

A	B	C	D	E	F	G	H	I	J	K	L	M
Transaction_date	Product	Price	Payment_Type	Name	City	State	Country	Account_Crea	Last_Login	Latitude	Longitude	
01-02-2009 06:17	Product1	1200	Mastercard	carolina	Basildon	England	United Kingdom	01-02-2009 06:00	01-02-2009 06:08	51.5	-1.11667	
01-02-2009 04:53	Product1	1200	Visa	Betina	Parkville	MO	United States	01-02-2009 04:42	01-02-2009 07:49	39.195	-94.6819	
01-02-2009 13:08	Product1	1200	Mastercard	Federica	Astoria	OR	United States	01-01-2009 16:21	01-03-2009 12:32	46.18806	-123.83	
01-03-2009 14:44	Product1	1200	Visa	Gouya	Echuca	Victoria	Australia	9/25/05 21:13	01-03-2009 14:22	-36.1333	144.75	
01-04-2009 12:56	Product2	3600	Visa	Gerd W	Cahaba Hc	AL	United States	11/15/08 15:47	01-04-2009 12:45	33.52056	-86.8025	
01-04-2009 13:19	Product1	1200	Visa	LAURENCE	Mickleton	NJ	United States	9/24/08 15:19	01-04-2009 13:04	39.79	-75.2381	
01-04-2009 20:11	Product1	1200	Mastercard	Fleur	Peoria	IL	United States	01-03-2009 09:38	01-04-2009 19:45	40.69361	-89.5889	
01-02-2009 20:09	Product1	1200	Mastercard	adam	Martin	TN	United States	01-02-2009 17:43	01-04-2009 20:01	36.34333	-88.8503	
01-04-2009 13:17	Product1	1200	Mastercard	Renee Elis	Tel Aviv	Tel Aviv	Israel	01-04-2009 13:03	01-04-2009 22:10	32.06667	34.76667	
01-04-2009 14:11	Product1	1200	Visa	Aidan	Chatou	Ile-de-Fra	France	06-03-2008 04:22	01-05-2009 01:17	48.88333	2.15	
01-05-2009 02:42	Product1	1200	Diners	Stacy	New York	NY	United States	01-05-2009 02:23	01-05-2009 04:59	40.71417	-74.0064	
01-05-2009 05:39	Product1	1200	Amex	Heidi	Eindhoven	Noord-Br	Netherlands	01-05-2009 04:55	01-05-2009 08:15	51.45	5.466667	
01-02-2009 09:16	Product1	1200	Mastercard	Sean	Shavano F	TX	United States	01-02-2009 08:32	01-05-2009 09:05	29.42389	-98.4933	
01-05-2009 10:08	Product1	1200	Visa	Georgia	Eagle	ID	United States	11-11-2008 15:53	01-05-2009 10:05	43.69556	-116.353	
01-02-2009 14:18	Product1	1200	Visa	Richard	Riverside	NJ	United States	12-09-2008 12:07	01-05-2009 11:01	40.03222	-74.9578	
01-04-2009 01:05	Product1	1200	Diners	Leanne	Julianstov	Meath	Ireland	01-04-2009 00:00	01-05-2009 13:36	53.67722	-6.31917	
01-05-2009 11:37	Product1	1200	Visa	Janet	Ottawa	Ontario	Canada	01-05-2009 09:35	01-05-2009 19:24	45.41667	-75.7	
01-06-2009 05:02	Product1	1200	Diners	barbara	Hyderabad	Andhra Pr	India	01-06-2009 02:41	01-06-2009 07:52	17.38333	78.46667	
01-06-2009 07:45	Product2	3600	Visa	Sabine	London	England	United Kingdom	01-06-2009 07:00	01-06-2009 09:17	51.52721	0.14559	
01-02-2009 07:35	Product1	1200	Diners	Hani	Salt Lake C	UT	United States	12/30/08 5:44	01-06-2009 10:52	40.76083	-111.89	
01-06-2009 12:56	Product1	1200	Visa	Jeremy	Manchest	England	United Kingdom	01-06-2009 10:58	01-06-2009 13:31	53.5	-2.21667	
01-03-2009 11:05	Product1	1200	Diners	Jojo	Ballyvaughan	Cork	Ireland	12-10-2007 12:37	01-07-2009 01:53	51.85206	-8.58	

Aim: Write MapReduce code to find countrywise and payment method wise total sales.

Country-wise

```
import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.io.LongWritable;
```

```

public class SalesCountryDriver {
    public static class SalesMapper
    extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private final Text word = new Text();
        // number of time mapper runs = no. of rows in csv file , done by recorder
        public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
            if (key.get() == 0) {
                return;
            } else {
                String valueString = value.toString();
                String[] SingleCountryData = valueString.split(",");
                context.write(new Text(SingleCountryData[7]), new
                IntWritable(Integer.parseInt(SingleCountryData[2])));
            }
        }
    }
}

```

```

    public static class SalesCountryReducer extends Reducer<Text,
    IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();
        public void reduce(Text t_key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException {

```

```
int sum = 0;
for (IntWritable val : values) {
    sum += val.get();
}
result.set(sum);
context.write(t_key, result);
}
}
```

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Sale per Country");
    job.setJarByClass(SalesCountryDriver.class);
    job.setMapperClass(SalesMapper.class);
    job.setCombinerClass(SalesCountryReducer.class);
    job.setReducerClass(SalesCountryReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[1]));
    FileOutputFormat.setOutputPath(job, new Path(args[2]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

Block information -- Block 0

Block ID: 1073741830

Block Pool ID: BP-666625786-127.0.0.1-1653975222207

Generation Stamp: 1006

Size: 820

Availability:

- localhost

File contents

Switzerland 76800

Thailand 4800

The Bahamas 2400

Turkey 7200

Ukraine 1200

United Arab Emirates 12000

United Kingdom 144000

United States 737000

Payment-wise

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.io.LongWritable;
```

```

public class SalesPaymentDriver {
    public static class SalesMapper
    extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private final Text word = new Text();
        // number of time mapper runs = no. of rows in csv file , done
        by recorder
        public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
            if (key.get() == 0) {
                return;
            } else {
                String valueString = value.toString();
                String[] SingleCountryData = valueString.split(",");
                context.write(new Text(SingleCountryData[3]), new
                IntWritable(Integer.parseInt(SingleCountryData[2])));
            }
        }
    }
}

```

```

    public static class SalesPaymentReducer extends Reducer<Text,
    IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();
        public void reduce(Text t_key, Iterable<IntWritable> values,

```

```

Context context) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
        sum += val.get();
    }
    result.set(sum);
    context.write(t_key, result);
}
}

```

```

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Sale per Payment");
    job.setJarByClass(SalesPaymentDriver.class);
    job.setMapperClass(SalesMapper.class);
    job.setCombinerClass(SalesPaymentReducer.class);
    job.setReducerClass(SalesPaymentReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[1]));
    FileOutputFormat.setOutputPath(job, new Path(args[2]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```

Block information -- Block 0 ⬆️⬆️

Block ID: 1073741831

Block Pool ID: BP-666625786-127.0.0.1-1653975222207

Generation Stamp: 1007

Size: 56

Availability:

- localhost

File contents

Amex	188900
Diners	133800
Mastercard	458450
Visa	836350