1. Create a schema based on the given dataset

create table agent\_performance\_bkp

(

sl\_no int,

ag\_date string,

agent\_name string,

total\_chats int,

average\_response\_time string,

average\_resolution\_time string,

average\_rating float,

total\_feedback int)

row format delimited

fields terminated by ','

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

create table agent\_performance

(

sl\_no int,

ag\_date date,

agent\_name string,

total\_chats int,

average\_response\_time bigint,

average\_resolution\_time bigint,

average\_rating float,

total\_feedback int)

row format delimited

fields terminated by ',';

create table agent\_loging\_report

(

sl\_no int,

agent\_name string,

ag\_date date,

login\_time bigint,

logout\_time bigint,

duration bigint)

row format delimited

fields terminated by ',';

create table agent\_loging\_report\_bkp

(

sl\_no int,

agent\_name string,

ag\_date string,

login\_time string,

logout\_time string,

duration string)

row format delimited

fields terminated by ','

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

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

load data local inpath '/Users/thestupidmonk/Downloads/AgentPerformance.csv' into table agent\_performance\_bkp;

insert into table agent\_performance select sl\_no,

from\_unixtime(unix\_timestamp(ag\_date,'mm/dd/yyyy'), 'yyyy-mm-dd'), agent\_name, total\_chats,

case when average\_response\_time like '%:%:%' then unix\_timestamp(average\_response\_time,'HH:mm:ss')

when average\_response\_time like '%:%' then unix\_timestamp(average\_response\_time,'mm:ss')

else average\_response\_time

end as average\_response\_time,

case when average\_resolution\_time like '%:%:%' then unix\_timestamp(average\_resolution\_time,'HH:mm:ss')

when average\_resolution\_time like '%:%' then unix\_timestamp(average\_resolution\_time,'mm:ss')

else average\_resolution\_time

end as average\_resolution\_time,

average\_rating, total\_feedback

from agent\_performance\_bkp;

load data local inpath '/Users/thestupidmonk/Downloads/AgentLogingReport.csv' into table agent\_loging\_report\_bkp;

insert into table agent\_loging\_report select sl\_no, agent\_name,

from\_unixtime(unix\_timestamp(substr(ag\_date,0,11),'dd-MMM-yy')),

case when login\_time like '%:%:%' then unix\_timestamp(login\_time,'HH:mm:ss')

when login\_time like '%:%' then unix\_timestamp(login\_time,'mm:ss')

else login\_time

end as login\_time,

case when logout\_time like '%:%:%' then unix\_timestamp(logout\_time,'HH:mm:ss')

when logout\_time like '%:%' then unix\_timestamp(logout\_time,'mm:ss')

else logout\_time

end as logout\_time,

case when duration like '%:%:%' then unix\_timestamp(duration,'HH:mm:ss')

when duration like '%:%' then unix\_timestamp(duration,'mm:ss')

else duration

end as duration

from agent\_loging\_report\_bkp;

3. List of all agents names.

select "agent\_loging\_report", count(DISTINCT(agent\_name)) as in\_log\_report from agent\_loging\_report

4. Find out agent average rating.

select agent\_name, ROUND(AVG(average\_rating), 2) as Average\_rating

from agent\_performance

group by agent\_name

order by Average\_rating desc;

5. Total working days for each agents

select "agent\_performance", agent\_name, count(distinct ag\_date) as working\_days

from agent\_performance

group by agent\_name;

6. Total query that each agent have taken

select agent\_name, SUM(total\_chats) as total\_queries

from agent\_performance

group by agent\_name;

7. Total Feedback that each agent have received

select agent\_name, SUM(total\_feedback) as total\_feedbacks

from agent\_performance

group by agent\_name

order by total\_feedbacks;

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

select agent\_name, AVG(average\_rating)

from agent\_performance

group by agent\_name

having ROUND(AVG(average\_rating),3) between 3.5 and 4

order by ROUND(AVG(average\_rating),1) desc;

select agent\_name, AVG(average\_rating)

from agent\_performance

group by agent\_name

having AVG(average\_rating) between 3.5 and 4;

9. Agent name who have rating less than 3.5

select agent\_name, AVG(average\_rating)

from agent\_performance

group by agent\_name

having AVG(average\_rating) < 3.5;

10. Agent name who have rating more than 4.5

select agent\_name, AVG(average\_rating)

from agent\_performance

group by agent\_name

having ROUND(AVG(average\_rating), 3) > 4.5;

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

select agent\_name, COUNT(average\_rating) AS frequency

from agent\_performance

WHERE average\_rating > 4.5

group by agent\_name;

12. average weekly response time for each agent

select agent\_name, weekofyear(ag\_date) as week\_no, AVG(average\_response\_time)

from agent\_performance

group by agent\_name, weekofyear(ag\_date);

13. average weekly resolution time for each agents

select agent\_name, weekofyear(ag\_date) as week\_no, AVG(average\_resolution\_time)

from agent\_performance

group by agent\_name, weekofyear(ag\_date);

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

select agent\_name, count(total\_chats)

from agent\_performance

where total\_chats <> 0

group by agent\_name;

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

select agent\_name, weekofyear(ag\_date) as week\_no, (sum(duration)/60)/60 as hours\_worked

from agent\_loging\_report

group by agent\_name, weekofyear(ag\_date);

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.

set hive.variable.substitute=true;

Inner Join:

with data1 as (select agent\_name, (sum(duration)/60)/60 as hours\_worked from agent\_loging\_report where agent\_name like 'Ayushi Mishra' group by agent\_name)

select ap.agent\_name, sum(ap.total\_chats) as total\_chats, sum(ap.average\_response\_time/60)/60 as response\_time, AVG(ap.average\_rating) as avg\_rating,

sum(ap.average\_resolution\_time/60)/60 as resolution\_time, sum(ap.total\_feedback) as total\_feedback, d.hours\_worked

from data1 d

inner join agent\_performance ap

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked;

OR

select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from agent\_performance ap

inner join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked;

Loading data from query into Local File system

hive -e 'set hive.cli.print.header=true; select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from hive\_class\_b1.agent\_performance ap

inner join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from hive\_class\_b1.agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked' | sed 's/[\t]/,/g' > /Users/thestupidmonk/Downloads/InnerJoinOutputfile.csv;

Alternate Way

insert overwrite local directory '/Users/thestupidmonk/Downloads/RightJoinOutputfile.csv' row format delimited fields terminated by ',' select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from agent\_performance ap

Right join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked;

Left Join:

select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from agent\_performance ap

left join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked;

Loading data from query into Local File system

hive -e 'set hive.cli.print.header=true; select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from hive\_class\_b1.agent\_performance ap

left join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from hive\_class\_b1.agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked' | sed 's/[\t]/,/g' > /Users/thestupidmonk/Downloads/LeftJoinOutputfile.csv;

Right Join:

select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from agent\_performance ap

Right join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked;

Loading data from query into Local File system

hive -e 'set hive.cli.print.header=true; select ap.agent\_name, sum(ap.total\_chats) as total\_chats, Round(sum(ap.average\_response\_time/60)/60, 3) as response\_time,

Round(sum(ap.average\_resolution\_time/60)/60, 3) as resolution\_time, Round(avg(average\_rating), 3) as avg\_rating, sum(ap.total\_feedback) as total\_feedback, Round(d.hours\_worked, 3) as hours\_served

from hive\_class\_b1.agent\_performance ap

Right join (select agent\_name, (sum(duration)/60)/60 as hours\_worked from hive\_class\_b1.agent\_loging\_report group by agent\_name) d

on d.agent\_name = ap.agent\_name

group by ap.agent\_name, d.hours\_worked' | sed 's/[\t]/,/g' > /Users/thestupidmonk/Downloads/LeftJoinOutputfile.csv;

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

First set the below properties

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.patition.mode=nonstrict;

Create Partition\_bucketed table:

create table partition\_bucketed\_loging

(

s\_no int,

date date,

login\_time string,

logout\_time string,

duration string

)

partitioned by (agent string)

clustered by(s\_no)

into 4 buckets

row format delimited

fields terminated by ','

stored as textfile;

Load data into Partition\_bucketed table:

insert overwrite table partition\_bucketed\_loging partition(agent) select s\_no, date, login\_time, logout\_time, duration, agent from agent\_loging;

s