AN MYSQL PROJECT REPORT

Healthcare Analytics with SQL

Project work done by
Thaslima Banu. S

Data Analytics

1. Write this query to create a database:

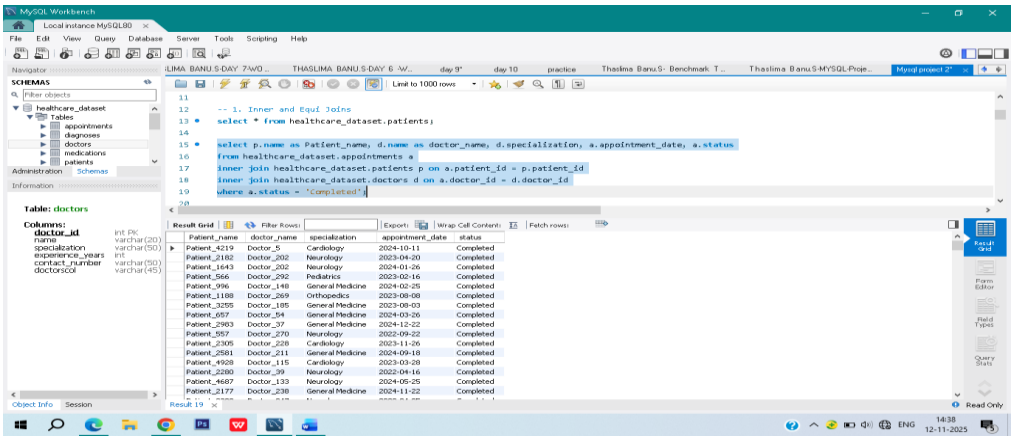
```
CREATE database Healthcare_Dataset;
use  Healthcare_Dataset;
```

2. Right click on the tables – Select – Table data Import Wizard to import the data from excel one by one.
3. Assign Primary Key(PK) and Foreign Key(FK) after importing table – Right click the imported table – Alter Table.

Tasks:

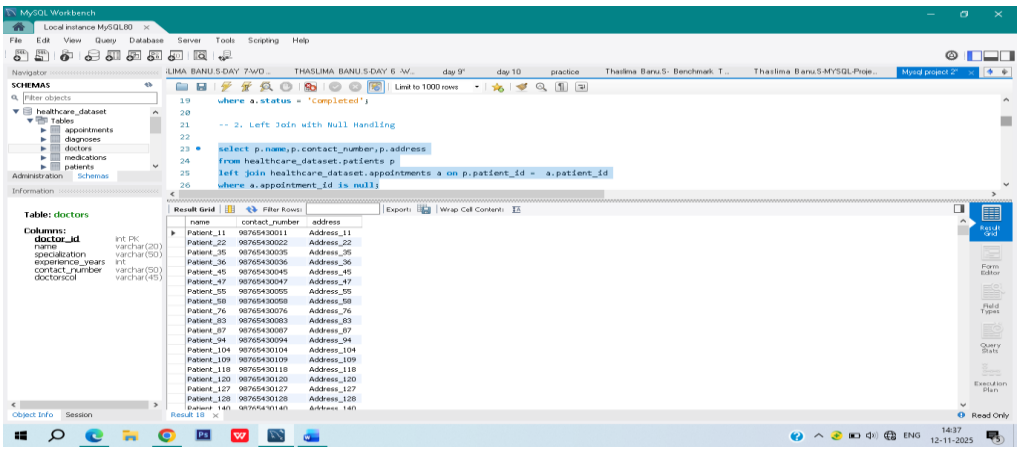
-- 1. Inner and Equi Joins

```
select p.name as patient_name, d.name as doctor_name, d.specialization, a.appointment_date, a.status
from healthcare_dataset.appointments a
inner join healthcare_dataset.patients p on a.patient_id = p.patient_id
inner join healthcare_dataset.doctors d on a.doctor_id = d.doctor_id
where a.status = “Completed”;
```



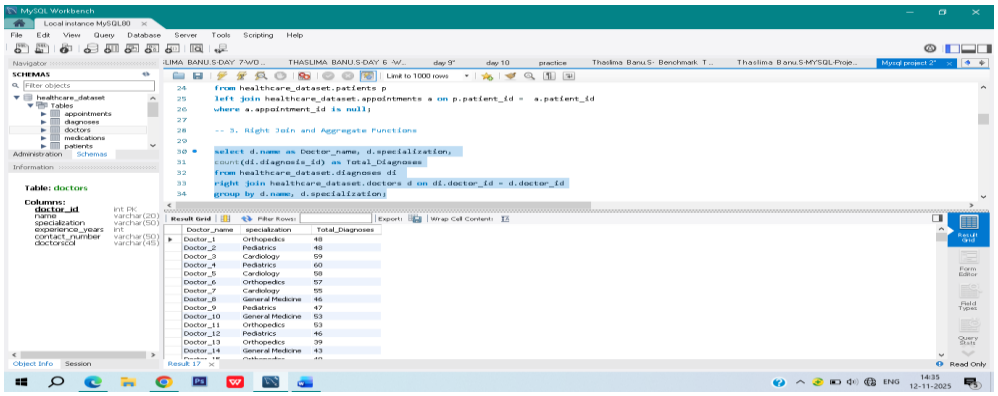
-- 2. Left Join with Null Handling

```
select p.name,p.contact_number,p.address
from healthcare_dataset.patients p
left join healthcare_dataset.appointments a on p.patient_id =  a.patient_id
where a.appointment_id is null;
```



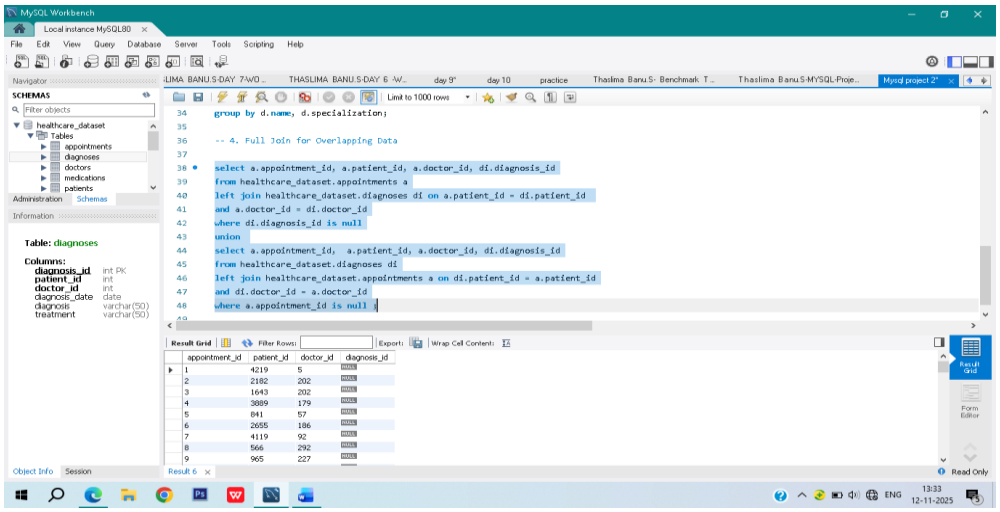
-- 3. Right Join and Aggregate Functions

```
select d.name as Doctor_name, d.specialization,
count(di.diagnosis_id) as Total_Diagnoses
from healthcare_dataset.diagnoses di
right join healthcare_dataset.doctors d on di.doctor_id = d.doctor_id
group by d.name, d.specialization;
```



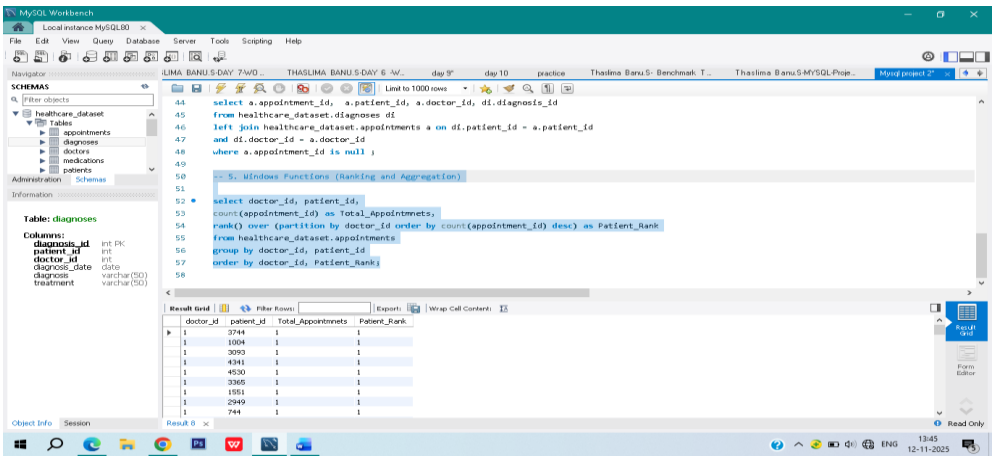
-- 4. Full Join for Overlapping Data

```
select a.appointment_id, a.patient_id, a.doctor_id, di.diagnosis_id
from healthcare_dataset.appointments a
left join healthcare_dataset.diagnoses di on a.patient_id = di.patient_id
and a.doctor_id = di.doctor_id
where di.diagnosis_id is null
union
select a.appointment_id,  a.patient_id, a.doctor_id, di.diagnosis_id
from healthcare_dataset.diagnoses di
left join healthcare_dataset.appointments a on di.patient_id = a.patient_id
and di.doctor_id = a.doctor_id
where a.appointment_id is null ;
```



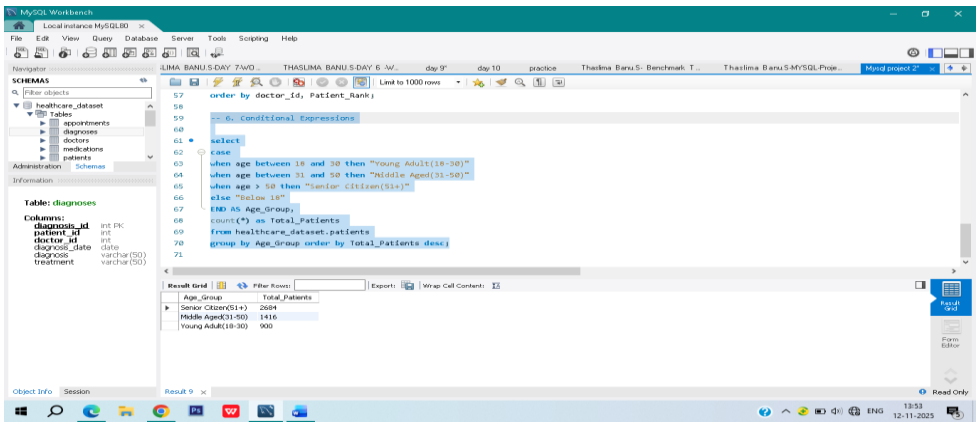
-- 5. Windows Functions (Ranking and Aggregation)

```
select doctor_id, patient_id,
count(appointment_id) as Total_Appointmnets,
rank() over (partition by doctor_id order by count(appointment_id) desc) as Patient_Rank
from healthcare_dataset.appointments
group by doctor_id, patient_id
order by doctor_id, Patient_Rank;
```



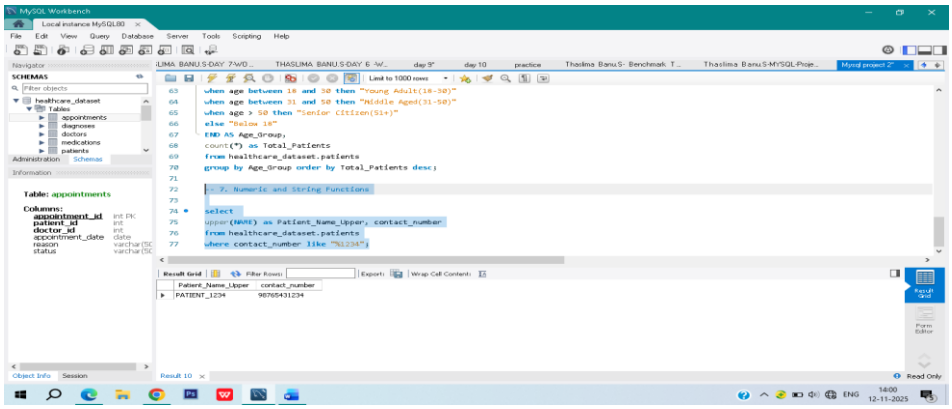
-- 6. Conditional Expressions

```
select
case
when age between 18 and 30 then "Young Adult(18-30)"
when age between 31 and 50 then "Middle Aged(31-50)"
when age > 50 then "Senior Citizen(51+)"
else "Below 18"
END AS Age_Group,
count(*) as Total_Patients
from healthcare_dataset.patients
group by Age_Group order by Total_Patients desc;
```



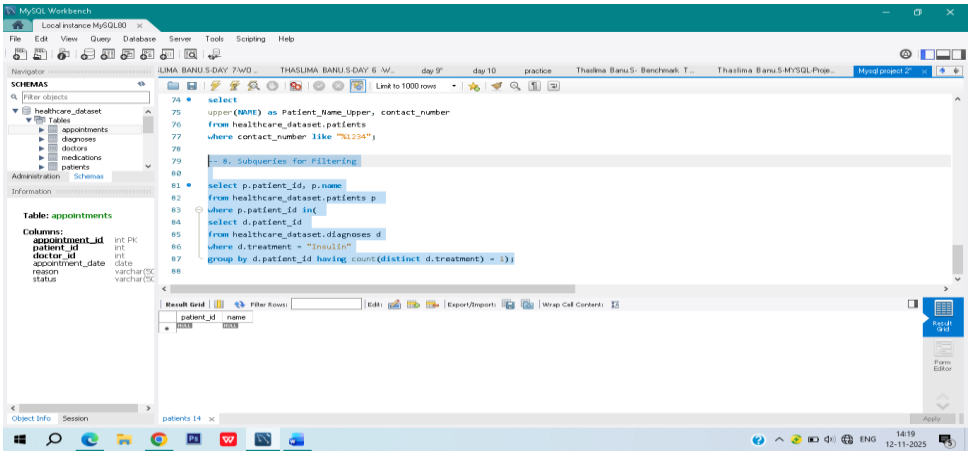
-- 7. Numeric and String Functions

```
select
upper(NAME) as Patient_Name_Upper, contact_number
from healthcare_dataset.patients
where contact_number like "%1234";
```



-- 8. Subqueries for Filtering

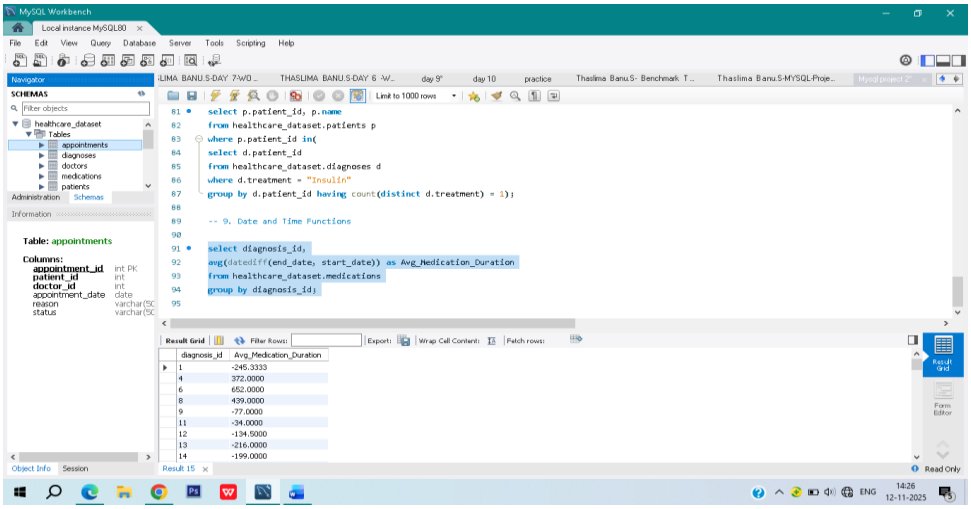
```
select p.patient_id, p.name
from healthcare_dataset.patients p
where p.patient_id in(
select d.patient_id
from healthcare_dataset.diagnoses d
where d.treatment = "Insulin"
group by d.patient_id having count(distinct d.treatment) = 1);
```



Based on the Dataset none of the patients received “Insulin” as the sole treatment.

-- 9. Date and Time Functions

```
select diagnosis_id,
avg(datediff(end_date, start_date)) as Avg_Medication_Duration
from healthcare_dataset.medications group by diagnosis_id;
```



-- 10. Complex Joins and Aggregation

```
select d.doctor_id, d.name, d.specialization,
count(distinct a.patient_id) as Unique_Patients
from healthcare_dataset.doctors d
join healthcare_dataset.appointments a on d.doctor_id = a.doctor_id
group by d.doctor_id, d.name, d.specialization
order by Unique_Patients desc limit 1;
```