

**SECD2523-DATABASE**

**20242025 - SEMESTER 1**

**PHASE 5**

**FACULTY OF MJIIT**

|  |  |
| --- | --- |
| **NAME** | **MATRIC ID** |
| **Liu Ruoyang** | **A23MJ4022** |
| **Pranto Anik Islam** | **A23MJ4024** |
| **Kahlan Sultan** | **A23MJ4021** |
| **Bu Guoshun** | **A23MJ4019** |

**Relational Database Schemas**

**1. Patient Table**

This table stores the basic information of patients.

* **Attributes**:
  + PatientID (Primary Key): Unique identifier for each patient.
  + FullName: Name of the patient.
  + DateOfBirth: Date of birth of the patient.
  + EmailAddress: Email address of the patient.
* **Relationships**:
  + Referenced by PatientContact, Visitor, Bill, LabTestInstances, Appointment, and Treatment tables.

**2. PatientContact Table**

This table manages contact information for patients.

* **Attributes**:
  + PatientID (Primary Key, Foreign Key): Links to Patient(PatientID).
  + ContactNumber: Contact number of the patient.
* **Relationships**:
  + One-to-One with the Patient table.

**3. Visitor Table**

This table records visitor information associated with patients.

* **Attributes**:
  + VisitorID: Unique identifier for the visitor.
  + PatientID (Foreign Key): Links to Patient(PatientID).
  + Relationship: Relationship of the visitor to the patient.
* **Relationships**:
  + Many-to-Many relationship between visitors and patients.

**4. Bill Table**

This table handles billing information for patients.

* **Attributes**:
  + BillID (Primary Key): Unique identifier for each bill.
  + Amount: The bill amount.
  + PaymentStatus: Status of the payment (e.g., Paid, Pending).
  + PatientID (Foreign Key): Links to Patient(PatientID).
* **Relationships**:
  + One-to-Many with the Patient table.

**5. LabTest Table**

This table contains information about the types of lab tests available.

* **Attributes**:
  + TestID (Primary Key): Unique identifier for the test.
  + TestName: Name of the test.
* **Relationships**:
  + Referenced by LabTestInfo and LabTestInstances.

**6. LabTestInfo Table**

This table provides additional information about lab tests.

* **Attributes**:
  + LabID (Primary Key): Unique identifier for the lab test info.
  + TestID (Foreign Key): Links to LabTest(TestID).
* **Relationships**:
  + One-to-Many relationship with the LabTest table.

**7. LabTestInstances Table**

This table records individual lab tests conducted for patients.

* **Attributes**:
  + TestID (Primary Key, Foreign Key): Links to LabTest(TestID).
  + PatientID (Primary Key, Foreign Key): Links to Patient(PatientID).
  + DateConducted: The date the lab test was conducted.
* **Relationships**:
  + Many-to-Many relationship between LabTest and Patient.

**8. Department Table**

This table manages information about hospital departments.

* **Attributes**:
  + DepartmentID (Primary Key): Unique identifier for the department.
  + DepartmentName: Name of the department.
  + Location: Location of the department.
* **Relationships**:
  + Referenced by the Staff table.

**9. Staff Table**

This table stores information about hospital staff.

* **Attributes**:
  + StaffID (Primary Key): Unique identifier for each staff member.
  + FullName: Name of the staff member.
  + ContactNumber: Contact number of the staff member.
  + DepartmentID (Foreign Key): Links to Department(DepartmentID).
* **Relationships**:
  + One-to-Many relationship with the Department table.
  + Referenced by Doctor and Nurse tables.

**10. Doctor Table**

This table records information about doctors.

* **Attributes**:
  + DoctorID (Primary Key): Unique identifier for each doctor.
  + Specialty: Specialization of the doctor.
  + StaffID (Unique, Foreign Key): Links to Staff(StaffID).
* **Relationships**:
  + One-to-One with the Staff table.
  + Referenced by Appointment and Treatment tables.

**11. Nurse Table**

This table records information about nurses.

* **Attributes**:
  + NurseID (Primary Key): Unique identifier for each nurse.
  + StaffID (Unique, Foreign Key): Links to Staff(StaffID).
* **Relationships**:
  + One-to-One with the Staff table.

**12. Appointment Table**

This table handles patient appointments with doctors.

* **Attributes**:
  + AppointmentID (Primary Key): Unique identifier for each appointment.
  + DateTime: Date and time of the appointment.
  + Status: Status of the appointment (e.g., Scheduled, Completed).
  + PatientID (Foreign Key): Links to Patient(PatientID).
  + DoctorID (Foreign Key): Links to Doctor(DoctorID).
* **Relationships**:
  + Many-to-Many relationship between Patient and Doctor.

**13. Treatment Table**

This table records treatments provided to patients.

* **Attributes**:
  + TreatmentID (Primary Key): Unique identifier for each treatment.
  + TreatmentDescription: Description of the treatment.
  + DateOfTreatment: Date the treatment was provided.
  + PatientID (Foreign Key): Links to Patient(PatientID).
  + DoctorID (Foreign Key): Links to Doctor(DoctorID).
* **Relationships**:
  + One-to-Many with Patient and Doctor tables.
  + Referenced by the TreatMedication table.

**14. Medication Table**

This table manages information about medications.

* **Attributes**:
  + MedicationID (Primary Key): Unique identifier for each medication.
  + Name: Name of the medication.
  + Dosage: Dosage details for the medication.
* **Relationships**:
  + Referenced by the TreatMedication table.

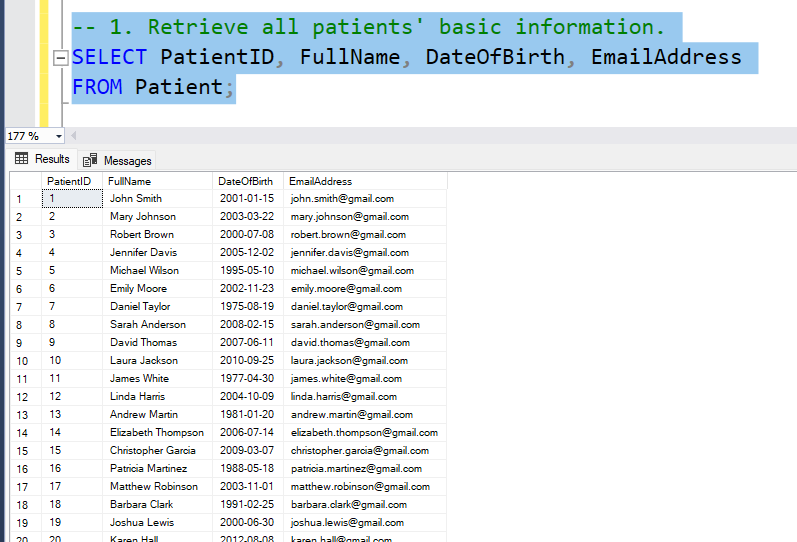
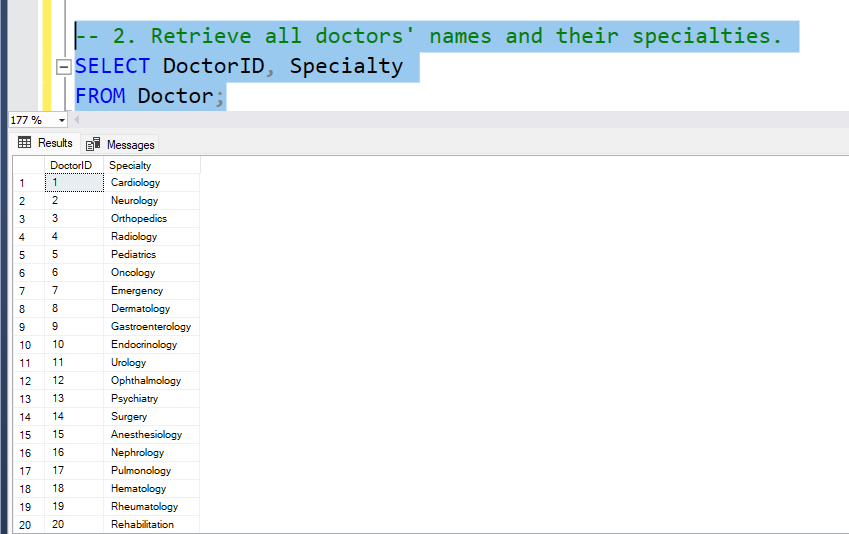
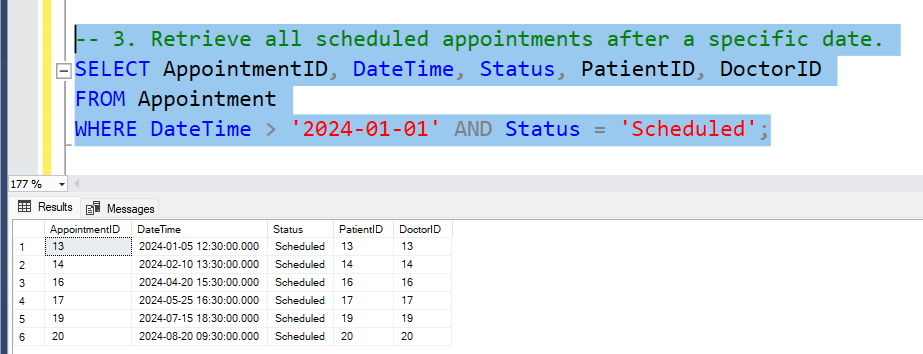
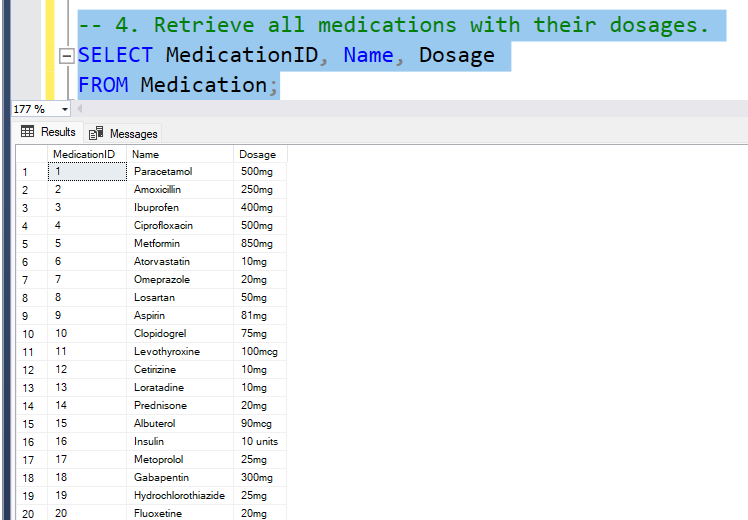
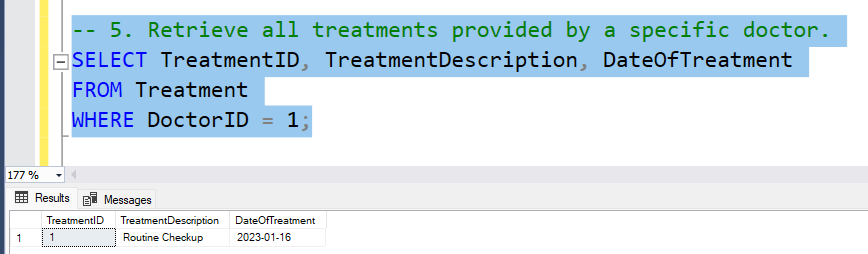
**15. TreatMedication Table**

This table represents the relationship between treatments and medications.

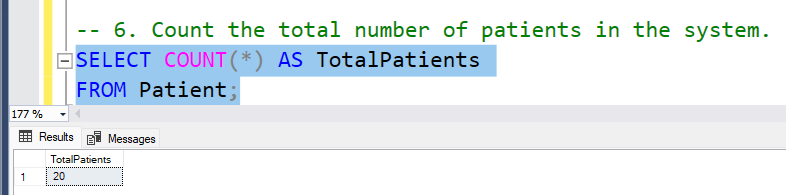
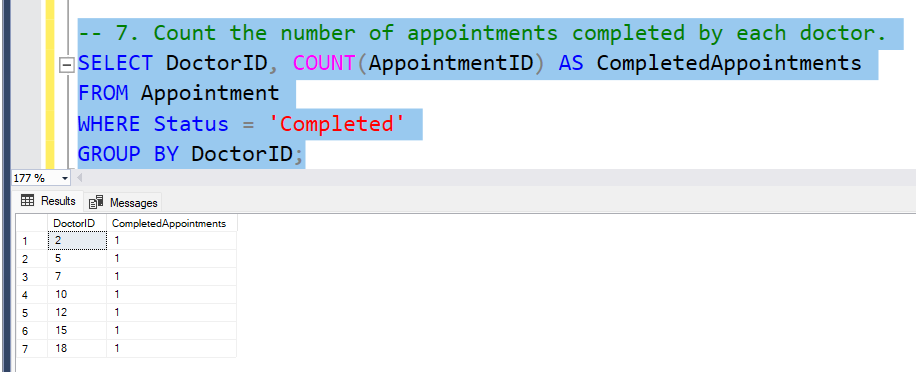
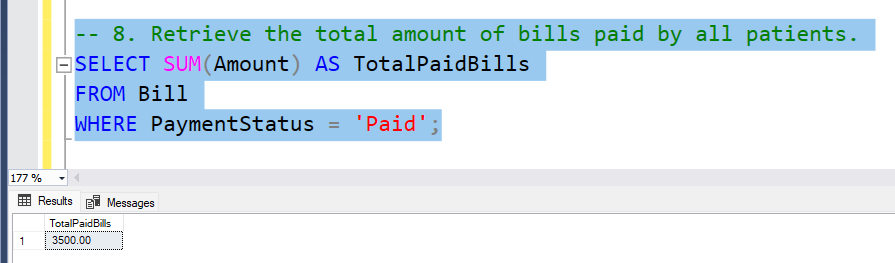
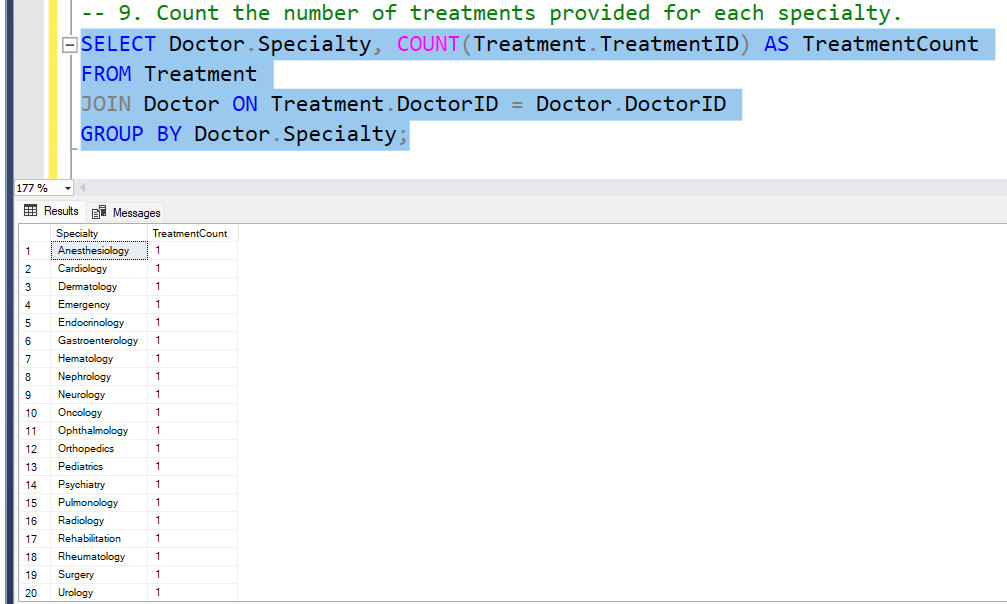
* **Attributes**:
  + TreatmentID (Primary Key, Foreign Key): Links to Treatment(TreatmentID).
  + MedicationID (Primary Key, Foreign Key): Links to Medication(MedicationID).
* **Relationships**:
  + Many-to-Many relationship between Treatment and Medication.

1. **Queries with Detailed Explanations**

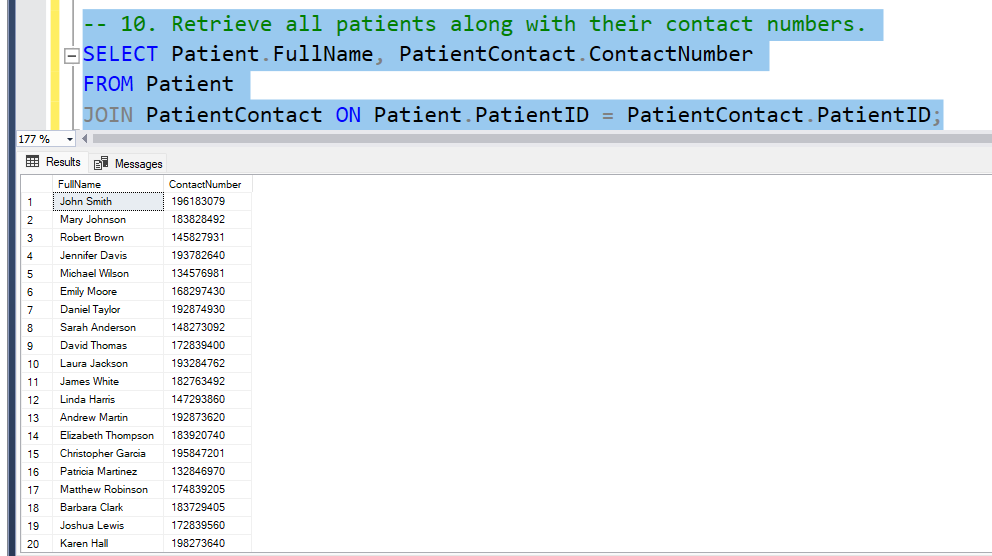
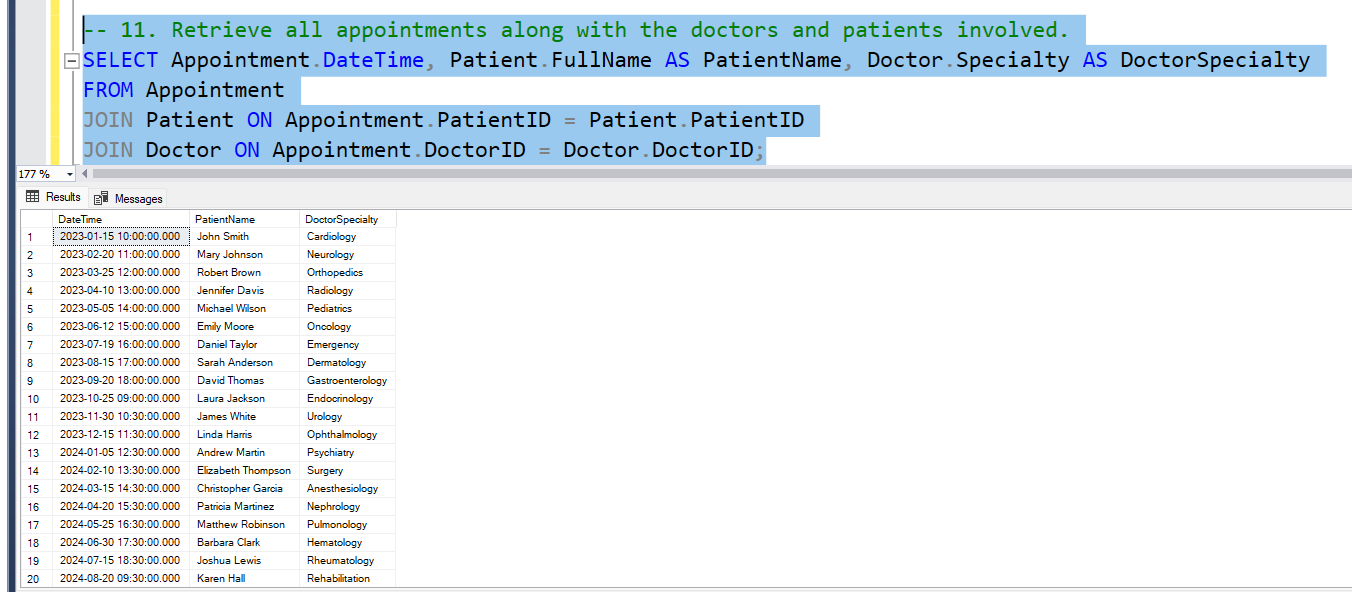
**Basic Queries**

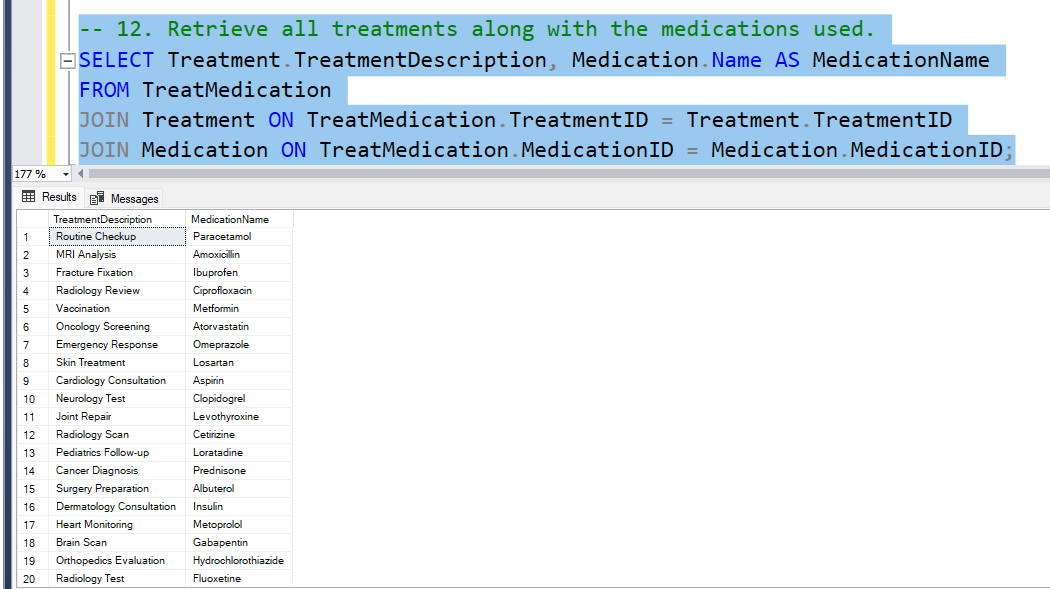
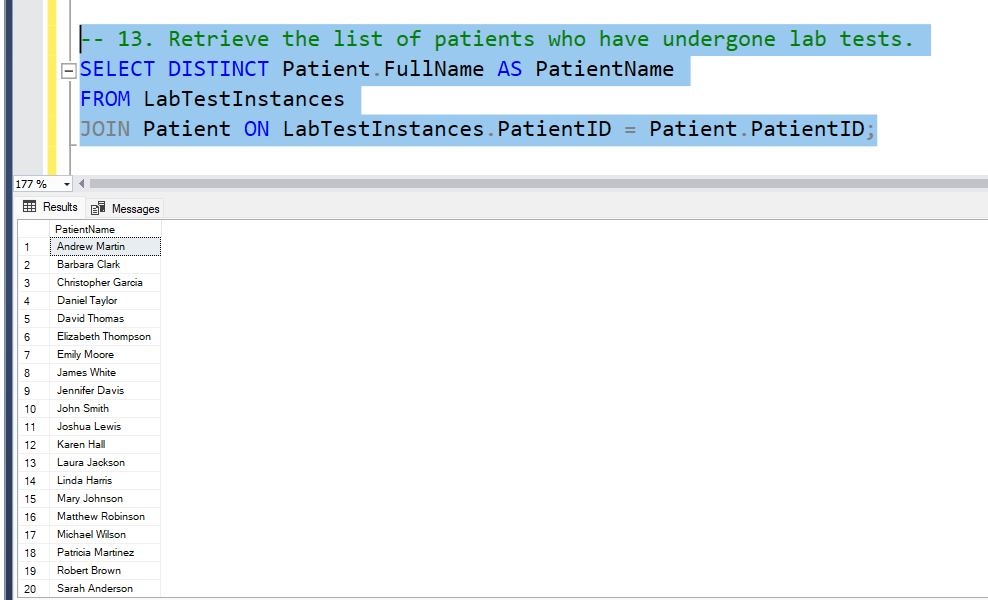
1. **Retrieve all patients' basic information.**
   * **Function**: Displays basic details for all patients, including ID, name, date of birth, and email address.
   * **Purpose**: Provides a quick overview of patient records.
2. **Retrieve all doctors' names and their specialties.**
   * **Function**: Shows the IDs and specialties of all doctors.
   * **Purpose**: Understands the distribution of medical specialties in the hospital.
3. **Retrieve all scheduled appointments after a specific date.**
   * **Function**: Lists appointments scheduled after a specific date, including appointment ID, time, status, patient, and doctor details.
   * **Purpose**: Tracks upcoming appointments.
4. **Retrieve all medications with their dosages.**
   * **Function**: Lists all medications and their dosages.
   * **Purpose**: Displays a catalog of available medications.
5. **Retrieve all treatments provided by a specific doctor.**
   * **Function**: Lists treatments conducted by a specific doctor (DoctorID = 1), including treatment description and date.
   * **Purpose**: Tracks the treatment history of an individual doctor.

**Aggregate and Statistical Queries**

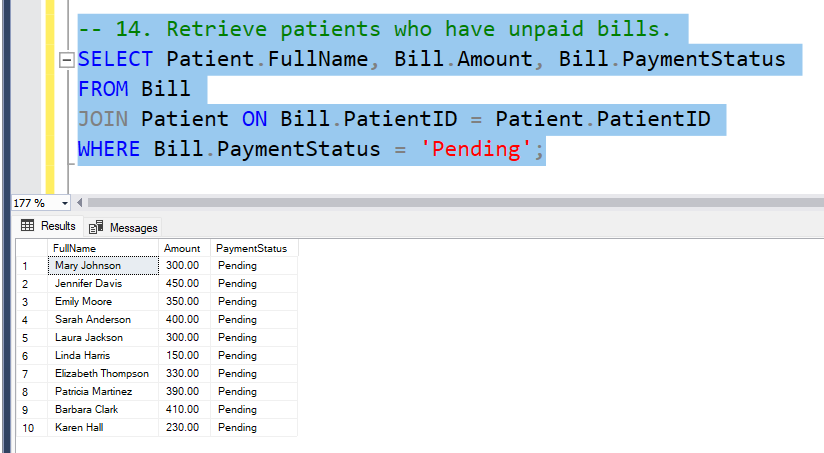
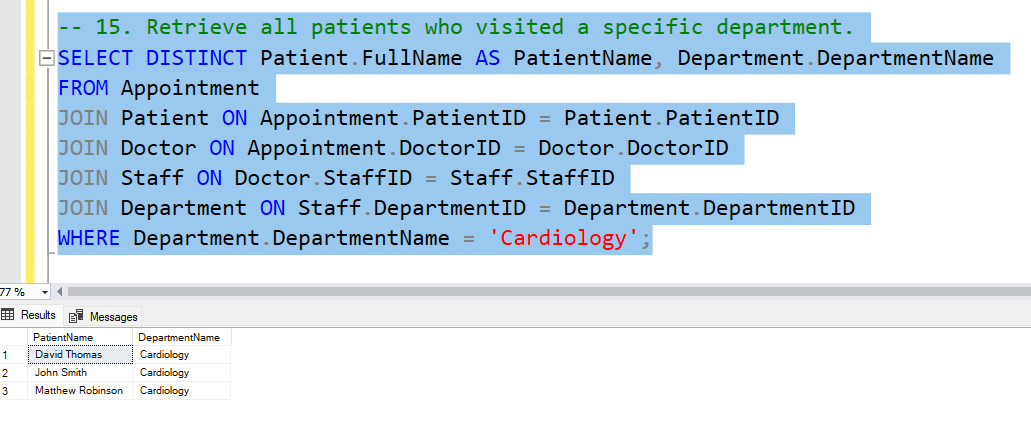
1. **Count the total number of patients in the system.**
   * **Function**: Counts the total number of patients in the hospital system.
   * **Purpose**: Provides an overall patient count.
2. **Count the number of appointments completed by each doctor.**
   * **Function**: Counts the number of appointments completed by each doctor.
   * **Purpose**: Analyzes doctor workload and completion rates.
3. **Retrieve the total amount of bills paid by all patients.**
   * **Function**: Calculates the total amount of all paid bills.
   * **Purpose**: Tracks total hospital revenue from paid bills.
4. **Count the number of treatments provided for each specialty.**
   * **Function**: Counts the treatments provided, grouped by doctor specialty.
   * **Purpose**: Evaluates the workload of each department or specialty.

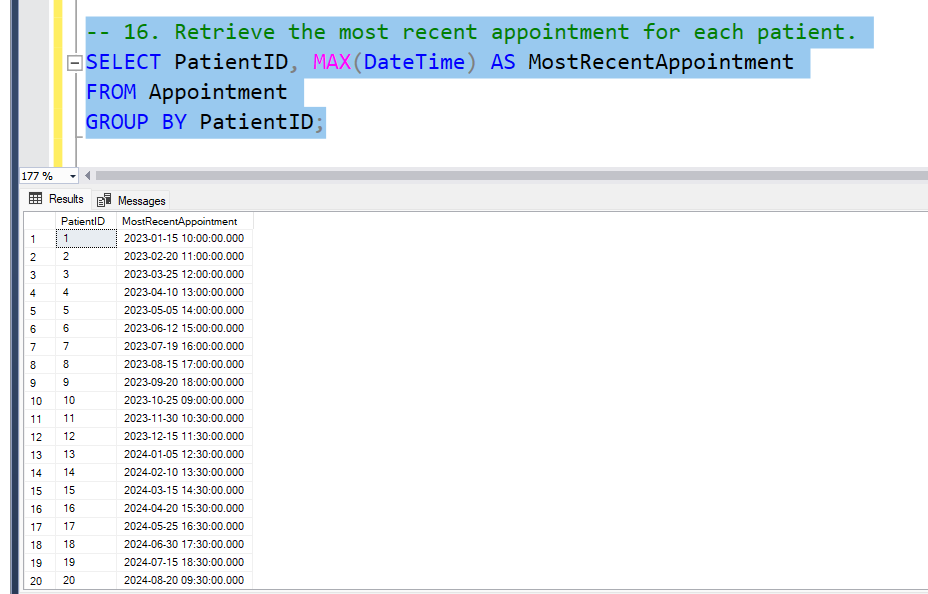
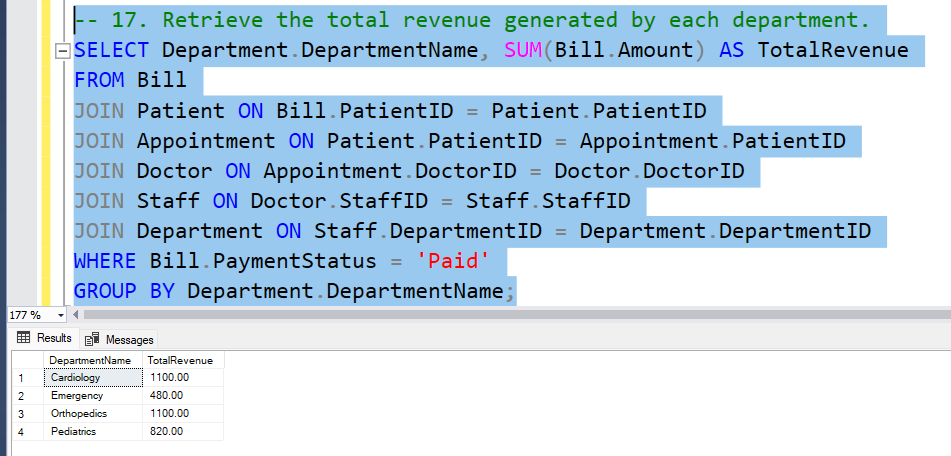
**Join Queries**

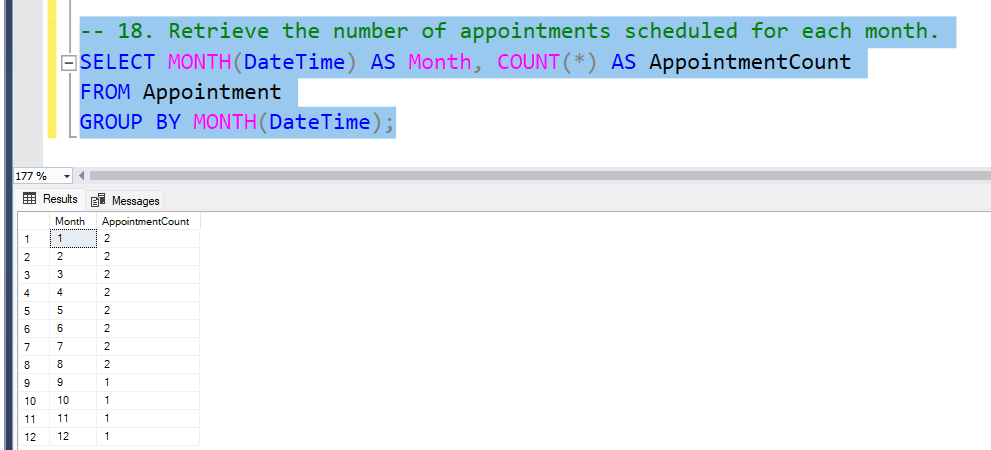
1. **Retrieve all patients along with their contact numbers.**
   * **Function**: Lists all patients with their contact numbers.
   * **Purpose**: Quickly accesses patient contact information.
2. **Retrieve all appointments along with the doctors and patients involved.**
   * **Function**: Shows the time of appointments, patient names, and doctor specialties.
   * **Purpose**: Visualizes appointment participants and schedules.

1. **Retrieve all treatments along with the medications used.**
   * **Function**: Lists treatments with the medications used for each treatment.
   * **Purpose**: Tracks the relationship between treatments and medications.
2. **Retrieve the list of patients who have undergone lab tests.**
   * **Function**: Displays a unique list of patients who have completed lab tests.
   * **Purpose**: Understands the coverage of lab testing among patients.

**Conditional and Advanced Queries**

1. **Retrieve patients who have unpaid bills.**
   * **Function**: Lists patients with unpaid bills, including the bill amount and status.
   * **Purpose**: Identifies overdue payments for follow-up.
2. **Retrieve all patients who visited a specific department.**
   * **Function**: Lists patients who visited a specific department (e.g., Cardiology).
   * **Purpose**: Tracks patient volume for a specific department.

1. **Retrieve the most recent appointment for each patient.**
   * **Function**: Displays the most recent appointment time for each patient.
   * **Purpose**: Tracks the latest interactions for each patient.
2. **Retrieve the total revenue generated by each department.**
   * **Function**: Calculates the total revenue for each department based on paid bills.
   * **Purpose**: Evaluates the financial contribution of each department.

1. **Retrieve the number of appointments scheduled for each month.**
   * **Function**: Counts the number of appointments grouped by month.
   * **Purpose**: Analyzes monthly trends in appointment scheduling.

**3.User Manual for Hospital Management System**

**Introduction**

This application is a simple Hospital Management System designed for managing patient data efficiently. It provides a graphical user interface (GUI) for performing basic database operations such as adding, updating, deleting, and viewing patient records.

**Main Features**

1. **Add Patient**: Add a new patient record to the database.
2. **Update Email:** Update the Email Address of an existing patient.
3. **Delete Patient**: Delete a patient record from the database.
4. **View Patients**: View all patient records or search for a specific patient by ID.

**How to Use the Application**

**1. Add Patient**图形用户界面, 应用程序

描述已自动生成

* **Steps to Add a Patient**:
  1. Click on the **"Add Patient"** button at the top of the screen.
  2. Fill in the required details in the form:
     + **Patient ID**: A unique identifier for the patient.
     + **Full Name**: The full name of the patient.
     + **Date of Birth**: Enter the date in YYYY-MM-DD format.
     + **Email Address**: Enter the patient's email address.
  3. Click the **"Add Patient"** button below the textboxes to save the record.
  4. If successful, a confirmation message will appear.

**2. Update Email**图形用户界面, 文本, 应用程序

描述已自动生成

* **Steps to Update a Patient's Email**:
  1. Click on the **"Update Email"** button at the top of the screen.
  2. Enter the following details:
     + **Patient ID**: The ID of the patient whose contact details need to be updated.
     + **New Email Address**: Enter the updated email address.
  3. Click the **"Update Email"** button below the textboxes.
  4. A confirmation message will appear if the update is successful.

**3. Delete Patient**图形用户界面, 文本, 应用程序

描述已自动生成

* **Steps to Delete a Patient**:
  1. Click on the **"Delete Patient"** button at the top of the screen.
  2. Enter the **Patient ID** of the record to be deleted.
  3. Click the **"Delete Patient"** button below the textbox.
  4. A confirmation message will appear if the deletion is successful.

表格

描述已自动生成**4. View Patients**

* **Steps to View All Patients**:
  1. Leave the textbox blank and click on the **"View Patients"** button below the textbox.
  2. All patient records will be displayed in a table format.
  3. Each record includes:
     + **ID**: Patient ID
     + **Name**: Patient's Name
     + **Date of Birth**: Patient's Date of Birth
     + **Email**: Patient's Email Address
* 图形用户界面, 文本, 应用程序

  描述已自动生成**Steps to Search for a Specific Patient**:
  1. Enter the **Patient ID** in the search field.
  2. Click the **"View Patients"** button.
  3. The record of the specific patient will be displayed.

图形用户界面, 表格

描述已自动生成

**Error Handling**

1. **Missing Required Fields**:
   * If any required field is left blank, an error message will be displayed.
   * Ensure all fields are filled before submitting the form.
2. **Invalid Input**:
   * Ensure the **Patient ID** is unique and matches the database record when updating or deleting.
   * Use the correct format for the **Date of Birth** (YYYY-MM-DD) and valid phone numbers/email addresses.
3. **Database Connection Error**:
   * If the application cannot connect to the database, check the server connection and retry.