This is a real time dataset of the ineuron 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 - M

Note: both files are csv files.

1. Create a schema based on the given dataset

Create table Agentlogreports

>(sl\_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

>(sl\_no int,date string,agent sring,totalchats int,average\_response string,Average\_resolution string,Average\_rating float,total\_feedback int) row formate delimited fields terminated by ‘,’ tblproperties(“skip.header.line.count”=”1”);

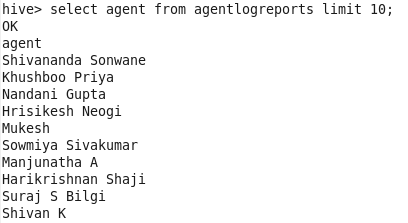
1. Dump the data inside the hdfs in the given schema location.

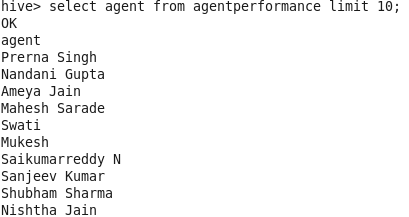
>load data inpath ‘/tmp/Reethesh\_hive\_challenge\_hdfs/AgentlogingReport.csv into table Agentlogreports;

> load data inpath ‘/tmp/Reethesh\_hive\_challenge\_hdfs/AgentPerformance.csv into table AgentPerformance;

1. List of all agents' names.

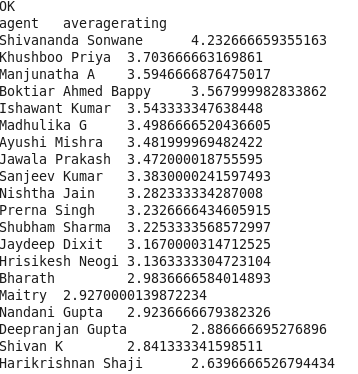
Select Agent from Agentlogreports;





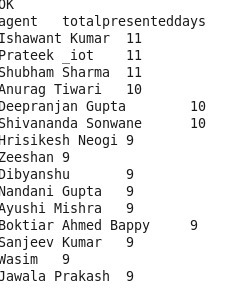
1. Find out agent average rating.

Select agent,avg(average\_rating) as averagerating from agent\_performance group by agent order by averagerating desc;



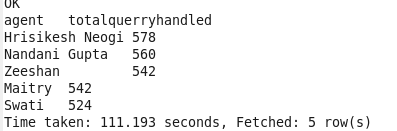
1. Total working days for each agents

Select agent,count(distinct date) as totalpresenteddays from agentlogreports group by agent order by totalpresenteddays desc;

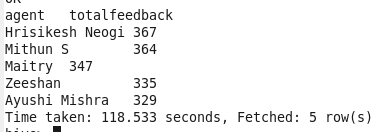


1. Total query that each agent have taken

Select agent,sum(total\_chats)as totalquerryhandled from agentperformance group by agent order by totalquerryhandled desc limit 5;



1. Total Feedback that each agent have received



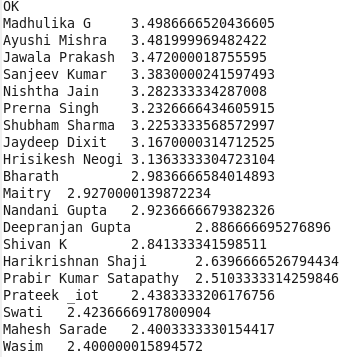
1. Agent name who have average rating between 3.5 to 4

Select agent,avg(average\_rating) as averagerating from agentperformance where average\_rating between 3.5 and 4 group by agent order by averagerating desc;



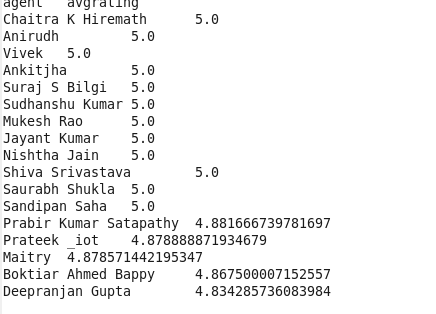
1. Agent name who have rating less than 3.5

Select agent,avg(average\_rating) as avgrating from agentperformance group by agent having avg(average\_rating)<3.5 order by avgrating desc;



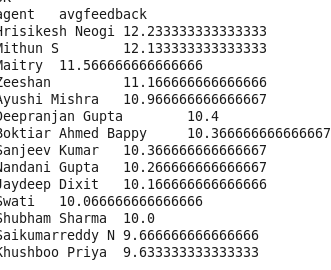
1. Agent name who have rating more than 4.5

Select agent,avg(average\_rating) as avgrating from agentperformance where average\_rating >3.5 group by agent order by avgrating ;



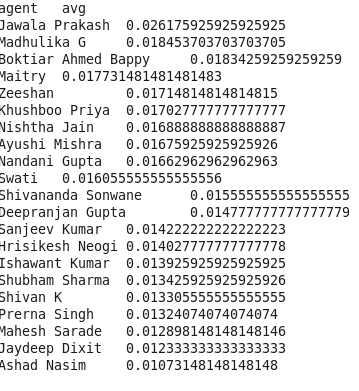
1. How many feedback agents have received more than 4.5 average

Select agent,avg(total\_feedback) as avgfeedback from agentperformance group by agent having avg(total\_feedback)>4.5 order by avgfeedback desc;



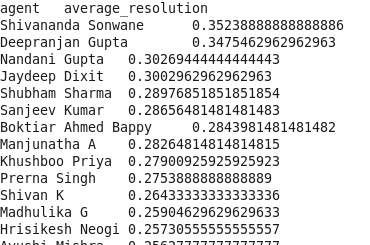
1. average weekly response time for each agent

Selectagent,avg(average\_response[0]\*3600+average\_response[1]\*60+substr(average\_response[2],1,2))/3600 as avg from (select agent,split(average\_response,’:’) as average\_response from agentperformance)s group by agent order by avg desc;



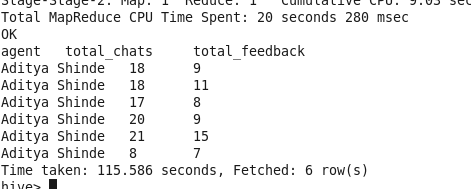
1. average weekly resolution time for each agents

Selectagent,avg(average\_resolution[0]\*3600+average\_resolution[1]\*60+substr(average\_resolution[2],1,2))/3600 as average\_resolution from (select agent,split(average\_resolution,’:’) as average\_resolution from agentperformance)s group by agent order by average\_resolution desc;



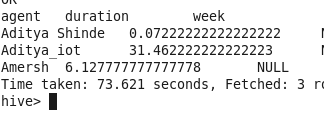
1. Find the number of chat on which they have received a feedback

Select agent,sum(total\_chats) as total\_chats from agentperformance where total\_feedback!=0 group by agent,total\_chats,total\_feedback;



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

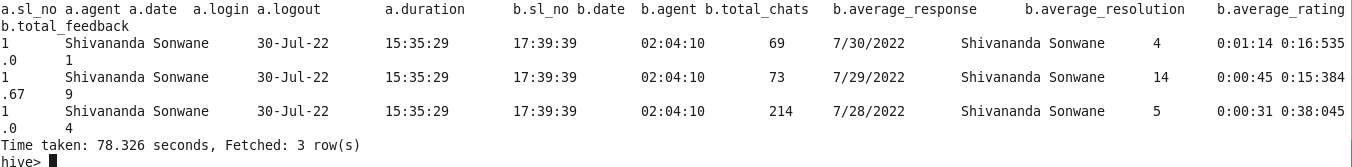
Select agent,sum(r1[0]\*3600+r1[1]\*60+r1[2])/3600 as duration,week from select(agent,split(duration,’:’) as r1,WeekOfYear(date) as week from agentlogreports)s group by agent,duration limit3;



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**

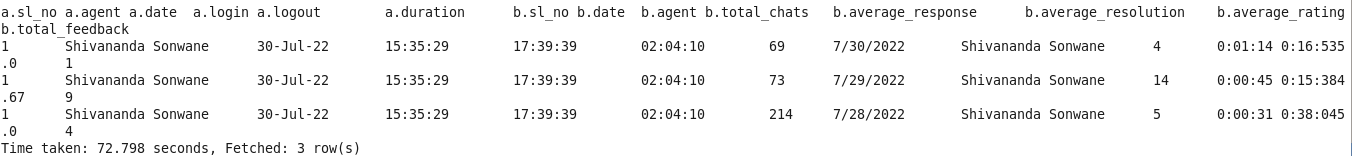
Select \* from agentlogreports as a inner join agentperformance as b on a.agent=b.agent limit 3;



**Left join**

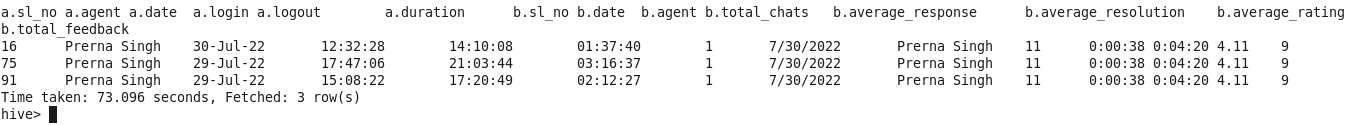
Select \* from agentlogreports as a left

join agentperformance as b on a.agent=b.agent limit 3;



**Right join**

Select \* from agentlogreports as a right join agentperformance as b on a.agent=b.agent limit 3;



**Ecport data into local as csv file (file path:/tmp/Reethesh\_revision)**

**Cloudera terminal:**

Hive –e ‘select \* from reethesh\_mini\_project.agentlogreports as a inner join agentperformance as b on a.agent = b. agent’ into out file /tmp/Reethesh\_revision/inner\_join.csv

Hive –e ‘select \* from reethesh\_mini\_project.agentlogreports as a right join agentperformance as b on a.agent = b. agent’ into out file /tmp/Reethesh\_revision/right\_join.csv

Hive –e ‘select \* from reethesh\_mini\_project.agentlogreports as a left join agentperformance as b on a.agent = b. agent’ into out file /tmp/Reethesh\_revision/left\_join.csv



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

Create table AgentLogingReport\_partitioned

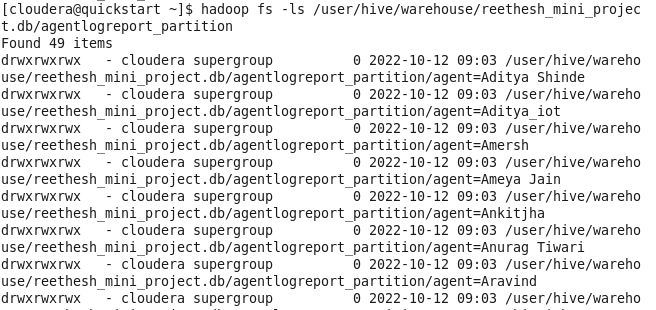
(sl\_no int,Date date,Login string,Logout string,Duration string)partitioned by (Agent string)Clustered by (Date) sorted by (date) into 4 BUCKETS

Row format delimited fields terminated by ',';

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

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

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



Hive> Create table AgentPerformance\_\_partition

(sl\_no int,Date date,total\_chats string,Average\_response string,

Average\_Resolution string,Average\_Rating float,total\_Feedback int)partitioned by (Agent string)

Clustered by (Date) sorted by (Date) into 8 BUCKETS

Row format delimited fields terminated by ',';

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

