**Mini Project-1**

1. **Create a schema based on the given dataset**
2. create table agent\_login(

sl\_no int

agent\_name string

day date

login\_time string

logout\_time string

duration string

)

row format delimited

fields terminated by ','

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

1. create table agent\_performance(

sl\_no int

day date

agent\_name string

total\_chats int

average\_response\_time string

average\_resolution\_time string

average\_rating float

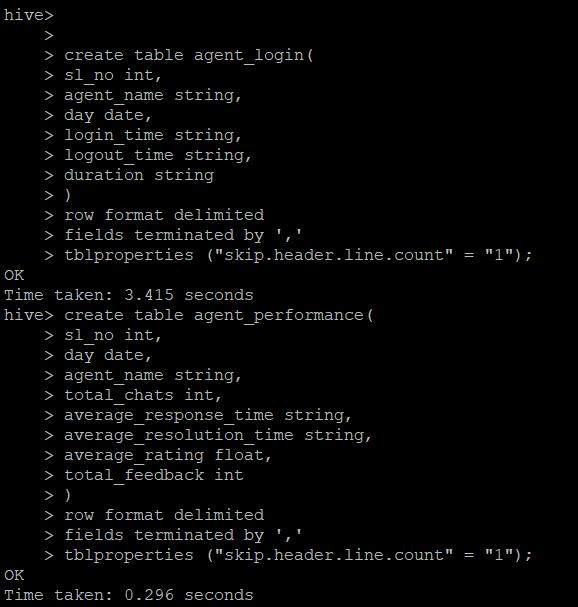
total\_feedback int

)

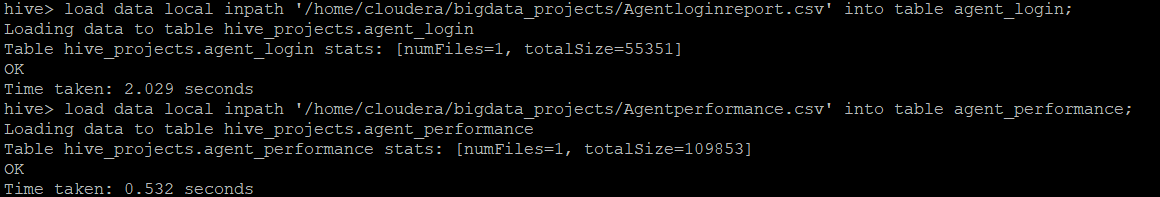
row format delimited

fields terminated by ','

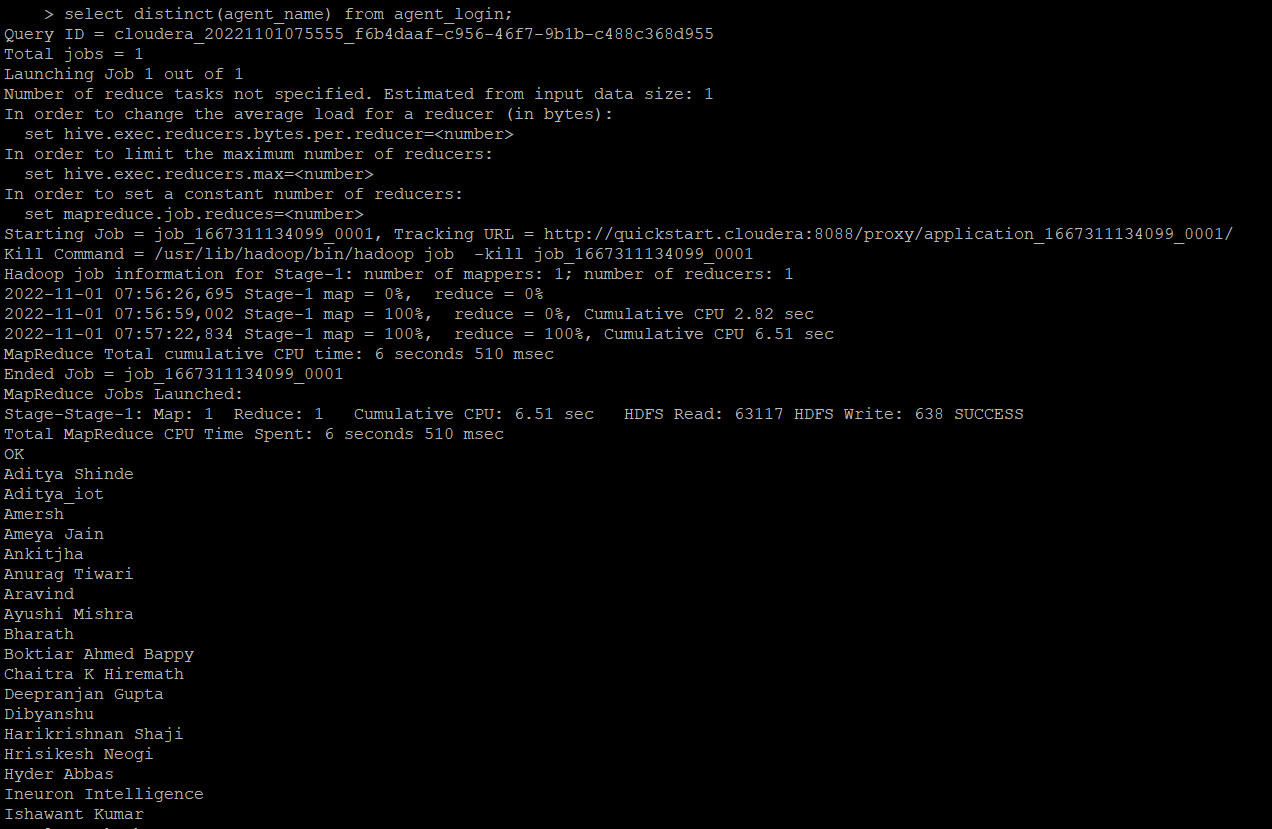
tblproperties ("skip.header.line.count" = "1");

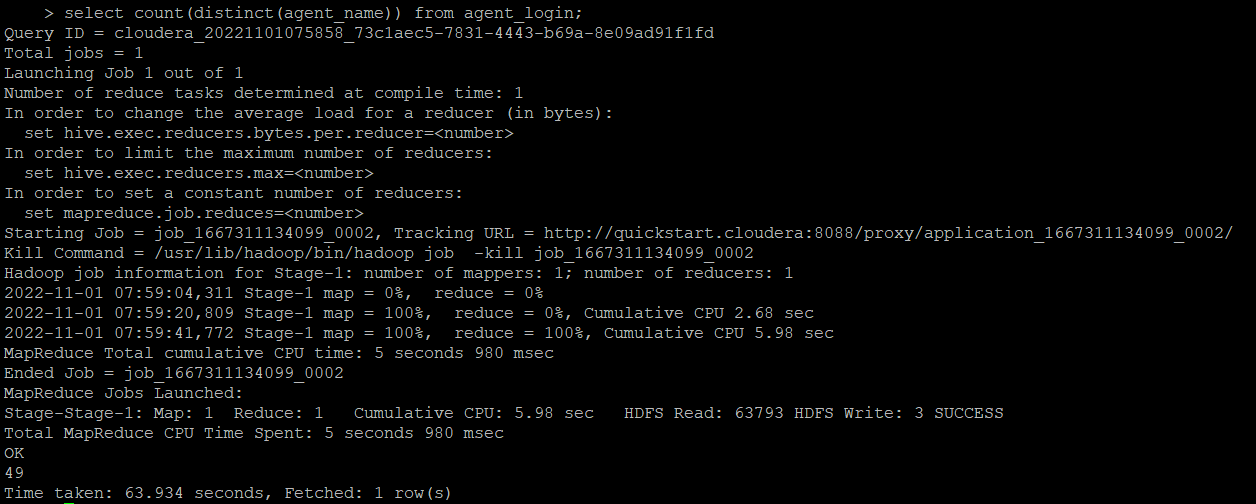


1. **Dump the data inside the hdfs in the given schema location.**

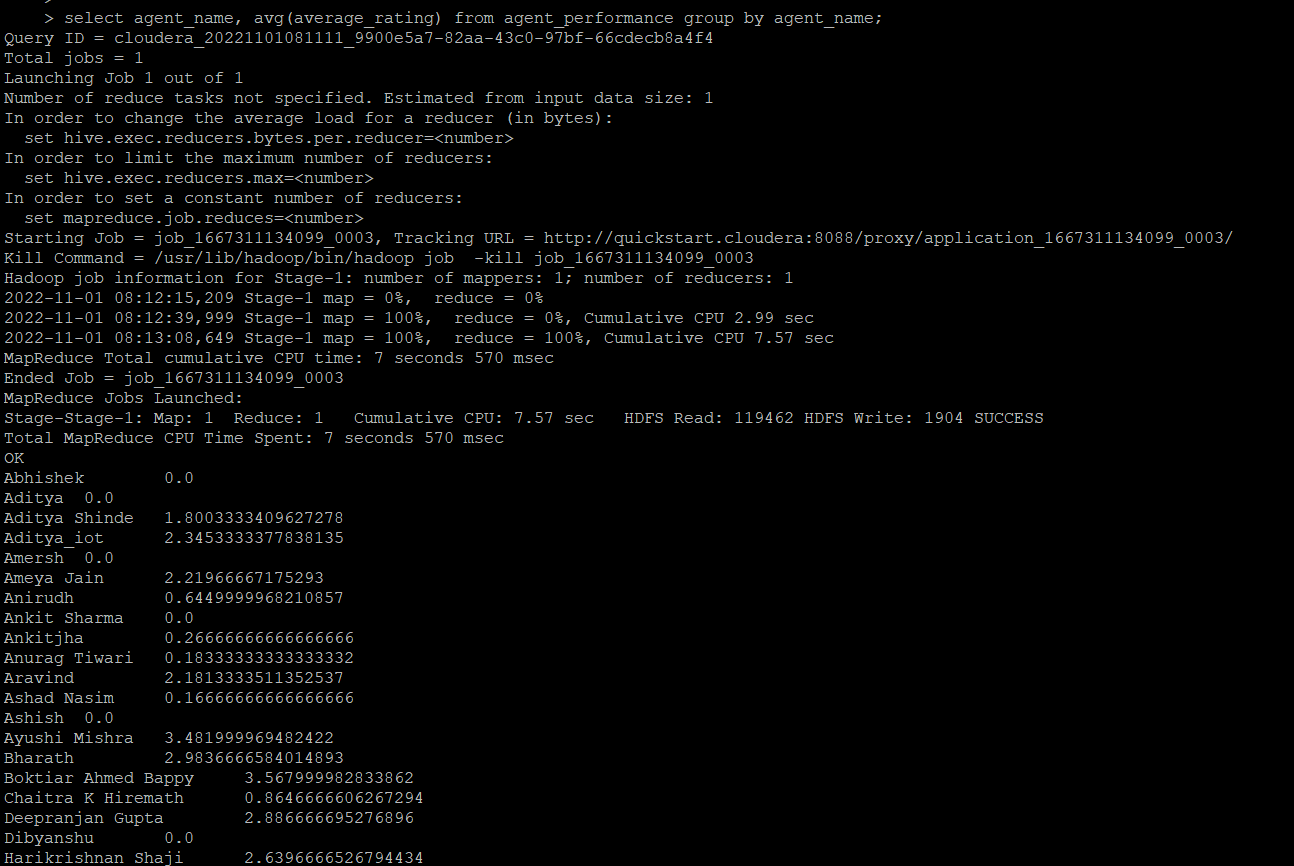


1. **List of all agents' names.**



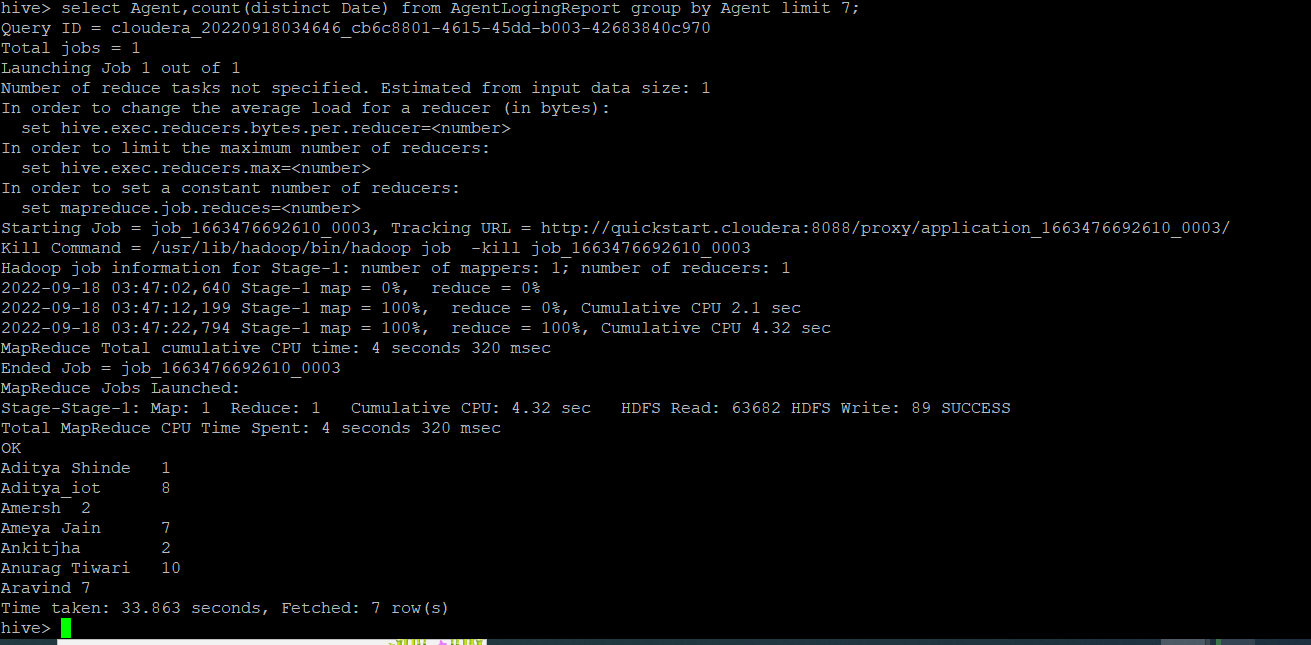


1. **Find out agent average rating.**



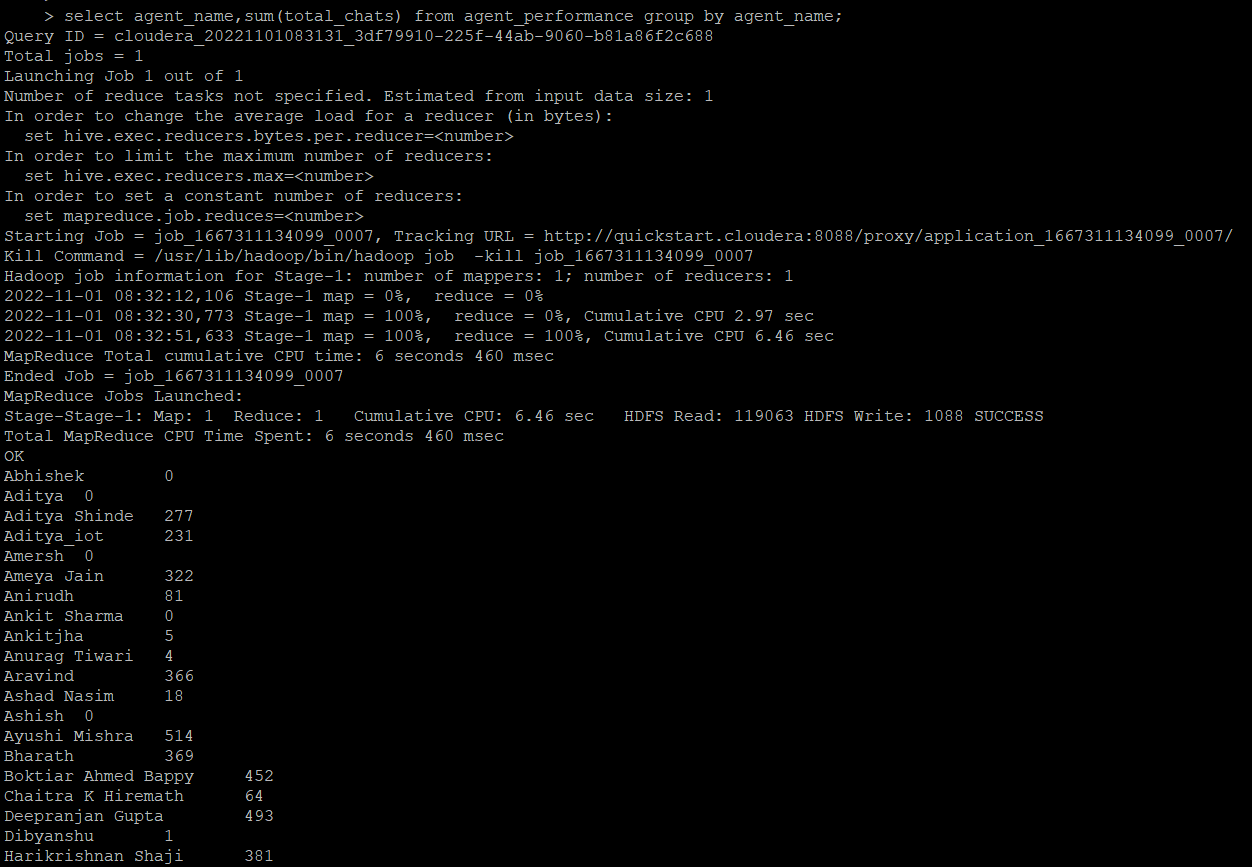
1. **Total working days for each agents**

select agent\_name,count(distinct day) from agent\_login group by agent\_name;



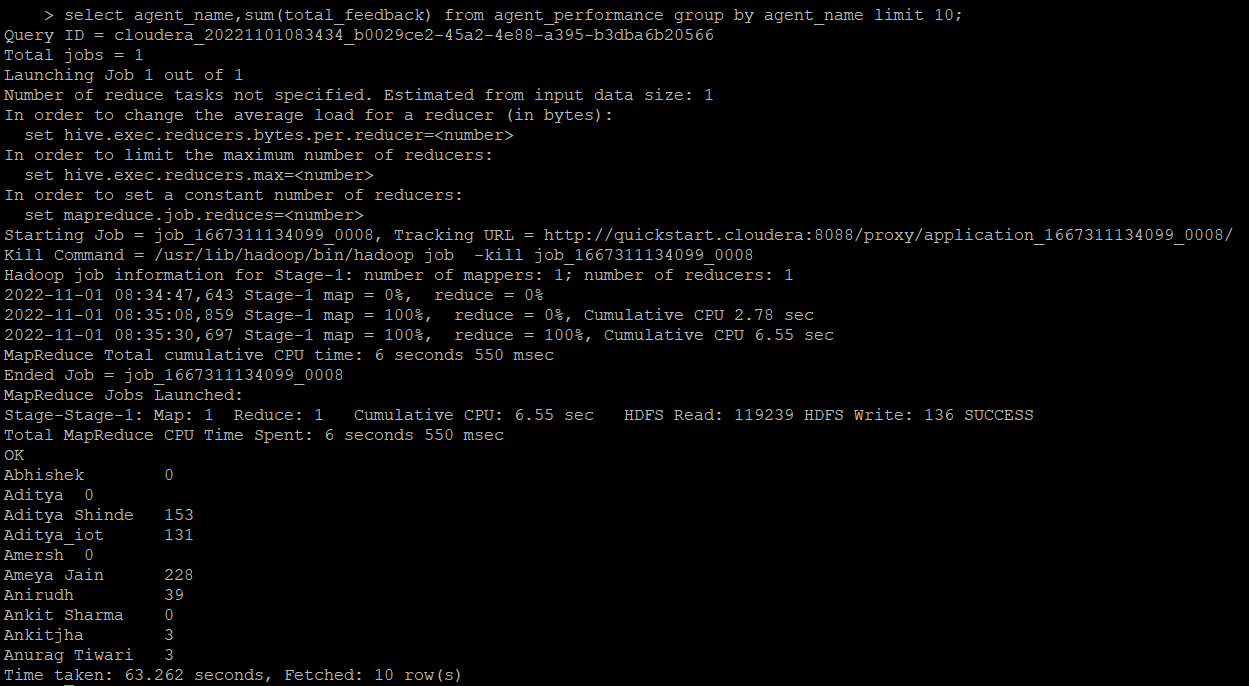
1. **Total query that each agent have taken**

select agent\_name,sum(total\_chats) from agent\_performance group by agent\_name;



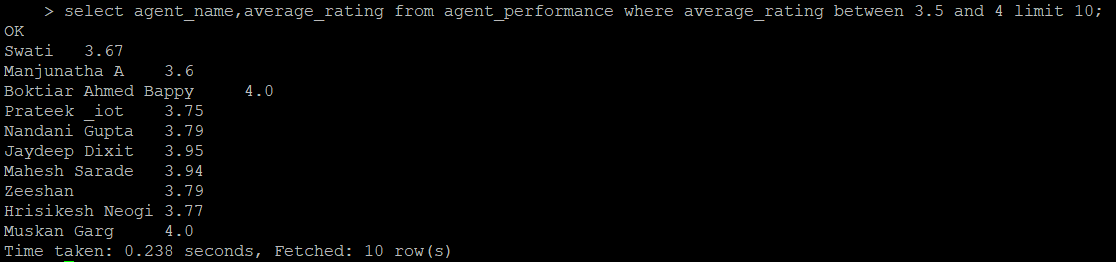
1. **Total Feedback that each agent have received**

select agent\_name,sum(total\_feedback) from agent\_performance group by agent\_name;



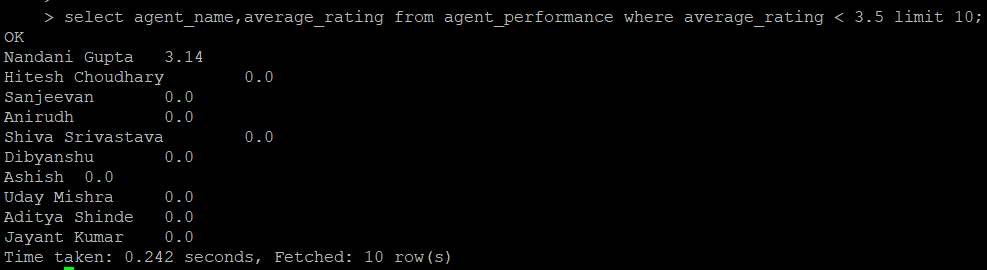
1. **Agent name who have average rating between 3.5 to 4**

select agent\_name,average\_rating from agent\_performance where average\_rating between 3.5 and 4;



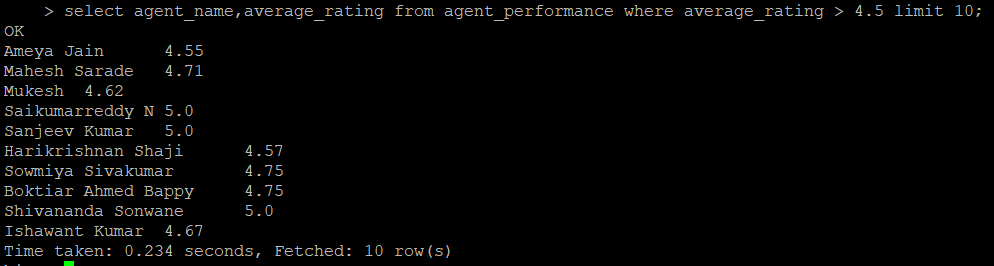
1. **Agent name who have rating less than 3.5**

select agent\_name,average\_rating from agent\_performance where average\_rating < 3.5;



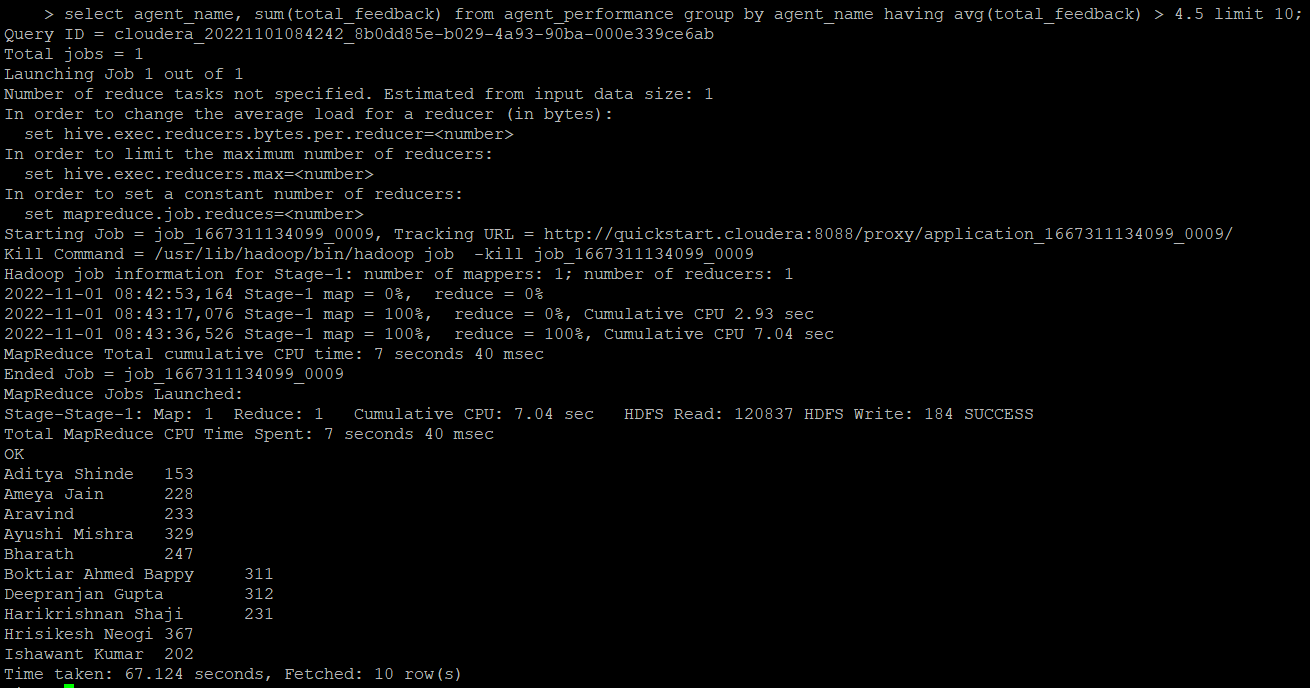
1. **Agent name who have rating more than 4.5**

select agent\_name,average\_rating from agent\_performance where average\_rating > 4.5;



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

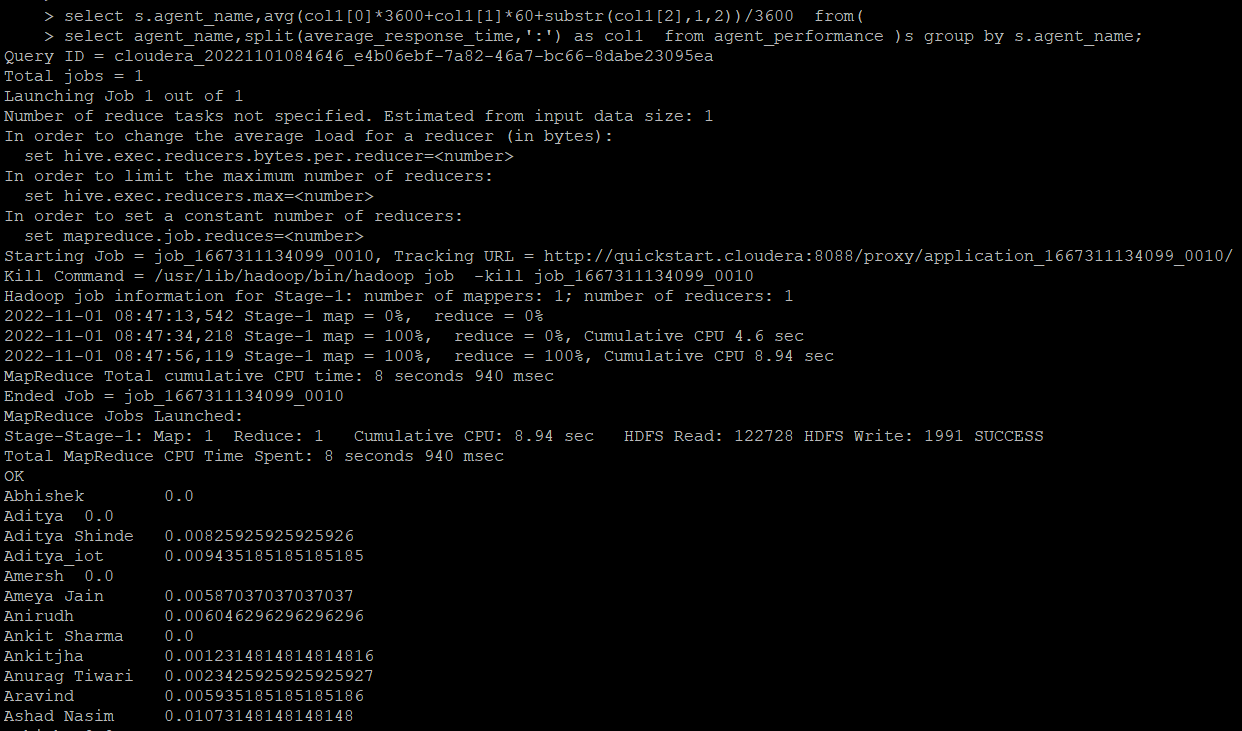
select agent\_name, sum(total\_feedback) from agent\_performance group by agent\_name having avg(total\_feedback) > 4.5;



1. **average weekly response time for each agent**

select s.agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/3600 from(

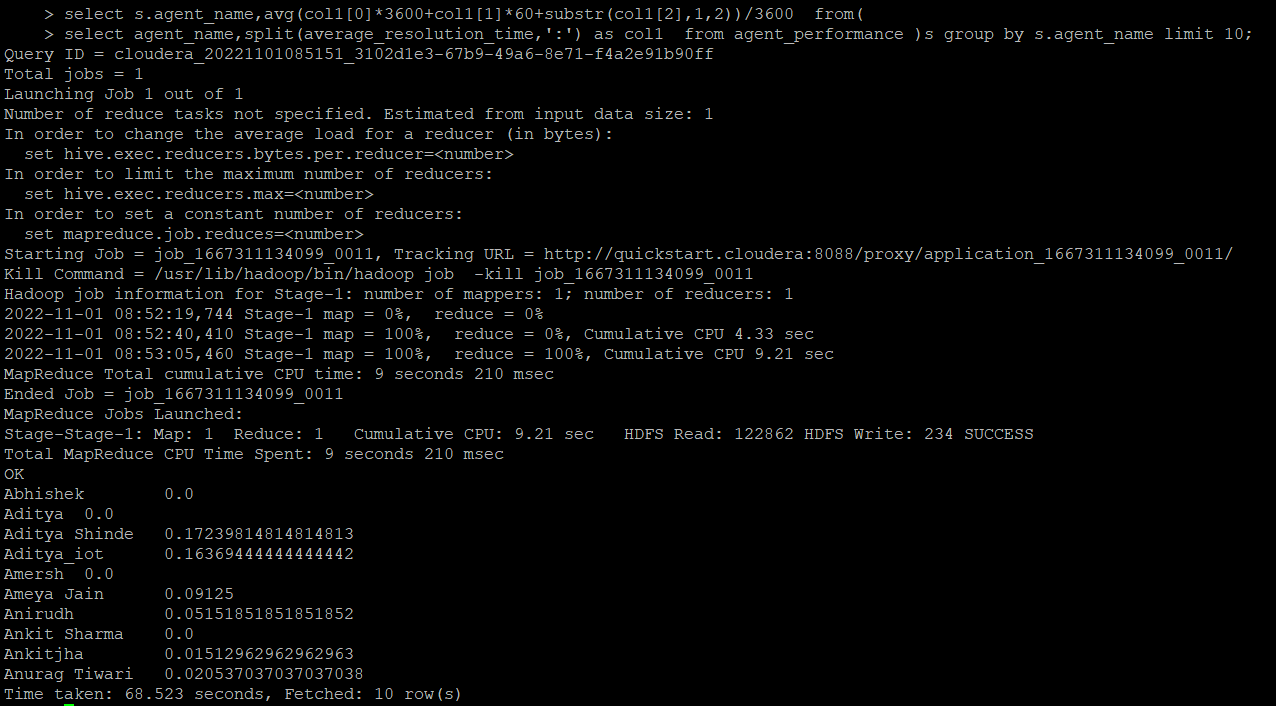
select agent\_name,split(average\_response\_time,':') as col1 from agent\_performance )s group by s.agent\_name;



1. **average weekly resolution time for each agents**

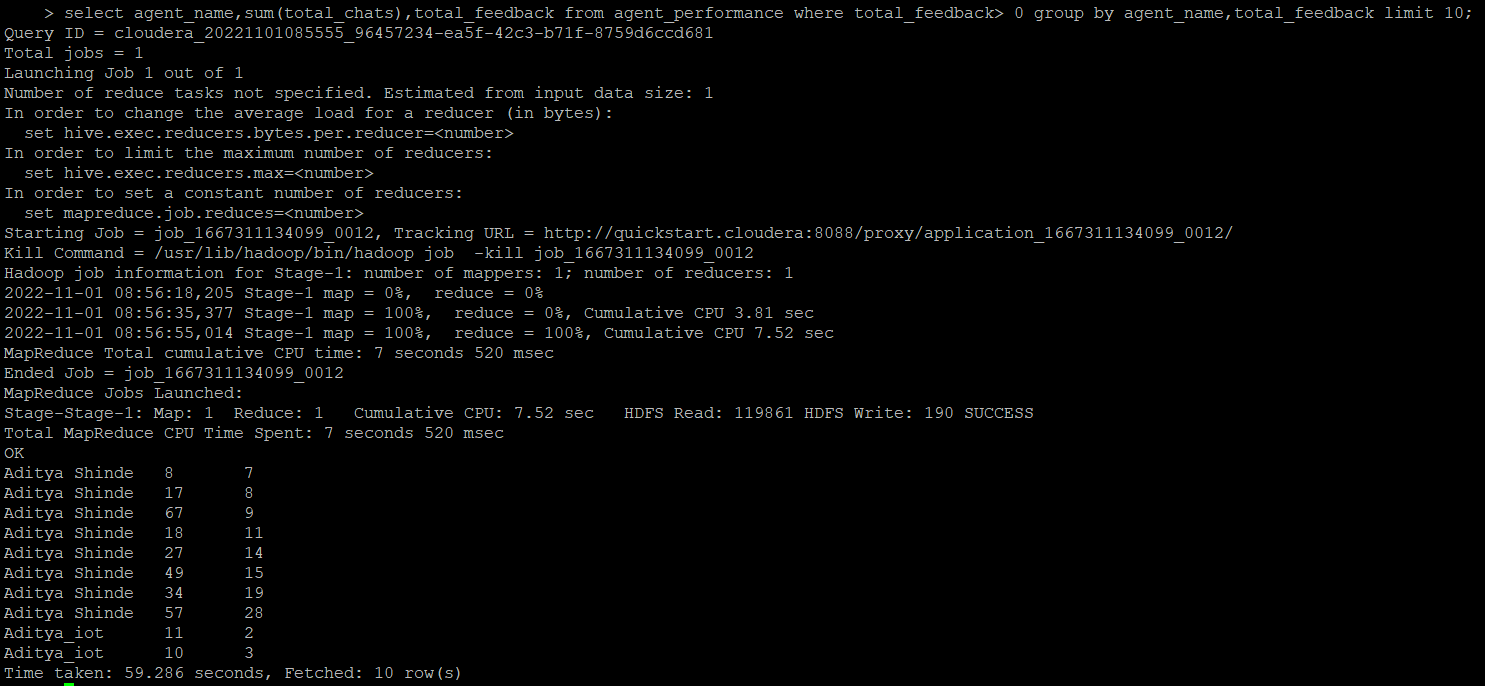
select s.agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/3600 from(

select agent\_name,split(average\_resolution\_time,':') as col1 from agent\_performance )s group by s.agent\_name;



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

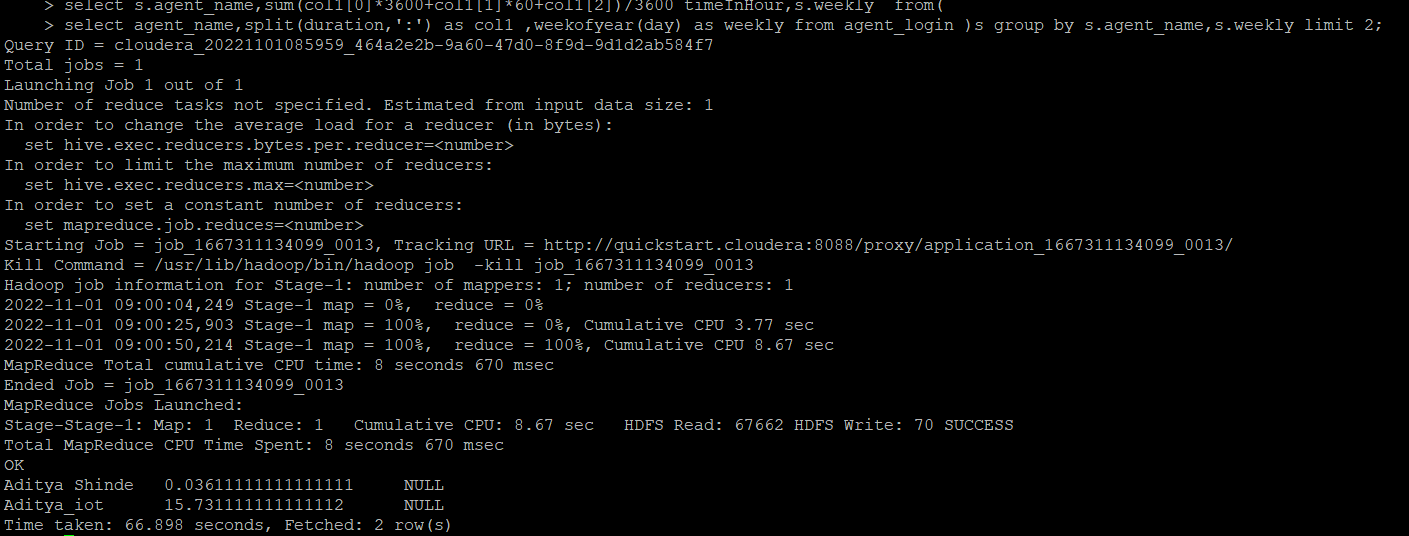
select agent\_name,sum(total\_chats),total\_feedback from agent\_performance where total\_feedback> 0 group by agent\_name,total\_feedback;



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

select s.agent\_name,sum(col1[0]\*3600+col1[1]\*60+col1[2])/3600 timeInHour,s.weekly from(

select agent\_name,split(duration,':') as col1 ,weekofyear(day) as weekly from agent\_login )s group by s.agent\_name,s.weekly limit 2;



1. **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.**

hive -e 'select a.agent\_name,a.day,a.duration,b.total\_chats,b.total\_feedback from challenge.agent\_login a join challenge.agent\_performance b on a.agent\_name = b.agent\_name' > /home/cloudera/bigdata\_projects/inner\_join.csv;

**left join:**

hive -e 'select a.agent\_name,a.date,a.Duration,b.total\_chats,b.total\_feedback from challenge.agent\_login a left join challenge.agent\_performance b on a.agent\_name = b.agent\_name' > /home/cloudera/bigdata\_projects/left\_join.csv;

**left join with performance improved due to /\*+ streamtable(a) \*/ hint:**

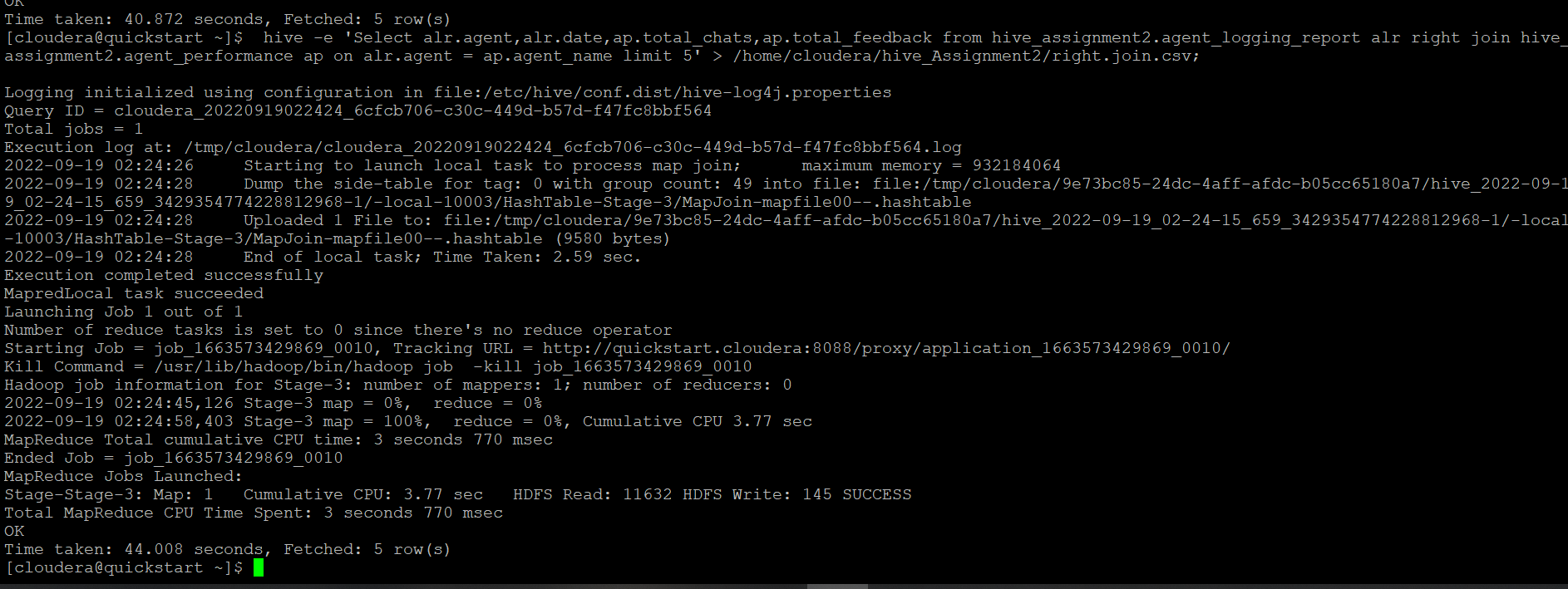
hive -e 'select /\*+ streamtable(a) \*/a.agent\_name,a.date,a.Duration,b.Total\_charts,b.total\_feedback from challenge.agent\_login a left join challenge.agent\_performance b on a.agent\_name = b.agent\_name' > /home/cloudera/bigdata\_projects/left\_join.csv;

**Right join:**

hive -e 'select a.agent\_name,a.date,a.Duration,b.Total\_charts,b.total\_feedback from challenge.agent\_login a right join challenge.agent\_performance b on a.agent\_name = b.agent\_name' > /home/cloudera/bigdata\_projects/right\_join.csv;

**Right join with performance improved due to /\*+ streamtable(a) \*/ hint:**

hive -e 'select /\*+ streamtable(a) \*/a.agent\_name,a.date,a.Duration,b.Total\_charts,b.total\_feedback from challenge.agent\_login a right join challenge.agent\_performance b on a.agent\_name = b.agent\_name' > /home/cloudera/bigdata\_projects/left\_join.csv; 17. Perform partitioning on top of the agent column and then on top of that perform bucketing for each partitioning.



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

Create table ALR\_partition\_Bucket

(

sr\_no int,

Date date,

Login\_time string,

Logout\_time string,

Duration string

)partitioned by (Agent string)

CLUSTERED BY (Date) sorted by (Date) INTO 4 BUCKETS

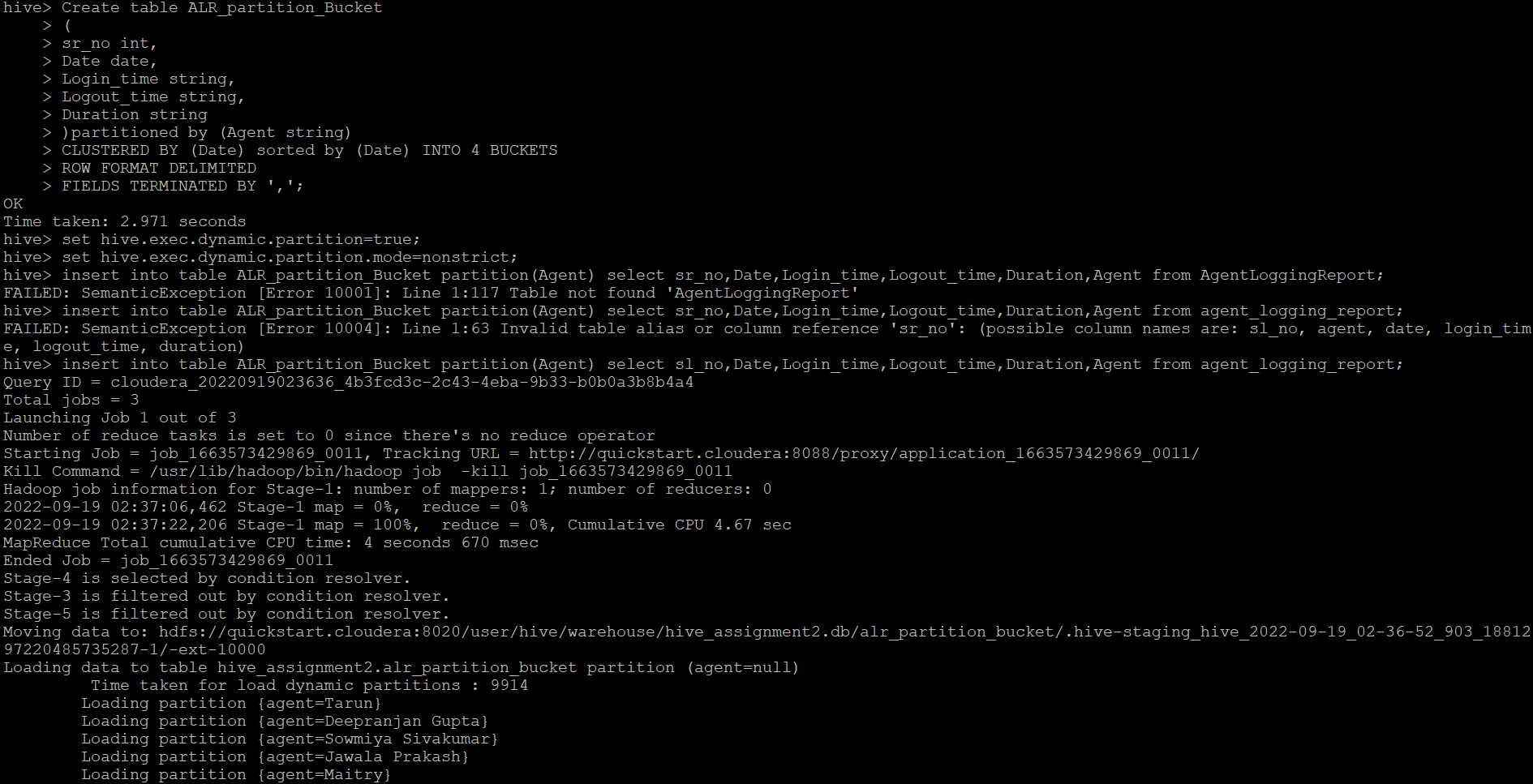
ROW FORMAT DELIMITED

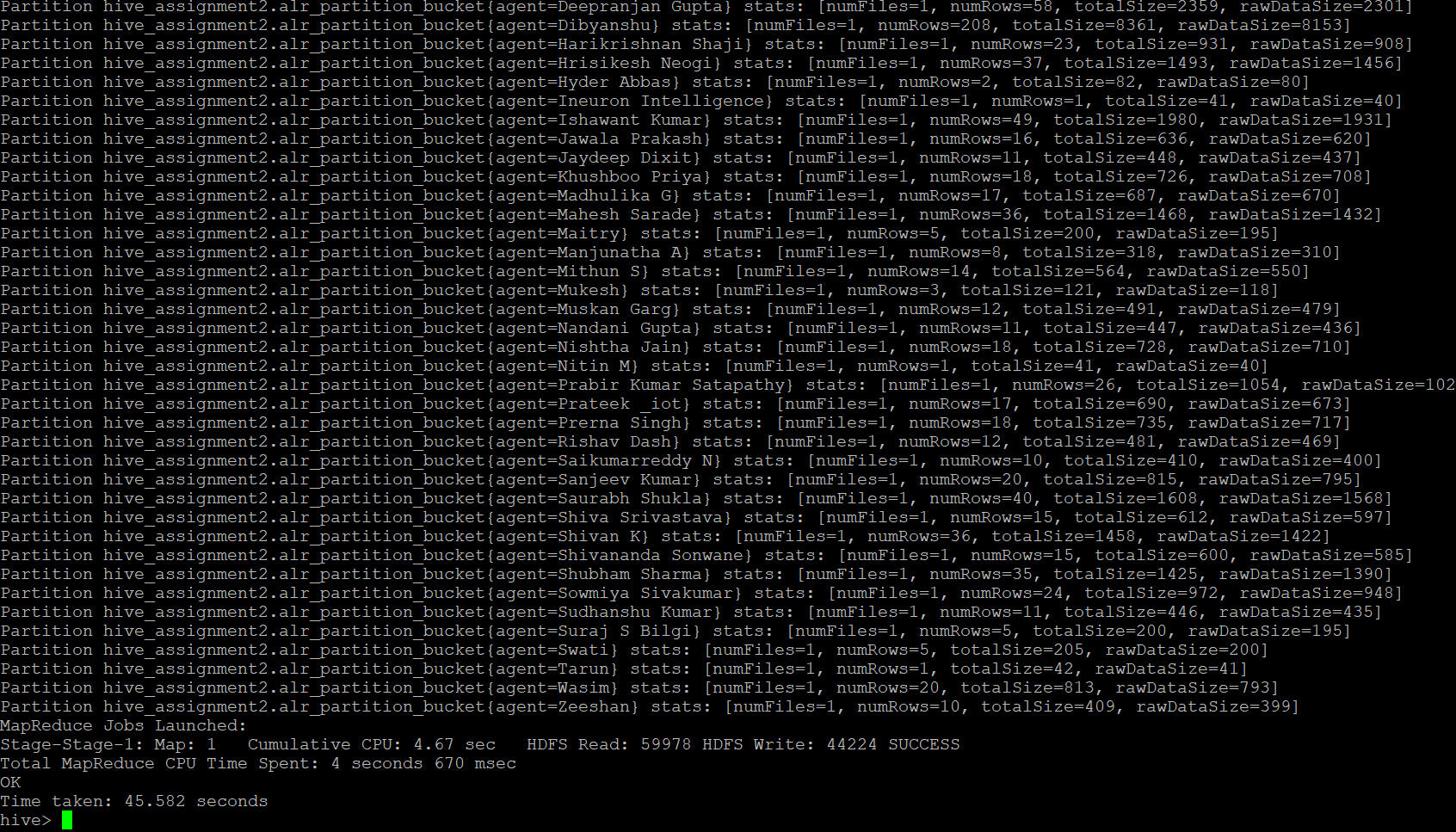
FIELDS TERMINATED BY ',';

set hive.exec.dynamic.partition=true;

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

hive> insert into table ALR\_partition\_Bucket partition(Agent) select sl\_no,Date,Login\_time,Logout\_time,Duration,Agent from agent\_logging\_report;





Create table AP\_partition\_Bucket

(

sr\_no int,

Date date,

Total\_chat string,

Average\_Response\_Time string,

Average\_Resolution\_Time string,

Average\_Rating float,

Total\_Feedback int

)partitioned by (agent\_name string)

CLUSTERED BY (Date) sorted by (Date) INTO 8 BUCKETS

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';

insert into table AP\_partition\_Bucket partition(agent\_name) select sl\_no,Date,Total\_chats, Average\_Response\_Time, Average\_Resolution,Average\_Rating,Total\_Feedback,Agent\_name from Agent\_performance;

