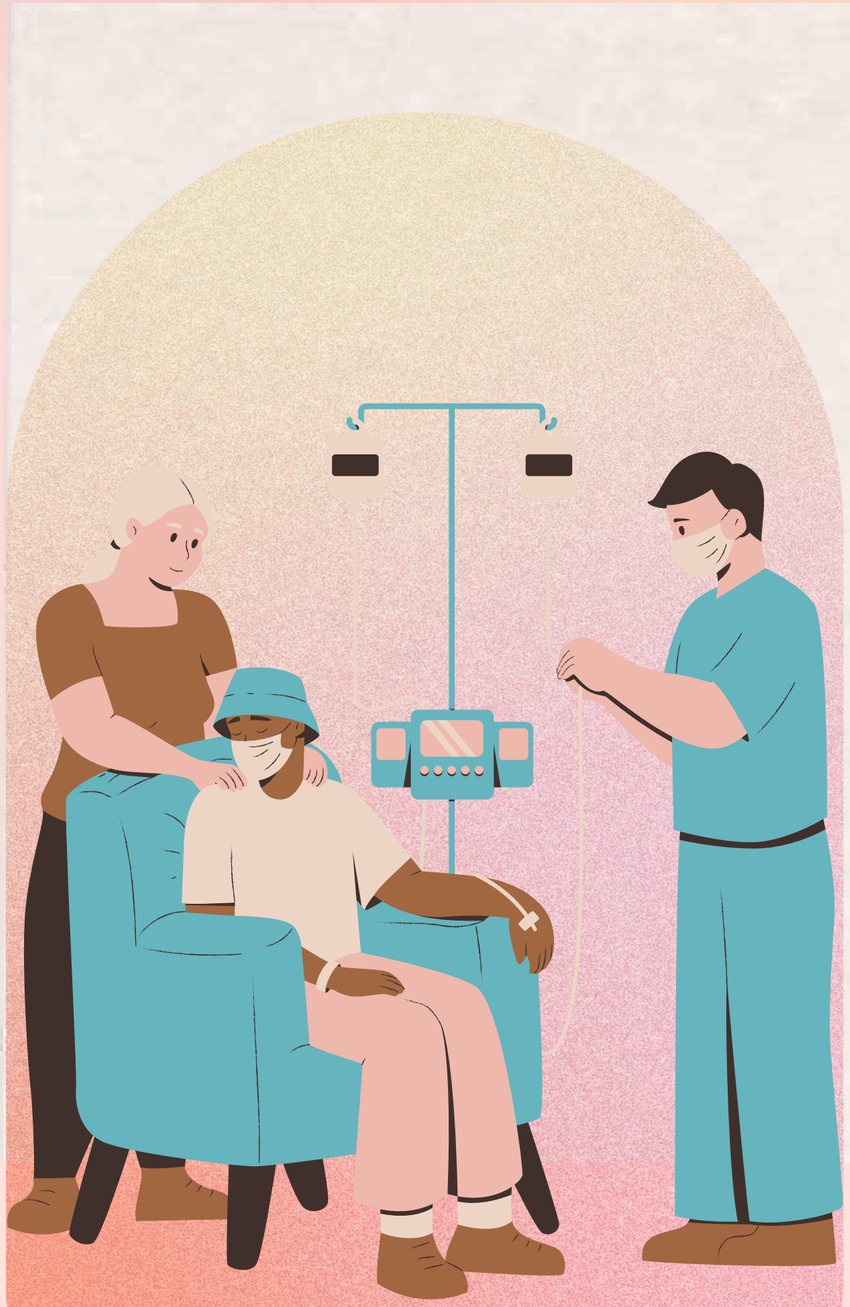
# MEDICAL RECORDS ANALYTICS PROJECT

Presented By :

**Paridhi Bhardwaj**



# PROJECT OVERVIEW

This project focuses on analyzing a hospital's relational database consisting of Patients, Appointments, and Medical Records. Using SQL, I performed data cleaning, joins, aggregations, and window functions to answer real-world healthcare queries. The analysis covered patient demographics, departmental appointment trends, doctor performance, and medical record insights. The project demonstrates how SQL can transform raw hospital data into actionable insights for process improvement and decision-making in healthcare management.

# ABOUT DATASET

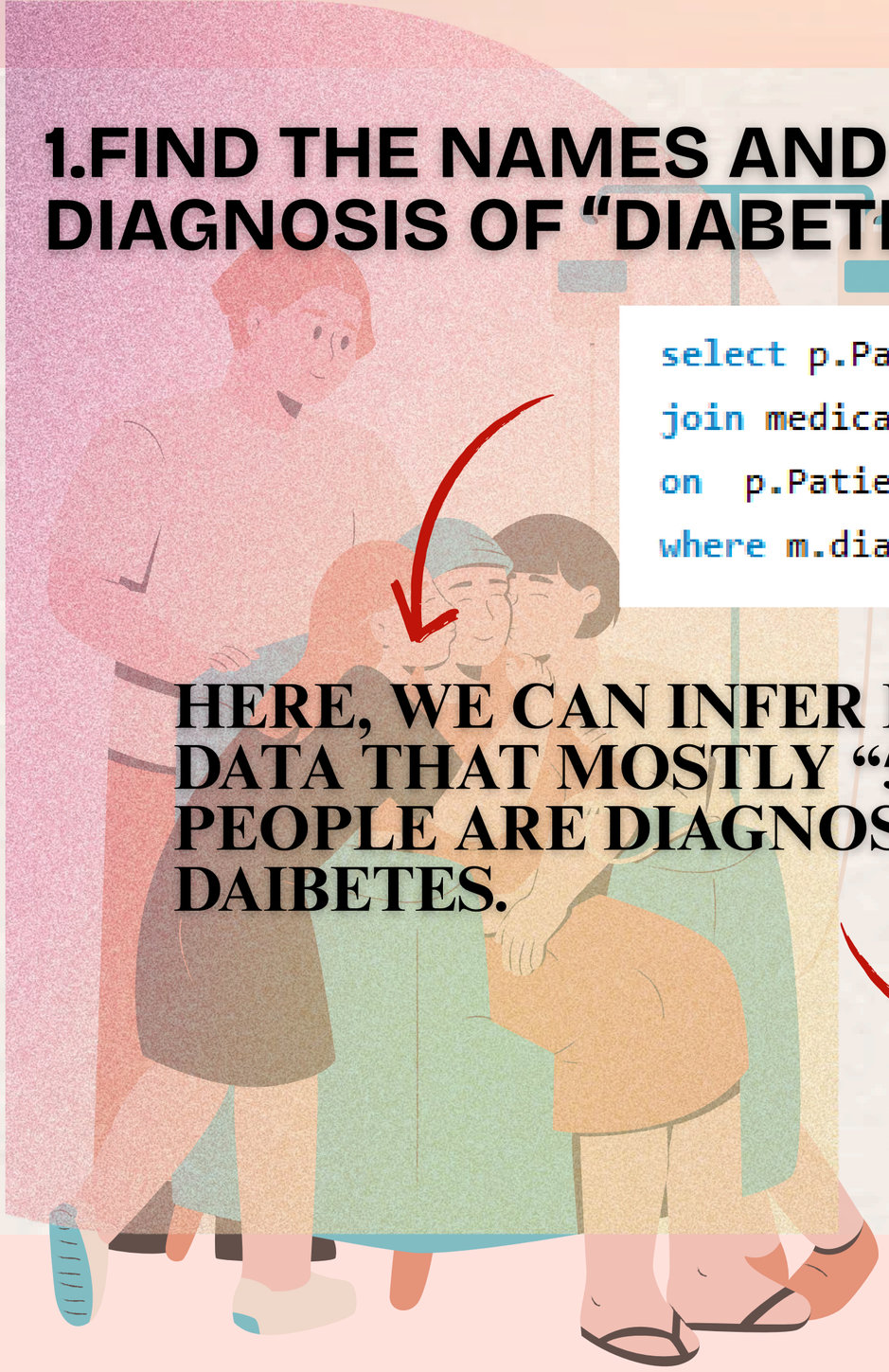
**Appointments:** AppointmentID, PatientID, DoctorID, AppointmentDate, Department, Status

**MedicalRecords:** RecordID, PatientID, Diagnosis, BloodPressure, Glucose

**Patients:** PatientID, Name, Age, Gender

# 1.FIND THE NAMES AND AGES OF PATIENTS WHO HAVE A DIAGNOSIS OF “DIABETES”.

```
select p.PatientID, p.name , p.age , m.Diagnosis from patients p
join medical_record  m
on  p.PatientID = m.PatientID
where m.diagnosis = "diabetes";
```



**HERE, WE CAN INFER FROM THE DATA THAT MOSTLY “50+” AGED PEOPLE ARE DIAGNOSED WITH DAIBETES.**

| PatientID | name | age | Diagnosis |
|---|---|---|---|
| 7 | Teresa Davis | 61 | Diabetes |
| 26 | Colleen Flores | 49 | Diabetes |
| 27 | Jennifer Mclean | 67 | Diabetes |
| 31 | Andrew Gross | 62 | Diabetes |
| 32 | Anna Robinson | 73 | Diabetes |
| 36 | Jon James | 62 | Diabetes |
| 44 | Johnathan Brown | 50 | Diabetes |
| 50 | Brandon Graham | 73 | Diabetes |
| 54 | Jennifer Dixon | 51 | Diabetes |

esult 20 ×

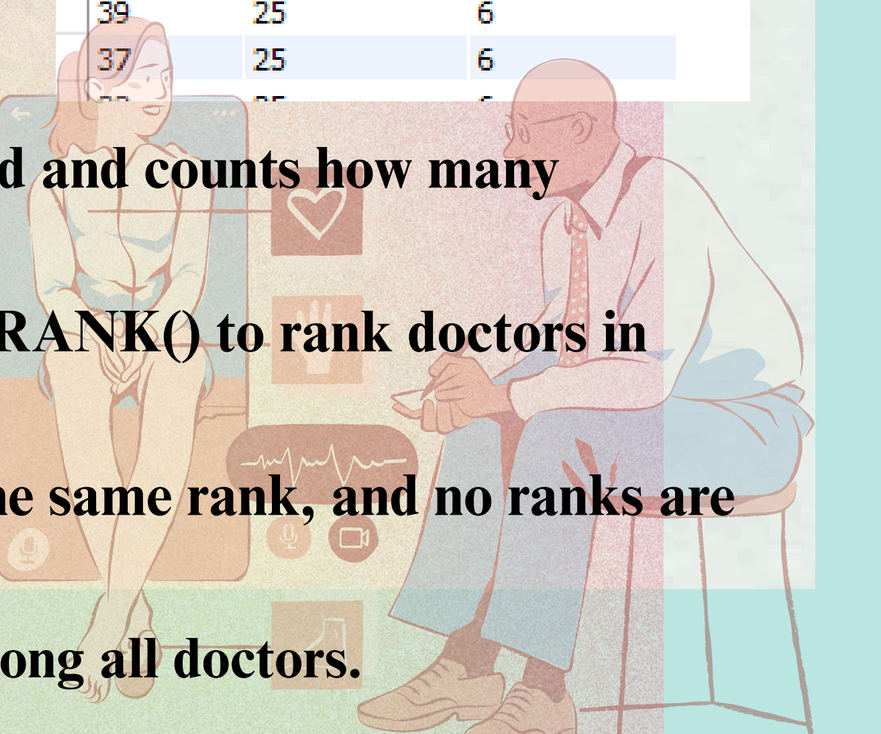
# 2. FOR EACH DOCTOR, CALCULATE THE NUMBER OF APPOINTMENTS THEY HANDLED AND THEIR RANK AMONG ALL DOCTORS.

```sql
with b as (select doctorid , count(patientid) as total_appoint
from appointments
group by doctorid)

select * ,
dense_rank() over (order by total_appoint desc)  as doctor_rank
from b;
```

Result Grid | Filter Rows:

| doctorid | total_appoint | doctor_rank |
|---|---|---|
| 26 | 30 | 1 |
| 28 | 29 | 2 |
| 35 | 28 | 3 |
| 33 | 27 | 4 |
| 49 | 27 | 4 |
| 47 | 26 | 5 |
| 39 | 25 | 6 |
| 37 | 25 | 6 |



- The CTE b first groups the appointments table by each doctorid and counts how many patients (appointments) each doctor handled → total_appoint.
- In the main query, it selects these results and applies DENSE_RANK() to rank doctors in descending order of total_appoint.
- This way, doctors with the same number of appointments get the same rank, and no ranks are skipped.
- Final output = each doctor's total appointments + their rank among all doctors.

# 3. FOR EACH DOCTOR, CALCULATE THE PERCENTAGE OF CANCELLED APPOINTMENTS.

- CTE a → counts how many appointments were cancelled for each doctorid.
- CTE b → counts the total number of appointments for each doctor using a window function.
- The main query joins a and b on doctorid.
- It calculates the percentage of cancelled appointments for each doctor with:
- Final output = doctor ID, cancelled appointments, total appointments, and % cancelled.

```
    with a as (select doctorid , count(status) cancelled_appoint
     from appointments
    where Status = "cancelled"
    group by doctorid ),

b as (select distinct DoctorID ,
count(*) over (PARTITION by doctorid) as total_appoints
      from appointments)

    select a.doctorid,a.cancelled_appoint , b.total_appoints ,
    round(cancelled_appoint * 100 /  total_appoints,2) as percentage
    from a left join b on
    a.doctorid = b.doctorid;
```

| doctorid | cancelled_appoint | total_appoints | percentage |
|---|---|---|---|
| 39 | 10 | 25 | 40.00 |
| 29 | 5 | 19 | 26.32 |
| 11 | 3 | 16 | 18.75 |
| 22 | 4 | 21 | 19.05 |
| 44 | 5 | 21 | 23.81 |

# 4. FOR EACH PATIENT, SHOW THEIR FIRST DIAGNOSIS AND MOST RECENT DIAGNOSIS .

```
with b as (select m.patientid , a.AppointmentDate,
   m.diagnosis , row_number() over (PARTITION by m.PatientID order by a.appointmentdate asc ) as rn
   from medical_record m join appointments a
   on m.PatientID = a.PatientID),

 c as (select m.patientid , a.appointmentdate,
  m.Diagnosis , row_number() over (PARTITION by m.PatientID order by a.appointmentdate desc ) as rn
   from medical_record m join appointments a
   on m.PatientID = a.PatientID)

select b.patientid,
       b.diagnosis as first_diagnosis,
       b.appointmentdate as first_appointment,
       c.diagnosis as last_diagnosis,
       c.appointmentdate as last_appointment
from b join c
on b.patientid = c.patientid
where c.rn=1 and b.rn = 1 ;
```

This query is designed to track the medical journey of each patient by identifying their first and last recorded diagnoses along with the corresponding appointment dates. In the first **CTE (b)**, it uses **ROW_NUMBER()** ordered by appointment date in ascending order to capture the earliest diagnosis for every patient. Similarly, in the second **CTE (c**), it orders appointments in descending order to capture the most recent diagnosis.

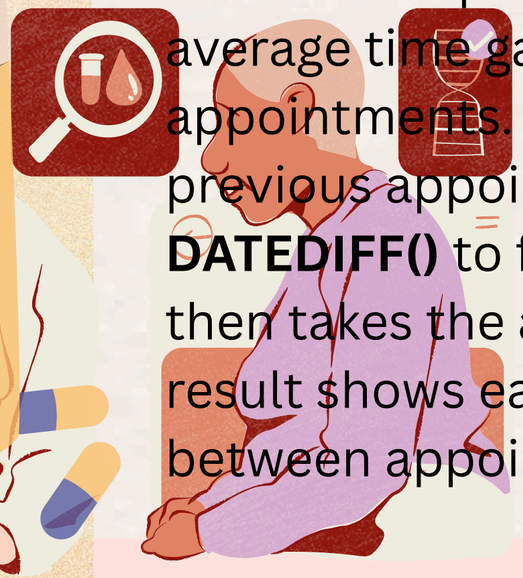
By joining these two sets on the patient ID and filtering for the first row (rn=1) in both, the final result neatly shows, for each patient, their **first diagnosis and appointment date alongside their most recent diagnosis and appointment date.** This provides a clear before-and-after snapshot of a patient's medical history.

| patientid | first_diagnosis | first_appointment | last_diagnosis | last_appointment |
|---|---|---|---|---|
| 1 | Cancer | 2024-01-01 00:00:00.000000000 | Cancer | 2024-01-01 00:00:00.000000000 |
| 2 | Cancer | 2024-01-01 08:46:07.567567567 | Cancer | 2024-01-01 08:46:07.567567567 |
| 3 | Heart Disease | 2024-01-01 17:32:15.135135135 | Heart Disease | 2024-01-01 17:32:15.135135135 |
| 4 | Asthma | 2024-01-02 02:18:22.702702702 | Asthma | 2024-01-02 02:18:22.702702702 |
| 5 | Heart Disease | 2024-01-02 11:04:30.270270270 | Heart Disease | 2024-01-02 11:04:30.270270270 |
| 6 | Hypertension | 2024-01-02 19:50:37.837837837 | Hypertension | 2024-01-02 19:50:37.837837837 |
| 7 | Diabetes | 2024-01-03 04:36:45.405405405 | Diabetes | 2024-01-03 04:36:45.405405405 |
| 8 | Flu | 2024-01-03 13:22:52.972972972 | Flu | 2024-01-03 13:22:52.972972972 |

# 5. FOR EACH DOCTOR, CALCULATE THE AVERAGE TIME GAP (IN DAYS) BETWEEN CONSECUTIVE APPOINTMENTS.ALTER

```sql
with a as (select doctorid , AppointmentDate,
   lag(appointmentdate) over (partition by doctorid order by appointmentdate) as prev_date
from appointments)

select doctorid,
round(avg(datediff(appointmentdate , prev_date )),2) as avg_gap
from a
group by doctorid ;
```



This query measures how frequently each doctor sees patients by calculating the average time gap between their consecutive appointments. It uses **LAG()** to capture the previous appointment date, applies **DATEDIFF()** to find the day difference, and then takes the average per doctor. The result shows each doctor's average interval between appointments.

| | doctorid | avg_gap |
|---|---|---|
| ▶ | 1 | 13.22 |
| | 2 | 16.90 |
| | 3 | 16.71 |
| | 4 | 22.60 |
| | 5 | 13.23 |
| | 6 | 21.25 |
| | 7 | 25.54 |
| | 8 | 17.10 |
| | 0 | 20.06 |

Result 25

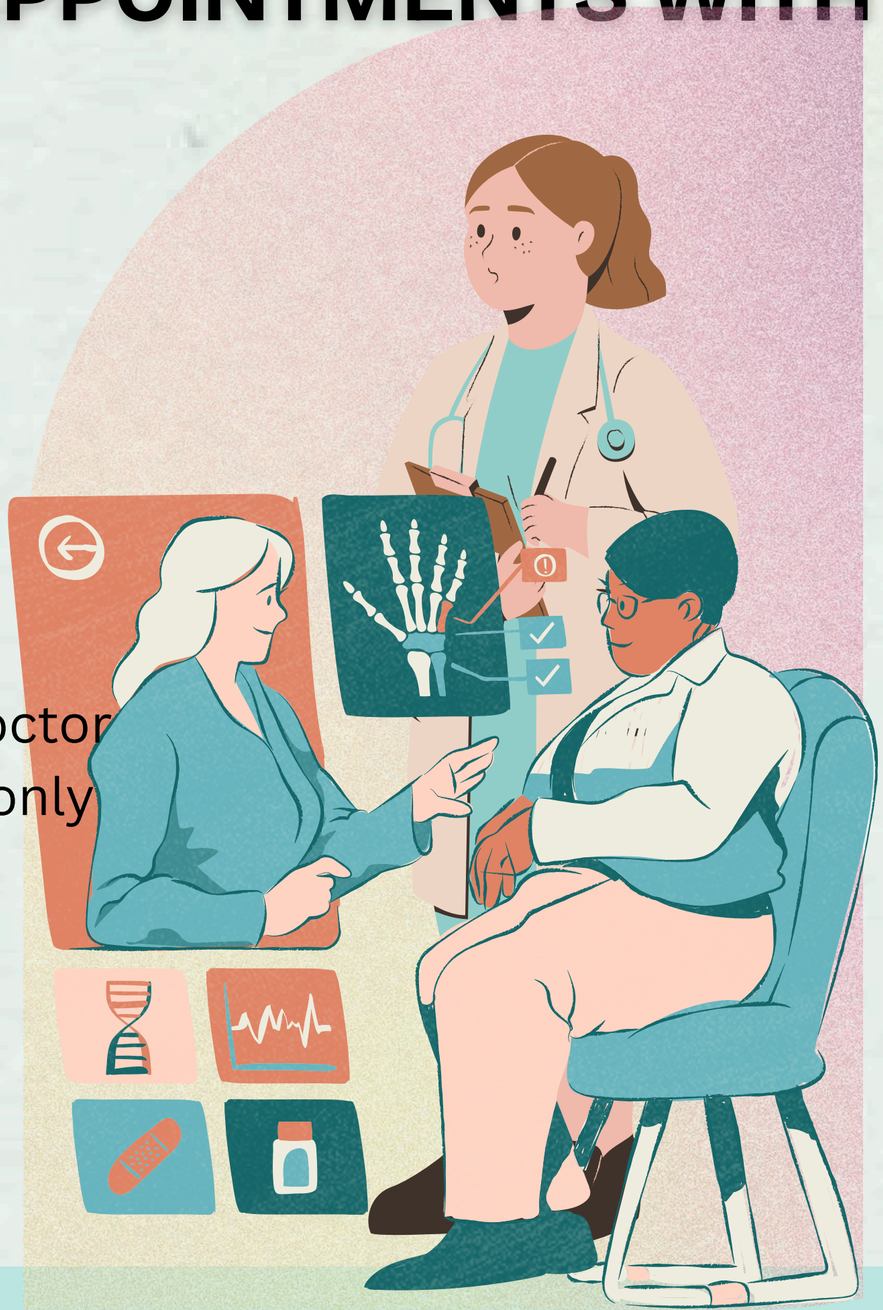
# 6. IDENTIFY PATIENTS WHO HAVE APPOINTMENTS WITH MORE THAN ONE DOCTOR

```
lect   m.patientid ,  count(distinct a.doctorid)
from medical_record m join appointments a
 on m.PatientID = a.PatientID
 group by m.patientid
 having count(a.doctorid) > 1
 order by m.patientid;
```

**Result**: **No patient** has consulted more than 1 distinct doctor — meaning each patient in your data is associated with only one doctor across their appointments.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

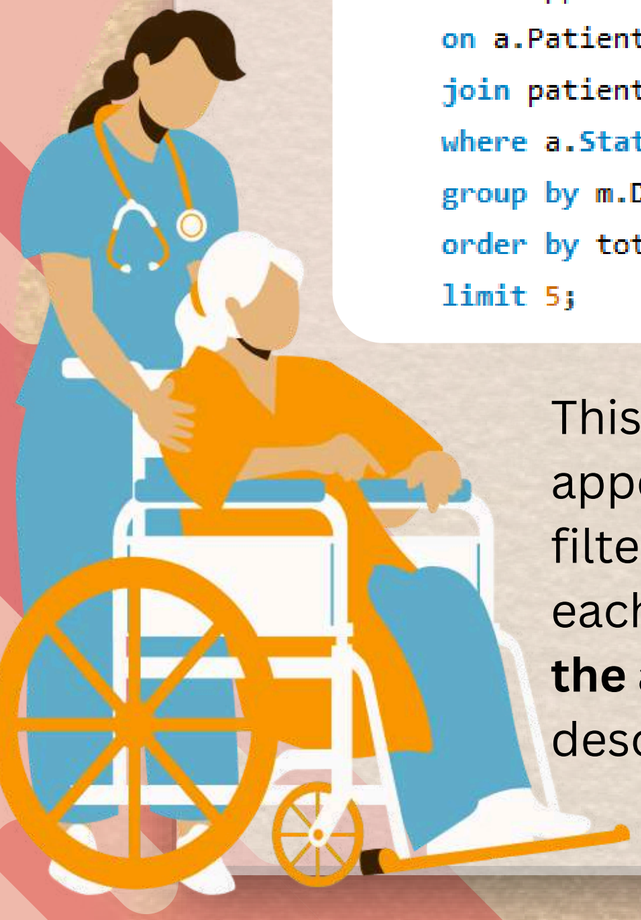
| patientid | count(distinct a.doctorid) |
|---|---|



# 7. FIND THE TOP 5 DIAGNOSES WITH THE HIGHEST NUMBER OF COMPLETED APPOINTMENTS,ALONG WITH THE AVERAGE AGE OF THOSE PATIENTS.

```sql
select count(a. appointmentid) as total_appointments ,
       round(avg(p.age),2) as age , m.diagnosis as diagnosis
       from appointments a join medical_record m
       on a.PatientID = m.PatientID
       join patients p on m.PatientID = p.PatientID
       where a.Status = "completed"
       group by m.Diagnosis
       order by total_appointments  desc
       limit 5;
```

| | total_appointments | age | diagnosis |
|---|---|---|---|
| ▶ | 52 | 47.10 | Cancer |
| | 40 | 50.35 | Diabetes |
| | 40 | 50.18 | Arthritis |
| | 35 | 47.06 | Asthma |
| | 35 | 49.77 | Hypertension |



This query identifies the top 5 diagnoses with the highest number of completed appointments. It joins the appointments, medical records, and patients tables, filters only completed appointments, and then groups results by diagnosis. For each diagnosis, it calculates the **total number of completed appointments and the average patient age**. Finally, it orders diagnoses by appointment count in descending order and returns the top five.

# 8. FIND THE DOCTORS IN EACH DEPARTMENT WHO HANDLED THE MOST COMPLETED APPOINTMENTS.

```
select doctorid , count(appointmentid) as total_appointments, department ,
dense_rank() over(partition by department order by count(appointmentid) desc ) as dept_rank
   from appointments
   group by  department,doctorid;
```

**DoctorID 33** and **DoctorID 46** each handled **6 appointments** → **tied at Rank 1.**
**DoctorID 8, 21, and 25** each handled **5 appointments** → **tied at Rank 2.**

| doctorid | total_appointments | department | dept_rank |
|---|---|---|---|
| 33 | 6 | Cardiology | 1 |
| 46 | 6 | Cardiology | 1 |
| 25 | 5 | Cardiology | 2 |
| 21 | 5 | Cardiology | 2 |
| 8 | 5 | Cardiology | 2 |
| 26 | 5 | Cardiology | 2 |
| 17 | 4 | Cardiology | 3 |
| 47 | 4 | Cardiology | 3 |

Result 29

- This shows how the query ranks doctors within the same department by appointment volume, giving equal rank if counts are the same.

# 9. COUNT HOW MANY APPOINTMENTS WERE CANCELLED VS COMPLETED.

```
select status , count(status) from appointments
group by status;
```

- the dataset shows that the highest number of appointments were Completed (270), while the fewest were Cancelled (227).

Result Grid | Filter Rows:

| status | count(status) |
|---|---|
| Cancelled | 227 |
| Scheduled | 249 |
| Completed | 270 |
| No-Show | 254 |

# 10. FOR EACH DEPARTMENT, FIND THE NUMBER OF UNIQUE PATIENTS, THE TOTAL NUMBER OF APPOINTMENTS, AND THE AVERAGE AGE OF PATIENTS WHO VISITED THAT DEPARTMENT.

- Cardiology had the most patients and appointments, with an older average age.
- Pediatrics patients are much younger (as expected).

```
select count(distinct a.patientid) distict_patients , a.department ,
count( a.appointmentid) total_appointmnt , round(avg(p.age),2) avg_age
from appointments a join patients p
on a.PatientID = p.PatientID
group by a.department ;
```

Result Grid | Filter Rows: | Export: | Wrap Ce

| distict_patients | department | total_appointmnt | avg_age |
|---|---|---|---|
| 145 | Cardiology | 145 | 49.63 |
| 141 | Dermatology | 141 | 47.15 |
| 151 | General Medicine | 151 | 49.61 |
| 140 | Neurology | 140 | 49.46 |
| 145 | Oncology | 145 | 49.75 |
| 153 | Orthopedics | 153 | 49.11 |
| 125 | Pediatrics | 125 | 47.93 |

- Neurology has fewer patients, but maybe those patients return more often.

# THANK YOU