→ Reduce side join

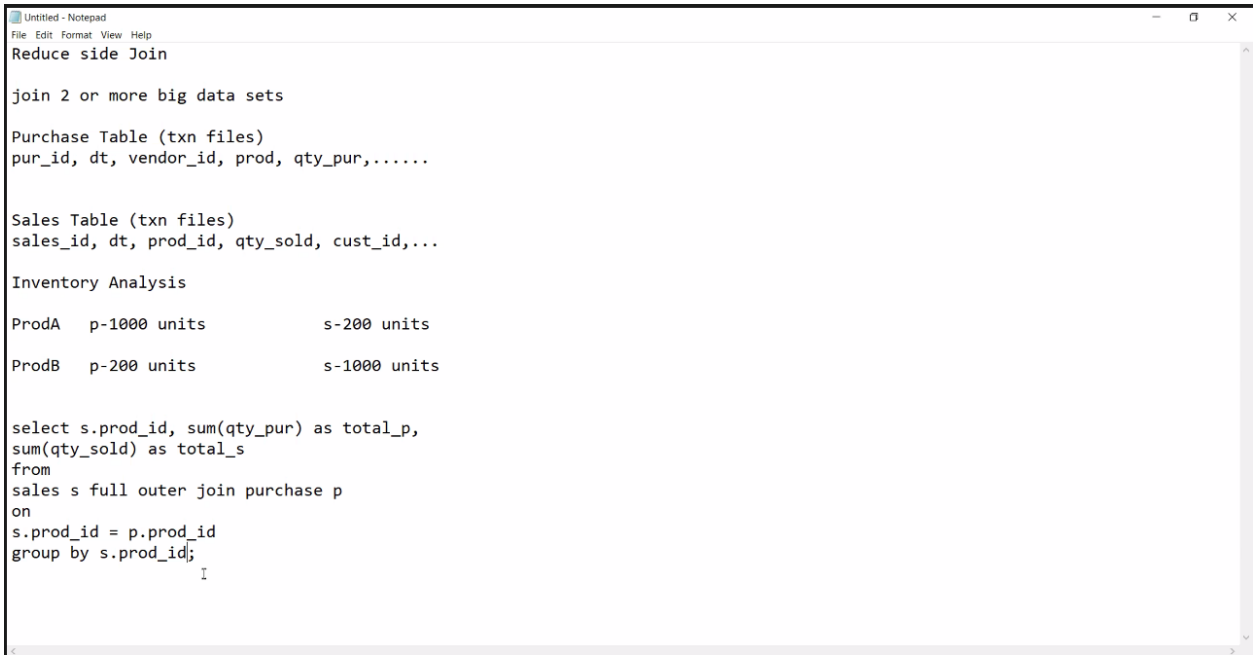
a. Joining 2 or more big data sets
b. 100TB with 100TB
c. It is always FULL OUTER JOIN result
d. 2 or more multiple mappers needed for reduce side join, but key should be same in all the mappers

Purchase Table
pur_id, dt, vendor_id, prod, qty_pur, …

Sales Table
sales_id, dt, prod_id, qty_sold, cust_id, …

Inventory analysis
ProdA p-1000 units s-200 units

```
SELECT s.prod_id, sum(qty_pur) , sum(qty_sold) FROM sales s
FULL OUTER JOIN purchase p ON s.prod_id= p.prod_id
GROUP BY s.prod_id;
```



e. 2 or more multiple mappers needed for reduce side join, but key should be same in all the mappers

Mapping stage:

sales_mapper	purchase _mapper
prod_id, qty_sold	prod_id, qty_pur
A, s.150	A, p.300
B, s.600	B, p.75
A, s.50	A, p.700
B, s.400	B, p.125

Shuffle Stage:

A, [s.150, s.50, p.300, p.700]
B, [s.600, s.400, p.75, p.125]

- An identifier is needed to identify which value is from which mapper, so we should make it like key, <mapper_identifier, some delimiter, value> instead of just key, value

Reducer stage:

A, Total_sale = 150+50	B, Total_sale = 600+400
A, Total_pur = 300+700	B, Total_pur = 75+125

A, (p-1000, s-200)
B, (p-200, s-1000)

- Identifier & delimiter is very much needed to identify which data is from which mapper, otherwise it is stuck in shuffle stage

→ Map-Side Join is faster than Reduce-Side Join

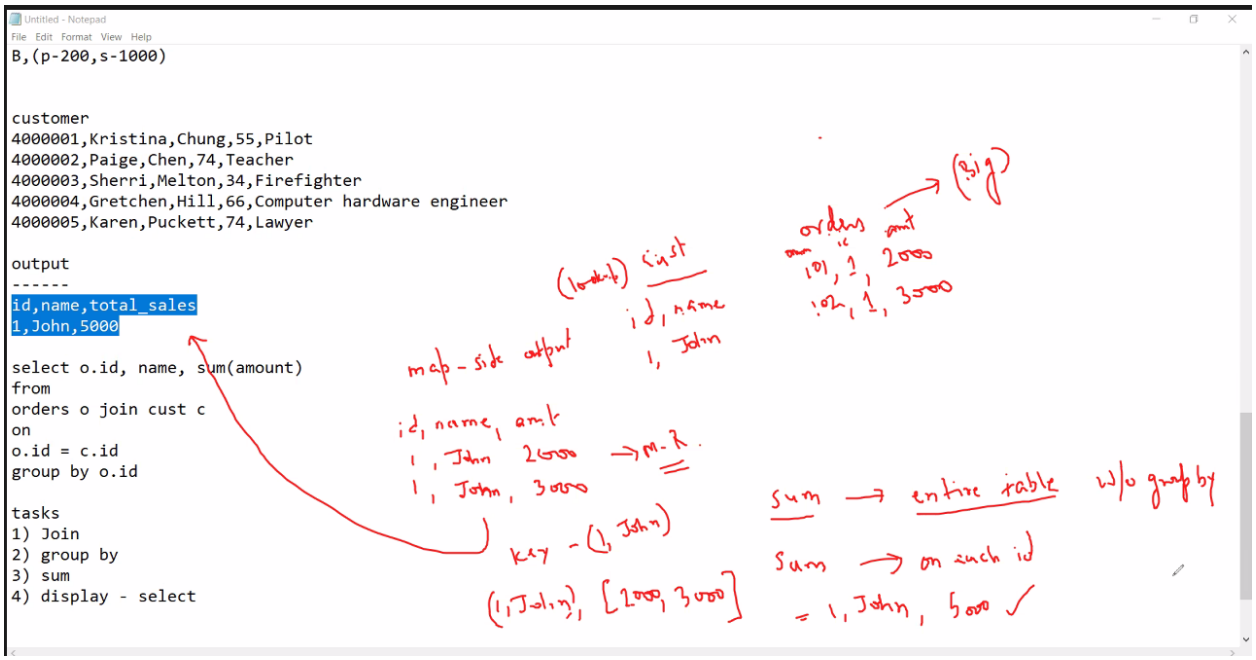
a. Only mapper is needed in map-side join
b. One lookup-file data is in memory in map-side join

(lookup) cust	(BigData) orders
id, name	order_id, id, amt
1, John	101, 1, 2000
	102, 1, 3000

SELECT c.id, c.name, sum(o.amt) FROM customers c
JOIN orders o ON c.id=o.id GROUP BY o.id;

→ Tasks in Reduce-Side Join

a. Join
b. Group By
c. Sum
d. Display - Select



Customer

```
4000001,Kristina,Chung,55,Pilot
4000002,Paige,Chen,74,Teacher
4000003,Sherri,Melton,34,Firefighter
4000004,Gretchen,Hill,66,Computer hardware engineer
4000005,Karen,Puckett,74,Lawyer
```

Txns

```
00000000,06-26-2011,4007024,040.33,Exercise & Fitness,Cardio Machine Accessories,Clarksville,Tennessee,credit
00000001,05-26-2011,4006742,198.44,Exercise & Fitness,Weightlifting Gloves,Long Beach,California,credit
00000002,06-01-2011,4009775,005.58,Exercise & Fitness,Weightlifting Machine Accessories,Anaheim,California,credit
00000003,06-05-2011,4002199,198.19,Gymnastics,Gymnastics Rings,Milwaukee,Wisconsin,credit
```

→ Problem statement: find total amount sold, total count of txns for each cust firstname
Kristina 238.77 2

Customer Mapper
4000001, (cust Kristina)

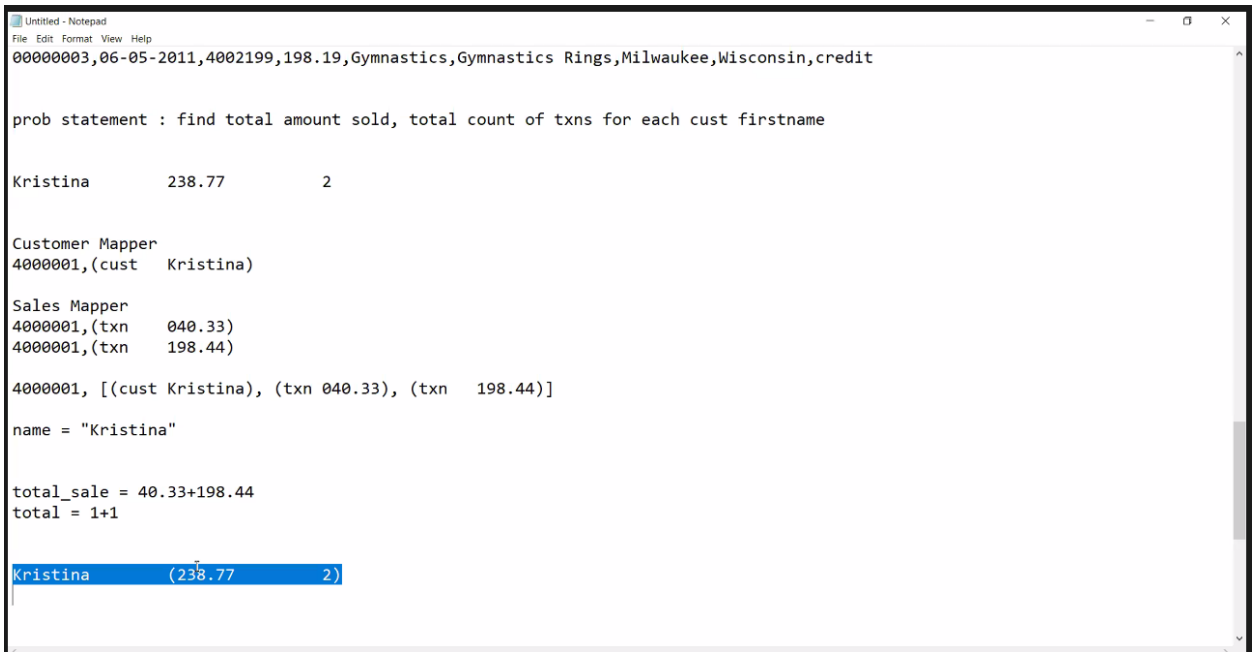
Sales Mapper
4000001, (txn, 040.33)
4000001, (txn, 198.44)

4000001, [(cust Kristina), (txn, 040.33), (txn, 198.44)]

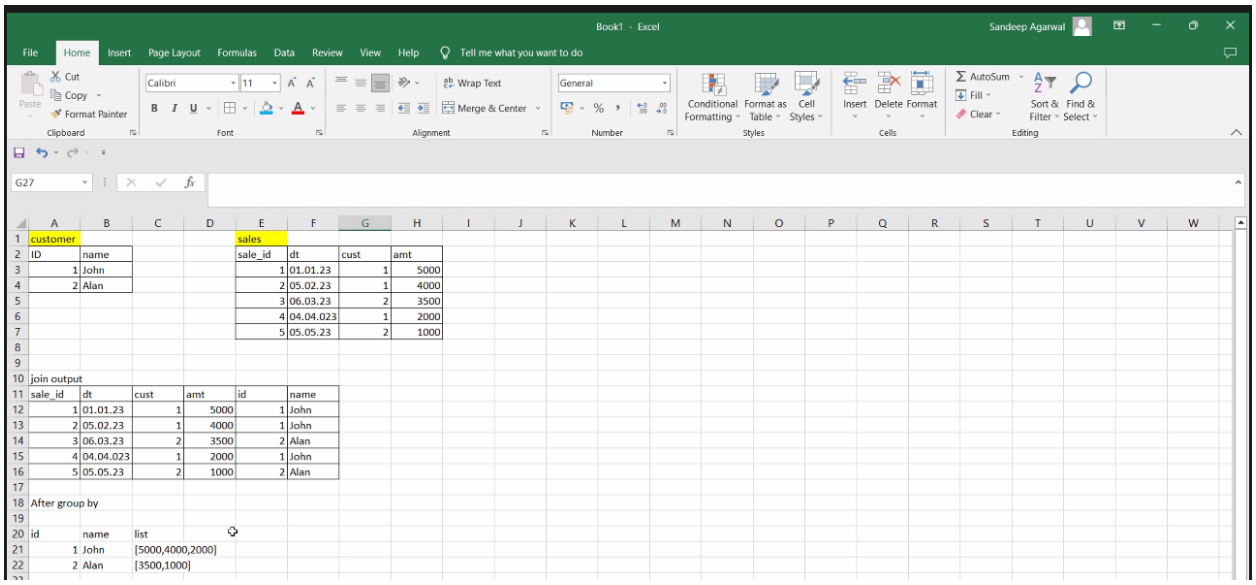
name = “Kristina”

Total_sale = 40.33 + 198.44
Total = 1 + 1

Kristina (237.77 2)



→



customer

ID	name
1	John
2	Alan

sales

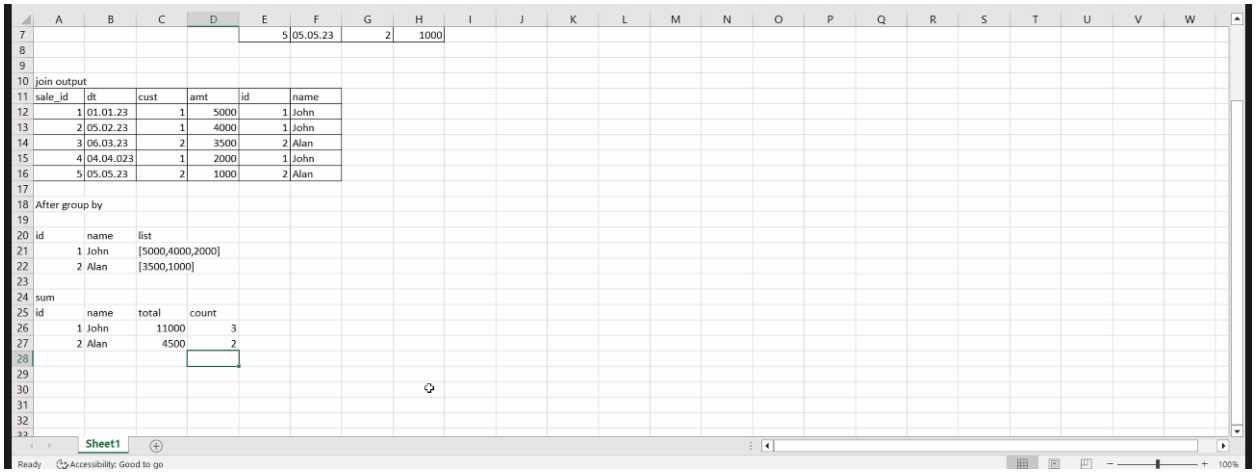
sale_id	dt	cust	amt
1	01.01.23	1	5000
2	05.02.23	1	4000
3	06.03.23	2	3500
4	04.04.023	1	2000
5	05.05.23	2	1000

join output

sale_id	dt	cust	amt	id	name
1	01.01.23	1	5000	1	John
2	05.02.23	1	4000	1	John
3	06.03.23	2	3500	2	Alan
4	04.04.023	1	2000	1	John
5	05.05.23	2	1000	2	Alan

After group by

id	name	list
1	John	[5000,4000,2000]
2	Alan	[3500,1000]



join output

sale_id	dt	cust	amt	id	name
1	01.01.23	1	5000	1	John
2	05.02.23	1	4000	1	John
3	06.03.23	2	3500	2	Alan
4	04.04.023	1	2000	1	John
5	05.05.23	2	1000	2	Alan

After group by

id	name	list
1	John	[5000,4000,2000]
2	Alan	[3500,1000]

sum

id	name	total	count
1	John	11000	3
2	Alan	4500	2

SELECT name, sum(amount), count(s.id) FROM
customers c FULL OUTER JOIN sales s ON c.id = s.id
GROUP BY s.id, name;

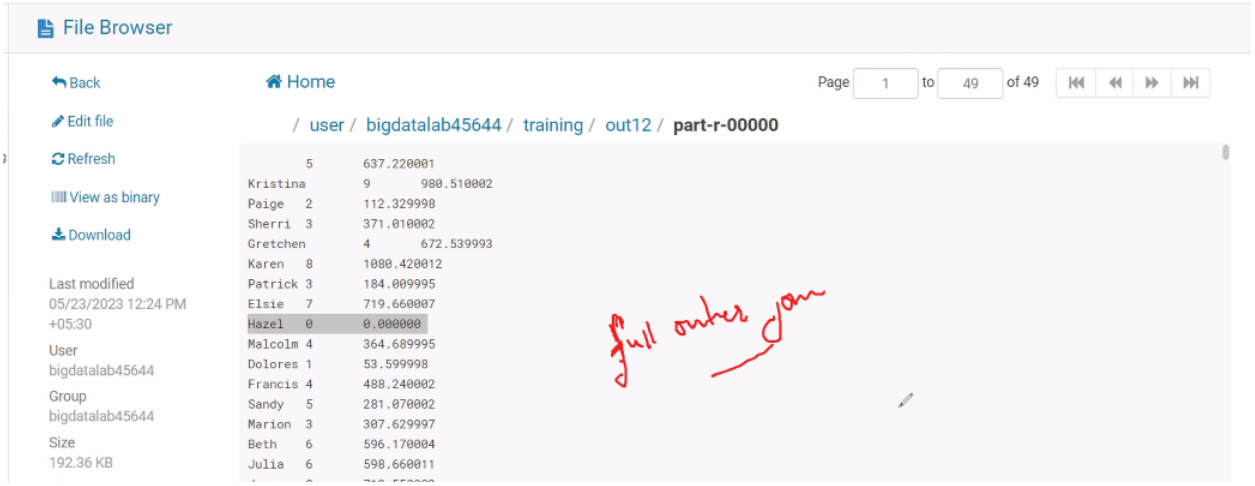
#

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop jar myjar.jar ReduceJoin
training/custs.txt training/txns1.txt training/out12
WARNING: Use "yarn jar" to launch YARN applications.
23/05/23 06:53:44 INFO client.RMProxy: Connecting to ResourceManager
at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/05/23 06:53:44 WARN mapreduce.JobResourceUploader: Hadoop
command-line option parsing not performed. Implement the Tool
interface and execute your application with T
oolRunner to remedy this.
```

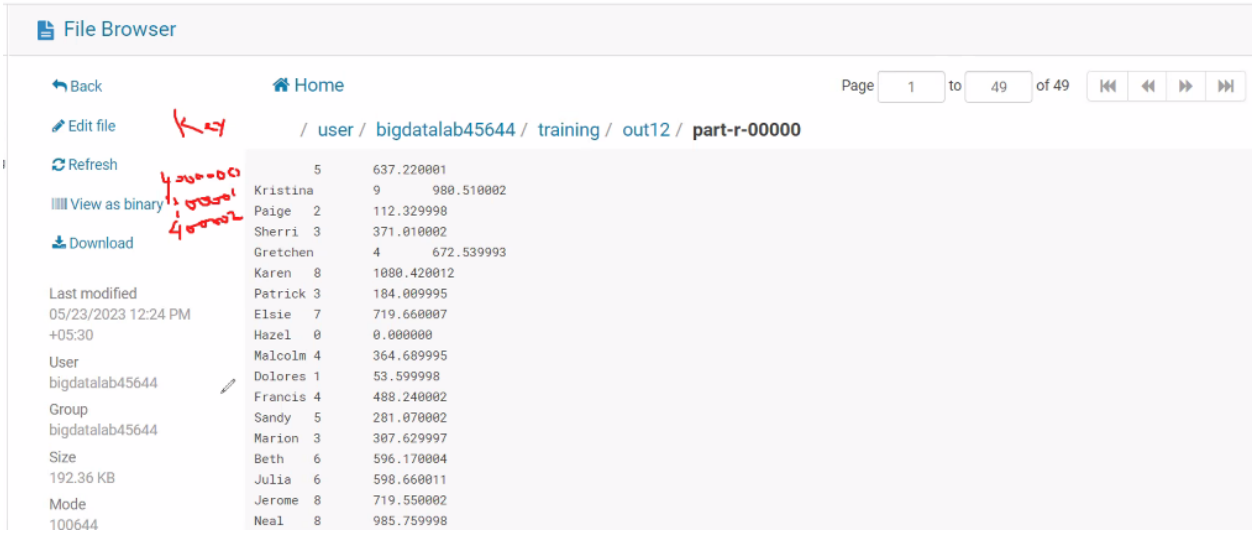
```
23/05/23 06:53:44 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path:
/user/bigdatalab456422/.staging/job_1684298513961_1195
23/05/23 06:53:45 INFO input.FileInputFormat: Total input files to
process : 1
23/05/23 06:53:45 INFO input.FileInputFormat: Total input files to
process : 1
23/05/23 06:53:45 INFO mapreduce.JobSubmitter: number of splits:2
23/05/23 06:53:45 INFO Configuration.deprecation:
yarn.resourcemanager.system-metrics-publisher.enabled is deprecated.
Instead, use yarn.system-metrics-publisher.enable
d
23/05/23 06:53:45 INFO mapreduce.JobSubmitter: Submitting tokens for
job: job_1684298513961_1195
23/05/23 06:53:45 INFO mapreduce.JobSubmitter: Executing with tokens:
[]
23/05/23 06:53:45 INFO conf.Configuration: resource-types.xml not
found
23/05/23 06:53:45 INFO resource.ResourceUtils: Unable to find
'resource-types.xml'.
23/05/23 06:53:45 INFO impl.YarnClientImpl: Submitted application
application_1684298513961_1195
23/05/23 06:53:45 INFO mapreduce.Job: The url to track the job:
http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/applicati
on_1684298513961_1195/
23/05/23 06:53:45 INFO mapreduce.Job: Running job:
job_1684298513961_1195
23/05/23 06:54:14 INFO mapreduce.Job: Job job_1684298513961_1195
running in uber mode : false
23/05/23 06:54:14 INFO mapreduce.Job:  map 0% reduce 0%
23/05/23 06:55:20 INFO mapreduce.Job:  map 50% reduce 0%
23/05/23 06:55:25 INFO mapreduce.Job:  map 100% reduce 0%
23/05/23 06:55:50 INFO mapreduce.Job:  map 100% reduce 100%
23/05/23 06:55:52 INFO mapreduce.Job: Job job_1684298513961_1195
completed successfully
23/05/23 06:55:53 INFO mapreduce.Job: Counters: 54
       File System Counters
               FILE: Number of bytes read=509016
               FILE: Number of bytes written=1648298
               FILE: Number of read operations=0
               FILE: Number of large read operations=0
               FILE: Number of write operations=0
               HDFS: Number of bytes read=4810551
               HDFS: Number of bytes written=196981
               HDFS: Number of read operations=11
```

```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots
(ms)=129660
                Total time spent by all reduces in occupied slots
(ms)=22300
                Total time spent by all map tasks (ms)=129660
                Total time spent by all reduce tasks (ms)=22300
                Total vcore-milliseconds taken by all map tasks=129660
                Total vcore-milliseconds taken by all reduce
tasks=22300
                Total megabyte-milliseconds taken by all map
tasks=132771840
                Total megabyte-milliseconds taken by all reduce
tasks=22835200
        Map-Reduce Framework
                Map input records=59999
                Map output records=59999
                Map output bytes=1206622
                Map output materialized bytes=470223
                Input split bytes=513
                Combine input records=0
                Combine output records=0
                Reduce input groups=10000
                Reduce shuffle bytes=470223
                Reduce input records=59999
                Reduce output records=10000
                Spilled Records=119998
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=2426
                CPU time spent (ms)=7290
                Physical memory (bytes) snapshot=1229955072
                Virtual memory (bytes) snapshot=7771475968
                Total committed heap usage (bytes)=1314390016
                Peak Map Physical memory (bytes)=499953664
                Peak Map Virtual memory (bytes)=2591252480
                Peak Reduce Physical memory (bytes)=270733312
                Peak Reduce Virtual memory (bytes)=2599899136
```

```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=196981
```



→ data is always sorted on key, but we're not printing key (key is cust_id), thus data is not sorted



→ Note:

a. HIVE tool takes SQL query, converts SQL query into JAR file internally, and launches jar file on ResourceManager