Hospital Management Database Project Documentation

Project Title: Hospital Management Database System

Database Engine: Microsoft SQL Server

Project Type: Medium-High Complexity Relational DBMS

1. Database Overview

This project simulates a hospital environment using a relational database that manages patients, doctors, appointments, rooms, billing, insurance, tests, prescriptions, and staff. The database includes logic implemented using stored procedures and triggers to support real-world automation.

2. Core Tables and Relationships

Table	Description
Patients	Stores patient info: name, DOB, gender, contact
Doctors	Doctors' info + foreign key to Department
Departments	Medical departments like Cardiology, Emergency
Appointments	Connects patients with doctors
Rooms	Rooms with type (ICU/General/Private) and availability
Admissions	Patient hospitalization info with diagnosis and room
Bills	Billing records, linked to patient and admission
Medications	Medication info (name, dosage, description)
Prescriptions	Linked to appointments, holds notes
PrescriptionDetails	M:N between Prescriptions and Medications
Tests	Lab/imaging results linked to appointments
Insurance	Insurance info per patient
Staff	Non-doctor hospital employees (nurses, admins)

3. Stored Procedures Implemented

Procedure Name	Description
AddAppointment	Adds appointment with patient & doctor validation
GetPatientBills	Returns all bills of a patient
DischargePatientAndFreeRoom	Updates discharge date and frees room automatically
GetAvailableRoomsByType	Returns all available rooms of specific type

4. Triggers Implemented

Trigger Name	Description
trg_FreeRoomOnDischarge	Updates room availability to 1 when patient is discharged
trg_OccupyRoomOnAdmission	Updates room availability to 0 when patient is admitted
trg_PreventDeleteDoctorWithAppointments	Prevents deleting a doctor with linked appointments
trg_GenerateBillOnDischarge	Automatically generates bill upon discharge based on stay duration

5. Sample Enhancements (Optional Features)

- - Users & roles for multi-user access and permissions
- - Audit logging table for inserts/updates/deletes
- - Feedback table to track patient satisfaction
- - Trigger to compute insurance coverage on bill insert
- - Views for financial reporting and occupancy tracking
- - Procedure to admit patient and automatically assign room if available
- - Cancelled Appointments tracking

6. Testing Scenario Summary

"A patient is admitted into a room. The room becomes unavailable. Upon discharge, the room is marked as available and a bill is generated automatically based on the stay. Doctors cannot be deleted if they have active appointments. Appointments and prescriptions are managed using validated procedures."

Prepared By: **Amr Amgad Adel**

Date: **4/19/2025**

7. ERD and Query Results

Entity Relationship Diagram (ERD)

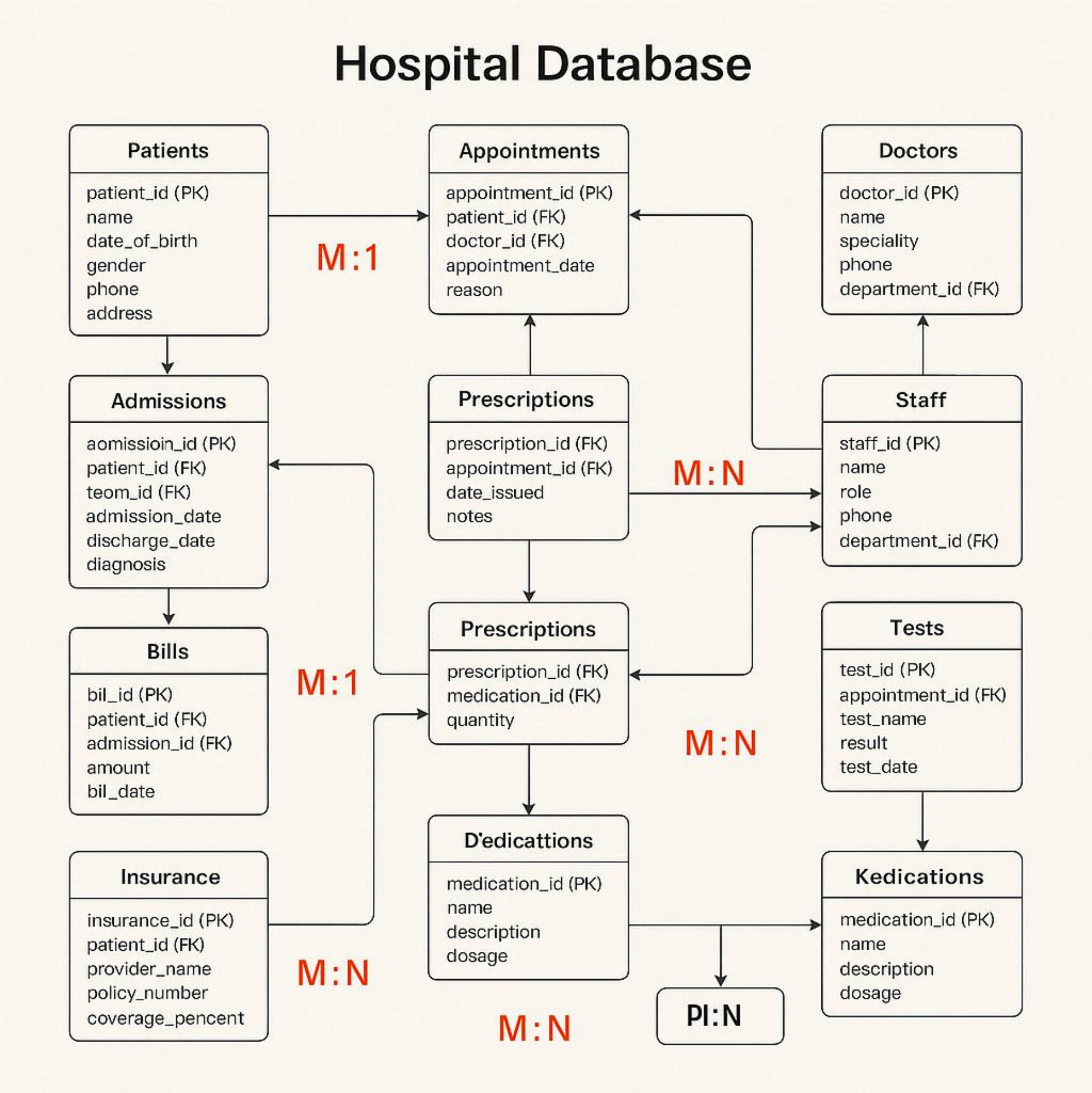
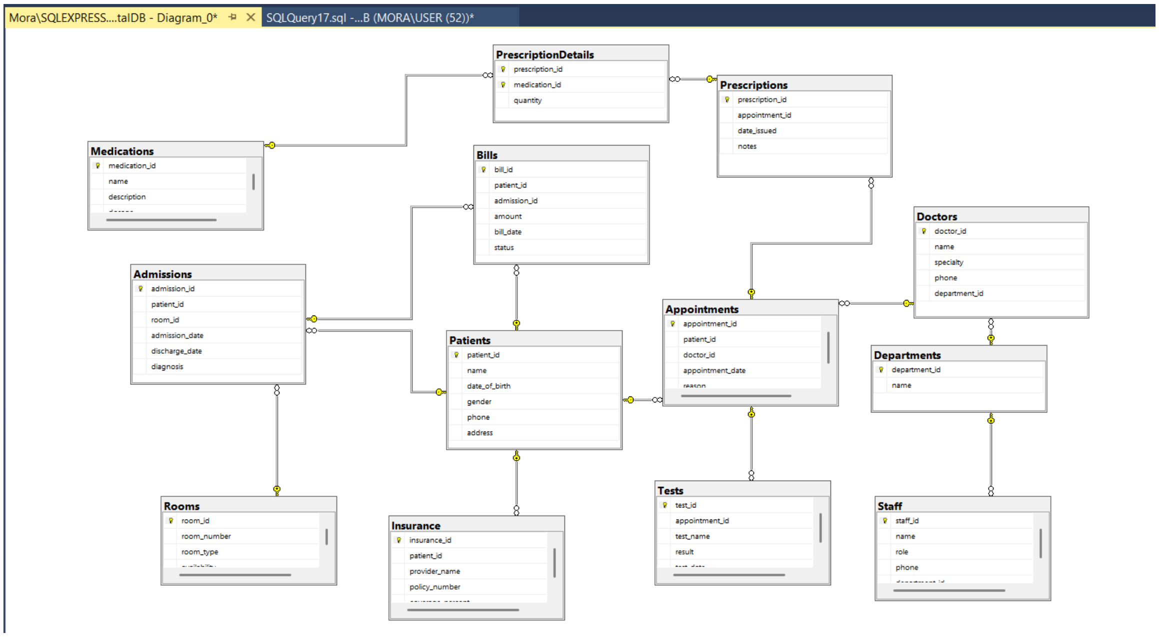
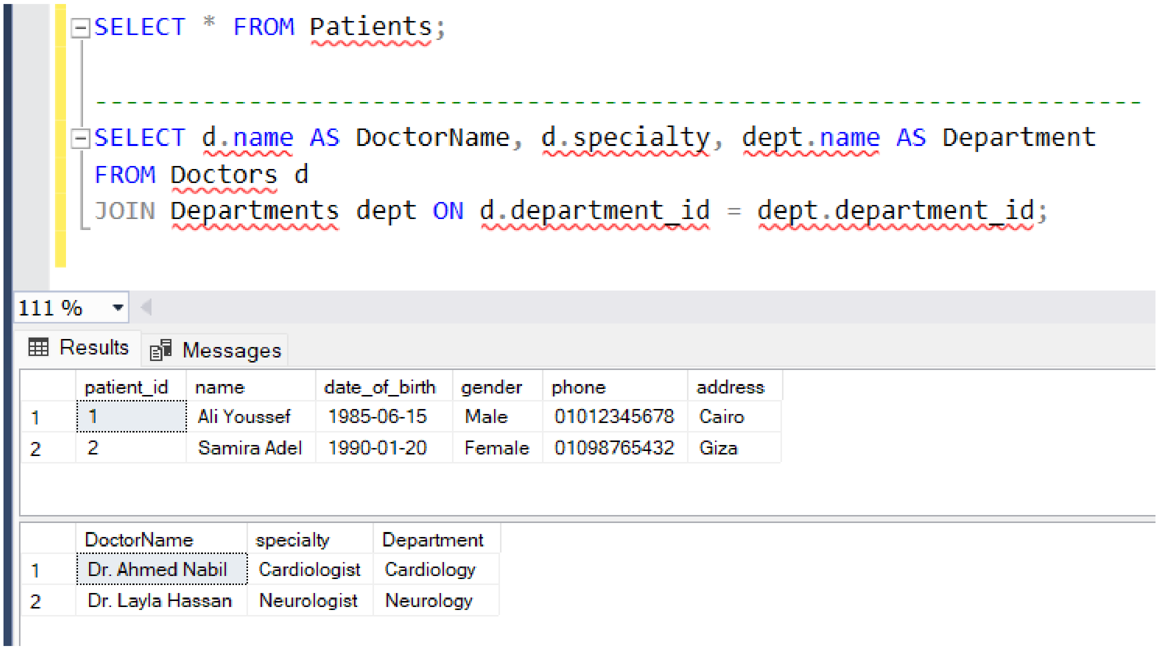


Figure 2: SQL Server ERD View from SSMS



Query 1: Display all patients and list of doctors with their departments.



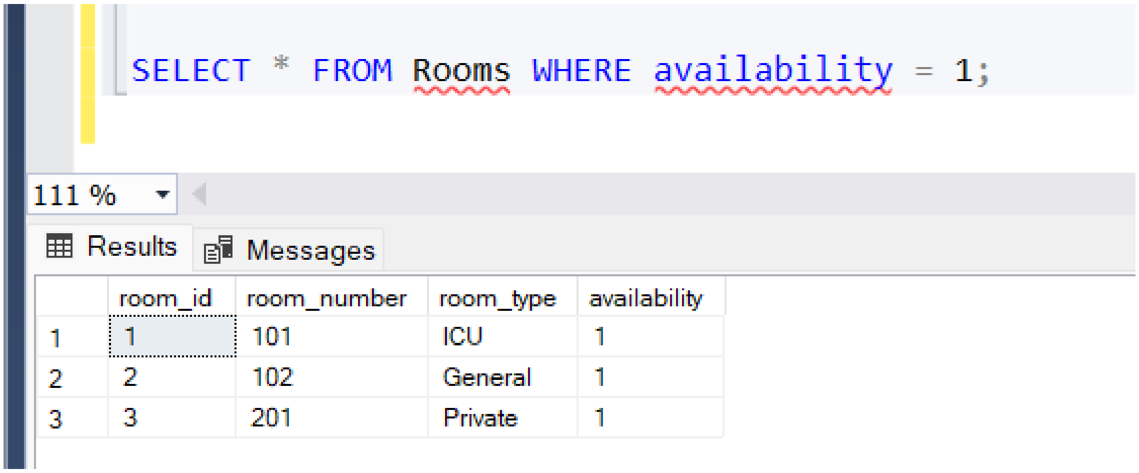
```
SELECT * FROM Patients;

---------------------------------------------------------------------------
SELECT d.name AS DoctorName, d.specialty, dept.name AS Department
FROM Doctors d
JOIN Departments dept ON d.department_id = dept.department_id;
```

	patient_id	name	date_of_birth	gender	phone	address
1	1	Ali Youssef	1985-06-15	Male	01012345678	Cairo
2	2	Samira Adel	1990-01-20	Female	01098765432	Giza

	DoctorName	specialty	Department
1	Dr. Ahmed Nabil	Cardiologist	Cardiology
2	Dr. Layla Hassan	Neurologist	Neurology

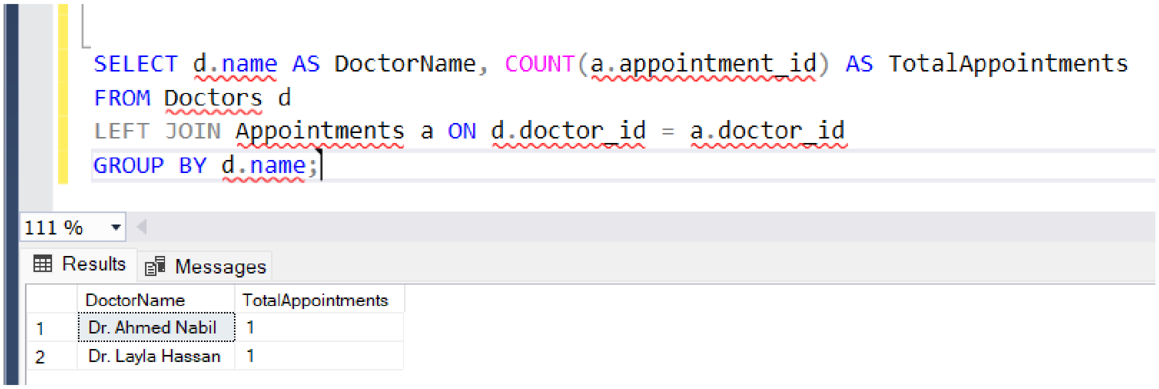
Query 2: Display all available rooms



```sql
SELECT * FROM Rooms WHERE availability = 1;
```

	room_id	room_number	room_type	availability
1	1	101	ICU	1
2	2	102	General	1
3	3	201	Private	1

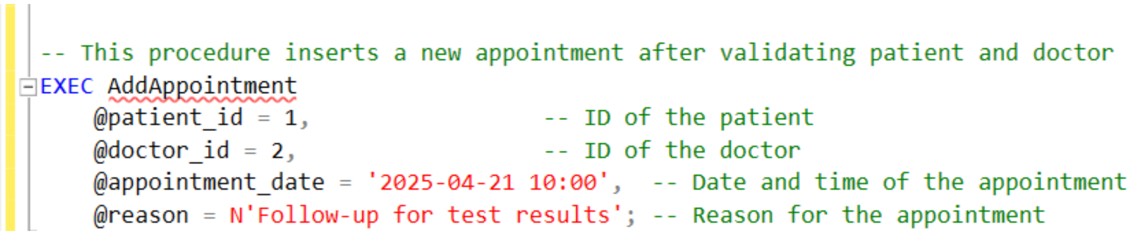
Query 3: Count number of appointments per doctor.



```sql
SELECT d.name AS DoctorName, COUNT(a.appointment_id) AS TotalAppointments
FROM Doctors d
LEFT JOIN Appointments a ON d.doctor_id = a.doctor_id
GROUP BY d.name;
```

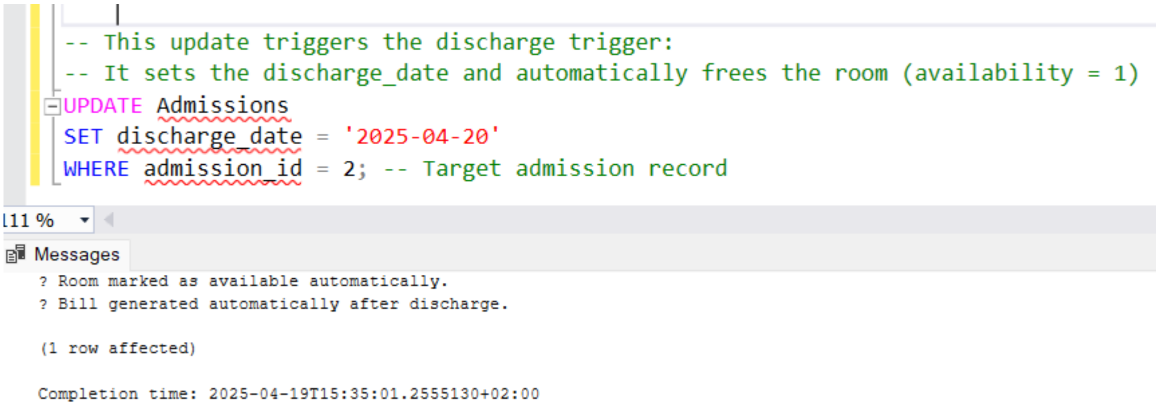
	DoctorName	TotalAppointments
1	Dr. Ahmed Nabil	1
2	Dr. Layla Hassan	1

Query 4: Execute the AddAppointment stored procedure.



```sql
-- This procedure inserts a new appointment after validating patient and doctor
EXEC AddAppointment
    @patient_id = 1,                    -- ID of the patient
    @doctor_id = 2,                     -- ID of the doctor
    @appointment_date = '2025-04-21 10:00',  -- Date and time of the appointment
    @reason = N'Follow-up for test results'; -- Reason for the appointment
```

Query 5: Trigger that fires on discharge to free the room and generate a bill.



```
-- This update triggers the discharge trigger:
-- It sets the discharge_date and automatically frees the room (availability = 1)
UPDATE Admissions
SET discharge_date = '2025-04-20'
WHERE admission_id = 2; -- Target admission record
```

```
? Room marked as available automatically.
? Bill generated automatically after discharge.

(1 row affected)

Completion time: 2025-04-19T15:35:01.2555130+02:00
```