**MINI PROJECT 1**

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

Create table AgentLogingReport

(

sr\_no int,

Agent string,

Date string,

Login string,

Logout string,

Duration string

)

row format delimited

fields terminated by ','

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

Create table AgentPerformance

(

sr\_no int,

Date string,

Agent\_Name string,

Total\_charts int,

Avg\_Response\_Time string,

Avg\_Resolution\_Time string,

Avg\_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.**

load data inpath '/tmp/a\_1/AgentLogingReport.csv' into table AgentLogingReport ;

load data inpath '/tmp/a\_1/AgentPerformance.csv' into table AgentPerformance;

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

select distinct(agent\_name) from AgentPerformance ;

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

select agent\_name , avg(avg\_rating) from AgentPerformance group by agent\_name;

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

select count(date) , agent from AgentLogingReport

group by agent;

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

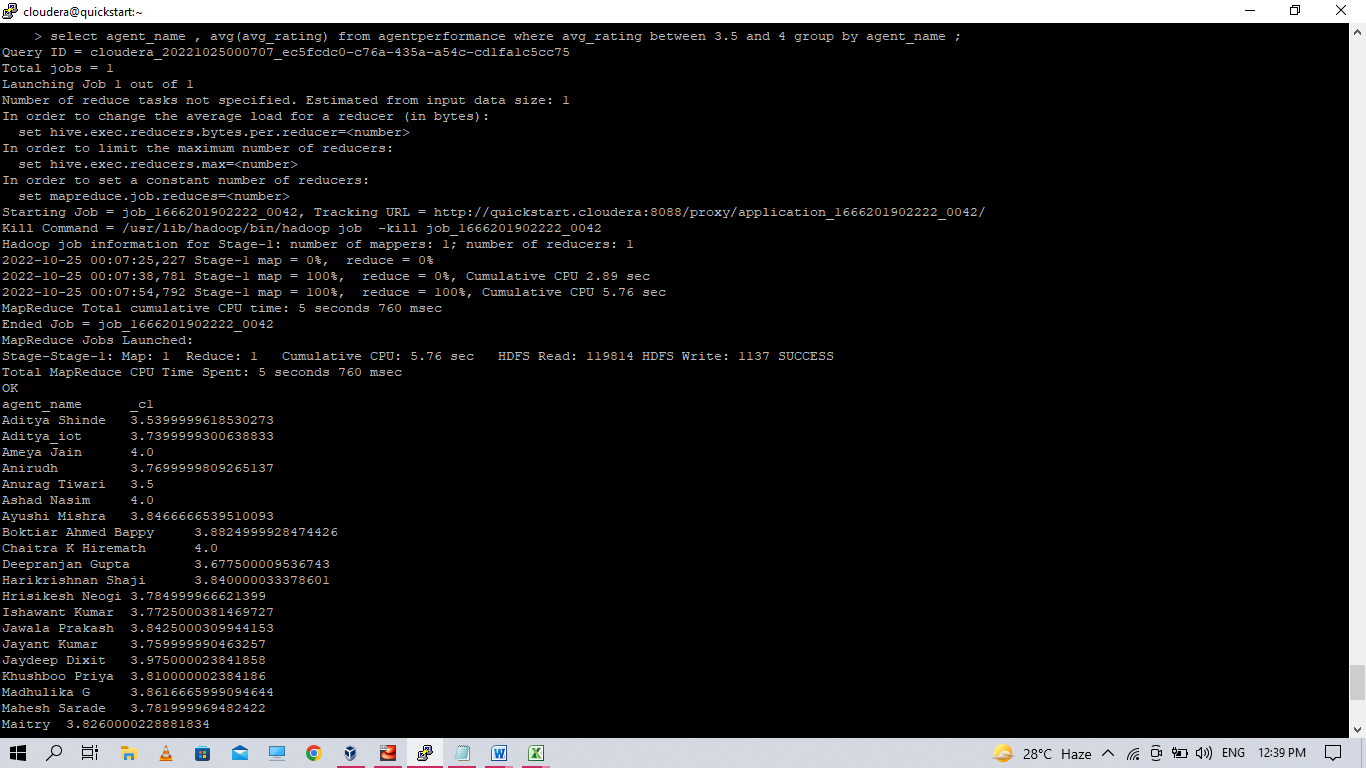
select agent\_name , sum(total\_chats) from AgentPerformance group by agent\_name;

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

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

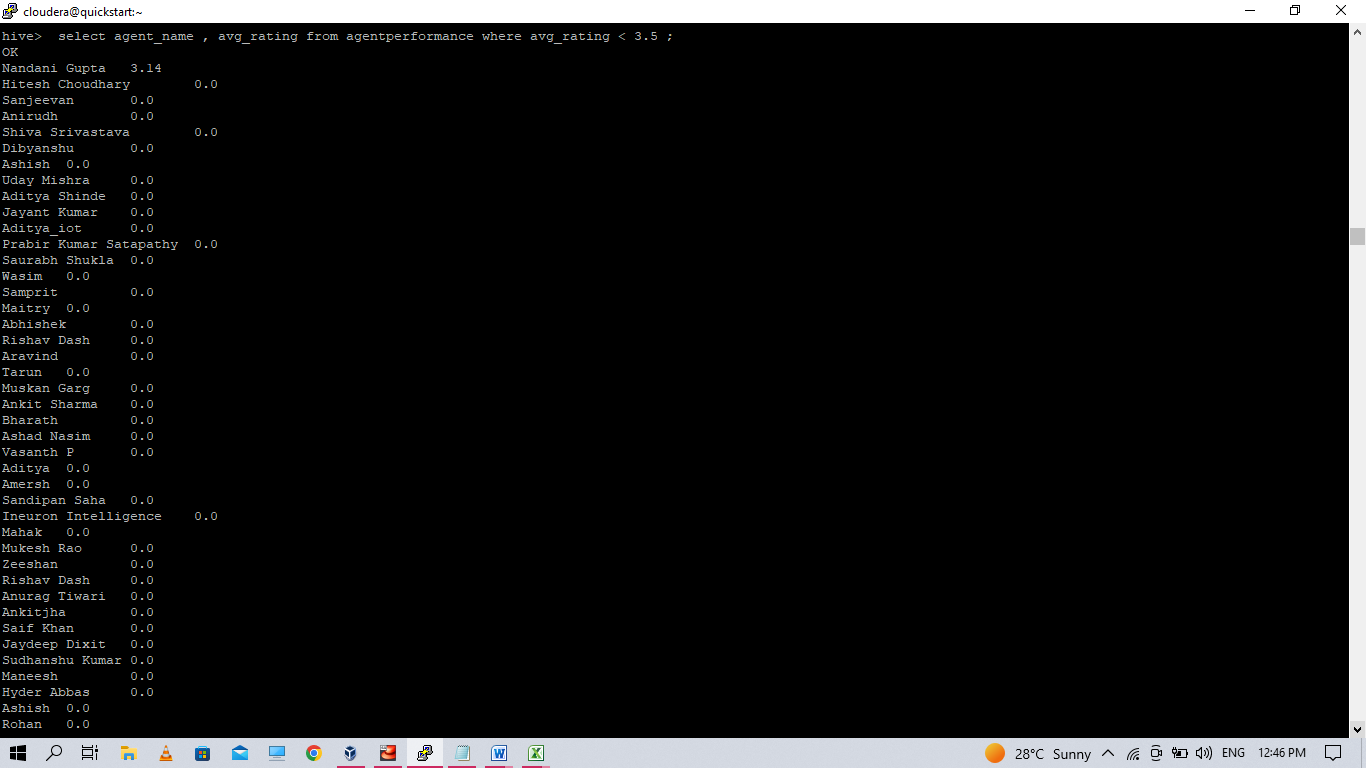
1. **Agent name who have average rating between 3.5 to 4**

select agent\_name , avg(avg\_rating) from agentperformance where avg\_rating between 3.5 and 4 group by agent\_name ;



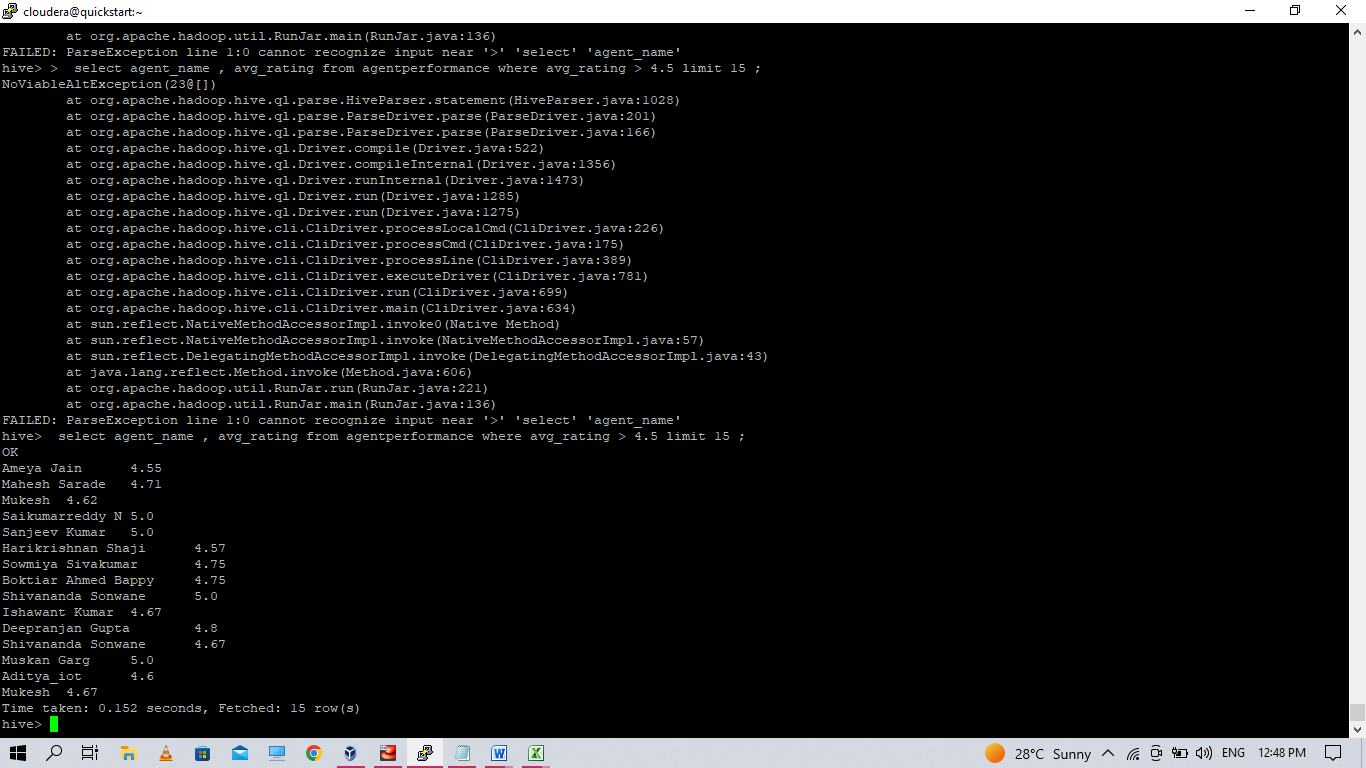
1. **Agent name who have rating less than 3.5**

select agent\_name , avg\_rating from agentperformance where avg\_rating < 3.5 ;



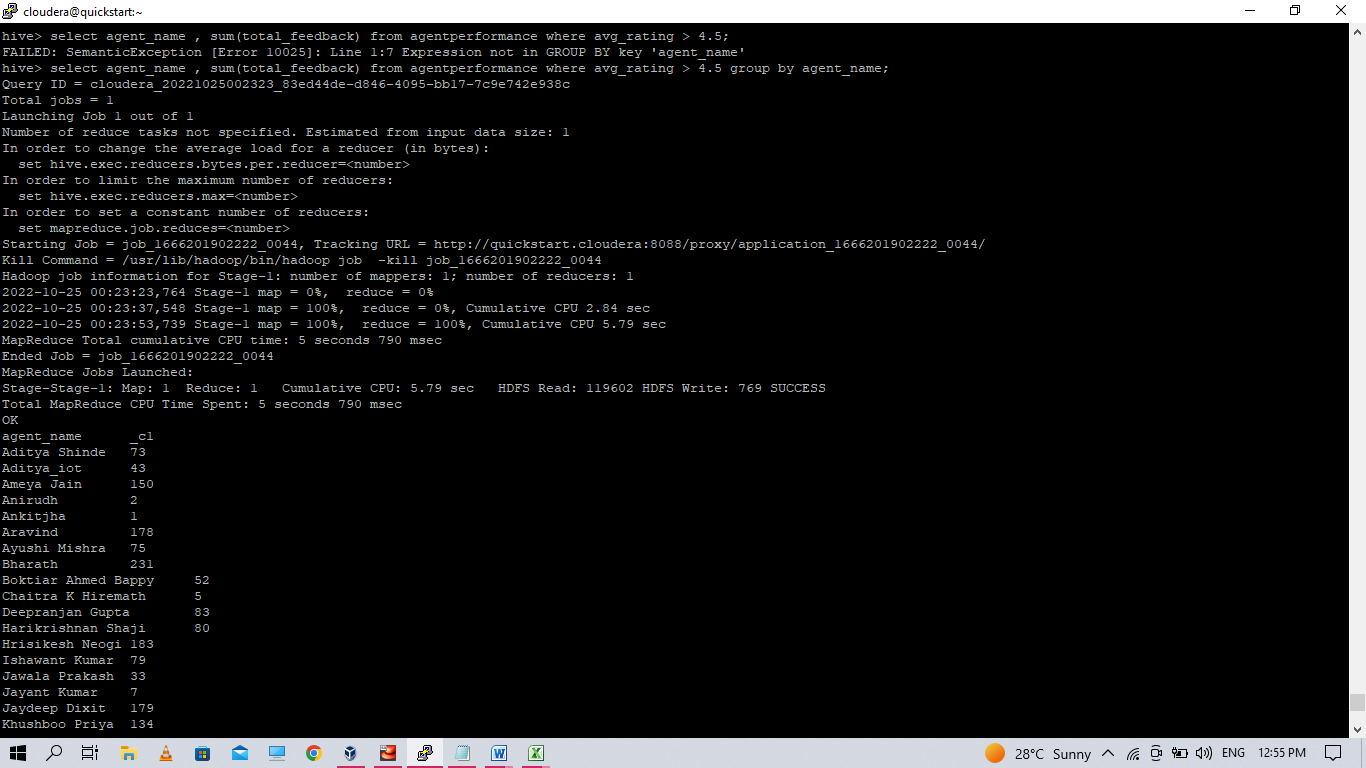
1. **Agent name who have rating more than 4.5**

select agent\_name , avg\_rating from agentperformance where avg\_rating > 4.5 limit 15 ;



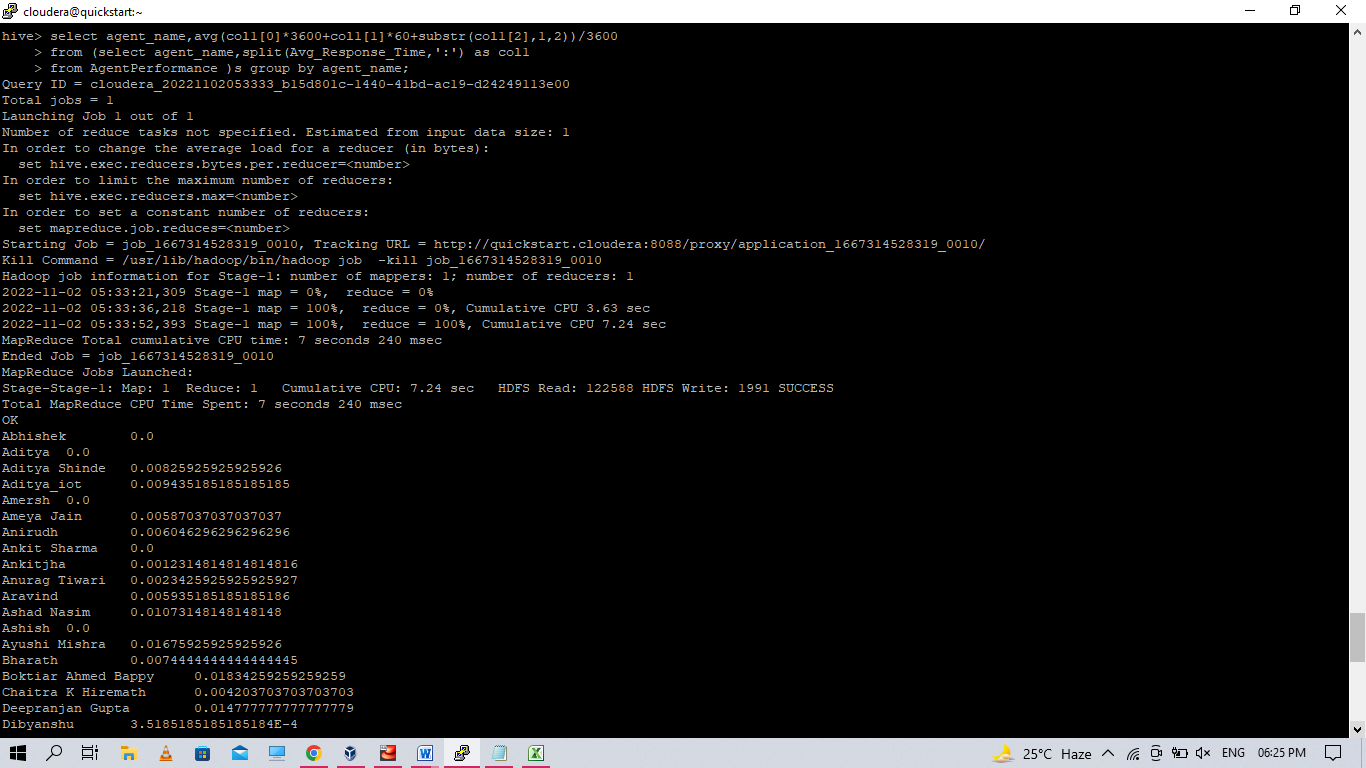
1. **How many feedback agents have received more than 4.5 average**

select agent\_name , sum(total\_feedback) from agentperformance where avg\_rating > 4.5 group by agent\_name ;

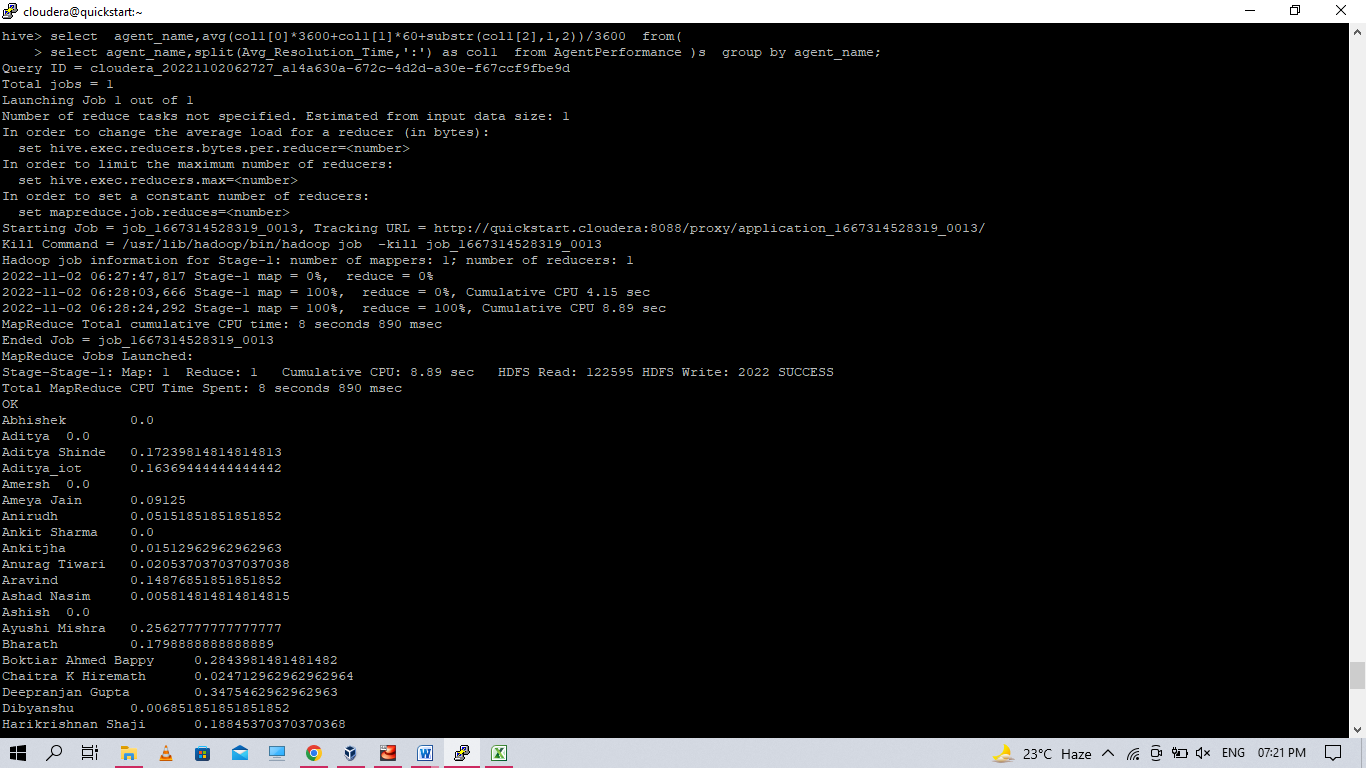


1. **average weekly response time for each agent**

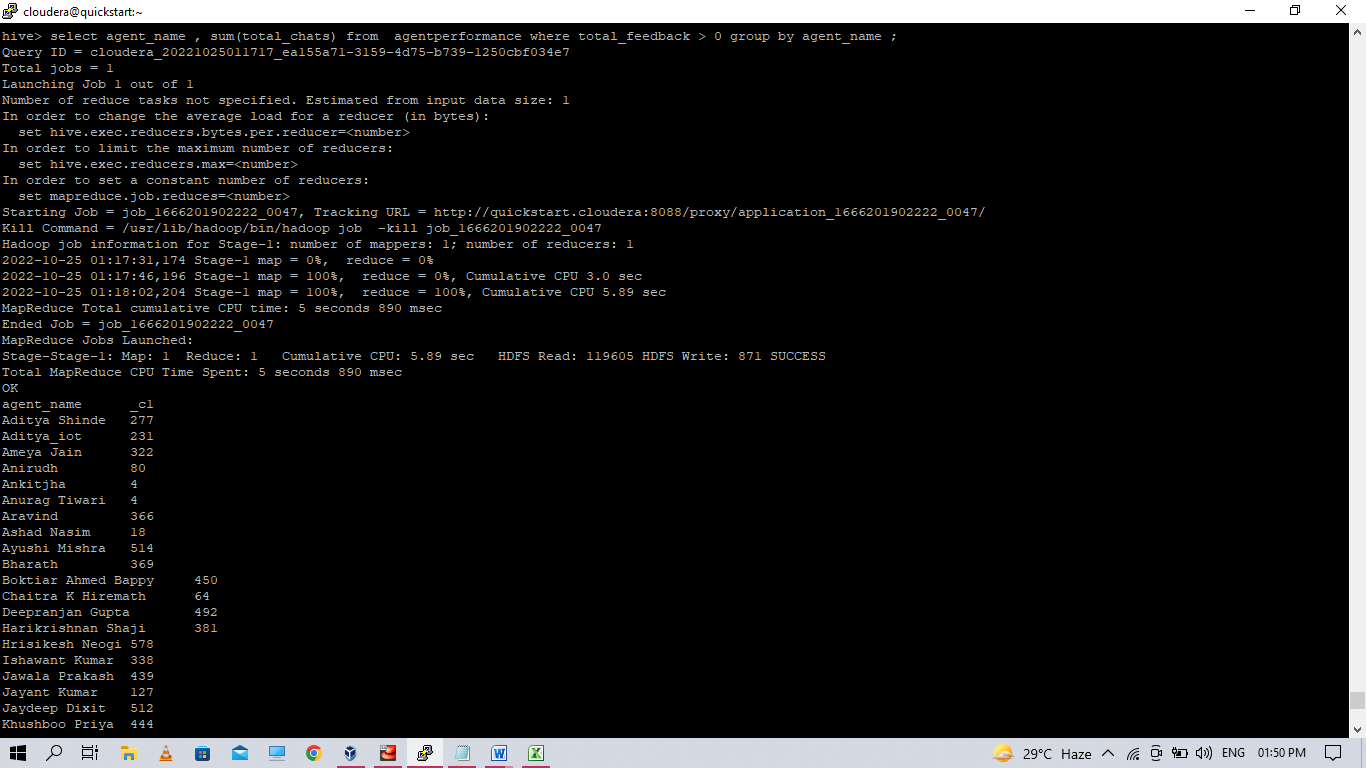
> select agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/360 from (select agent\_name,split(Avg\_Response\_Time,':') as col1 from AgentPerformance )s group by agent\_name;



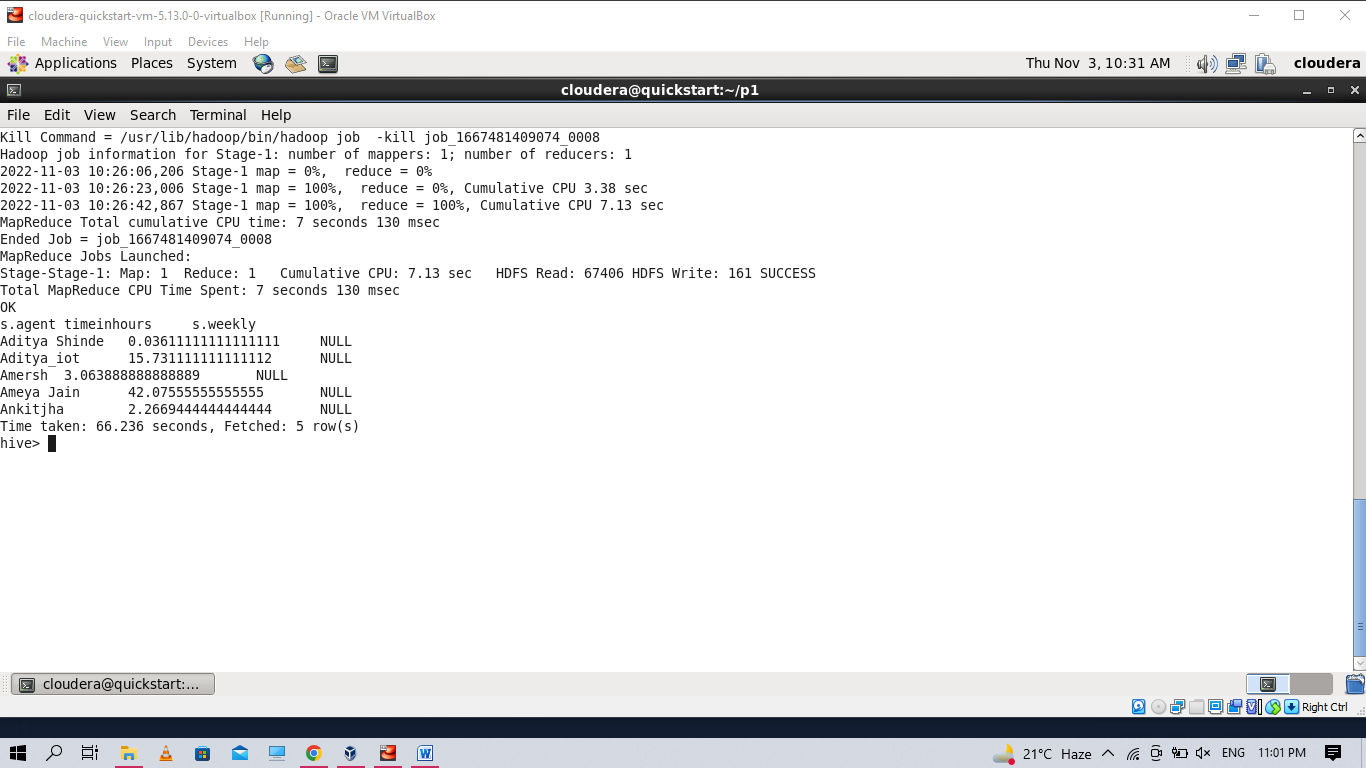
1. **average weekly resolution time for each agents**

select agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/3600 from(select agent\_name,split(Avg\_Resolution\_Time,':') as col1 from AgentPerformance )s group by agent\_name;

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

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

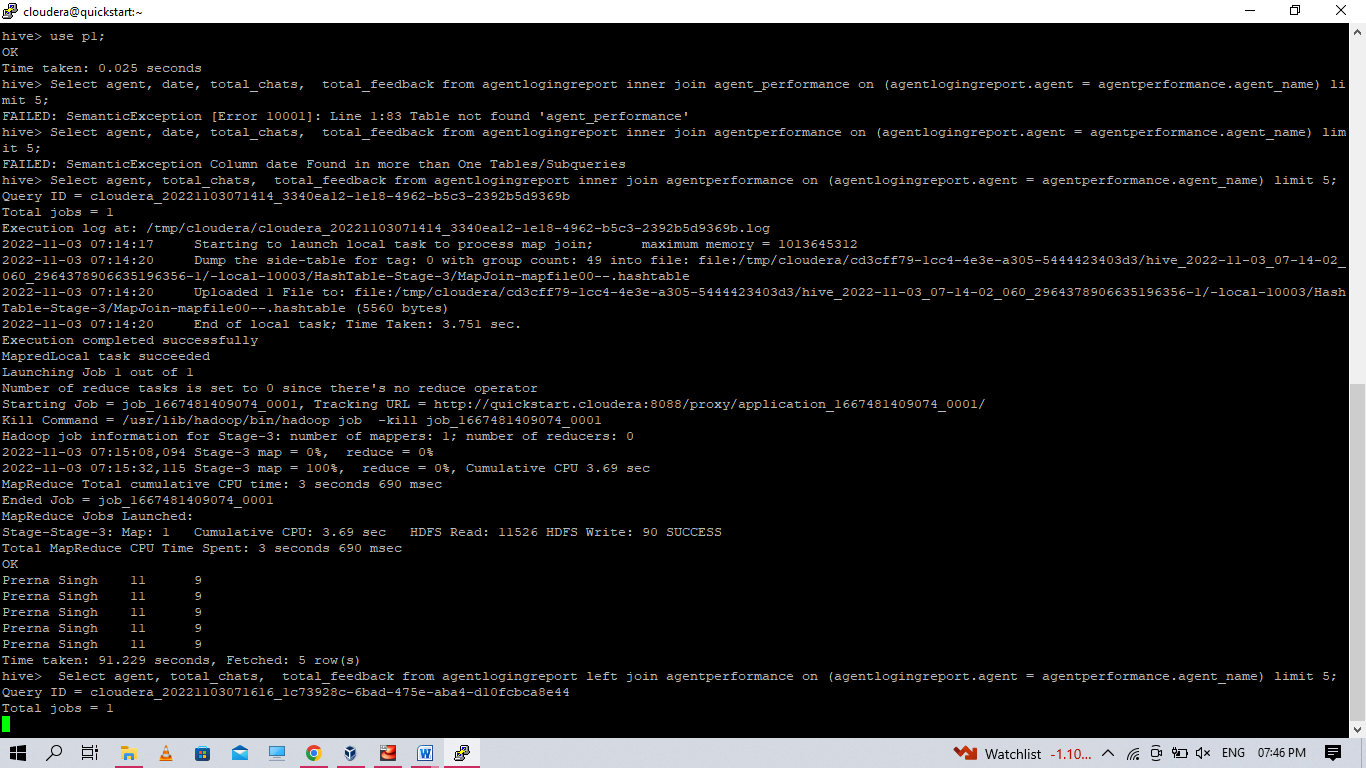
1. **Total contribution hour for each and every agents weekly basis**

select s.agent,sum(col1[0]\*3600+col1[1]\*60+col1[2])/3600 timeInHour,s.weekly from (select agent,split(duration,':') as col1 ,weekofyear(Date) as weekly from AgentLogingReport )s group by s.agent,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.**

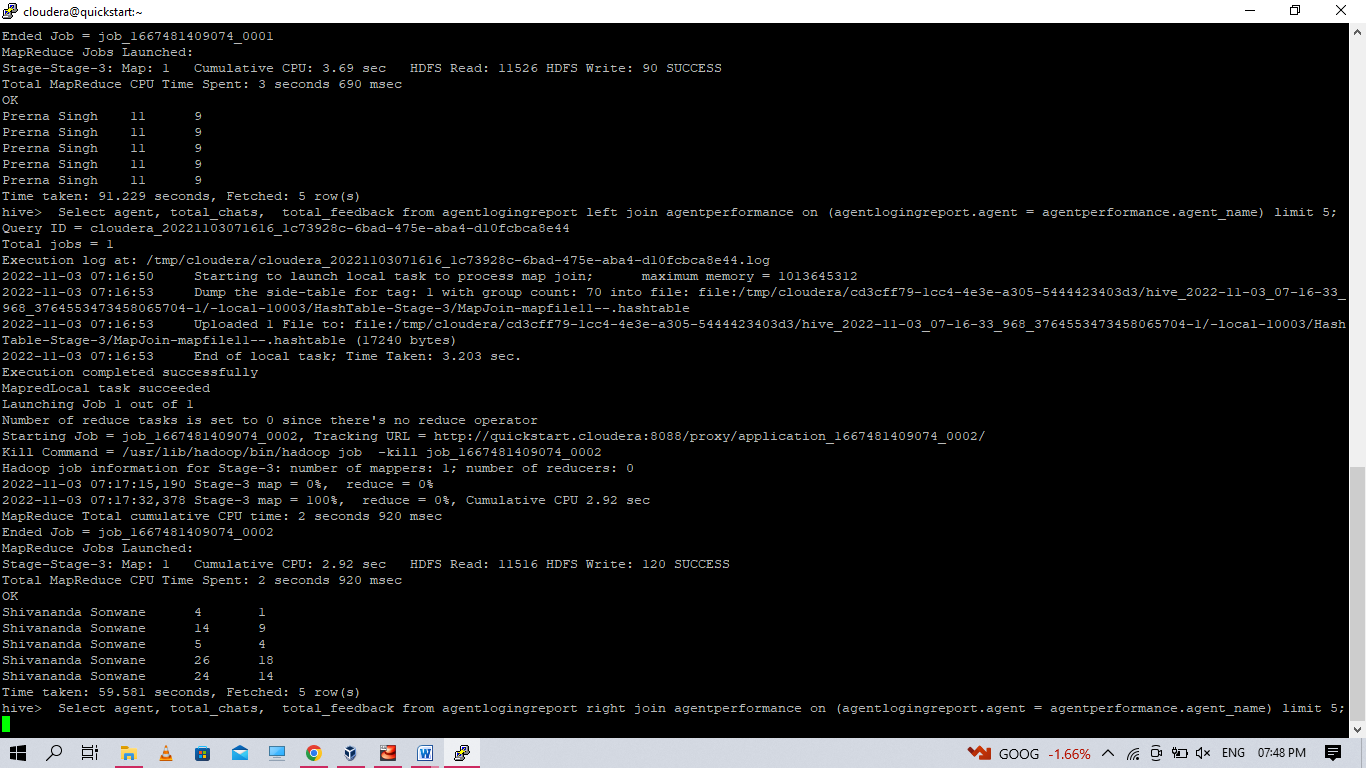
**INNER JOIN**

Select agent, total\_chats, total\_feedback from agentlogingreport inner join agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5;



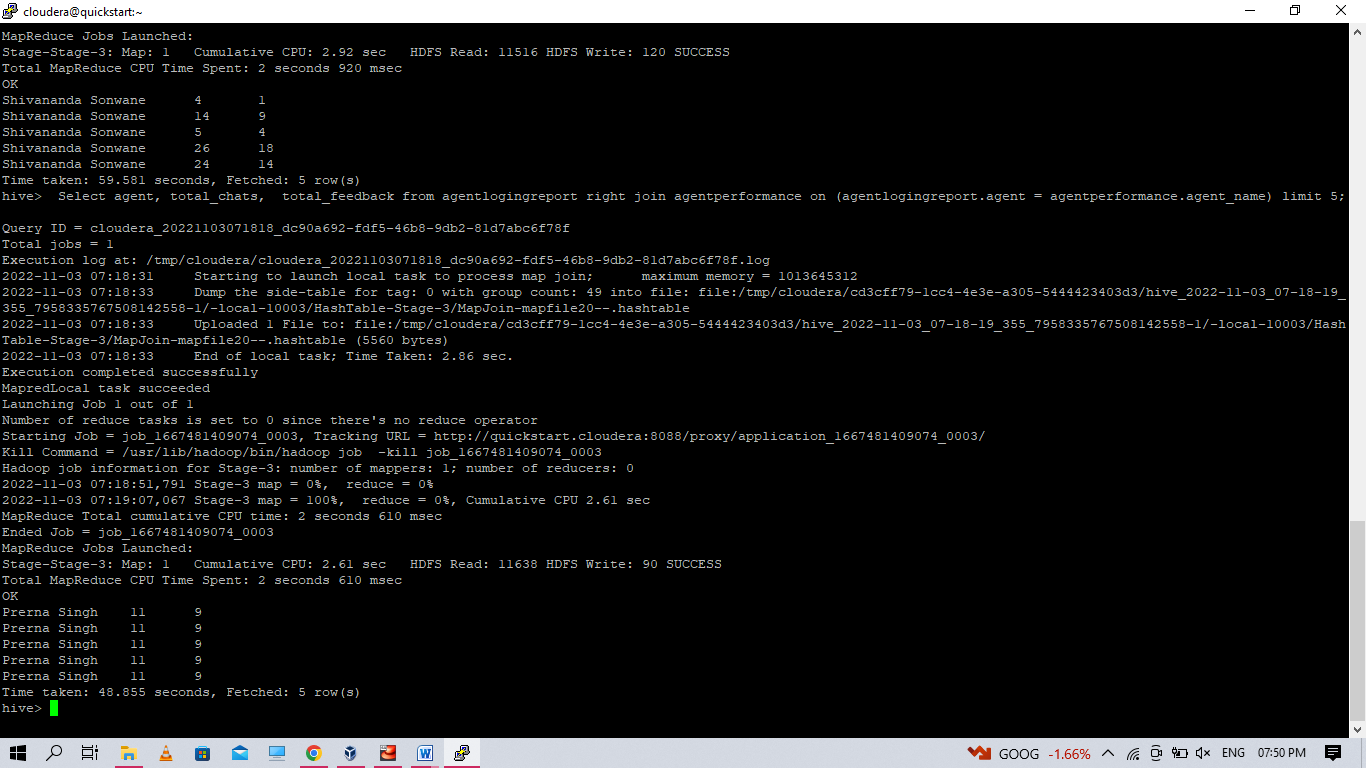
**LEFT JOIN**

Select agent, total\_chats, total\_feedback from agentlogingreport left join agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5



**RIGHT JOIN**

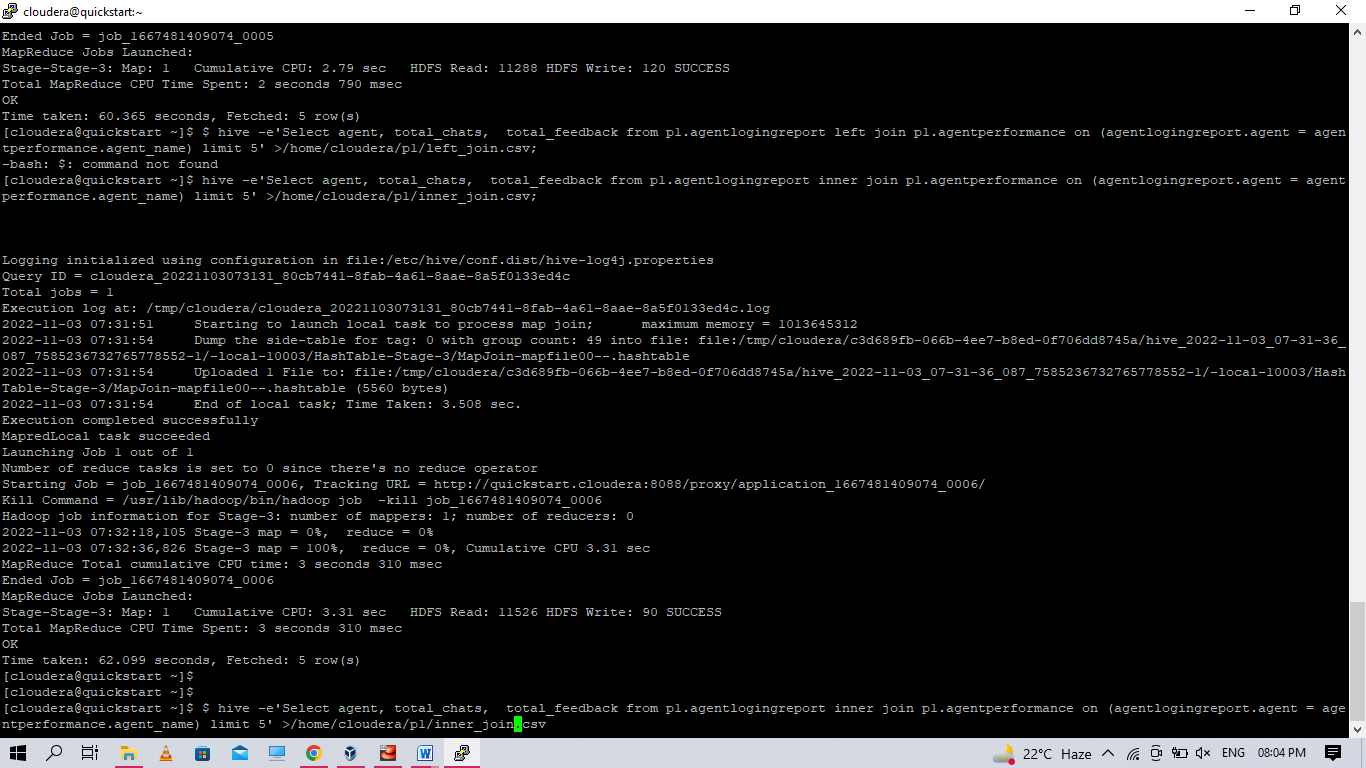
Select agent, total\_chats, total\_feedback from agentlogingreport right join agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5;



**EXPORT THE OUPUT DATA INTO LOCAL SYSTEM**

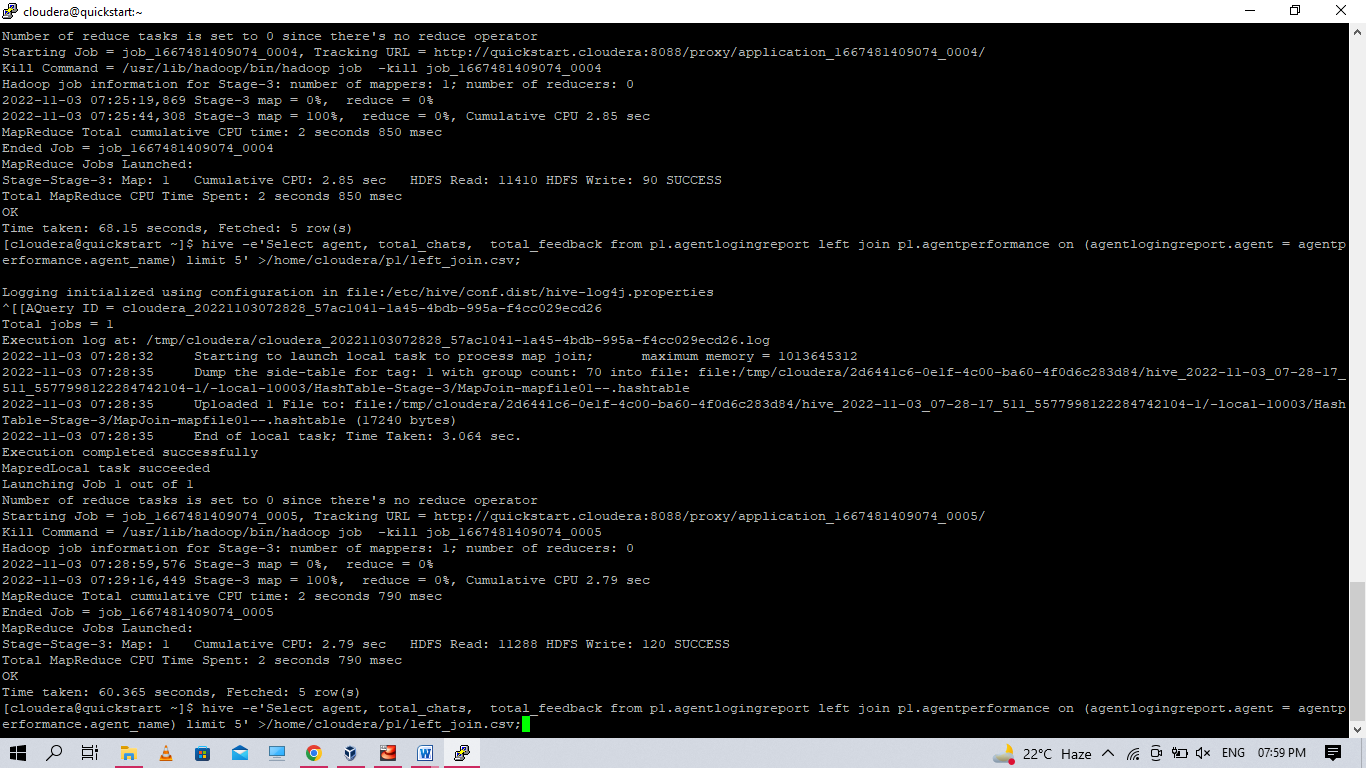
**INNER JOIN**

$ hive -e'Select agent, total\_chats, total\_feedback from p1.agentlogingreport inner join p1.agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5' >/home/cloudera/p1/inner\_join.csv



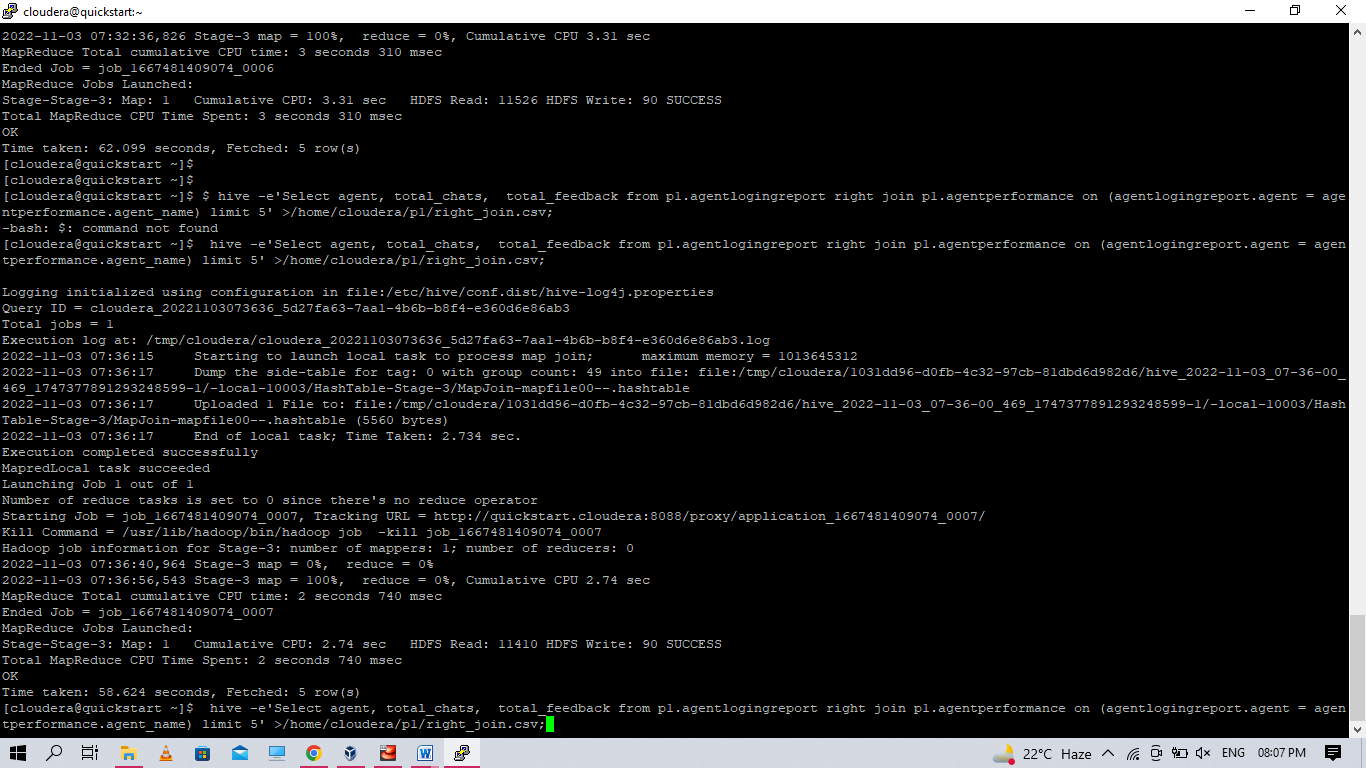
**LEFT JOIN**

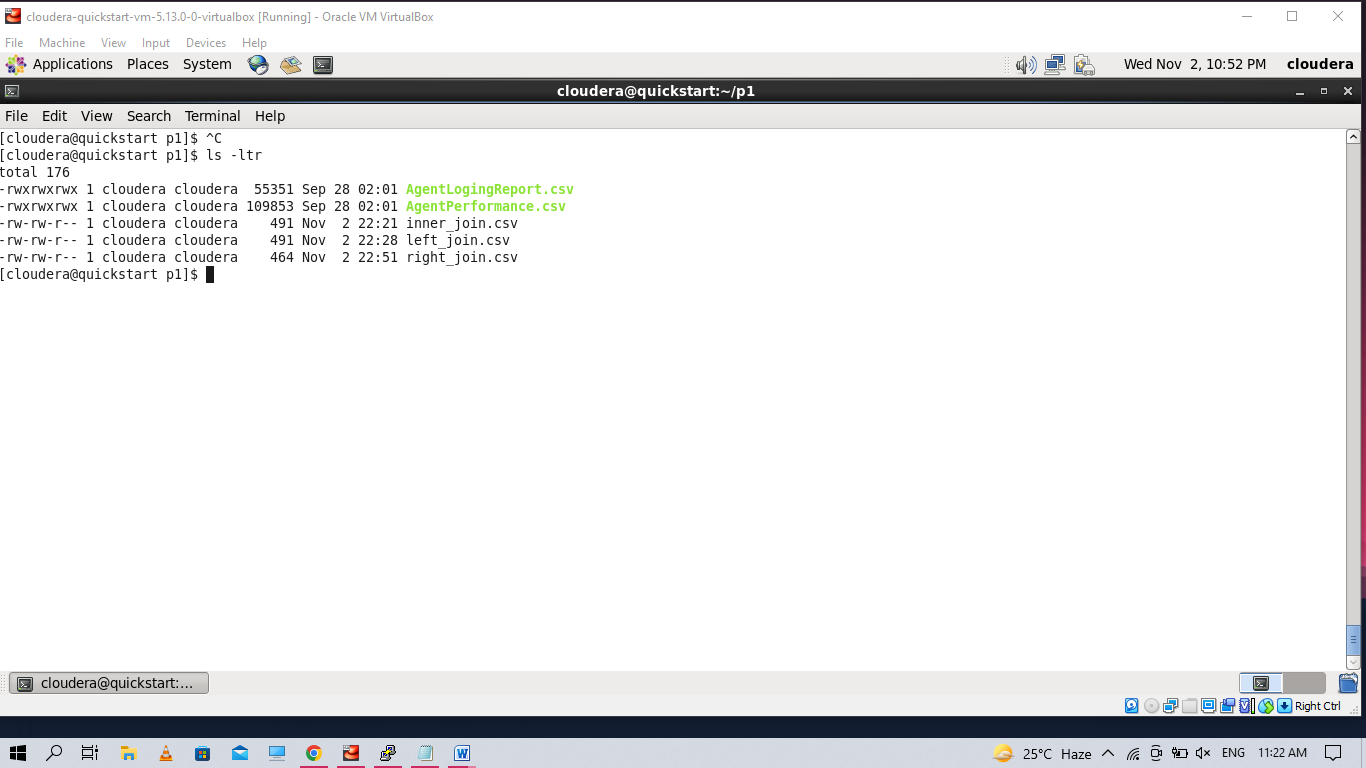
$ hive -e'Select agent, total\_chats, total\_feedback from p1.agentlogingreport left join p1.agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5' >/home/cloudera/p1/left\_join.csv;



**RIGHT JOIN**

hive -e'Select agent, total\_chats, total\_feedback from p1.agentlogingreport right join p1.agentperformance on (agentlogingreport.agent = agentperformance.agent\_name) limit 5' >/home/cloudera/p1/right\_join.csv;



****

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

hive> set hive.exec.dynamic.partition=true;

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

hive> set hive.enforce.bucketing=true;

Create table AgentLogingReport\_partition

(

sr\_no int,

Date date,

Login string,

Logout string,

Duration string

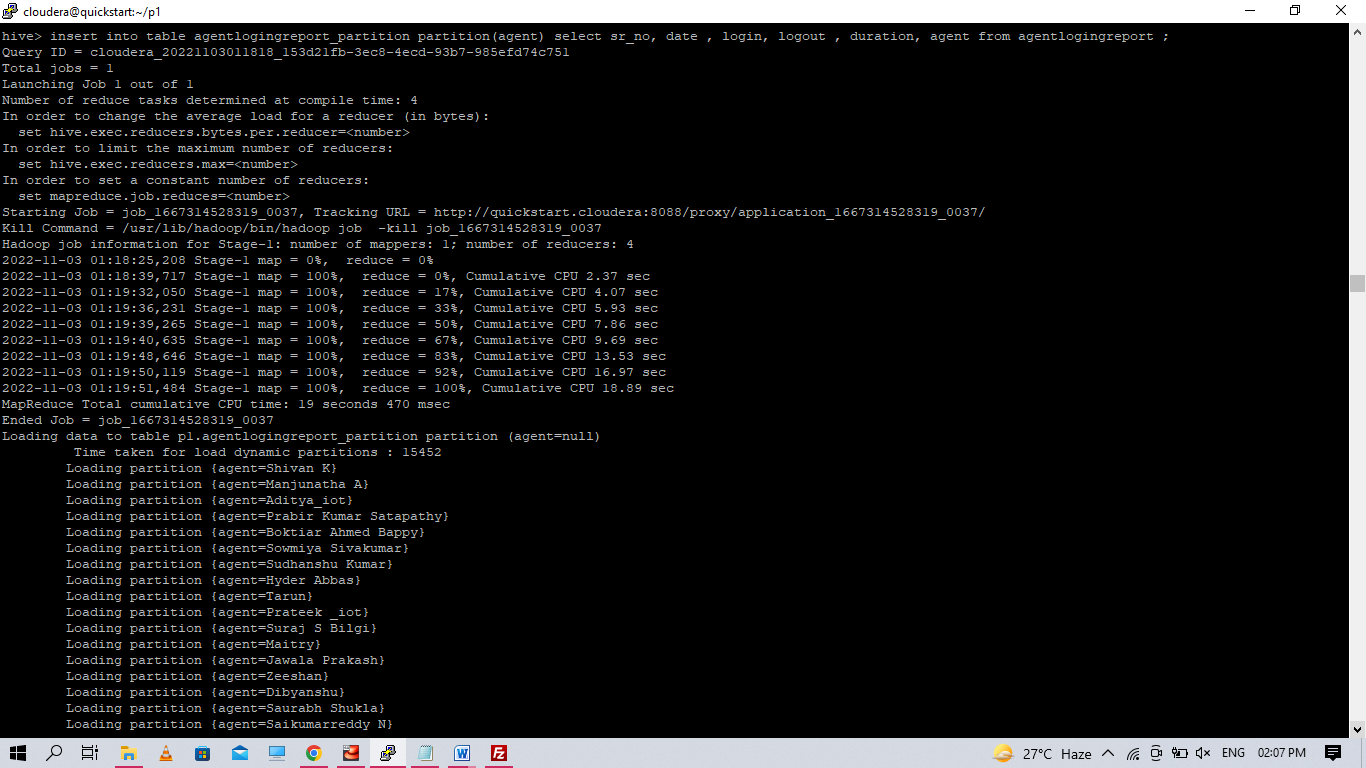
)partitioned by (Agent string)

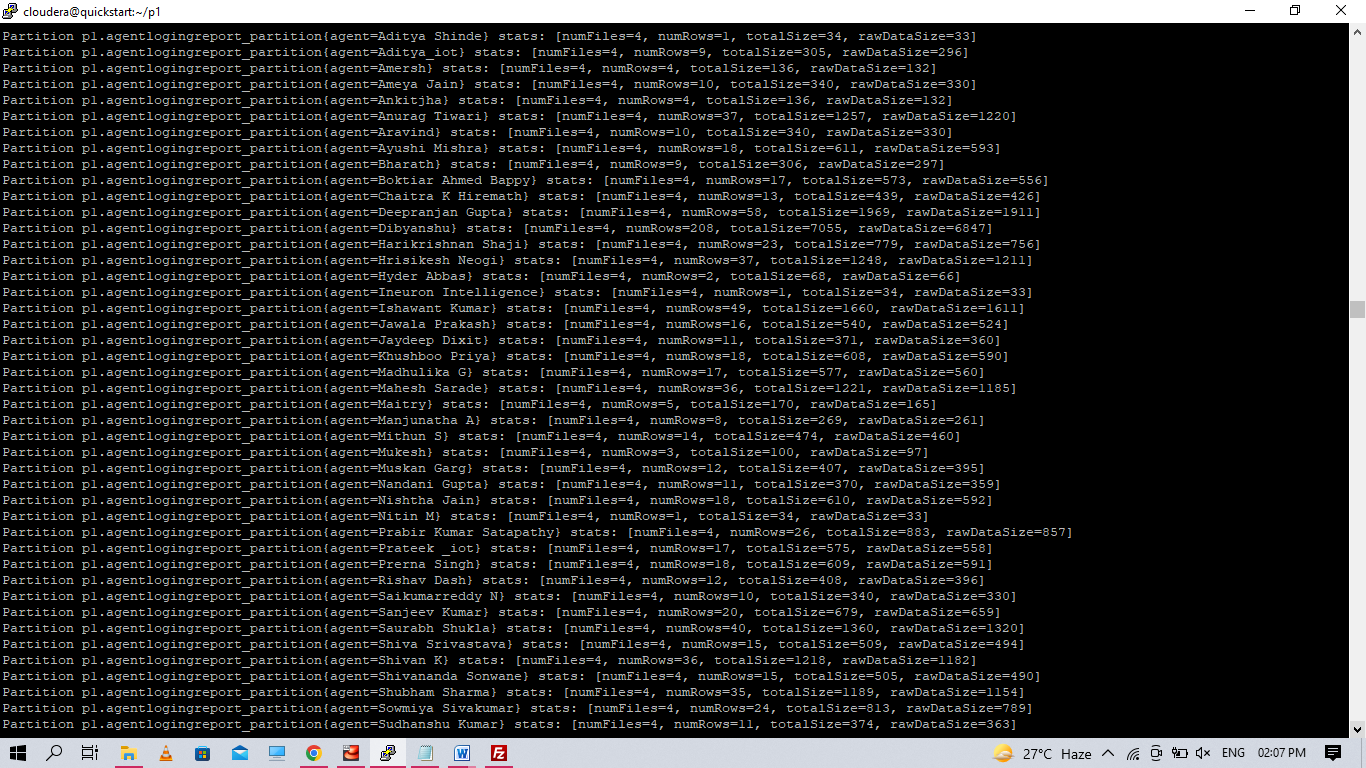
CLUSTERED BY (Date) sorted by (Date) INTO 4 BUCKETS

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';

insert into table AgentLogingReport\_partition partition(Agent) select sr\_no,Date,Login,Logout,Duration,Agent from AgentLogingReport;

****

****

create table AgentPerformance\_partition

(

sr\_no int,

Date date,

Total\_chats string,

Avg\_Response\_Time string,

Avg\_Resolution\_Time string,

Avg\_Rating float,

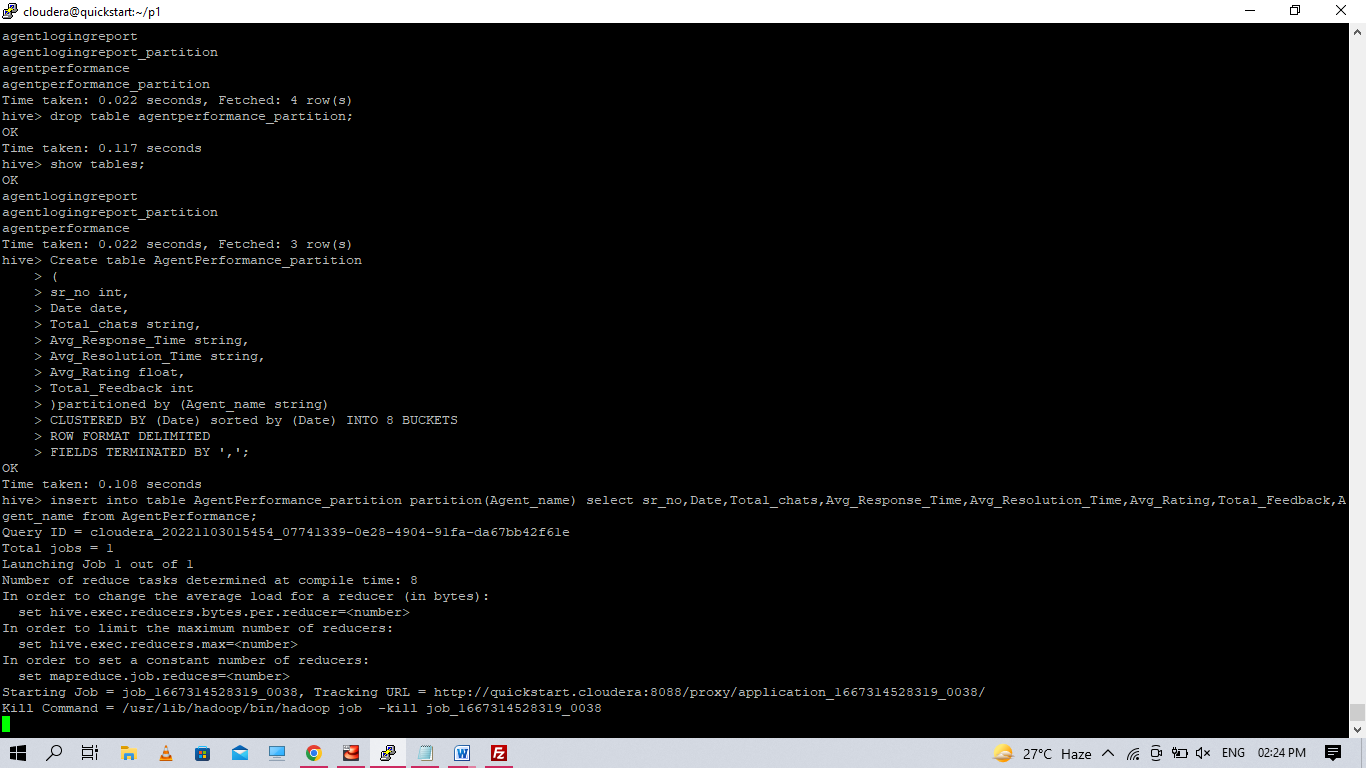
Total\_Feedback int

)partitioned by (Agent\_name string)

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

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';



Hive> insert into table AgentPerformance\_partition partition(Agent\_name) select sr\_no,Date,Total\_chats,Avg\_Response\_Time,Avg\_Resolution\_Time,Avg\_Rating,Total\_Feedback,Agent\_name from AgentPerformance;

