**What we have done:**

**Implementation:**

**Step 1:** Installation of Cloudera CDH3 virtual platform with VM Player.



**Step 2: Then we have done demo for our understanding of the system.**

Here is the files that we inserted in to the HDFS:

****

Sample Text File demo.txt contains the set of statements as below:



Then we load this file into HDFS:

****

After that we apply Map task on the file to get the word count of a particular word. The source of Map program is http://www.cloudera.com/content/cloudera/en/documentation/HadoopTutorial/CDH4/Hadoop-

Tutorial/ht\_wordcount1\_source.html

The detailed execution of Map task is shown below:

****

The result of the Map task is as follows:

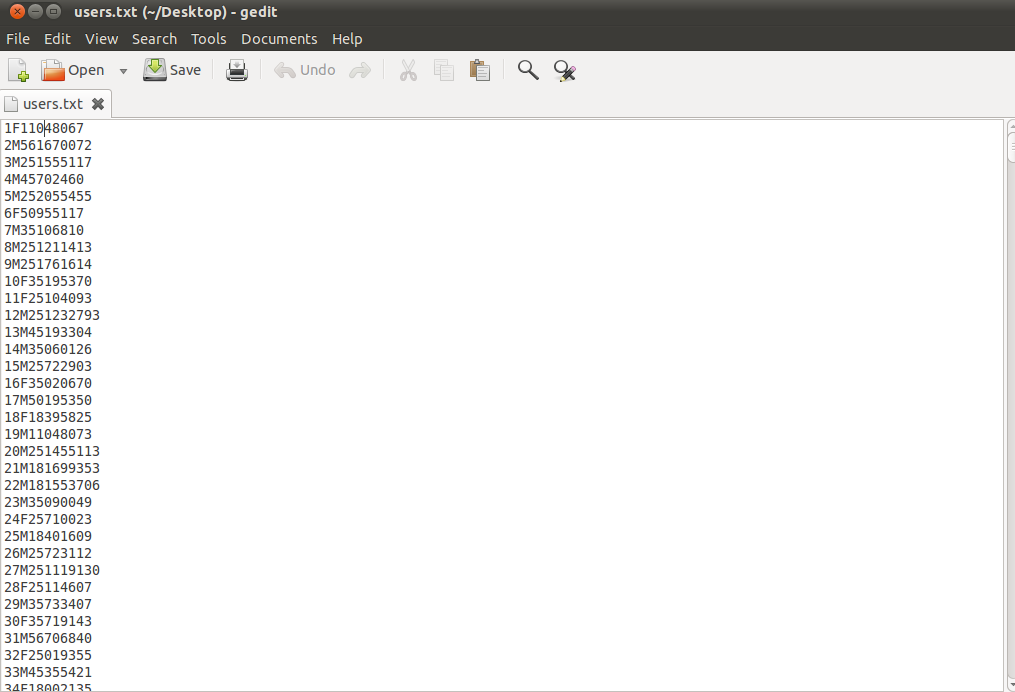
****

**Step 3:** Now we started using the system according to our requirements.

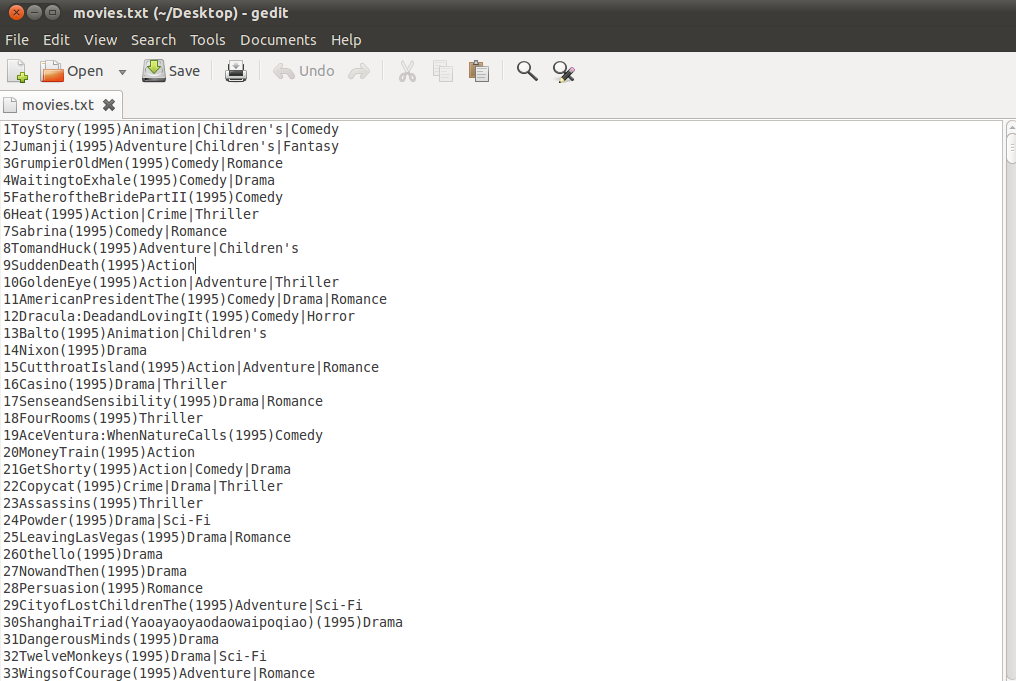
At first, we transfer our datasets from our local machine to Cloudera’s VM through FileZilla FTP client.

And then we stored this unstructured data files in to the HDFS.

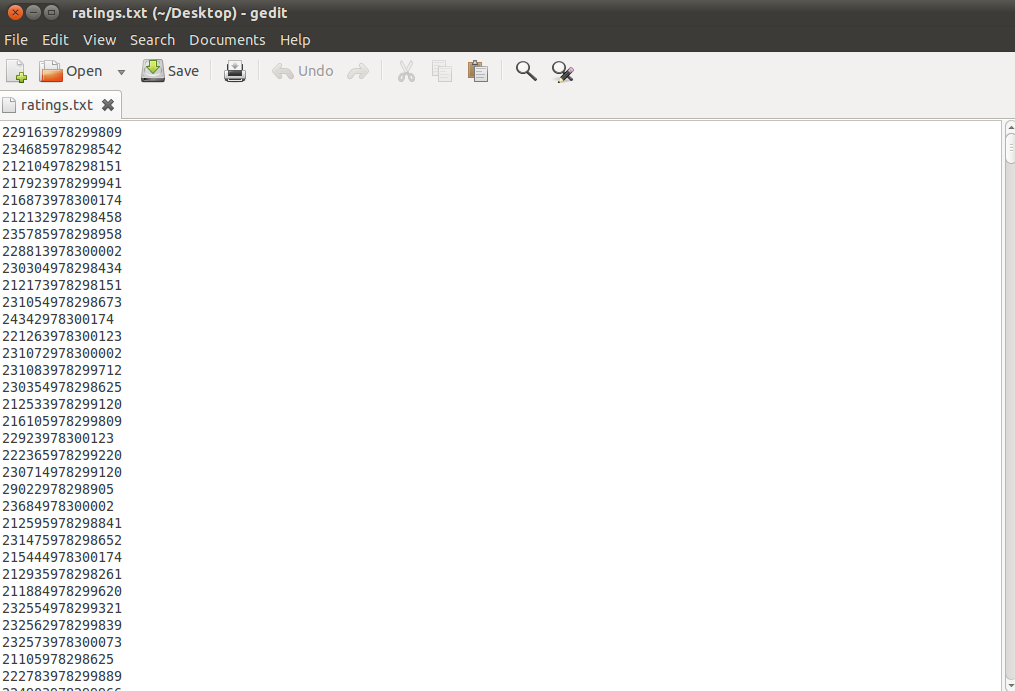
1. User.txt



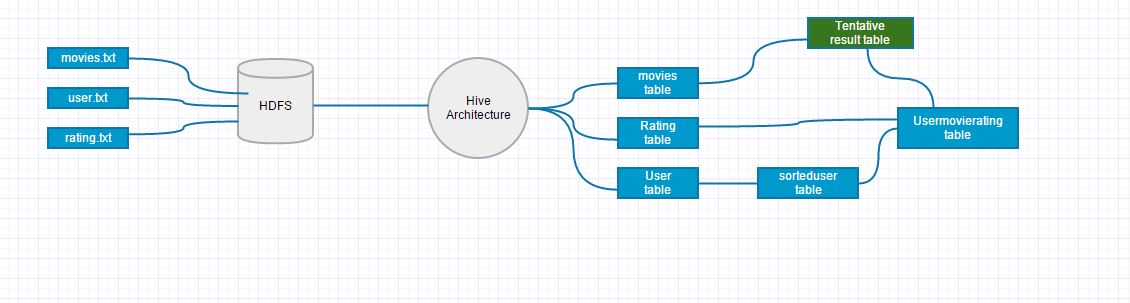
2. Movies.txt



3. Rating.txt

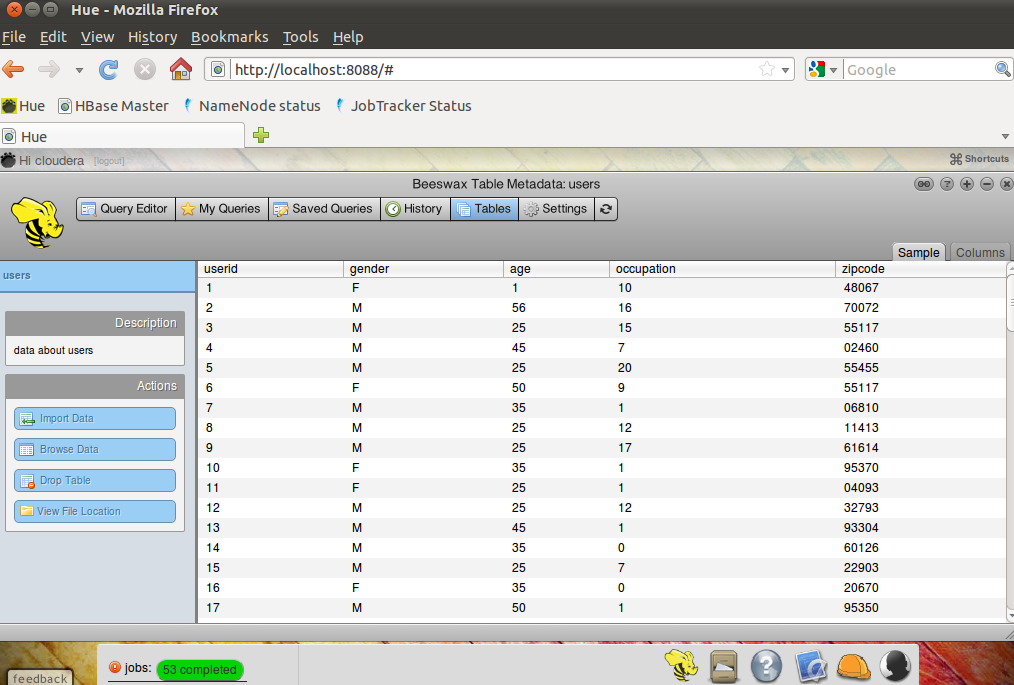


**Here is the Data flow of the Overall Project:**

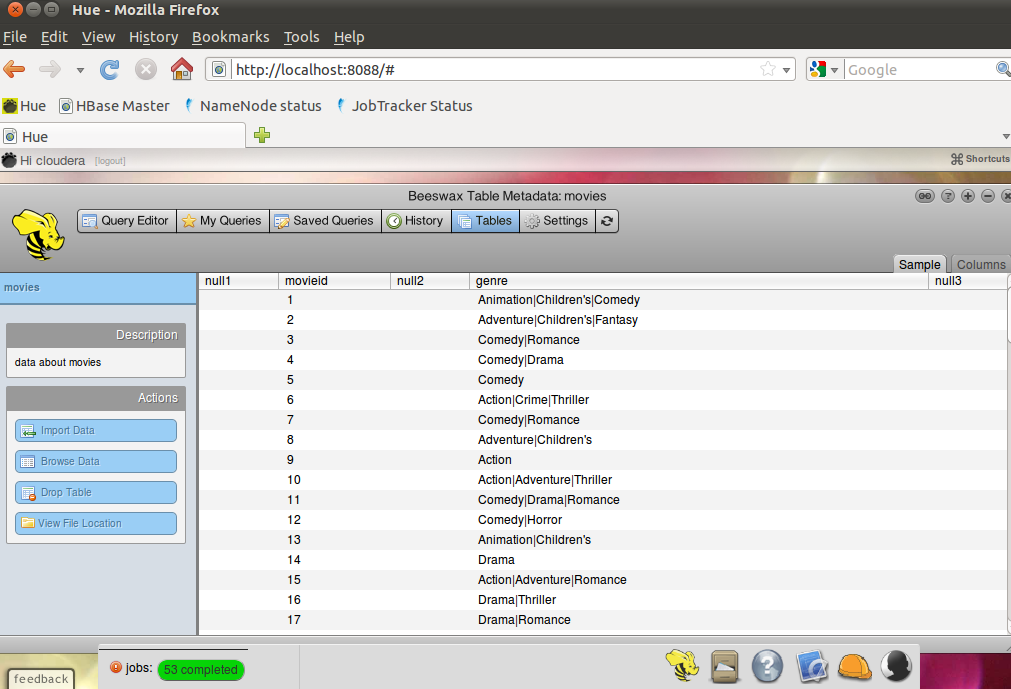
****

According to the data flow we proceed and convert the unstructured datasets in to a structured format like tables.

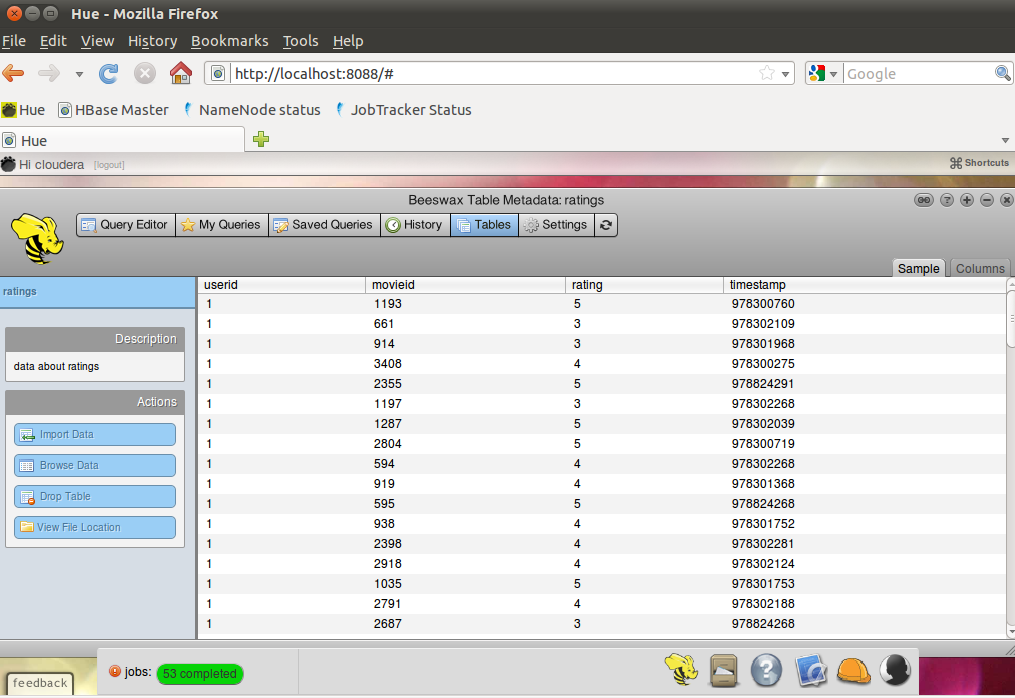
1. User Table.



2. Movies Table.



3. Ratings Table.



After that we apply some queries to the tables and came with the certain sets of Tables, by using certain delimiters and Serializable and Deserializable interfaces.

Some queries that we used are:-

* To sort the user with age group 18 (18 to 24yrs) and generate **sorteduser** tables.

Select \* from **usertable** where age =’18’;

* To create a new table called **usermovierating** from **sorteduser** and **rating** table**.**

Select userid, gender, age, movieid, rating from **sortedusers** join **rating** on (users.userid=movies,userid) where ratings.rating=’5’;

* To generate the **tentativeresult** table from **movies** and **usermovierating** table.

Select movieid, userid, gender, age, rating, genre from **movies** join **usermovierating** on (movies.movieid=usermovierating.movieid) where movies.movieid=usermovierating.movieid;

* To calculate the number of user under the age group 18 (18 to 24 yrs). i.e. **1098**

Select distinct userid, gender, age, rating, genre from tentativeresult;

* To calculate the number of females from (18 to 24 yrs). i.e. **298**

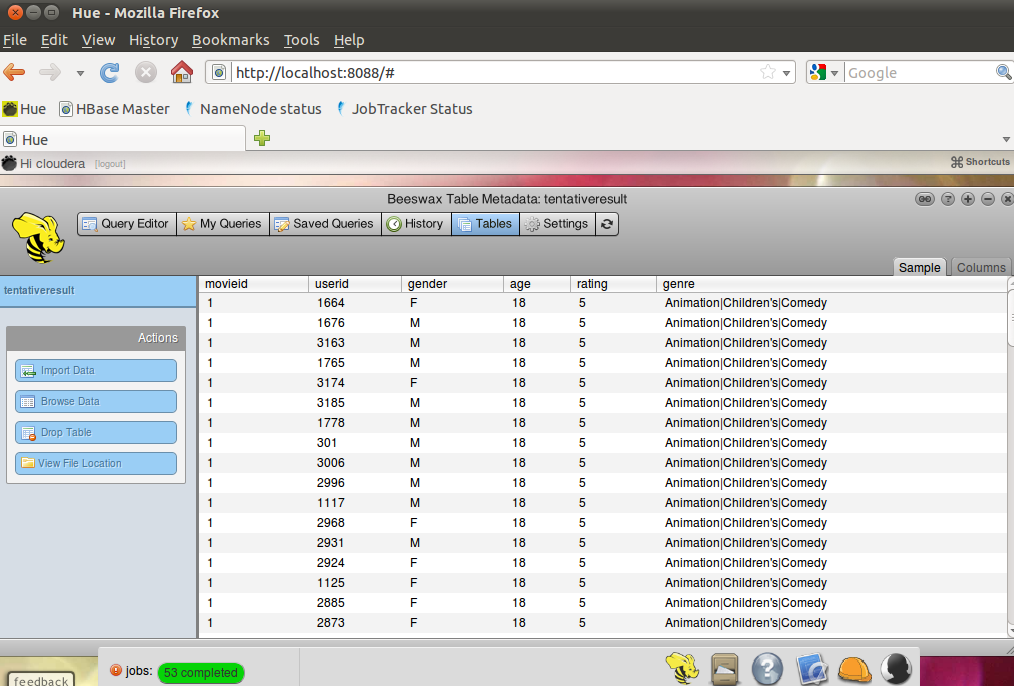
Select \* from tentativeresult where gender=’F’;

* To calculate the number of males from (18 to 24 yrs) i.e. **800**

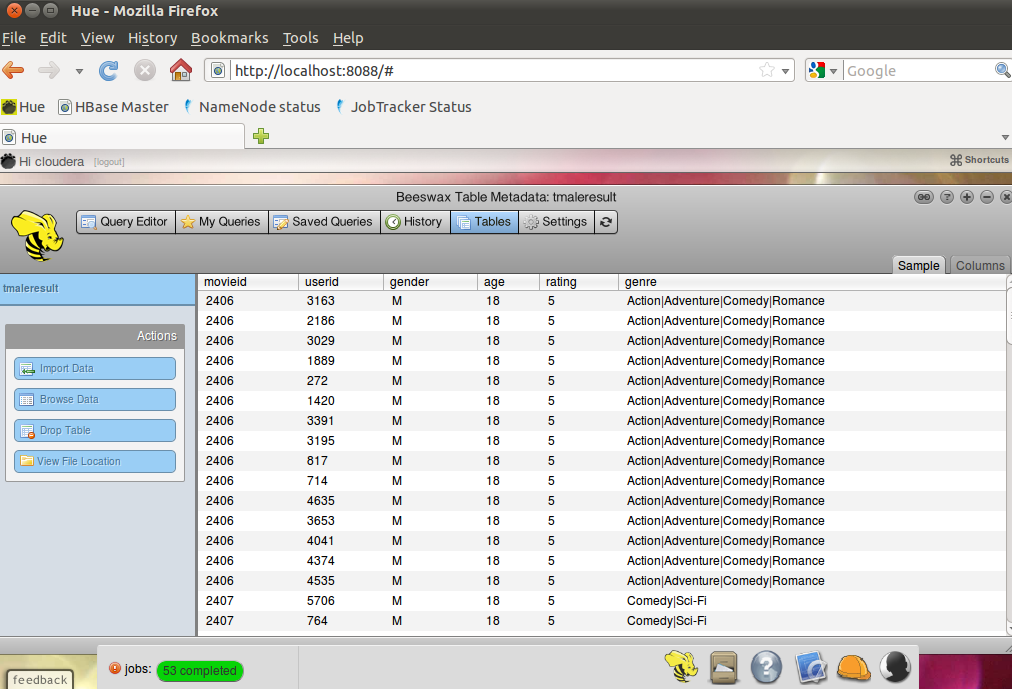
Select \* from tentativeresult where gender=’M’;

Some of the Screenshots of our result tables are:

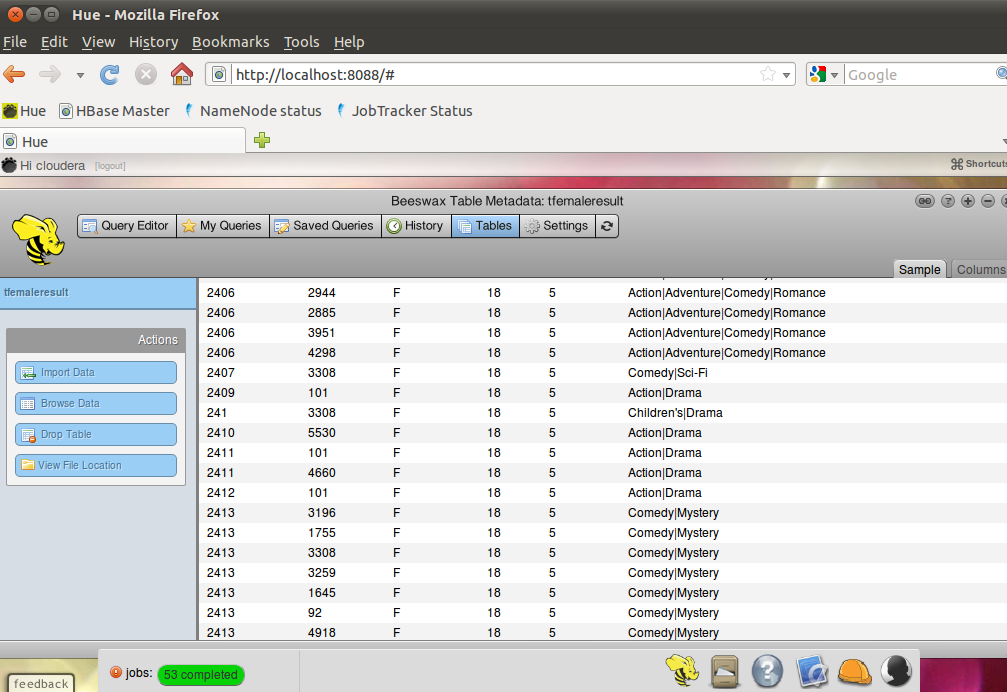
1. Tentative Result.



2. Male Table.



3. Female Table.



The above result tables are derived from the unstructured datasets that we have at the starting and with the help of this result tables the analyst can easily analyze the data according to the requirements which would be very hard to do with the unorganized data sets that we have earlier.