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

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

create table agent\_logging\_report (

sl\_no int,

Agent string,

Date date,

login\_time string,

logout\_time string,

duration string)

row format delimited

fields terminated by ','

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

);

create table agent\_performance(

sl\_no int,

Date date,

Agent\_name string,

total\_chats int,

average\_response\_time string,

average\_resolution string,

average\_rating float,

total\_feedback int

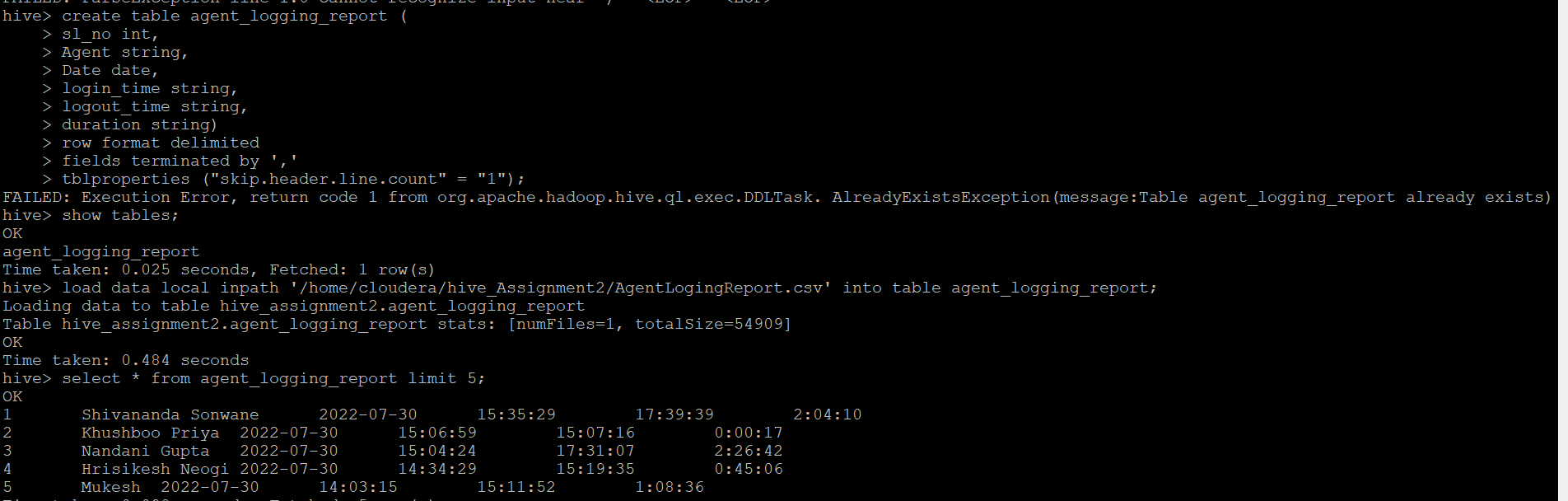
)

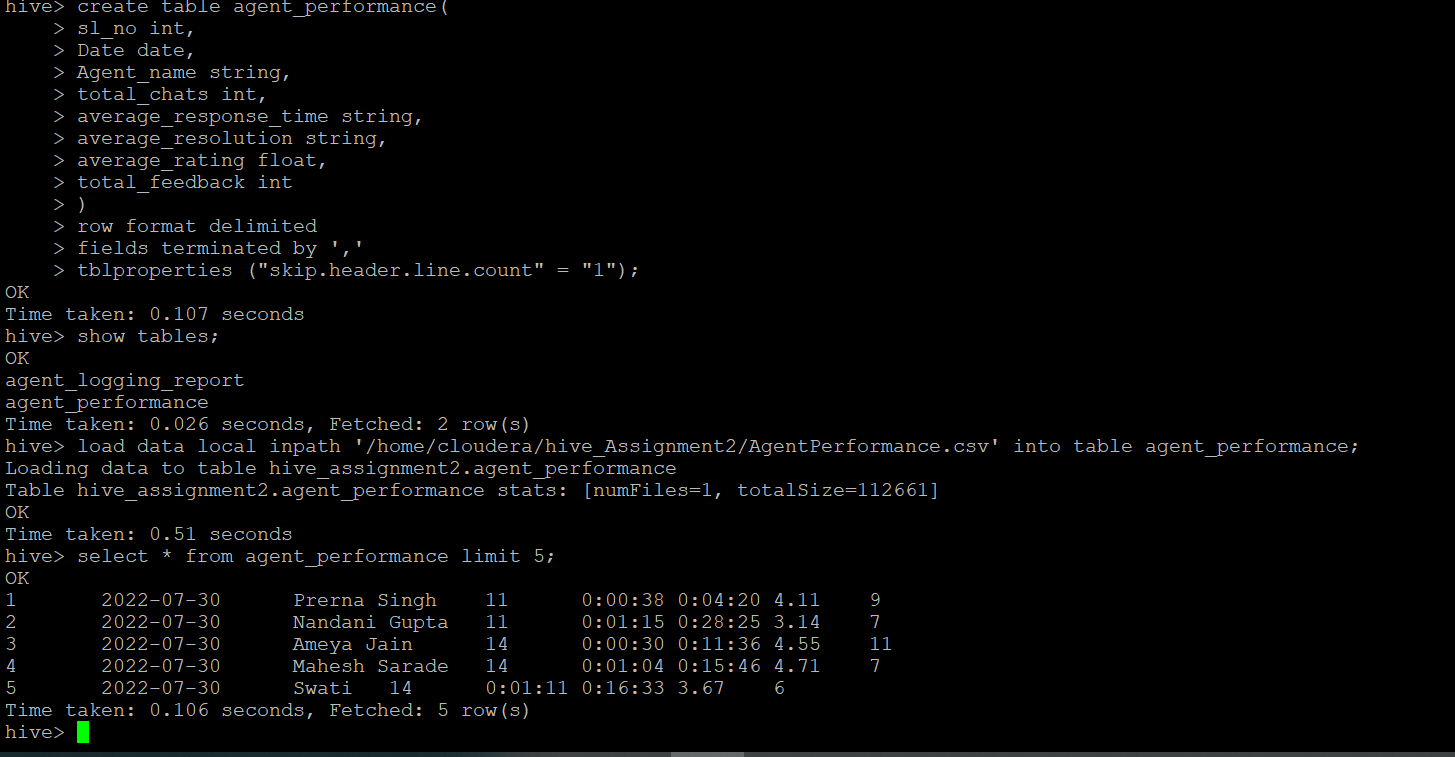
row format delimited

fields terminated by ','

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

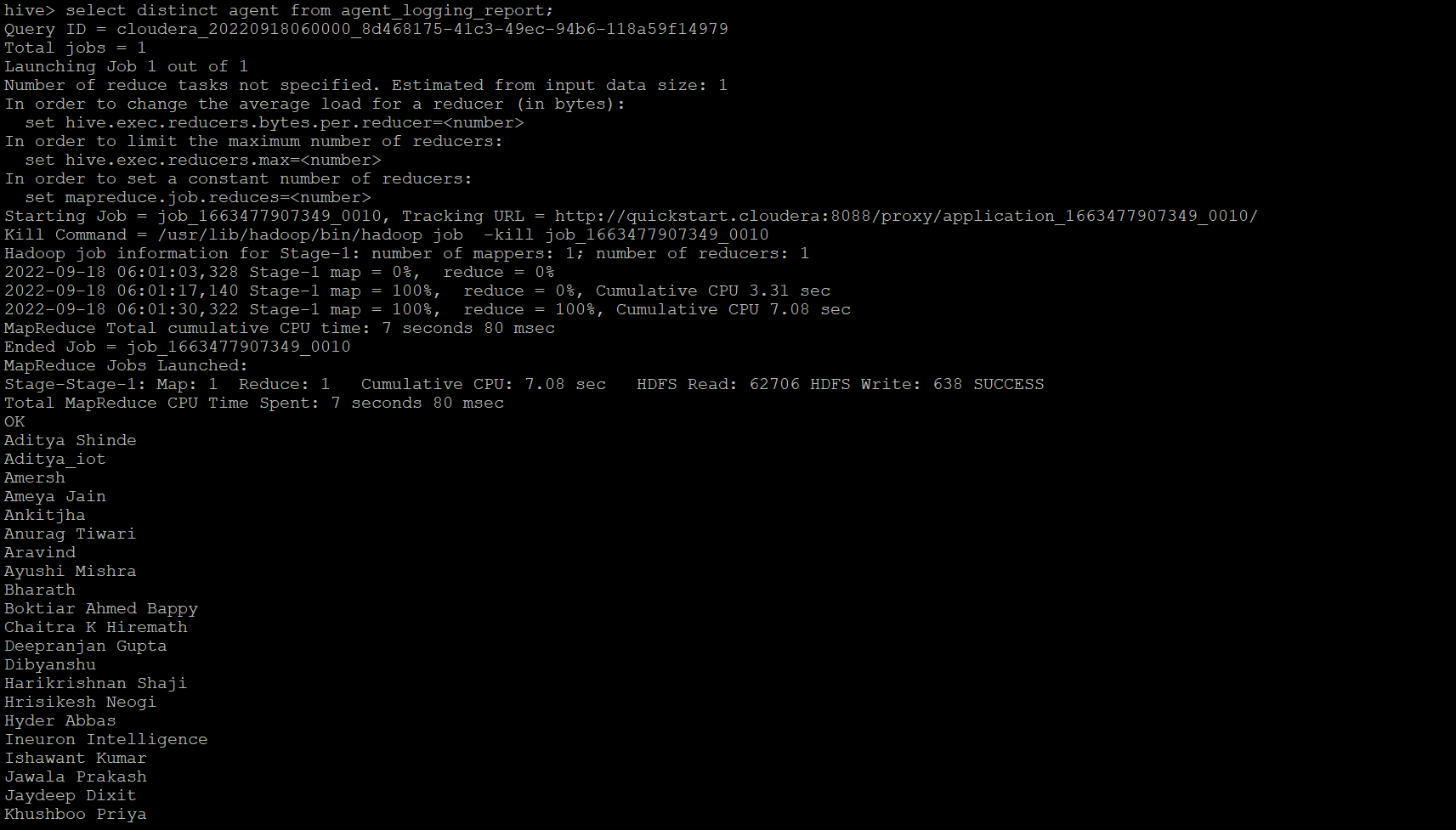
);





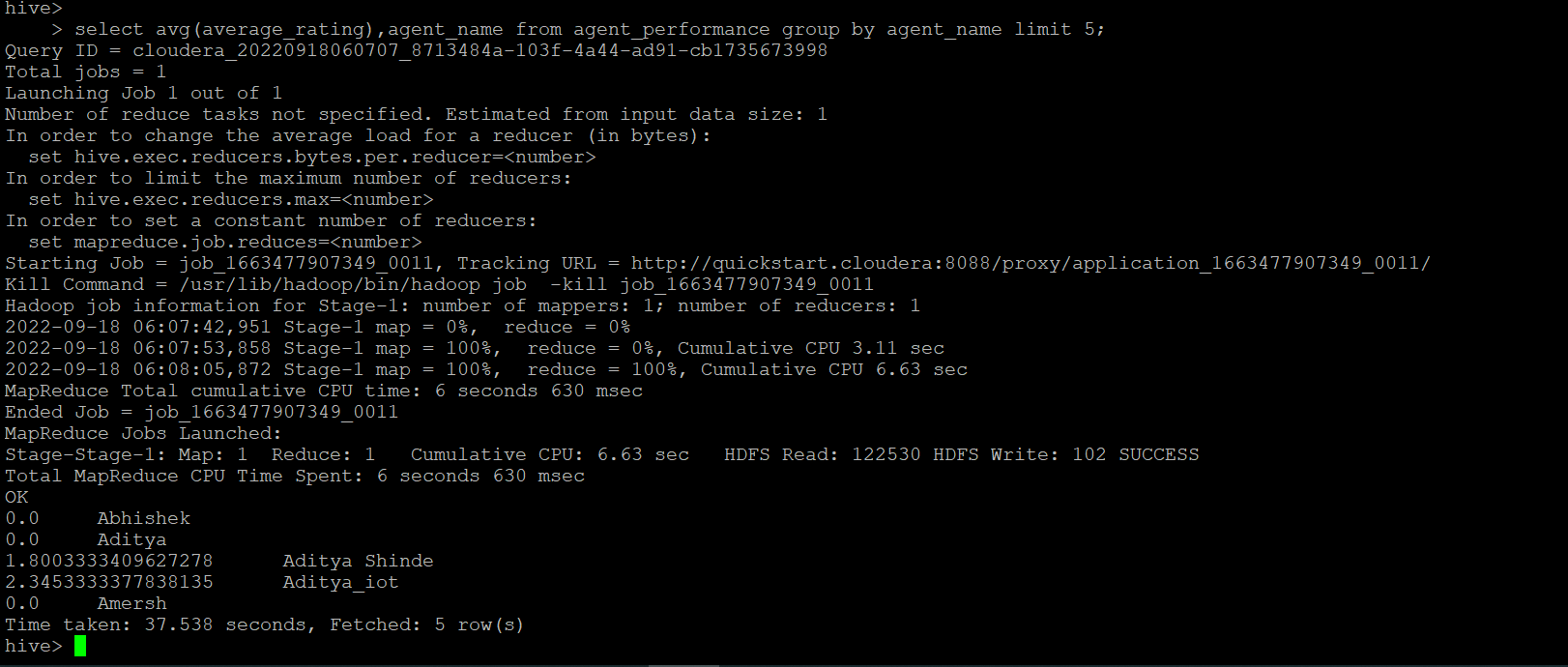
**3. List of all agents' names.**

select distinct agent from agent\_logging\_report ;



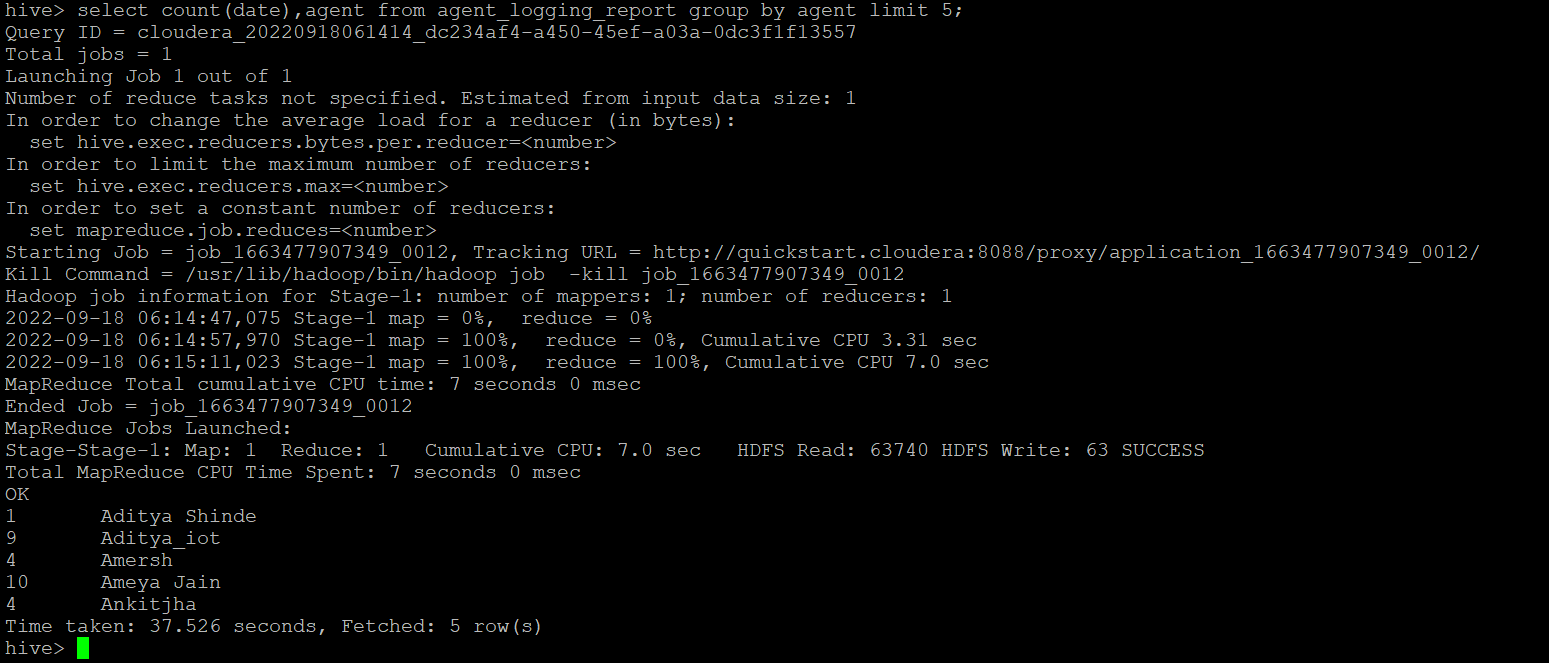
**4. Find out agent average rating.**

select avg(average\_rating),agent\_name from agent\_performance group by agent\_name limit 5;



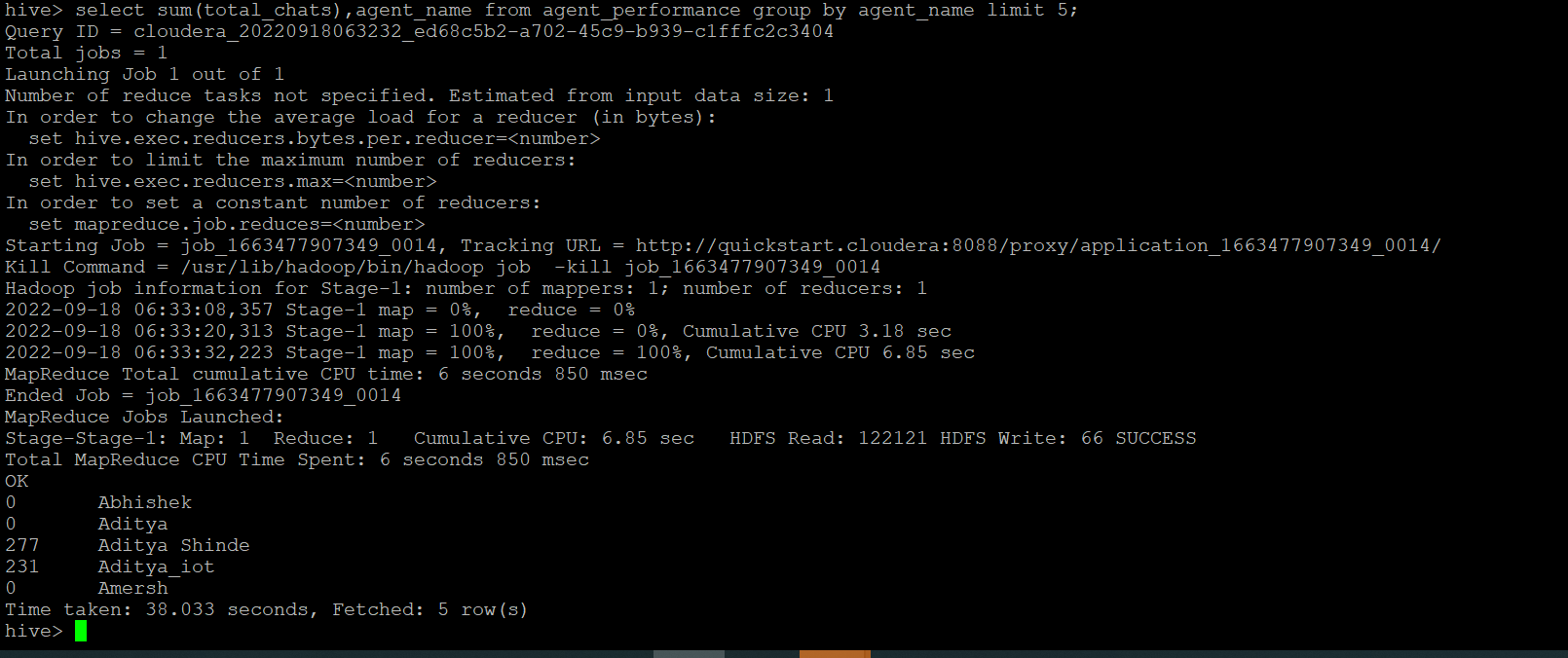
**5. Total working days for each agents**

select count(date),agent from agent\_logging\_report group by agent limit 5;



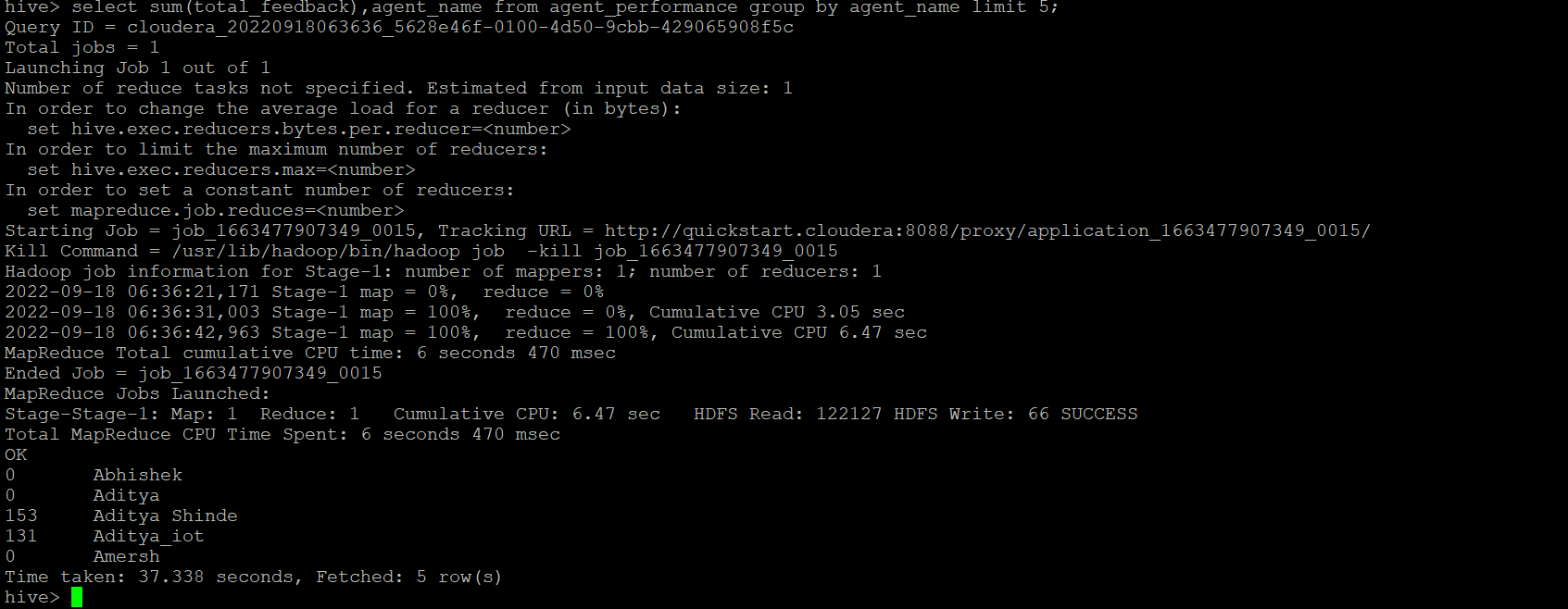
**6. Total query that each agent have taken**

select sum(total\_chats),agent\_name from agent\_performance group by agent\_name limit 5;



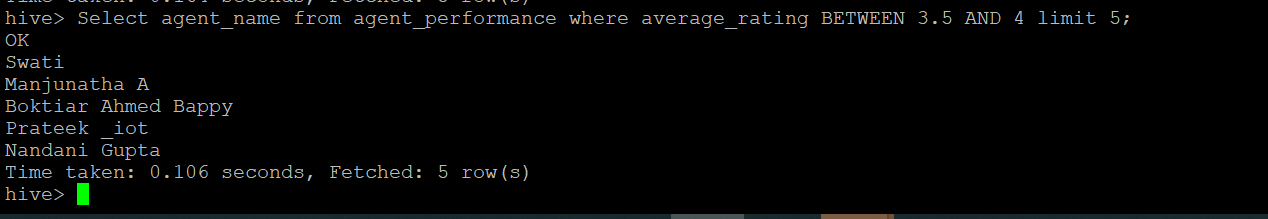
**7. Total Feedback that each agent have received**

select sum(total\_feedback),agent\_name from agent\_performance group by agent\_name limit 5;



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

Select agent\_name from agent\_performance where average\_rating BETWEEN 3.5 AND 4 limit 5;

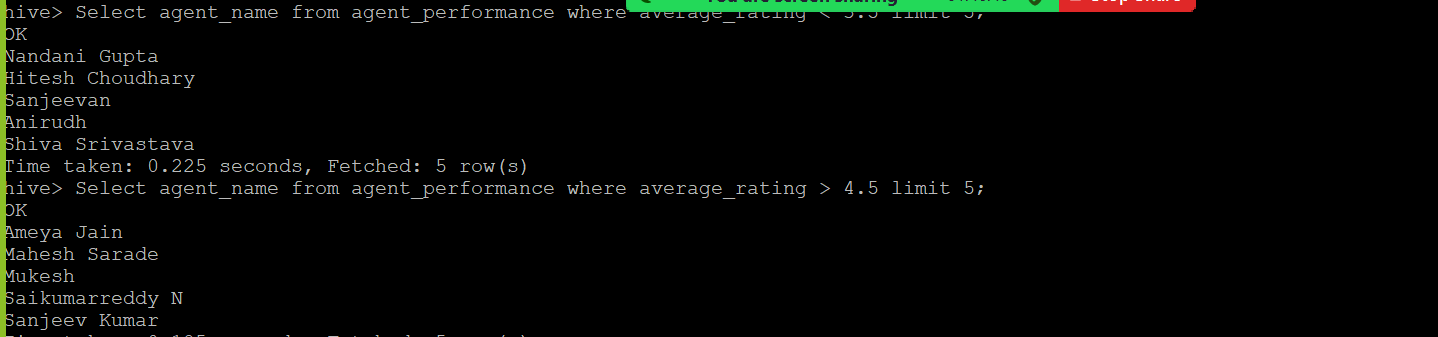


**9. Agent name who have rating less than 3.5**

Select agent\_name from agent\_performance where average\_rating < 3.5 limit 5;

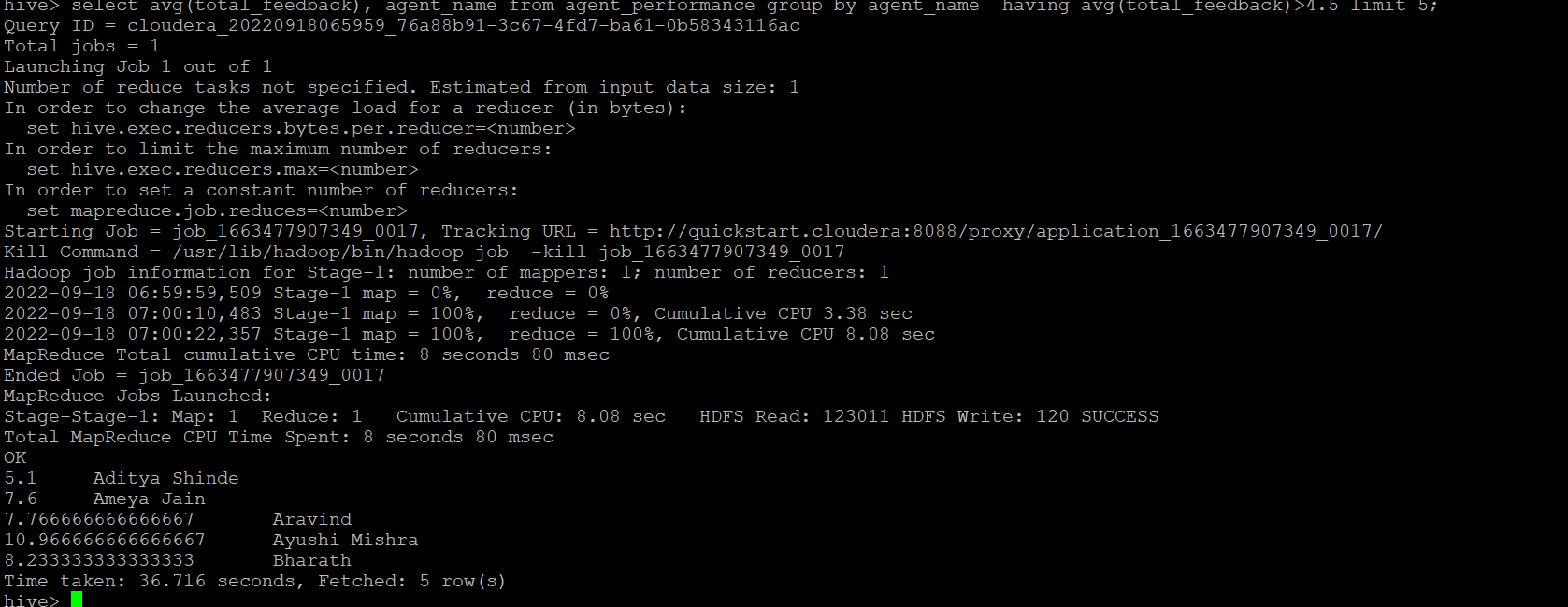
**10. Agent name who have rating more than 4.5**

Select agent\_name from agent\_performance where average\_rating > 4.5 limit 5;



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

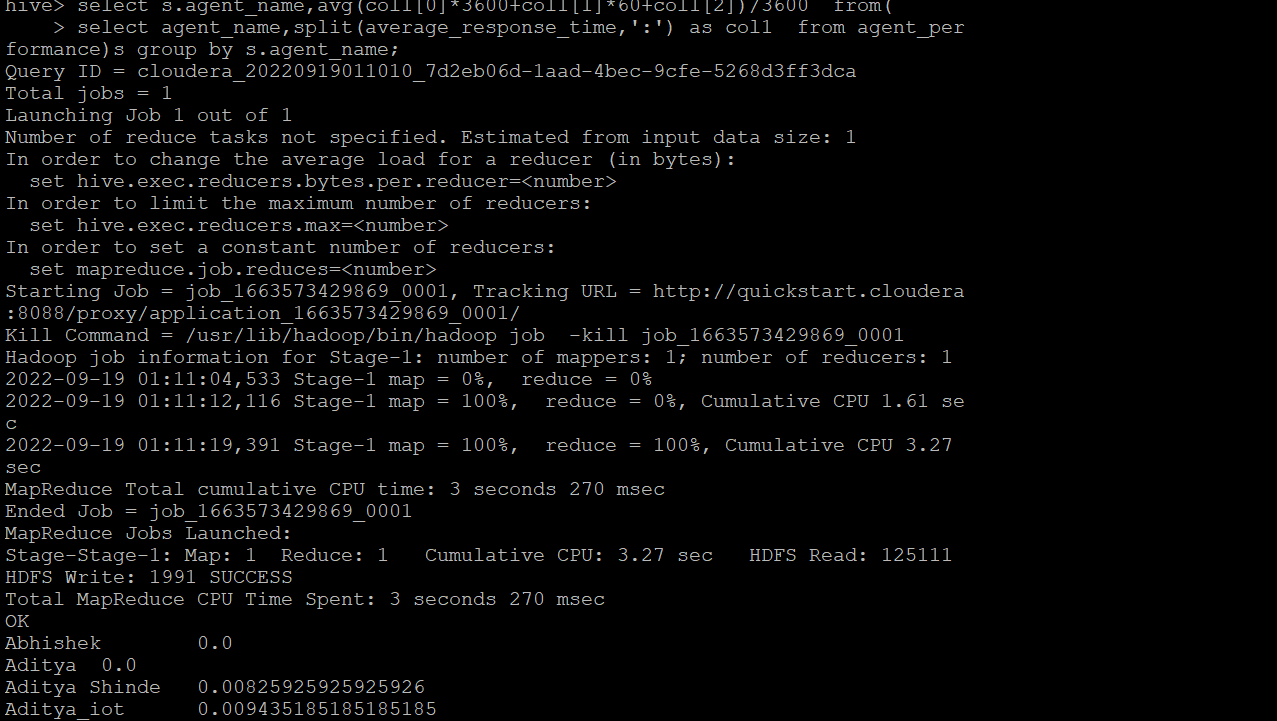
select avg(total\_feedback), agent\_name from agent\_performance group by agent\_name having avg(total\_feedback)>4.5;



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

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

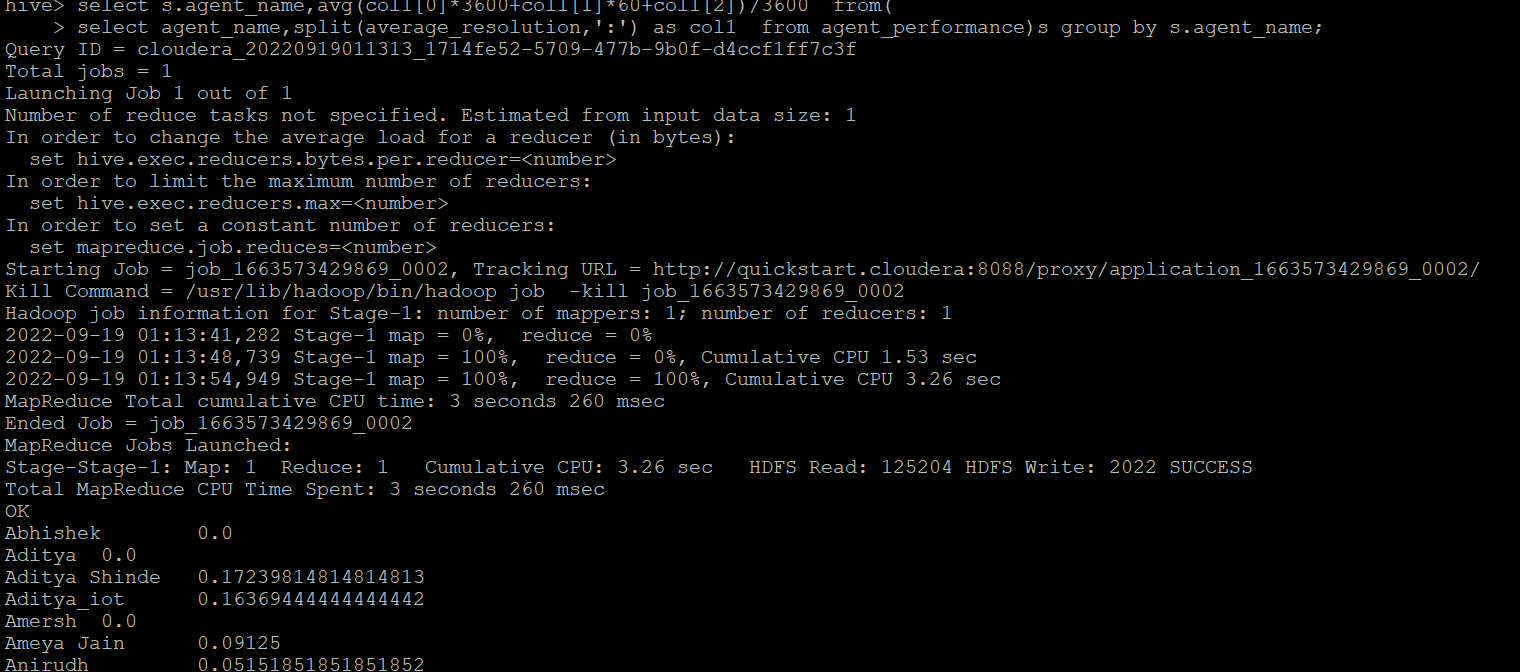
select agent\_name,split(average\_response\_time,':') as col1 from agent\_performance)s group by s.agent\_name;



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

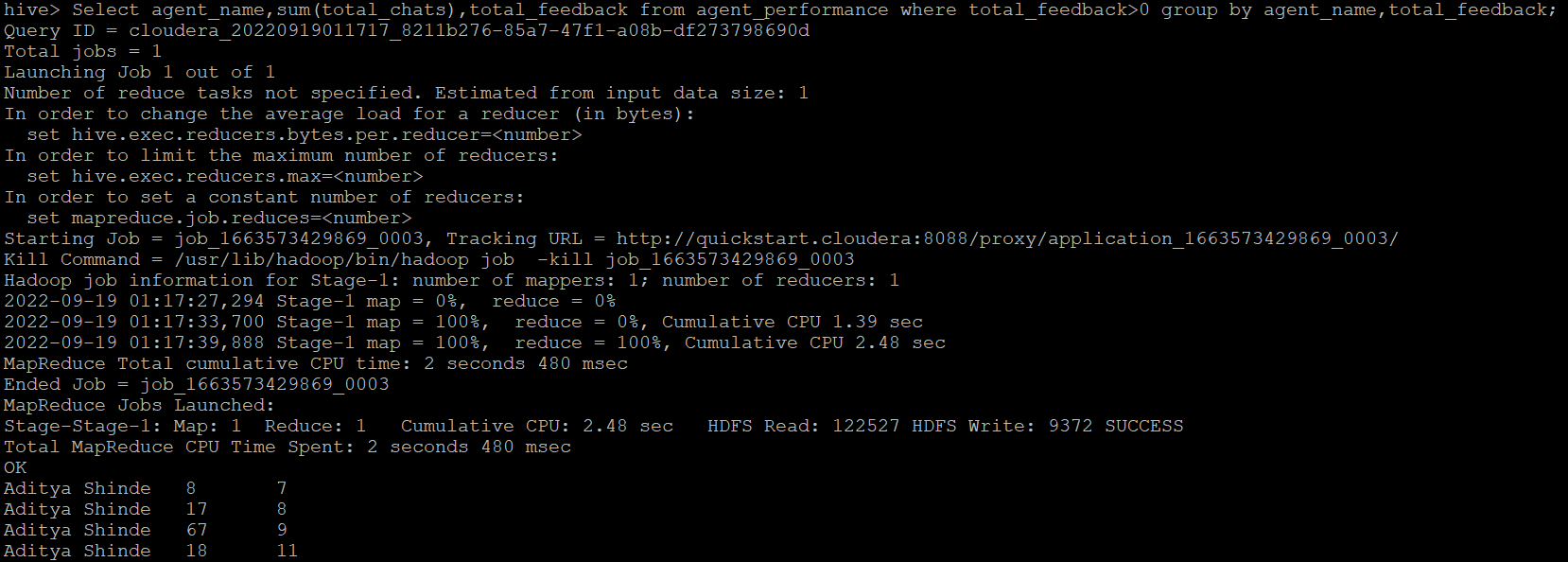
select s.agent\_name,avg(col1[0]\*3600+col1[1]\*60+col1[2])/3600 from(

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



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

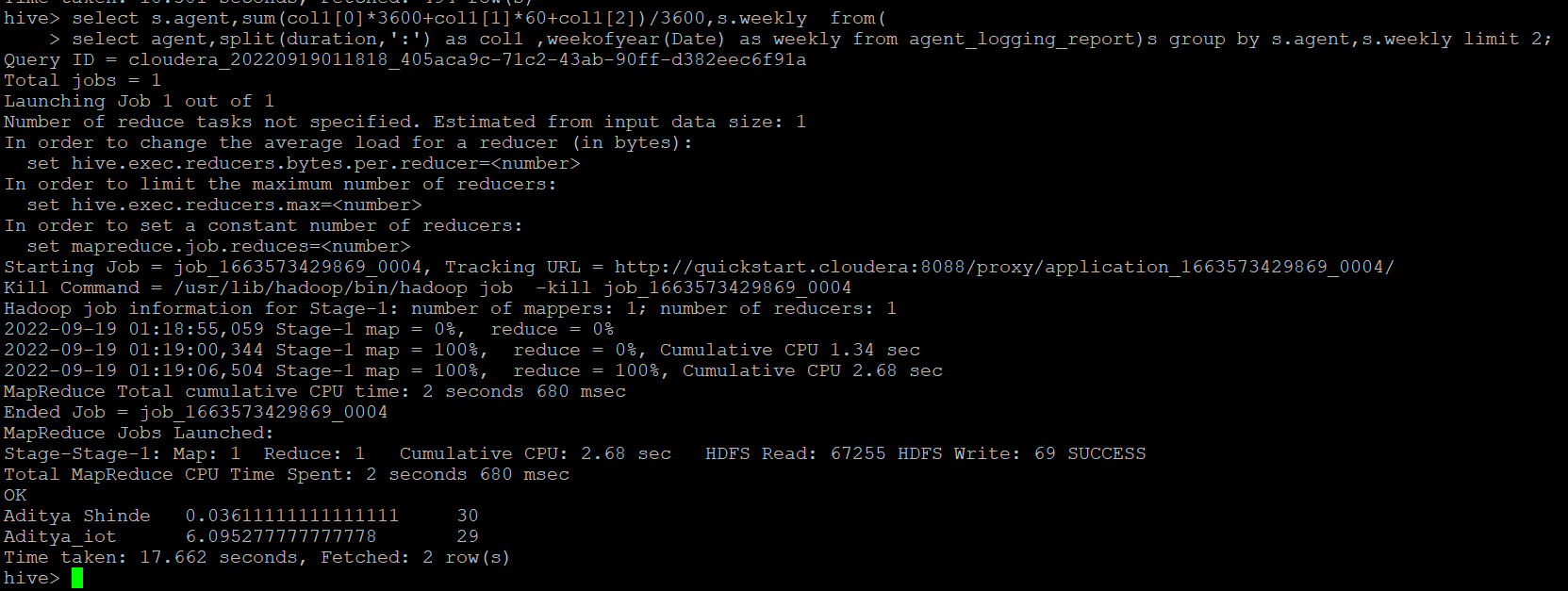
Select agent\_name,sum(total\_chats),total\_feedback from agent\_performance where total\_feedback>0 group by agent\_name,total\_feedback;



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

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

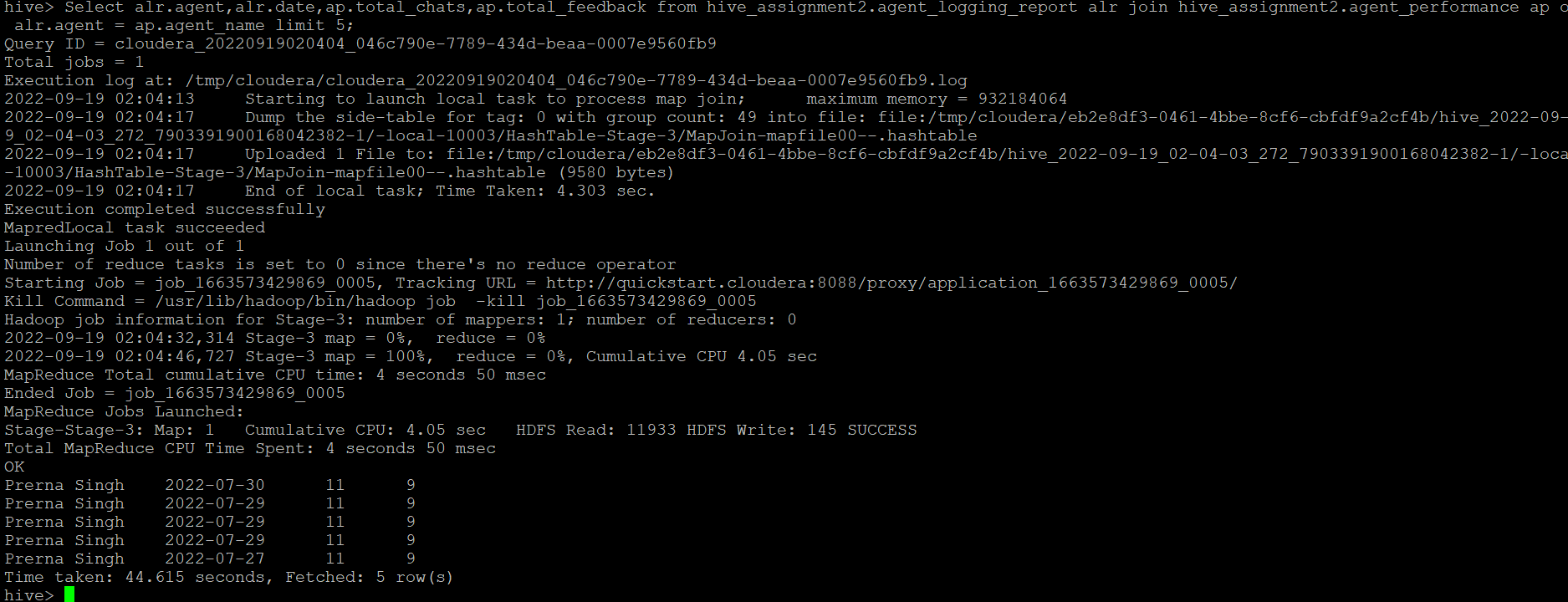
select agent,split(duration,':') as col1 ,weekofyear(Date) as weekly from agent\_logging\_report)s group by s.agent,s.weekly limit 2;



**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.**

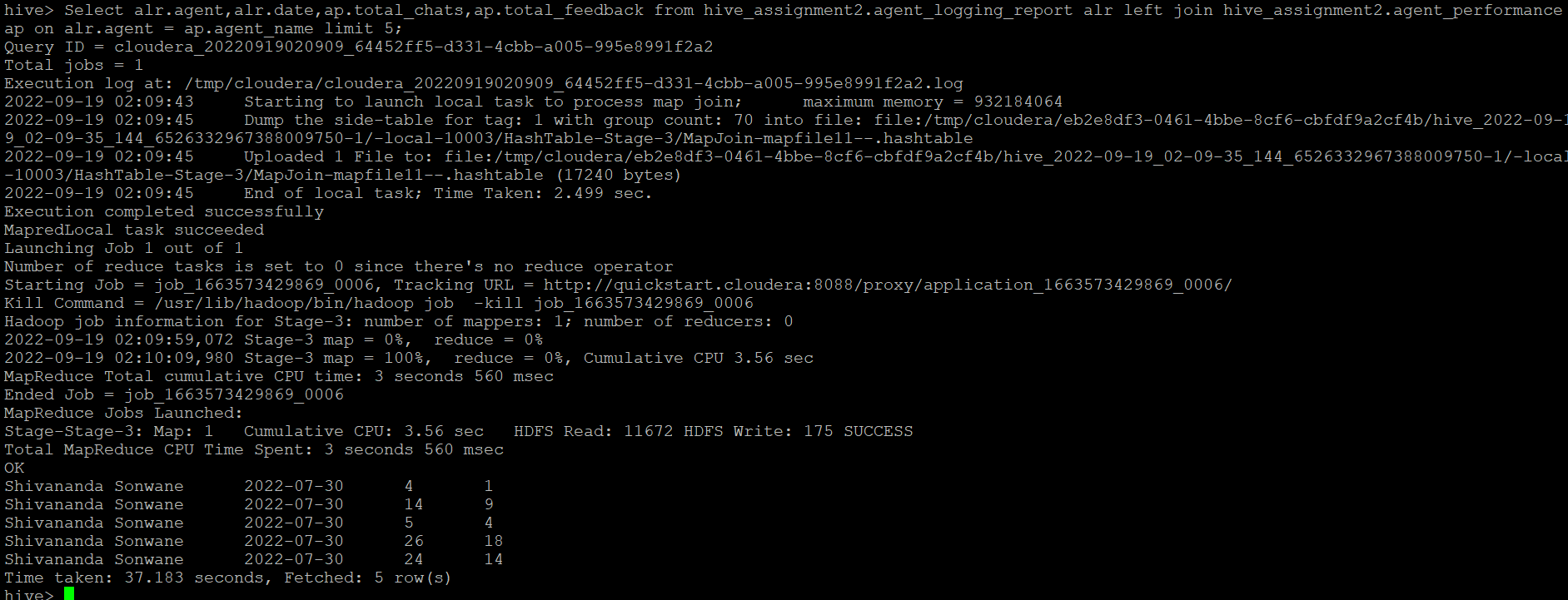
**INNER JOIN:**

Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5;



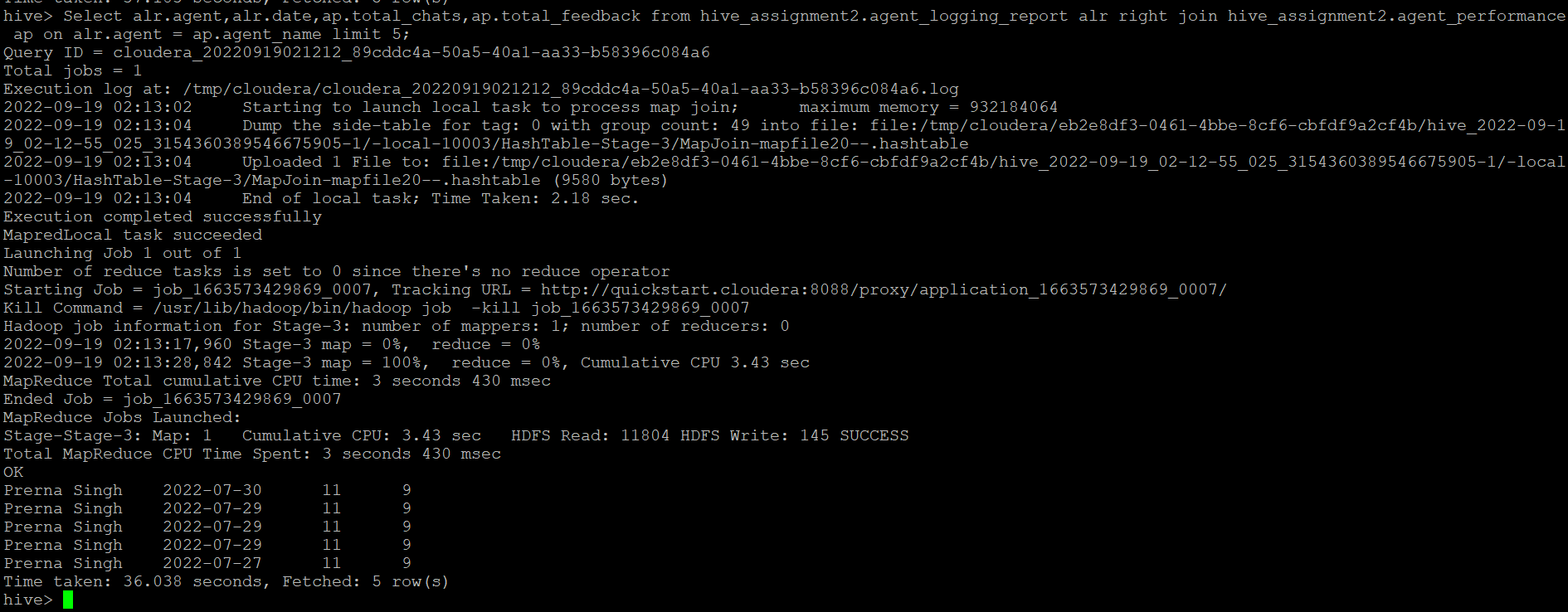
**LEFT JOIN:**

Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr left join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5;



**RIGHT JOIN:**

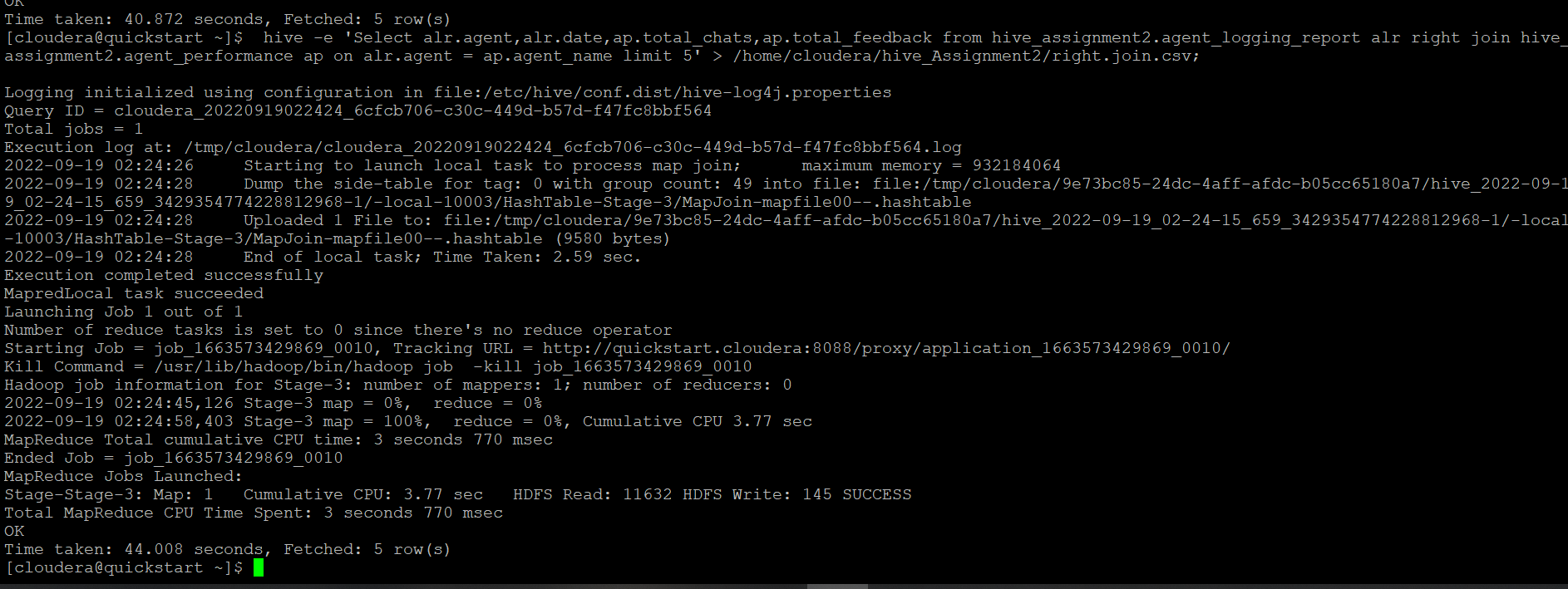
Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr right join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5;



$ hive -e 'Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5' > /home/cloudera/hive\_Assignment2/inner.join.csv;

$ hive -e 'Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr left join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5' > /home/cloudera/hive\_Assignment2/left.join.csv;

$ hive -e 'Select alr.agent,alr.date,ap.total\_chats,ap.total\_feedback from hive\_assignment2.agent\_logging\_report alr right join hive\_assignment2.agent\_performance ap on alr.agent = ap.agent\_name limit 5' > /home/cloudera/hive\_Assignment2/right.join.csv;



**17. 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

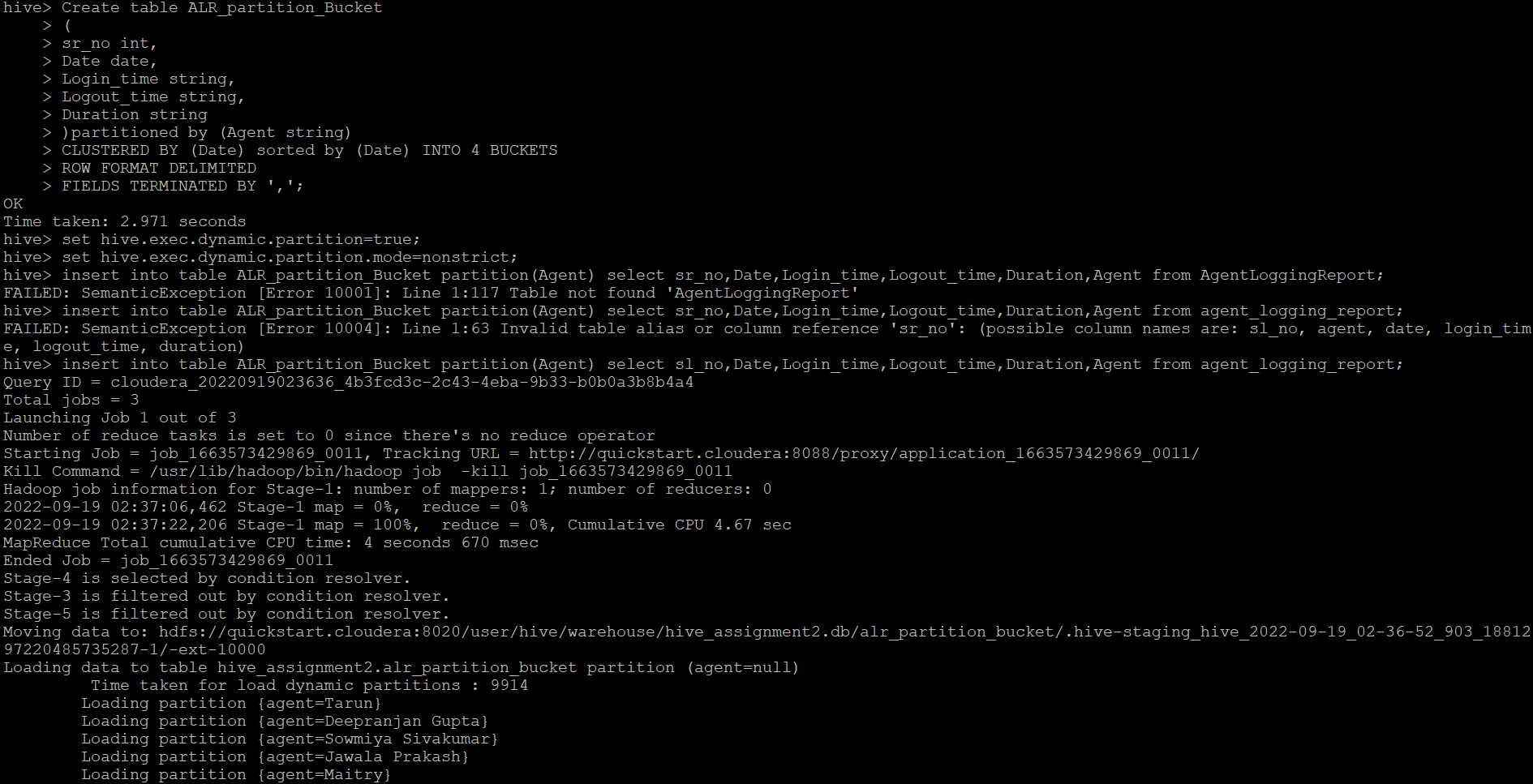
ROW FORMAT DELIMITED

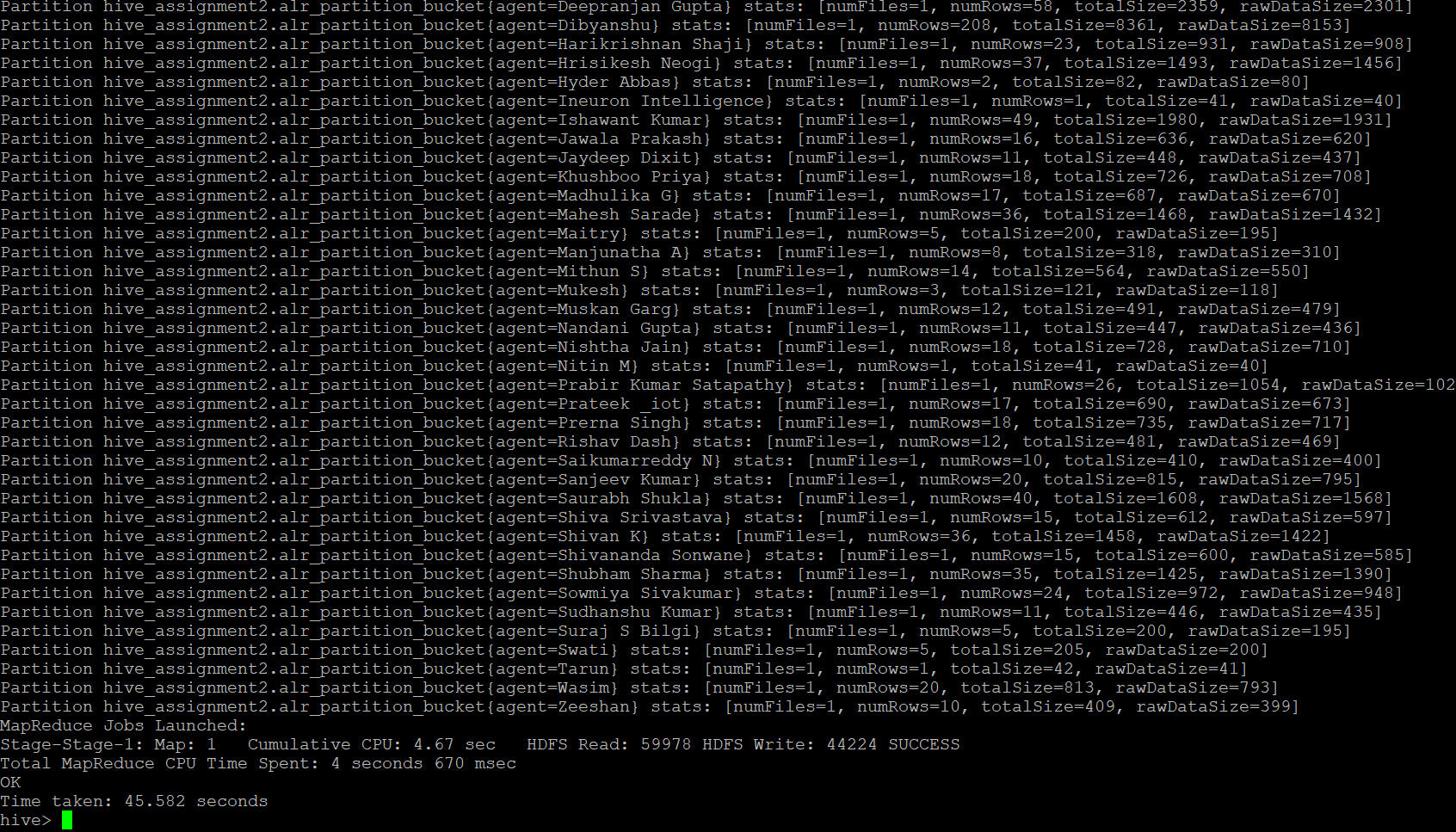
FIELDS TERMINATED BY ',';

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;

