



Presentation on the Project:
Database Mmaking of

Medical Management

A project for the course “Database Management System Sessional”

Course Code: CSE - 3114





COURSE INSTRUCTOR

DR. Shafkat Kibria

Assistant professor

Department of Computer Science & Engineering,
Leading University

PRESENTED BY

ID	NAME
0182220012101098	Pulock Deb Roy
0182220012101129	Fabliha Zafar Prima
0182220012101102	Delwar Hussin Sagor
0182220012101120	Suborna Deb

TEAM : 4

List of Contents

01. INTRODUCTION
02. USES & USERS OF Medical MANAGEMENT
03. ER DIAGRAM
04. TABLE SCHEMA
05. TABLE SCRIPTS & Their Implementation
06. SQL QUERIES
07. CONCLUSION



USES & USERS OF Medical MANAGEMENT

USES OF Medical MANAGEMENT:

1. Efficient Resource Allocation
2. Patient and Doctor Interaction
3. Reporting and Analytics
4. Billing and Payment Processing
5. Database Integrity and Security
6. Clinical Workflow and Operations

USER OF Medical MANAGEMENT:

1. Patient User
2. Doctor User
3. Admin User



ER DIAGRAM

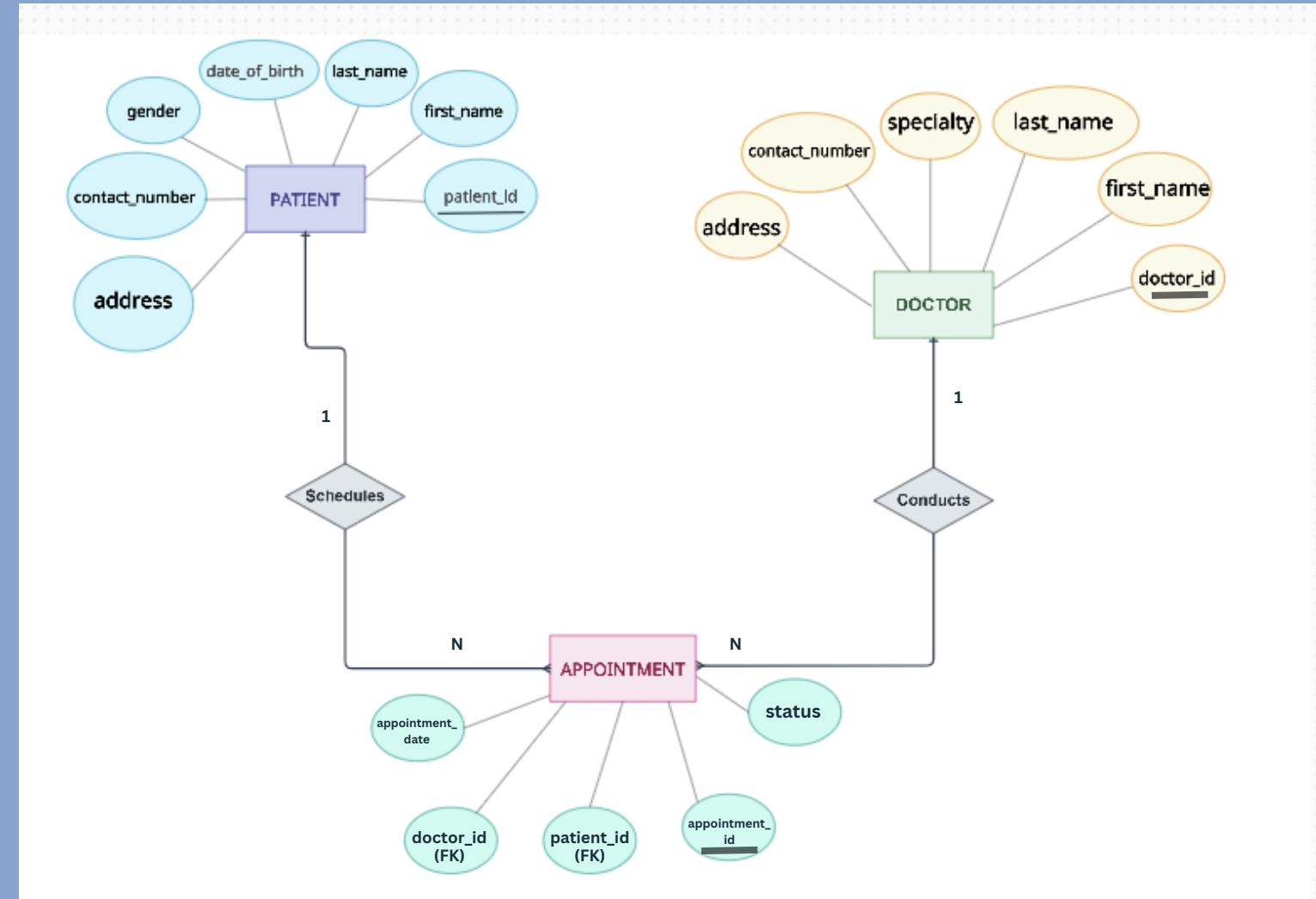


Table Schema

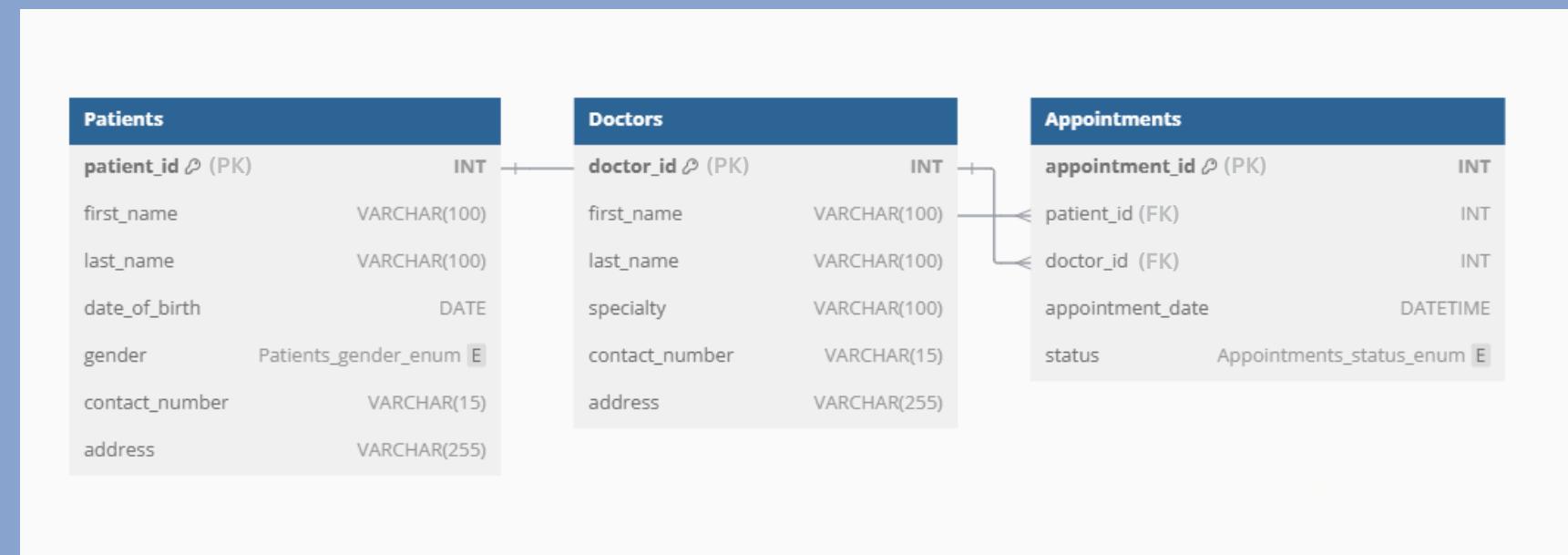


Table Scripts & Their Implementation

Table: 1

```
create database Medical_management;
CREATE TABLE Patients (
    patient_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    date_of_birth DATE,
    gender ENUM('Male', 'Female', 'Other'),
    contact_number VARCHAR(15),
    address VARCHAR(255)
);
INSERT INTO Patients (first_name, last_name, date_of_birth, gender, contact_number, address)
VALUES
    ('NurJahan','Nura','2003-06-20','Female','12345678','Sylhet, Bangladesh'),
    ('Fabliha', 'Prima', '2000-06-25', 'Female', '1234567890', 'Sylhet, Bangladesh'),
    ('Suborna','Deb','1971-07-26','Female','12345678','Sylhet, Bangladesh'),
    ('Delwar','Sagor','2010-07-27','Male','12345678','Sylhet, Bangladesh'),
    ('Pulock','Dip','2001-06-25','Male','12345678','Sylhet, Bangladesh');
select * from patients;
```

The screenshot shows the SQL Server Management Studio interface. The top pane displays the T-SQL script for creating the 'Patients' table and inserting data into it. The bottom pane shows the 'Result Grid' containing the inserted data.

patient_id	first_name	last_name	date_of_birth	gender	contact_number	address
2	NurJahan	Nura	2003-06-20	Female	12345678	Sylhet, Bangladesh
4	Pulock	Dip	2001-06-25	Male	12345678	Sylhet, Bangladesh
5	Fabliha	Prima	2000-06-25	Female	1234567890	Sylhet, Bangladesh
7	Suborna	Deb	1971-07-26	Female	12345678	Sylhet, Bangladesh
9	Delwar	Sagor	2010-07-27	Male	12345678	Sylhet, Bangladesh

Table Scripts & Their Implementation

Table: 2

```
CREATE TABLE Doctors (
    doctor_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    specialty VARCHAR(100),
    contact_number VARCHAR(15),
    address VARCHAR(255)
);

INSERT INTO Doctors (first_name, last_name, specialty, contact_number, address)
VALUES
('Dristy', 'Islam', 'Obstetrics', '0987654321', 'Mymensingh, Bangladesh'),
('Anisa', 'Sultana', 'Gynecology', '0987654321', 'Barishal, Bangladesh'),
('Neel', 'Islam', 'Cardiology', '0987654321', 'Dhaka, Bangladesh'),
('Samira', 'Begum', 'Neurology', '0987654321', 'Dhaka, Bangladesh'),
('Badar', 'Khan', 'Oncology', '0987654321', 'Sylhet, Bangladesh');

select * from doctors;
```

The screenshot shows the MySQL Workbench interface with two panes. The top pane displays the SQL code for creating the 'Doctors' table and inserting data. The bottom pane shows the resulting data in a grid.

doctor_id	first_name	last_name	specialty	contact_number	address
1	Anisa	Sultana	Gynecology	0987654321	Barishal, Bangladesh
2	Neel	Islam	Cardiology	0987654321	Dhaka, Bangladesh
3	Samira	Begum	Neurology	0987654321	Dhaka, Bangladesh
4	Badar	Khan	Oncology	0987654321	Sylhet, Bangladesh
5	Dristy	Islam	Obstetrics	0987654321	Mymensingh, Bangladesh

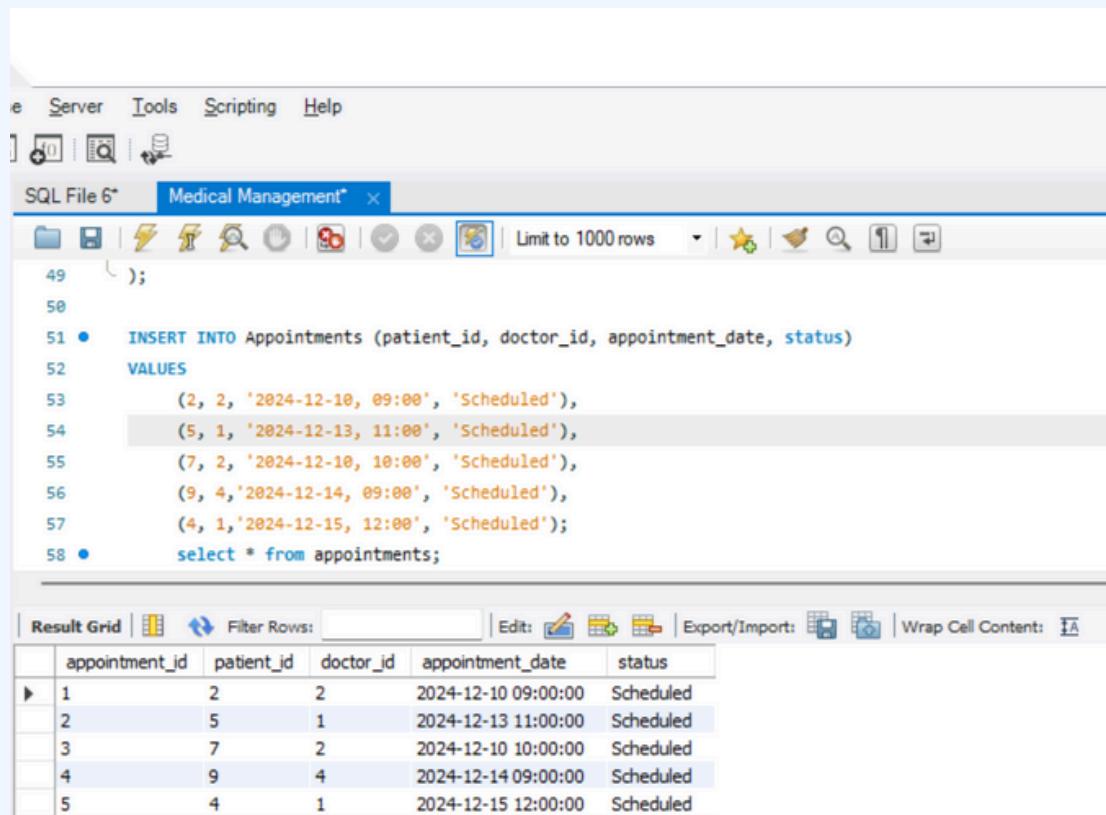
Table Scripts & Their Implementation

Table: 3

```
CREATE TABLE Appointments (
    appointment_id INT PRIMARY KEY AUTO_INCREMENT,
    patient_id INT,
    doctor_id INT,
    appointment_date DATETIME,
    status ENUM('Scheduled', 'Completed', 'Cancelled'),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

```
INSERT INTO Appointments (appointment_date, status)
VALUES
    ('2024-12-10, 09:00', 'Scheduled'),
    ('2024-12-13, 11:00', 'Scheduled'),
    ('2024-12-10, 10:00', 'Scheduled'),
    ('2024-12-14, 09:00', 'Scheduled'),
    ('2024-12-15, 12:00', 'Scheduled');

select * from appointments;
```



The screenshot shows a SQL Server Management Studio window with the following details:

- Scripting Tab:** The script pane displays the table creation and data insertion statements.
- Execution Results:** The results pane shows the output of the `select * from appointments;` query, resulting in the following table:

appointment_id	patient_id	doctor_id	appointment_date	status
1	2	2	2024-12-10 09:00:00	Scheduled
2	5	1	2024-12-13 11:00:00	Scheduled
3	7	2	2024-12-10 10:00:00	Scheduled
4	9	4	2024-12-14 09:00:00	Scheduled
5	4	1	2024-12-15 12:00:00	Scheduled

SQL QUERIES : 1

Q1: Write an SQL query to insert a new patient named "John Doe", born on "1985-05-15", male, with contact number "9876543210", and address "Dhaka, Bangladesh" into the Patients table.

```
INSERT INTO Patients (first_name, last_name, date_of_birth, gender, contact_number, address)
VALUES ('John', 'Doe', '1985-05-15', 'Male', '9876543210', 'Dhaka, Bangladesh');
```

	patient_id	first_name	last_name	date_of_birth	gender	contact_number	address
▶	2	NurJahan	Nura	2003-06-20	Female	12345678	Sylhet, Bangladesh
	4	Pulock	Dip	2001-06-25	Male	12345678	Sylhet, Bangladesh
	5	Fabiha	Prima	2000-06-25	Female	1234567890	Sylhet, Bangladesh
	7	Suborna	Deb	1971-07-26	Female	12345678	Sylhet, Bangladesh
	9	Delwar	Sagor	2010-07-27	Male	12345678	Sylhet, Bangladesh
	10	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh
	11	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh
	12	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh

SQL QUERIES : 2

Q2: Write an SQL query to update the contact number of the patient with patient_id 12 to "9876543210".

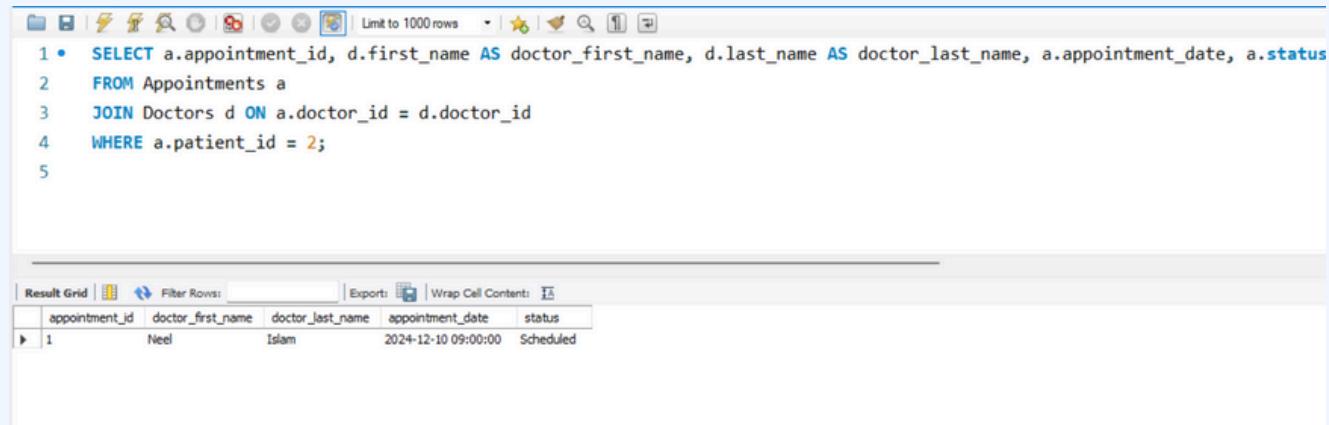
```
UPDATE Patients  
SET contact_number = '9876543210'  
WHERE patient_id = 12;
```

patient_id	first_name	last_name	date_of_birth	gender	contact_number	address
2	NurJahan	Nura	2003-06-20	Female	12345678	Sylhet, Bangladesh
4	Pulock	Dip	2001-06-25	Male	12345678	Sylhet, Bangladesh
5	Fabiha	Prima	2000-06-25	Female	1234567890	Sylhet, Bangladesh
7	Suborna	Deb	1971-07-26	Female	12345678	Sylhet, Bangladesh
9	Delwar	Sagor	2010-07-27	Male	12345678	Sylhet, Bangladesh
10	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh
11	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh
12	John	Doe	1985-05-15	Male	9876543210	Dhaka, Bangladesh

SQL QUERIES : 3

Q3: Write an SQL query to select the appointment_id, first_name and last_name of the doctor, appointment_date, and status for all appointments where the patient_id is 2. Use a JOIN to combine Appointments and Doctors.

```
SELECT a.appointment_id, d.first_name AS doctor_first_name, d.last_name AS doctor_last_name,  
       a.appointment_date, a.status  
  FROM Appointments a  
 JOIN Doctors d ON a.doctor_id = d.doctor_id  
 WHERE a.patient_id = 2;
```



The screenshot shows the MySQL Workbench interface. The SQL editor window contains the following code:

```
1 •  SELECT a.appointment_id, d.first_name AS doctor_first_name, d.last_name AS doctor_last_name, a.appointment_date, a.status  
2   FROM Appointments a  
3   JOIN Doctors d ON a.doctor_id = d.doctor_id  
4  WHERE a.patient_id = 2;  
5
```

Below the editor is the Result Grid window, which displays the query results:

	appointment_id	doctor_first_name	doctor_last_name	appointment_date	status
▶	1	Neel	Islam	2024-12-10 09:00:00	Scheduled

SQL QUERIES : 4

Q4: Write an SQL query to select the appointment_id, first_name and last_name of the patient, appointment_date, and status for all appointments where the doctor_id is 1.

```
SELECT a.appointment_id, p.first_name AS patient_first_name, p.last_name AS patient_last_name,  
       a.appointment_date, a.status  
  FROM Appointments a  
 JOIN Patients p ON a.patient_id = p.patient_id  
 WHERE a.doctor_id = 1;
```

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** The query is displayed in the SQL editor window under the tab "SQL File 6".
- Result Grid:** Below the editor, the result grid displays the following data:

	appointment_id	patient_first_name	patient_last_name	appointment_date	status
▶	2	Fabila	Prima	2024-12-13 11:00:00	Scheduled
	5	Pulock	Dip	2024-12-15 12:00:00	Scheduled

SQL QUERIES : 5

Q5: Write an SQL query to select the first_name, last_name, and date_of_birth of all patients in the Patients table, sorted by date_of_birth in ascending order.

```
SELECT first_name, last_name, date_of_birth  
      FROM Patients  
 ORDER BY date_of_birth ASC;
```

The screenshot shows an SQL IDE interface with the following details:

- SQL File 6***: The current file tab.
- Medical Management***: The database connection name.
- Query Editor Content:**

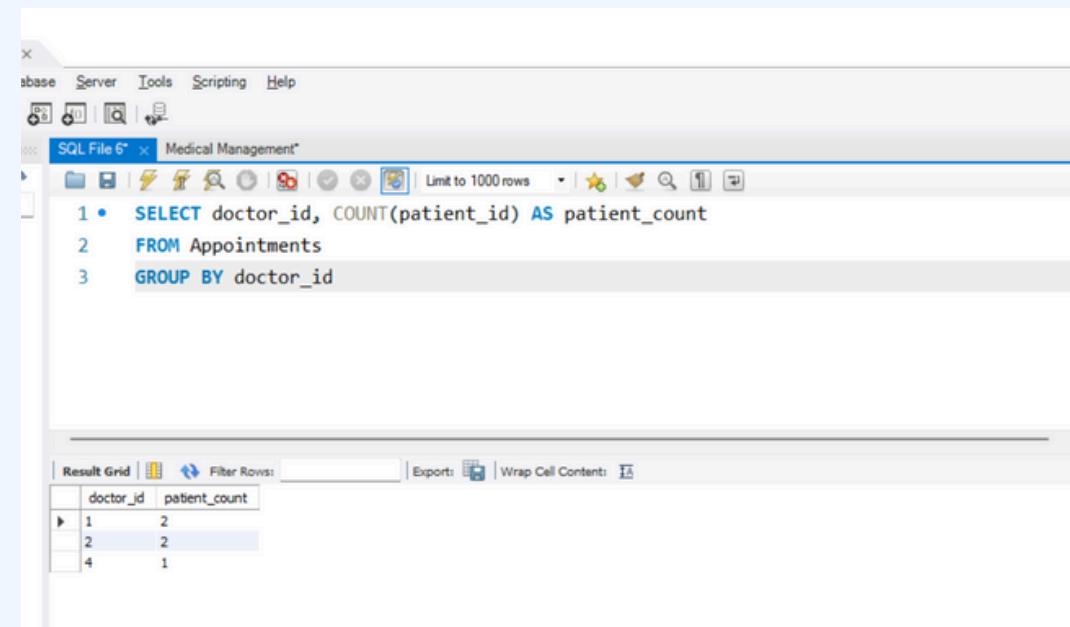
```
1 •  SELECT first_name, last_name, date_of_birth  
2   FROM Patients  
3   ORDER BY date_of_birth ASC;  
4
```
- Result Grid:** A table displaying the query results.
- Table Headers:** first_name, last_name, date_of_birth.
- Table Data:**

	first_name	last_name	date_of_birth
▶	Suborna	Deb	1971-07-26
	John	Doe	1985-05-15
	John	Doe	1985-05-15
	John	Doe	1985-05-15
	Fabila	Prima	2000-06-25
	Pulok	Dip	2001-06-25
	NurJahan	Nura	2003-06-20
	Delwar	Sagor	2010-07-27
- Result Grid Buttons:** Filter Rows, Export, Wrap Cell Content.
- Patients 16**: The current result set tab.

SQL QUERIES : 6

Q6: Write an SQL query to count the number of patients assigned to each doctor. The result should display the doctor_id and patient_count

```
SELECT doctor_id, COUNT(patient_id) AS patient_count  
      FROM Appointments  
     GROUP BY doctor_id
```



The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
1 •  SELECT doctor_id, COUNT(patient_id) AS patient_count  
2   FROM Appointments  
3   GROUP BY doctor_id
```

The results are displayed in a Result Grid:

doctor_id	patient_count
1	2
2	2
4	1

SQL QUERIES : 7

Q7: Write an SQL query to retrieve all appointments where the appointment_date is scheduled after "2024-12-12".

```
SELECT appointment_id, patient_id, doctor_id, appointment_date, status  
      FROM Appointments  
     WHERE appointment_date > '2024-12-12';
```

The screenshot shows the MySQL Workbench interface. The top window is titled "SQL File 6* | Medical Management*". It contains the following SQL code:

```
1 •  SELECT appointment_id, patient_id, doctor_id, appointment_date, status  
2   FROM Appointments  
3  WHERE appointment_date > '2024-12-12';  
4
```

Below the code is a "Result Grid" table with the following data:

	appointment_id	patient_id	doctor_id	appointment_date	status
▶	2	5	1	2024-12-13 11:00:00	Scheduled
	4	9	4	2024-12-14 09:00:00	Scheduled
	5	4	1	2024-12-15 12:00:00	Scheduled

SQL QUERIES : 8

Q8: Write an SQL query to find the number of appointments scheduled for each status ('Scheduled','Canceled'). The result should display the status and the count of appointments for each status.

```
SELECT status, COUNT(appointment_id) AS appointment_count  
      FROM Appointments  
      GROUP BY status;
```

The screenshot shows the SQL Server Management Studio interface. The query window contains the following SQL code:

```
1 •  SELECT status, COUNT(appointment_id) AS appointment_count  
2   FROM Appointments  
3   GROUP BY status;
```

The results pane displays the following table:

status	appointment_count
Scheduled	5

Conclusion



The SQL code successfully sets up a medical management system with Patients, Doctors, and Appointments tables

Thank you for your attention

