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

hive> create table agent\_loging\_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");

hive> create table agent\_performance

> (

> sl\_no int,

> date date,

> name string,

> total\_chats 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");

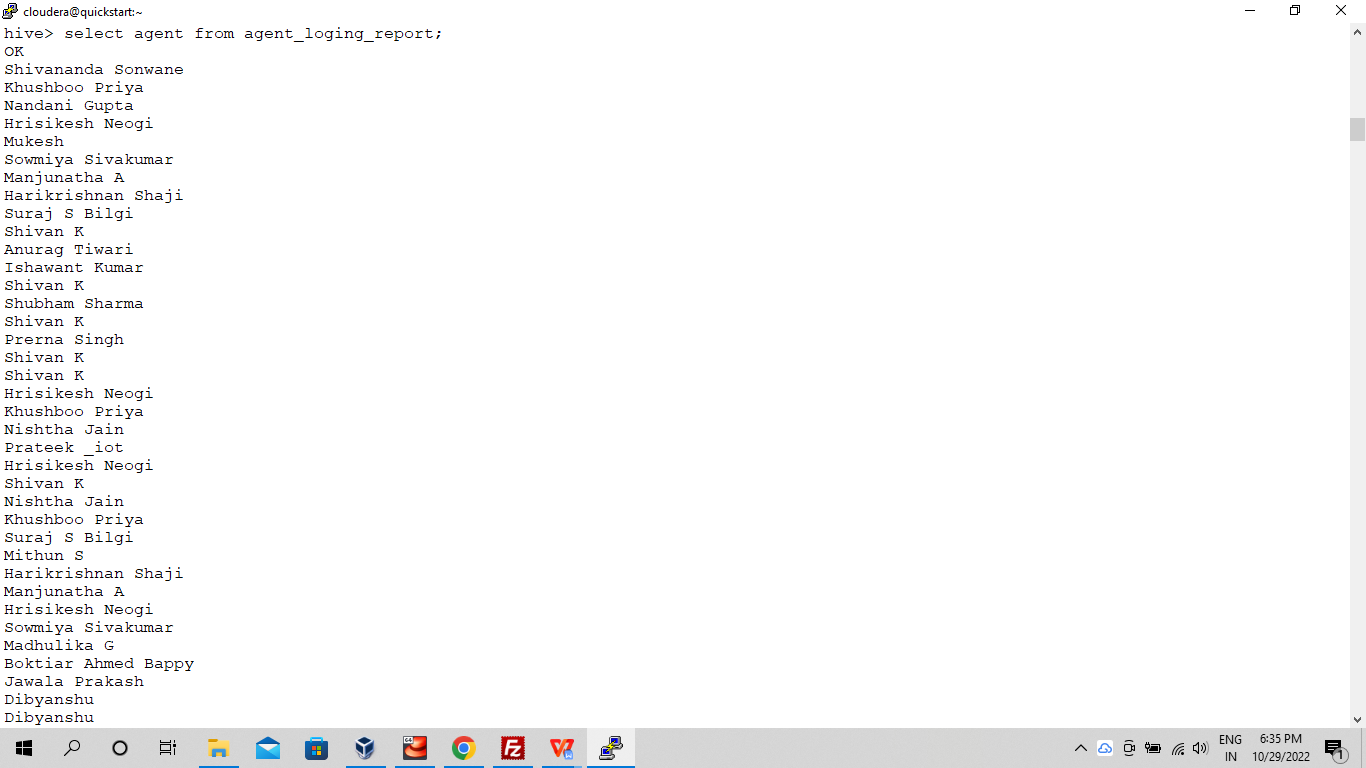
1. **Dump the data inside the hdfs in the given schema location**

hive> load data local inpath '/home/cloudera/data/AgentLogingReport.csv' into table agent\_loging\_report;

hive> load data local inpath '/home/cloudera/data/AgentPerformance.csv' into table agent\_performance;

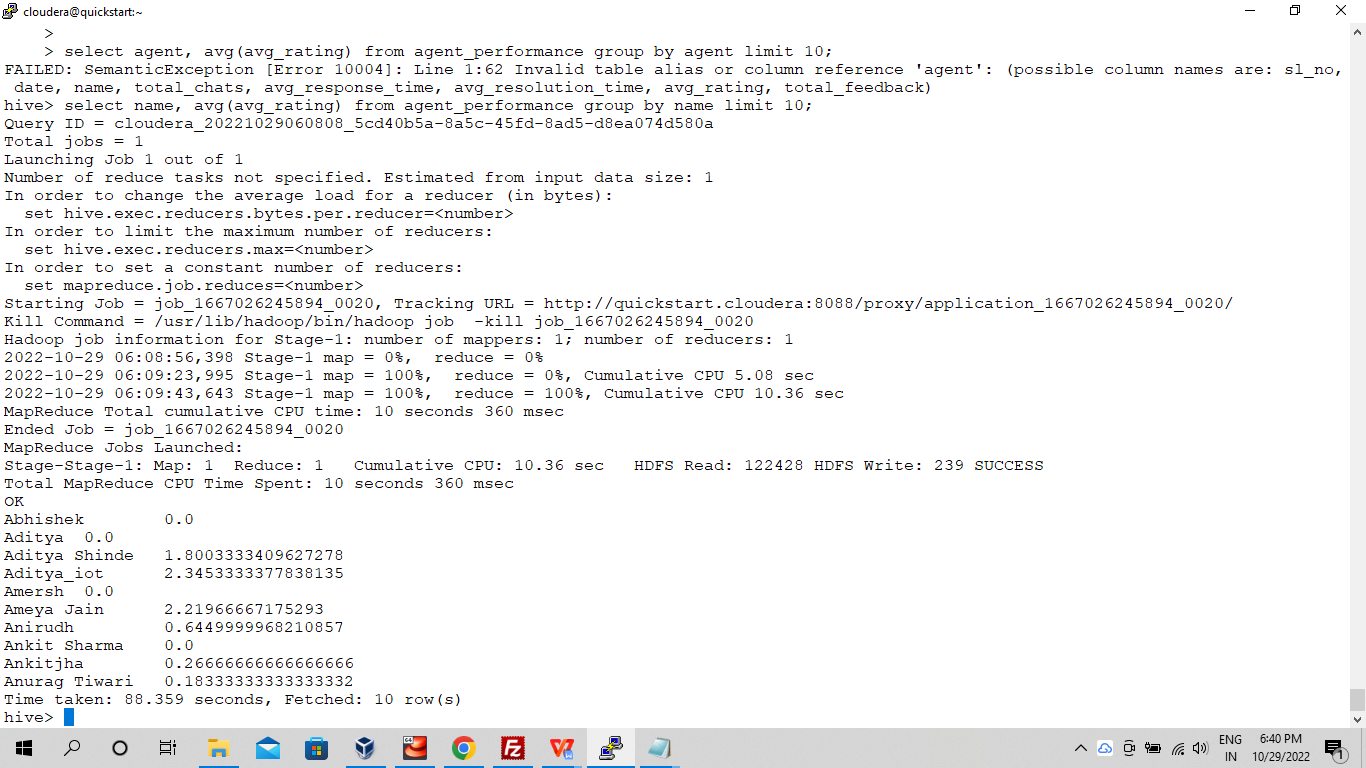
1. **List of all agents' names**

select agent from agent\_loging\_report;



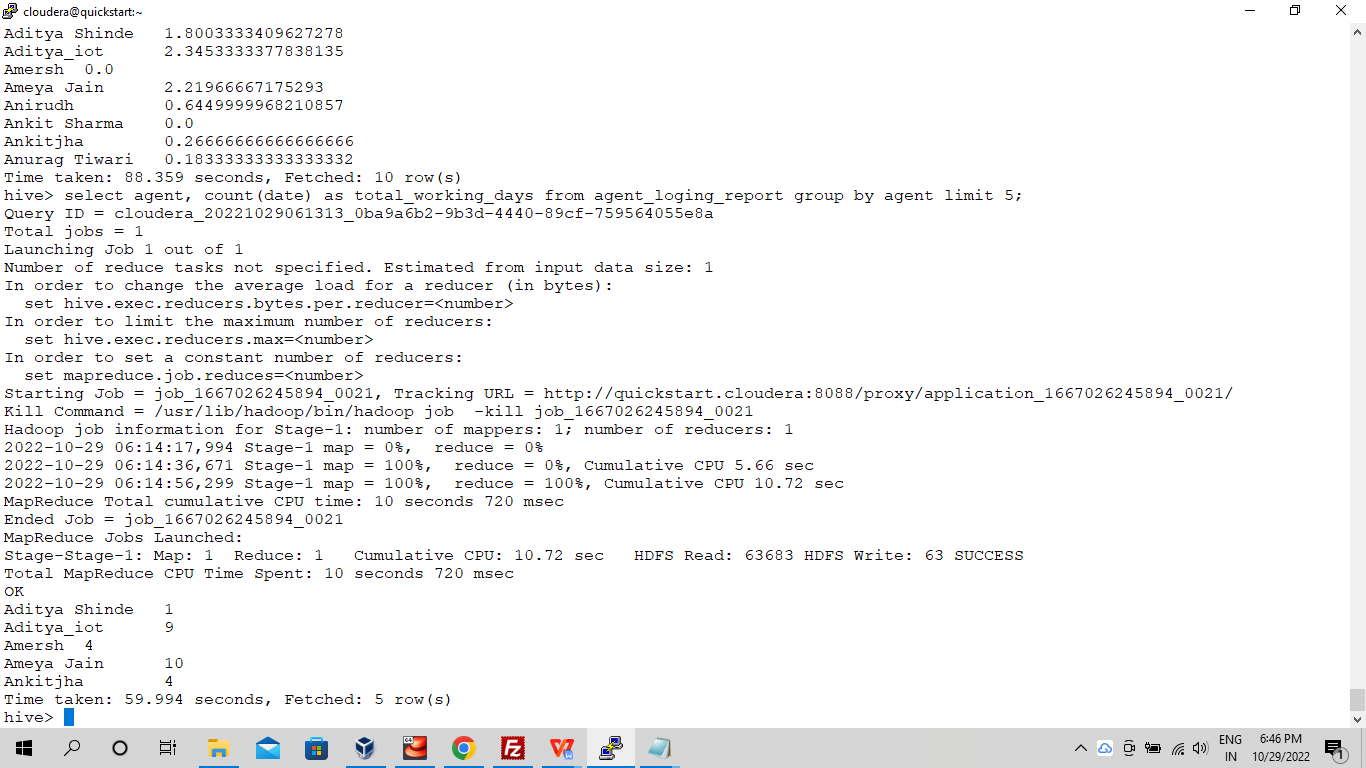
1. **Find out agent average rating**

select name, avg(avg\_rating) from agent\_performance group by name limit 10;



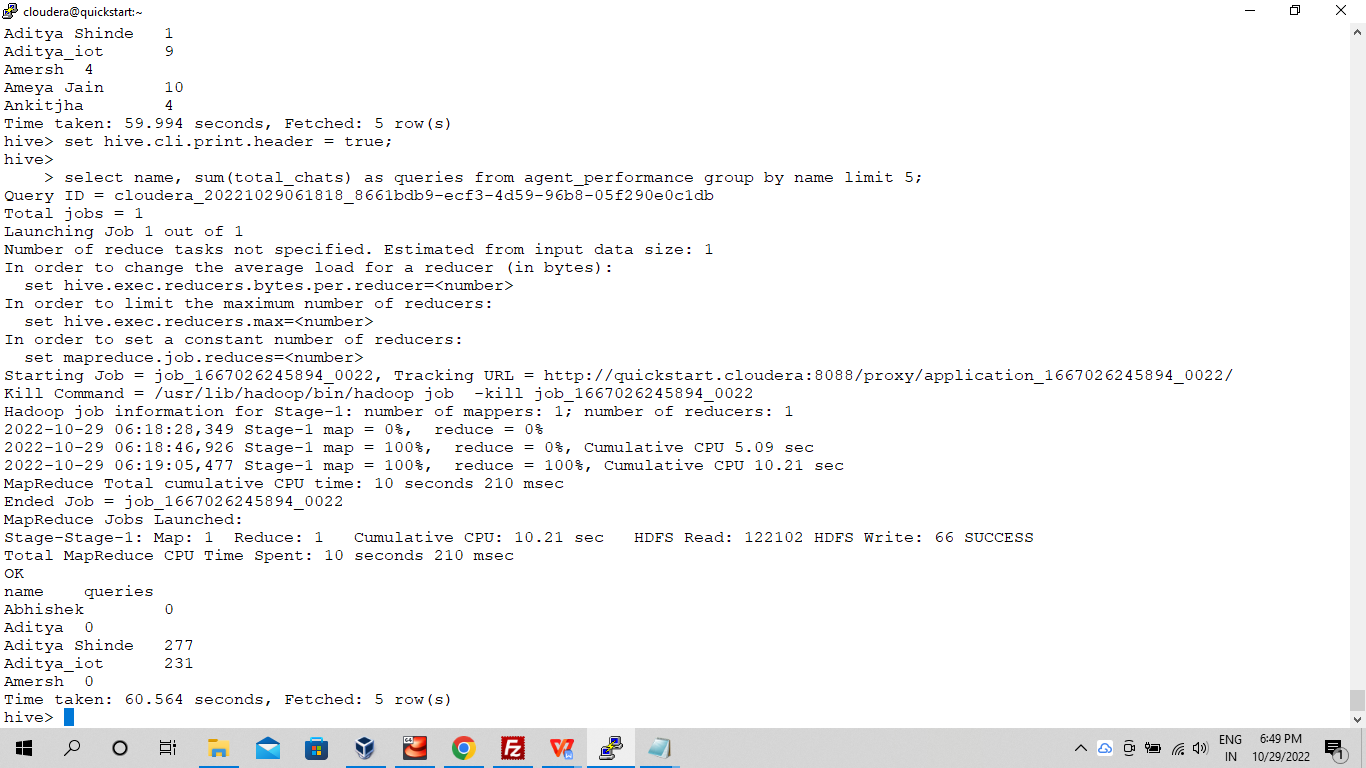
1. **Total working days for each agents**

select agent, count(date) as total\_working\_days from agent\_loging\_report group by agent limit 5;



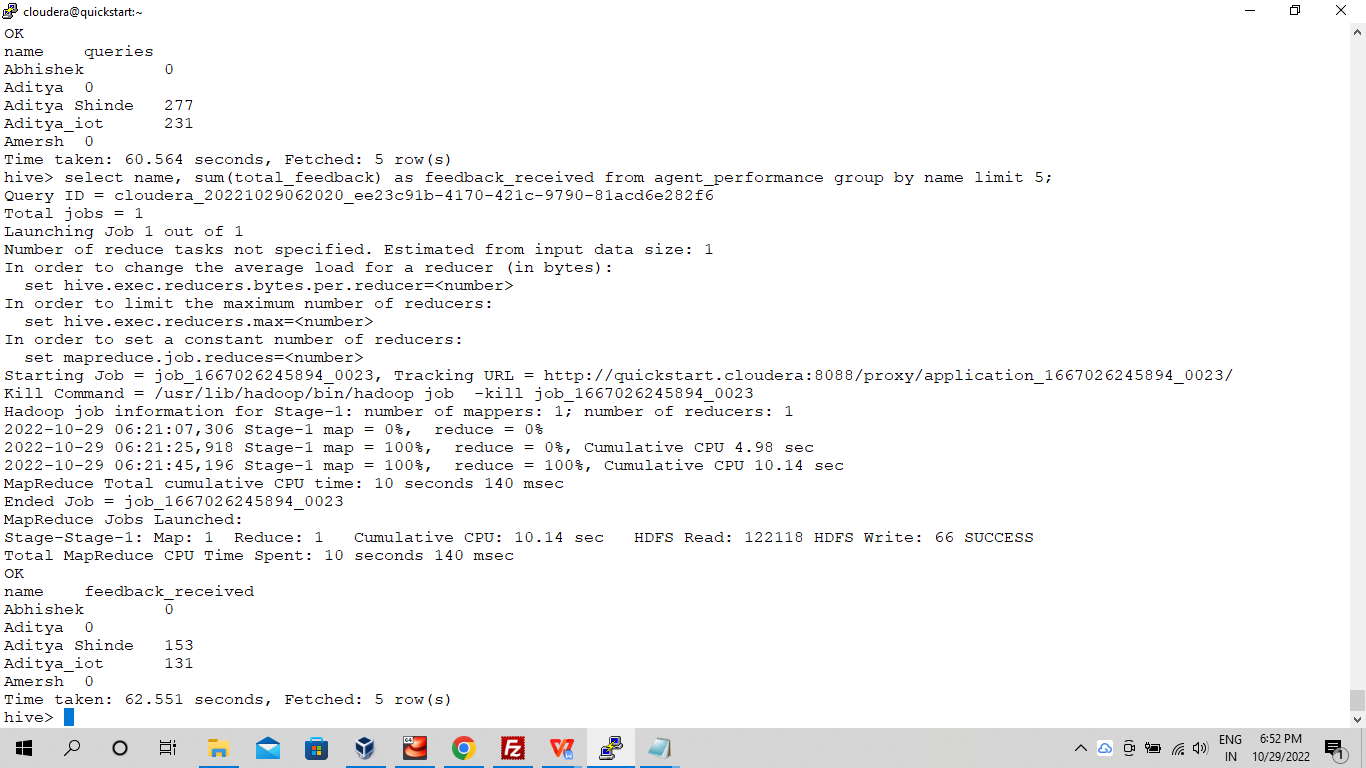
1. **Total query that each agent have taken**

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



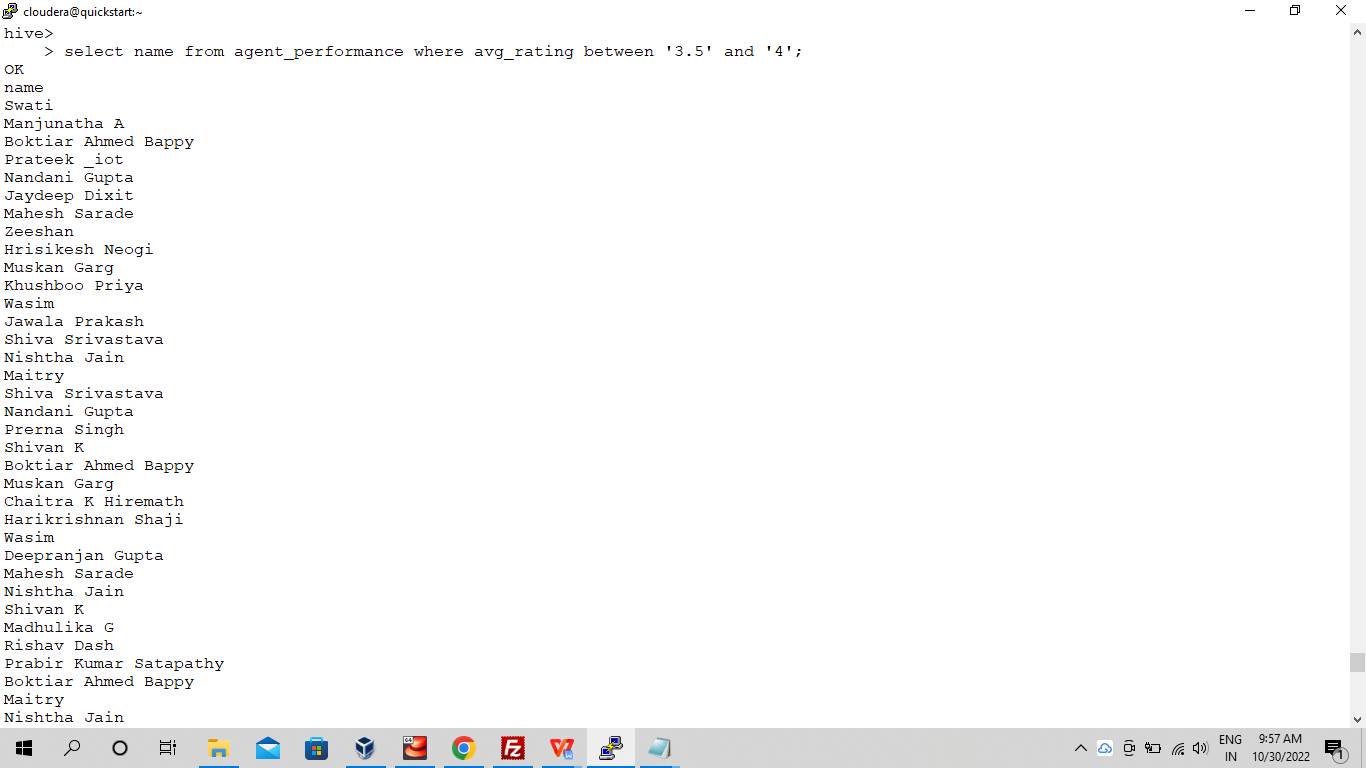
1. **Total Feedback that each agent have received**

select name, sum(total\_feedback) as feedback\_received from agent\_performance group by name limit 5;



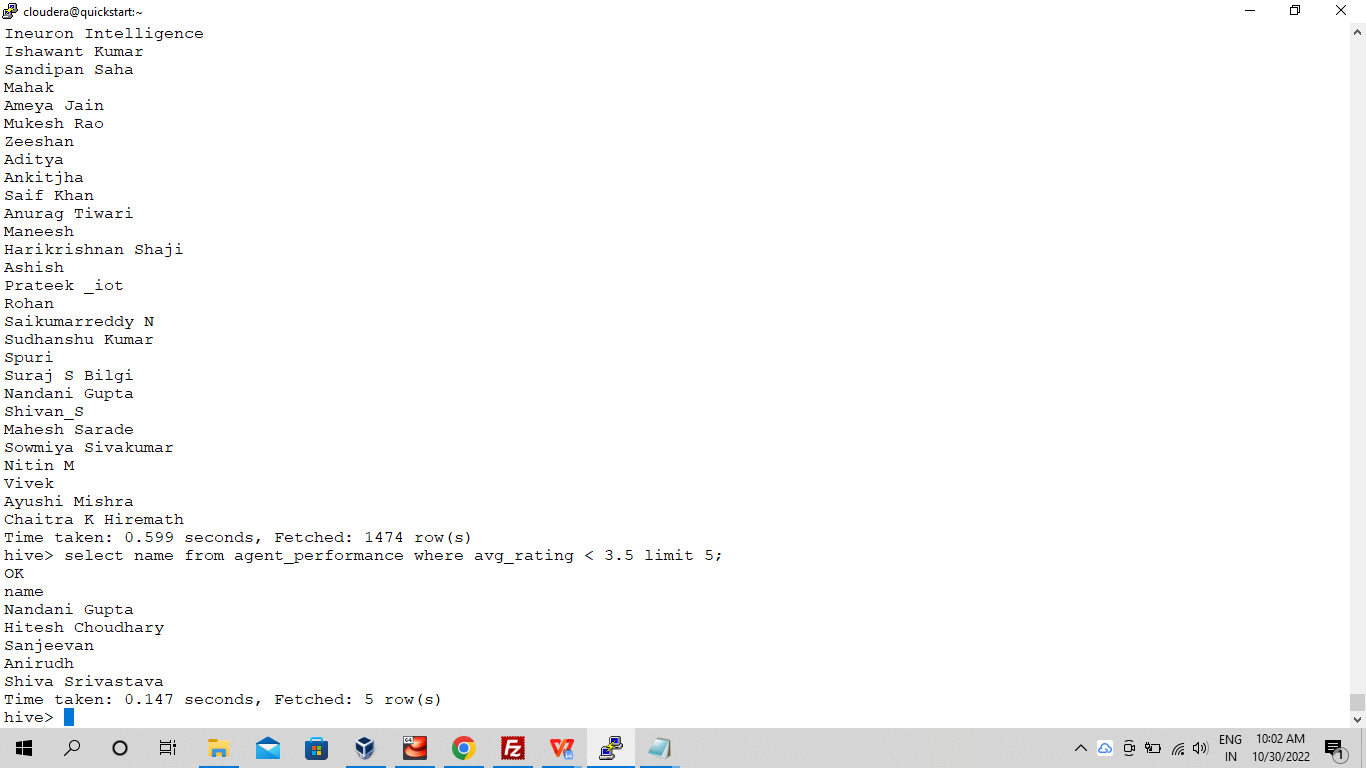
1. **Agent name who have average rating between 3.5 to 4**

select name from agent\_performance where avg\_rating between '3.5' and '4';



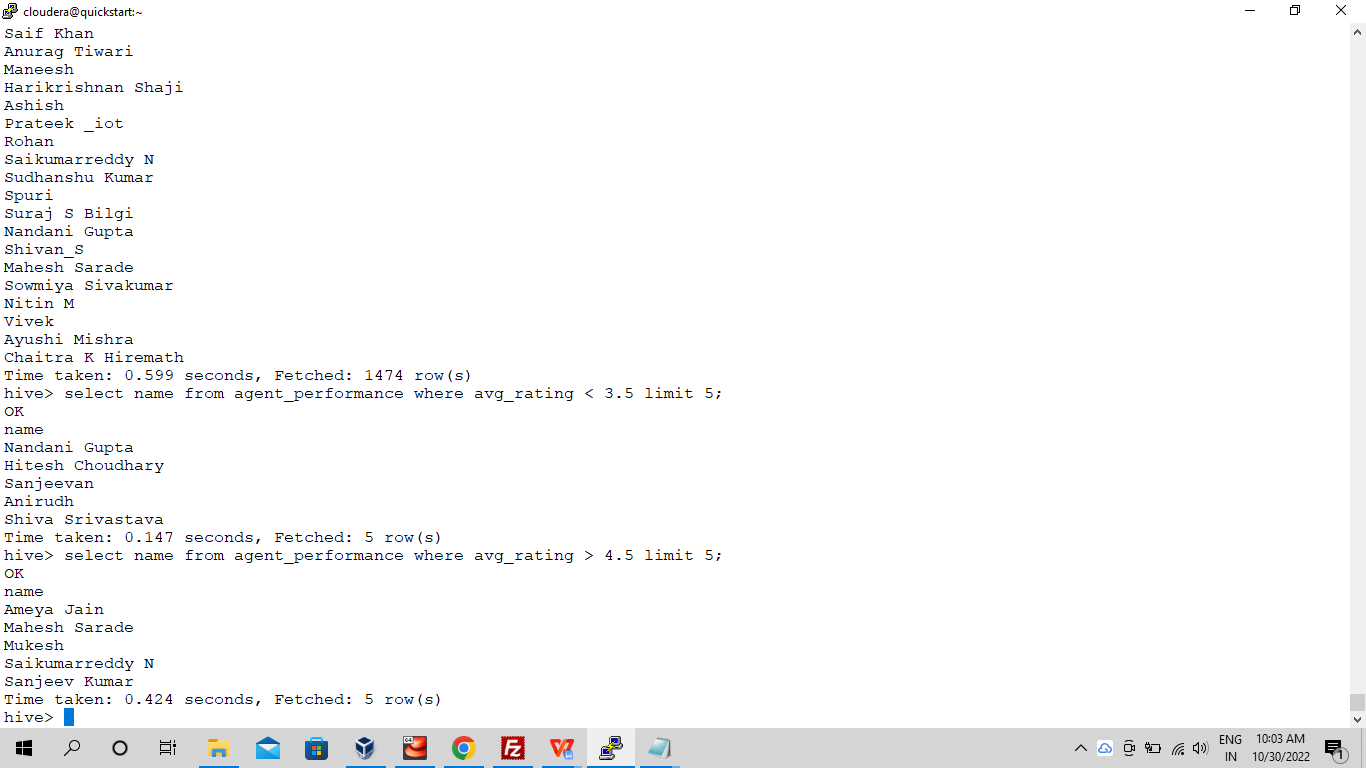
1. **Agent name who have rating less than 3.5**

select name from agent\_performance where avg\_rating < 3.5 limit 5;



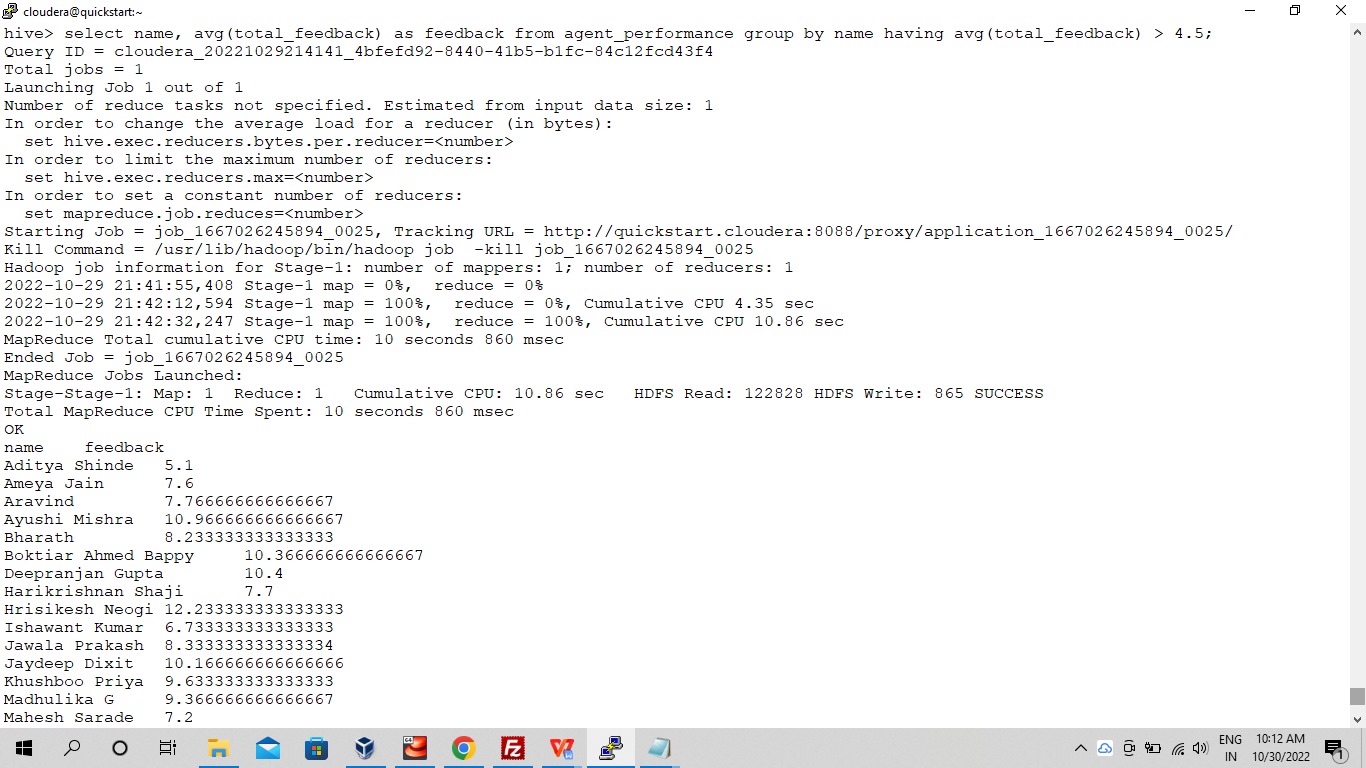
1. Agent name who have rating more than 4.5

select name from agent\_performance where avg\_rating > 4.5 limit 5;



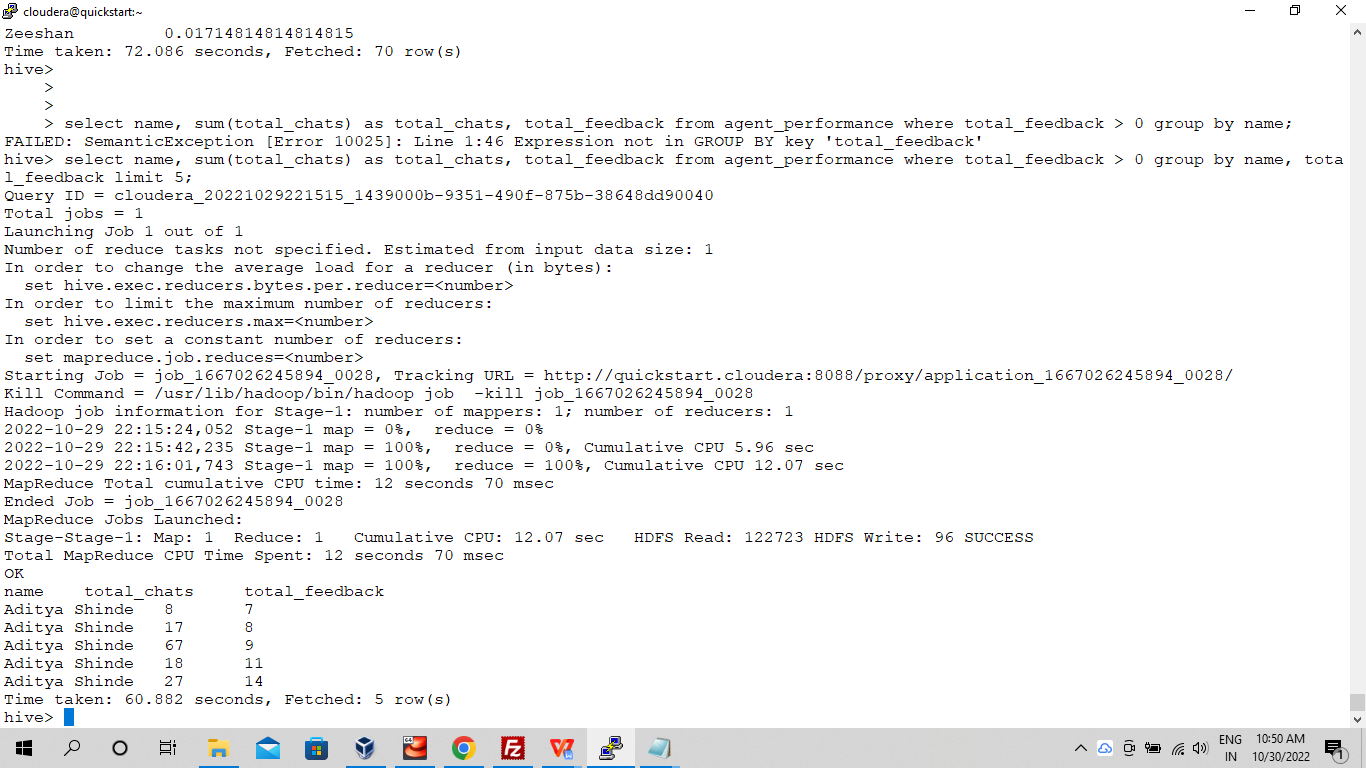
1. **How many feedback agents have received more than 4.5 average**

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



1. **average weekly response time for each agent**
2. **average weekly resolution time for each agents**
3. **Find the number of chat on which they have received a feedback**

select name, sum(total\_chats) as total\_chats, total\_feedback from agent\_performance where total\_feedback > 0 group by name, total\_feedback limit 5;



1. **Total contribution hour for each and every agents weekly basis**
2. **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 l.agent, a.total\_chats, a.avg\_response\_time from mini\_project\_1.agent\_loging\_report l inner join mini\_project\_1.agent\_performance a on l.agent = a.name limit 5;' > /home/cloudera/data/inner\_join.csv;

hive -e 'select l.agent, a.total\_chats, a.avg\_response\_time from mini\_project\_1.agent\_loging\_report l left join mini\_project\_1.agent\_performance a on l.agent = a.name limit 5;' > /home/cloudera/data/left\_join.csv;

hive -e 'select l.agent, a.total\_chats, a.avg\_response\_time from mini\_project\_1.agent\_loging\_report l right join mini\_project\_1.agent\_performance a on l.agent = a.name limit 5;' > /home/cloudera/data/right\_join.csv;

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

hive> create table agent\_details

> (

> sl\_no int,

> total\_chats int,avg\_response\_time string

> )

> partitioned by (name string)

> clustered by (sl\_no)

> sorted by (sl\_no)

> into 2 buckets;

hive> insert overwrite table agent\_details partition(name) select sl\_no, total\_chats, avg\_response\_time, name from agent\_performance;

