292	DKK	MasterCard	Withdrawal		
9	2017	January	1	Sunday	0	Inactive	12	NCR	Ã˜sterÃ¥ Duus	Ã˜sterÃ¥	12	9000	57.04852	9.92193	DKK	Mastercard - on-	Withdrawal		
0	2017	January	1	Sunday	0	Inactive	16	NCR	Skive	Adelgade	8	7800	56.56678	9.026604	DKK	Visa Dankort	Withdrawal		
1	2017	January	1	Sunday	0	Active	17	NCR	Randers	Ã˜stervold	16	8900	56.46188	10.03779	DKK	VISA	Withdrawal		

- **Weather-Dataset**

  This dataset contains the weather report data.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
zipcode	weather_lat	weather_lon	weather_city_id	weather_city_name	temp	pressure	humidity	wind_spee	wind_deg	rain_3h	clouds_all	weather_i	weather_r	weather_d
4700	55.229919	11.76092	2616038	Naestved	281.15	1014	87	7	260	0.215	92	500	Rain	light rain
9000	57.04834	9.9352	2616235	NÃ¸rresundby	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9000	57.04834	9.9352	2616235	NÃ¸rresundby	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
7430	56.138828	9.15768	2619426	Ikast	281.15	1011	100	6	240		75	300	Drizzle	light intensi
4000	55.641521	12.08035	2614481	Roskilde	280.61	1014	87	7	260		88	701	Mist	mist
9240	56.981499	9.63917	2616483	Nibe	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
7000	55.565681	9.75257	2621951	Fredericia	281.15	1014	93	7	230	0.29	92	500	Rain	light rain
9320	57.164661	10.14571	2620275	Hjallerup	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
7870	56.793339	8.85282	2615964	Nykobing Mors	281.15	1011	100	6	240		75	300	Drizzle	light intensi
9560	56.714821	10.11682	2620952	Hadsund	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9400	57.048	9.9187	2624886	Aalborg	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9210	57.04834	9.9352	2616235	NÃ¸rresundby	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9000	57.04834	9.9352	2616235	NÃ¸rresundby	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9000	57.048	9.9187	2624886	Aalborg	280.64	1020	93	9	250	0.59	92	500	Rain	light rain
9300	57.440731	10.53661	2621927	Frederikshavn	281.14	1019	94	12	251	1.275	92	500	Rain	light rain

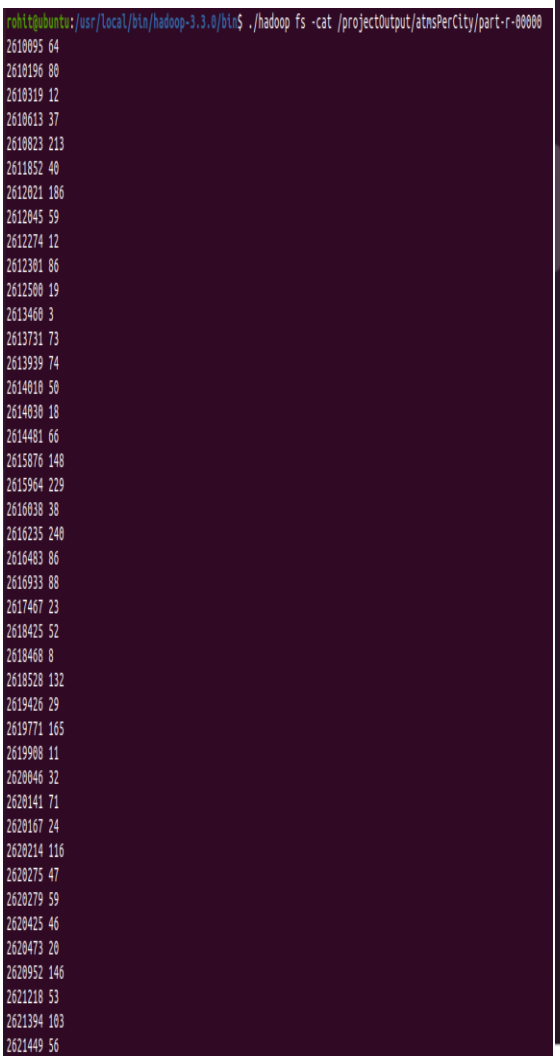
MapReduce and Pig Scripts Analysis performed-

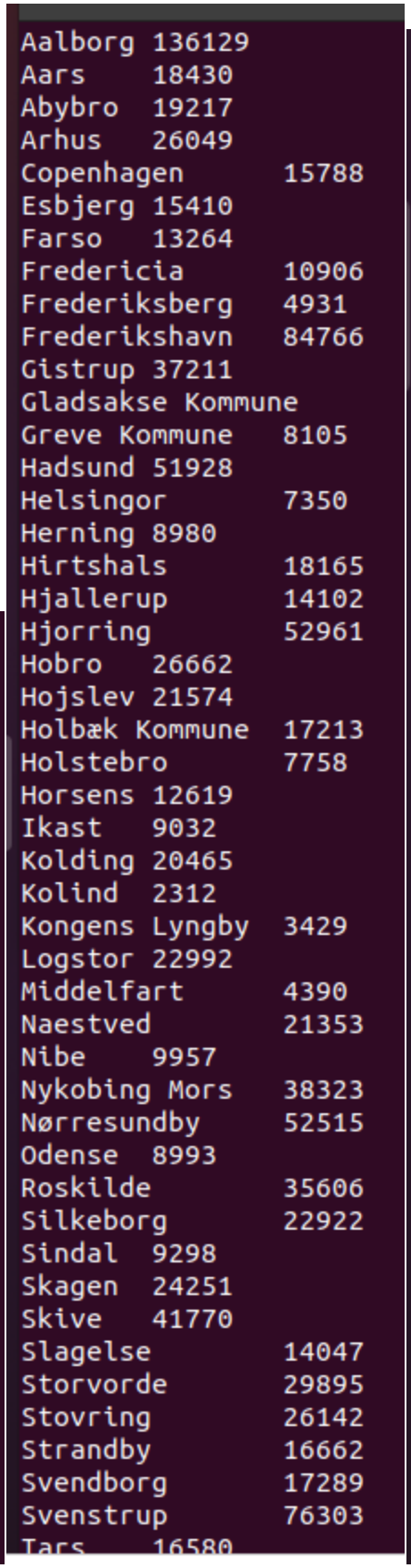
- Number of Transactions Per City
- Top Ten Cities with most ATM traffic
- Transactions with Active/Inactive status per city
- Join weather-dataset with atm-dataset
- Get specified ATM's error/malfunctioned transactions
- Monthly Trend, Weekly Trend
- Hourly Traffic

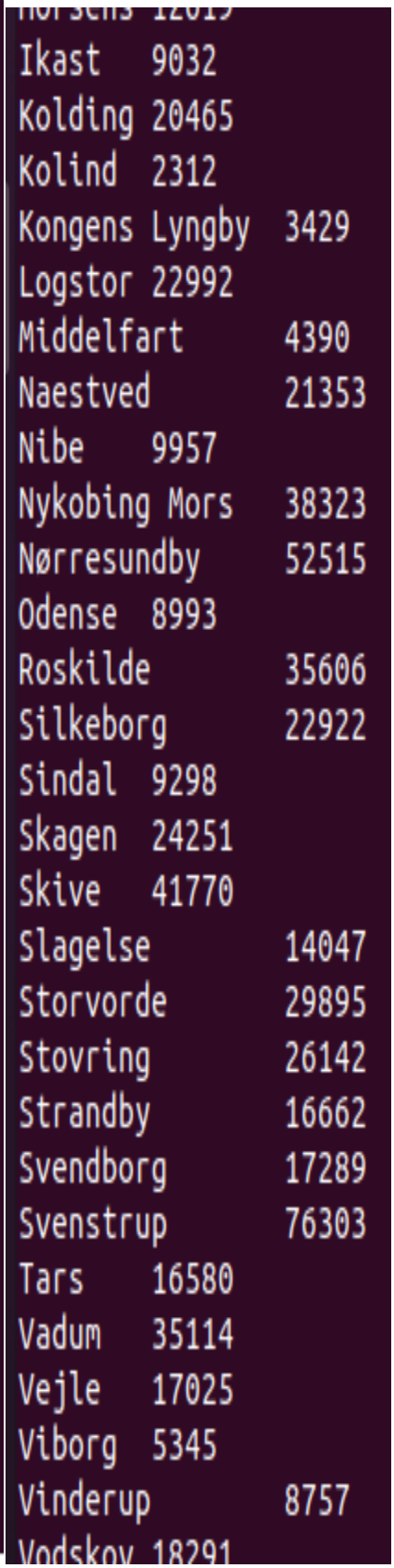
Number of Transactions across all cities

Analysis takes the count of the number of ATMs per city.

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/atmsPerCity.jar atmsPerCity.DriverClass /Project/atm_data_part1.csv /projectOutput/atmsPerCity
2021-04-23 08:13:35,015 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-23 08:13:36,275 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-23 08:13:36,347 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619186614317_0002
2021-04-23 08:13:37,062 INFO input.FileInputFormat: Total input files to process : 1
2021-04-23 08:13:37,298 INFO mapreduce.JobSubmitter: number of splits:2
2021-04-23 08:13:37,889 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619186614317_0002
2021-04-23 08:13:37,890 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-23 08:13:38,674 INFO conf.Configuration: resource-types.xml not found
2021-04-23 08:13:38,674 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-23 08:13:39,078 INFO impl.YarnClientImpl: Submitted application application_1619186614317_0002
2021-04-23 08:13:39,207 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619186614317_0002/
2021-04-23 08:13:39,209 INFO mapreduce.Job: Running job: job_1619186614317_0002
2021-04-23 08:13:55,954 INFO mapreduce.Job: Job job_1619186614317_0002 running in uber mode : false
2021-04-23 08:13:55,956 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-23 08:14:39,291 INFO mapreduce.Job:  map 9% reduce 0%
2021-04-23 08:14:47,620 INFO mapreduce.Job:  map 14% reduce 0%
2021-04-23 08:20:03,131 INFO mapreduce.Job:  map 17% reduce 0%
2021-04-23 08:20:08,204 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-23 08:20:14,241 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-23 08:20:15,262 INFO mapreduce.Job: Job job_1619186614317_0002 completed successfully
```







Observation --> The analysis gives transaction count for each city. The output gives us a good insight on how the ATM traffic is based on each city.

```java
package atmsPerCity;

import org.apache.hadoop.io.IntWritable;

public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws I

        String line = value.toString();
        String cityName = "";


        try {
            cityName = line.split(",")[22];
        } catch (Exception e) {
            cityName = "Ambiguous data";

        }



        if(cityName.length() > 0) {
            context.write(new Text(cityName),one);
        }
    }

}
```

```java
public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    // just like in mongoDB values is iterable
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context)

        int sum=0;
        for(IntWritable v: values){
            sum += v.get();
        }

        context.write(key, new IntWritable(sum));

    }
}
```

Top Ten Cities with most traffic

To get the top ten cities with most ATMs, I have used the output of previous analysis and calculated the top ten cities.

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/topTenAtmsCity.jar topTenAtmsCities.DriverClass /Project/atm_data_part1.csv /projectOutput/AtmsAllCity /projectOut
put/AtmsAllCity/part-r-00000 /projectOutput/TopTenAtmsPerCity
2021-04-29 09:59:11,650 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-29 09:59:12,723 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-29 09:59:12,758 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619713408179_0008
2021-04-29 09:59:13,681 INFO input.FileInputFormat: Total input files to process : 1
2021-04-29 09:59:14,039 INFO mapreduce.JobSubmitter: number of splits:2
2021-04-29 09:59:14,685 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619713408179_0008
2021-04-29 09:59:14,686 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-29 09:59:15,638 INFO conf.Configuration: resource-types.xml not found
2021-04-29 09:59:15,639 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-29 09:59:15,959 INFO impl.YarnClientImpl: Submitted application application_1619713408179_0008
2021-04-29 09:59:16,148 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619713408179_0008/
2021-04-29 09:59:16,158 INFO mapreduce.Job: Running job: job_1619713408179_0008
2021-04-29 09:59:37,247 INFO mapreduce.Job: Job job_1619713408179_0008 running in uber mode : false
2021-04-29 09:59:37,249 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-29 10:00:00,782 INFO mapreduce.Job:  map 67% reduce 0%
2021-04-29 10:00:01,932 INFO mapreduce.Job:  map 83% reduce 0%
2021-04-29 10:00:02,950 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-29 10:00:19,213 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-29 10:00:21,262 INFO mapreduce.Job: Job job_1619713408179_0008 completed successfully
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -head /projectOutput/TopTenAtmsPerCity/part-r-00000
Aalborg 136129
Frederikshavn   84766
Gistrup 37211
Hadsund 51928
Hjorring        52961
Nykobing Mors   38323
Nørresundby     52515
Roskilde        35606
Skive   41770
Svenstrup       76303
```

Observation → Here we have top ten cities with maximum number of transactions, the bank can focus on these cities to increase more ATMs, or open up new branches.

Driver Class

```
public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException{


    Configuration conf = new Configuration();
    Configuration conf2 = new Configuration();
    // Create a new Job
    Job job = Job.getInstance(conf,"toptentransactionsPerCity");
    job.setJarByClass(DriverClass.class);

    // Specify various job-specific parameters
    job.setJobName("toptentransactionsPerCity");


    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));


    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(IntWritable.class);

    job.setMapperClass(MapperClass.class);
    job.setReducerClass(ReducerClass.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    boolean complete = job.waitForCompletion(true);

    Job job2 = Job.getInstance(conf2, "Chaining");

    if (complete) {
        FileInputFormat.addInputPath(job2, new Path(args[2]));
        FileOutputFormat.setOutputPath(job2, new Path(args[3]));
        job2.setJarByClass(DriverClass.class);
        job2.setMapperClass(MapperClass2.class);
        job2.setMapOutputKeyClass(Text.class);
        job2.setMapOutputValueClass(LongWritable.class);
        System.exit(job2.waitForCompletion(true) ? 0 : 1);

    }
}
```

MapperClass2

```
    @Override
    protected void setup(Mapper<LongWritable, Text, Text, LongWritable>.Context context)
            throws IOException, InterruptedException {
        // TODO Auto-generated method stub
    //    super.setup(context);
    }



    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException, InterruptedException {

        String row[] = value.toString().split("\\t");
//        Text city_name = new Text(row[0]);

        String city_name = row[0].toString();
        Integer atmsCount = Integer.parseInt(row[1]);

        tm.put(atmsCount, city_name);

        if(tm.size() > 10)
            tm.remove(tm.firstKey());
    }


    @Override
    protected void cleanup(Mapper<LongWritable, Text, Text, LongWritable>.Context context)
            throws IOException, InterruptedException {


        for (Map.Entry<Integer, String> entry : tm.entrySet())
        {

            long count = entry.getKey();
            String name = entry.getValue();

            context.write(new Text(name), new LongWritable(count));
        }
```

How many Active/In-Active ATMs?

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/activeInactiveAtmsAcrossCities.jar  atmStatusAcrossCities.DriverClass /Project/atm_data_part1.csv /projectOutput/A
ctiveInactiveAtmscities
2021-04-29 11:26:11,678 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-29 11:26:12,025 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-29 11:26:12,040 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619719060148_0002
2021-04-29 11:26:12,260 INFO input.FileInputFormat: Total input files to process : 2
2021-04-29 11:26:12,347 INFO mapreduce.JobSubmitter: number of splits:4
2021-04-29 11:26:12,469 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619719060148_0002
2021-04-29 11:26:12,470 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-29 11:26:12,662 INFO conf.Configuration: resource-types.xml not found
2021-04-29 11:26:12,662 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-29 11:26:12,766 INFO impl.YarnClientImpl: Submitted application application_1619719060148_0002
2021-04-29 11:26:12,799 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619719060148_0002/
2021-04-29 11:26:12,800 INFO mapreduce.Job: Running job: job_1619719060148_0002
2021-04-29 11:26:18,912 INFO mapreduce.Job: Job job_1619719060148_0002 running in uber mode : false
2021-04-29 11:26:18,913 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-29 11:26:38,125 INFO mapreduce.Job:  map 75% reduce 0%
2021-04-29 11:26:39,131 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-29 11:26:46,166 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-29 11:26:47,180 INFO mapreduce.Job: Job job_1619719060148_0002 completed successfully
```

```
Gladsakse Kommune                4655    2718
Greve Kommune           8105    0
Hadsund         51928   0
Helsingor               7350    0
Herning         8980    0
Hirtshals               18165   0
Hjallerup               12792   1310
Hjorring                50432   2529
Hobro           26662   0
Hojslev         21574   0
Holbæk Kommune          14655   2558
Holstebro               7758    0
Horsens         12619   0
Ikast           0       9032
Kolding         20465   0
Kolind          2312    0
Kongens Lyngby          3429    0
Logstor         22992   0
Middelfart              4390    0
Naestved                21353   0
Nibe            9957    0
Nykobing Mors           15129   23194
Nørresundby             29528   22987
Odense          8425    568
Roskilde                35606   0
Silkeborg               22922   0
Sindal          9298    0
Skagen          19840   4411
Skive           12689   29081
Slagelse                14047   0
Storvorde               29895   0
Stovring                26142   0
Strandby                16662   0
Svendborg               17289   0
Svenstrup               42246   34057
Tars            16580   0
Vadum           30646   4468
Vejle           17025   0
Viborg          5345    0
Vinderup                8757    0
Vodskov         18291   0
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$
```

Observations → This analysis gives us the ATM status for the transactions. How many times it remained inactive, and active. The results would give insight on the ATM status activity.

MapperClass

```
public class MapperClass extends Mapper<Object, Text, Text, ActiveInactiveTuple> {

    private ActiveInactiveTuple outputTuple = new ActiveInactiveTuple();

    @Override
    protected void map(Object key, Text value, Context context) throws IOException, Int

        String[] atmLine= value.toString().split(",");
        if(!atmLine[0].equals("weather_city_name")){
            String city = atmLine[22];
            String status = atmLine[5];
            try {
                if(status.equals("Active")) {
                    outputTuple.setActive(1);
                    outputTuple.setInactive(0);
                    context.write(new Text(city), outputTuple);
                }
                else if(status.equals("Inactive")) {
                    outputTuple.setActive(0);
                    outputTuple.setInactive(1);
                    context.write(new Text(city), outputTuple);
                }
             //   context.write(new Text(city), outputTuple);

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
```

ReducerClass

```
public class
ReducerClass extends Reducer<Text, ActiveInactiveTuple, Text, ActiveInactiveTuple>{

     private ActiveInactiveTuple outputTuple = new ActiveInactiveTuple();

    @Override
    protected void reduce(Text key, Iterable<ActiveInactiveTuple> values, Context cont

        // First Count
        long active = 0;
        long inactive = 0;

        for(ActiveInactiveTuple v: values){
            active += v.getActive();
            inactive += v.getInactive();
        }
        active = active/2;
        inactive = inactive/2;

        outputTuple.setActive(active);
        outputTuple.setInactive(inactive);

        // Emit it
        context.write(key, outputTuple);
    }


}
```

Join with weather data

For a given ATM-ID, get the weather report across all transactions

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/ReduceJoin.jar reduceSideJoin.DriverClass 3 /Project/atm-dataset /Project/weather-dataset /projectOutput/JoinOutpu
t
2021-04-29 19:30:15,860 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-29 19:30:16,215 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-29 19:30:16,254 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619749796274_0001
2021-04-29 19:30:16,963 INFO input.FileInputFormat: Total input files to process : 1
2021-04-29 19:30:16,995 INFO input.FileInputFormat: Total input files to process : 1
2021-04-29 19:30:17,474 INFO mapreduce.JobSubmitter: number of splits:3
2021-04-29 19:30:17,603 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619749796274_0001
2021-04-29 19:30:17,604 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-29 19:30:17,783 INFO conf.Configuration: resource-types.xml not found
2021-04-29 19:30:17,783 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-29 19:30:18,074 INFO impl.YarnClientImpl: Submitted application application_1619749796274_0001
2021-04-29 19:30:18,102 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619749796274_0001/
2021-04-29 19:30:18,103 INFO mapreduce.Job: Running job: job_1619749796274_0001
2021-04-29 19:30:25,254 INFO mapreduce.Job: Job job_1619749796274_0001 running in uber mode : false
2021-04-29 19:30:25,254 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-29 19:30:46,006 INFO mapreduce.Job:  map 17% reduce 0%
2021-04-29 19:30:49,190 INFO mapreduce.Job:  map 38% reduce 0%
2021-04-29 19:30:52,212 INFO mapreduce.Job:  map 46% reduce 0%
2021-04-29 19:30:58,261 INFO mapreduce.Job:  map 57% reduce 0%
2021-04-29 19:31:09,696 INFO mapreduce.Job:  map 64% reduce 0%
2021-04-29 19:31:10,750 INFO mapreduce.Job:  map 71% reduce 0%
2021-04-29 19:31:30,864 INFO mapreduce.Job:  map 84% reduce 0%
2021-04-29 19:31:31,876 INFO mapreduce.Job:  map 88% reduce 11%
2021-04-29 19:31:35,963 INFO mapreduce.Job:  map 100% reduce 11%
2021-04-29 19:31:37,974 INFO mapreduce.Job:  map 100% reduce 67%
2021-04-29 19:31:39,990 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-29 19:31:42,021 INFO mapreduce.Job: Job job_1619749796274_0001 completed successfully
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -head /projectOutput/LeftReduceJoinOutput/part-r-00000
Ikast,Clouds    3,2017,April,0,7430,MasterCard,DKK
Ikast,Clouds    3,2017,April,0,7430,Visa Dankort,DKK
Ikast,Clouds    3,2017,April,0,7430,MasterCard,DKK
Ikast,Clouds    3,2017,April,0,7430,Visa Dankort,DKK
Ikast,Clear     3,2017,April,1,7430,Mastercard - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,VISA,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort,DKK
Ikast,Clear     3,2017,April,1,7430,Mastercard - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,VISA,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Clear     3,2017,April,1,7430,Visa Dankort,DKK
Ikast,Rain      3,2017,April,1,7430,Mastercard - on-us,DKK
Ikast,Rain      3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Rain      3,2017,April,1,7430,VISA,DKK
Ikast,Rain      3,2017,April,1,7430,Visa Dankort - on-us,DKK
Ikast,Rain      3,2017,April,1,7430,Visa Dankort,DKK
Ikast,Rain      3,2017,April,1,7430,Mastercard - on-us,DKK
```

Observations → This analysis gives the weather report at the time of each transactions for the given ATM-ID. The code picks up 4 arguments, 1st argument takes the ATM_ID, 2nd argument is the atm dataset input file path, 3rd argument takes the weather dataset input file path and 4th argument is the output file path.

AtmMapperClass

```
    @Override
    protected void setup(Mapper<LongWritable, Text, Text, Text>.Context context)
            throws IOException, InterruptedException {

        getATM = context.getConfiguration().get("atm");

//      getATM = "1";
    }
    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException, Inte

        String line = value.toString();
        String atm = "";

        try {
              atm = line.split(",")[6];
              String zipcode = line.split(",")[11];

              String year = line.split(",")[0];
              String month = line.split(",")[1];
              String currency = line.split(",")[14];
              String hour = line.split(",")[4];
              String card_type = line.split(",")[15];

              Text val = new Text(year+","+month+","+hour+","+zipcode+"\t"+card_type+","+currency);
              Text newKey = new Text(year+","+month+","+hour+","+zipcode);

              if(atm.equals(getATM)) {
                    context.write(newKey, new Text("A"+atm+" "+val));
              }


        } catch (Exception e) {
            atm = "Ambiguous data";

        }
```

WeatherMapperClass

```
@Override
protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException, Int

    String line = value.toString();
    String atm = "";

    try {

        String year = line.split(",")[0];
        String month = line.split(",")[1];
        String hour = line.split(",")[4];
        String zipcode = line.split(",")[5];

        String cityName = line.split(",")[9];
        String weather_main = line.split(",")[18];

        Text val = new Text("W"+cityName+","+weather_main);

        context.write(new Text(year+","+month+","+hour+","+zipcode), new Text(val));


    } catch (Exception e) {
        atm = "Ambiguous data";

    }
```

ReducerClass

```
@Override
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
    listATM.clear();
    listW.clear();
    for(Text text : values){
        if(text.charAt(0) == 'A'){
            listATM.add(new Text(text.toString().substring(1)));
        } else if(text.charAt(0) == 'W'){
            listW.add(new Text(text.toString().substring(1)));
        }
    }
    executeJoinLogic(context);
}



public void executeJoinLogic(Context context) throws IOException, InterruptedException {
    String joinType = context.getConfiguration().get("join.type");

    //Inner Join
    if(joinType.equalsIgnoreCase("inner")){
        if(!listATM.isEmpty() && !listW.isEmpty()){
            for(Text atmData : listATM){
                for(Text wData : listW){
                    context.write(atmData, wData);
                }
            }
        }
```

Get specified ATM errors/issues data

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/filterATM.jar filteringAtms.DriverClass 2 /Project/atm-dataset /projectOutput/filteredAtms2
2021-04-29 16:32:09,590 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-29 16:32:09,907 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-29 16:32:09,919 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619738209556_0006
2021-04-29 16:32:10,105 INFO input.FileInputFormat: Total input files to process : 1
2021-04-29 16:32:10,569 INFO mapreduce.JobSubmitter: number of splits:2
2021-04-29 16:32:10,676 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619738209556_0006
2021-04-29 16:32:10,676 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-29 16:32:10,824 INFO conf.Configuration: resource-types.xml not found
2021-04-29 16:32:10,824 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-29 16:32:10,878 INFO impl.YarnClientImpl: Submitted application application_1619738209556_0006
2021-04-29 16:32:10,912 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619738209556_0006/
2021-04-29 16:32:10,913 INFO mapreduce.Job: Running job: job_1619738209556_0006
2021-04-29 16:32:15,983 INFO mapreduce.Job: Job job_1619738209556_0006 running in uber mode : false
2021-04-29 16:32:15,984 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-29 16:32:25,086 INFO mapreduce.Job:  map 50% reduce 0%
2021-04-29 16:32:31,155 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-29 16:32:33,169 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-29 16:32:33,178 INFO mapreduce.Job: Job job_1619738209556_0006 completed successfully
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -head /projectOutput/filteredAtms2/part-r-00000
2       July,8,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,6,9000,VISA,,DKK,"Suspected malfunction
2       June,5,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,5,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,15,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,15,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,15,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,7,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,6,9000,Hævekort - on-us,,DKK,"Suspected malfunction
2       June,6,9000,Hævekort - on-us,,DKK,"Suspected malfunction
2       June,21,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,21,9000,MasterCard,,DKK,"Suspected malfunction
2       June,21,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,11,9000,Mastercard - on-us,,DKK,"Suspected malfunction
2       June,11,9000,MasterCard,,DKK,"Suspected malfunction
2       June,11,9000,MasterCard,,DKK,"Suspected malfunction
2       June,10,9000,Visa Dankort - on-us,,DKK,"Suspected malfunction
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -tail /projectOutput/filteredAtms3/part-r-00000
sa Dankort,,DKK,"Suspected malfunction
3       May,12,7430,Visa Dankort,,DKK,"Suspected malfunction
3       May,7,7430,Hævekort - on-us,,DKK,"Suspected malfunction
3       May,7,7430,Visa Dankort,,DKK,"Suspected malfunction
3       May,7,7430,Visa Dankort,,DKK,"Suspected malfunction
3       May,6,7430,MasterCard,,DKK,"Suspected malfunction
3       May,8,7430,Mastercard - on-us,,DKK,"Suspected malfunction
3       May,8,7430,Visa Dankort - on-us,,DKK,"Suspected malfunction
3       May,8,7430,Visa Dankort - on-us,,DKK,"Suspected malfunction
3       May,8,7430,MasterCard,,DKK,"Suspected malfunction
3       May,21,7430,MasterCard,,DKK,"Suspected malfunction
3       May,14,7430,Dankort - on-us,,DKK,"Suspected malfunction
3       April,12,7430,Visa Dankort,,DKK,"Suspected malfunction
3       April,12,7430,Visa Dankort,,DKK,"Suspected malfunction
3       April,12,7430,Visa Dankort - on-us,,DKK,"Suspected malfunction
3       April,12,7430,Visa Dankort - on-us,,DKK,"Suspected malfunction
3       April,12,7430,Visa Dankort - on-us,,DKK,"Suspected malfunction
3       April,23,7430,MasterCard,,DKK,Timed-out taking money
```

Observations → The analysis provides reports when the given ATM malfunctioned/or had any issues. We provide the ATM_ID and it the MapReduce job will get the malfunctioned transactions.

```
    @Override
    protected void setup(Mapper<LongWritable, Text, Text, Text>.Context context)
            throws IOException, InterruptedException {

        getATM = context.getConfiguration().get("atm");

//      getATM = "1";
    }
    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException, InterruptedE

        String line = value.toString();
        String atm = "";

        try {
              atm = line.split(",")[6];
              String zipcode = line.split(",")[11];

              String year = line.split(",")[0];
              String month = line.split(",")[1];
              String currency = line.split(",")[14];
              String hour = line.split(",")[4];
              String card_type = line.split(",")[15];

              Text val = new Text(year+","+month+","+hour+","+zipcode+"\t"+card_type+","+currency);


              if(atm.equals(getATM)) {
                    context.write(new Text(atm), new Text(val));
              }


        } catch (Exception e) {
            atm = "Ambiguous data";

        }
```

Number of Transactions/Day of Week

The below analysis gets the number of transactions for each day. The Mapper takes the weekday as the key. MapReduce result gives us the count of transactions for each day of the week. Below are the results:

Output

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -ls /projectOutput
Found 1 items
drwxr-xr-x   - rohit supergroup          0 2021-04-11 12:47 /projectOutput/weekdaycount
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -ls /projectOutput/weekdaycount/
Found 2 items
-rw-r--r--   1 rohit supergroup          0 2021-04-11 12:47 /projectOutput/weekdaycount/_SUCCESS
-rw-r--r--   1 rohit supergroup        116 2021-04-11 12:47 /projectOutput/weekdaycount/part-r-00000
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -cat /projectOutput/weekdaycount/part-r-00000
Friday  482954
Monday  328844
Saturday        337716
Sunday  226540
Thursday        392251
Tuesday 347907
Wednesday       352360
weekday 2
```

We can observe following from the above result:

1. Maximum number of transactions happened during Thursdays, Fridays
2. Minimum number of transactions happened Saturdays, Sundays and Mondays

Monthly Trend

This analysis gives us the transactions trend over the month. The analysis gives insight on how the transactions happen based on the time of the month, that is, number of transactions for first 10 days, middle 10 days and end of the month. This will give some perspective on MapReduce job produces output

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/monthlytrend.jar transactionMonthlyTrend.DriverClass /project/dataset  /projectOutput/monthlytrend
2021-04-12 20:50:51,419 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032onthlyTrend.DriverClass   /projectOutput/monthlytrend
2021-04-12 20:50:51,859 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this
2021-04-12 20:50:51,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1618285456996_0001
2021-04-12 20:50:52,703 INFO input.FileInputFormat: Total input files to process : 1
2021-04-12 20:50:52,808 INFO mapreduce.JobSubmitter: number of splits:4
2021-04-12 20:50:52,991 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1618285456996_0001
2021-04-12 20:50:52,991 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-12 20:50:53,204 INFO conf.Configuration: resource-types.xml not found
2021-04-12 20:50:53,205 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-12 20:50:53,638 INFO impl.YarnClientImpl: Submitted application application_1618285456996_0001
2021-04-12 20:50:53,695 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1618285456996_0001/
2021-04-12 20:50:53,695 INFO mapreduce.Job: Running job: job_1618285456996_0001
2021-04-12 20:51:00,840 INFO mapreduce.Job: Job job_1618285456996_0001 running in uber mode : false
2021-04-12 20:51:00,842 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-12 20:51:29,842 INFO mapreduce.Job:  map 44% reduce 0%
2021-04-12 20:51:33,886 INFO mapreduce.Job:  map 84% reduce 0%
2021-04-12 20:51:34,898 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-12 20:51:40,938 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-12 20:51:41,951 INFO mapreduce.Job: Job job_1618285456996_0001 completed successfully
2021-04-12 20:51:42,059 INFO mapreduce.Job: Counters: 55
```

Output

```
                Bytes Written=103
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -ls /projectOutput/
Found 2 items
drwxr-xr-x   - rohit supergroup          0 2021-04-12 20:51 /projectOutput/monthlytrend
drwxr-xr-x   - rohit supergroup          0 2021-04-11 12:47 /projectOutput/weekdaycount
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -ls /projectOutput/monthlytrend
Found 2 items
-rw-r--r--   1 rohit supergroup          0 2021-04-12 20:51 /projectOutput/monthlytrend/_SUCCESS
-rw-r--r--   1 rohit supergroup        103 2021-04-12 20:51 /projectOutput/monthlytrend/part-r-00000
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -cat /projectOutput/monthlytrend/part-r-00000
Ambiguous data  2
End of the month        839941
First Ten days of the month     869789
Middle of the month     758842
```

Observations →

Max number of transactions are at the beginning of the month and at the end of the month

MapperClass

```
    @Override
    protected void map(LongWritable key, Text value, Mapper.Context con

        String line = value.toString();
        String monthDay = "";


        try {
            Integer day = Integer.parseInt(line.split(",")[2]);
            if(day <= 10)
                monthDay = "First Ten days of the month";
            else if(day > 10 && day <= 20)
                monthDay = "Middle of the month";
            else if(day > 20 && day <= 31)
                monthDay = "End of the month";
        } catch (NumberFormatException e) {
            monthDay = "Ambiguous data";

        }
```

ATMs with Hourly Traffic

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/atmswithmaxtraffic.jar toptenwithmaxtraffic.DriverClass /project/dataset  /projectOutput/atmswithmosttraffic
2021-04-13 10:58:13,628 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-13 10:58:13,999 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1618334179131_0003
2021-04-13 10:58:14,179 INFO input.FileInputFormat: Total input files to process : 1
2021-04-13 10:58:14,681 INFO mapreduce.JobSubmitter: number of splits:4
2021-04-13 10:58:14,835 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1618334179131_0003
2021-04-13 10:58:14,836 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-13 10:58:15,001 INFO conf.Configuration: resource-types.xml not found
2021-04-13 10:58:15,001 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-13 10:58:15,120 INFO impl.YarnClientImpl: Submitted application application_1618334179131_0003
2021-04-13 10:58:15,176 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1618334179131_0003/
2021-04-13 10:58:15,177 INFO mapreduce.Job: Running job: job_1618334179131_0003
2021-04-13 10:58:21,278 INFO mapreduce.Job: Job job_1618334179131_0003 running in uber mode : false
2021-04-13 10:58:21,281 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-13 10:58:54,728 INFO mapreduce.Job:  map 41% reduce 0%
2021-04-13 10:58:59,864 INFO mapreduce.Job:  map 100% reduce 0%
2021-04-13 10:59:07,923 INFO mapreduce.Job:  map 100% reduce 100%
2021-04-13 10:59:07,929 INFO mapreduce.Job: Job job_1618334179131_0003 completed successfully
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -ls /projectOutput/
Found 2 items
drwxr-xr-x   - rohit supergroup          0 2021-04-12 20:51 /projectOutput/monthlytrend
drwxr-xr-x   - rohit supergroup          0 2021-04-11 12:47 /projectOutput/weekdaycount
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop jar ~/Desktop/Project/Project_Jar/hourlyTraffic.jar hourlyTraffic.DriverClass /project/dataset /projectOutput/hourlyTraffic
2021-04-23 07:07:14,902 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-04-23 07:07:16,541 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-04-23 07:07:16,666 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/rohit/.staging/job_1619186614317_0001
2021-04-23 07:07:18,898 INFO input.FileInputFormat: Total input files to process : 1
2021-04-23 07:07:19,151 INFO mapreduce.JobSubmitter: number of splits:4
2021-04-23 07:07:19,899 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1619186614317_0001
2021-04-23 07:07:19,901 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-04-23 07:07:20,792 INFO conf.Configuration: resource-types.xml not found
2021-04-23 07:07:20,793 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-04-23 07:07:21,972 INFO impl.YarnClientImpl: Submitted application application_1619186614317_0001
2021-04-23 07:07:22,144 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1619186614317_0001/
2021-04-23 07:07:22,146 INFO mapreduce.Job: Running job: job_1619186614317_0001
2021-04-23 07:07:50,250 INFO mapreduce.Job: Job job_1619186614317_0001 running in uber mode : false
2021-04-23 07:07:50,253 INFO mapreduce.Job:  map 0% reduce 0%
2021-04-23 07:08:52,646 INFO mapreduce.Job:  map 16% reduce 0%
```

```
rohit@ubuntu:/usr/local/bin/hadoop-3.3.0/bin$ ./hadoop fs -cat /projectOutput/hourlyTraffic/part-r-00000
Ambiguous data  2
Evening 419850
Late night      190320
Midday  882919
Morning 894520
```

MapperClass

```
@Override
protected void map(LongWritable key, Text value, Mapper.Context context

    String line = value.toString();
    String hourOfTheDay = "";


    try {
        Integer hour = Integer.parseInt(line.split(",")[4]);
        if(hour > 0 && hour <= 8)
            hourOfTheDay = "Late night";
        else if(hour > 8 && hour <= 12)
            hourOfTheDay = "Morning";
        else if(hour > 12 && hour <= 16)
            hourOfTheDay = "Midday";
        else if(hour > 16 && hour <= 20)
            hourOfTheDay = "Evening";
        else if(hour > 20 && hour <= 0)
            hourOfTheDay = "Night";
    } catch (NumberFormatException e) {
        hourOfTheDay = "Ambiguous data";

    }



    if(hourOfTheDay.length() > 0) {
        context.write(new Text(hourOfTheDay),one);
    }
}
```

Pig Scripts

Top Ten ATMs with Maximum Traffic

Analysis gives top ten ATMs with maximum traffic and the transaction count.

```
 GNU nano 4.8                                                    ATMTraffic.pig
atm_data = LOAD '/home/rohit/Desktop/Project/Dataset/atm-dataset/atm_data_part1.csv' USING PigStorage(',');

selected_stuff = FOREACH atm_data GENERATE $6 AS atm_id, $16 AS service;


-- filtered = LIMIT selected_stuff 5;
-- DUMP filtered;


group_up = GROUP selected_stuff BY atm_id;

-- group_1 = LIMIT group_up 5;

-- DUMP group_1;

atmstraffic = FOREACH group_up GENERATE group AS ATM_ID, COUNT($1) as traffic;

atmstraffic_ordered = ORDER atmstraffic BY $1 DESC;

result = LIMIT atmstraffic_ordered 10;

filtered_atm = result;


STORE filtered_atm INTO 'ResultFiles/ATMTrafficTopTen/';
```

```
rohit@ubuntu:~/Desktop/Project/pig-scripts/ResultFiles/ATMTrafficTopTen$ tail part-r-00000
39      30453
45      27961
20      27599
24      26686
10      26408
2       23102
16      21522
1       21391
13      20483
48      20439
```

Top Ten Cities with maximum ATM traffic

Analysis gives top ten cities with maximum ATM traffic.

```
 GNU nano 4.8                                                    ATMTrafficCityTopTen.pig
atm_data = LOAD '/home/rohit/Desktop/Project/Dataset/atm-dataset/atm_data_part1.csv' USING PigStorage(',');

-- filtered = LIMIT atm_data 5;


-- DUMP filtered;


selected_stuff = FOREACH atm_data GENERATE $22 AS city, $4 AS hour;


-- filtered = LIMIT selected_stuff 5;
-- DUMP filtered;


group_up = GROUP selected_stuff BY city;

-- group_1 = LIMIT group_up 5;

-- DUMP group_1;

atmstraffic = FOREACH group_up GENERATE group AS CITY, COUNT($1) as traffic;

atmstraffic_ordered = ORDER atmstraffic BY $1 DESC;

result = LIMIT atmstraffic_ordered 10;

filtered_atm = result;


STORE filtered_atm INTO 'ResultFiles/ATMTrafficPerCityTopTen/';
```

```
rohit@ubuntu:~/Desktop/Project/pig-scripts/ResultFiles/ATMTrafficPerCityTopTen$ head part-r-00000
Aalborg 136129
Frederikshavn   84766
Svenstrup       76303
Hjorring        52961
Nørresundby     52515
Hadsund 51928
Skive   41770
Nykobing Mors   38323
Gistrup 37211
Roskilde        35606
```

Join Operation – ATM dataset and Weather Dataset

Join Operation on ATM dataset and weather dataset

```
rohit@ubuntu: ~/Desktop/Project/pig-scripts          rohit@ubuntu: ~/Desktop/Project/pig-
  GNU nano 4.8                                                        JoinOperation.pig
atm_data = LOAD '/home/rohit/Desktop/Project/atm-dataset' USING PigStorage(',');
weather_data = LOAD '/home/rohit/Desktop/Project/weather-dataset' USING PigStorage(',');

joined_data = JOIN atm_data BY ($0, $1, $2, $3, $4,$11), weather_data BY ($0,$1, $2, $3, $4, $5);

filtered_atm = joined_data;

STORE filtered_atm INTO 'ResultFiles/JoinedDatasetOutput/';
```

Execution -

```
Counters:
Total records written : 24655291
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local590043933_0001


2021-04-29 21:35:29,313 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2021-04-29 21:35:29,313 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2021-04-29 21:35:29,314 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
2021-04-29 21:35:29,324 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2021-04-29 21:35:29,406 [main] INFO  org.apache.pig.Main - Pig script completed in 1 minute, 53 seconds and 501 milliseconds (113501 ms)
```

Total time taken – 1 minute, 53 seconds

Output -

```
rohit@ubuntu:~/Desktop/Project/pig-scripts/ResultFiles/JoinedDatasetOutput$ head part-r-00000
2017    May     1       Monday  0       Active  78      Diebold Nixdorf Nyborg  Vestergade      35      5800    55.31843268     10.78110914     DKK     VISA    Withdrawal                      2017    May
1       Monday  0       5800    55.059818       10.60677        2612045 Svendborg       280.15  1017    81      8       100             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  34      NCR     Skipperen       Vestre Alle     2       9000    57.03389256     9.90814567      DKK     MasterCard      Withdrawal                      201
7       May     1       Monday  0       9000    57.048  9.9187  2624886 Aalborg 278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  34      NCR     Skipperen       Vestre Alle     2       9000    57.03389256     9.90814567      DKK     MasterCard      Withdrawal                      201
7       May     1       Monday  0       9000    57.048  9.9187  2624886 Aalborg 278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  15      NCR     Vestre  Kastetvej       36      9000    57.05344705     9.90529246      DKK     Mastercard - on-us      Withdrawal                      201
7       May     1       Monday  0       9000    57.048  9.9187  2624886 Aalborg 278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  15      NCR     Vestre  Kastetvej       36      9000    57.05344705     9.90529246      DKK     Mastercard - on-us      Withdrawal                      201
7       May     1       Monday  0       9000    57.048  9.9187  2624886 Aalborg 278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  11      NCR     Sauersvej       Fridtjof Nansens Vej    2       9210    57.0232364      9.9403567       DKK     Mastercard - on-us      Withdrawal      2
017     May     1       Monday  0       9210    57.04834        9.9352  2616235 Nørresundby     278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  11      NCR     Sauersvej       Fridtjof Nansens Vej    2       9210    57.0232364      9.9403567       DKK     Mastercard - on-us      Withdrawal      2
017     May     1       Monday  0       9210    57.04834        9.9352  2616235 Nørresundby     278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  11      NCR     Sauersvej       Fridtjof Nansens Vej    2       9210    57.0232364      9.9403567       DKK     MasterCard      Withdrawal      2
017     May     1       Monday  0       9210    57.04834        9.9352  2616235 Nørresundby     278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  11      NCR     Sauersvej       Fridtjof Nansens Vej    2       9210    57.0232364      9.9403567       DKK     MasterCard      Withdrawal      2
017     May     1       Monday  0       9210    57.04834        9.9352  2616235 Nørresundby     278.208 1035    72      5       139             0       800     Clear   Sky is Clear
2017    May     1       Monday  0       Active  39      NCR     Svenstrup       Godthåbsvej     14      9230    56.9728006      9.8511237       DKK     Visa Dankort    Withdrawal                      201
7       May     1       Monday  0       9230    56.972301       9.84806 2612021 Svenstrup       278.208 1035    72      5       139             0       800     Clear   Sky is Clear
```

Number of Transactions with issues

```
 GNU nano 4.8                                                    TransactionErrors.pig
atm_data = LOAD '/home/rohit/Desktop/Project/Dataset/atm-dataset/atm_data_part1.csv' USING PigStorage(',');

filter_stuff = FILTER atm_data BY $18 != '';

selected_stuff = FOREACH filter_stuff GENERATE $7 AS manufacturer, $18 AS error_message, $4 AS hour;

group_up = GROUP selected_stuff BY (manufacturer, error_message);

manufacturer_with_errors = FOREACH group_up GENERATE group AS manufacturer, COUNT($1) as traffic;

filtered_atm = manufacturer_with_errors;

STORE filtered_atm INTO 'ResultFiles/manufacturer_errors';
```

```
rohit@ubuntu:~/Desktop/Project/pig-scripts/ResultFiles/manufacturer_errors$ cat part-r-00000
(NCR,Suspected malfunction)     87
(NCR,"Suspected malfunction)    3313
(NCR,"Timed-out taking card)    39
(NCR,Timed-out taking money)    509
(NCR,No response received from host)    36
(Diebold Nixdorf,Suspected malfunction) 35
(Diebold Nixdorf,"Suspected malfunction)        664
(Diebold Nixdorf,"Timed-out taking card)        35
(Diebold Nixdorf,Timed-out taking money)        97
(Diebold Nixdorf,No response received from host)        15
(atm_manufacturer,message_text) 1
```

UDF Implementation

```java
package udfs;

import java.io.IOException;

import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;

public class PigEvalFunc extends EvalFunc<String>{

        public String exec(Tuple input) throws IOException {
           if (input == null || input.size() == 0)
           return "SUCCESS";
           String str = (String)input.get(0);
           return str.toUpperCase();
        }
    }
```

Pig Script

```
REGISTER udfs.jar;

atm_data = LOAD '/home/rohit/Desktop/Project/Dataset/atm-dataset' USING PigStorage(',') as (year:chararray,month:cha
atm_location:chararray,atm_streetname:chararray,atm_street_number:chararray,atm_zipcode:chararray,atm_lat:chararray,

selected_stuff = FOREACH atm_data GENERATE $1 AS messages, $5 AS status;

-- temp = FOREACH selected_stuff GENERATE udfs.PigEvalFunc(messages) as msgs, status;

group_up = GROUP selected_stuff BY messages;

result = FOREACH group_up GENERATE udfs.PigEvalFunc(group) AS messages, COUNT($1) as traffic;

--number_of_atms = FOREACH group_up GENERATE group AS messages, COUNT($1) as status;

filtered_atm = result;

STORE filtered_atm INTO 'ResultFiles/PigUDFOutput/';
```

Output

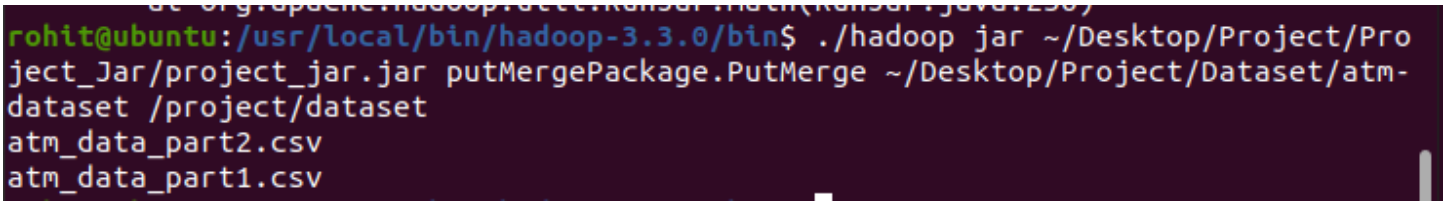
```
rohit@ubuntu:~/Desktop/Project/pig-scripts/ResultFiles/PigUDFOutput$ cat part-r-00000
MAY     667254
JULY    249904
JUNE    675498
APRIL   656595
MARCH   628758
MONTH   4
AUGUST  217218
JANUARY 540585
OCTOBER 191667
DECEMBER        197048
FEBRUARY        547977
NOVEMBER        193967
SEPTEMBER       202101
```

Implementing PutMerge

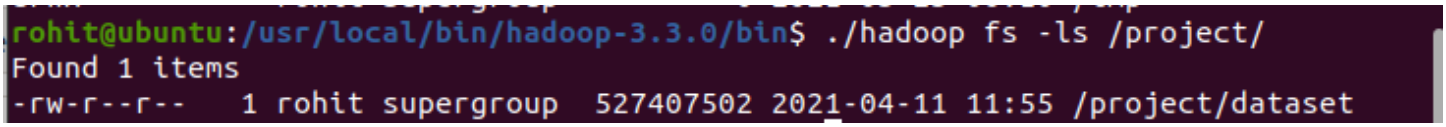
PutMerge.class

```
1  package putMergePackage;
2
3  import java.io.IOException;
4
5  import org.apache.hadoop.conf.Configuration;
6  import org.apache.hadoop.fs.FSDataInputStream;
7  import org.apache.hadoop.fs.FSDataOutputStream;
8  import org.apache.hadoop.fs.FileStatus;
9  import org.apache.hadoop.fs.FileSystem;
10 import org.apache.hadoop.fs.Path;
11
12 public class PutMerge {
13
14     public static void main(String[] args) throws IOException {
15
16         Configuration conf = new Configuration();
17         FileSystem hdfs = FileSystem.get(conf);
18         FileSystem local = FileSystem.getLocal(conf);
19
20         Path inputDir = new Path(args[0]);
21         Path hdfsFile = new Path(args[1]);
22
23         try {
24             FileStatus[] inputFiles = local.listStatus(inputDir);
25             FSDataOutputStream out = hdfs.create(hdfsFile);
26             for(int i = 0; i < inputFiles.length; i++) {
27                 System.out.println(inputFiles[i].getPath().getName());
28                 FSDataInputStream in = local.open(inputFiles[i].getPath());
29                 byte buffer[] = new byte[256];
30
31                 int bytesRead = 0;
32                 while((bytesRead = in.read(buffer)) > 0) {
33                     out.write(buffer, 0, bytesRead);
34                 }
35                 in.close();
36
37             }
38         out.close();
39         }
40         catch (IOException e) {
41             e.printStackTrace();
42         }
43     }
```

Running the jar file



Output



APPENDIX

ATMs/City

DriverClass

```
public class DriverClass {

    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException{


        Configuration conf = new Configuration();
        // Create a new Job
        Job job = Job.getInstance(conf,"atmsTransPerCity");
        job.setJarByClass(DriverClass.class);

        // Specify various job-specific parameters
        job.setJobName("atmsTransPerCity");


        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));


        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);


        job.setMapperClass(MapperClass.class);
        job.setReducerClass(ReducerClass.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        // Submit the job, then poll for progress until the job is complete
        System.exit(job.waitForCompletion(true)?0:1);

    }
}
```

MapperClass

```
public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String cityName = "";
```

```
        try {
             cityName = line.split(",")[22];
        } catch (Exception e) {
             cityName = "Ambiguous data";

        }



        if(cityName.length() > 0) {
             context.write(new Text(cityName),one);
        }
    }

}
```

ReducerClass

```
public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    // just like in mongoDB values is iterable
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {

        int sum=0;
        for(IntWritable v: values){
            sum += v.get();
        }

        context.write(key, new IntWritable(sum));

    }
}
```

Active/InActive ATM status

ActiveInactiveATMTuple

```
public class ActiveInactiveTuple implements Writable{


    private long active;
    private long inactive;




    public long getActive() {
             return active;
      }

      public void setActive(long active) {
             this.active = active;
      }
```

```
	public long getInactive() {
		return inactive;
	}

	public void setInactive(long inactive) {
		this.inactive = inactive;
	}

	@Override
	public String toString() {
		return "\t" + active + "\t" + inactive;
	}


    public void write(DataOutput out) throws IOException {

        out.writeLong(active);
        out.writeLong(inactive);

    }

    public void readFields(DataInput in) throws IOException {

        try {


            active = in.readLong();
            inactive = in.readLong();

        }catch (EOFException e){

        }

    }
}
```

DriverClass

```
public class DriverClass {

	public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException {

    // Create a new Job
    Job job = Job.getInstance();//(conf,"MinMaxStock");
    job.setJarByClass(DriverClass.class);

    // Specify various job-specific parameters
    job.setJobName("myjob");
    job.setMapperClass(MapperClass.class);
    job.setReducerClass(ReducerClass.class);
    job.setNumReduceTasks(1);
    TextInputFormat.addInputPath(job, new Path(args[0]));

    job.setInputFormatClass(TextInputFormat.class);

    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(ActiveInactiveTuple.class);
```

```
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(ActiveInactiveTuple.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    // Submit the job, then poll for progress until the job is complete
    System.exit(job.waitForCompletion(true)?0:1);

       }

}
```

MapperClass

```
public class MapperClass extends Mapper<Object, Text, Text, ActiveInactiveTuple> {

    private ActiveInactiveTuple outputTuple = new ActiveInactiveTuple();

    @Override
    protected void map(Object key, Text value, Context context) throws IOException,
InterruptedException {

        String[] atmLine= value.toString().split(",");
        if(!atmLine[0].equals("weather_city_name")){
            String city = atmLine[22];
            String status = atmLine[5];
            try {
             if(status.equals("Active")) {
                    outputTuple.setActive(1);
                    outputTuple.setInactive(0);
                    context.write(new Text(city), outputTuple);
             }
             else if(status.equals("Inactive")) {
                    outputTuple.setActive(0);
                    outputTuple.setInactive(1);
                    context.write(new Text(city), outputTuple);
             }
             //   context.write(new Text(city), outputTuple);

            } catch (Exception e) {
                e.printStackTrace();
            }
        }

    }

}
```

ReducerClass

```
public class
ReducerClass extends Reducer<Text, ActiveInactiveTuple, Text, ActiveInactiveTuple>{
```

```
     private ActiveInactiveTuple outputTuple = new ActiveInactiveTuple();

    @Override
    protected void reduce(Text key, Iterable<ActiveInactiveTuple> values, Context context) throws
IOException, InterruptedException {

       // First Count
        long active = 0;
        long inactive = 0;

        for(ActiveInactiveTuple v: values){
            active += v.getActive();
            inactive += v.getInactive();
        }
       active = active/2;
       inactive = inactive/2;

        outputTuple.setActive(active);
        outputTuple.setInactive(inactive);

       // Emit it
        context.write(key, outputTuple);
    }


}
```

Join Operation

DriverClass

```
public class
ReducerClass extends Reducer<Text, ActiveInactiveTuple, Text, ActiveInactiveTuple>{

     private ActiveInactiveTuple outputTuple = new ActiveInactiveTuple();

    @Override
    protected void reduce(Text key, Iterable<ActiveInactiveTuple> values, Context context) throws
IOException, InterruptedException {

       // First Count
        long active = 0;
        long inactive = 0;

        for(ActiveInactiveTuple v: values){
            active += v.getActive();
            inactive += v.getInactive();
        }
       active = active/2;
       inactive = inactive/2;

        outputTuple.setActive(active);
        outputTuple.setInactive(inactive);

       // Emit it
        context.write(key, outputTuple);
    }
```

```
}
```

AtmMapperClass

```
public class AtmsMapperClass extends Mapper<LongWritable, Text, Text, Text>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);
    private String getATM;

    @Override
      protected void setup(Mapper<LongWritable, Text, Text, Text>.Context context)
                  throws IOException, InterruptedException {

      getATM = context.getConfiguration().get("atm");

//     getATM = "1";
      }
    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String atm = "";

        try {
               atm = line.split(",")[6];
               String zipcode = line.split(",")[11];

               String year = line.split(",")[0];
              String month = line.split(",")[1];
              String currency = line.split(",")[14];
              String hour = line.split(",")[4];
              String card_type = line.split(",")[15];

               Text val = new
Text(atm+","+year+","+month+","+hour+","+zipcode+","+card_type+","+currency);
               Text newKey = new Text(year+","+month+","+hour+","+zipcode);

               if(atm.equals(getATM)) {
                          context.write(newKey, new Text("A"+val));
              }


        } catch (Exception e) {
             atm = "Ambiguous data";

        }




    }

}
```

WeatherClass

```
public class WeatherMapperClass extends Mapper<LongWritable, Text, Text, Text>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String atm = "";

        try {

               String year = line.split(",")[0];
              String month = line.split(",")[1];
              String hour = line.split(",")[4];
              String zipcode = line.split(",")[5];

              String cityName = line.split(",")[9];
              String weather_main = line.split(",")[18];

              Text val = new Text("W"+cityName+","+weather_main);
              Text newKey = new Text(year+","+month+","+hour+","+zipcode);

              context.write(newKey, new Text(val));


        } catch (Exception e) {
             atm = "Ambiguous data";

        }




    }

}
```

ReducerClass

```
public class ReducerClass extends Reducer<Text, Text, Text, Text> {

    private ArrayList<Text> listATM = new ArrayList<Text>();
    private ArrayList<Text> listW = new ArrayList<Text>();
    private static final Text EMPTY_TEXT = new Text("");



    @Override
```

```
    protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {
       listATM.clear();
       listW.clear();
        for(Text text : values){
            if(text.charAt(0) == 'A'){
             listATM.add(new Text(text.toString().substring(1)));
            } else if(text.charAt(0) == 'W'){
             listW.add(new Text(text.toString().substring(1)));
            }
        }
        joinMethod(context);
    }



    public void joinMethod(Context context) throws IOException, InterruptedException {
        String joinType = context.getConfiguration().get("join.type");

        //Inner Join
        if(joinType.equalsIgnoreCase("inner")){
            if(!listATM.isEmpty() && !listW.isEmpty()){
                for(Text atmData : listATM){
                    for(Text wData : listW){
                        context.write(atmData, wData);
                    }
                }
            }
        }

        else if(joinType.equalsIgnoreCase("leftOuter")){
            for(Text atmData : listATM){
                if(!listW.isEmpty()){
                    for(Text wData : listW){
                        context.write(atmData, wData);
                    }
                } else{
                    context.write(atmData, EMPTY_TEXT);
                }
            }
        }
    }
}
```

TopTenATMsTraffic

DriverClass

```
public class DriverClass {

    public static void main(String[] args) throws IOException, InterruptedException,
ClassNotFoundException{


        Configuration conf = new Configuration();
        Configuration conf2 = new Configuration();
        // Create a new Job
        Job job = Job.getInstance(conf,"toptentransactionsPerCity");
        job.setJarByClass(DriverClass.class);
```

```
        // Specify various job-specific parameters
        job.setJobName("toptentransactionsPerCity");


        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));


        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setMapperClass(MapperClass.class);
        job.setReducerClass(ReducerClass.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        boolean complete = job.waitForCompletion(true);

        Job job2 = Job.getInstance(conf2, "Chaining");

        if (complete) {
            FileInputFormat.addInputPath(job2, new Path(args[2]));
            FileOutputFormat.setOutputPath(job2, new Path(args[3]));
            job2.setJarByClass(DriverClass.class);
            job2.setMapperClass(MapperClass2.class);
            job2.setMapOutputKeyClass(Text.class);
            job2.setMapOutputValueClass(LongWritable.class);
            System.exit(job2.waitForCompletion(true) ? 0 : 1);

        }
    }
}
```

MapperClass

```
public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String cityName = "";


        try {
             cityName = line.split(",")[22];
        } catch (Exception e) {
             cityName = "Ambiguous data";

        }
```

```
        if(cityName.length() > 0) {
             context.write(new Text(cityName),one);
        }
    }

}
```

MapperClass2

```
public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String cityName = "";


        try {
             cityName = line.split(",")[22];
        } catch (Exception e) {
             cityName = "Ambiguous data";

        }



        if(cityName.length() > 0) {
             context.write(new Text(cityName),one);
        }
    }

}
```

ReducerClass

```
public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    // just like in mongoDB values is iterable
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {

        int sum=0;
        for(IntWritable v: values){
            sum += v.get();
        }

        context.write(key, new IntWritable(sum));

    }
}
```

MonthlyTrend

MapperClass

```
public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();
        String monthDay = "";


        try {
             Integer day = Integer.parseInt(line.split(",")[2]);
             if(day <= 10)
             monthDay = "First Ten days of the month";
            else if(day > 10 && day <= 20)
             monthDay = "Middle of the month";
            else if(day > 20 && day <= 31)
             monthDay = "End of the month";
        } catch (NumberFormatException e) {
             monthDay = "Ambiguous data";

        }



        if(monthDay.length() > 0) {
             context.write(new Text(monthDay),one);
        }
    }

}
```

ReducerClass

```
public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    // just like in mongoDB values is iterable
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {

        int sum=0;
        for(IntWritable v: values){
            sum += v.get();
            // can we use this--  Integer.parseInt(v.toString());
        }

        context.write(key, new IntWritable(sum));

    }

}
```

Weekday Transaction count

MapperClass

```
public class MapperClass extends Mapper<LongWritable, Text, Text, IntWritable>{

    //    hadoop datatype
    Text word = new Text();
    IntWritable one = new IntWritable(1);

    @Override
    protected void map(LongWritable key, Text value, Mapper.Context context) throws IOException,
InterruptedException {

        String line = value.toString();

        String weekDay = line.split(",")[3];

        if(weekDay.length() > 0) {
    //    if(logs.length>0){
  //      String ipAddress = logs[0];

        //        word.set(ipAddress);
             context.write(new Text(weekDay),one);

        }
        //super.map(key, value, context); //To change body of generated methods, choose Tools |
Templates.
    }

}
```

ReducerClass

```
public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    // just like in mongoDB values is iterable
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {

        int sum=0;
        for(IntWritable v: values){
            sum += v.get();
            // can we use this--  Integer.parseInt(v.toString());
        }

        context.write(key, new IntWritable(sum));

        //super.reduce(key, values, context); //To change body of generated methods, choose Tools
| Templates.
    }

}
```