**Hive Mini Project-1**

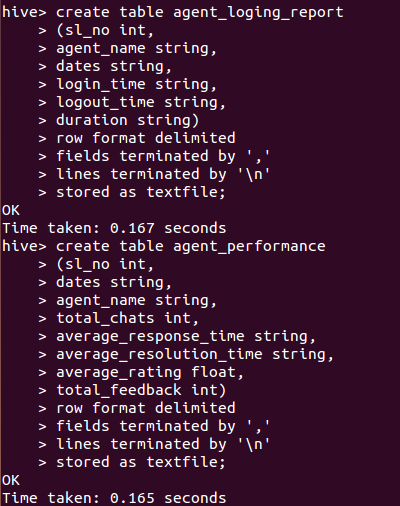
**This is a real time dataset of the I-neuron technical consultant team. You have to perform hive analysis on this given dataset.**

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

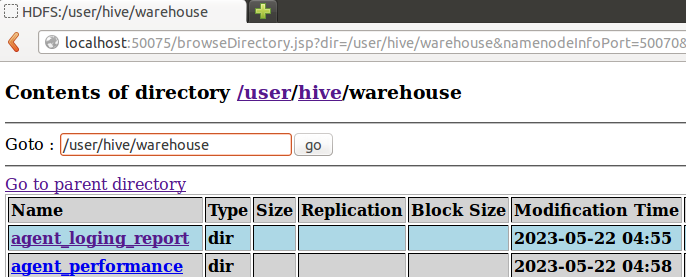
**Download Dataset 2 -** [**https://drive.google.com/file/d/1-JIPCZ34dyN6k9CqJa-Y8yxIGq6vTVXU/view**](https://drive.google.com/file/d/1-JIPCZ34dyN6k9CqJa-Y8yxIGq6vTVXU/view)

**Note: both files are csv files.**

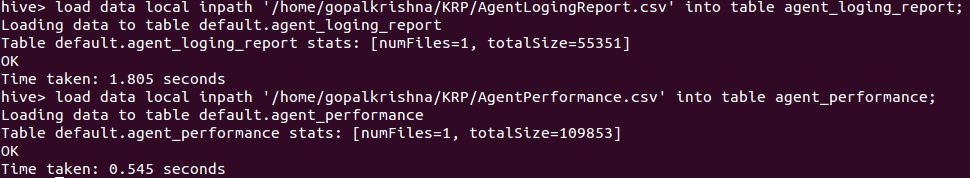
1. **Create a schema based on the given dataset.**

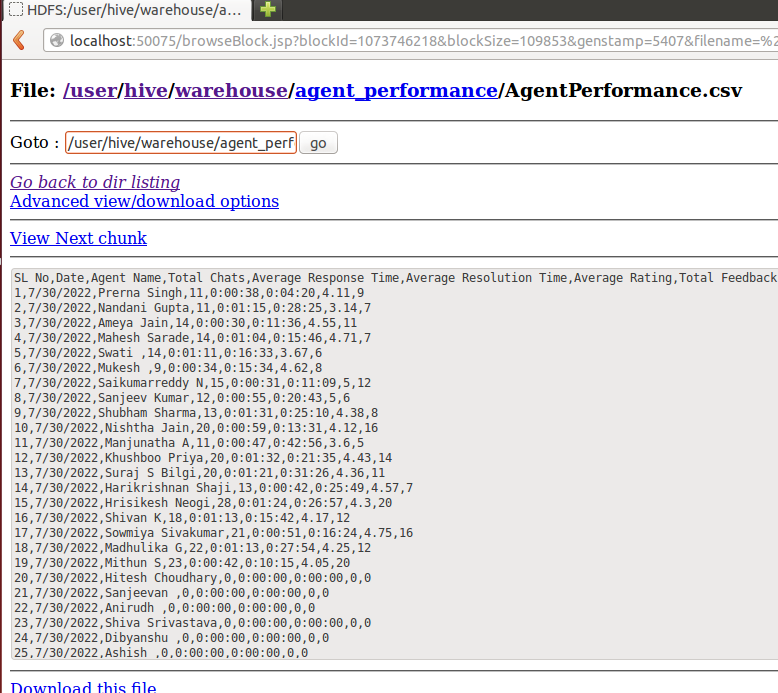
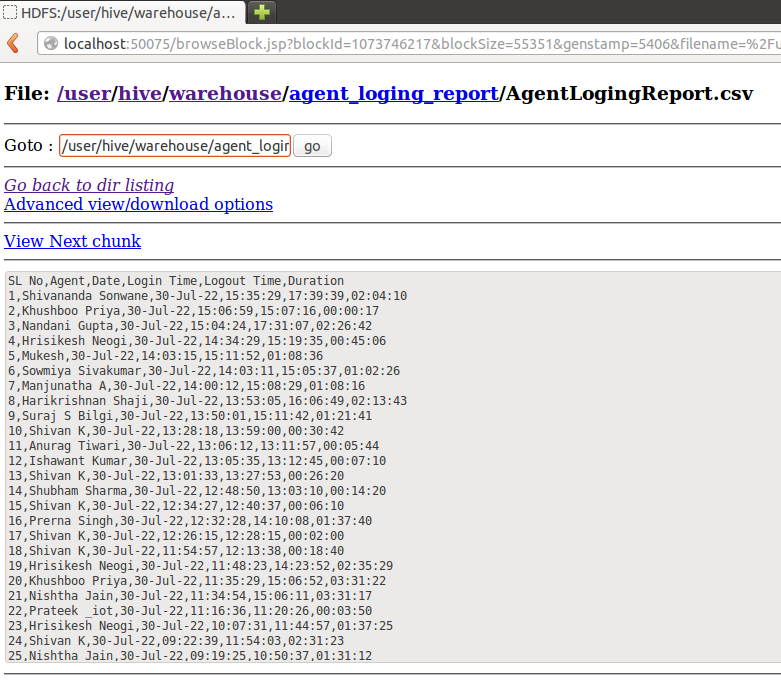
****

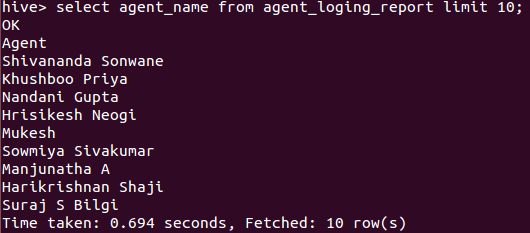
See below the directory created in HDFS.

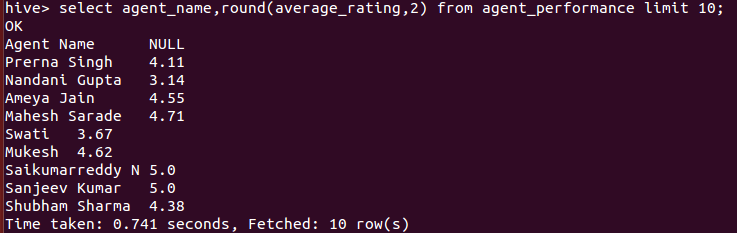
****

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

****

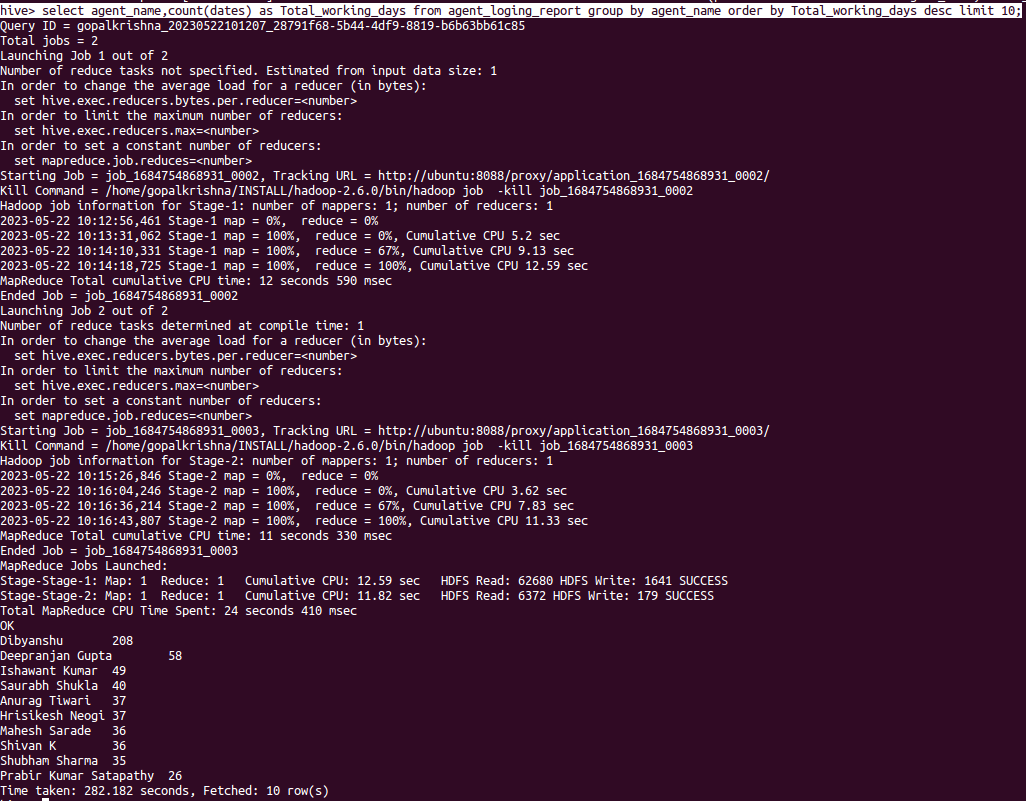
See below the data is loaded to HDFS.****

1. **List of all agents' name**
2. **Find out agent average rating.**

****

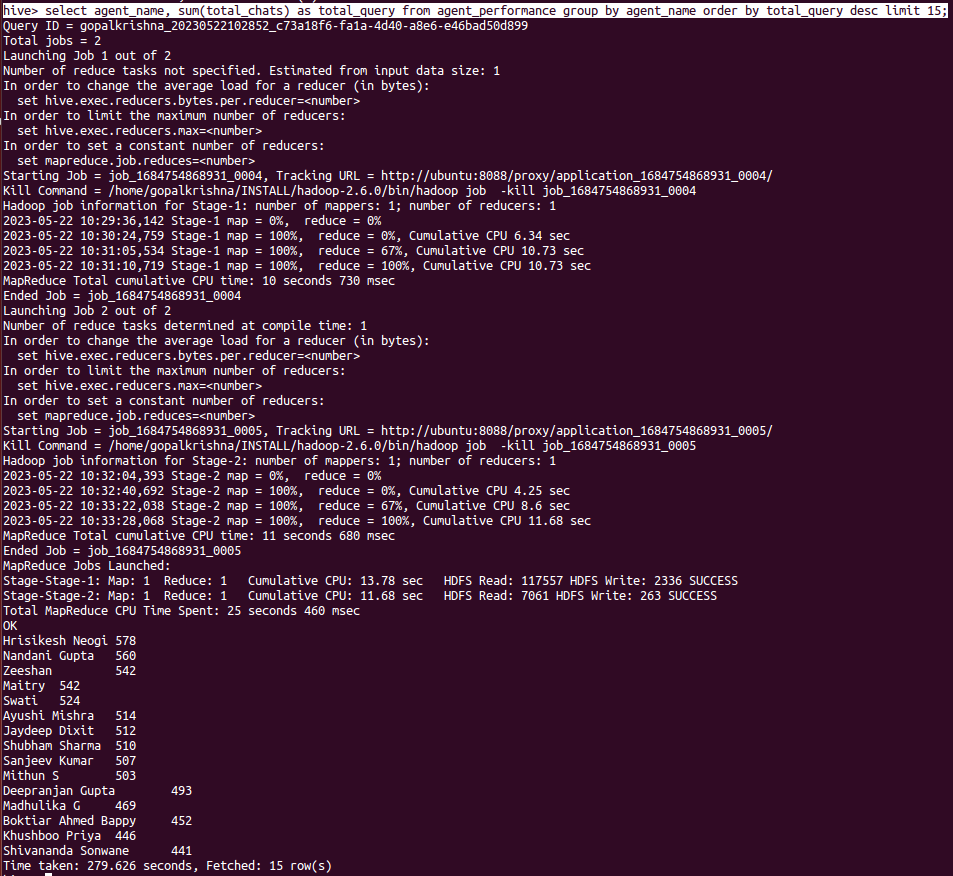
1. **Total working days for each agents.**

hive> select agent\_name,count(dates) as Total\_working\_days from agent\_loging\_report group by agent\_name order by Total\_working\_days desc limit 10;



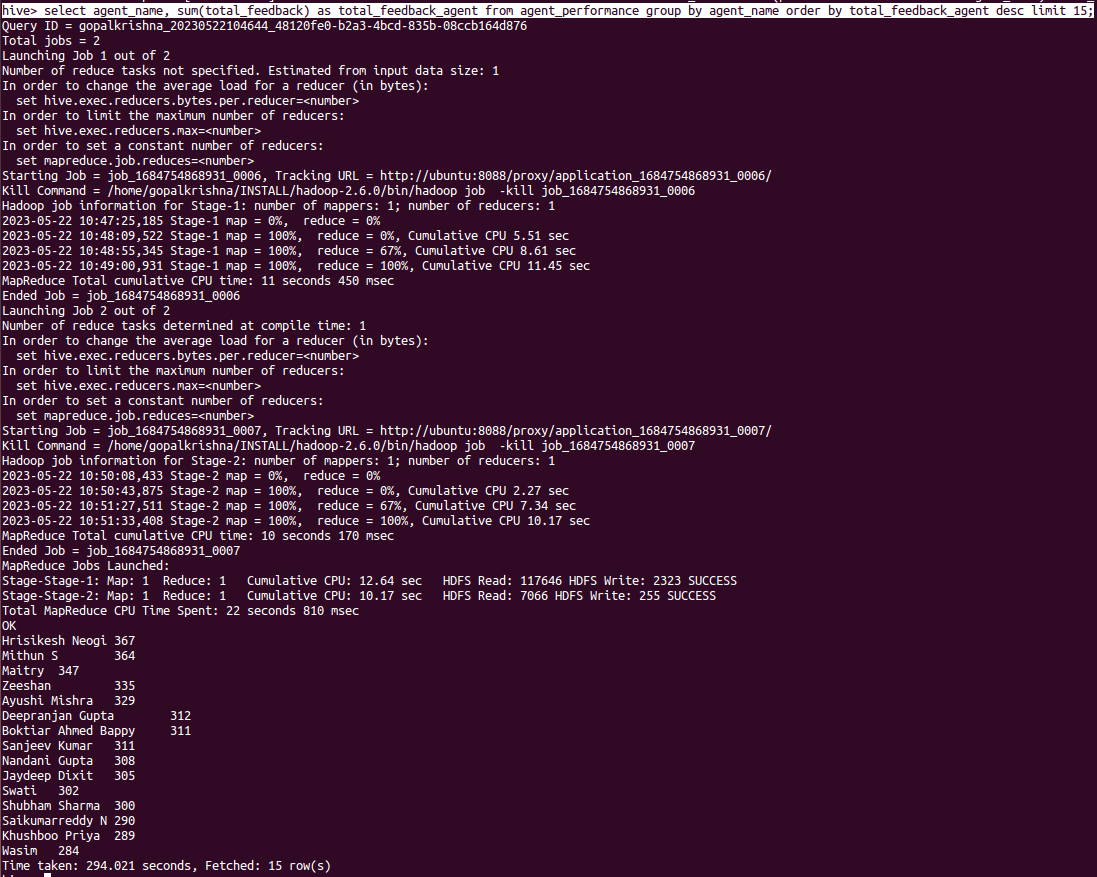
1. **Total query that each agent have taken.**

hive> select agent\_name, sum(total\_chats) as total\_query from agent\_performance group by agent\_name order by total\_query desc limit 15;

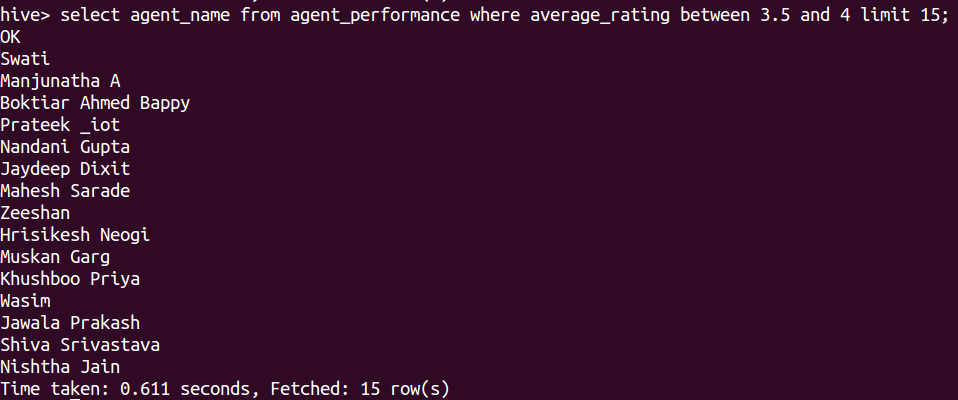


1. **Total Feedback that each agent have received.**

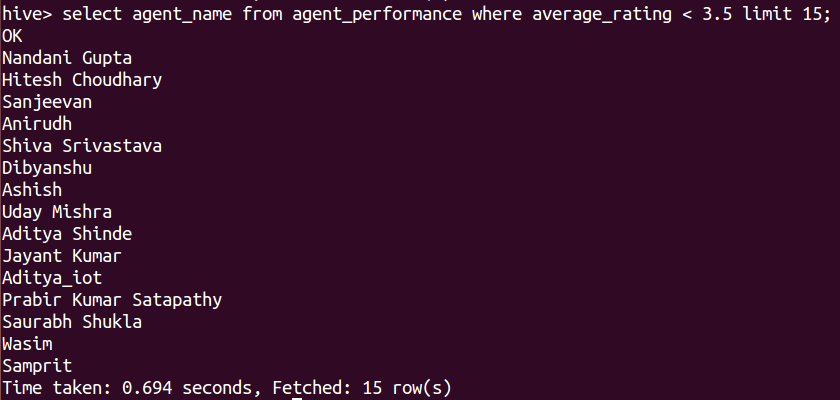
hive> select agent\_name, sum(total\_feedback) as total\_feedback\_agent from agent\_performance group by agent\_name order by total\_feedback\_agent desc limit 15;



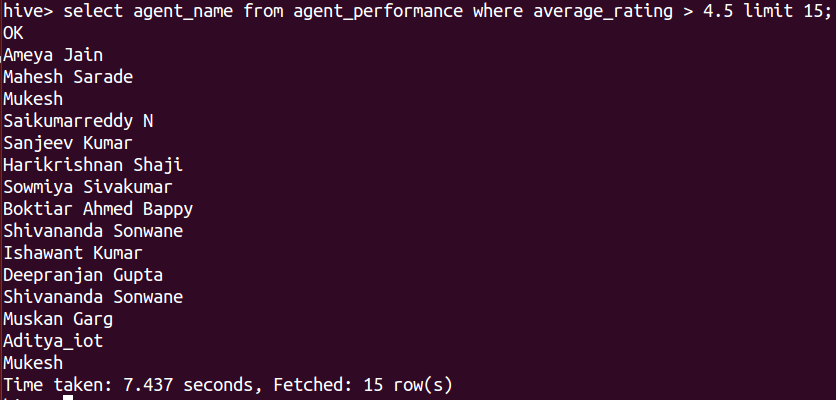
1. **Agent name who have average rating between 3.5 to 4.**

****

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

****

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

****

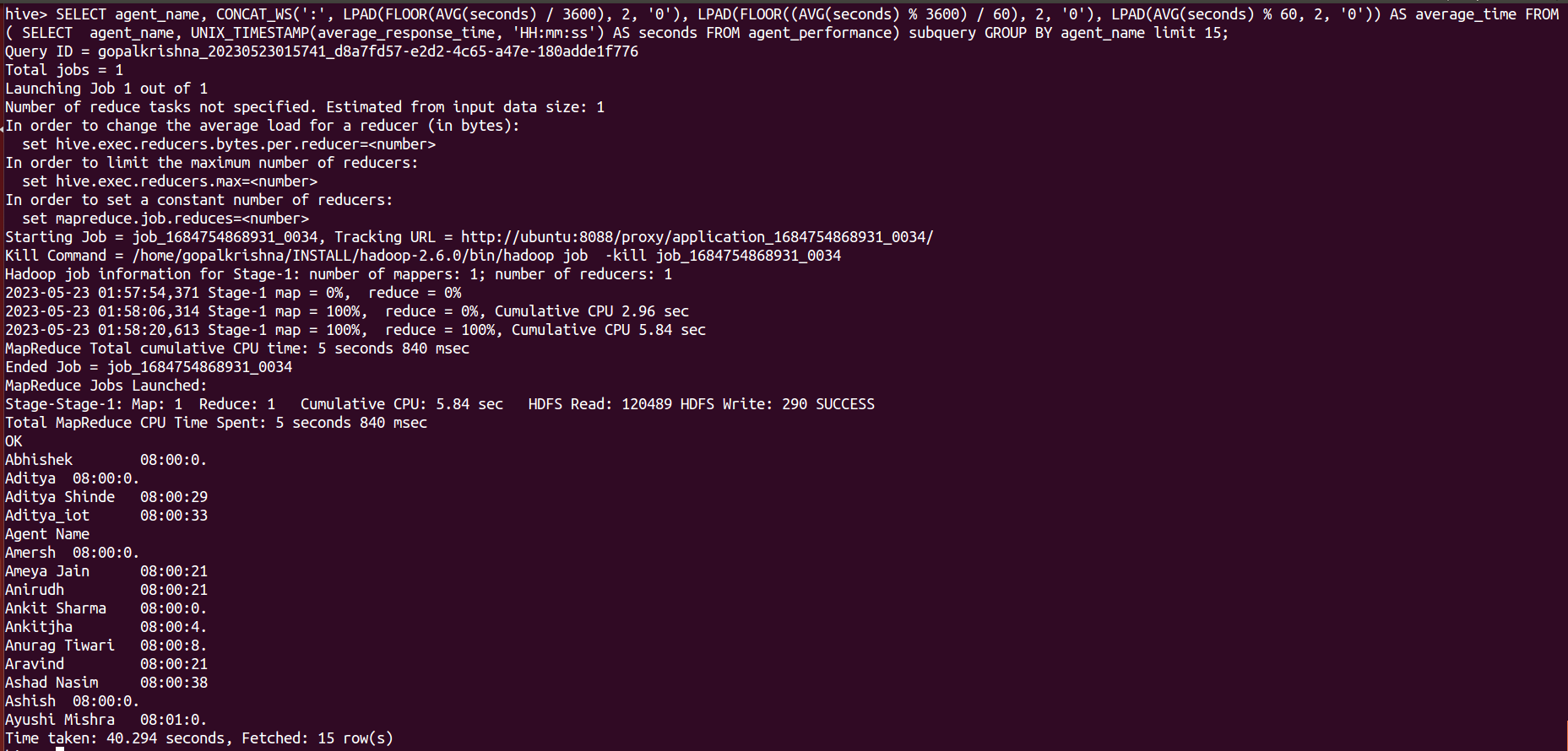
1. **How many feedback agents have received more than 4.5 average.**

hive> select agent\_name,round(avg(total\_feedback),2) from agent\_performance group by agent\_name having round(avg(total\_feedback),2) > 4.5 limit 15;

****

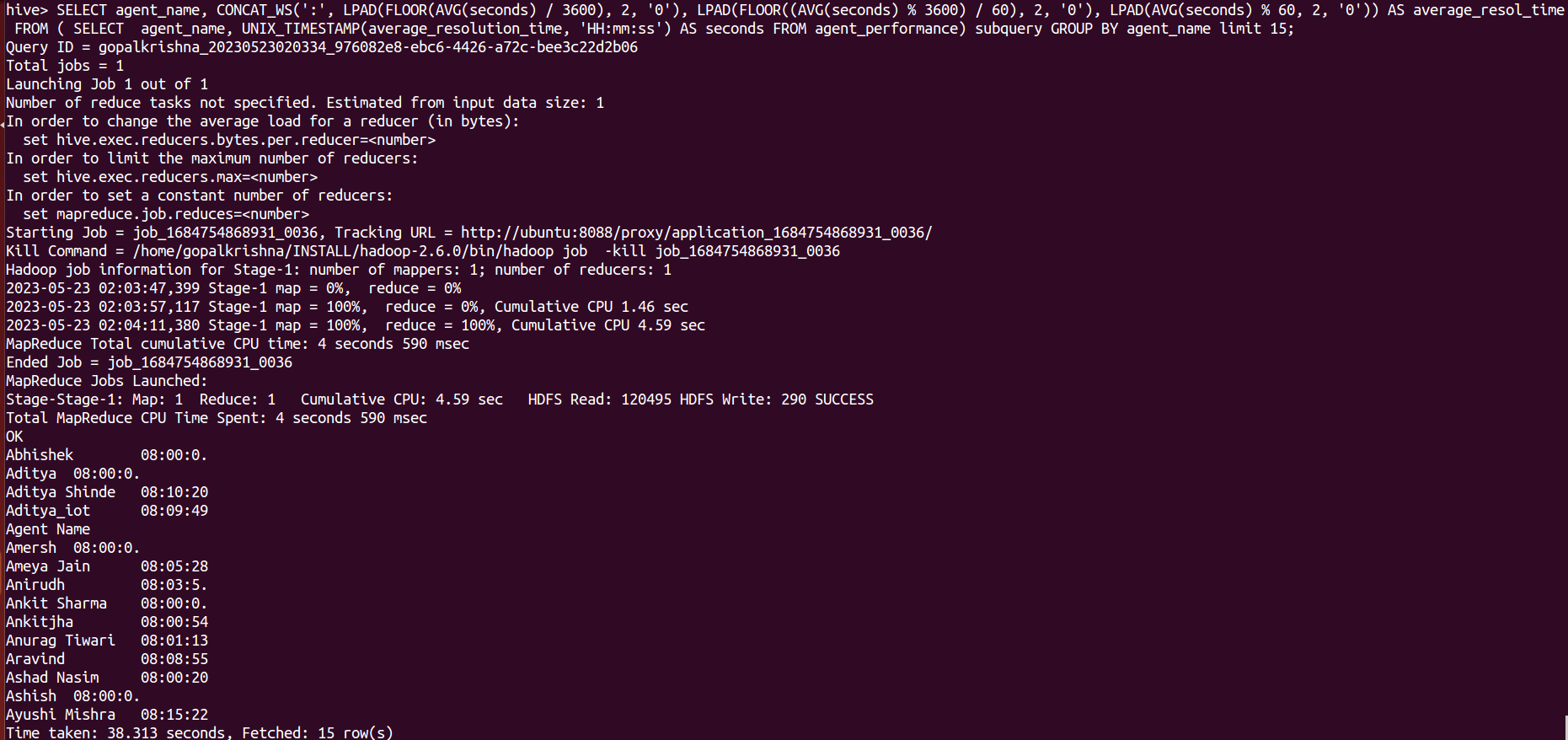
1. **average weekly response time for each agent.**

hive> SELECT agent\_name, CONCAT\_WS(':', LPAD(FLOOR(AVG(seconds) / 3600), 2, '0'), LPAD(FLOOR((AVG(seconds) % 3600) / 60), 2, '0'), LPAD(AVG(seconds) % 60, 2, '0')) AS average\_time FROM ( SELECT agent\_name, UNIX\_TIMESTAMP(average\_response\_time, 'HH:mm:ss') AS seconds FROM agent\_performance) subquery GROUP BY agent\_name limit 15;

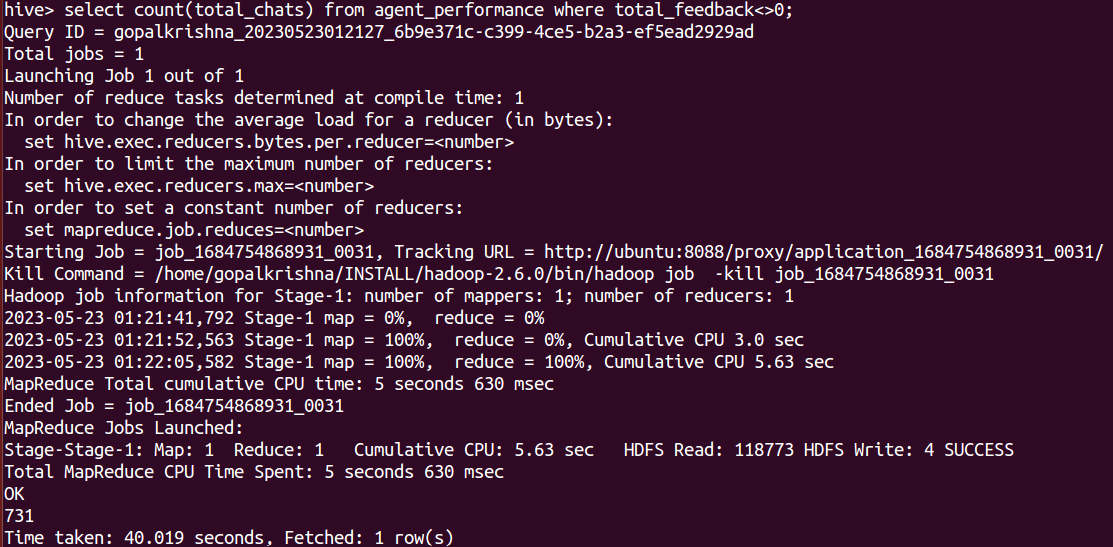


1. **average weekly resolution time for each agents.**

hive> SELECT agent\_name, CONCAT\_WS(':', LPAD(FLOOR(AVG(seconds) / 3600), 2, '0'), LPAD(FLOOR((AVG(seconds) % 3600) / 60), 2, '0'), LPAD(AVG(seconds) % 60, 2, '0')) AS average\_resol\_time FROM ( SELECT agent\_name, UNIX\_TIMESTAMP(average\_resolution\_time, 'HH:mm:ss') AS seconds FROM agent\_performance) subquery GROUP BY agent\_name limit 15;

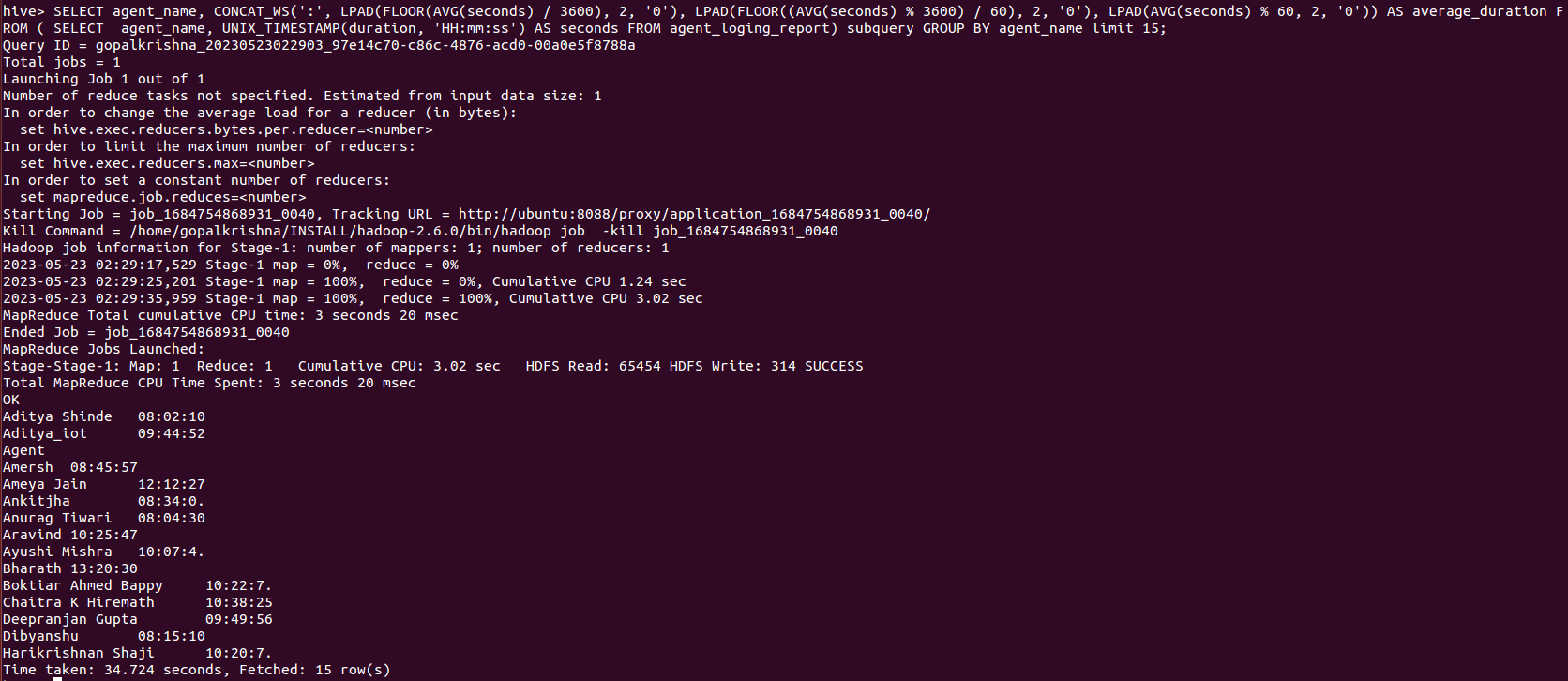
****

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

****

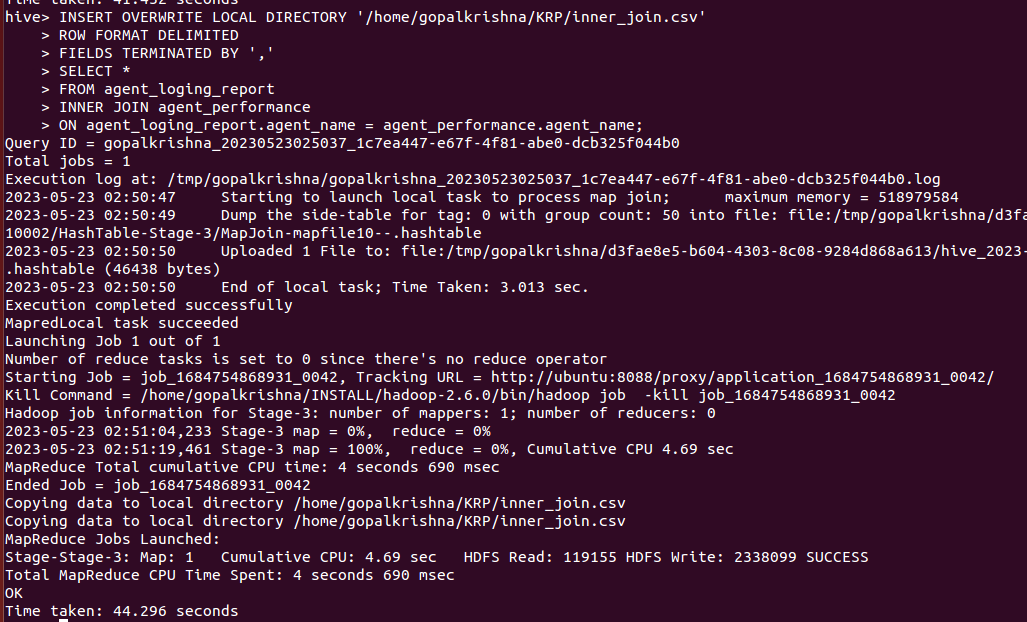
1. **Total contribution hour for each and every agents weekly basis.**

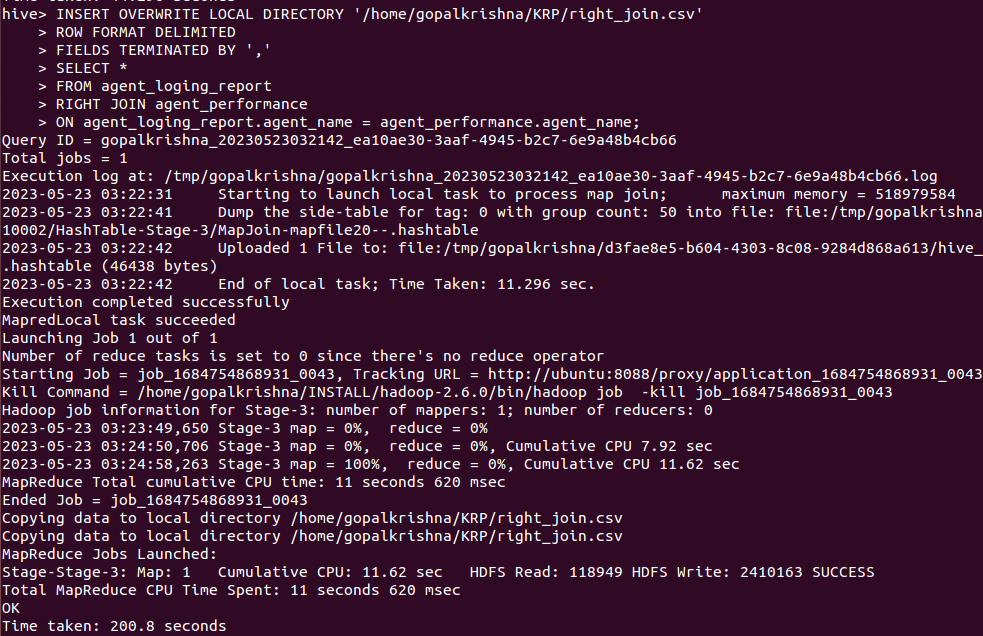
hive> SELECT agent\_name, CONCAT\_WS(':', LPAD(FLOOR(AVG(seconds) / 3600), 2, '0'), LPAD(FLOOR((AVG(seconds) % 3600) / 60), 2, '0'), LPAD(AVG(seconds) % 60, 2, '0')) AS average\_duration FROM ( SELECT agent\_name, UNIX\_TIMESTAMP(duration, 'HH:mm:ss') AS seconds FROM agent\_loging\_report) subquery GROUP BY agent\_name limit 15;



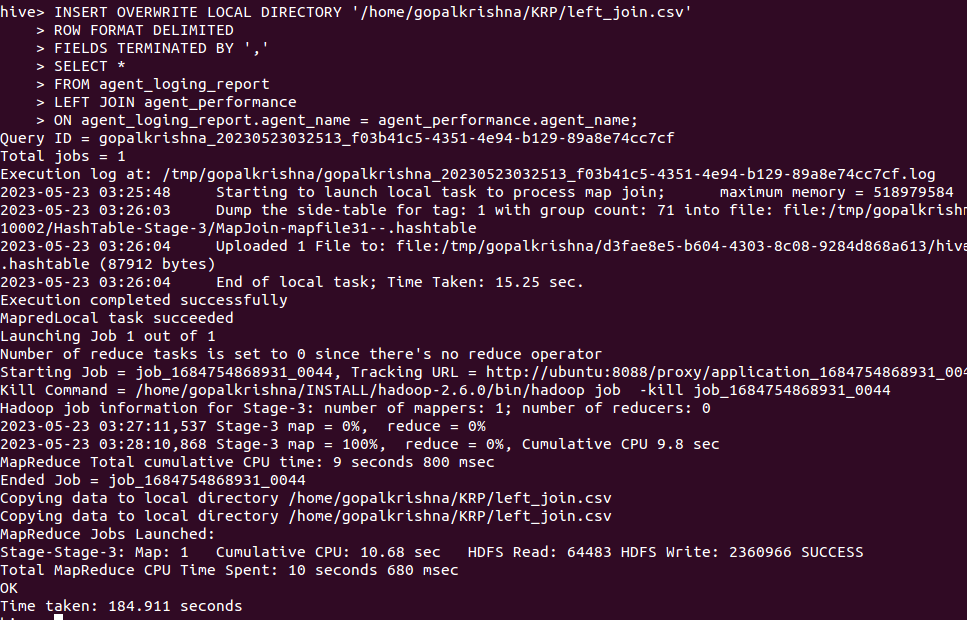
1. **Perform inner join, left join and right join based on the agent column and after joining the table export that data into your local system.**

INNER JOIN:-

****

RIGHT JOIN:-****

LEFT JOIN:-

****

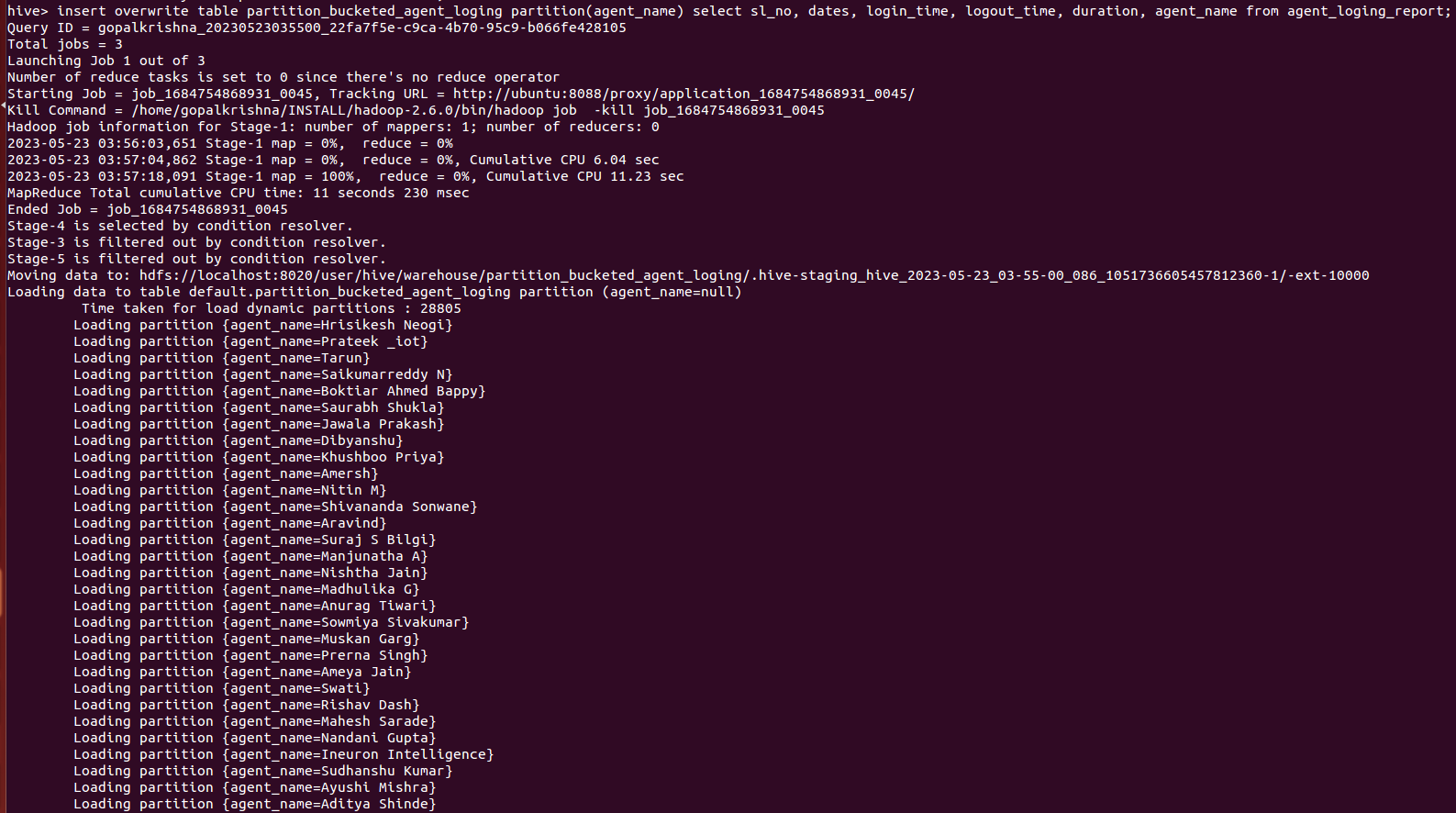
1. **Perform partitioning on top of the agent column and then on top of that perform bucketing for each partitioning.**

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

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



hive> insert overwrite table partition\_bucketed\_agent\_loging partition(agent\_name) select sl\_no, dates, login\_time, logout\_time, duration, agent\_name from agent\_loging\_report;

****