Total Marks 100
Deadline: 23-05-2025

This dataset contains real-world medical appointment records collected from a US state Virginia. It includes information on 10,000 appointments, with demographic and clinical details about patients and whether they showed up for their scheduled appointments.

Key Variables:

- PatientId: Unique identifier for each patient
- AppointmentID: Unique identifier for each appointment
- Gender: Gender of the patient
- ScheduledDay: Date and time when the appointment was scheduled
- AppointmentDay: Date when the appointment was to take place
- Age: Age of the patient
- Neighbourhood: Area where the patient lives
- Scholarship: Indicates whether the patient is enrolled in a welfare program
- Hipertension, Diabetes, Alcoholism, Handcap: Indicators of health conditions
- SMS_received: Whether the patient received an SMS reminder
- Showed_up: Whether the patient attended the appointment
- Date.diff: Calculated difference in days between the scheduled and appointment date

Scenario:

Design and run SQL queries to explore, analyze, and derive insights from the appointment data to understand factors affecting patient attendance.
The ultimate goal is to help the clinic reduce no-shows, improve communication with patients, and optimize resource allocation by identifying key patterns using SQL-based analysis.

Deliverables:

The deliverables expected for this project are .sql solution files and a comprehensive report detailing your findings from the analysis.

Hint: Avoid solely relying on LLM like Chatgpt. Use your own creativity.

Basic SQL & Data Retrieval

1. Retrieve all columns from the Appointments table.
2. List the first 10 appointments where the patient is older than 60.
3. Show the unique neighborhoods from which patients came.
4. Find all female patients who received an SMS reminder. Give count of them
5. Display all appointments scheduled on or after '2023-05-01' and before '2023-06-01'.

Data Modification & Filtering

6. Update the 'Showed_up' status to 'Yes' where it is null or empty
7. Add a new column AppointmentStatus using a CASE statement:
   - 'No Show' if Showed_up = 'No'
   - 'Attended' otherwise
8. Filter appointments for diabetic patients with hypertension.
9. Order the records by Age in descending order and show only the top 5 oldest patients.
10. Limit results to the first 5 appointments for patients under age 18.

Aggregation & Grouping

11. Find the average age of patients for each gender.
12. Count how many patients received SMS reminders, grouped by Showed_up status.
13. Count no-show appointments in each neighborhood using GROUP BY.
14. Show neighborhoods with more than 100 total appointments (HAVING clause).
15. Use CASE to calculate the total number of:

- children (Age < 12)
- adults (Age BETWEEN 12 AND 60)
- seniors (Age > 60)

Window Functions

16. Tracks how appointments accumulate over time in each neighbourhood. (Running Total of Appointments per Day) In simple words: How many appointments were there each day and how do the total appointments keep adding up over time in each neighborhood?

17. Use Dense_Rank() to rank patients by age within each gender group.

18. How many days have passed since the last appointment in the same neighborhood? (Hint: DATEDIFF and Lag) (This helps to see how frequently appointments are happening in each neighborhood.)

19. Which neighborhoods have the highest number of missed appointments? Use DENSE_RANK() to rank neighborhoods based on the number of no-show appointments.

20. Are patients more likely to miss appointments on certain days of the week?
Steps to follow for question # 20

- (Use the AppointmentDay column in function dayname() to extract the day name (like Monday, Tuesday, etc.).
- Count how many appointments were scheduled, how many showed up (showed_up = "yes") and how many were missed (Showed_up = 'No') on each day.
- Calculate the percentage of shows and no-shows for better comparison between days.
- Formula: (count of Showed_up = 'yes' / total appointment count) * 100, Use round function to show upto two decimal points
- Sort the result by No_Show_Percent in descending order to see the worst-performing days first.

Happy Querying!