**Name : Omkar Gharat   
  
Download Dataset 1 -** [https://drive.google.com/file/d/1WrG-9qv6atP-W3P\_gYln1hHyFKRKMHP/view](https://drive.google.com/file/d/1WrG-9qv6atP-W3P_-gYln1hHyFKRKMHP/view)

**Download Dataset 2 -**<https://drive.google.com/file/d/1-JIPCZ34dyN6k9CqJa-Y8yxIGq6vTVXU/view>

These are datasets for completing the project.

**1) Create a schema based on the given dataset. AgentPerformance Table Schema** create table agentperformance(

sno int, date Timestamp, agentname string, TotalChats int,

avgresponsetime Timestamp, avgresolutiontime Timestamp, avgrating int,

TotalFeedback int

)

row format delimited fields terminated by ‘,’ stored as textfile

**AgentLogingReport Table Schema**

create table if not exists agentloging( sno int,

agentname string, Date Timestamp,

LoginTime Timestamp,

LogoutTime Timestamp,

Duration Timestamp

)

row format delimited fields terminated by ‘,’ stored as textfile tblproperties ("skip.header.line.count" = "1")'

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

load data local inpath ‘/home/cloudera/Desktop/AgentPerformance.csv’ into table agentperformance;

load data local inpath ‘/home/cloudera/Desktop/AgentLogingReport.csv’ into table agentloging;

1. **List of all agents’ names.**

Select DISTINCT agentname from agentperformance;

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

Select agentname , avg(avgrating) from agentperformance group by agentname;

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

Select date,agentname from agentperformance group by date ,agentname;

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

Select agentname , sum(totalchats) from agentperformance group by agentname;

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

Select agentname , sum(totalfeedbacks) from agentperformance group by agentname;

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

Select agentname, avgrating from agentperformance group by agentname having avg(avgrating)>=3.5 AND avg(avgrating)<=4;

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

Select agentname, avgrating from agentperformance group by agentname having avg(avgrating)<3.5;

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

Select agentname, avgrating from agentperformance group by agentname having avg(avgrating)>4.5;

1. **How many feedback agents have received more than 4.5 average.**  Select count(avgrating) from agentperformance where avgrating> 4.5;
2. **Average weekly response time for each agent.**

hive -e "with weekresponse as (select agentname,

weekofyear(from\_unixtime(unix\_timestamp(date, 'MM/dd/yyyy'),'yyyy-MM-dd')) as week,\round((hour(avgresponsetime)\*3600+minute(avgresponsetime)\*60+second(avgr esponsetime))/60,2) as responsetime \ from agentperformance)

select agentName, week, avg(responsetime) from weekresponse group by agentname,week";

**13.Average weekly resolution time for each agent.**hive -e "with weekresolution as (

select agentname,\ weekofyear(from\_unixtime(unix\_timestamp(Date, 'MM/dd/yyyy'),'yyyy-MMdd')) as week,\round((hour(avgresolutiontime)\*3600+minute(avgresolutiontime

e)\*60+second(avgresolutiontime))/60,2) as resolutiontime\ from agentperformance); select agentname, week, avg(resolutiontime) from weekresolution group by agentname,week"

1. **Find the number of chats on which they have received feedback.** select sum (totalchats), AgentName from AgentPerformance where totalfeedback! =0 group by agentname

1. **Total contribution hour for each and every agent’s weekly basis.**

hive -e "with TotalContribution as (

select agentname, \weekofyear(from\_unixtime(unix\_timestamp(Date, 'dd-MMM-yy'),'yyyy-MM-dd')) as week,\

round((hour(Duration)\*3600+minute(Duration)\*60+second(Duration))/3600,2) as

hours \ from AgentLogingReport)

select agentname, week, sum(hours) from TotalContribution group by agentname,week;

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

**INNER JOIN**

Select agentperformance.\* , agentloging.\* from agentperformance JOIN agentlogging ON

(agentperformance.agentname =agentloging.agentname);  
 **Exporting data into the local System**

insert overwrite local directory ‘/home/cloudera/Desktop/innerjoin.csv’ Select agentperformance.\* , agentloging.\* from agentperformance JOIN agentlogging ON (agentperformance.agentname =agentloging.agentname);

**LEFT JOIN**

Select agentperformance.\* , agentloging.\* from agentperformance LEFT JOIN agentlogging ON (agentperformance.agentname =agentloging.agentname);   
**Exporting data into the local System**

insert overwrite local directory ‘/home/cloudera/Desktop/leftjoin.csv’ Select agentperformance.\* , agentloging.\* from agentperformance JOIN agentlogging ON (agentperformance.agentname =agentloging.agentname);

**RIGHT JOIN**

Select agentperformance.\* , agentloging.\* from agentperformance RIGHT JOIN agentlogging ON (agentperformance.agentname =agentloging.agentname); **Exporting data into the local System**

insert overwrite local directory ‘/home/cloudera/Desktop/rightjoin.csv’ Select agentperformance.\* , agentloging.\* from agentperformance JOIN agentlogging ON (agentperformance.agentname =agentloging.agentname);

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

SET hive.exec.dynamic.partition=true set hive.exec.dynamic.partition.mode=nonstrict SET hive.exec.dynamic.partition.mode=nonstrict set hive.enforce.bucketing=true create table DummyAgentPerformance(

sno int,

Date Timestamp, TotalChats int,

avgresponsetime Timestamp, avgresolutiontime Timestamp, avgrating int,

TotalFeedback int

)

partitioned by (agentname string) clustered by (Date) sorted by (Date) into 3 buckets row format delimited terminated by fields ‘,’ stored as textfile tblproperties ("skip.header.line.count" = "1");

INSERT OVERWRITE TABLE DummyAgentPerformance PARTITION(agentname) select \* from AgentPerformance;