Hive challenge -1

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 - 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**

create table agent (

id int, agent string, date string, login\_time string, logout\_time string, duration string)

row format delimited

fields terminated by ','

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

create table performance (id int, date string, agentname string, total\_chats int, averange\_response\_time string, averange\_resolution\_time string, average\_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.**

hadoop fs -copyFromLocal 'agent.csv' '/user/hive/warehouse/agent/'

hadoop fs -copyFromLocal 'performance.csv' '/user/hive/warehouse/performace/'

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

select distinct(agent) as agent\_names from agent

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

select avg(average\_rating)as average\_rating,agent\_name from performance group by agent limit 5;

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

select agent, count(date) as working\_days from agent group by agent;

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

select sum(total\_chats),agent\_name from performance group by agent\_name limit 5;

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

select agentname, sum(total\_feedback) as total\_feedbacks from performance where total\_feedback > 0 group by agentname;

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

select agentname,average\_rating from performance where average\_rating between '3.5' and '4';

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

select agentname,average\_rating from performance where average\_rating < '3.5';

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

select agentname,average\_rating from performance where average\_rating > '4.5';

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

select agentname,count(total\_feedback) as total\_feedback from performance where average\_rating > '4.5' group by agentname;

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 agentname,sum(total\_chats) as from performance where total\_feedback > 0 group by agent\_name, total\_feedback;

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.**

**Set hive.auto.convert.join=true;**

Insert overwrite local directory '/tmp/results1.csv' row format delimited fields terminated by ‘,’ select \* from agent a inner join performance p where a.agent=p.agentname ;

Insert overwrite local directory '/tmp/results2.csv' row format delimited fields terminated by ‘,’ select \* from agent a right join performance p where a.agent=p.agentname ;

Insert overwrite local directory '/tmp/results3.csv' row format delimited fields terminated by ‘,’ select \* from agent a left join performance p where a.agent=p.agentname ;

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

create table performance\_bucketing (id int, date string, total\_chats int, averange\_response\_time string, averange\_resolution\_time string, average\_rating float, total\_feedback int)partitioned by (agentname string) clustered by (id) sorted by (id) into 7 buckets

stored as ORC;

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

SET hive.enforce.bucketing=true;

insert overwrite table performance\_bucketing partition (agentname) select id, date, total\_chats, averange\_response\_time, averange\_resolution\_time , average\_rating, total\_feedback, agentname from performance;

**#how\_to\_decide\_bucketing\_number**

#https://stackoverflow.com/questions/30730567/how-can-we-decide-the-total-no-of-buckets-for-a-hive-table

#https://www.youtube.com/watch?v=pHvyhfWK43s&ab\_channel=DataEngineering

Lets take a scenario Where table size is: 2300 MB, HDFS Block Size: 128 MB

Now, Divide 2300/128=17.96

Now, remember number of bucket will always be in the power of 2.

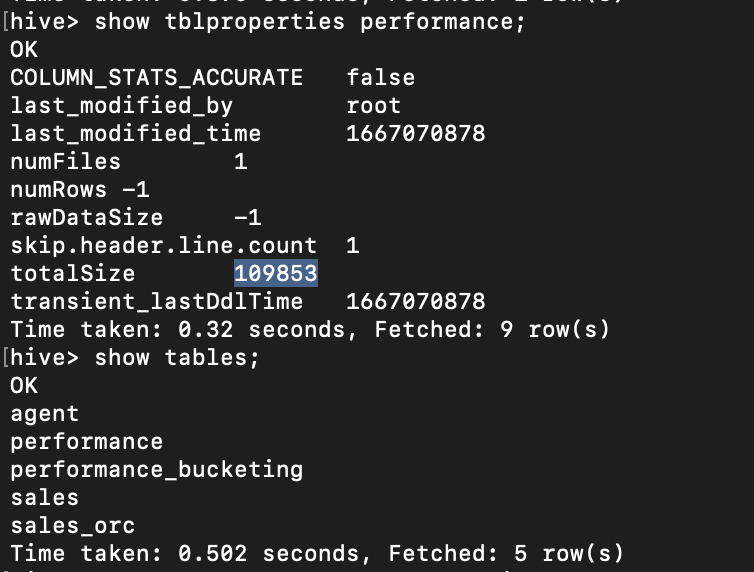
So we need to find n such that 2^n > 17.96

n=5.

So, I am going to use number of buckets as 2^5=32

**#from the above explanation we have created we have assigned a bucket number**

**Show tblproperties performance #to\_see\_table\_size**

****

Now, Divide 109853/128(HDFS block size)=858.22

Log(858)= 2.93

2^n => 2^2.93 = 7.6. so I took around 7 as bucket number