**Title - Hive Mini Project 1 – Big Data class ineuron**

Name -Mayur Thele

Email – [Mayur.thele1@gmail.com](mailto:Mayur.thele1@gmail.com)

**Project info and requirement –**

This is a real time dataset of the ineuron technical consultant team. You have to perform hive analysis on this given dataset.

Download Dataset 1 - https://drive.google.com/file/d/1WrG-9qv6atP-W3P\_-gYln1hHyFKRKMHP/view

Download Dataset 2 - https://drive.google.com/file/d/1-JIPCZ34dyN6k9CqJa-Y8yxIGq6vTVXU/view

Note: both files are csv files.

**Problem Statement 1-**

**Create a schema based on the given dataset**

**Solution –**

**Table 1 –**

create table tbl\_AgentLogingReport(

SL\_No int,

Agent string,

Date string,

Login\_Time string,

Logout\_Time string,

Duration string)

row format delimited

fields terminated by ','

tblproperties("skip.header.line.count"="1");

**Table 2 –**

create table tbl\_AgentPerformance(

SL\_No int,

Date string,

Agent\_Name string,

Total\_Chats int,

Average\_Response\_Time string,

Average\_Resolution\_Time string,

Average\_Rating float,

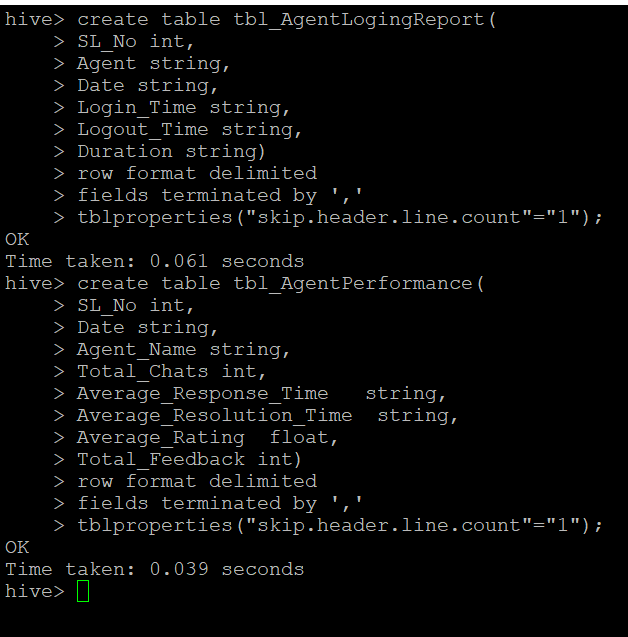
Total\_Feedback int)

row format delimited

fields terminated by ','

tblproperties("skip.header.line.count"="1");

**CLI Output –**



**Problem Statement 2-**

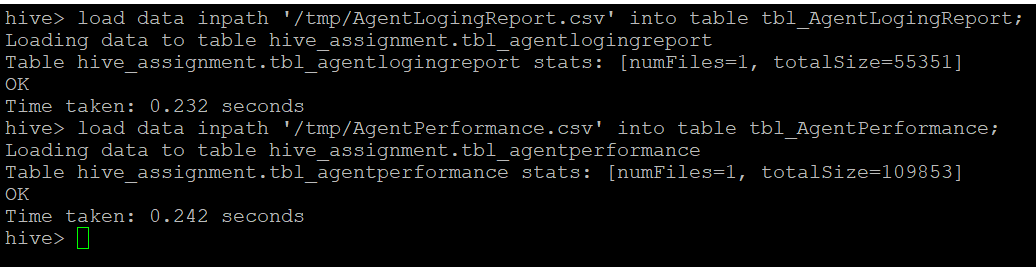
**Dump the data inside the hdfs in the given schema location.**

**Solution –**

load data inpath '/tmp/AgentPerformance.csv' into table tbl\_AgentPerformance;

load data inpath '/tmp/AgentLogingReport.csv' into table tbl\_AgentLogingReport;

**CLI Output –**



**Problem Statement 3-**

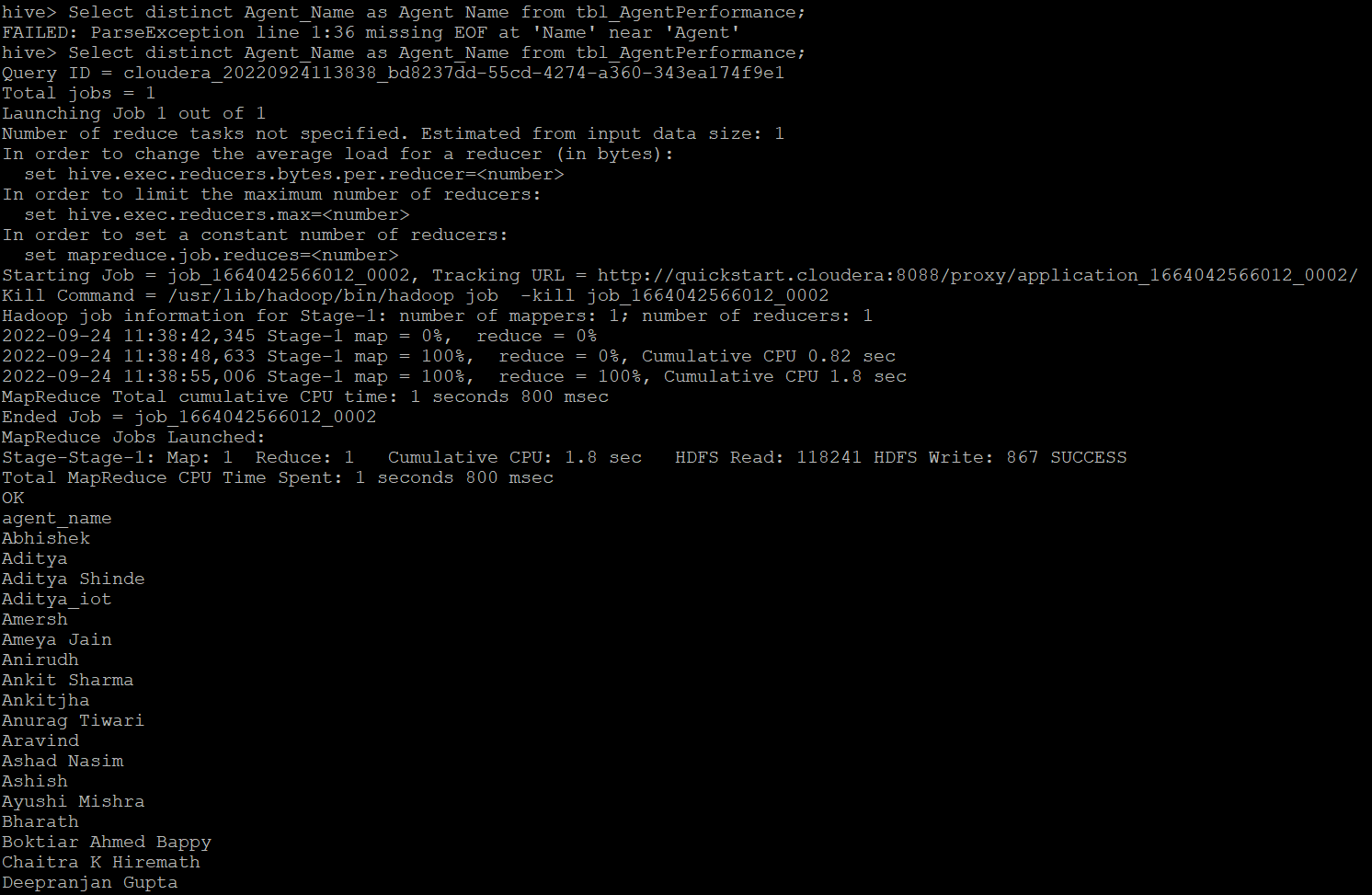
**List of all agents' names.**

**Solution –**

Select distinct Agent\_Name as Agent\_Name from tbl\_AgentPerformance;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below -





**Problem Statement 4-**

**Find out agent average rating.**

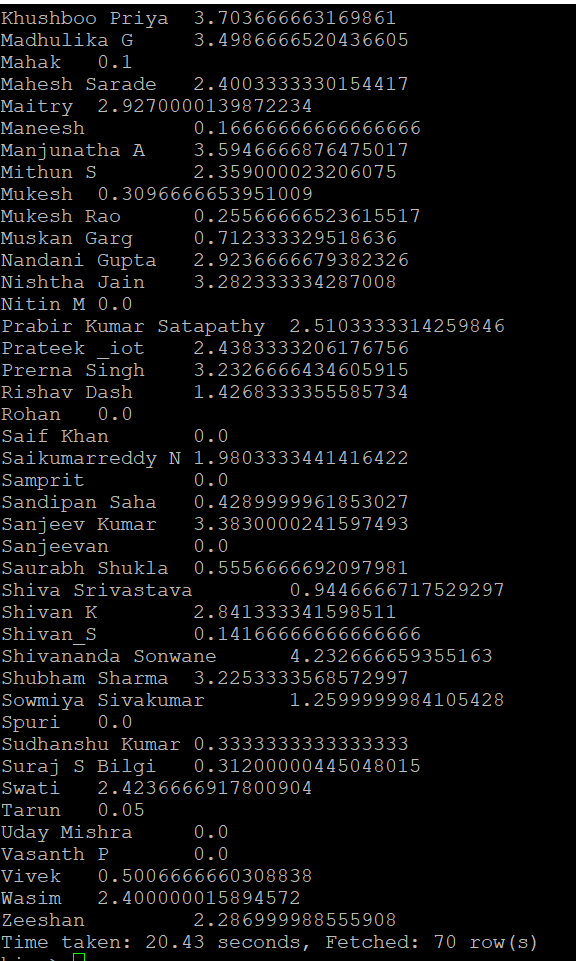
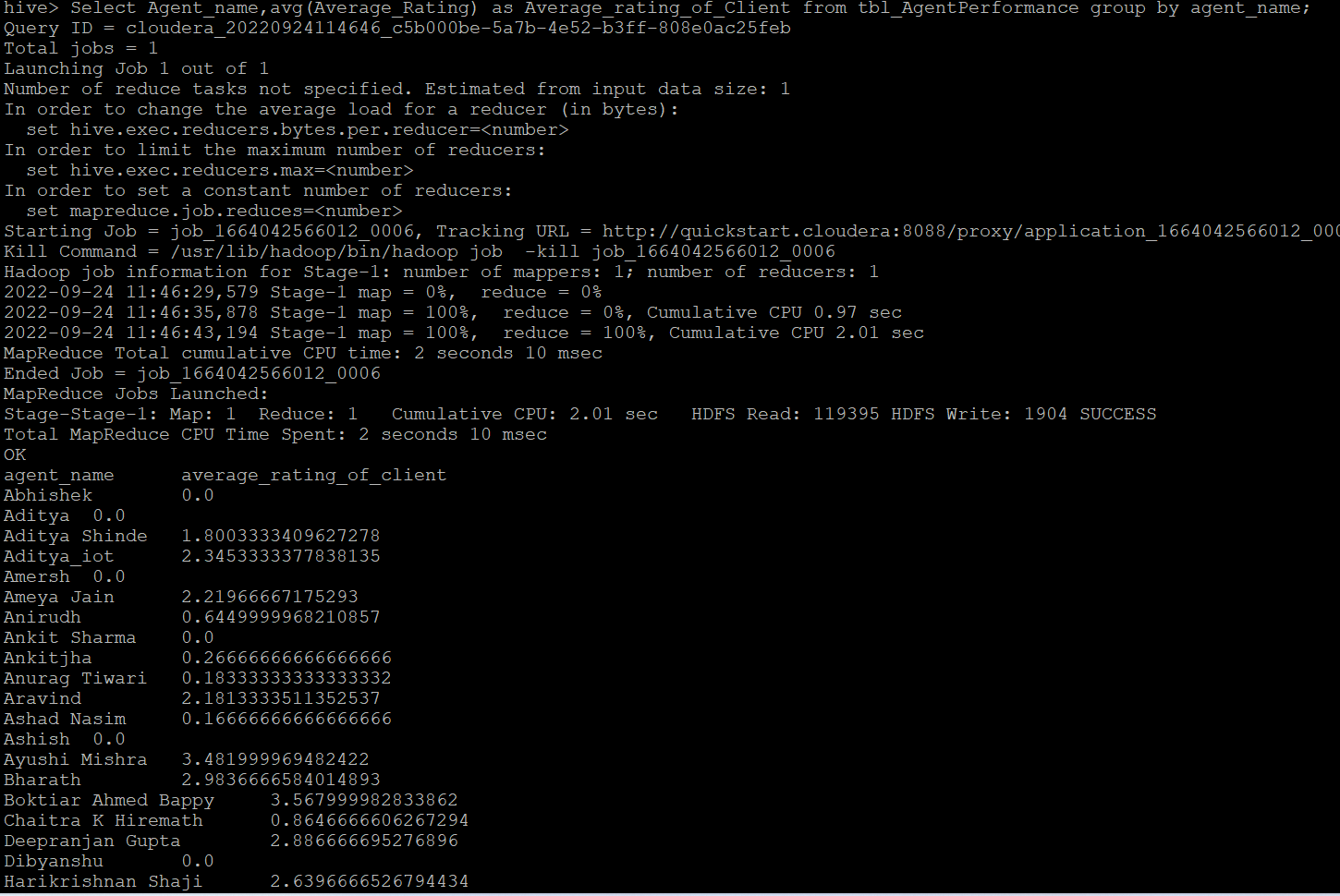
**Solution –**

As per problem solution we need to find the average rating for each agent

Select Agent\_name,avg(Average\_Rating) as Average\_rating\_of\_Client from tbl\_AgentPerformance group by agent\_name;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below



**Problem Statement 5-**

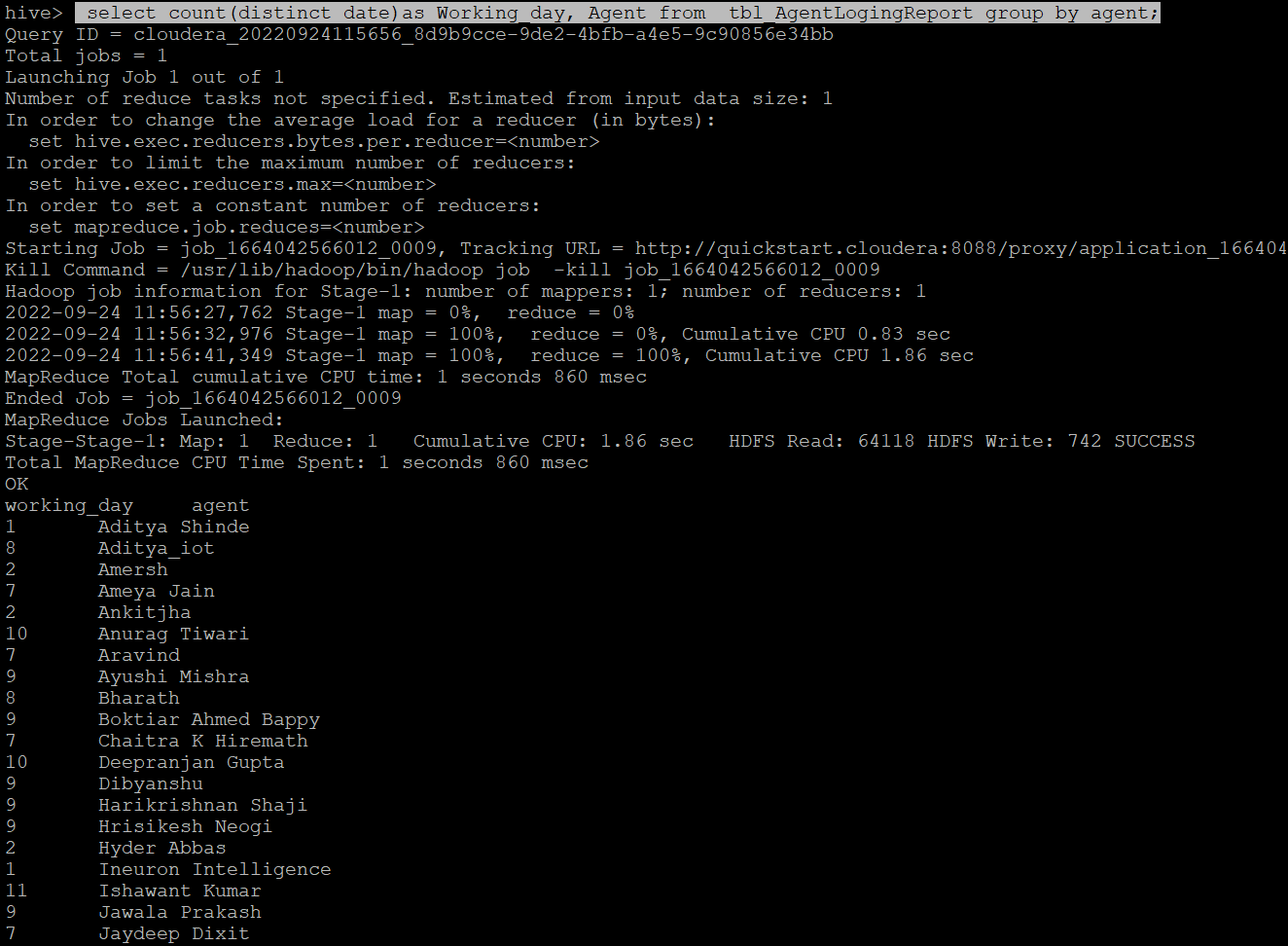
**Total working days for each agents –**

**Solution –**

select count(distinct date)as Working\_day, Agent from tbl\_AgentLogingReport group by agent;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 6-**

**Total query that each agent have taken -**

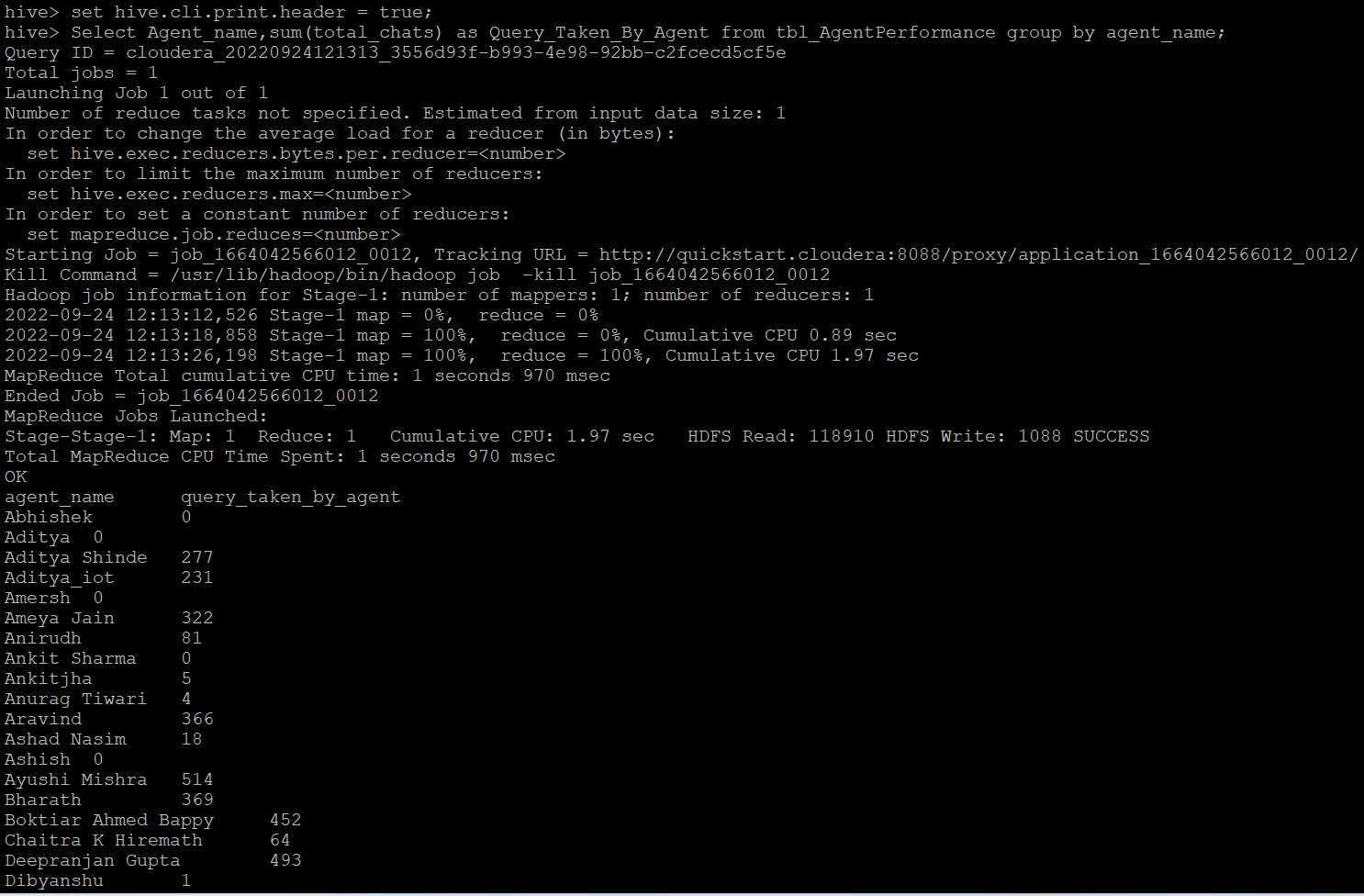
**Solution –**

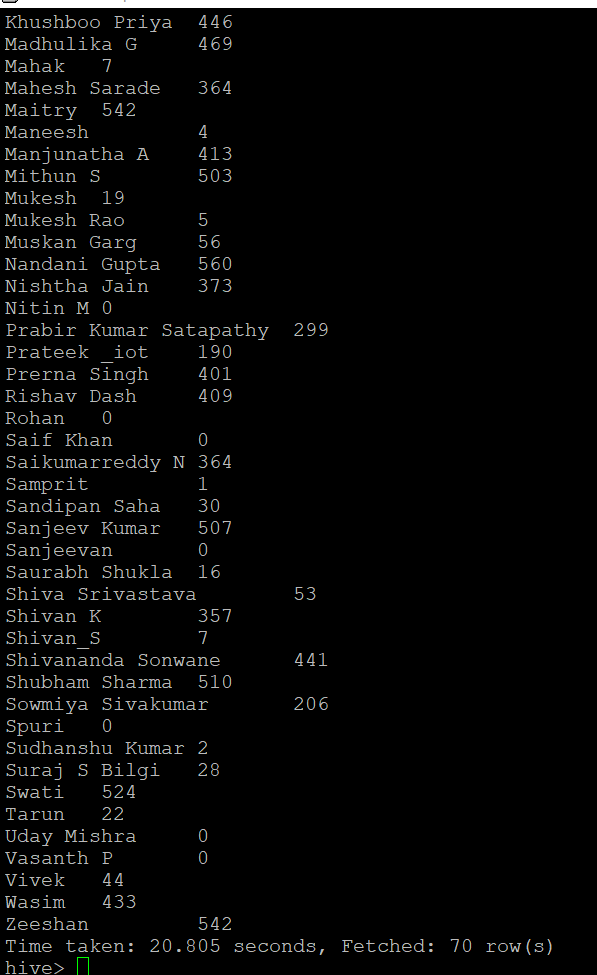
*Considering query means the questions so equivalent to chat*

Select Agent\_name,sum(total\_chats) as Query\_Taken\_By\_Agent from tbl\_AgentPerformance group by agent\_name;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 7-**

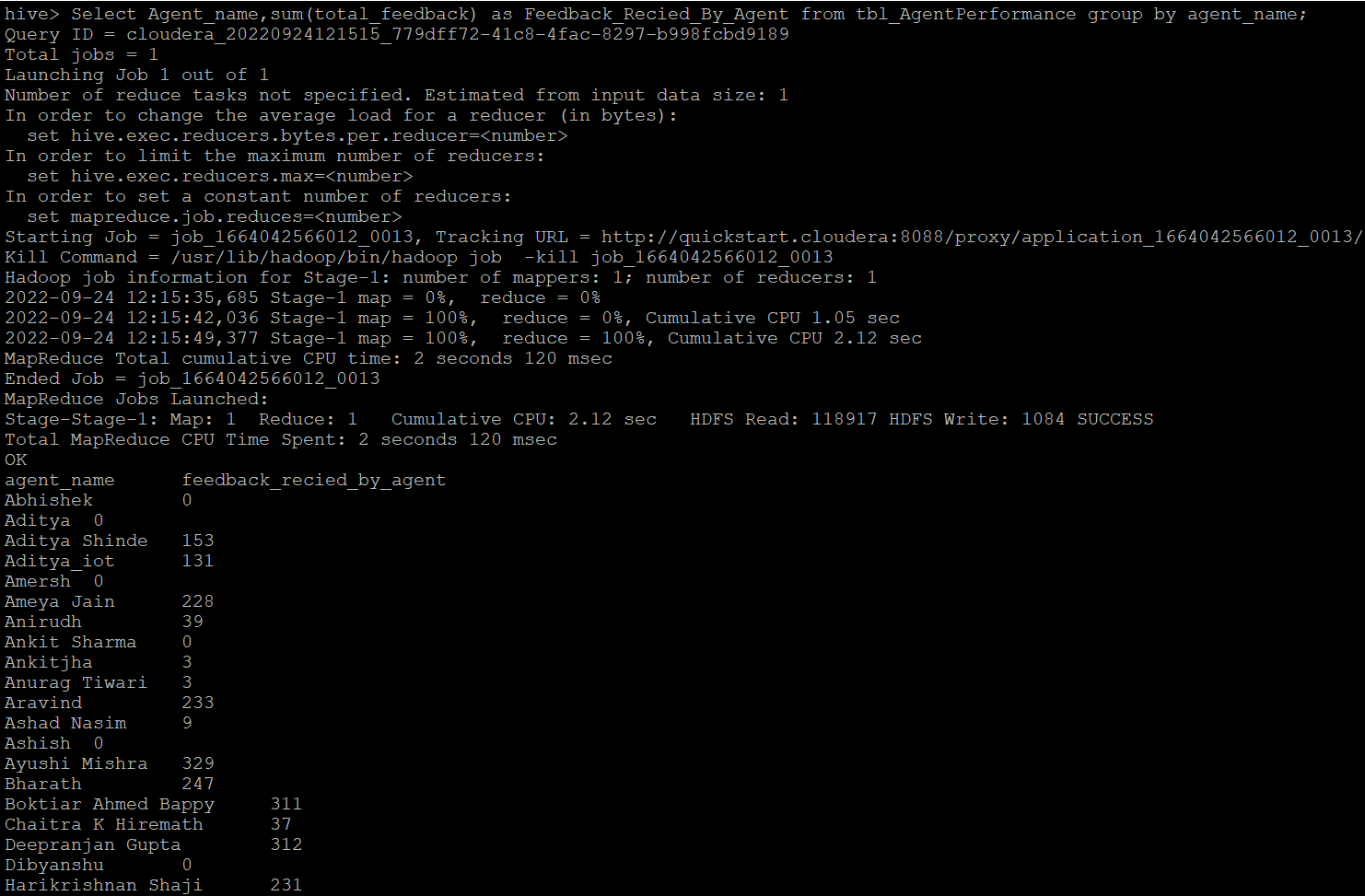
**Total Feedback that each agent have received -**

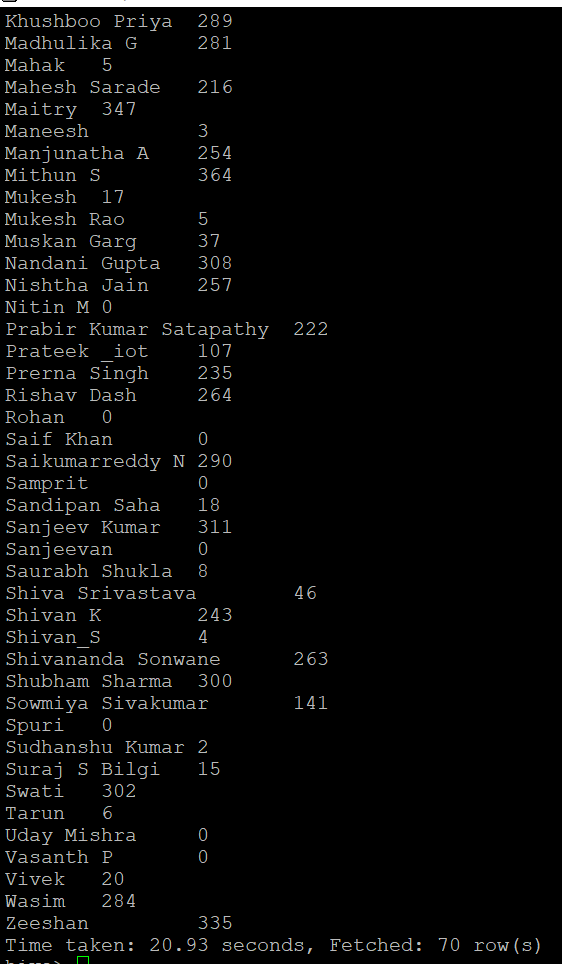
**Solution –**

Select Agent\_name,sum(total\_feedback) as Feedback\_Recied\_By\_Agent from tbl\_AgentPerformance group by agent\_name;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 8-**

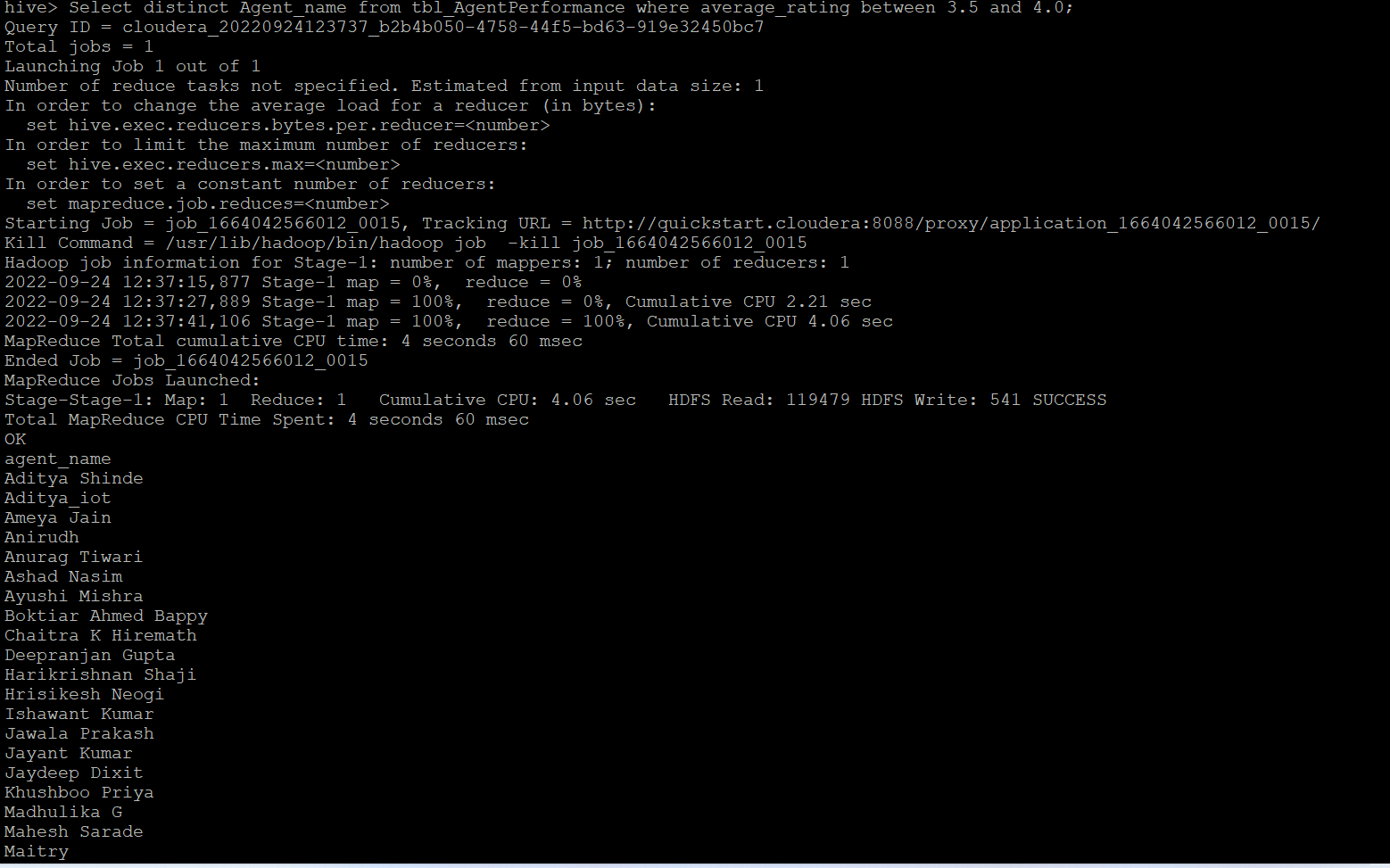
**Agent name who have average rating between 3.5 to 4 -**

**Solution –**

Select distinct Agent\_name from tbl\_AgentPerformance where average\_rating between 3.5 and 4.0;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 9-**

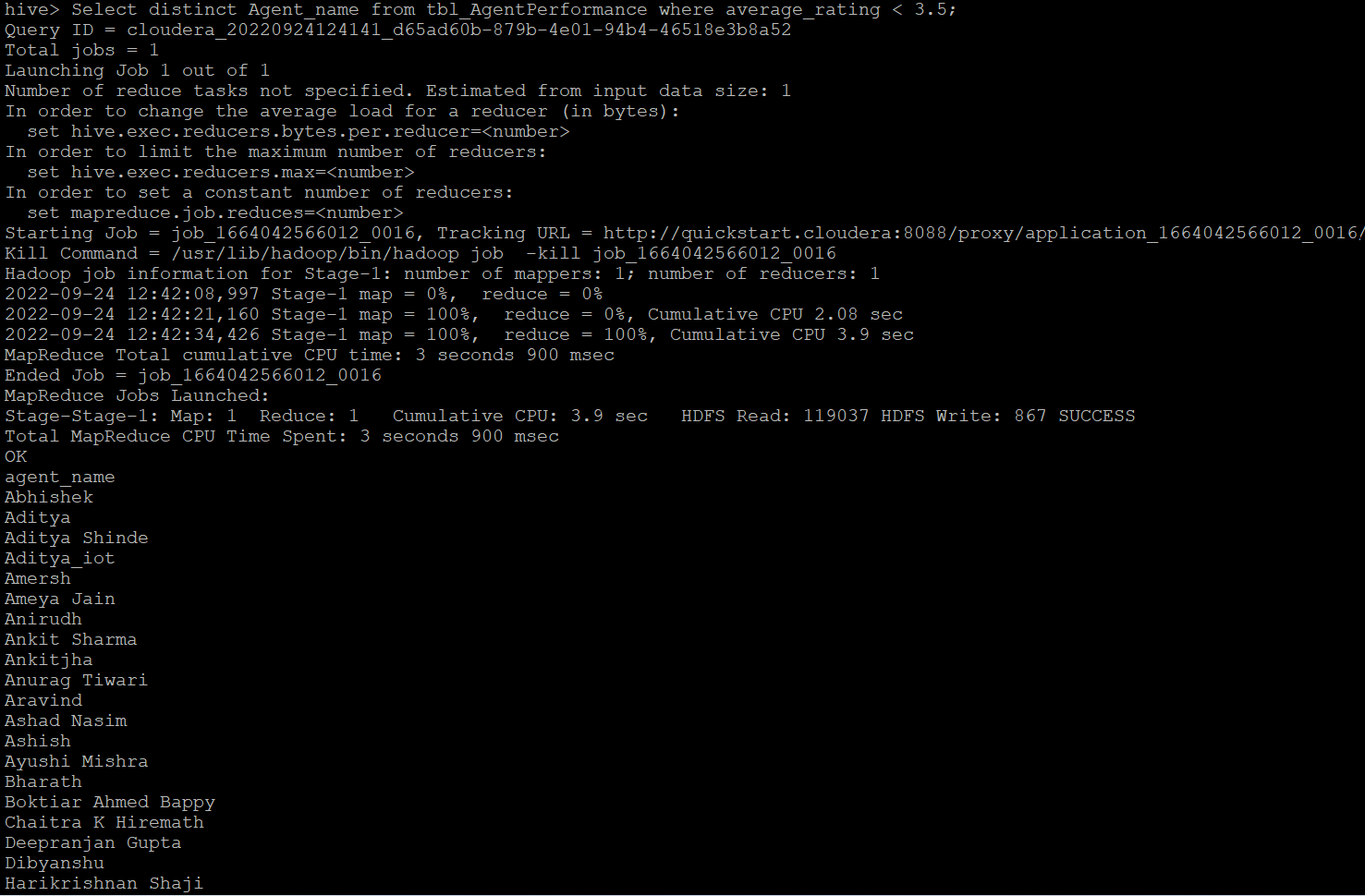
**Agent name who have rating less than 3.5 –**

**Solution –**

Select distinct Agent\_name from tbl\_AgentPerformance where average\_rating < 3.5;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 10-**

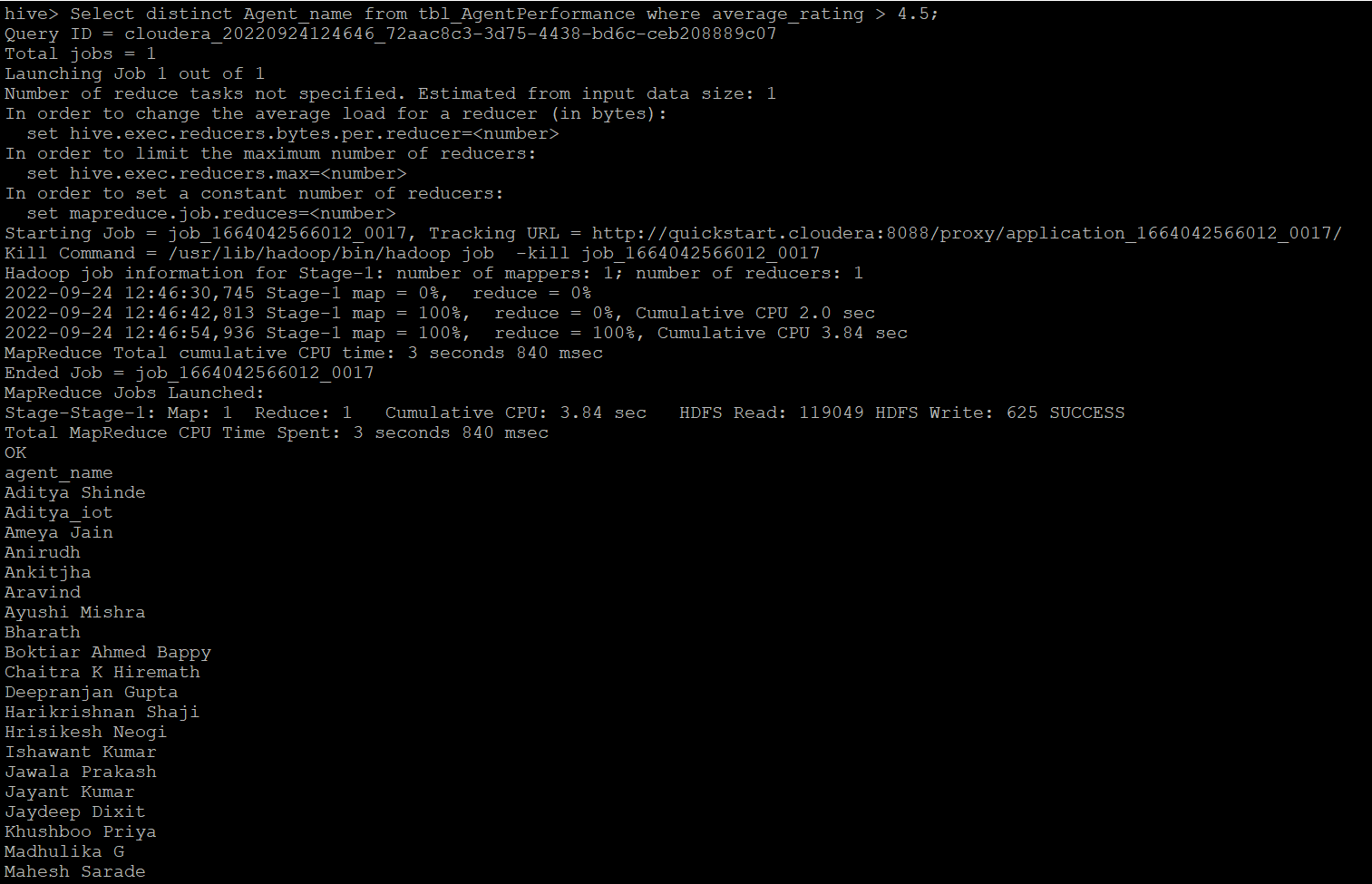
**Agent name who have rating more than 4.5 -**

**Solution –**

Select distinct Agent\_name from tbl\_AgentPerformance where average\_rating > 4.5;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 11-**

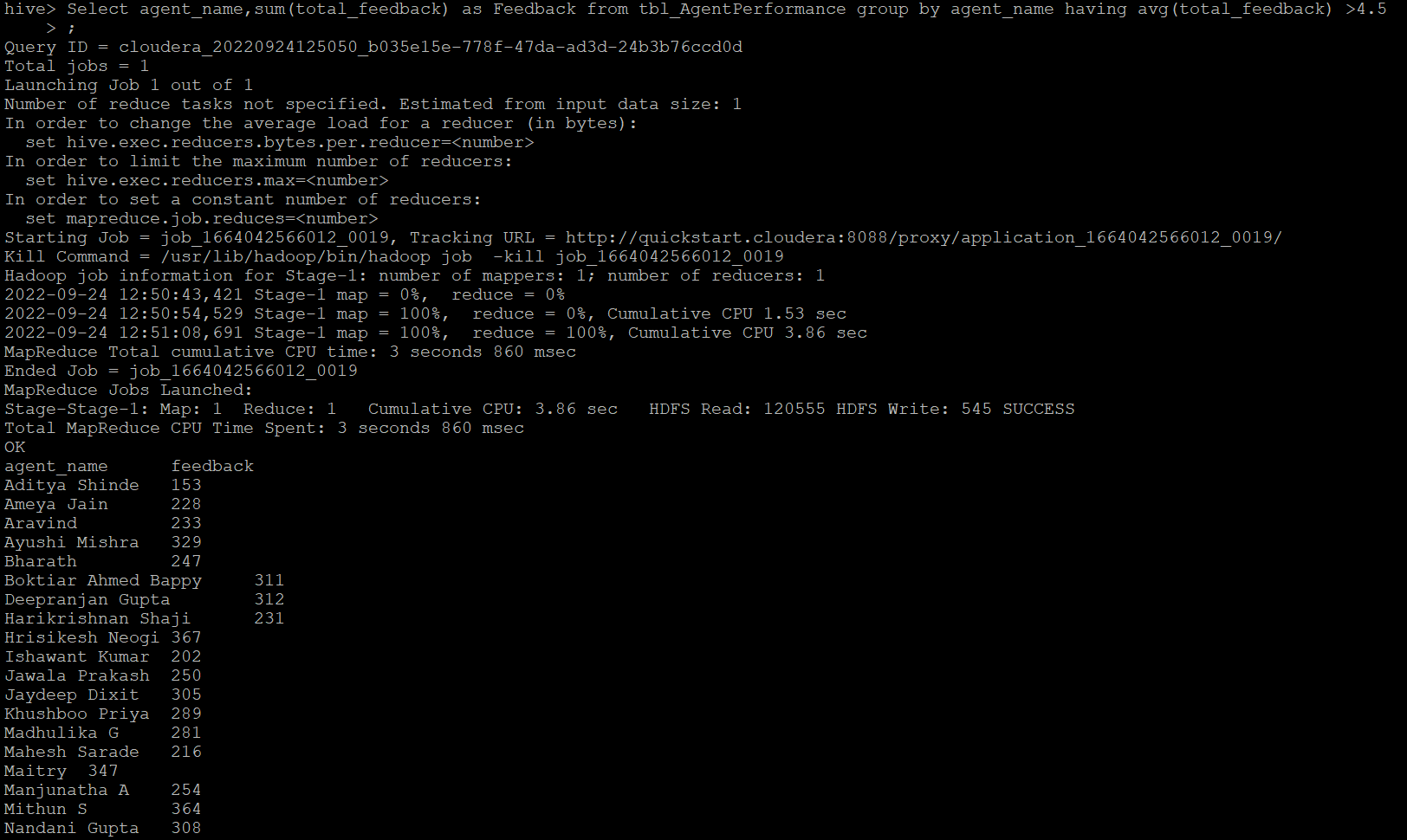
**How many feedback agents have received more than 4.5 average**

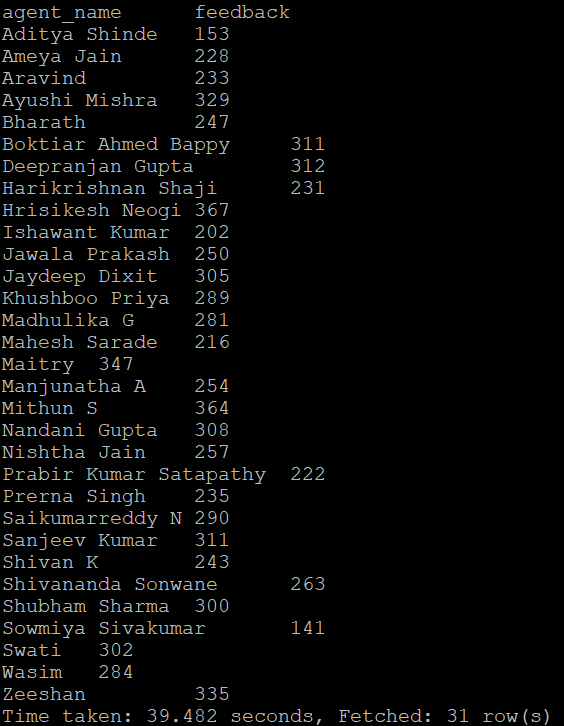
**Solution –**

Select agent\_name,sum(total\_feedback) as Feedback from tbl\_AgentPerformance group by agent\_name having avg(total\_feedback) >4.5

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 12-**

**Average weekly response time for each agent**

**Solution –** *we have used unix\_timestamp and from\_unixtime function to convet string into time or date data. And Cast function to convert date/time data after aggregation into BigInt and then to string for print purpose*

select from\_unixtime(cast(avg(unix\_timestamp(Average\_Response\_Time,'HH:mm:ss'))as bigint),'HH:mm:ss') as Average\_Response\_Time, week, agent\_name

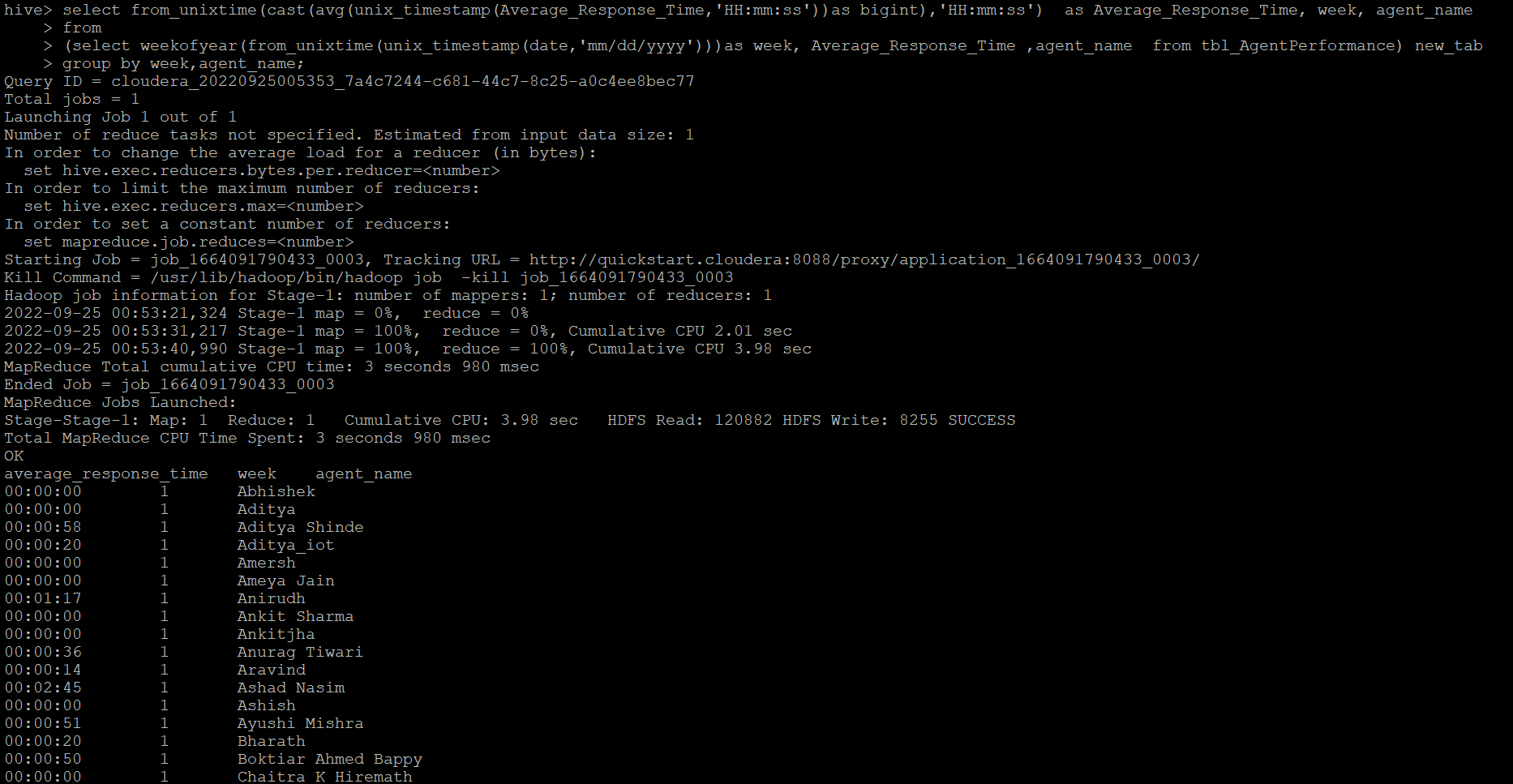
from

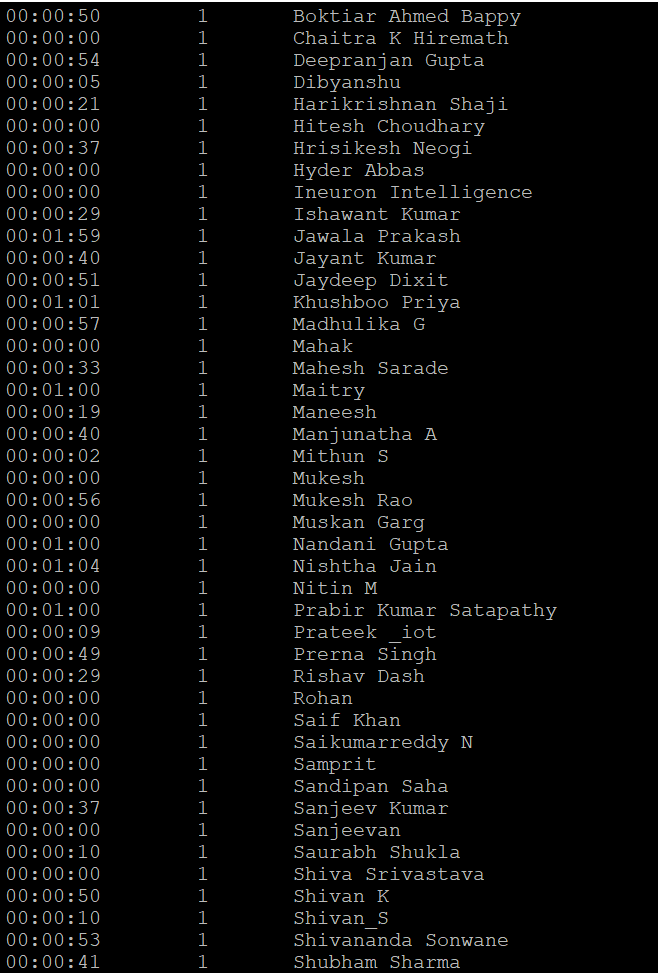
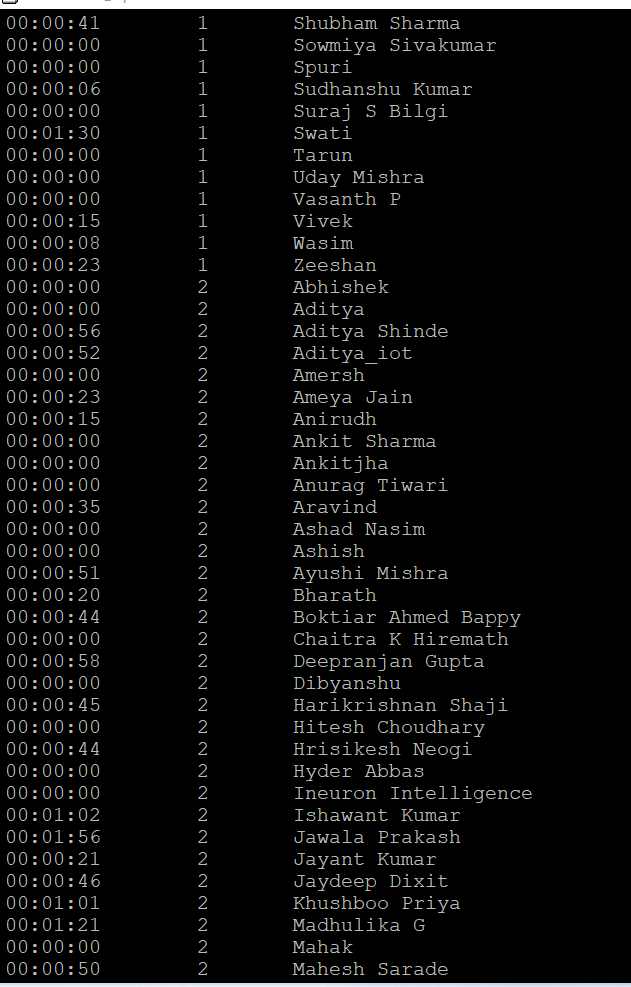
(select weekofyear(from\_unixtime(unix\_timestamp(date,'mm/dd/yyyy')))as week, Average\_Response\_Time ,agent\_name from tbl\_AgentPerformance) new\_tab

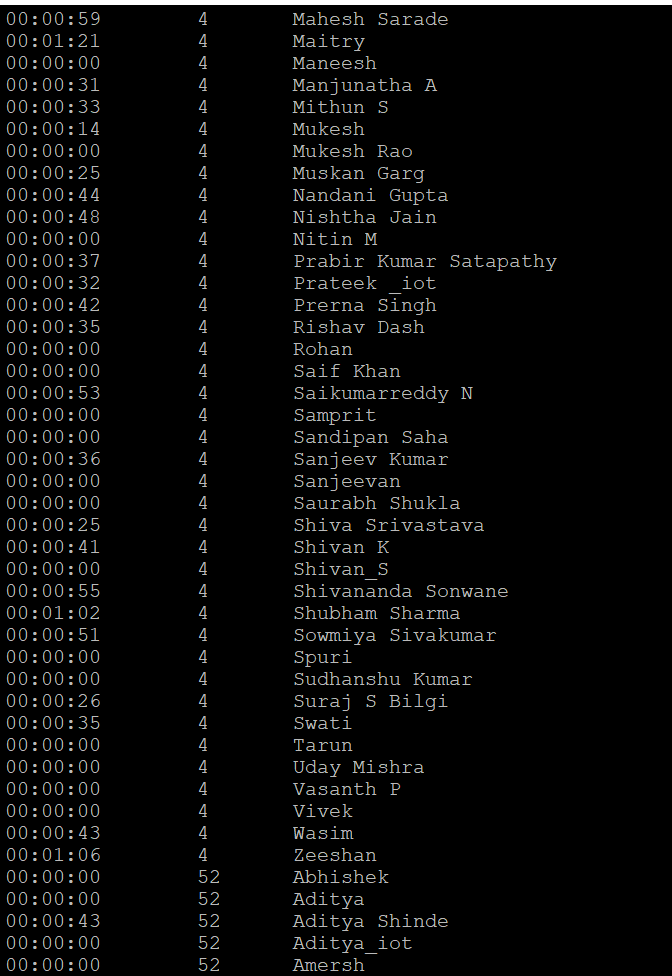
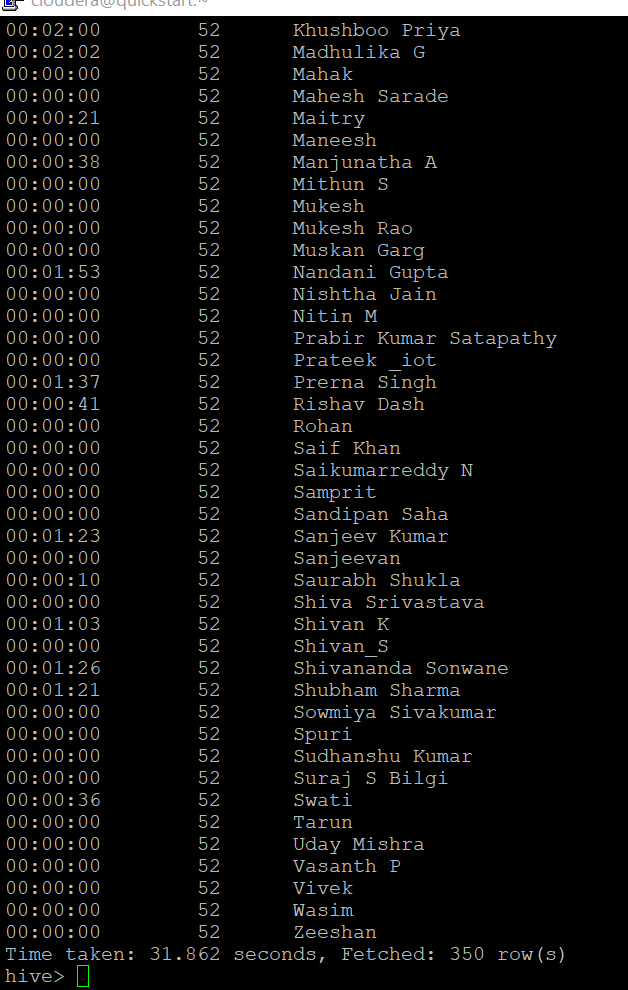
group by week,agent\_name;

**CLI Output –**

Complete output with respective quarter



**Problem Statement 13-**

**Average weekly resolution time for each agents**

**Solution –** *we have used unix\_timestamp and from\_unixtime function to convet string into time or date data. And Cast function to convert date/time data after aggregation into BigInt and then to string for print purpose*

select from\_unixtime(cast(avg(unix\_timestamp(Average\_Resolution\_Time,'HH:mm:ss'))as bigint),'HH:mm:ss') as Average\_Resolution\_Time, week, agent\_name

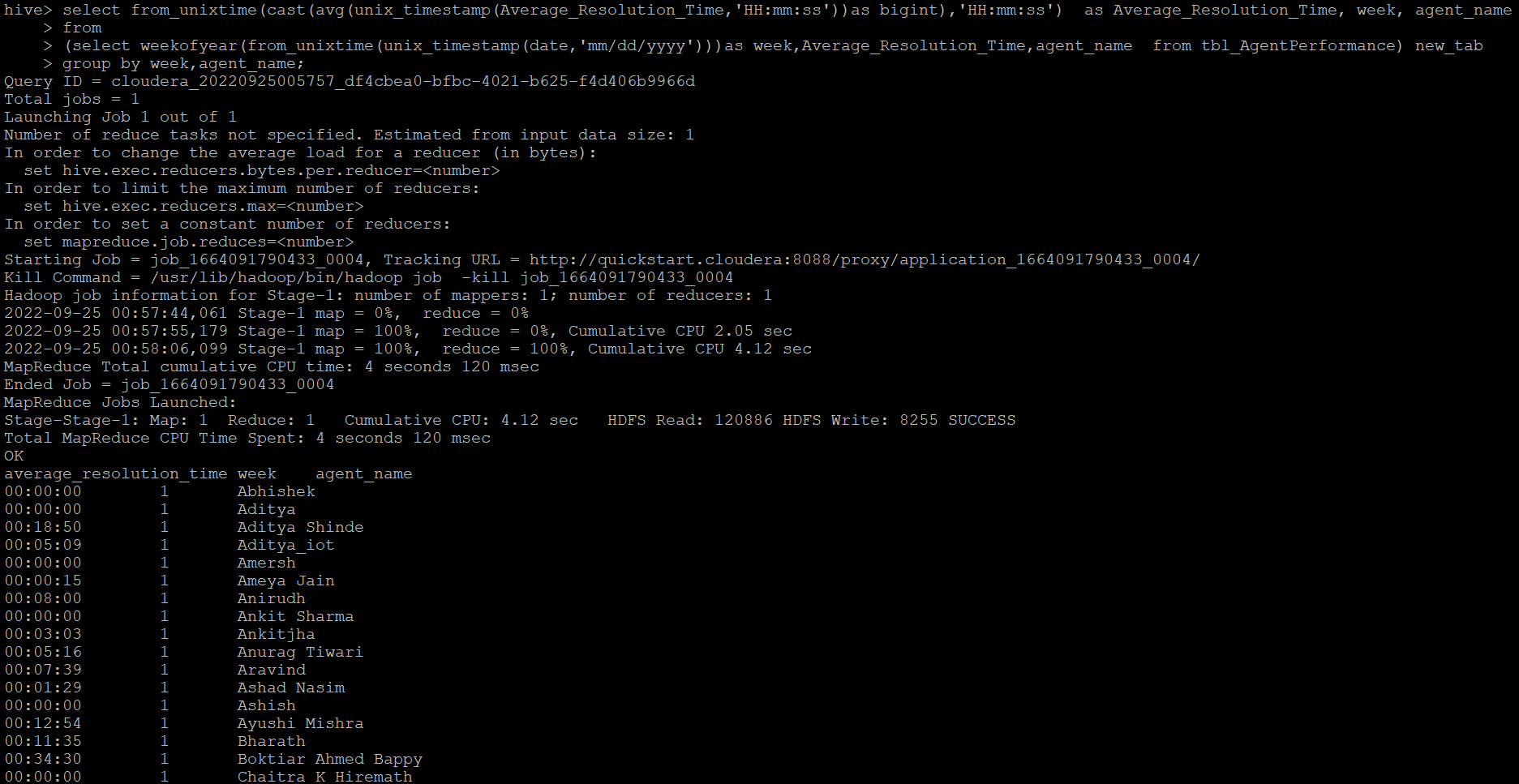
from

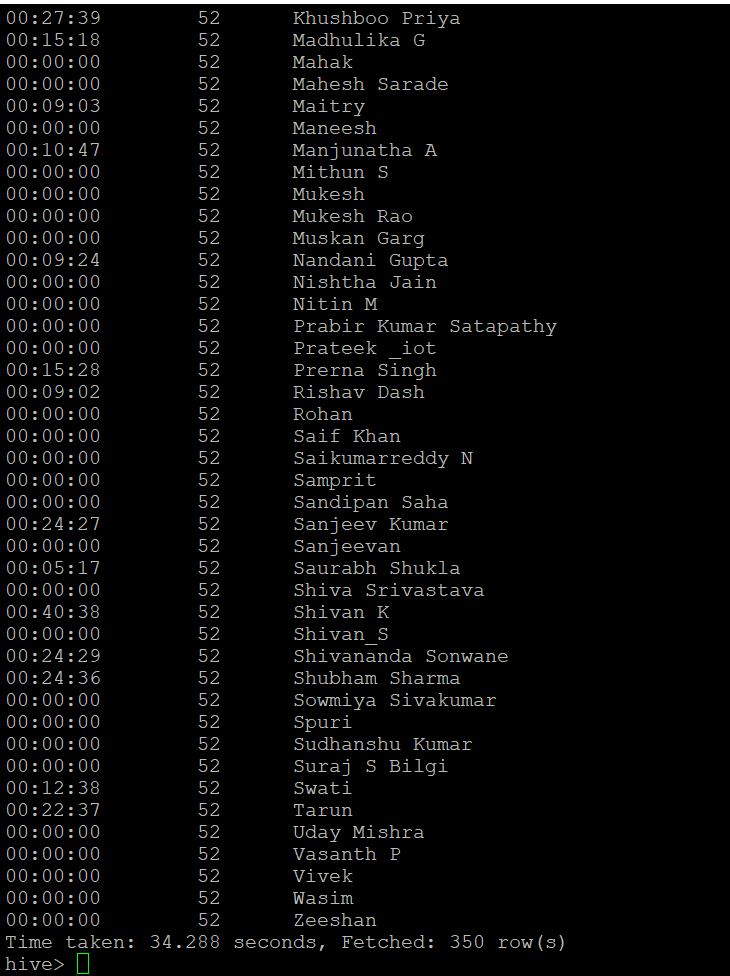
(select weekofyear(from\_unixtime(unix\_timestamp(date,'mm/dd/yyyy')))as week,Average\_Resolution\_Time,agent\_name from tbl\_AgentPerformance) new\_tab

group by week,agent\_name;

**CLI Output –**

Complete output



**Problem Statement 14-**

**Find the number of chat on which they have received a feedback**

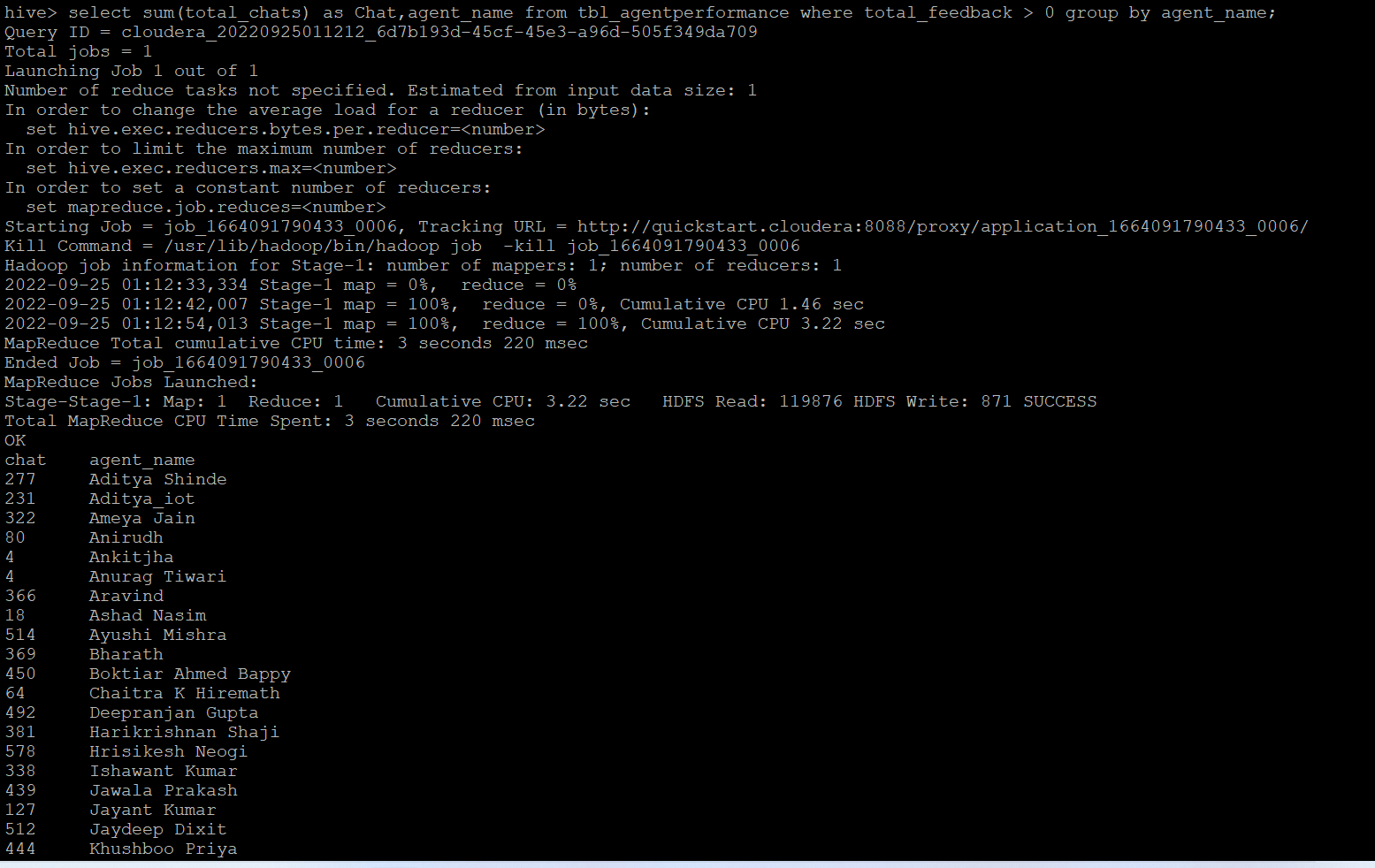
**Solution –**

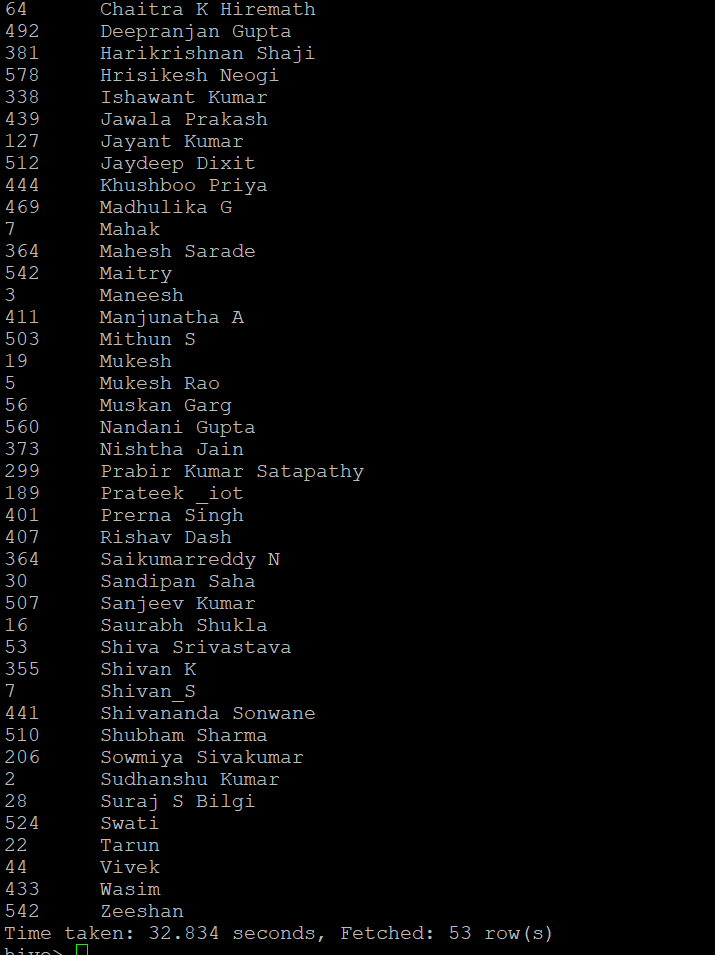
*As per problem statement ‘they’ means “agent” so we have found the feedback per ‘agent ‘*

select sum(total\_chats) as Chat,agent\_name from tbl\_agentperformance where total\_feedback > 0 group by agent\_name;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below





**Problem Statement 15-**

**Total contribution hour for each and every agents weekly basis**

**Solution –** *we have used unix\_timestamp and from\_unixtime function to convet string into time or date data. And Cast function to convert date/time data after aggregation into BigInt and then to string for print purpose*

select from\_unixtime(cast(avg(unix\_timestamp(duration,'HH:mm:ss'))as bigint),'HH') as contribution\_in\_Hours , week, agent as agent\_name

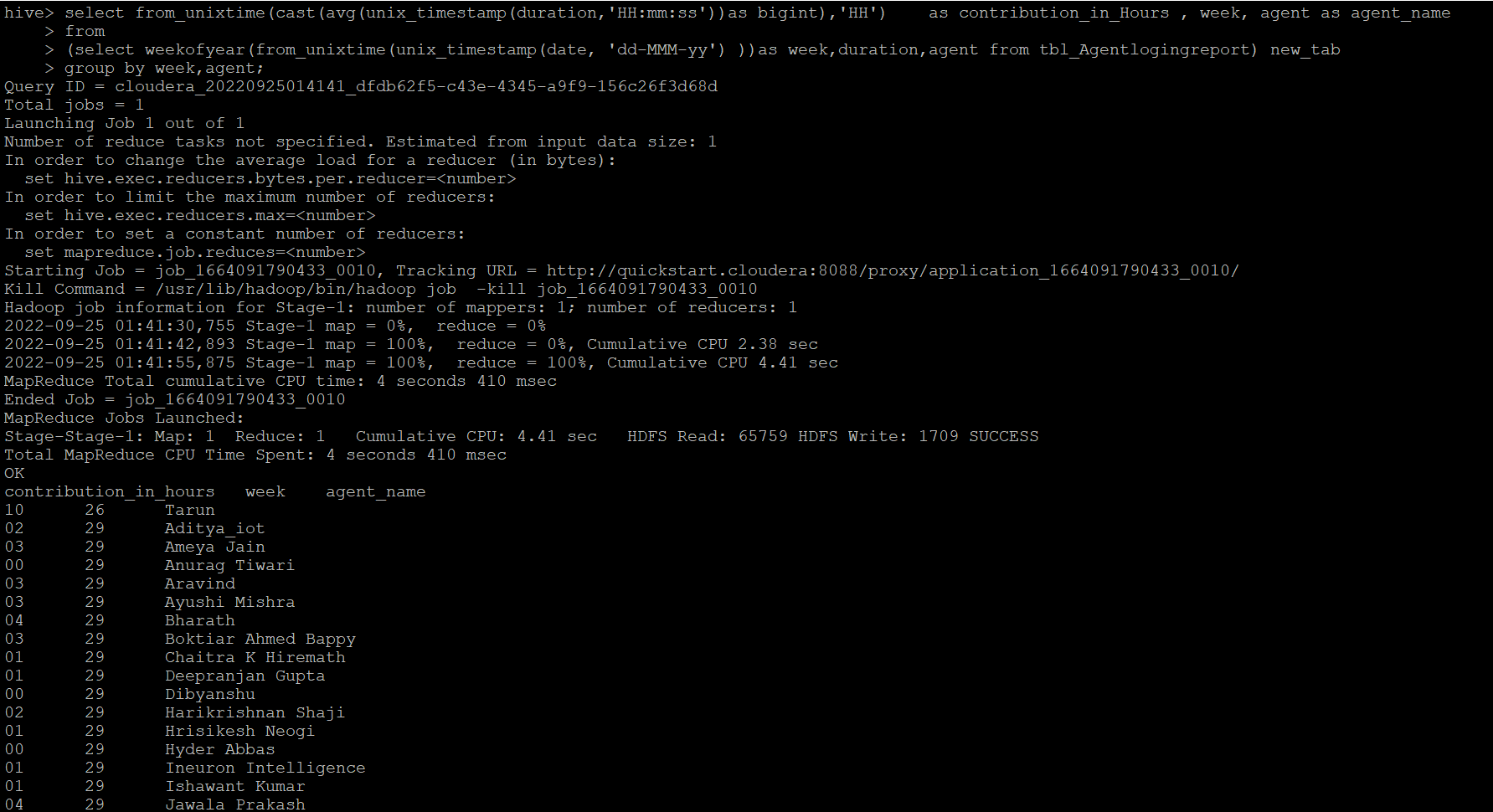
from

(select weekofyear(from\_unixtime(unix\_timestamp(date, 'dd-MMM-yy') ))as week,duration,agent from tbl\_Agentlogingreport) new\_tab

group by week,agent;

**CLI Output –**

Output is too big to capture in single window so the initial and final line snapshot is attached below



****

**Problem Statement 16-**

**Perform inner join, left join and right join based on the agent column and after joining the table export that data into your local system.-**

**Solution –**

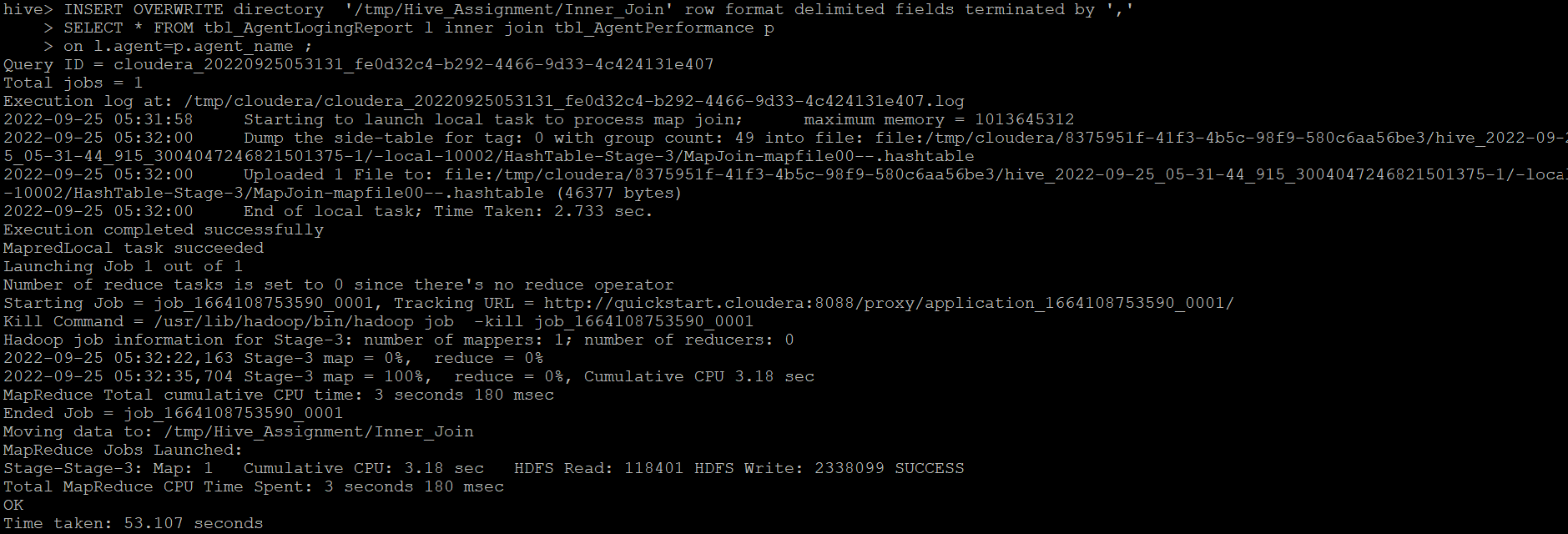
**Inner Join Query –**

INSERT OVERWRITE directory '/tmp/Hive\_Assignment/Inner\_Join' row format delimited fields terminated by ','

SELECT \* FROM tbl\_AgentLogingReport l inner join tbl\_AgentPerformance p

on l.agent=p.agent\_name ;

**CLI Output –**



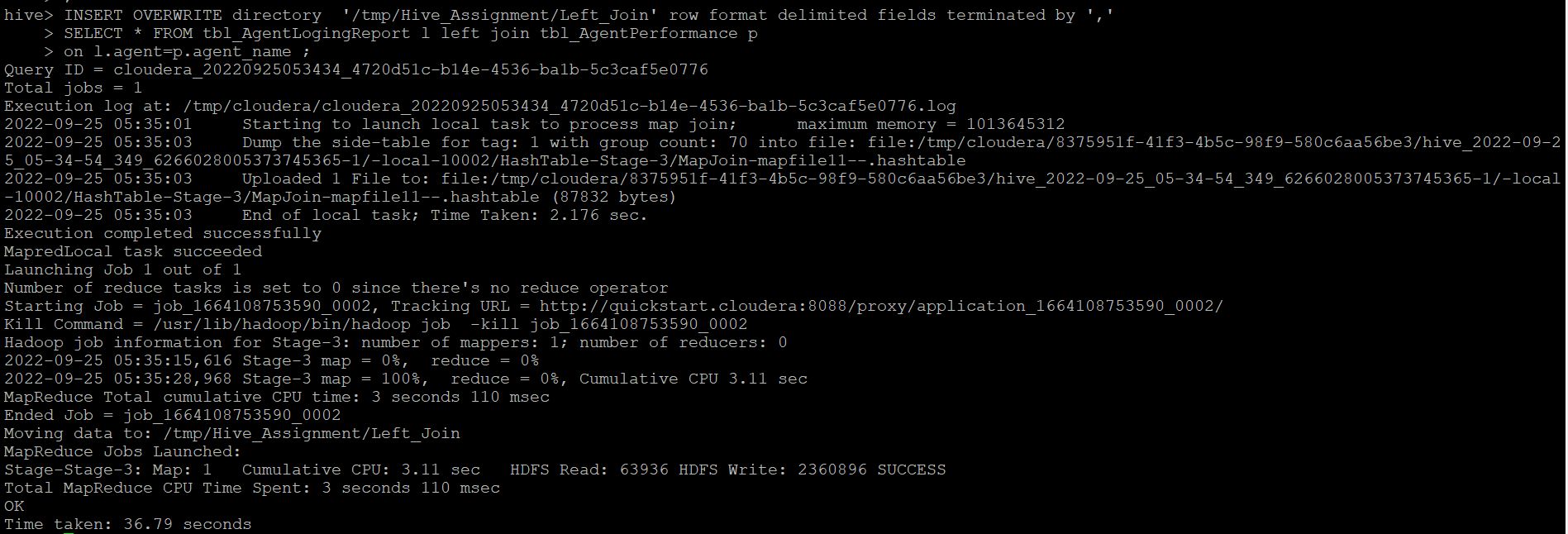
**Left Join Query –**

INSERT OVERWRITE directory '/tmp/Hive\_Assignment/Left\_Join' row format delimited fields terminated by ','

SELECT \* FROM tbl\_AgentLogingReport l left join tbl\_AgentPerformance p

on l.agent=p.agent\_name ;

**CLI Output –**



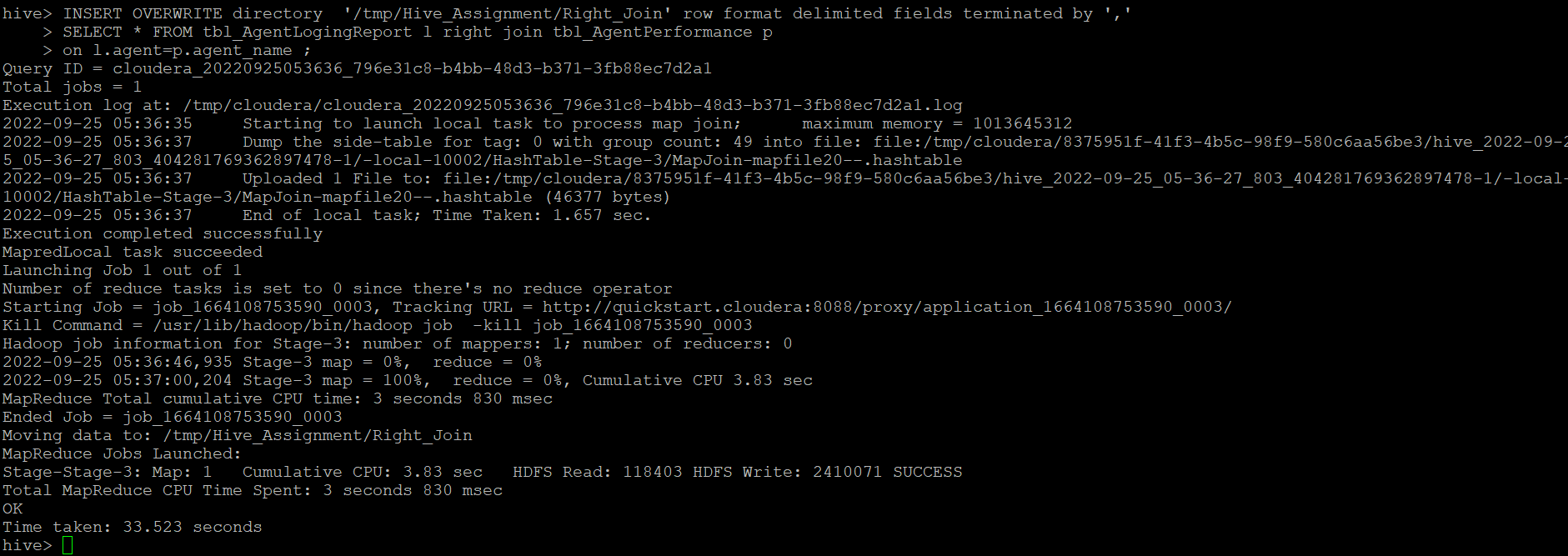
**Right Join Query –**

INSERT OVERWRITE directory '/tmp/Hive\_Assignment/Right\_Join' Join' row format delimited fields terminated by ','

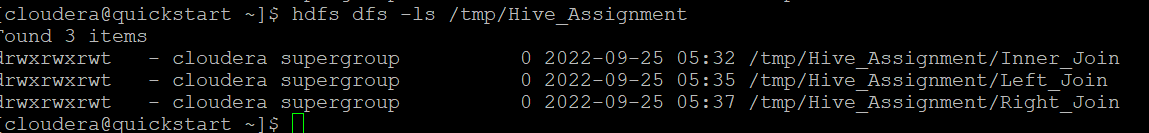
SELECT \* FROM tbl\_AgentLogingReport l right join tbl\_AgentPerformance p

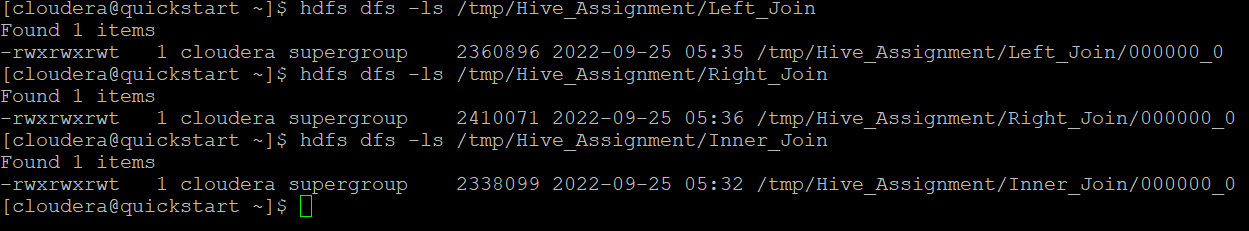
on l.agent=p.agent\_name ;

**CLI Output –**

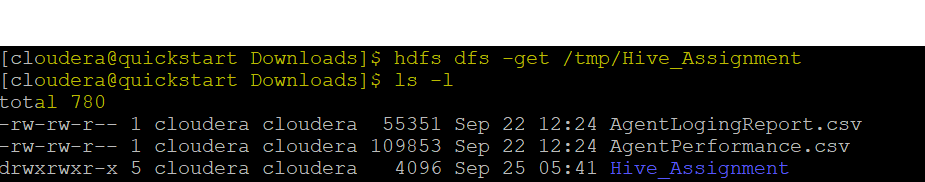


**Files stored in HDFS location -**

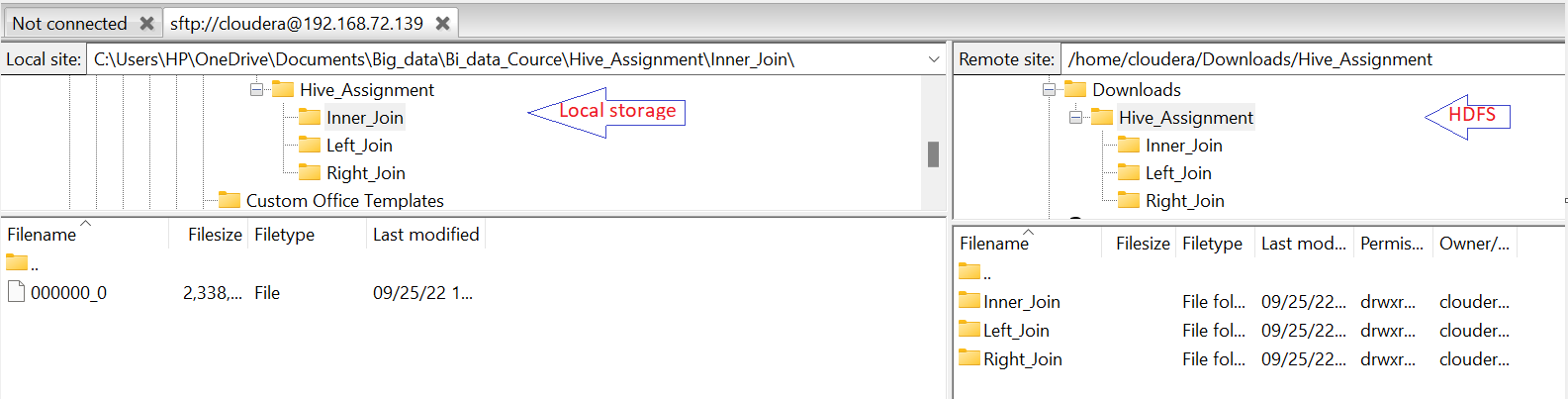




**Files loaded into ClouderaVM from HDFS location -**



**Files loaded into local Pc using FileZilla-**



**Problem Statement 17-**

**Perform partitioning on top of the agent column and then on top of that perform bucketing for each partitioning.**

**Solution –**

**Table 1 Partitioning and Bucketing –**

create table tbl\_AgentLogingReport\_Partion\_buck(

SL\_No int,

Date string,

Login\_Time string,

Logout\_Time string,

Duration string)

partitioned by (Agent string)

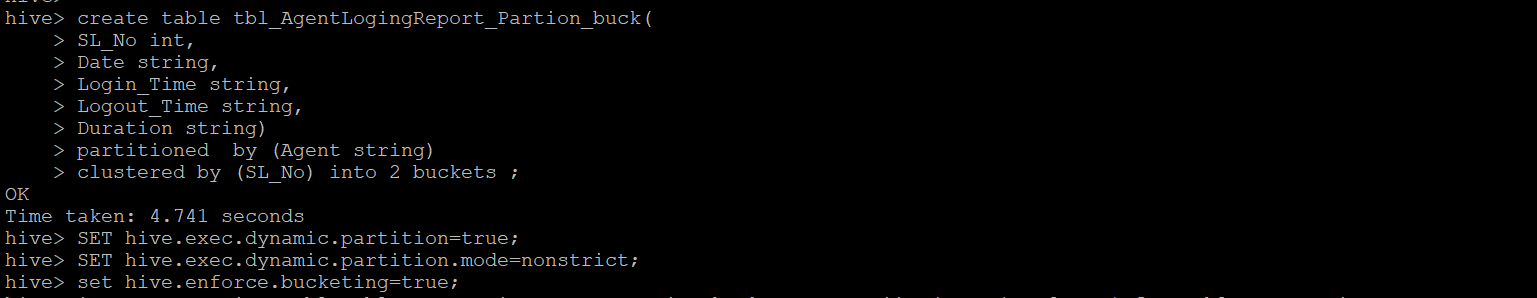
clustered by (SL\_No) into 2 buckets ;

**Set Necessary properties –**

SET hive.enforce.bucketing=true;

SET hive.exec.dynamic.partition=true;

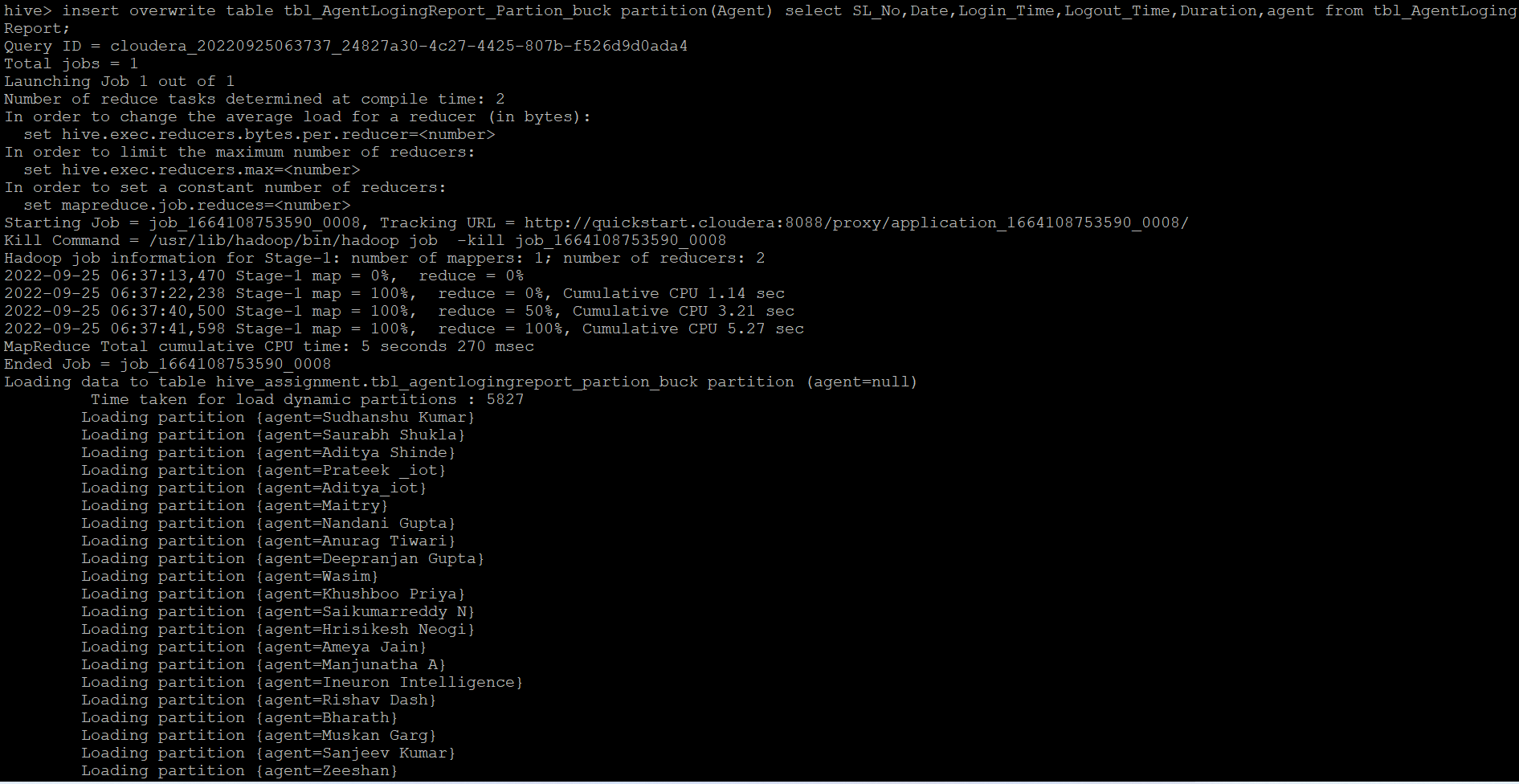
SET hive.exec.dynamic.partition.mode=nonstrict;

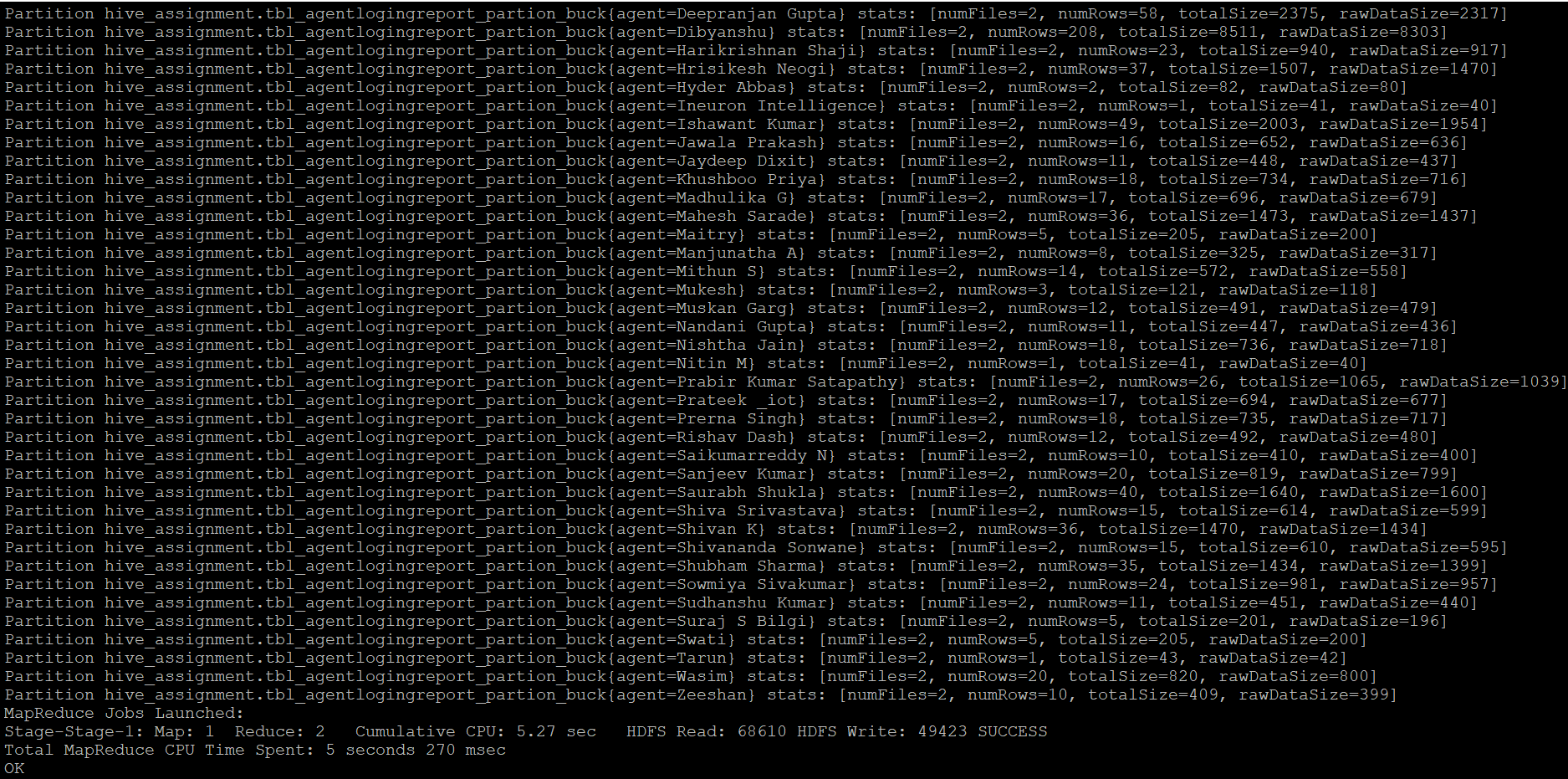


**Data Insert from stg table to Partitioning and Bucketing table –**

insert overwrite tbl\_AgentLogingReport\_Partion\_buck partition(Agent) select SL\_No,Date, Login\_Time, Logout\_Time, Duration,agent from tbl\_AgentLogingReport;

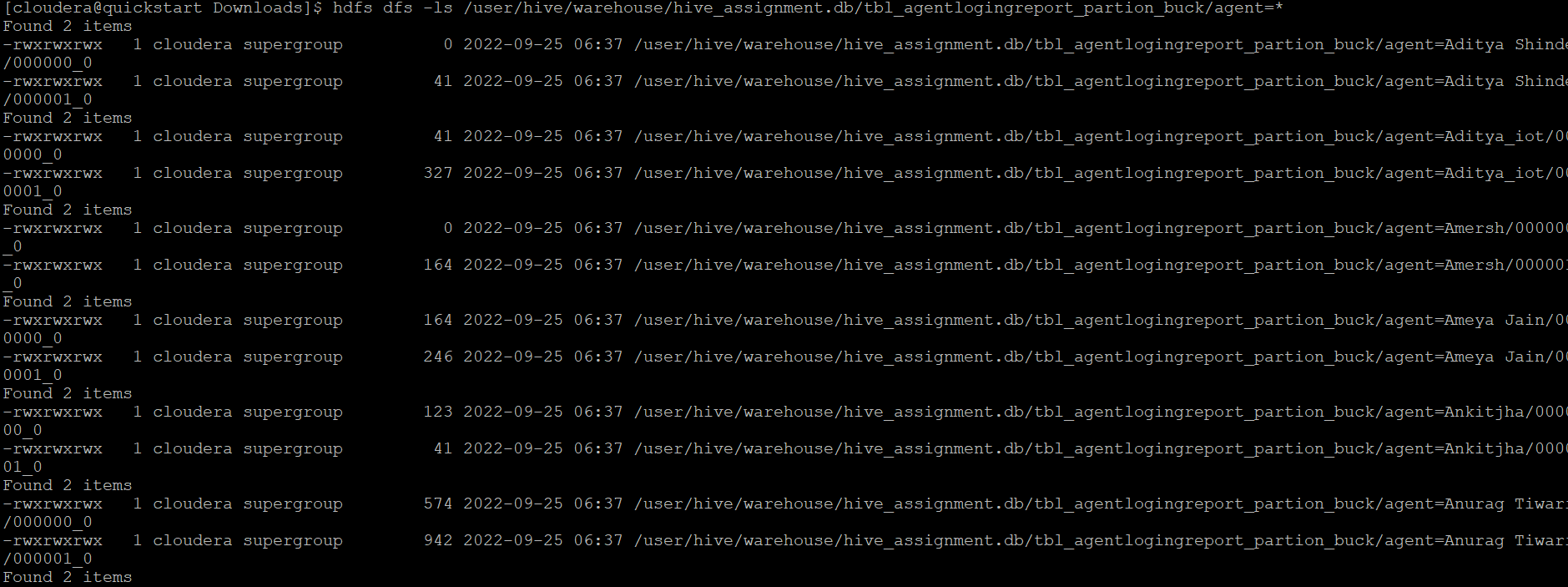
**CLI Output –**





**Partitioning and Bucketing got created at Table location HDFS –**

hdfs dfs -ls /user/hive/warehouse/hive\_assignment.db/tbl\_AgentLogingReport\_Partion\_buck/agent=\*



**Table 2 Partitioning and Bucketing –**

create table tbl\_AgentPerformance\_Partion\_buck

(

SL\_No int,

Date string,

Total\_Chats int,

Average\_Response\_Time string,

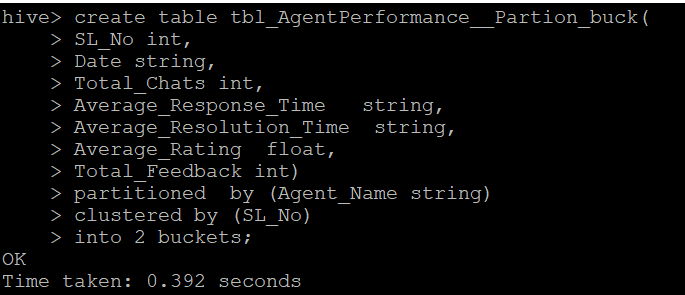
Average\_Resolution\_Time string,

Average\_Rating float,

Total\_Feedback int)

partitioned by (Agent\_Name string)

clustered by (SL\_No) into 2 buckets;

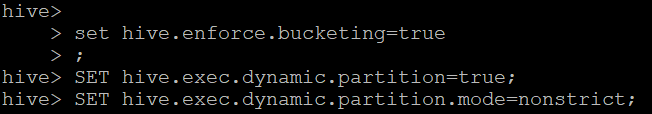


**Set Necessary properties –**

SET hive.enforce.bucketing=true;

SET hive.exec.dynamic.partition=true;

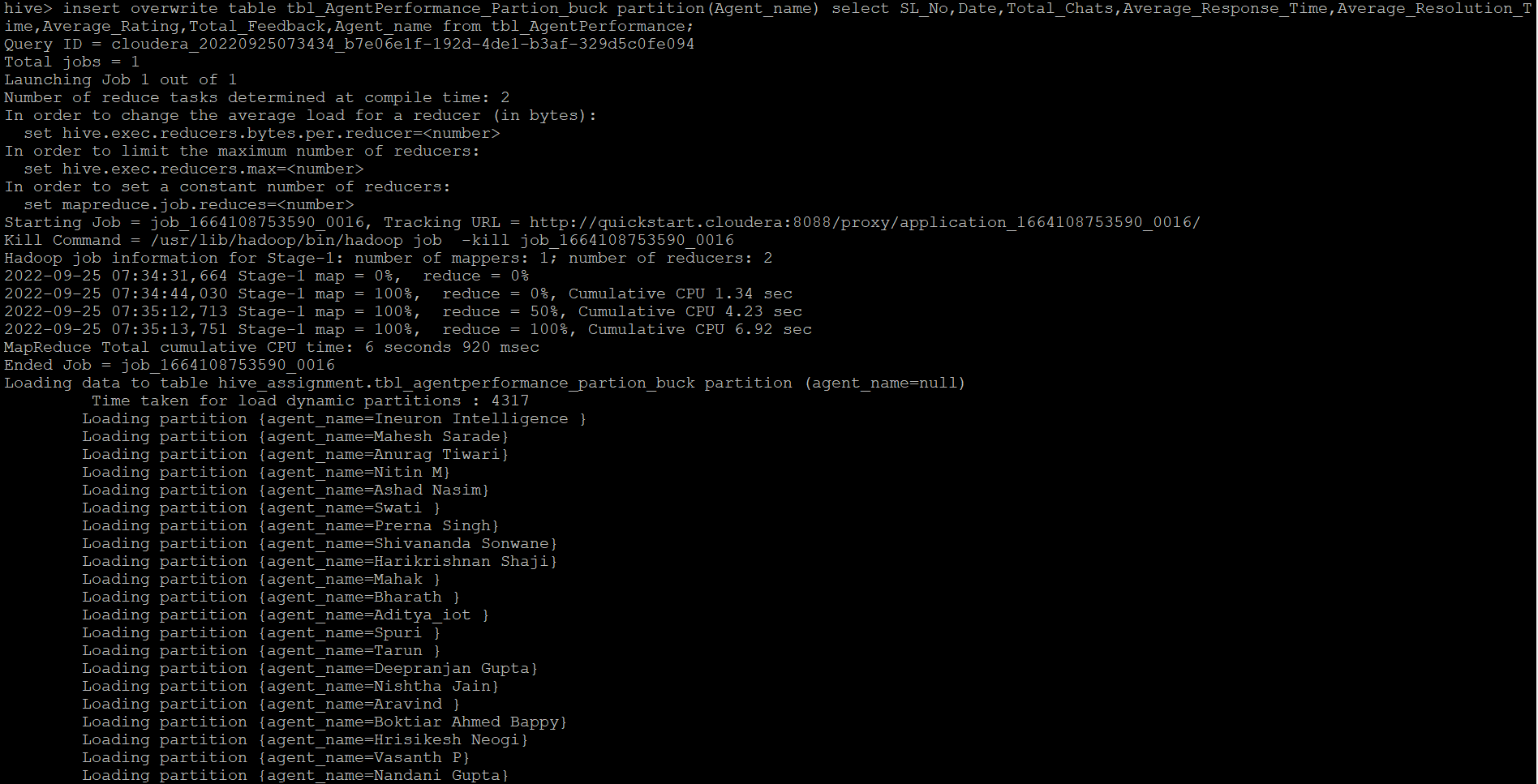
SET hive.exec.dynamic.partition.mode=nonstrict;

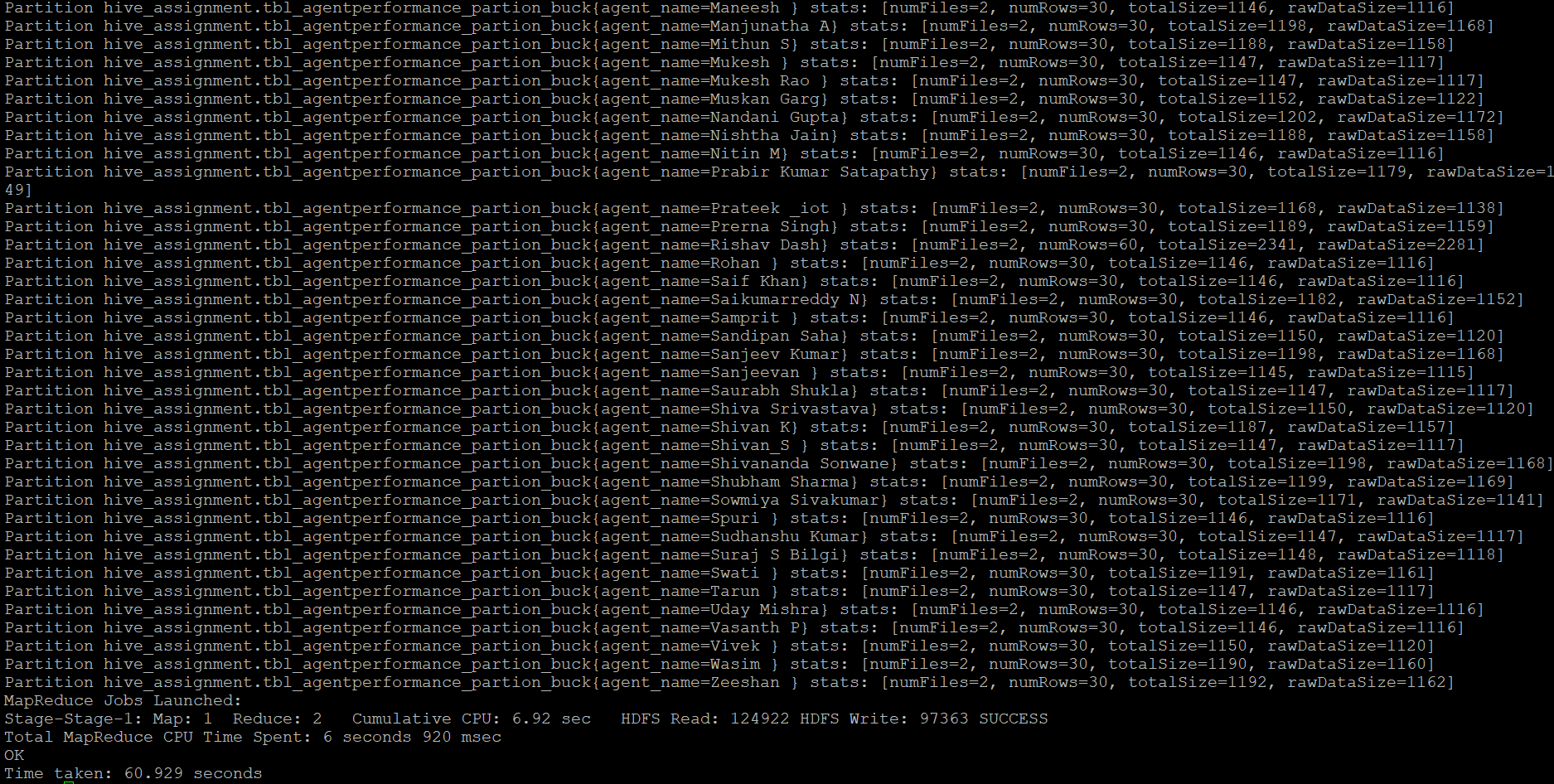


**Data Insert from stg table to Partitioning and Bucketing table –**

insert overwrite table tbl\_AgentPerformance\_Partion\_buck partition(Agent\_name) select SL\_No,Date,Total\_Chats,Average\_Response\_Time,Average\_Resolution\_Time,Average\_Rating,Total\_Feedback,Agent\_name from tbl\_AgentPerformance;

**CLI Output –**





**Partitioning and Bucketing got created at Table location HDFS –**

hdfs dfs -ls /user/hive/warehouse/hive\_assignment.db/tbl\_agentperformance\_partion\_buck/agent\_name\*

