**SALES OPTIMIZATION**

**THROUGH HIGH-PERFORMANCE DATA ANALYSIS WITH HADOOP**

# INTRODUCTION

MapReduce is a programming model for handling large amounts of data. Hadoop is an open- source software framework that stores and processes large amounts of data using the MapReduce model. It is intended to handle large amounts of data by dividing the workload among multiple servers. Hadoop is made up of two main components: HDFS, a distributed file system for storing data, and MapReduce, a data processing engine. Hadoop is widely used for big data analysis and processing in industries such as finance, healthcare, and retail.

# DATASET AND RESEARCH QUESTION

Dataset Description of Sale\_sample.txt file

Order ID => This is the Order ID that is unique to each Customer. Order Date => This is Order Date of the product.

Ship Date => Shipping Date of the each Product.

Ship Mode=> Shipping Mode specified by the each Customer. Customer ID => This is the Unique ID to identify each Customer. Customer Name => This field is the Name of the each Customer. Segment => The segment to which the Customer belongs.

Country => This represents Customer's country of residence. City => This is the city of residence of the Customer.

State => The Customer's state of residence.

Postal Code => Postal Code of each Customer.

Region => This tells the Region where each Customer belong. Product ID => This is the Product ID that is unique to each customer. Category => This is the Category of the product ordered.

Sub-Category => It represents Sub-Category of the product ordered. Sales => The product's sales.

Quantity => The amount of the product.

Discount => It tells us the amount of discount provided. Profit => Profit/loss was made.

# Question:

The research question for this paper is to determine the total sales of each state by category and which subcategory contributed the most to those sales. The sales manager can use this information for "Strategic Planning" to plan future sales strategies for the state. For example, if the subcategory with the highest contribution was electronics, the sales manager may plan to increase the state's emphasis on electronics sales. Additionally for "Resource Allocation," the sales manager can allocate resources and efforts to the subcategories that contributed the most to the state's sales.

# DESIGN AND IMPLEMENTATION

















# EXPLANATION

This proposed framework includes four jobs: job 1, job 2, job 3, and job 4. The information is retrieved from the text file salesanalysis.txt. Job 1's task is to calculate the total sales of a specific state. Similarly, job 2's task is to determine total sales based on the combination of state and sub-category. So, from the same text file, Sales\_mapper1 will extract information such as state and sales, and Sales\_mapper2 will extract information such as state, sub-category, and sales. Now, Sales\_reducer1 of job 1 will return total sales for the state, and Sales\_reducer2 of job 2 will return total sales for each pair of state and sub-category. Job 3's Sales\_mapper3 will now read the output of Sales\_reducer2 and provide the output key as state and value as sub-category and sales. Then, Sales\_reducer3 of job 3 will determine the sub-category with the highest sales based on state. Job4's Sales\_mapper4 and Sales\_mapper5 will now read the outputs of Sales\_reducer1 and Sales\_reducer3. Both mappers will generate an output key of state and include a word called prefix in values to inform Sales\_reducer4 that the data it is reading is of total sales of state or the subcategory with the highest sales according to state. Sales\_reducer4 will now generate output key as state and value as information containing total sales of a state and subcategory with the highest sales and its sales value.

Formation of key value pairs for each mapper-reducer:

| Component | Input | Input Type | Output | Output Type |
| --- | --- | --- | --- | --- |
| Sales\_mapper1 | Will read data | Key = | Key = state | Key = Text |
|  | from file | LongWritable | and category | Value = |
|  | Salesanalysis.txt | Value = Text | Value = | DoubleWritable |
|  | Key = location |  | Sales |  |
|  | of the file |  |  |  |
|  | Value = a row |  |  |  |
|  | of the file |  |  |  |
| Sales\_reducer1 | Key = state | Key = Text | Key = state | Key = Text |
|  | Value = array of | Value = Iterable | and category | Value = |
|  | all Sales for a | <DoubleWritable> | Value = | DoubleWritable |
|  | combination of |  | totalsales |  |
|  | state and |  |  |  |
|  | category |  |  |  |

| Component | Input | Input Type | Output | Output Type |
| --- | --- | --- | --- | --- |
| Sales\_mapper | Will read data | Key = | Key = pair of | Key = Text |
| 2 | from file | LongWritable | state, category | Value = |
|  | Salesanalysis.tx | Value = Text | ,subcategory | DoubleWritabl |
|  | t |  | Value = Sales | e |
|  | Key = location |  |  |  |
|  | of the file |  |  |  |
|  | Value = a row |  |  |  |
|  | of the file |  |  |  |

| Sales\_reducer | Key = pair of | Key = Text | Key = state, | Key = Text |
| --- | --- | --- | --- | --- |
| 2 | state, category, | Value = Iterable | category, | Value = |
|  | subcategory | <DoubleWritable | subcategory | DoubleWritabl |
|  | Value = array | > | Value = | e |
|  | of all Sales fora |  | subcat\_totalsale |  |
|  | pair of state, |  | s |  |
|  | category, |  |  |  |
|  | subcategory |  |  |  |

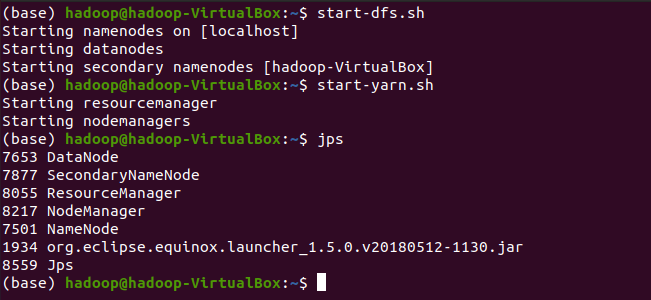
| Component | Input | Input Type | Output | Output Type |
| --- | --- | --- | --- | --- |
| Sales\_mapper3 | Will read data | Key = | Key = state and | Key = Text |
|  | from output file | LongWritable | category | Value = |
|  | of | Value = Text | Value = | Text |
|  | Sales\_reducer2. |  | subcategory and |  |
|  | Key = location |  | subcat\_total |  |
|  | of the file |  | Sales |  |
|  | Value = a row |  |  |  |
|  | of the file |  |  |  |
| Sales\_reducer3 | Key = state and | Key = Text | Key = state and | Key = Text |
|  | category | Value = | category | Value = |
|  | Value = array of | Iterable | Value = | Text |
|  | all subacategory | <Text> | (subcategory |  |
|  | and |  | with the highest |  |
|  | subcat\_totalsales |  | sales, and |  |
|  | as a text for a |  | subcat\_totalsales) |  |
|  | combination of |  |  |  |
|  | state and |  |  |  |
|  | category |  |  |  |

| Component | Input | Input Type | Output | Output Type |
| --- | --- | --- | --- | --- |
| Sales\_reducer4 | Reads output of Sales\_reducer1 Key = file location  Value = a row of file | Key = LongWritable Value = Text | Key = State and category Value =  “totalsales”  ,total\_sales | Key = Text Value =Text |
| Sales\_mapper5 | Reads output of Sales\_reducer3 Key = file location  Value = a row of file | Key = LongWritable Value = Text | Key = State and category Value =  “topsubcategory”  and subcat\_totalsales | Key = Text Value =Text |
| Sales\_reducer4 | Key = State and category  Value = array of an information | Key = Text Value = Iterable  <Text> | Key = State and category Value = a  representation of  information of | Key = Text Value = Text |

|  | of a pair of state and category |  | totalsales and highest subcategory and subcat\_totalsales for a pair of  State and category |  |
| --- | --- | --- | --- | --- |

# RESULTS AND EVALUATION

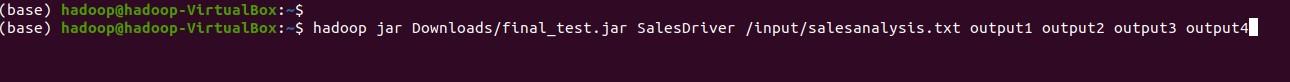
Connect to Hadoop



Creating input directory and copying salesanalysis.txt file into input folder that is Hadoop storage.



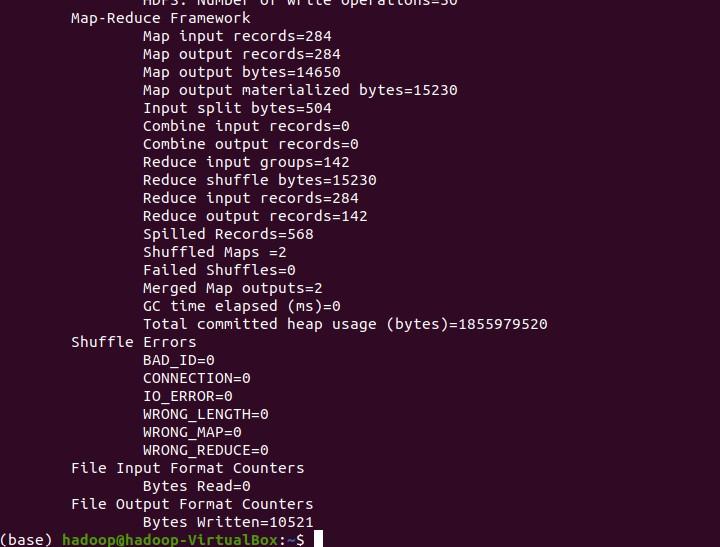
This is the code with which we will execute out .jar file to txt file :



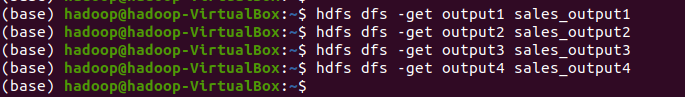
Here, “Downloads/” is a file path which contains out .jar file called final\_test.jar which contains our map reducer framework. SalesDriver is a driver class of that jar file remaining arguments are for input and output paths for framework workers(mappers-reducers). That is

* /input/salesanalysis.txt is input path for Sales\_mapper1 and Sales\_mapper2.
* output1, output2, output3 and output4 are output path for Sales\_reducer1, Sales\_reducer2, Sales\_reducer3, and Sales\_reducer3 respectively.

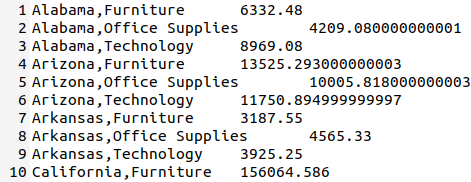
To see if our code executed successfully or not :



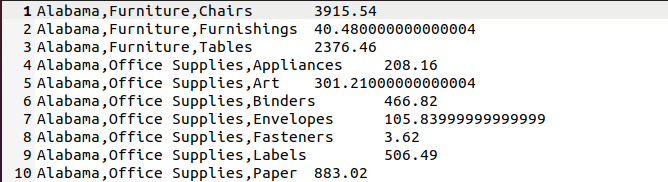
To get the outputs of all reducer from Hadoop storage to local storage we need to run this :



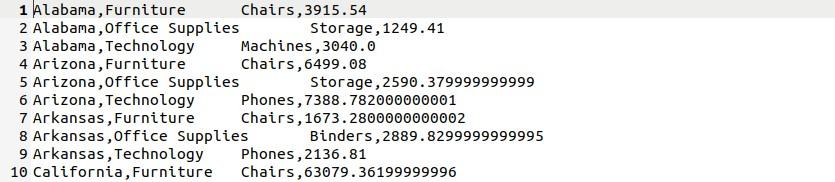
Output of Sales\_reducer1 : which gives total sales for each pair of State and Category :



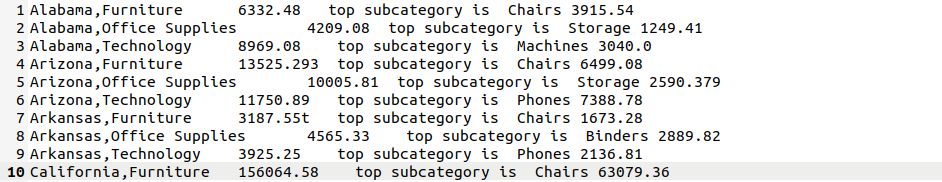
Output of Sales\_reducer2 : which gives total sales for each pair of State ,Category and Sub\_category:



Output of Sales\_reducer3 : which generates pair of totalsales and subcategory while keeping of State and Category :



Output of Sales\_reducer4 : which gives total sales and subcategory which had contributed the most in those sales for each pair of State and Category :



Explanation of output of all phases with one example:

SALES\_MAPPER**1** SALES\_M**APPER**2

| **KEY = ALABAMA, FURNITURE** | Key = Alabama, Furniture, chairs |
| --- | --- |
| **VALUE = 2804** | Value = 800 |

# SHUFFLE & SORTING

| **KEY = ALABAMA, FURNITURE VALUE = {2804, 600,2008, …}** | Key = Alabama, Furniture, chairs Value = {200, 1256, 456, …} |
| --- | --- |

**PARALLEL\_REDUCER\_2** SALES**\_**REDUCER2

| **KEY = ALABAMA, FURNITURE** | Key = Alabama, Furniture, chairs |
| --- | --- |
| **VALUE = 6332.48** | Value = 3915.54 |

SALES**\_**M**APPER**3



# SHUFFLE & SORTING

**SALES\_REDUCER3**

SALES\_MAPPER4 SALES\_MAPPER5

| **KEY = ALABAMA, FURNITURE** | Key = Alabama, Furniture |
| --- | --- |
| **VALUE = 6332.48** | Value = (Chairs, 3915.54) |

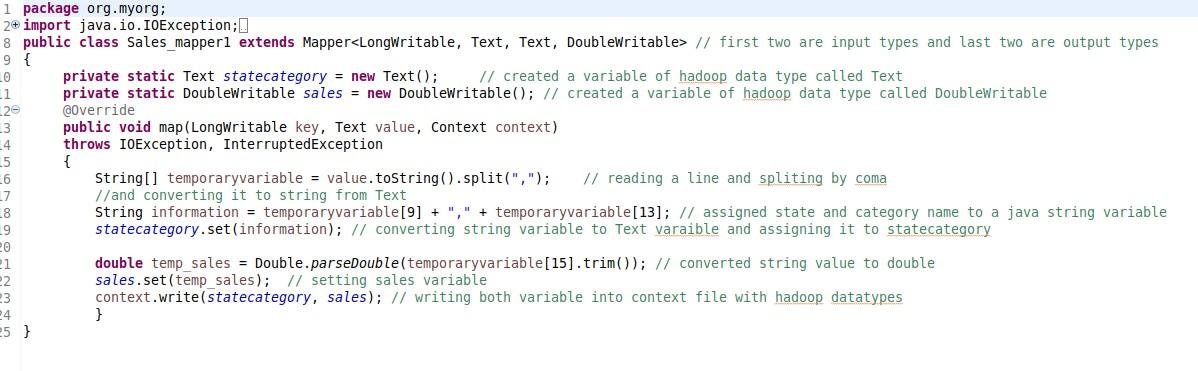


# SHUFFLE & SORTING

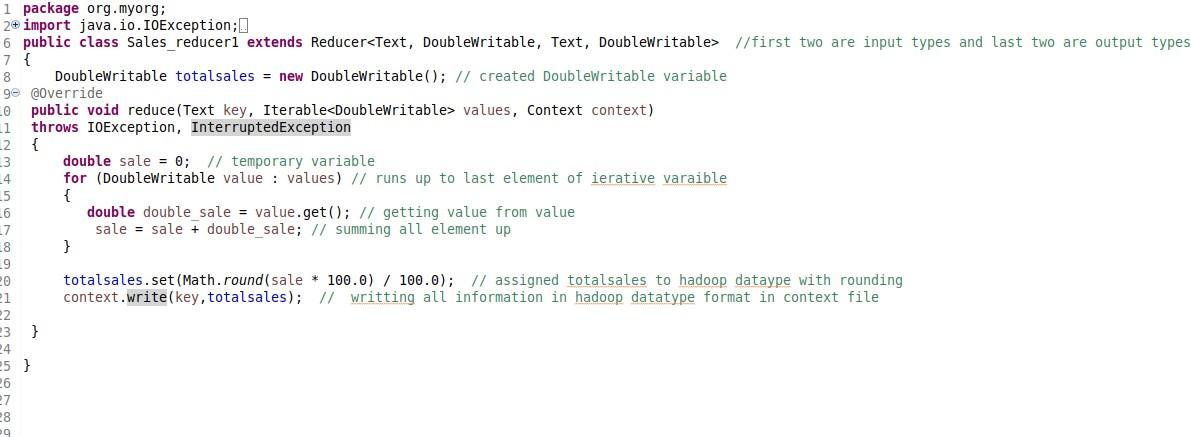


**SALES\_REDUCER**4

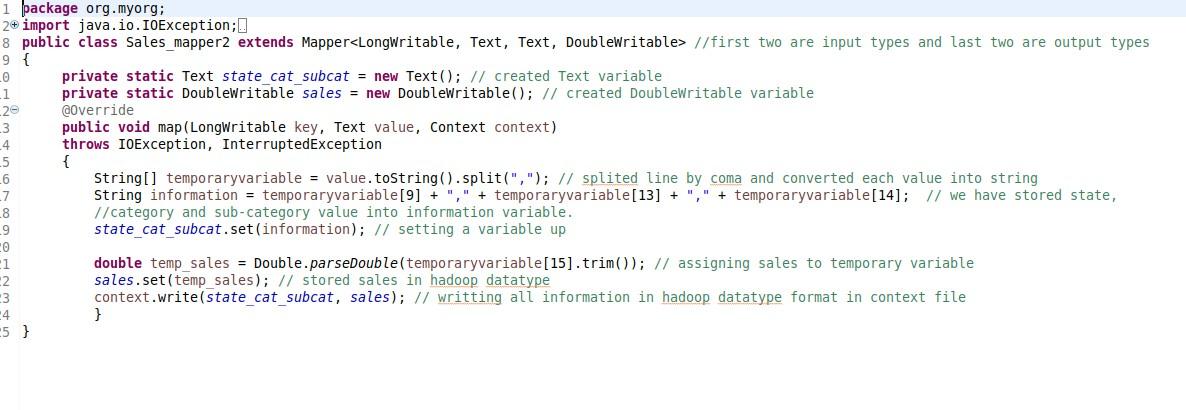
# Sales\_mapper1



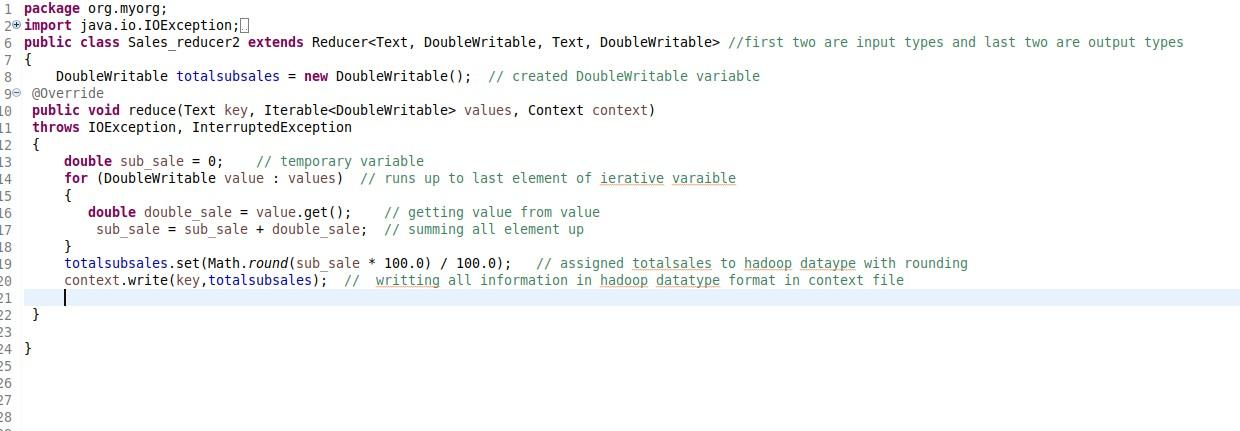
**Sales\_reducer1**



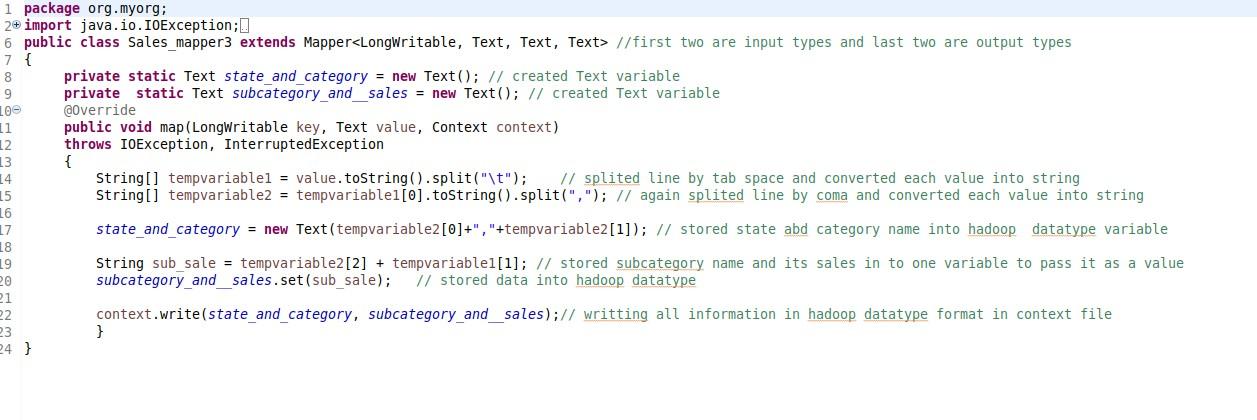
# Sales\_mapper2



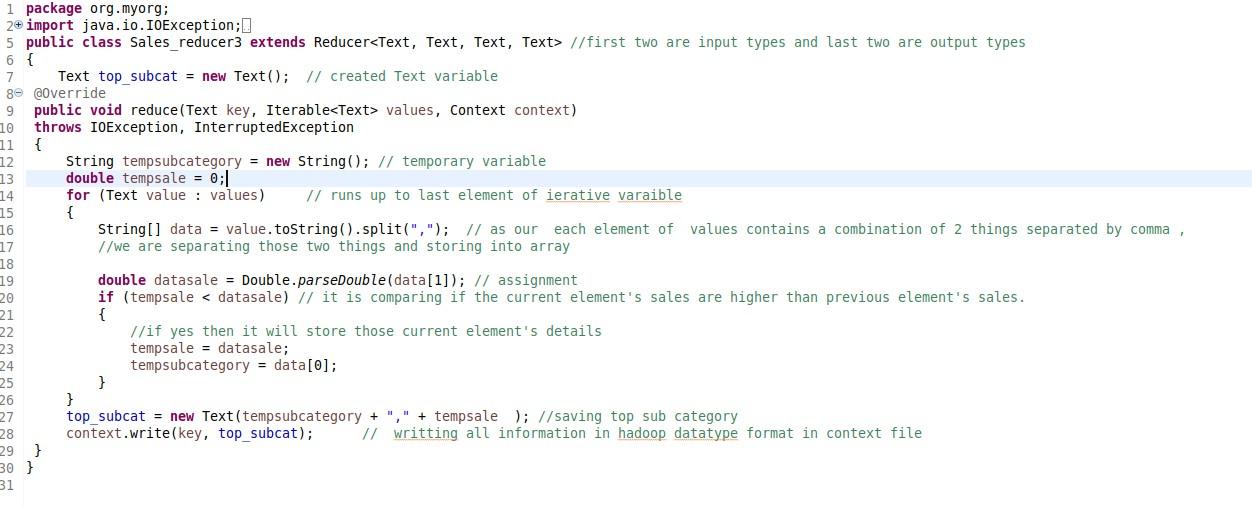
**Sales\_reducer2**



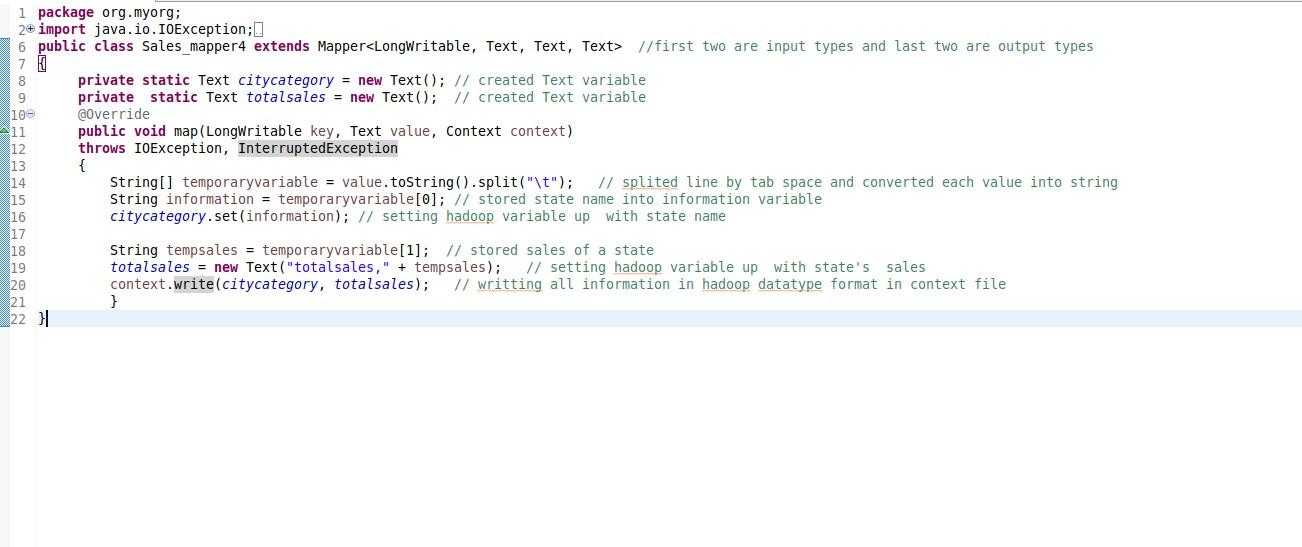
# Sales\_mapper3



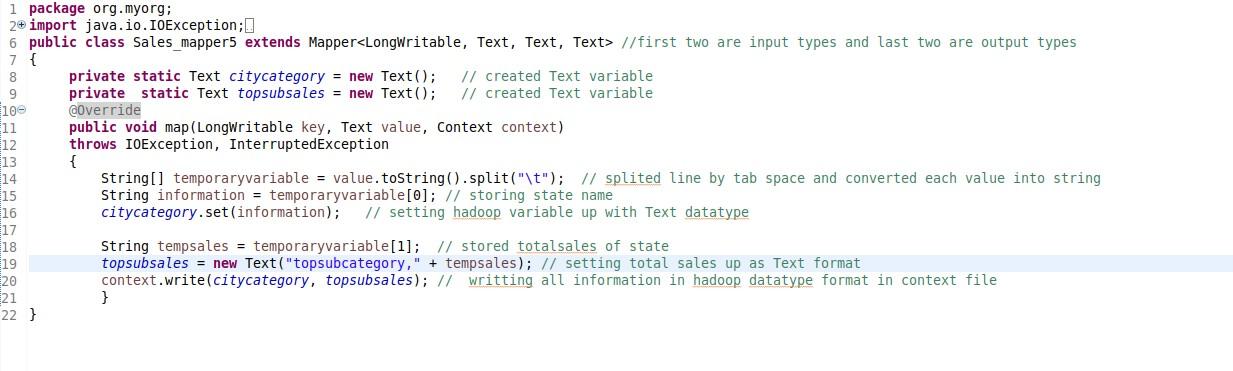
**Sales\_reducer3**



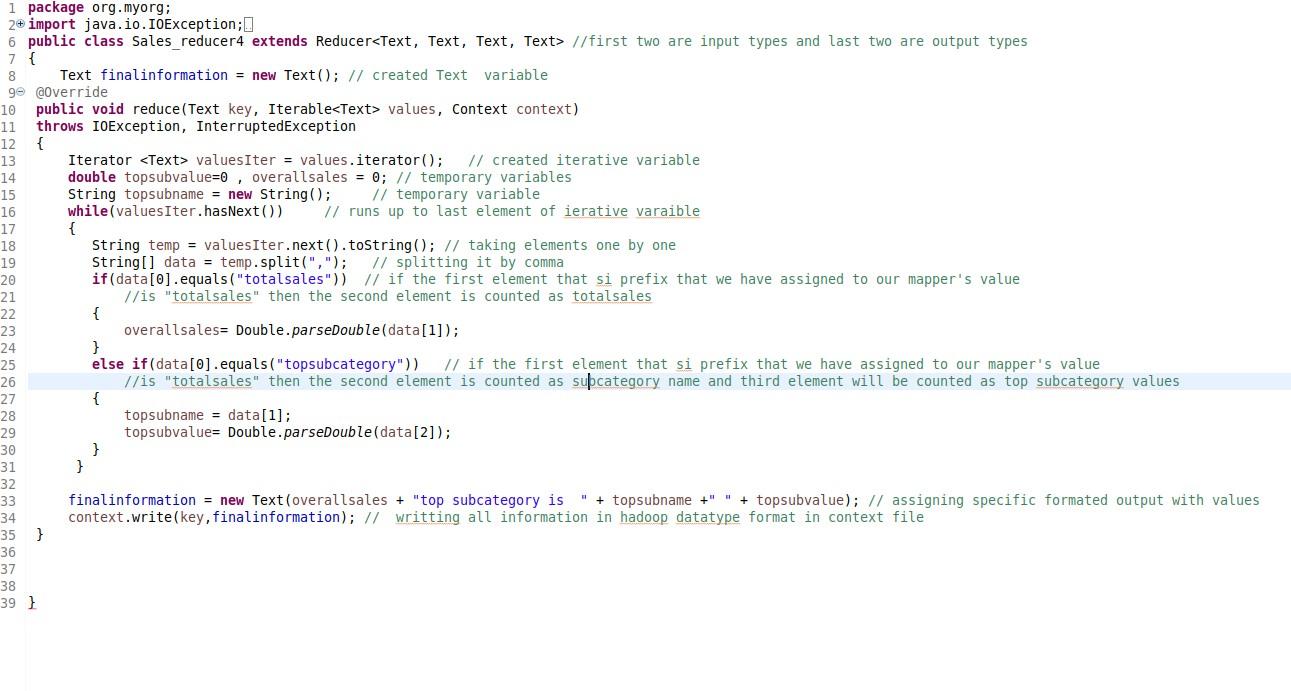
# Sales\_mapper4



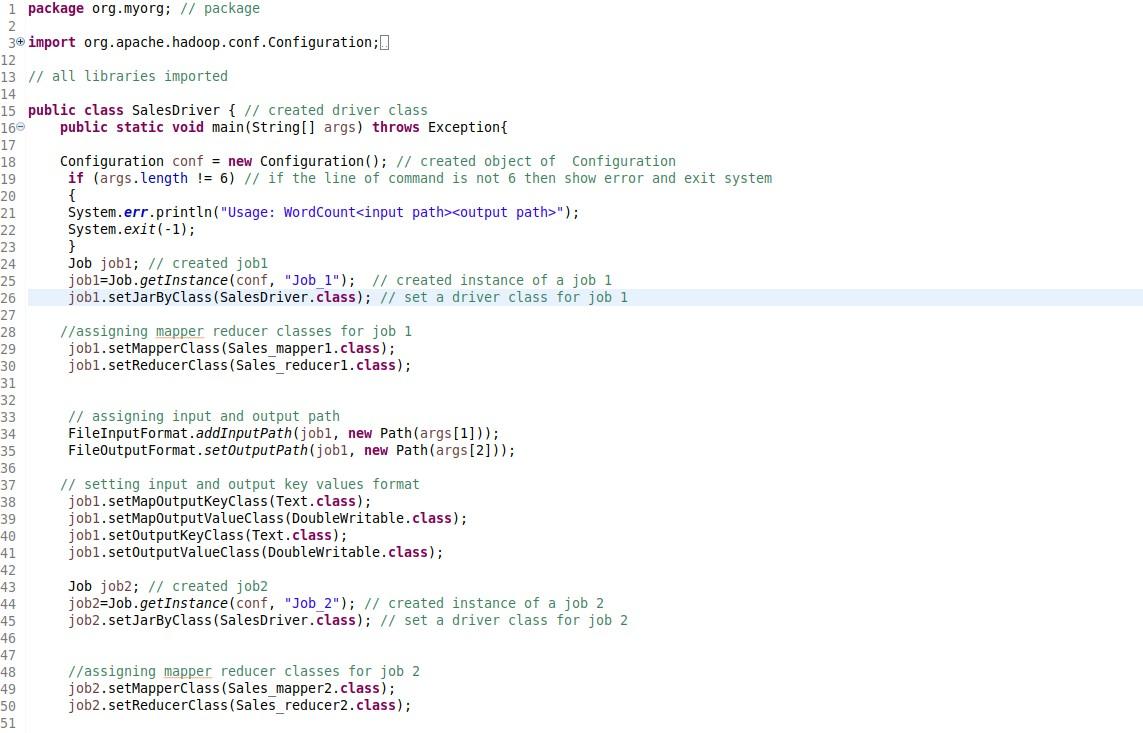
**Sales\_mapper5**

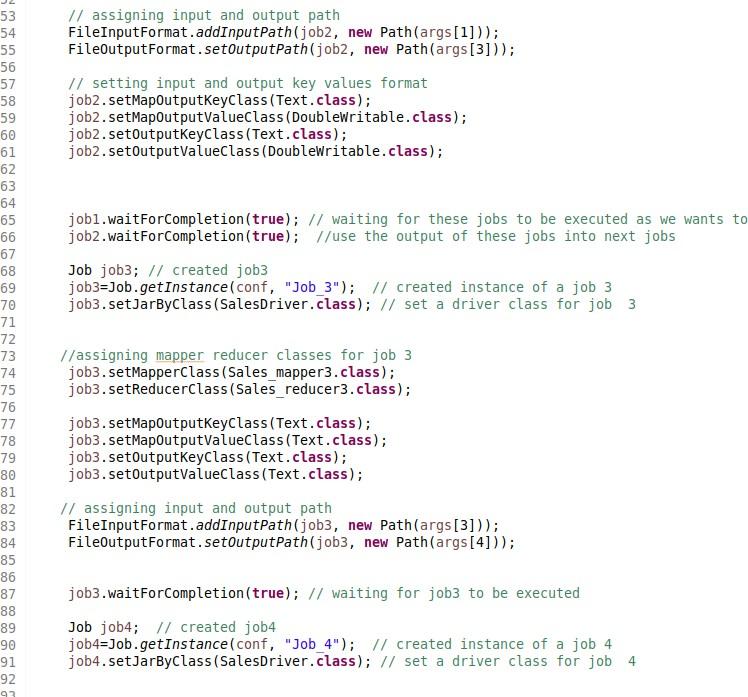


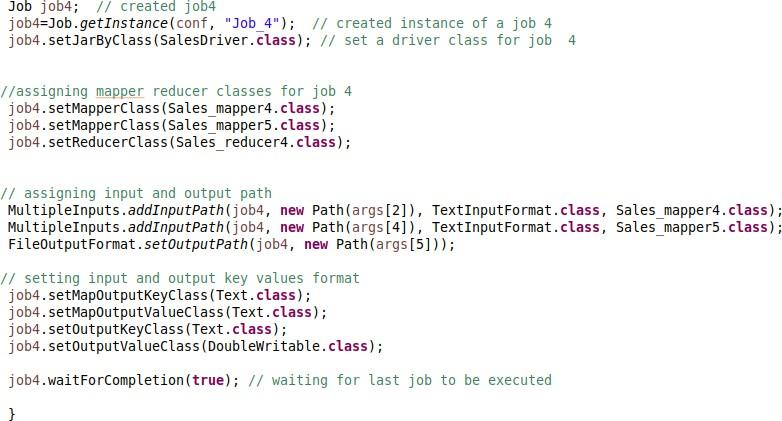
# Sales\_reducer4



**SalesDriver**







# Visual Representation:

My results are matching with the visualisations of tableau which represents the Albama state with category furniture and subcategory of chair with total sales 6332.