Project 1

Prashanth

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 IF NOT EXISTS
AgentPerformance (
                    SL_No int,
                    Date to_date,
                    Agent_Name String,
                    Total_Chats int,
                    Average_Response_Time TIME_STAMP,
                    Average_Rating Float,
                    Total Feedback int
                    )
                    ROW FORMAT DELIMITED
                    FIELDS TERMINATED BY '\t'
                    LINES TERMINATED BY '\n'
                    STORED AS TEXTFILE;
```

---

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

LOAD DATA INPATH '/path/to/data/file' INTO TABLE AgentPerformance;

---

3. List of all Agent_Name.

SELECT DISTINCT Agent_Name FROM AgentPerformance;

---

4. Find out agent Average_Rating.

SELECT Agent_Name, AVG(Average_Rating) as Avg_Rating FROM AgentPerformance GROUP BY Agent_Name;

---

5. Total working days for each agents

SELECT Agent_Name, COUNT(DISTINCT Date) as Total_Working_Days FROM AgentPerformance GROUP BY Agent_Name;

---

6. Total query that each agent have taken

SELECT Agent_Name, SUM(Total_Chats) as Total_Queries FROM AgentPerformance GROUP BY Agent_Name;

---

7. Total Feedback that each agent have received

SELECT Agent_Name, SUM(Total_Feedback) as Total_Feedback_Received FROM AgentPerformance GROUP BY Agent_Name;

---

8. Agent name who have Average_Rating between 3.5 to 4

SELECT Agent_Name FROM AgentPerformance WHERE Average_Rating >= 3.5 AND Average_Rating <= 4;

---

9. Agent name who have rating less than 3.5

SELECT Agent_Name FROM AgentPerformance WHERE Average_Rating < 3.5;

---

10. Agent name who have rating more than 4.5

SELECT Agent_Name FROM AgentPerformance WHERE Average_Rating > 4.5;

---

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

SELECT COUNT(*) FROM AgentPerformance WHERE Average_Rating > 4.5 AND Total_Feedback > 0;

---

12. average weekly response time for each agent

SELECT Agent_Name, WEEK(Date) as Week_Number, AVG(Average_Response_Time) as Avg_Response_Time FROM AgentPerformance GROUP BY Agent_Name, WEEK(Date);

---

13. average weekly resolution time for each agents

SELECT Agent_Name, WEEK(Date) as Week_Number, AVG(Average_Rating) as Avg_Rating FROM AgentPerformance GROUP BY Agent_Name, WEEK(Date);

---

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

SELECT Agent_Name, SUM(Total_Feedback) as Total_Feedback_Count FROM AgentPerformance GROUP BY Agent_Name;

---

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

SELECT Agent_Name, WEEK(Date) as Week_Number, SUM(Total_Chats) + SUM(Average_Response_Time) as Total_Contribution_Hours FROM AgentPerformance GROUP BY Agent_Name, WEEK(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.

Inner

SELECT * FROM AgentPerformance a INNER JOIN OtherTable b ON a.Agent_Name = b.Agent_Name;

Left

SELECT * FROM AgentPerformance a LEFT JOIN OtherTable b ON a.Agent_Name = b.Agent_Name;

Right

SELECT * FROM AgentPerformance a RIGHT JOIN OtherTable b ON a.Agent_Name = b.Agent_Name;

INSERT OVERWRITE LOCAL DIRECTORY '/path/to/output/directory' SELECT * FROM JoinedTable;

---

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

```
CREATE TABLE AgentPerformance_Partitioned (
   SL_No int,
   Date to_date,
   Total_Chats int,
   Average_Response_Time TIME_STAMP,
   Average_Rating Float,
   Total_Feedback int
)
PARTITIONED BY (Agent_Name string)
CLUSTERED BY (SL_No) INTO 4 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```