**Hive\_Project\_01**

This is a real time dataset of the iNeuron technical consultant team. We 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

Note: both files are csv files.

1. Create a schema based on the given dataset

**Hive> Create table AgentLogingReport**

**(**

**SL\_No int,**

**Agent string,**

**Dates date,**

**Login string,**

**Logout string,**

**Duration string**

**)**

**row format delimited**

**fields terminated by ','**

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

**Hive> Create table AgentPerformance**

**(**

**SL\_No int,**

**Dates date,**

**Agent\_Name string,**

**Total\_chats int,**

**Avg\_Response\_Time string,**

**Avg\_Resolution\_Time string,**

**Avg\_Rating float,**

**Total\_Feedback int**

**)**

**row format delimited**

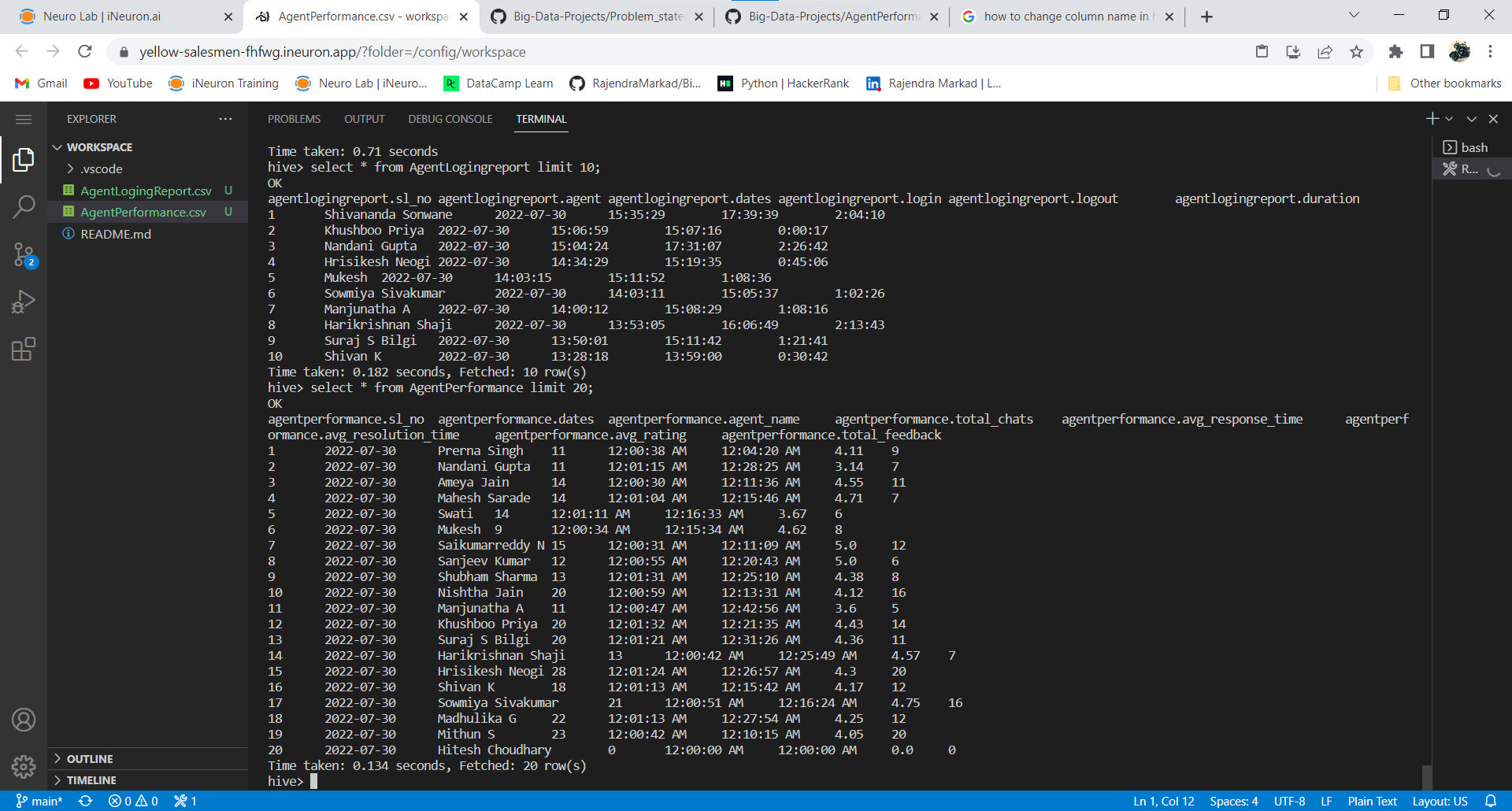
**fields terminated by ','**

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

2. load data in the given schema/tables.

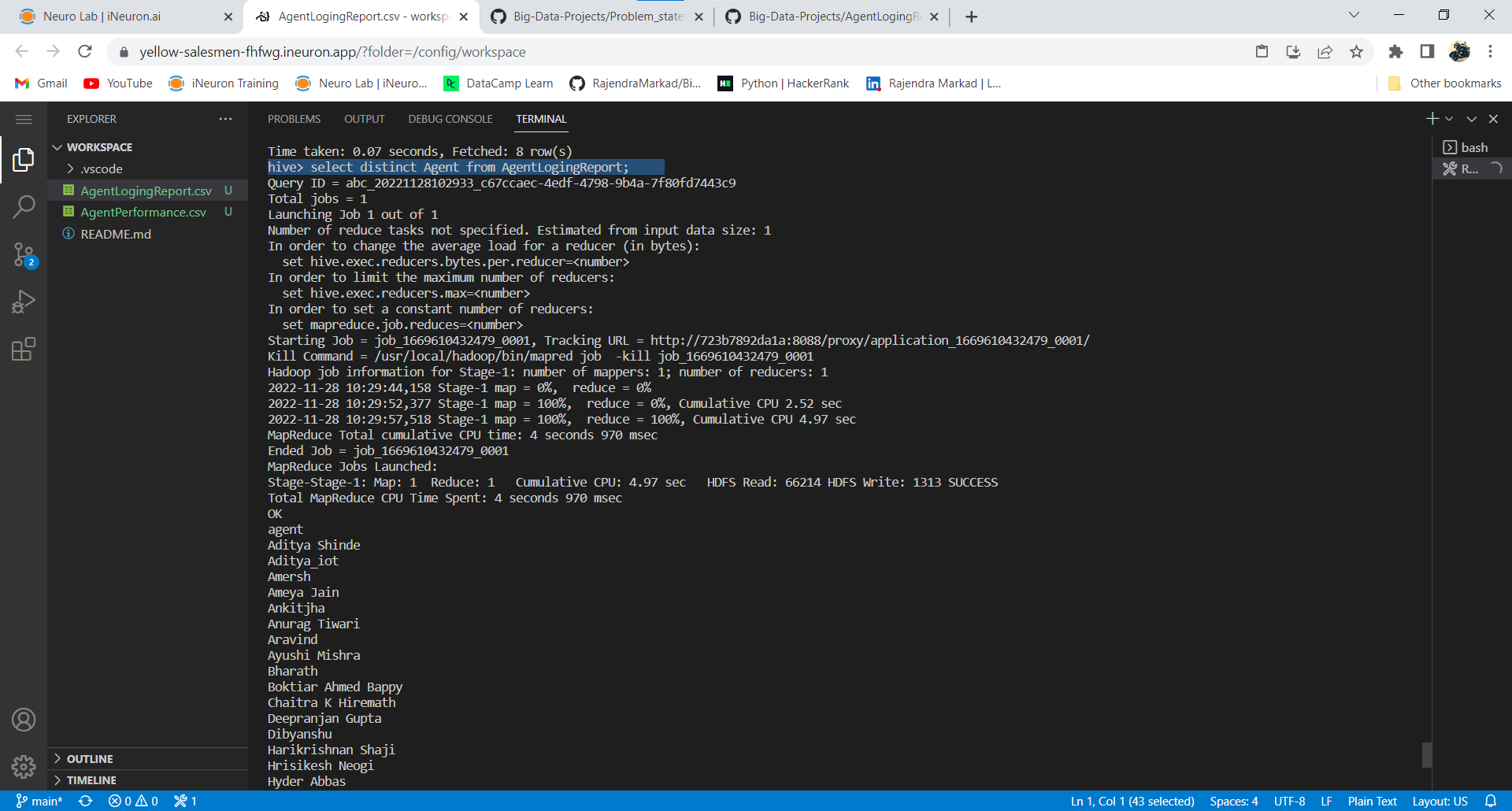
**Load data local inpath '/config/workspace/AgentLogingReport.csv' into table AgentLogingReport;**

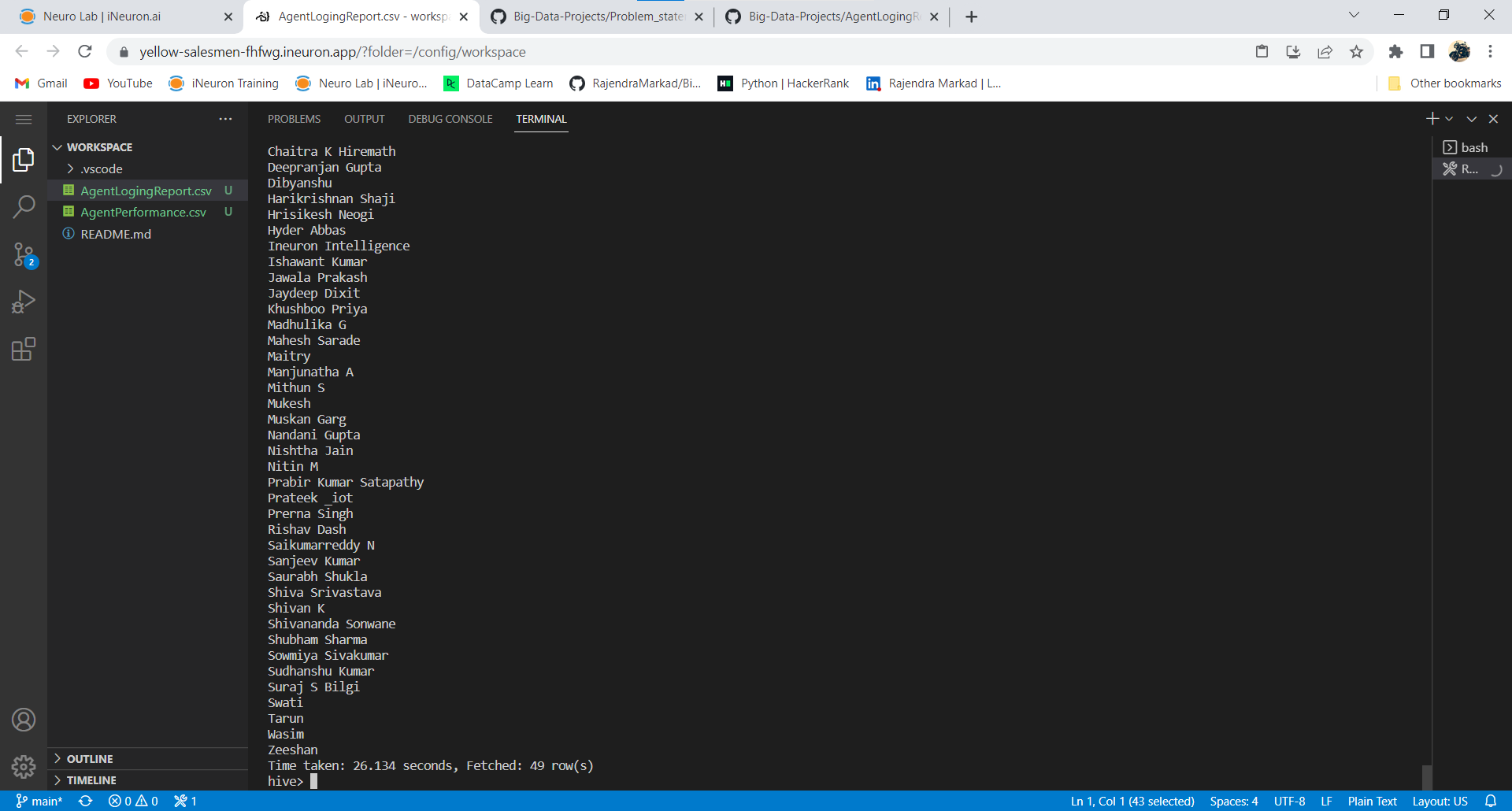
**Load data local inpath '/config/workspace/AgentPerformance.csv' into table AgentPerformance;**



3. List of all agents names.

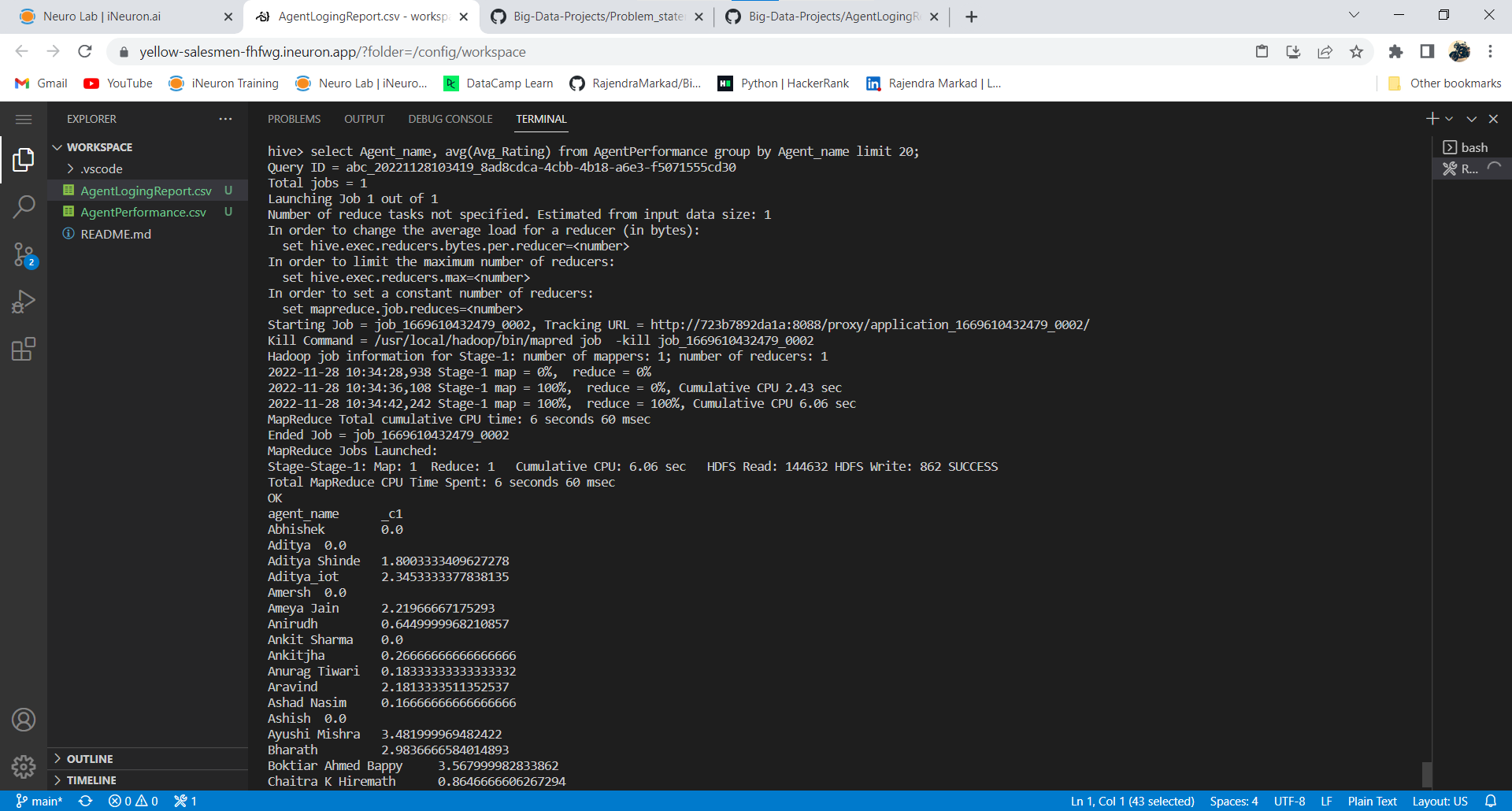
**Hive> select distinct Agent from AgentLogingReport;**





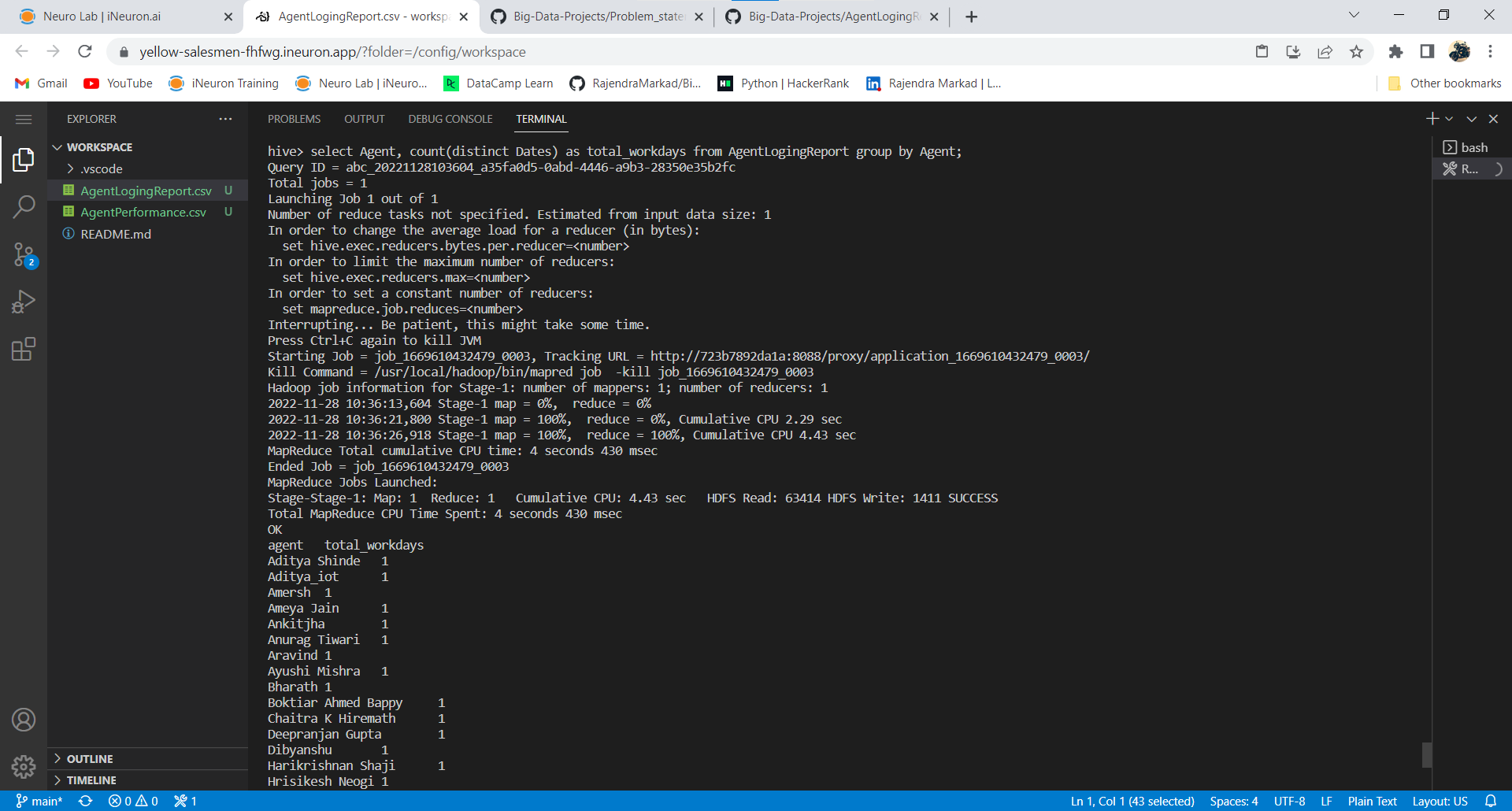
4. Find out agent average rating.

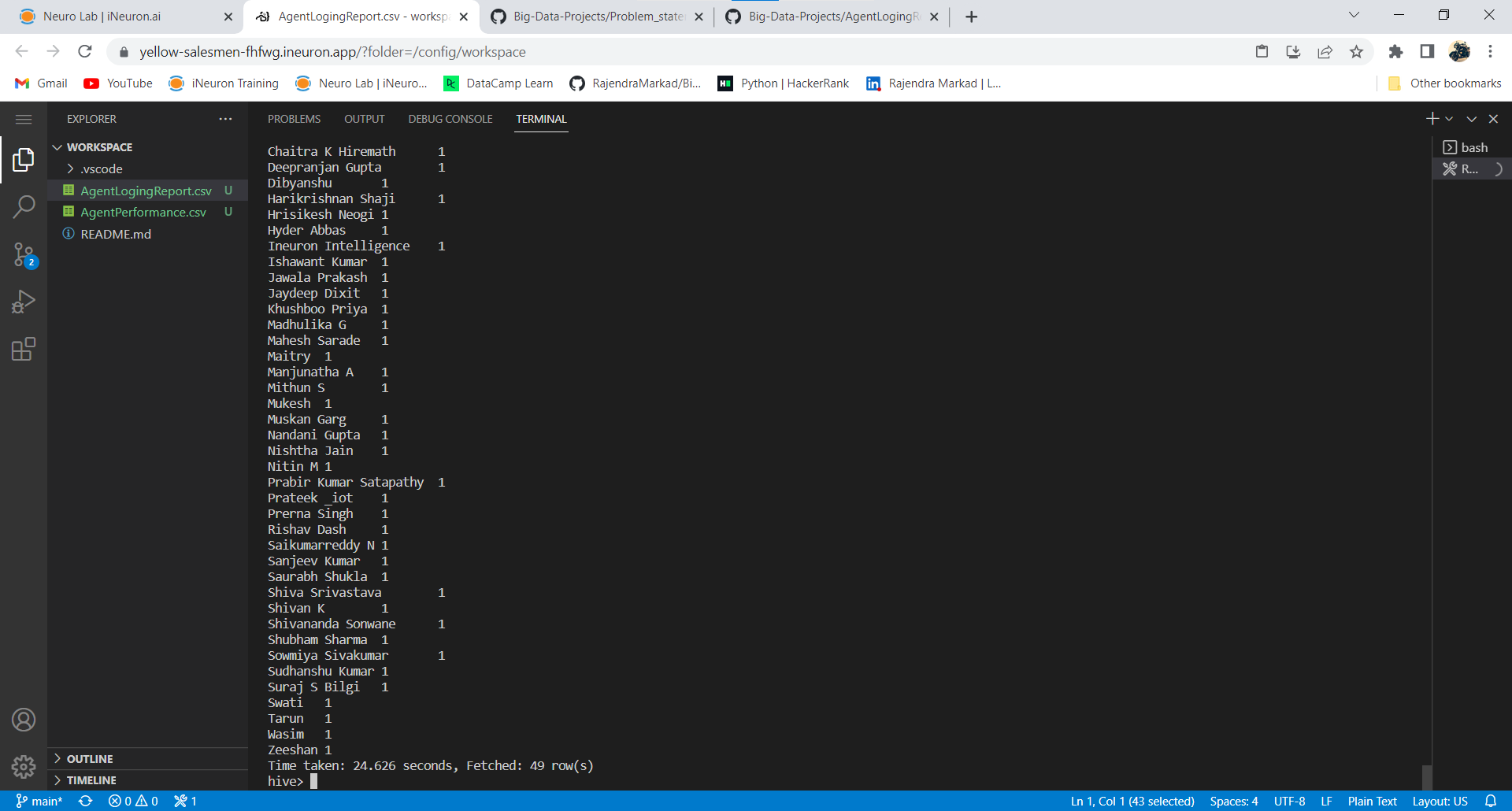
**Hive> select Agent\_name, avg(Avg\_Rating) from AgentPerformance group by Agent\_name;**



5. Total working days for each agents

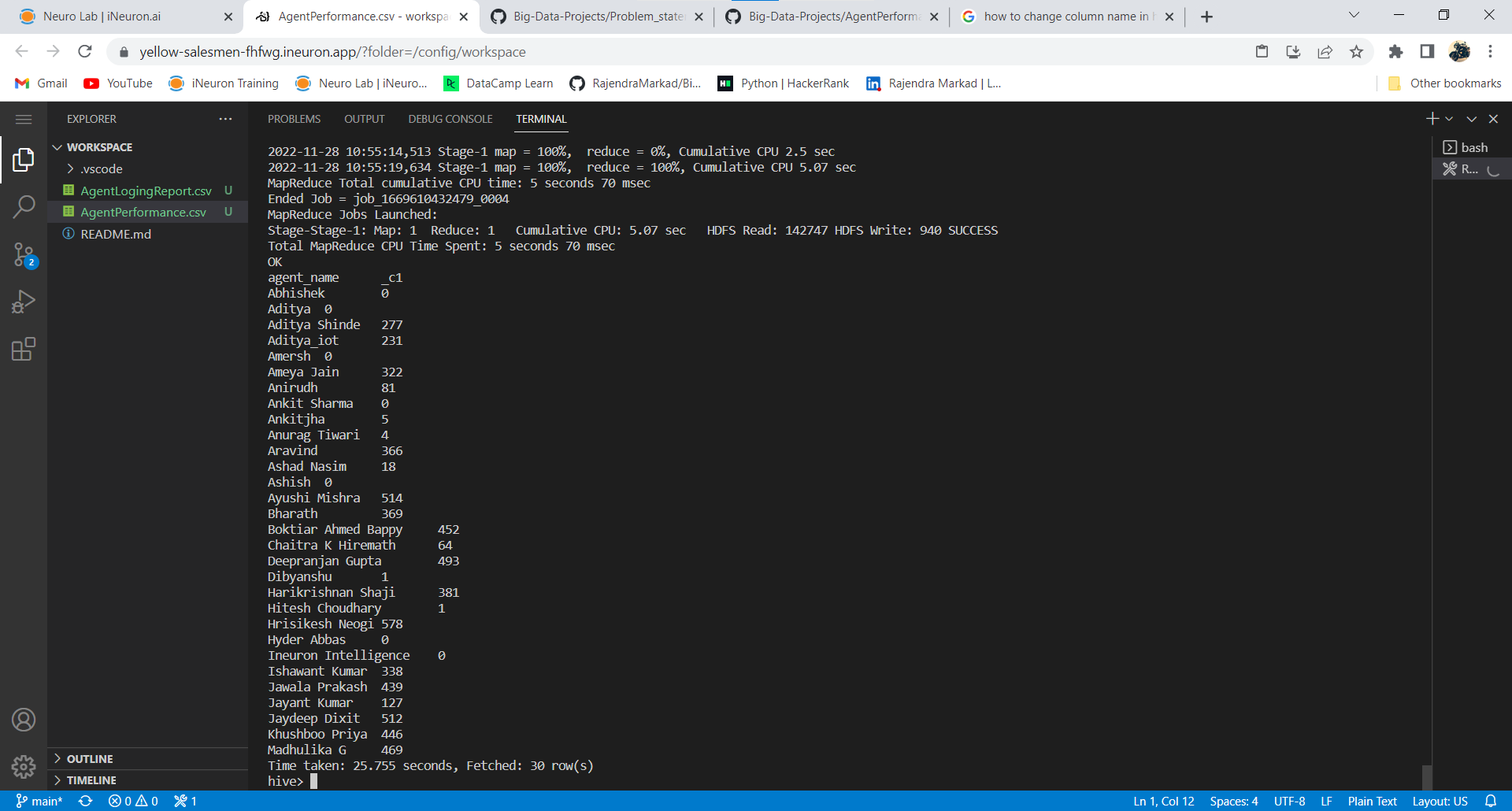
**Hive> select Agent, count(distinct Dates) as total\_workdays from AgentLogingReport group by Agent;**





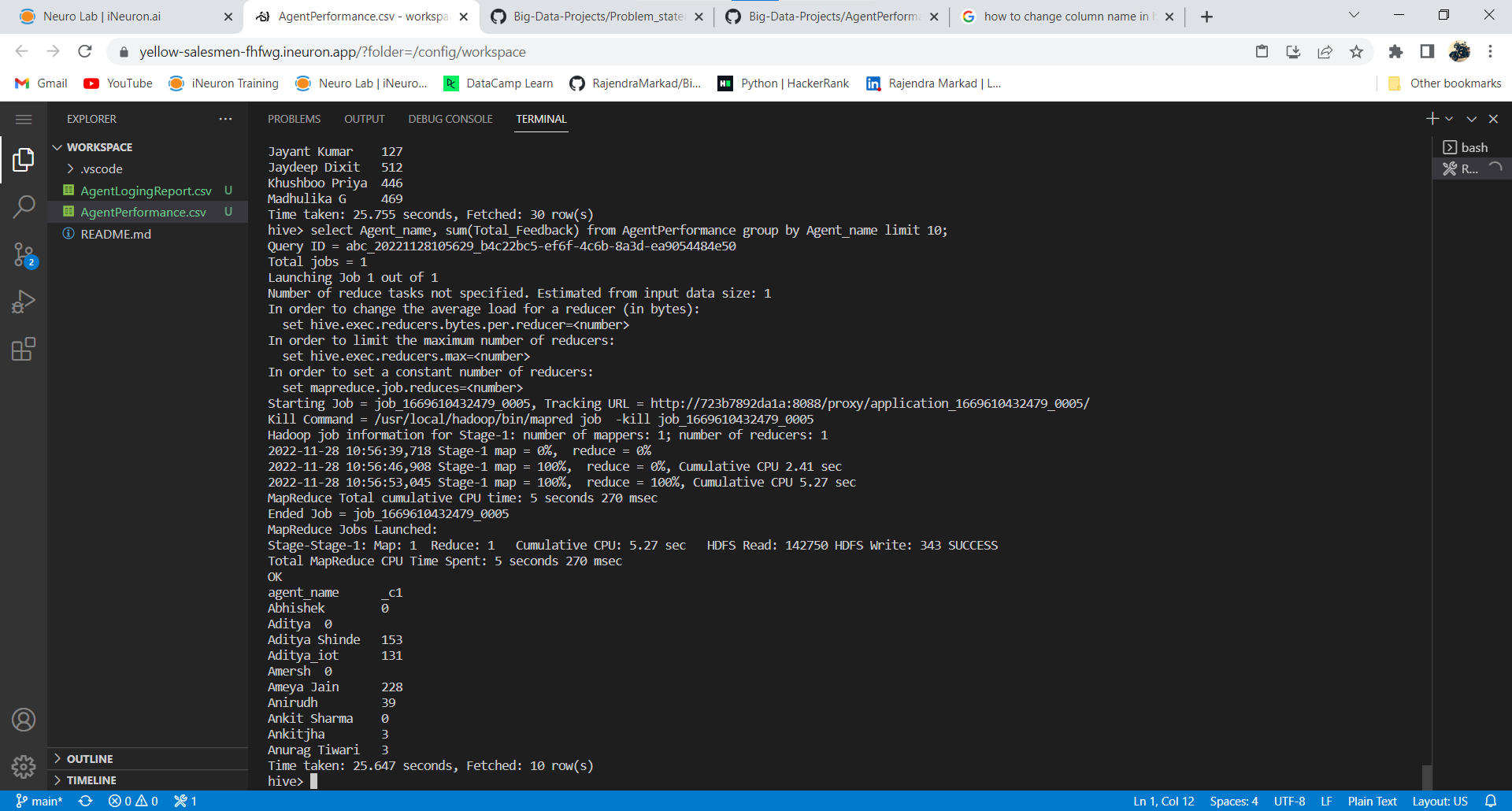
6. Total query that each agent have taken

**Hive> select Agent\_name, sum(total\_chats) from AgentPerformance group by Agent\_name;**

****

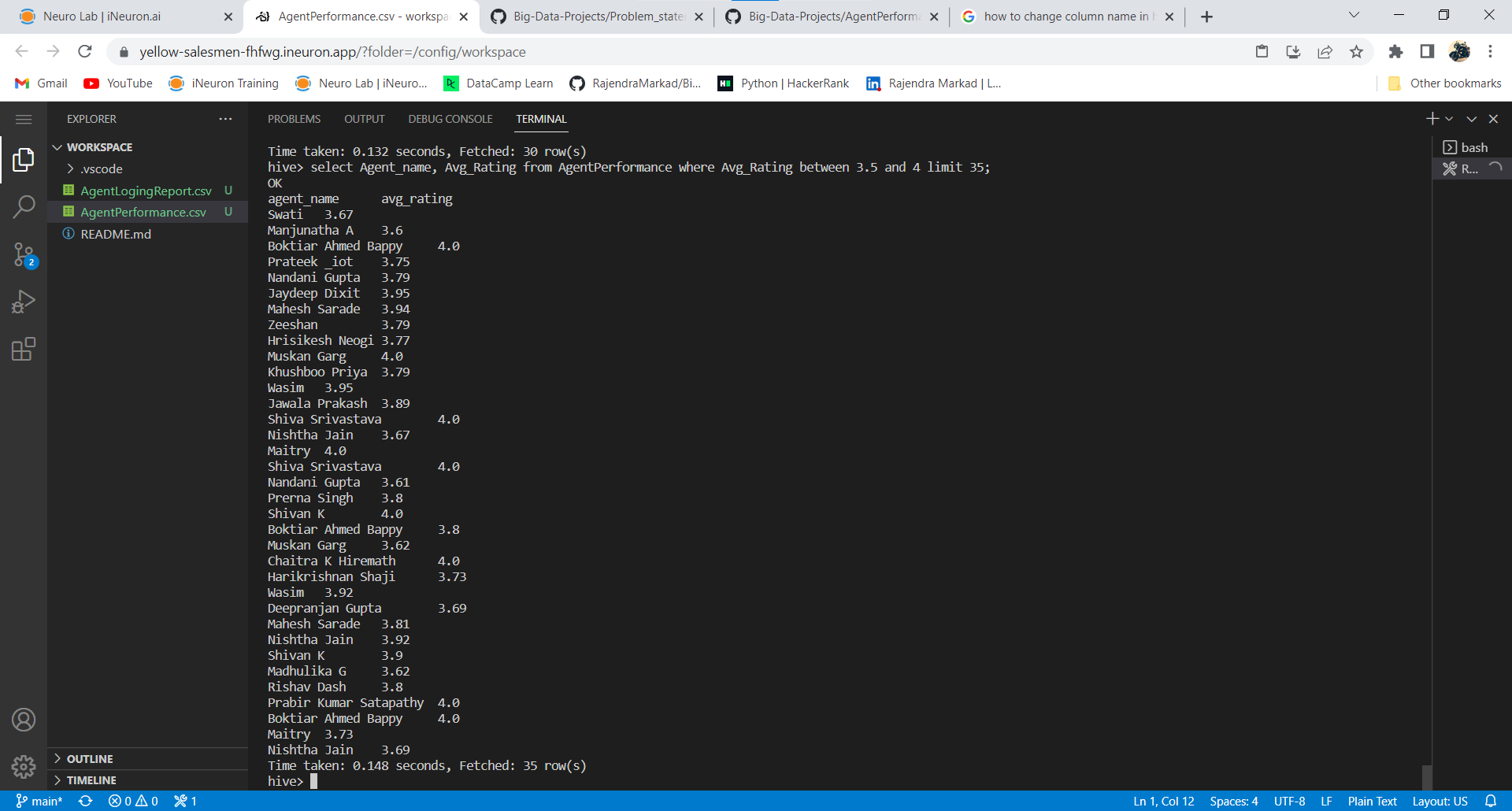
7. Total Feedback that each agent have received

**Hive> select Agent\_name, sum(Total\_Feedback) from AgentPerformance group by Agent\_name;**



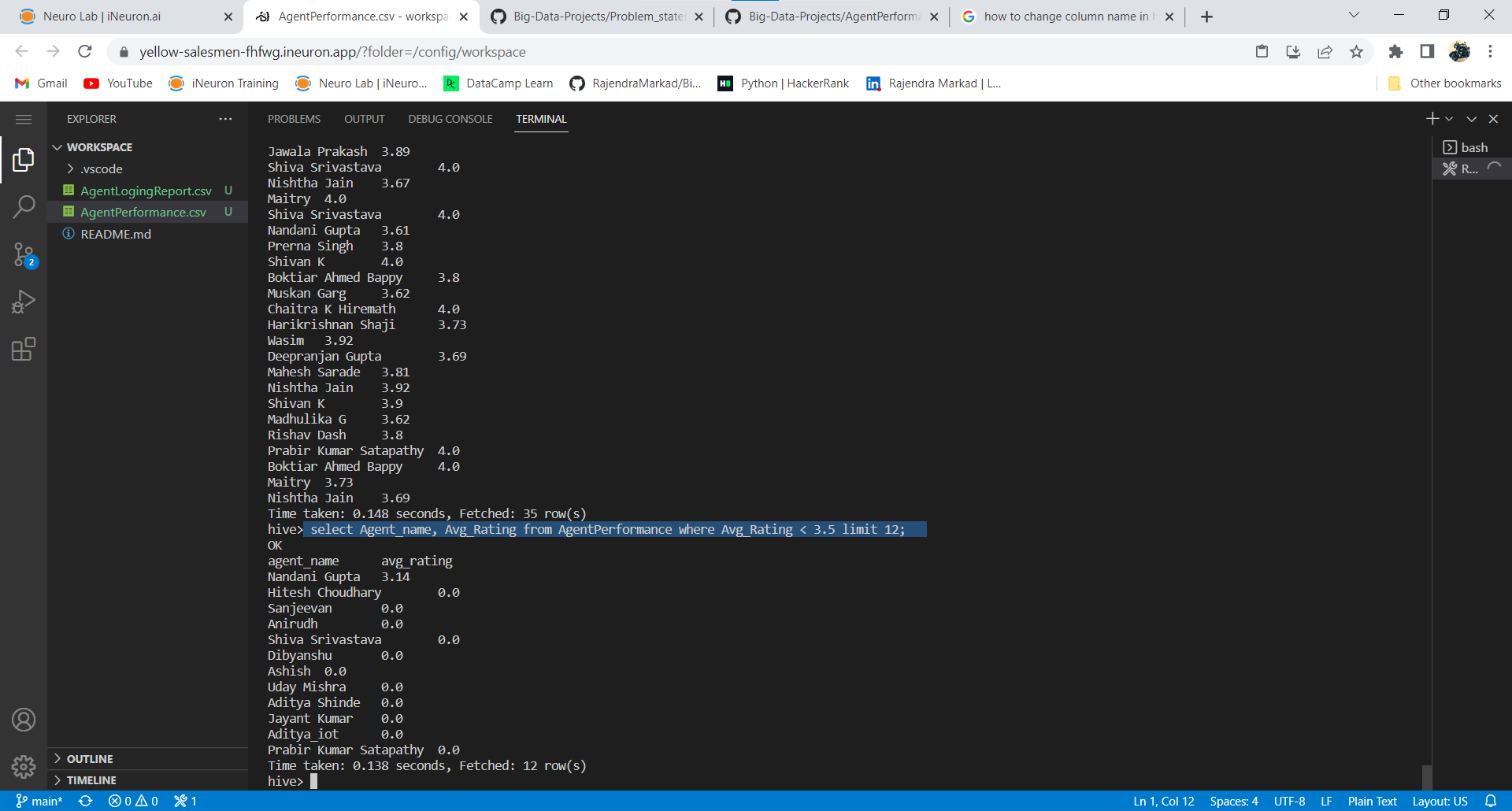
8. Agent name who have average rating between 3.5 to 4

**Hive> select Agent\_name, Avg\_Rating from AgentPerformance where Avg\_Rating between 3.5 and 4;**



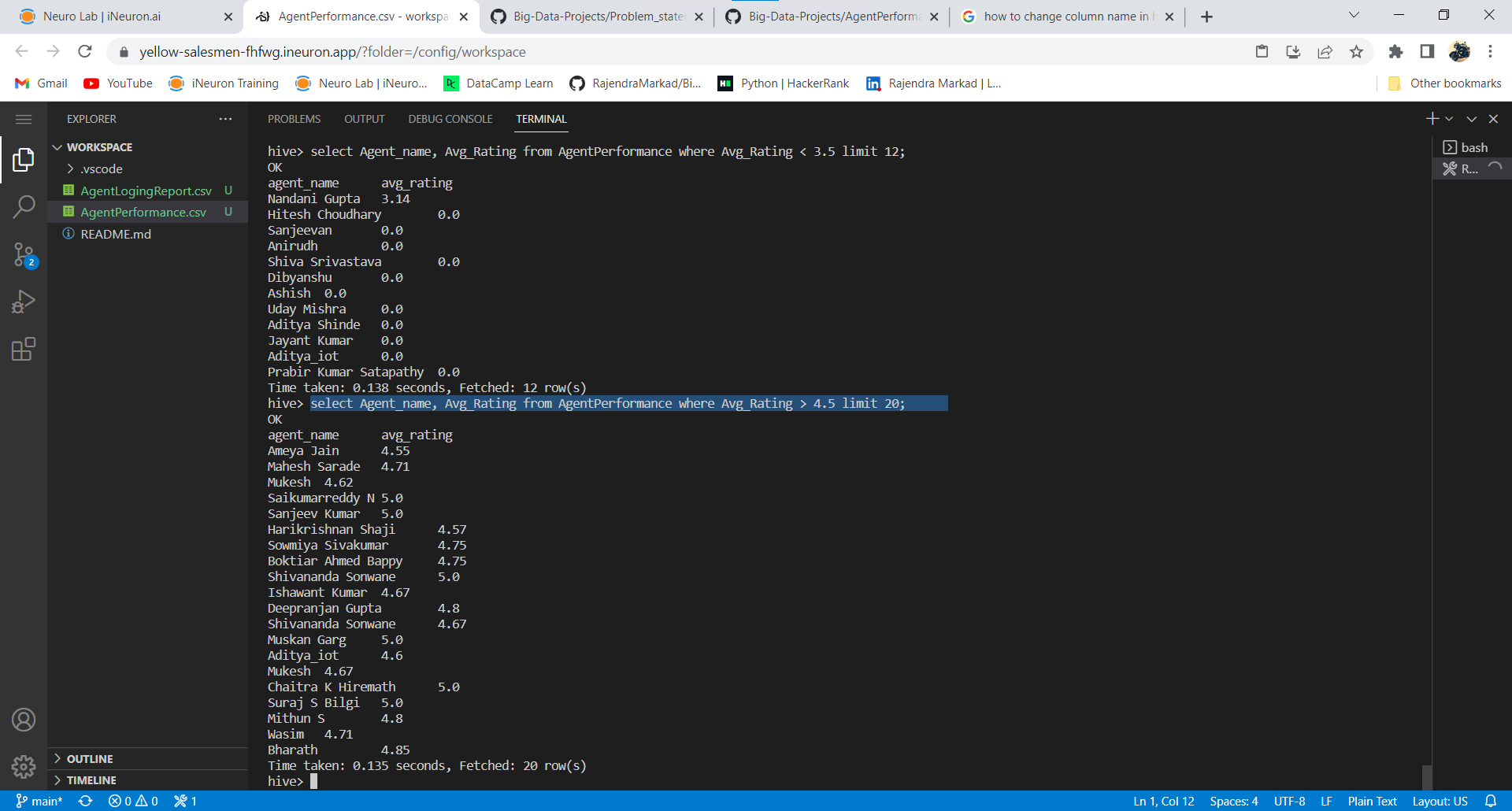
9. Agent name who have rating less than 3.5

**Hive> select Agent\_name, Avg\_Rating from AgentPerformance where Avg\_Rating < 3.5;**



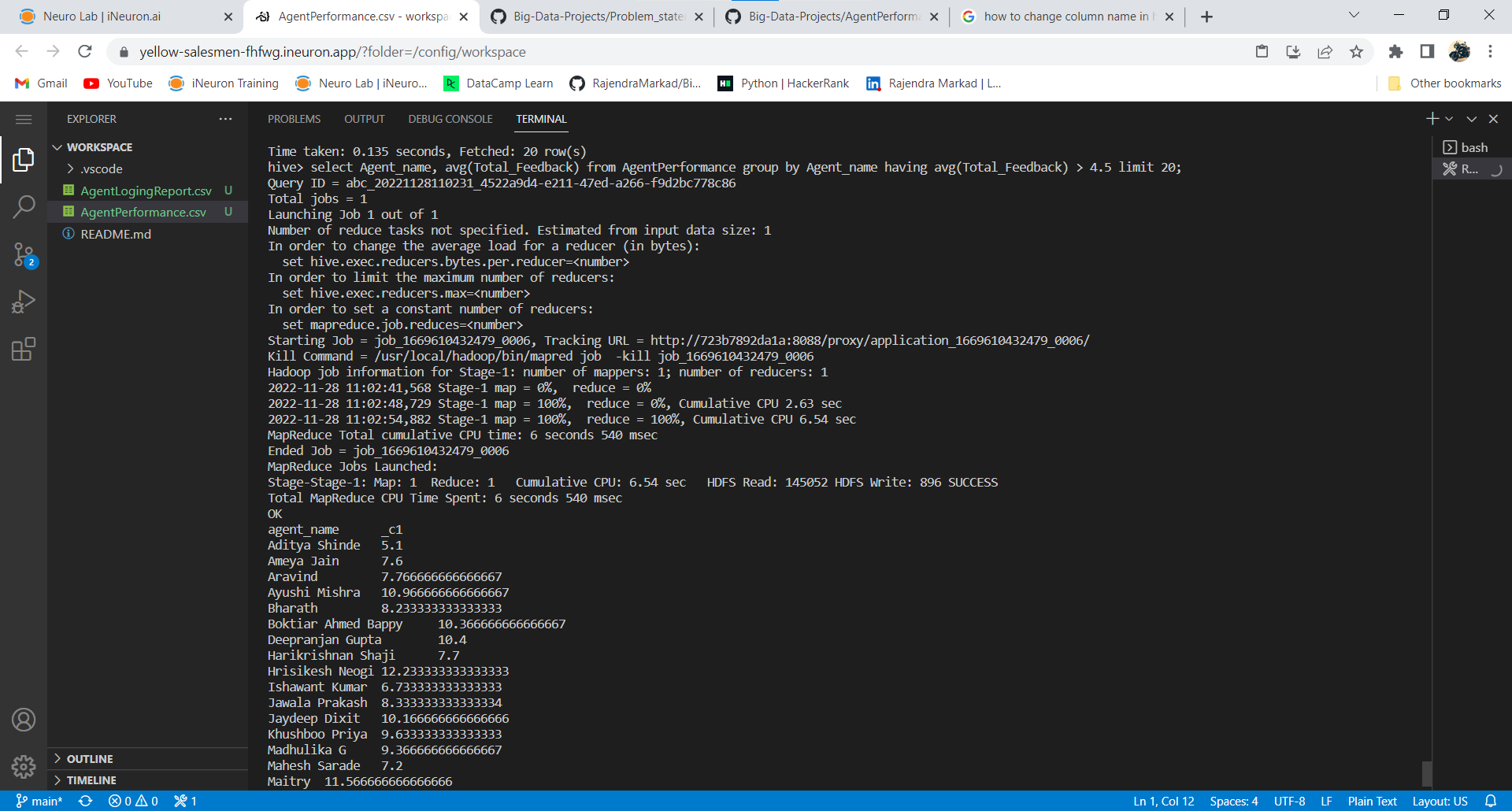
10. Agent name who have rating more than 4.5

**Hive> select Agent\_name, Avg\_Rating from AgentPerformance where Avg\_Rating > 4.5;**



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

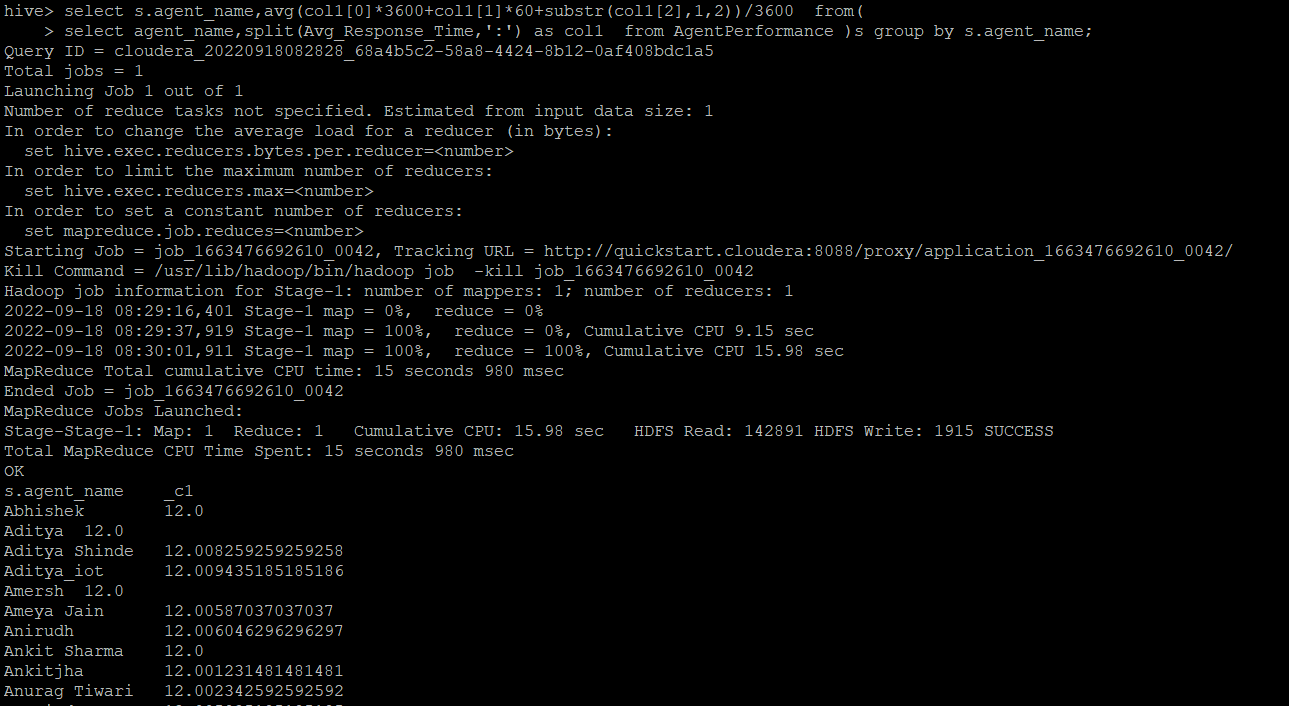
**Hive> select Agent\_name, avg(Total\_Feedback) from AgentPerformance group by Agent\_name having avg(Total\_Feedback) > 4.5;**

****

12. average weekly response time for each agent.

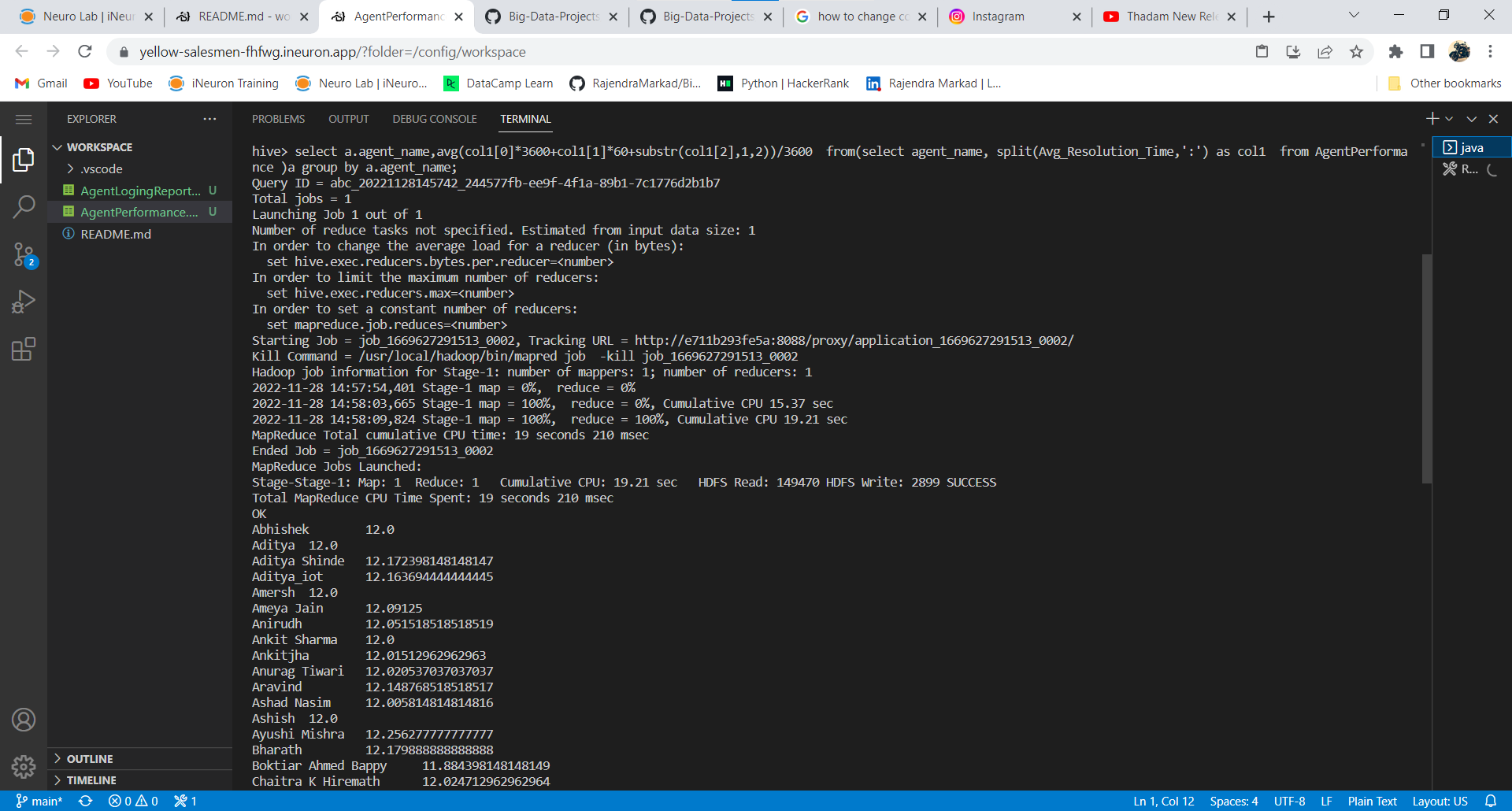
**Hive> select s.agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/3600 from(**

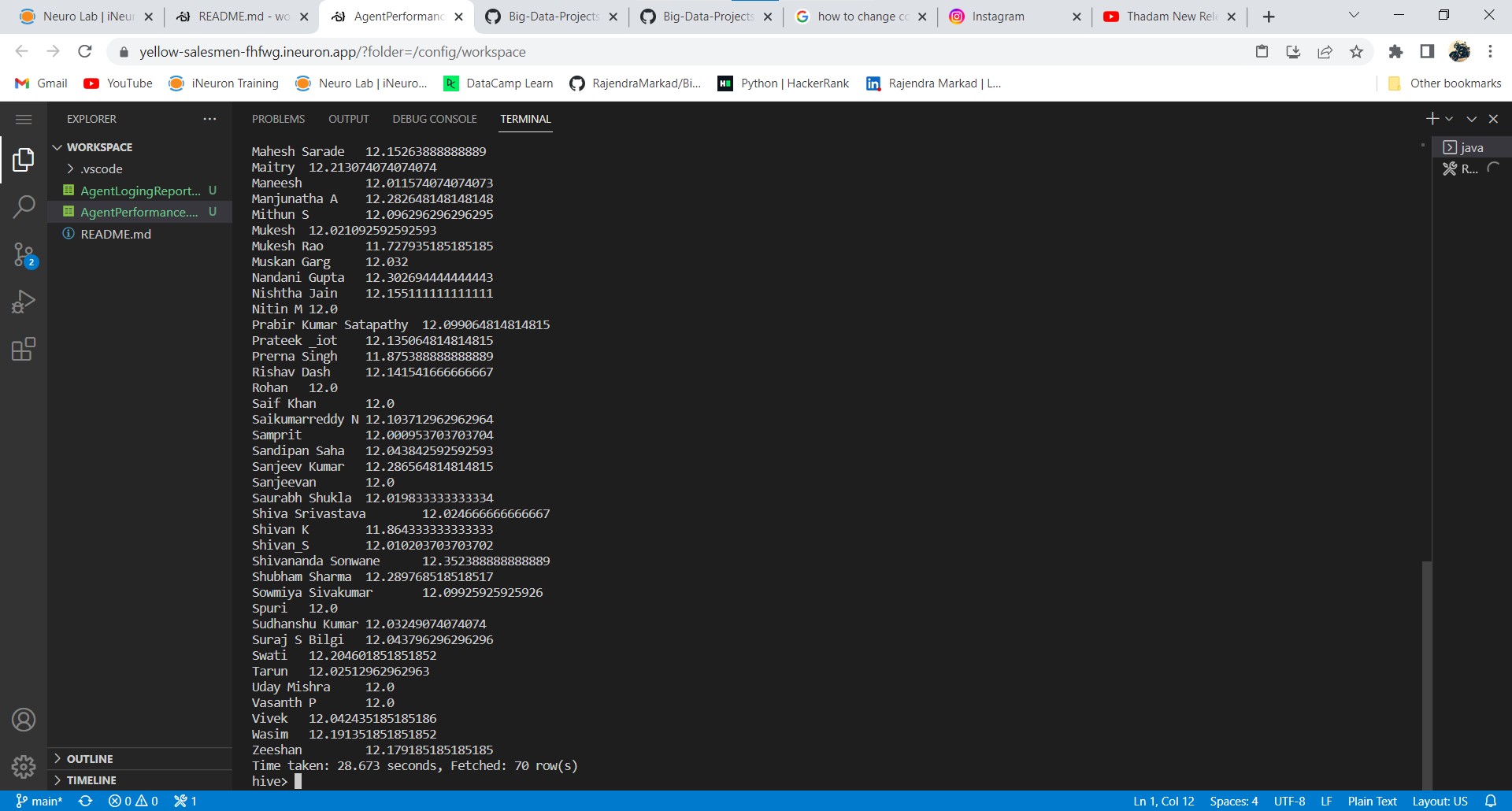
**select agent\_name, split(Avg\_Response\_Time,':') as col1 from AgentPerformance )s group by s.agent\_name;**



13. average weekly resolution time for each agents

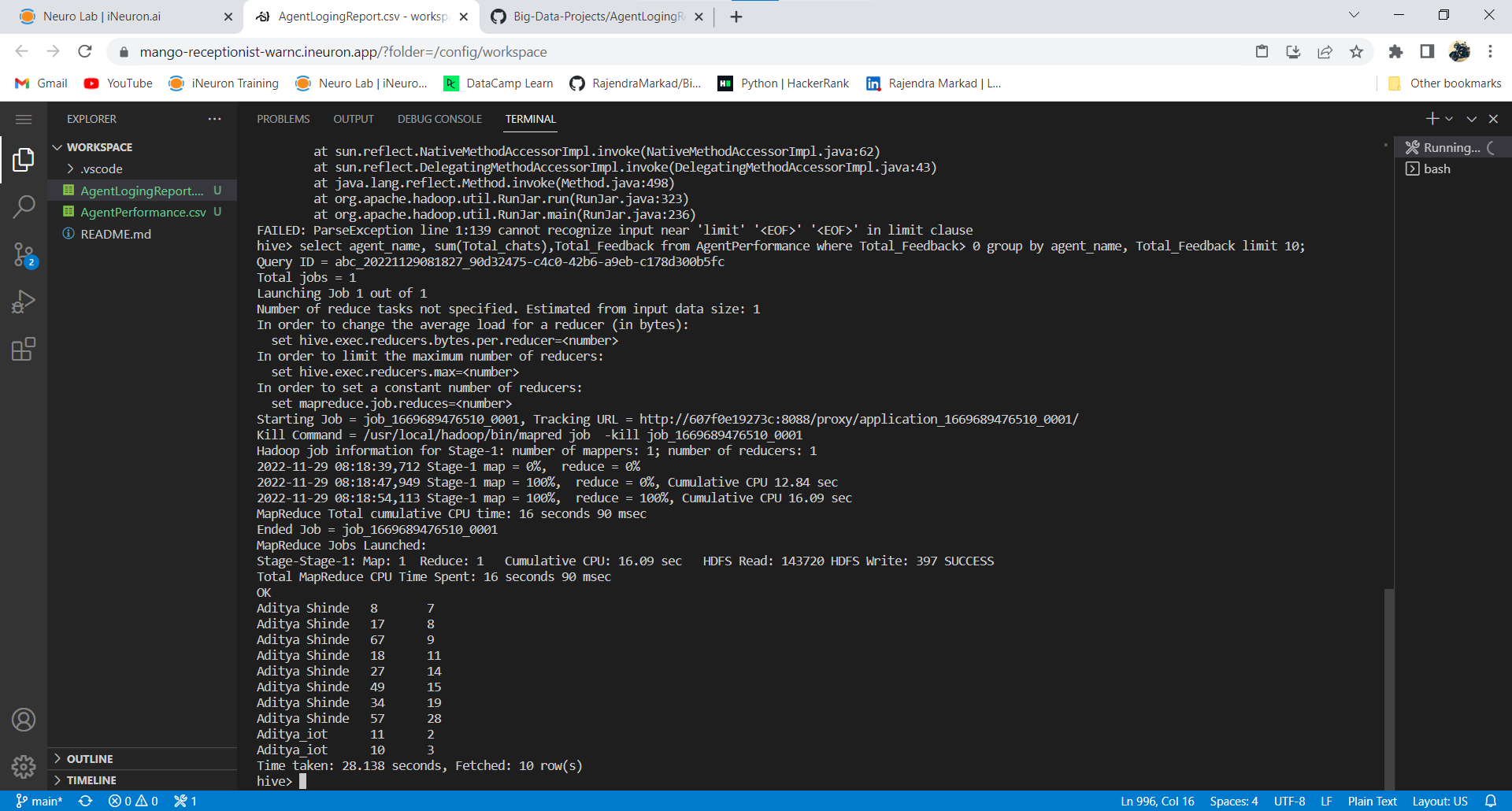
**Hive> select a.agent\_name,avg(col1[0]\*3600+col1[1]\*60+substr(col1[2],1,2))/3600 from(select agent\_name, split(Avg\_Resolution\_Time,':') as col1 from AgentPerformance )a group by a.agent\_name;**





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

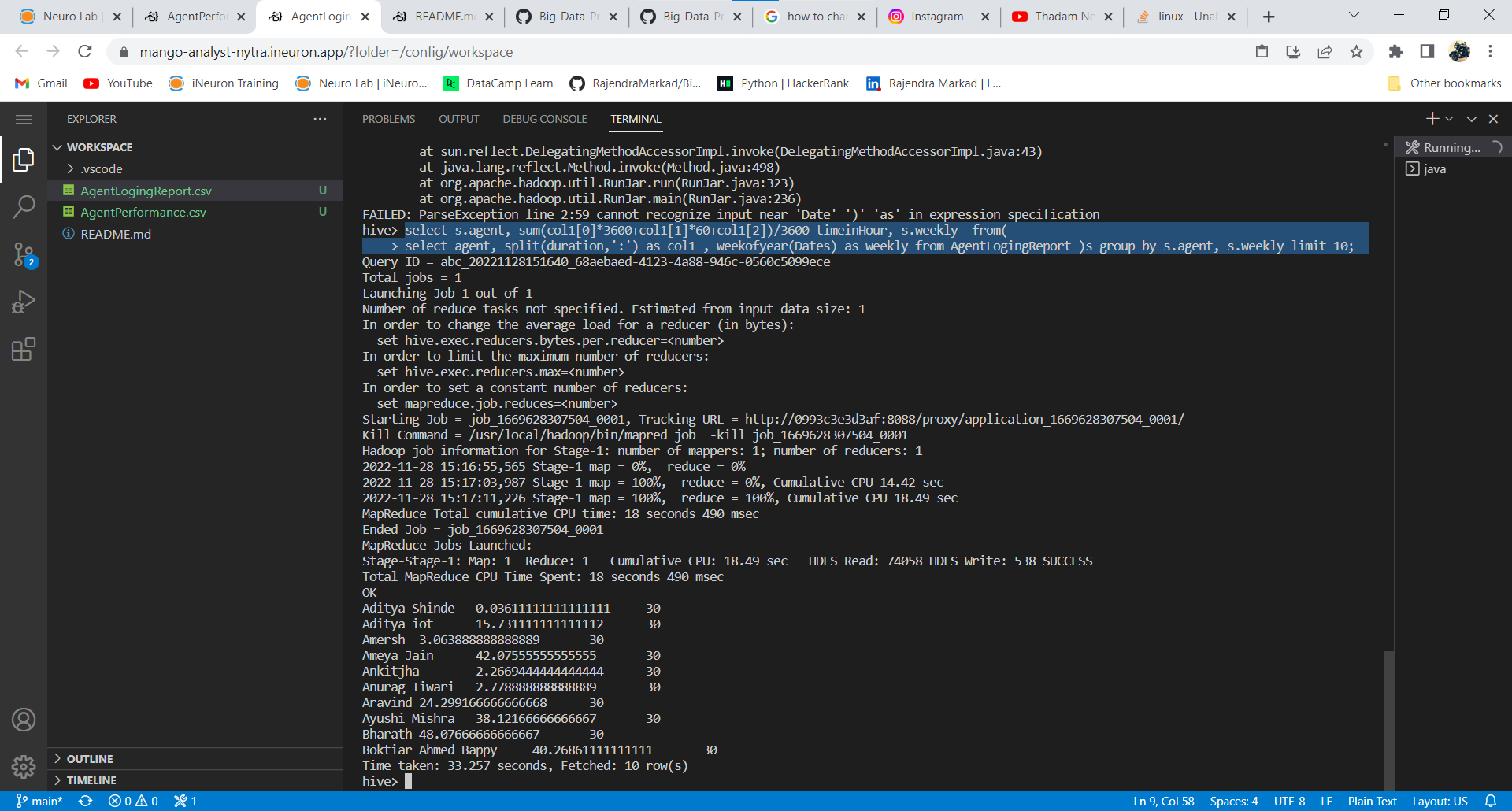
**Hive> select agent\_name, sum(Total\_chats),Total\_Feedback from AgentPerformance where Total\_Feedback> 0 group by agent\_name, Total\_Feedback;**



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

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

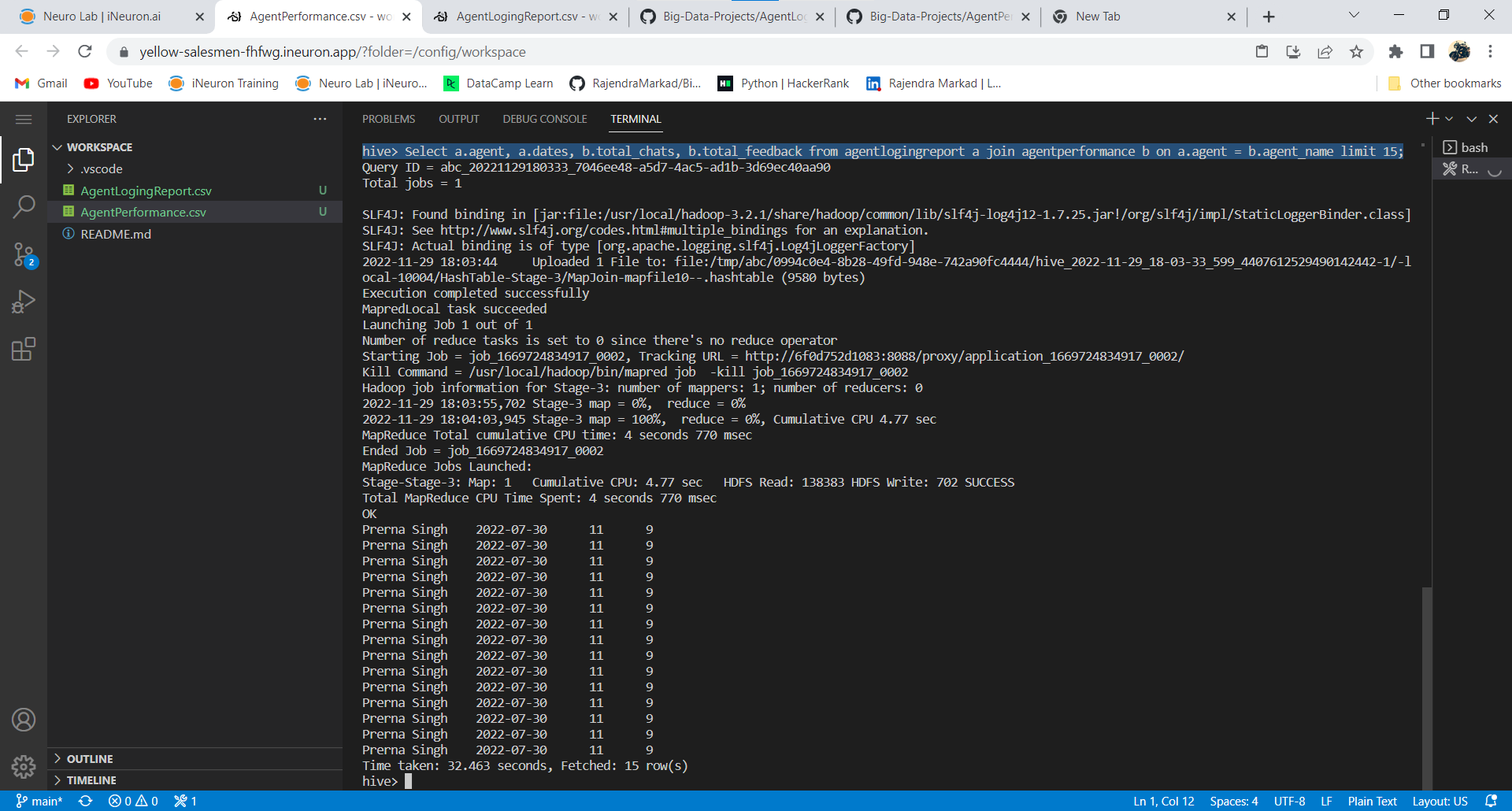
**select agent, split(duration,':') as col1 , weekofyear(Dates) as weekly from AgentLogingReport )s group by s.agent, s.weekly limit 10;**



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.

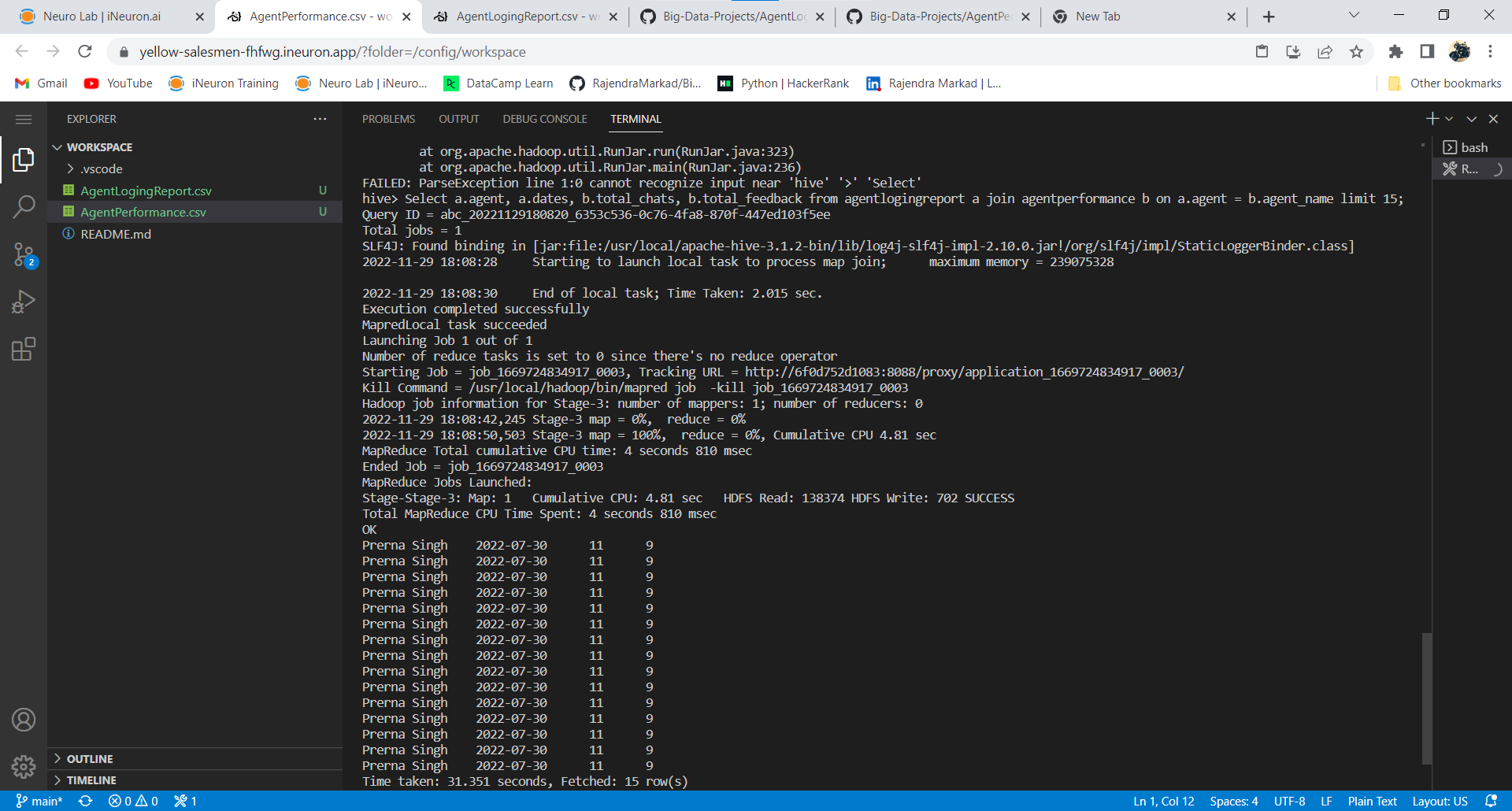
**Inner join:**

**Select a.agent, a.dates, b.total\_chats, b.total\_feedback from agentlogingreport a join agentperformance b on a.agent = b.agent\_name limit 15;**



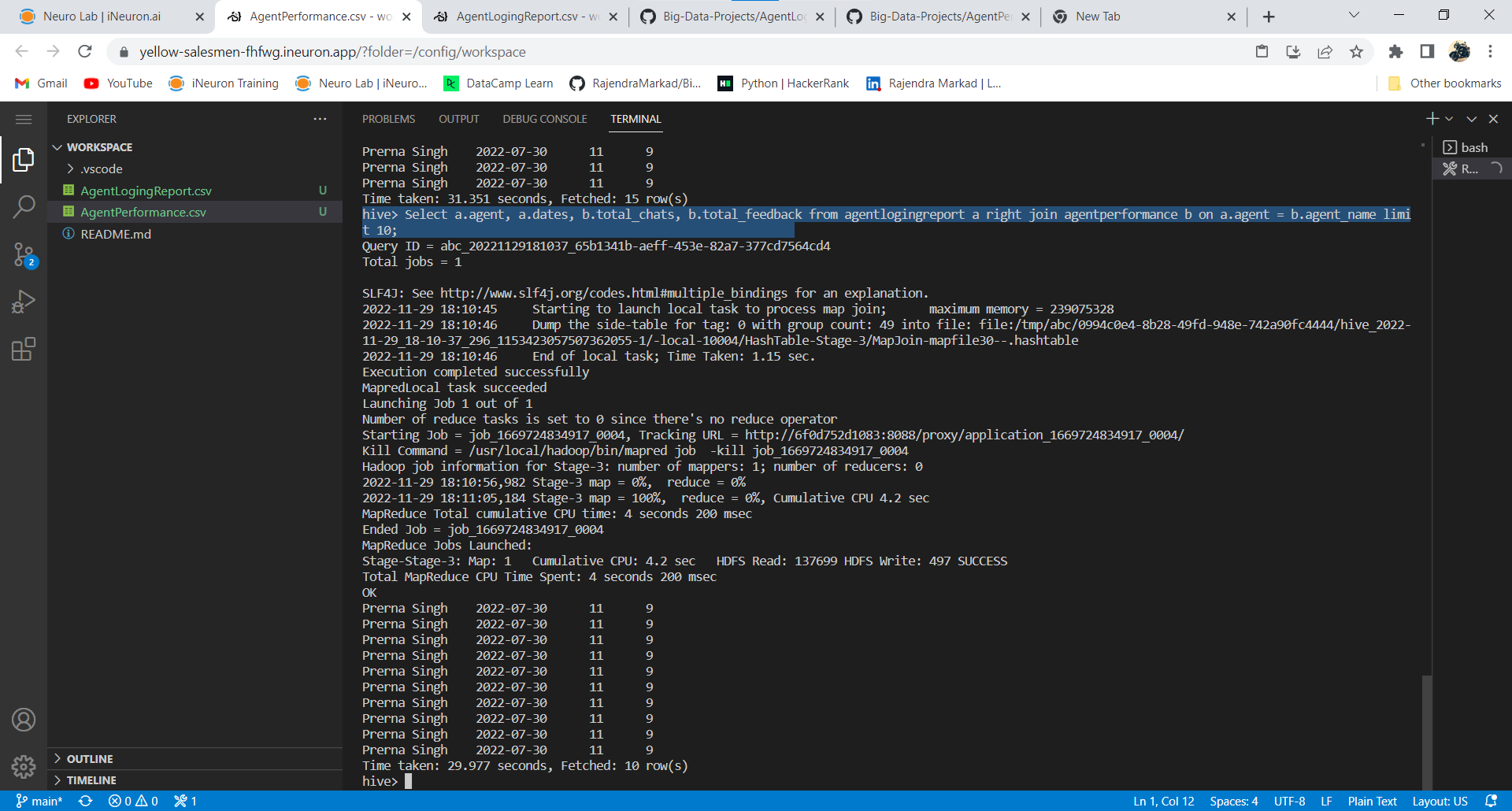
**left join:**

**hive> Select a.agent, a.dates, b.total\_chats, b.total\_feedback from agentlogingreport a join agentperformance b on a.agent = b.agent\_name limit 15;**

****

**Right join:**

**hive> Select a.agent, a.dates, b.total\_chats, b.total\_feedback from agentlogingreport a right join agentperformance b on a.agent = b.agent\_name limit 10;**

****

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

**Create table AgentLogingReport\_partitioned**

**(**

**SL\_no int,**

**Dates date,**

**Login string,**

**Logout string,**

**Duration string**

**)partitioned by (Agent string)**

**CLUSTERED BY (Dates)**

**sorted by (Dates)**

**INTO 4 BUCKETS**

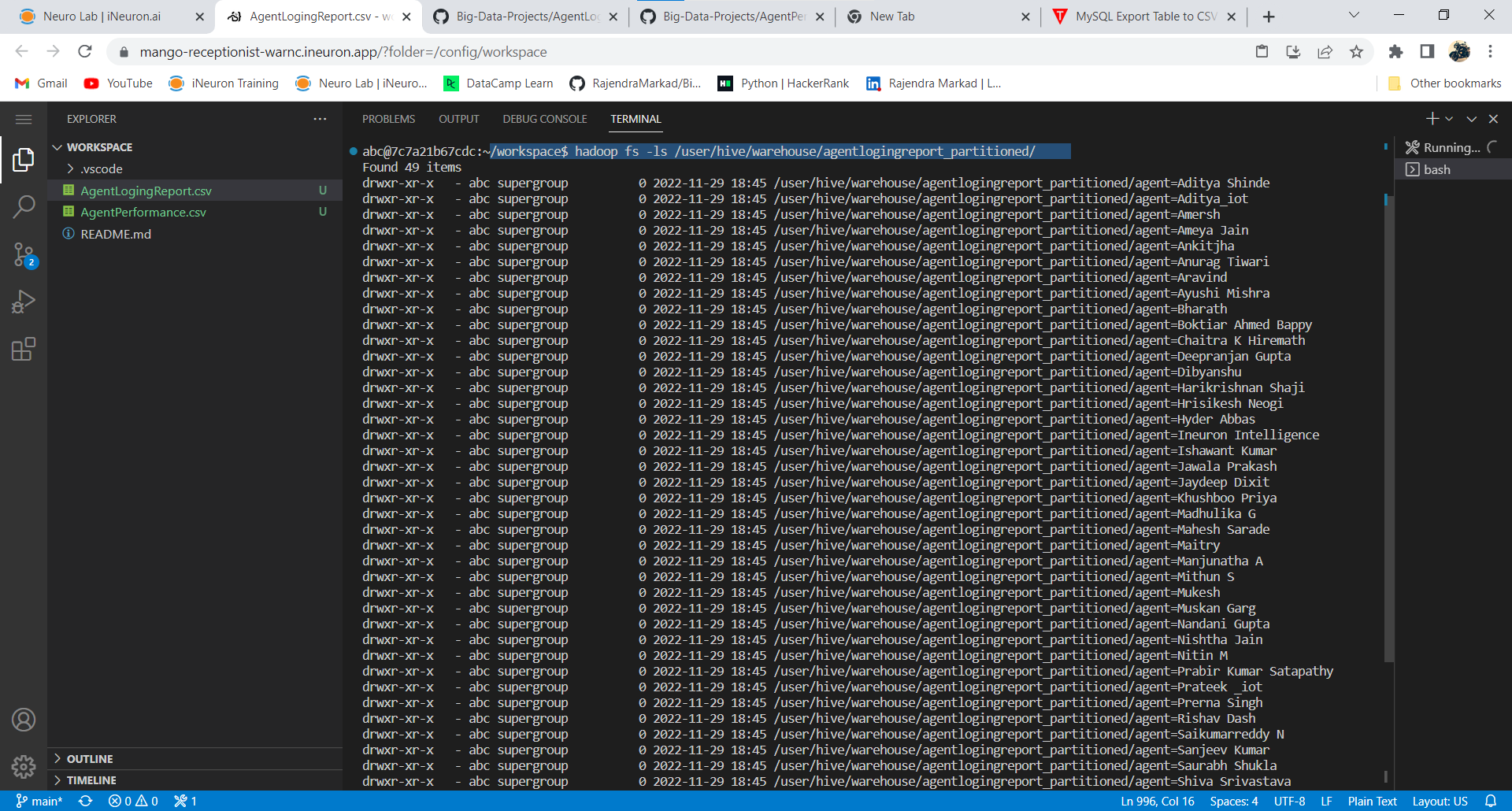
**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY ',';**

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

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

**hive> insert into table AgentLogingReport\_partitioned partition(Agent) select sl\_no,Dates,Login,Logout,Duration,Agent from AgentLogingReport;**

****

**Hive> Create table AgentPerformance\_partitioned**

**(**

**SL\_no int,**

**Dates date,**

**Total\_charts string,**

**Avg\_Response\_Time string,**

**Avg\_Resolution\_Time string,**

**Avg\_Rating float,**

**Total\_Feedback int**

**) partitioned by (Agent\_name string)**

**CLUSTERED BY (Dates)**

**sorted by (Dates) INTO 8 BUCKETS**

**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY ',';**

**Hive> insert into table AgentPerformance\_partitioned partition(Agent\_name) select sl\_no,Dates,Total\_chats,Avg\_Response\_Time,Avg\_Resolution\_Time,Avg\_Rating,Total\_Feedback,Agent\_name from AgentPerformance;**

