Lab-1 CSEE 5590 BIG DATA PROGRAMMING

Team members: ID-4

1. Nagababu chillukuri- 4
2. Grace stalin - 30
3. Pavan -

**1.Hadoop MapReduce Algorithm**

Implement MapReduce algorithm for finding Facebook common friends problem and run the MapReduce job on Apache Hadoop. Show your implementation through map-reduce diagram as shown in Lesson Plan 2.

**2.Use Case: Counting and Summing**

**3. Consider one of the following use cases,**

1. Zomato Restaurants Data

<https://www.kaggle.com/shrutimehta/zomato-restaurants-data>

1. Super Heros Dataset

<https://www.kaggle.com/claudiodavi/superhero-set/data>

1. Google Job Skills Dataset

<https://www.kaggle.com/niyamatalmass/google-job-skills/data>

1. Seinfeld Chronicles Dataset

<https://www.kaggle.com/thec03u5/seinfeld-chronicles/data>

***HIVE USECASE***

1. Create a Hive Table including Complex Data Types
2. Use built-in functions in your queries
3. Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box.

***SOLR USECASE***

* 1. Create a Solr Collection including our own Field Types
  2. Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box. Implement at least 5 nested queries among the 10.
  3. Record the time execution for the queries.

**Objective:**

1.Implement Map Reduce algorithm for finding Facebook common

Friend’s problem and run the Map Reduce job on Apache Hadoop.

Show your implementation through map-reduce diagram.

Approach:

Let us take the use case example given in problem statement to implement Map Reduce approach.

Given input is the user and friends list User --> [friends List]

Input:

10 20,30,50

20 30,40,10

30 10,20

Step 1: The Map Reduce approach will divide this into different phases as described below:

Map Phase: Using above input Split Mapper will create key, value pair.

**Combiner/Shuffle Sort Phase:** This phase will group them by their keys before sending it to reducer:

Step1:

10,20 20,30,50

10,30 20,30,50

10,50 20,30,50

20,30 30,40,10

20,40 30,40,10

10,20 30,40,10

10,30 10,20

20,30 10,20

Step 2: Reducer Phase: The reducer will take all the grouped key and output the same key with the result of the intersection.

(10,20) -> (20,30,50) (30,40,10)

(10,30) -> (20,30,50) (10,20)

(10,50) -> (20,30,50)

(20,30) -> (30,40,10) (10,20)

(20,40) -> (30,40,10)

Output: Result after reduction:

(10 20) -> (30)

(10 30) -> (20)

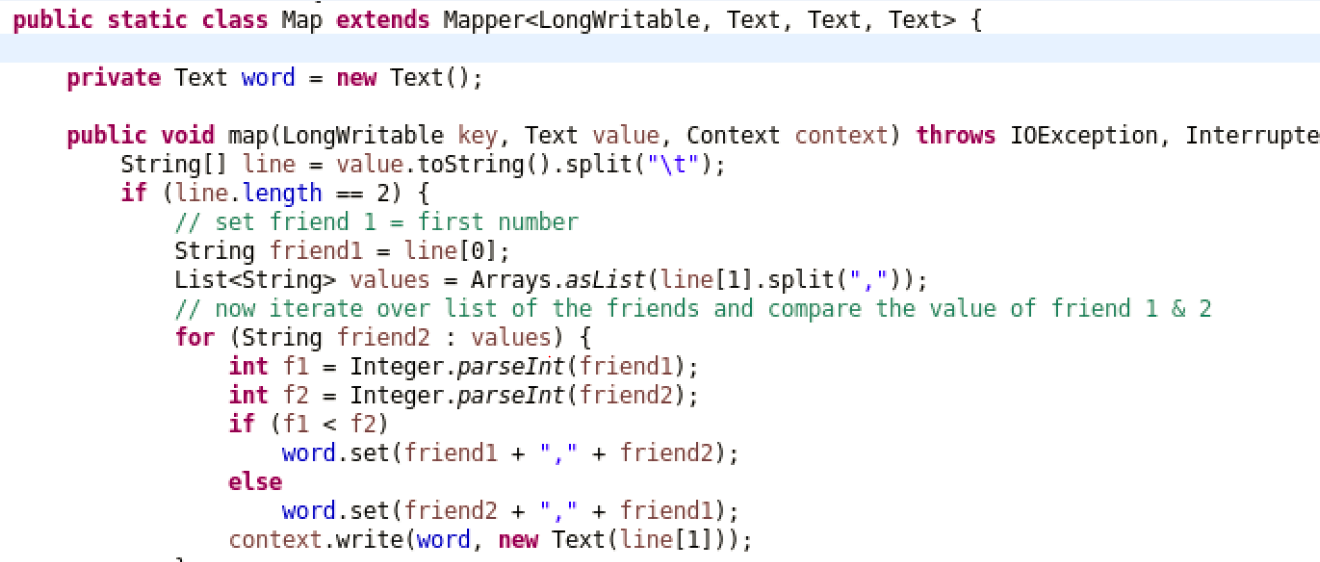
(10,50) ->

(20 30) -> (10)

(20,40) ->

WorkFlow:

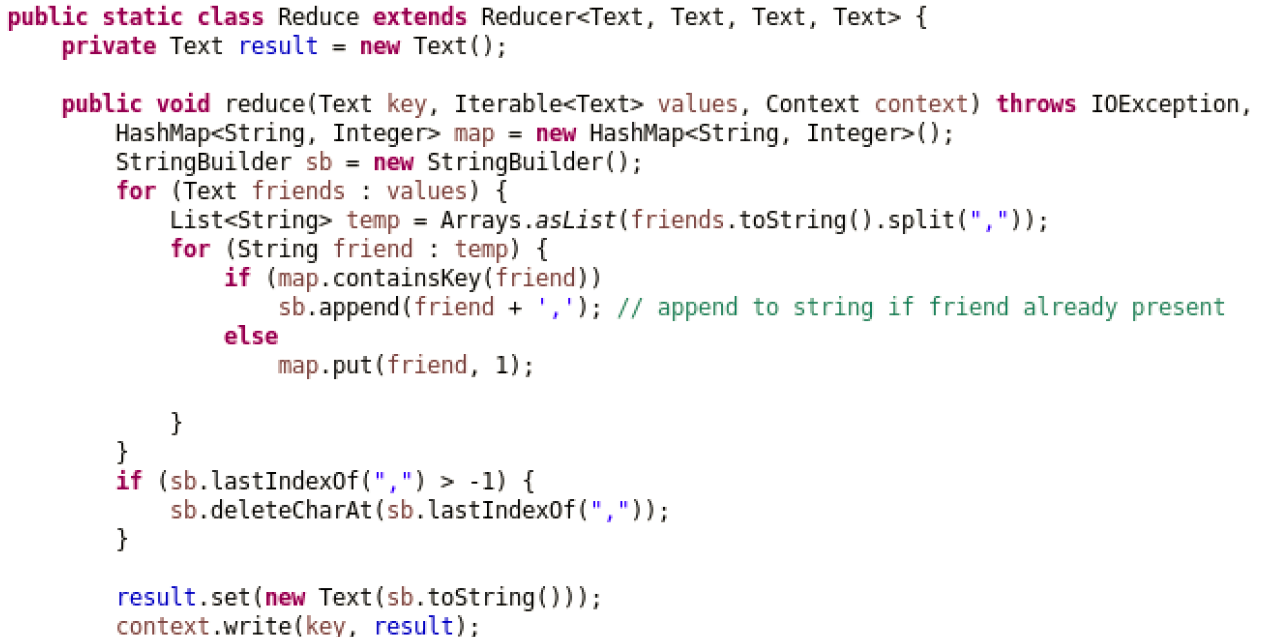
Create a mapper class as shown in the code snippet below. Each line of the input file split based on “tab” delimiter. Then its length computed as two, where the first part is source or base user and the rest of the split considered as list of friends of the user. Then the keys are prepared as (A, B) or (B, A) based on the integer values of A & B in the input.



Create a reducer class where the data grouped based on the key values

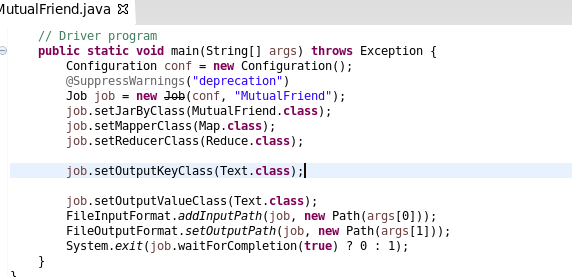
(A, B) or (B, C) and their list of friends as produced. Then finally reduced to

Find the mutual friends of (A, B).



A main method which acts as a driver to set mapper and reducer class

Which takes the input and produces the output.



**Data set and Parameter**

Please find input file and Output File at

[www.github](http://www.github)

Mapper split

(10,20) -> (20,30,50) (30,40,10)

(10,30) -> (20,30,50) (10,20)

(10,50) -> (20,30,50)

(20,30) -> (30,40,10) (10,20)

(20,40) -> (30,40,10)

(10 20) -> (30)

(10 30) -> (20)

(10,50) ->

(20 30) -> (10)

(20,40) ->

Algorithm:

10,20=20,30,50

10,30=20,30,50

10,50=20,30,50

Input splits

10 20, 30, 50

20,30 =30,40,50

20,40=30,40,50

10,20=30,40,50

10 20,30,50

20 30,40,10

30 10,20

20 30, 40, 10

10,30=10,20

20,30=10,20

30 10,20

Final output

Combiner/shuffle sort

**Objective**

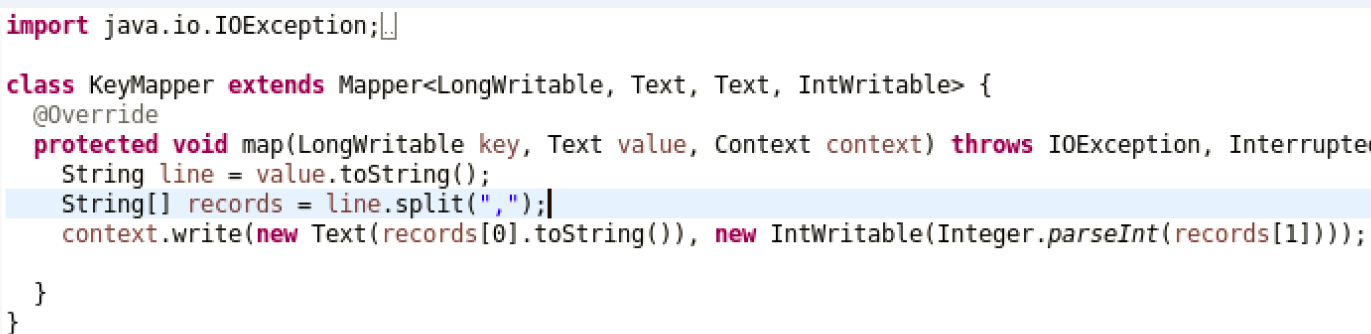
1. Use Case: Counting and Summing:

Approach:

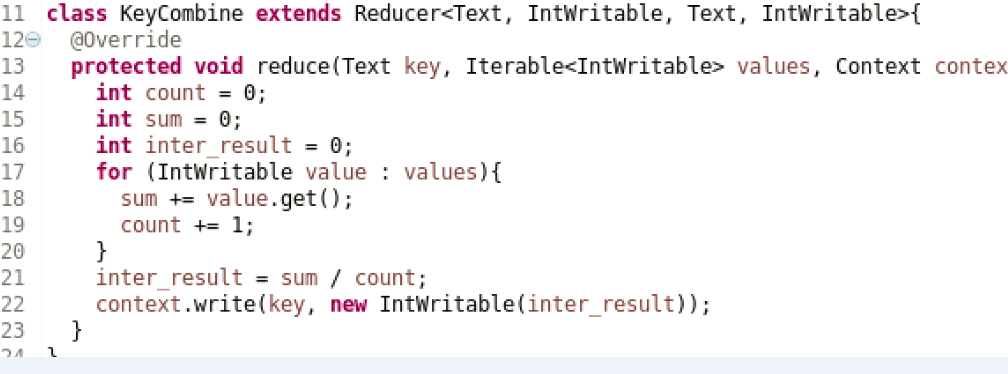
In order to find the average of each key given in input file the logic is quite simple: If all the number have the same key, then the mapper sent all the values you want to find the average of with that same key. Because of this, in the combiner (mini reducer) and reducer you can sum the values in the iterator. You can then keep a counter on number time the iterator works, which solves the issue of how many items, are to get average. Finally, after the iterator, you can find the average by dividing the sum by the number of items.

Workflow:

Create a mapper class as shown in the code snippet below. Each line of the input file split based on “,” delimiter where the first part is key (Node) and second part is numeric value of response time. Then in map the keys are prepared as (N1 10, 20) or (N2 20, 20) (N3 30, 30).



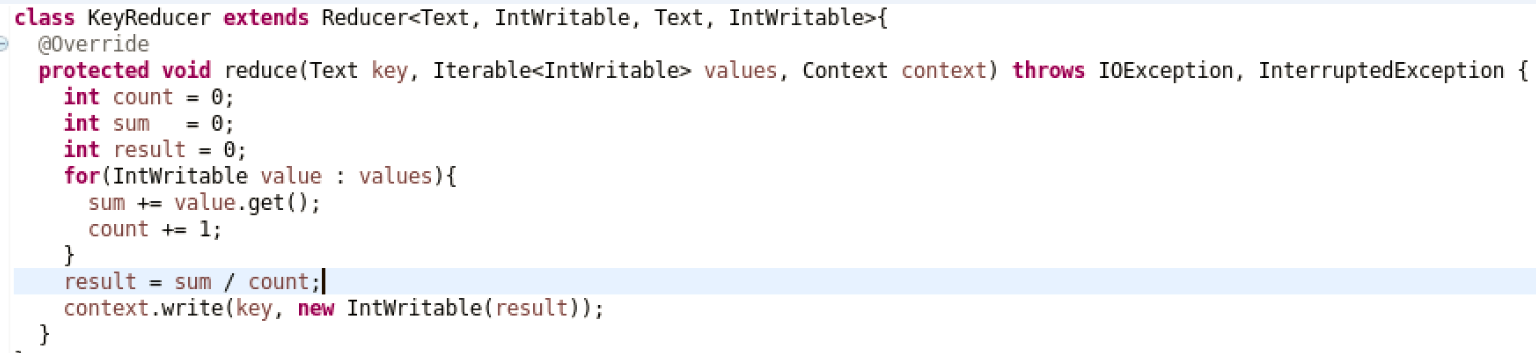
Create a Combiner also called mini reducer average, count, sum calculated based on the key values. The combiner step will decreases the amount of data that need to be processed by the reducer. Combiner improves the overall performance of the reducer.



The reducer will need to process only few of key value pair and generate output.

A main method, which acts as a driver to set mapper, combiner and reducer class

Which takes the input and produces the output.



**Data set and Parameter**

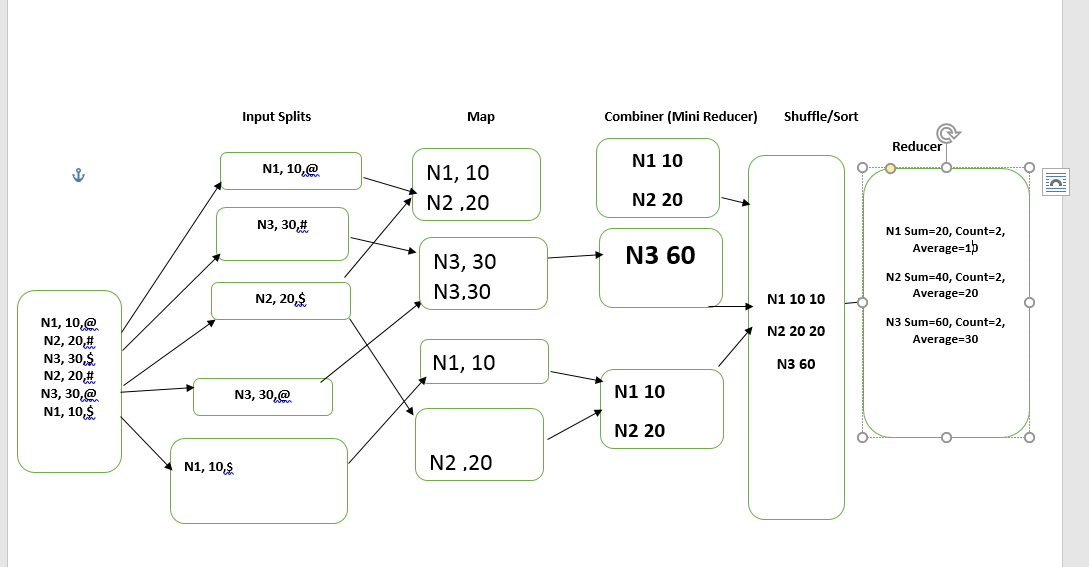
Please find input file at below location:

[www.github](http://www.github)

**Evaluation**

Hadoop map reduce with combiner is very efficient in finding the average of key given in input file. We can say that Map Reduce Combiner plays a key role in reducing network congestion. Map Reduce combiner improves the overall performance of the reducer by summarizing the output of Mapper.

Algorithm:



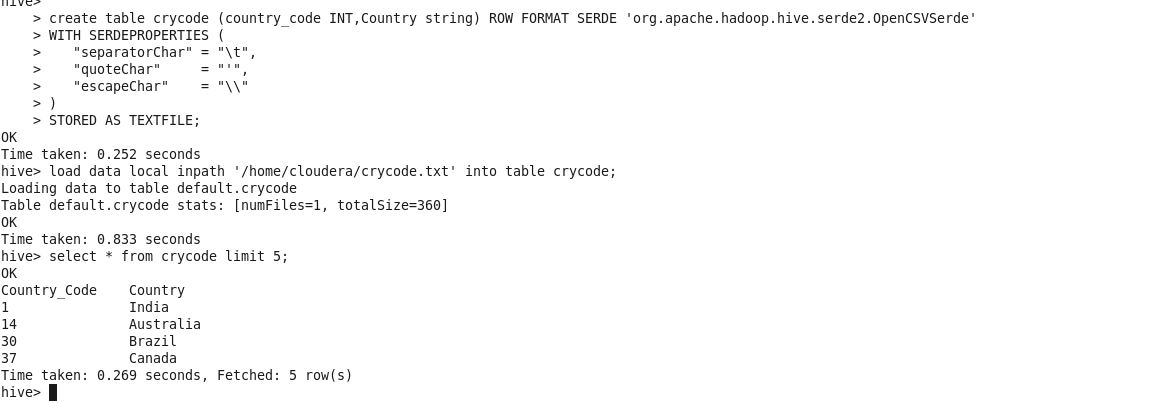
1. Create a Hive Table including Complex Data Types:

1.Creating hive table using zomato with complex data types as follows:



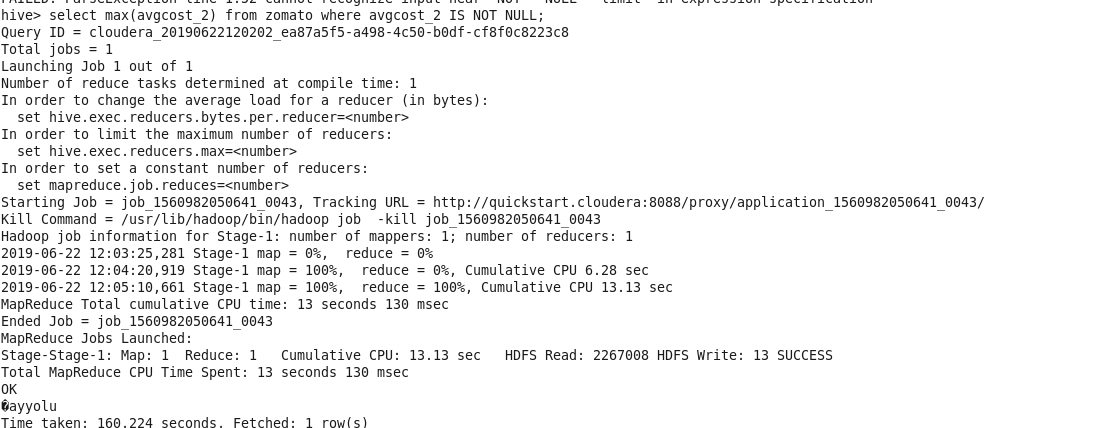
Here table is created using comples datatypes with serde properties to handle quotes and comma delimiter in zomato dataset.

2. Creating second table and upload it in the hive as follows:

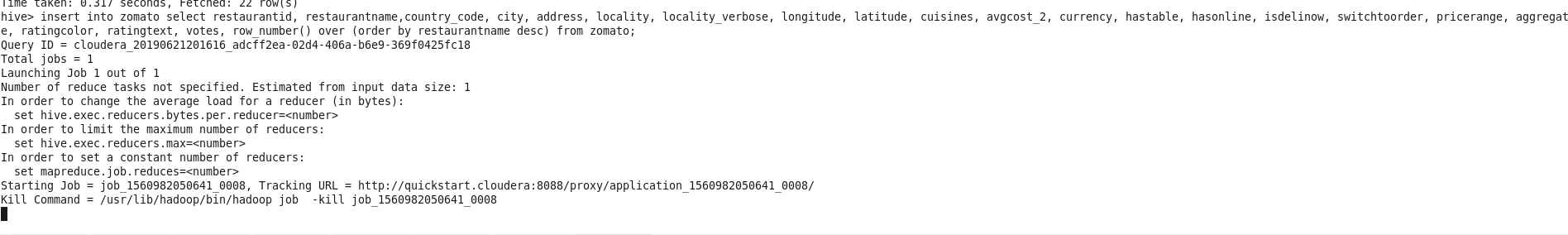


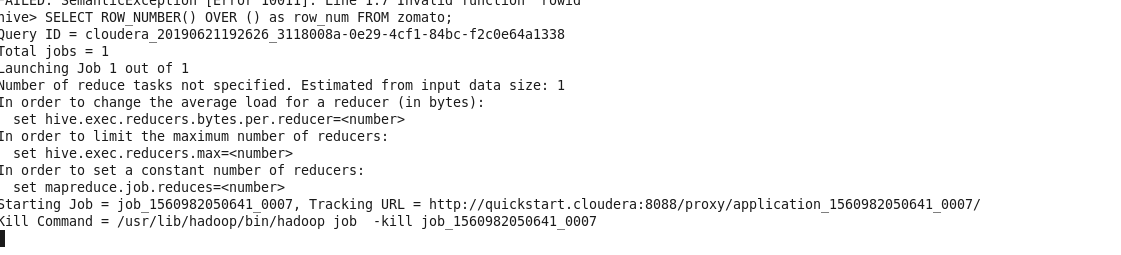
B.Use built-in functions in your queries:

1.Max() and row\_number() are the 2 built in functions we have used in hive:

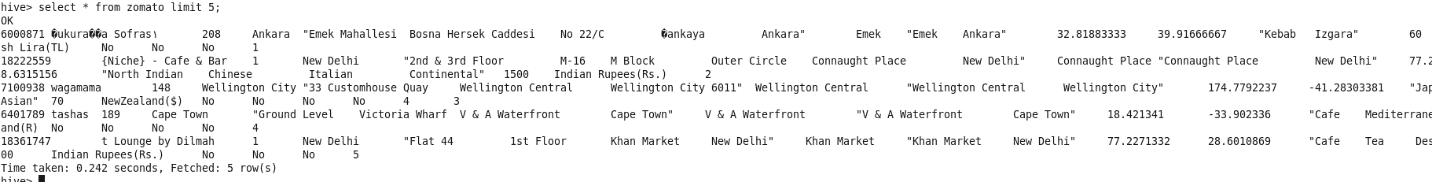
****

2.The row\_number() function is useful while handling sql query in database:



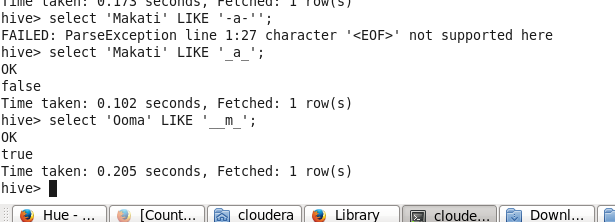


The rowid is column is inserted at the end of the column as follows:

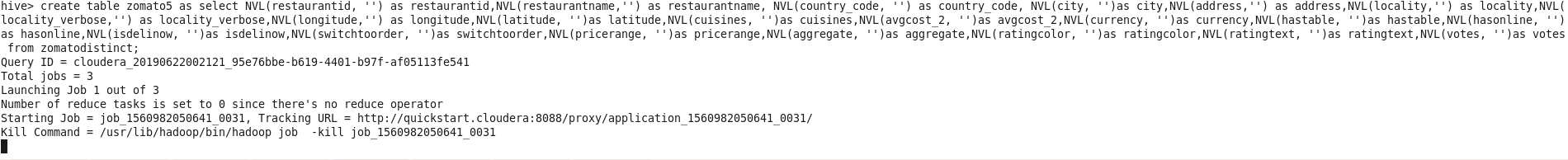


C.Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box.

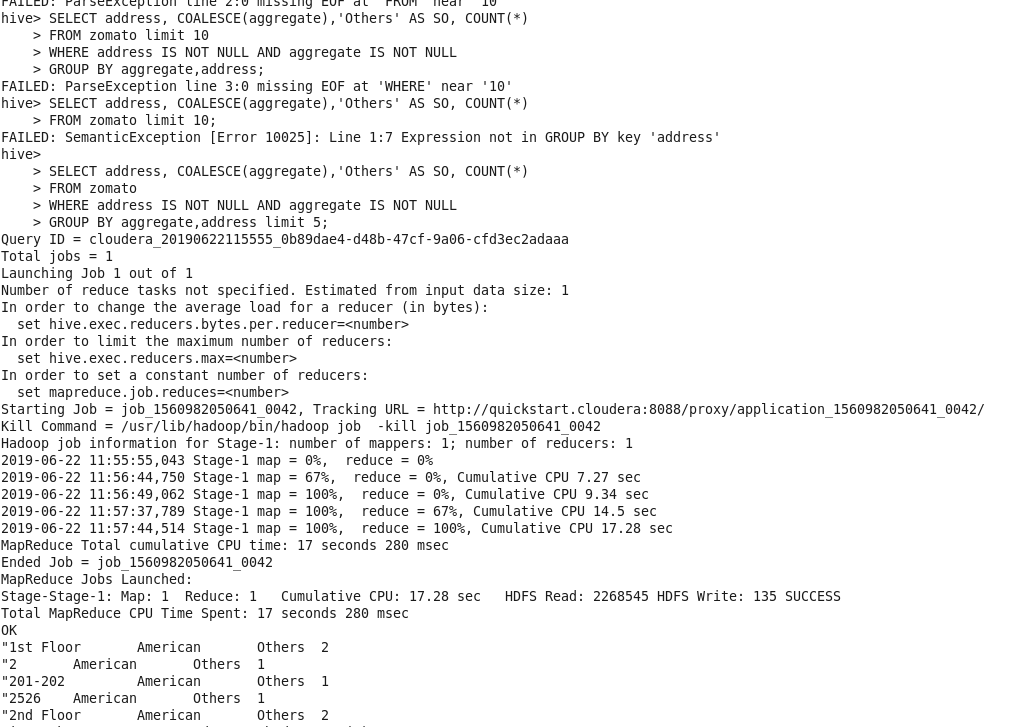
3. Pattern recognition is done as follows:



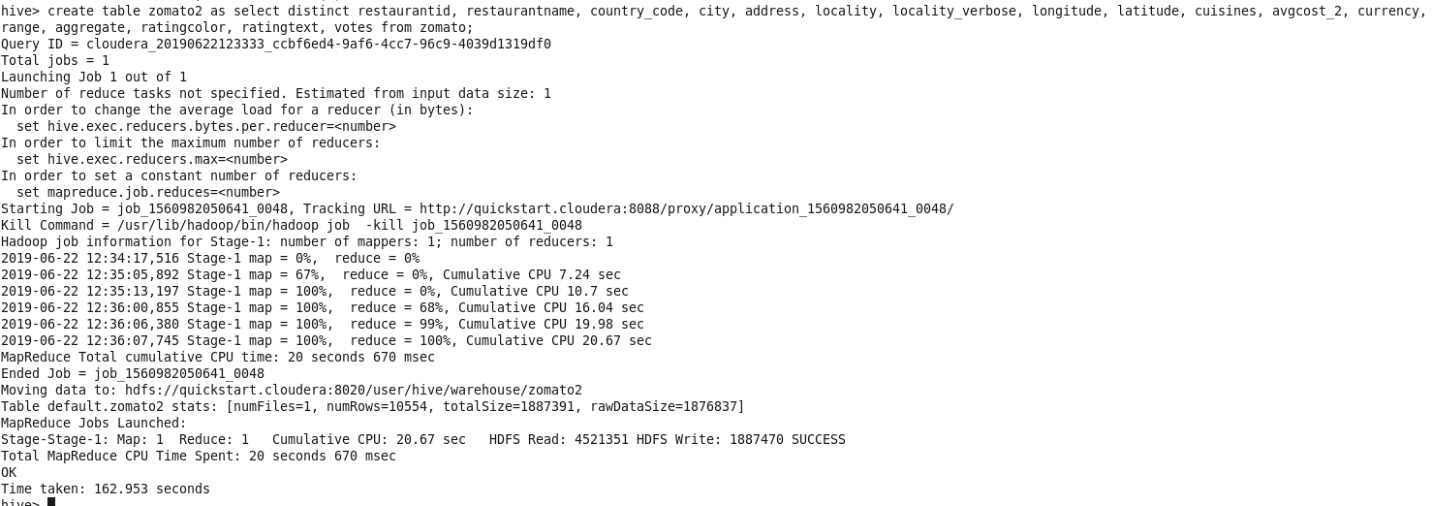
4. Nullvalue deletion in the hive table:



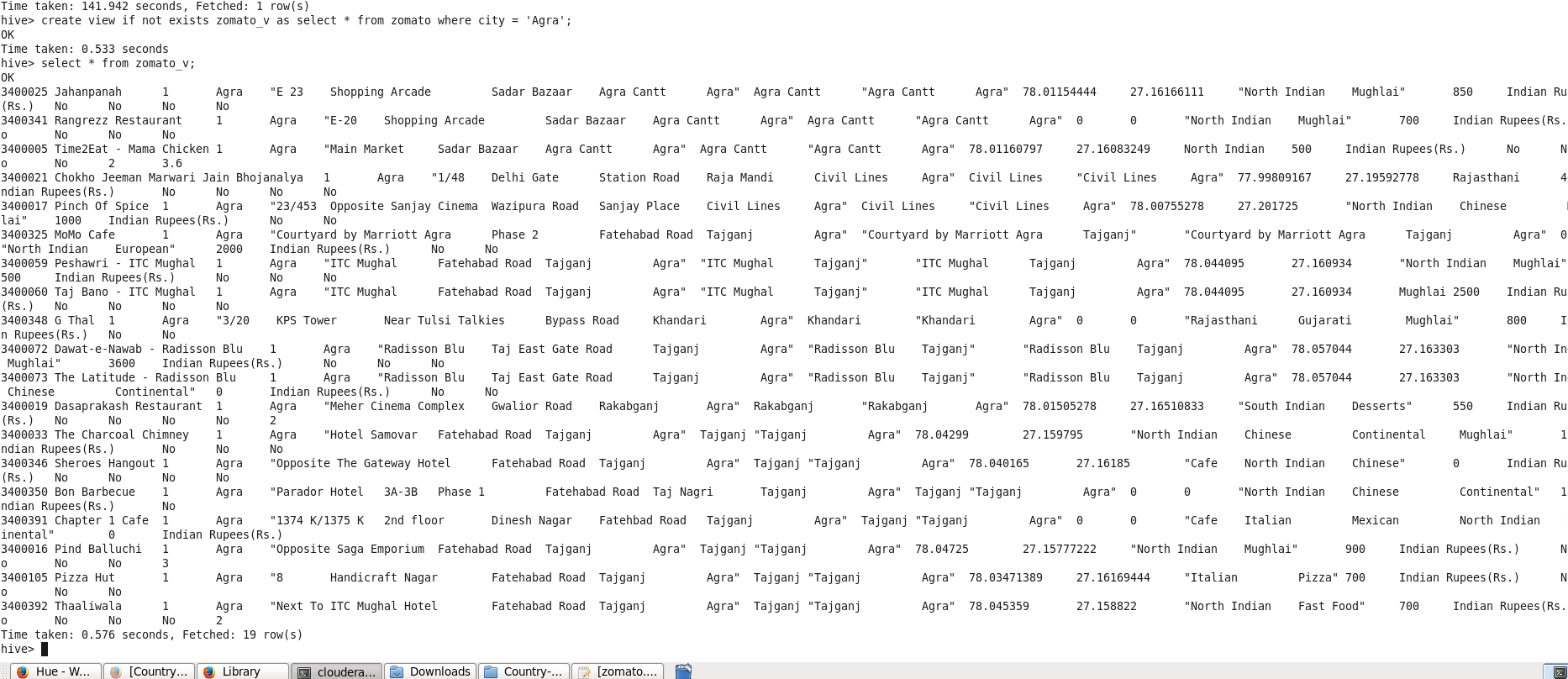
5.By selecting the notnull values and creating the rownumber we will delete most of the null values inside the table while doing the table cleaning process:



6.By creating the distinct table so that the duplicate values are deleted in the table which is also a cleaning process:

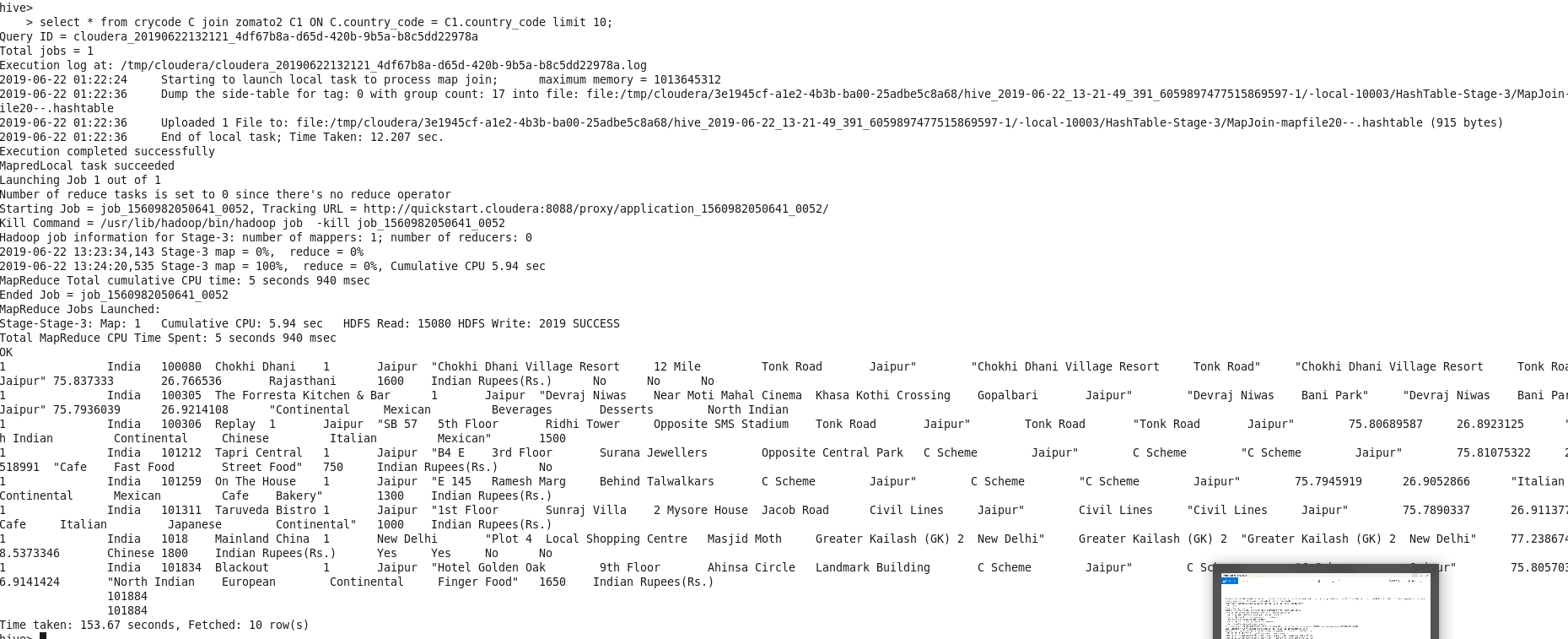


7.Creating views in a database is always known for hiding certain personal information so that everyone cannot see the hidden view in a database.

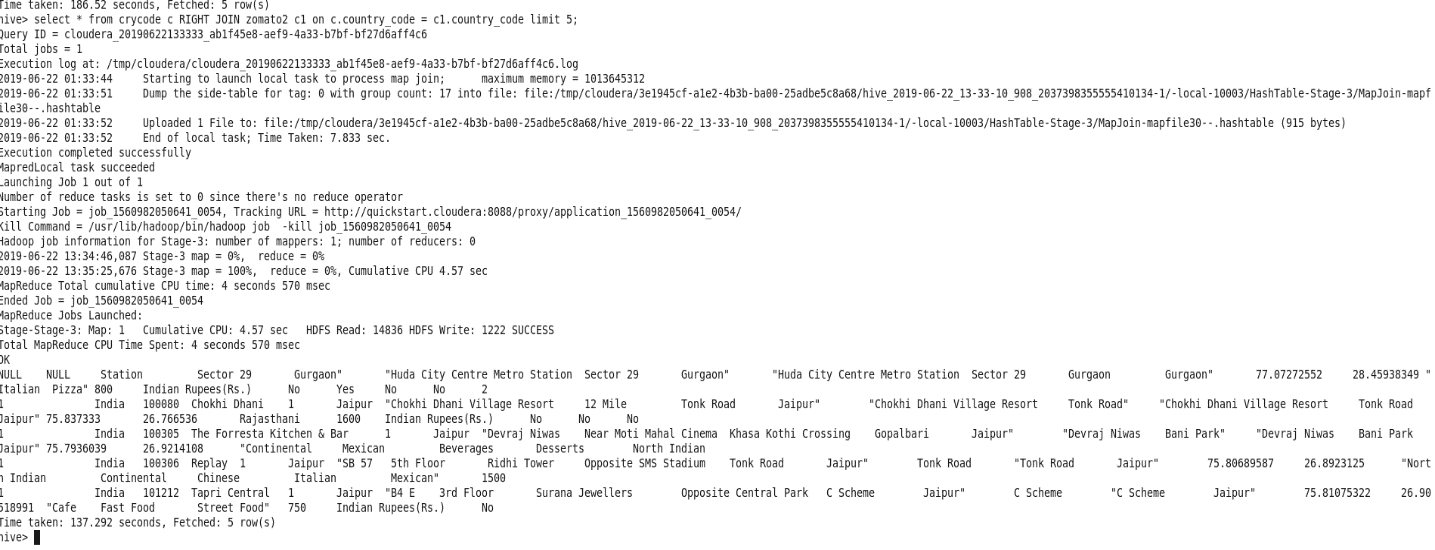


8.AS in the columns in the dataset are similar hence we perform joints as follows:

1.Equijoin:



2.Right outer join:



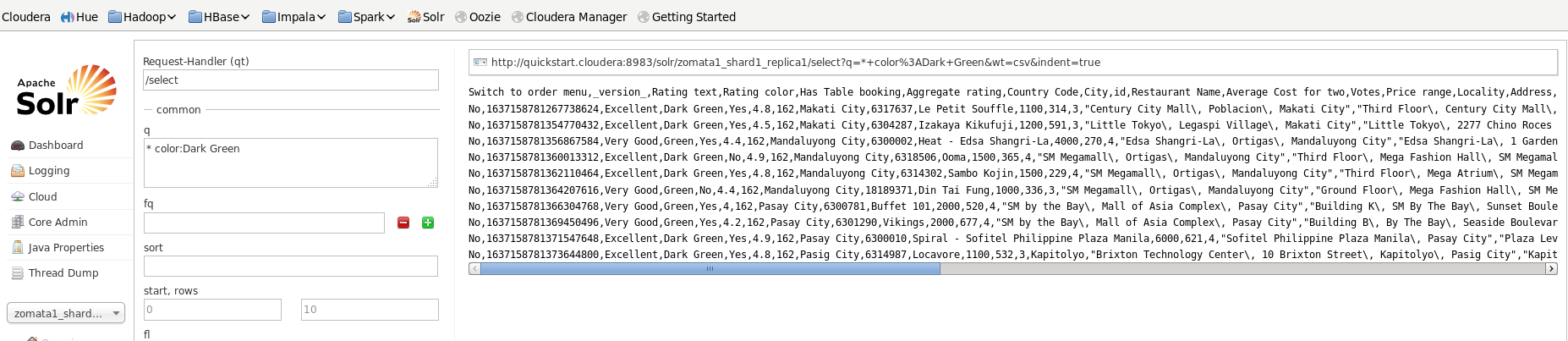
Thus using the dataset of 9558 rows and 22 column including rowid we have cleaned the dataset and got the output for all important queries that is necessary for the corporate culture.

***SOLR USECASE***

* 1. Create a Solr Collection including our own Field Types:

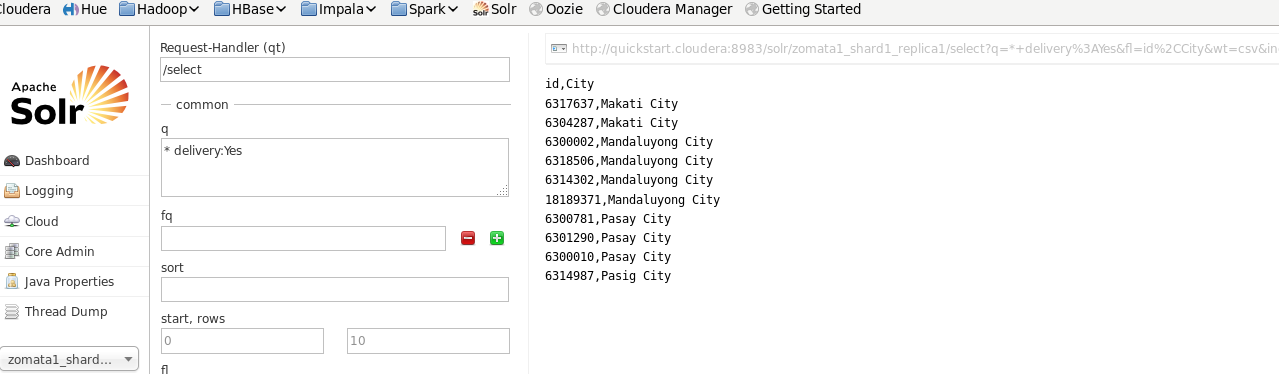
Taking Zomato restaurant:

<https://www.kaggle.com/shrutimehta/zomato-restaurants-data>.



2. Number of rows with range two people 100 to 1000:





.