A

Project Report submitted in fulfillment of the requirements for the Internship. **CELEBAL TECHNOLOGIES**

VIACOM18- 90-D

(Big Data)

by

# Student Name Roll Number

Yash Mohan Garg R172218128

Yuvraj Monga R172218130

Drishay Arora R103218046 Isha Khera R103218061



under the guidance of

# Dr. Hitesh Kumar Sharma

University of Petroleum & Energy Studies Bidholi, Via Prem Nagar, Dehradun, UK

June 7,2021-July 22, 2021

# Contents:

1. Introduction-

* Problem Statement
* Apache Hive
* Hive on Cloudera

1. Data Sources-

* Data represented in a comma-separated .csv file

1. Approach to the problem statement-

* Data Flow Diagram

1. Query Implementation-

* Creation table and loading data from .csv file
* Calculating Viewers
* Calculating Visitors
* Combining both Queries

1. Results-

* Displaying Final Result
* Displaying Similar Outcomes for different Combinations.

# References

# Introduction:

**Problem Statement:**

Create a Hive table containing combination for from date and to date where a count of distinct visitor and viewer for that combination of time stamps. (From\_time < To\_time)

**WHAT IS APACHE HIVE?**

Apache Hive is an open source data warehouse software for reading, writing and managing large data set files that are stored directly in either the Apache Hadoop Distributed File System (HDFS) or other data storage systems such as Apache HBase. Hive enables SQL developers to write Hive Query Language (HQL) statements that are similar to standard SQL statements for data query and analysis. It is designed to make MapReduce programming easier because you don’t have to know and write lengthy Java code. Instead, you can write queries more simply in HQL, and Hive can then create the map and reduce the functions

**HIVE ON CLOUDERA:**

Hive data warehouse software enables reading, writing, and managing large datasets in distributed storage. Using the Hive query language (HiveQL), which is very similar to SQL, queries are converted into a series of jobs that execute on a Hadoop cluster through MapReduce or Apache Spark. Users can run batch processing workloads with Hive while also analyzing the same data for interactive SQL or machine-learning workloads using tools like Apache Impala or Apache Spark—all within a single platform.

Note : We have considered Cloudera installation and Apache Hive configuration on Cloudera, a prerequisite for the given project.

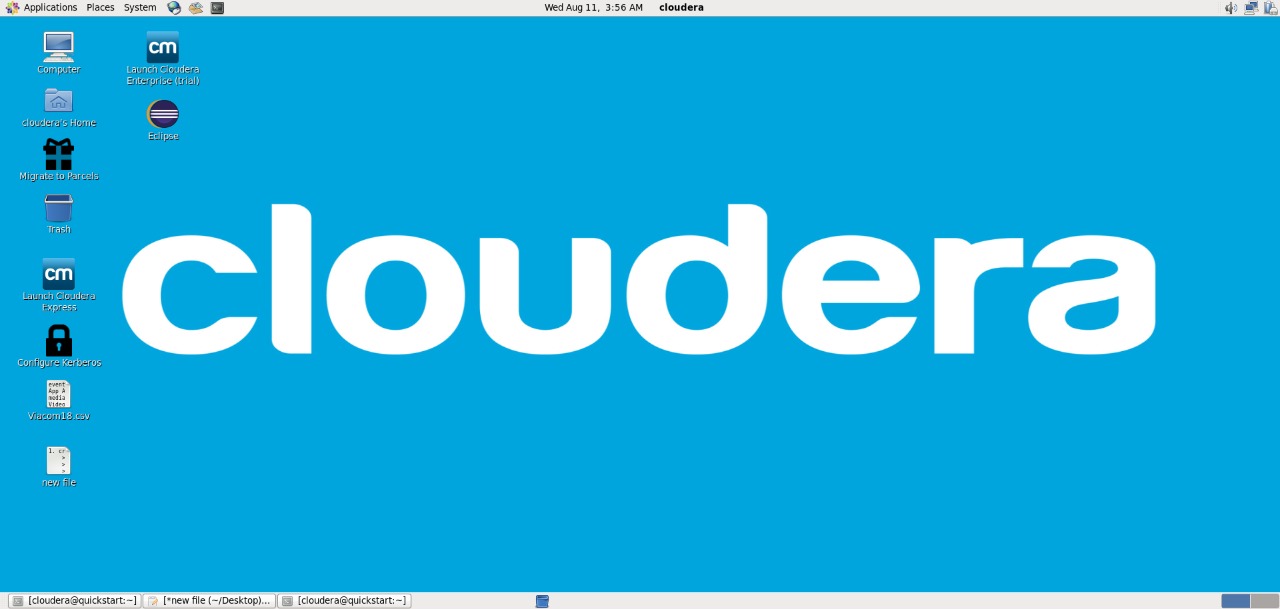


Fig1: Cloudera Interface

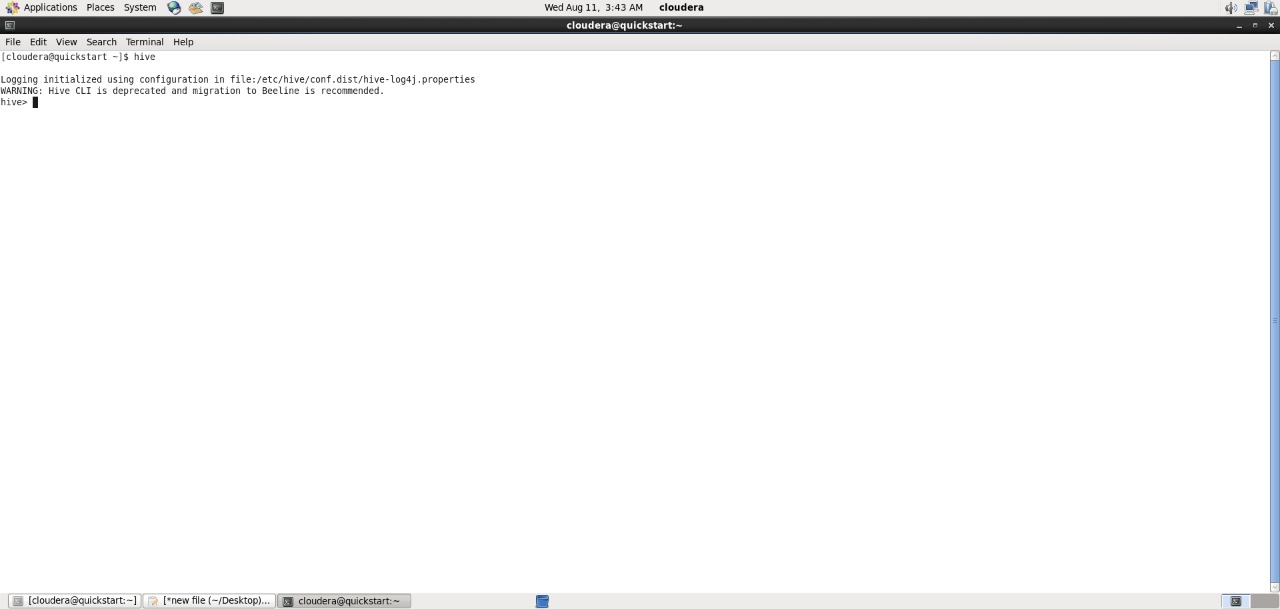


Fig2: Hive on CLI

**DATA SOURCES:**

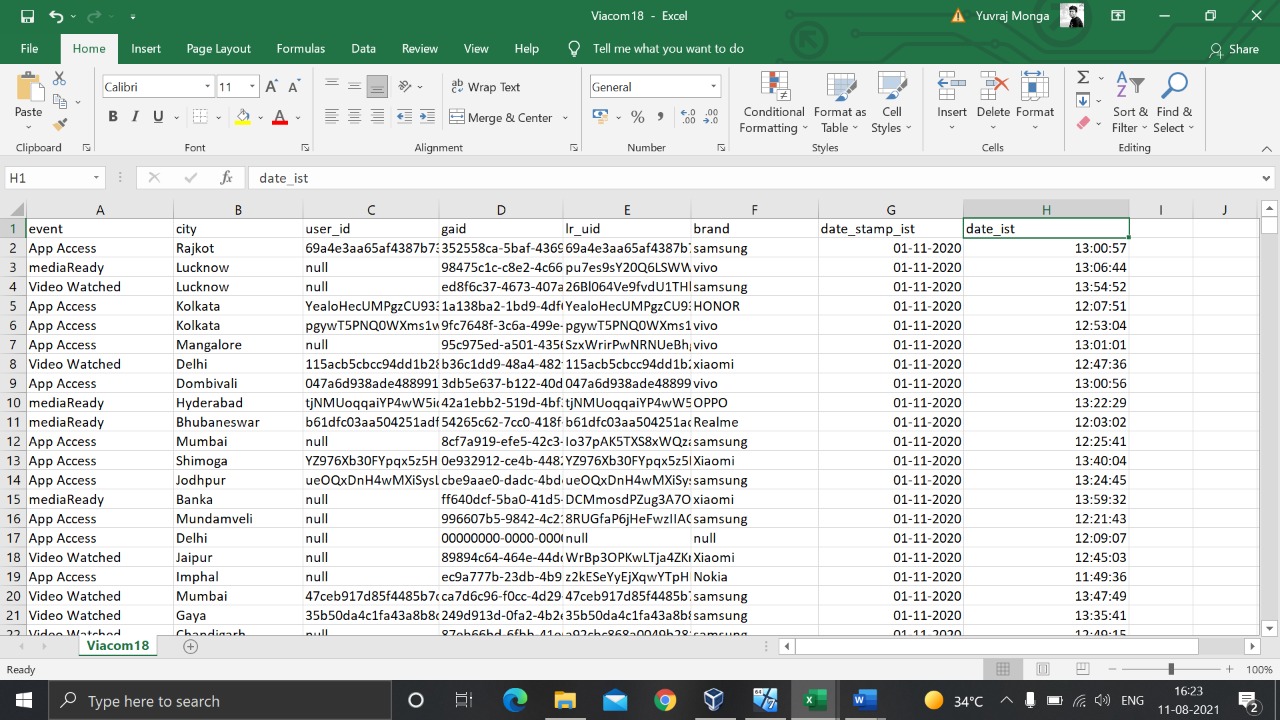


Fig 3: Excel File

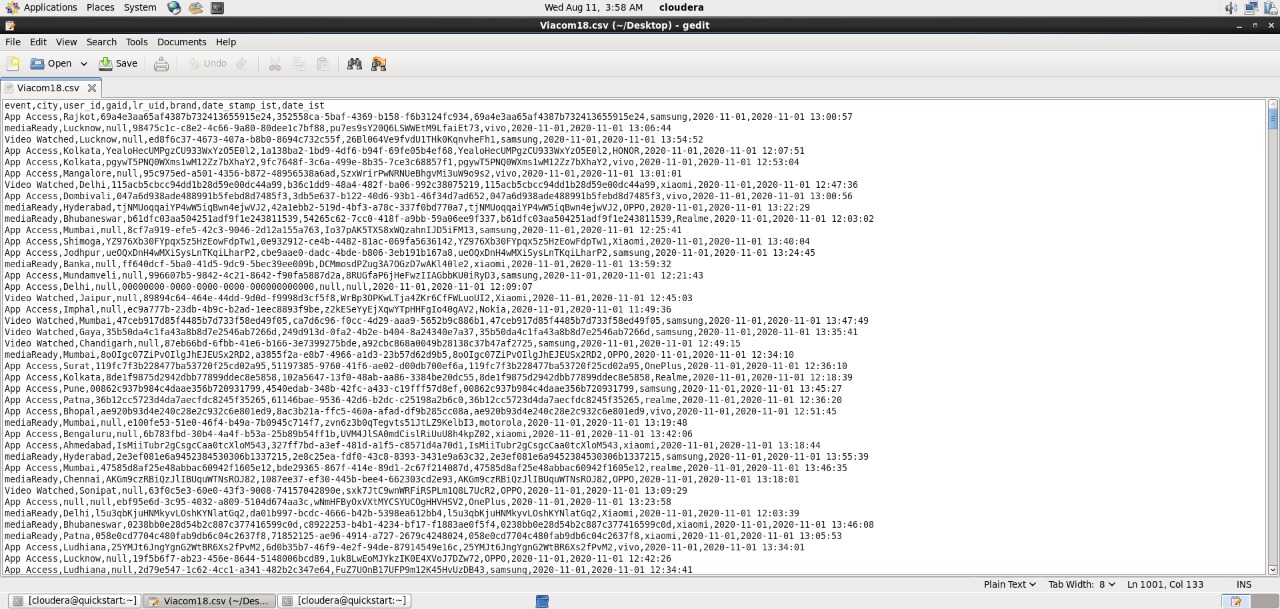
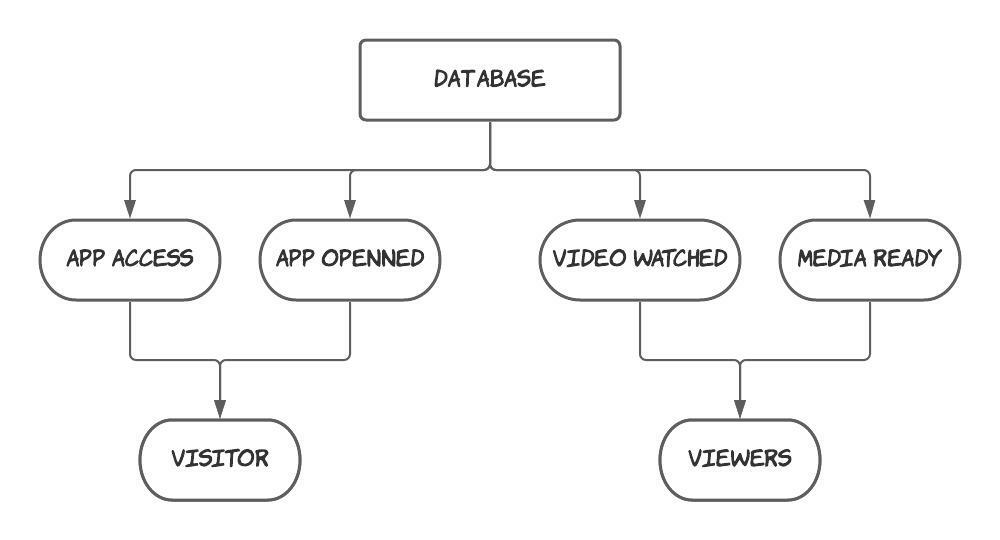


Fig 4: CSV file

**Approach to solve the problem statement:**

****

**Data Flow Diagram**

**Query Implementation:**

**Creating Table and Loading Data:**

*create table viacom18(event char(20), city char(20), user\_id char(20), gaid char(20), lr\_id char(20), brand char(15), date\_stamp\_ist date, date\_ist timestamp)*

*>row format delimited*

*>fields terminated by ',';*

*load data local inpath*

*> '/home/cloudera/Desktop/Viacom18.csv'*

*> into table viacom18;*

We started by creating a table viacom18 which includes attributes namely event, city,user\_id, gaid, lr\_id, brand, date\_stamp\_ist, date\_ist.

Then, we load the data into the table from the csv file present on the desktop.

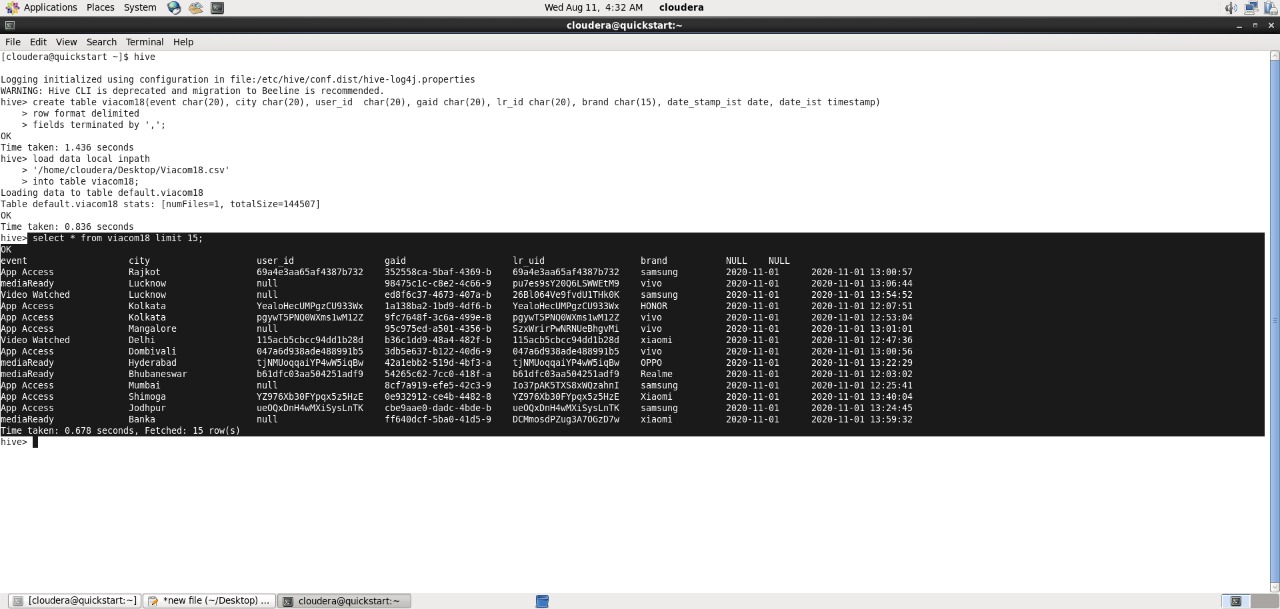
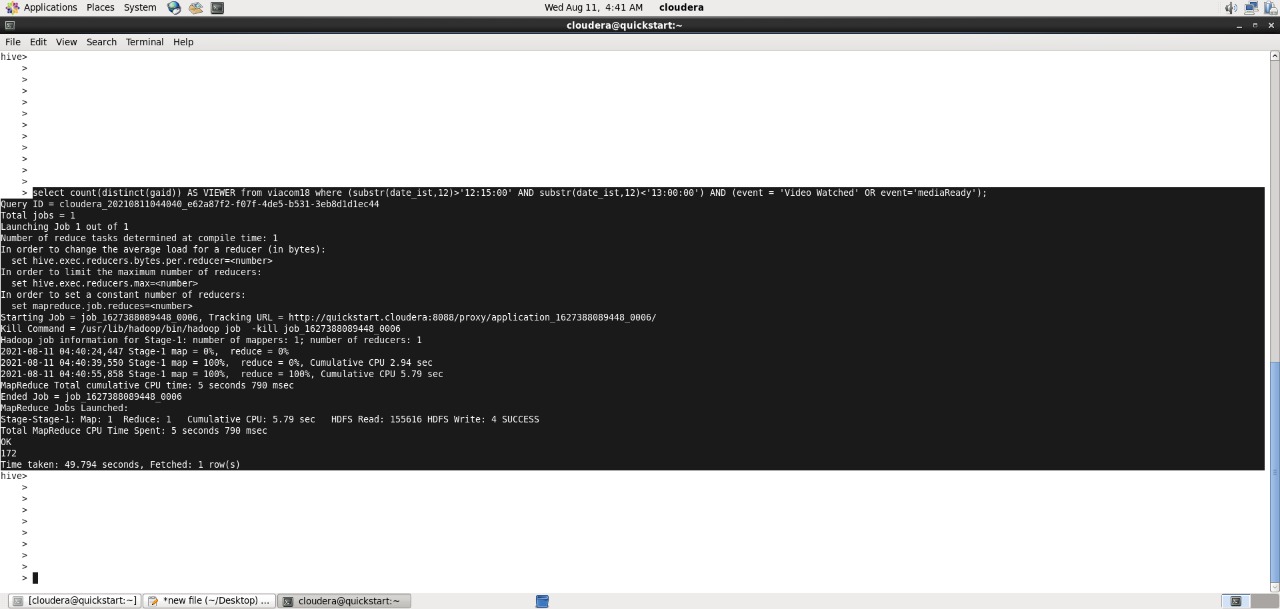


Fig 5: Data Sources.

**Viewers:**

*select count(distinct(gaid)) AS VIEWER from viacom18 where (substr(date\_ist,12)>'12:15:00' AND substr(date\_ist,12)<'13:00:00') AND (event = 'Video Watched' OR event='mediaReady');*

Following query helps us to count the distinct number of viewers, the attributes ‘Video Watched’ and ‘Media Ready’ are considered as viewers.

 Fig 6.Query Execution (Viewers)

OUTPUT : 175 Viewers.

**Visitors:**

*select count(distinct(gaid)) AS VISITORS from viacom18 where (substr(date\_ist,12)>'12:15:00' AND substr(date\_ist,12)<'13:00:00') AND (event = 'App Access' OR event='App Opened');*

Following query helps us to count the distinct number of visitors, the attributes ‘App Access’ and ‘App Opened’ are considered as visitors.

**DISTINCT()** function is used in SELECT statement in HIVE **to fetch only unique rows**. The row does not mean entire row in the table but it means "row" as per column listed in the SELECT statement.

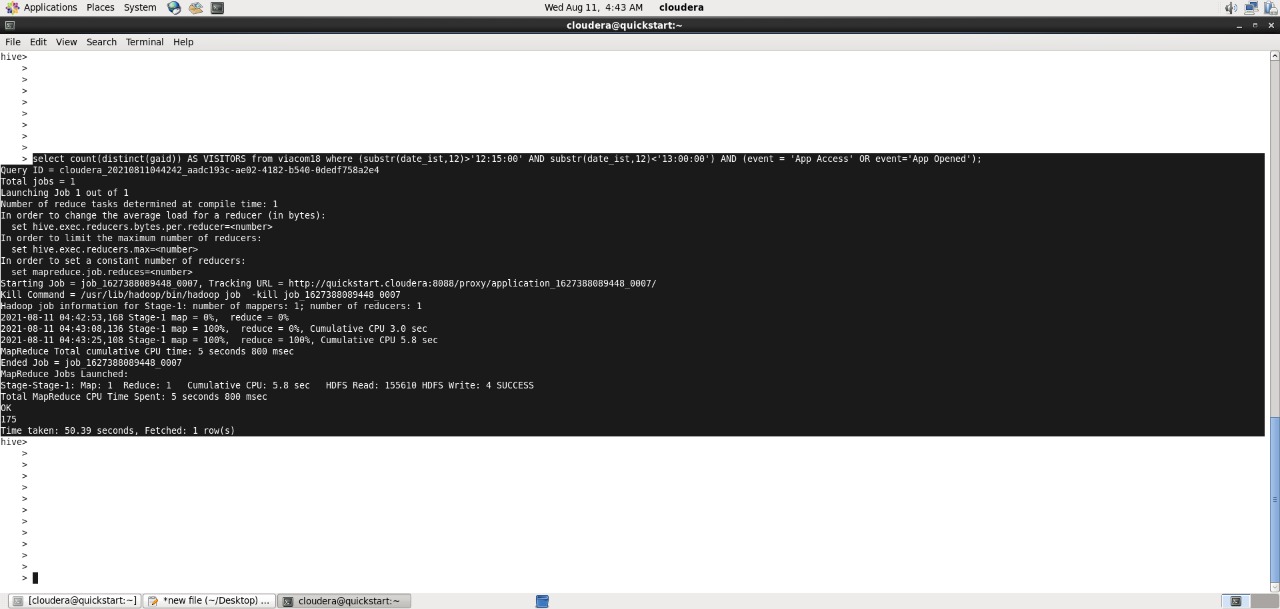


Fig 7. Query Execution (Visitors)

OUTPUT: 172 Visitors.

Combination of both queries to get the result at the same time.

*create table project as*

*> with viewers as (select count(distinct(gaid)) from viacom18 where(substr(date\_ist,12)>'12:15:00' AND substr(date\_ist,12)<'13:00:00') AND (event='Video Watched' OR event='mediaReady')),*

*> visitors as (select count(distinct(gaid)) from viacom18 where(substr(date\_ist,12)>'12:15:00' AND substr(date\_ist,12)<'13:00:00') AND (event='App Access' OR event='App Opened'))*

*> select \* from viewers UNION ALL select \* from visitors;*

The UNION set operation combines the results of two or more similar sub-queries into a single result set that contains the rows that are returned by all SELECT statements. Data types of the column that you are trying to combine should match. The Hive UNION set operation is different from JOIN, which combine the columns from two tables.

Therefore, the following query helps in achieving a combined result at the same time.

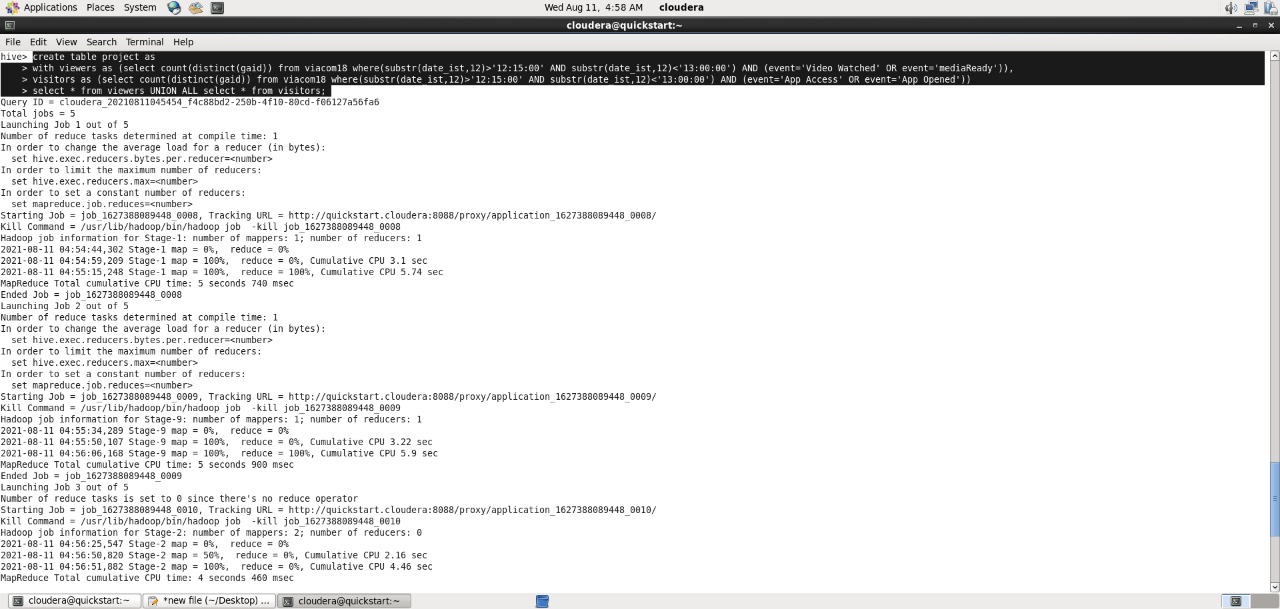


Fig 8. Combination of both queries to get result at the same time.

**Results:**

*select \* from project;*

SELECT statement is used to retrieve the data from a table.SELECT \* the simplest query is effectively one which returns the contents of the whole table

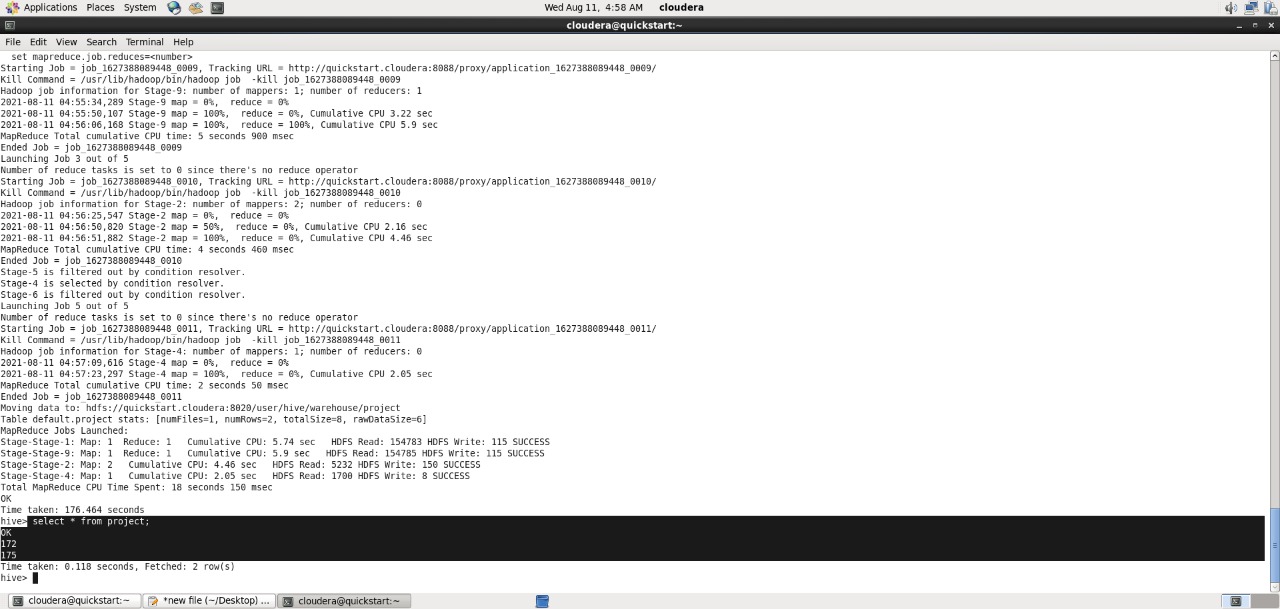


Fig 9

**Similar Outcomes:**

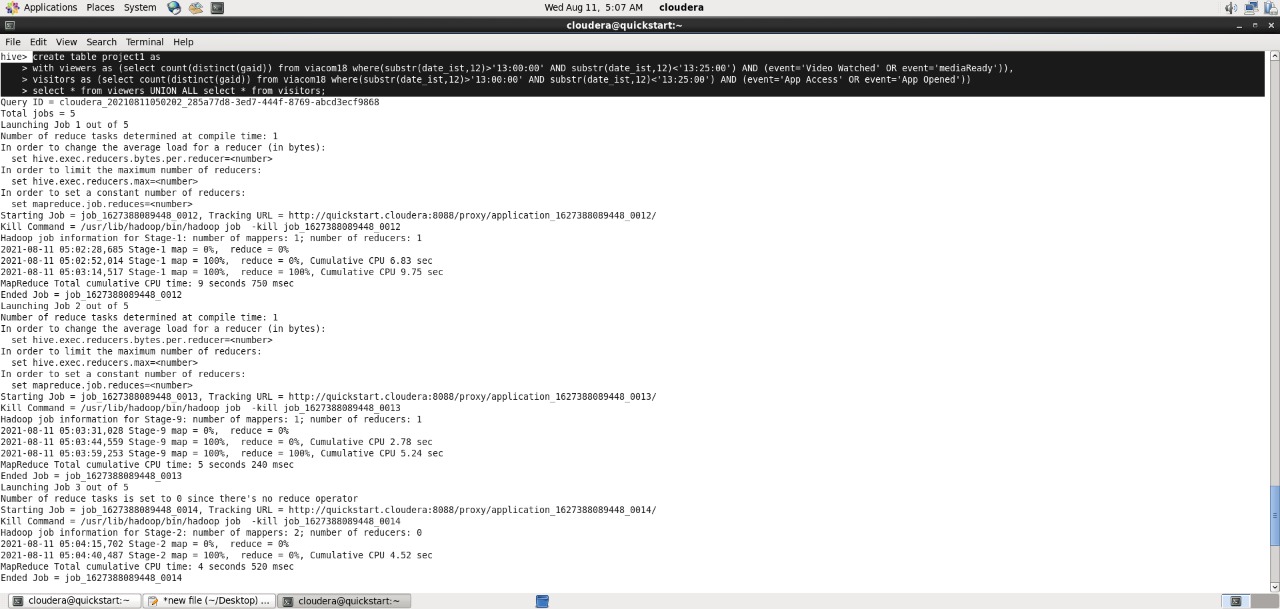
The above query can also be executed with different time combinations (From\_time < To\_time)

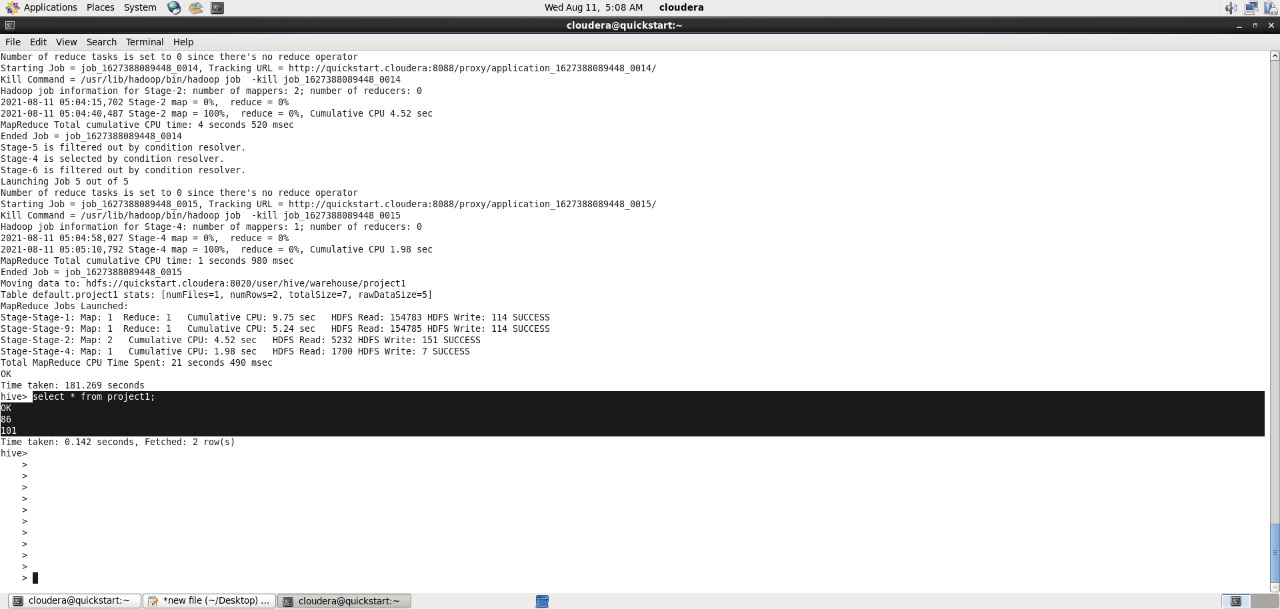
*with viewers as (select count(distinct(gaid)) from viacom18 where(substr(date\_ist,12)>'XX:XX:XX' AND substr(date\_ist,12)<'XX:XX:XX') AND (event='Video Watched' OR event='mediaReady')),*

*> visitors as (select count(distinct(gaid)) from viacom18 where(substr(date\_ist,12)>'XX:XX:XX' AND substr(date\_ist,12)<'XX:XX:XX') AND (event='App Access' OR event='App Opened'))*

*> select \* from viewers UNION ALL select \* from visitors;*

Here ‘XX:XX:XX’ denotes time





**References:**

**[1]** [**https://cwiki.apache.org/confluence/display/HIVE**](https://cwiki.apache.org/confluence/display/HIVE)

**-** Official documentation for Apache Hive to study basic DDL and DML commands.

**[2]** [**https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/hive\_intro.html**](https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/hive_intro.html)

* An Overview of Hive on Cloudera interface.

**[3]** [**https://community.cloudera.com/t5/Support-Questions/Importing-data-from-two-tables-to-a-single-table-in-Hive/td-p/112445**](https://community.cloudera.com/t5/Support-Questions/Importing-data-from-two-tables-to-a-single-table-in-Hive/td-p/112445)

**[4]** [**https://adilmca.wordpress.com/2016/08/13/hive-create-table-with-data-from-2-different-tables/**](https://adilmca.wordpress.com/2016/08/13/hive-create-table-with-data-from-2-different-tables/)

* [3] and [4] contain the piece of information that has helped us to bring results from two different queries into a single table.