DATABASE PROGRAMMING FINAL PROJECT USING STRUCTURED QUERY LANGUAGE (SQL)

HOSPITAL MANAGEMENT SYSTEM DATABASE

TABLE OF CONTENTS:

STORYLINE:

At **Tyrell Health Centre**, the management team faces increasing challenges in efficiently coordinating and tracking patient care across a growing number of specializations and cases. The hospital serves patients with diverse medical needs, ranging from routine check-ups to the management of complex chronic conditions. With specialists like Dr. Boateng (Cardiologist), Dr. Armstrong (General Practitioner), Dr. Patel (Endocrinologist), and Dr. Taylor (Neurologist), the hospital provides comprehensive care, addressing heart conditions, hypertension, neurological disorders, and diabetes.

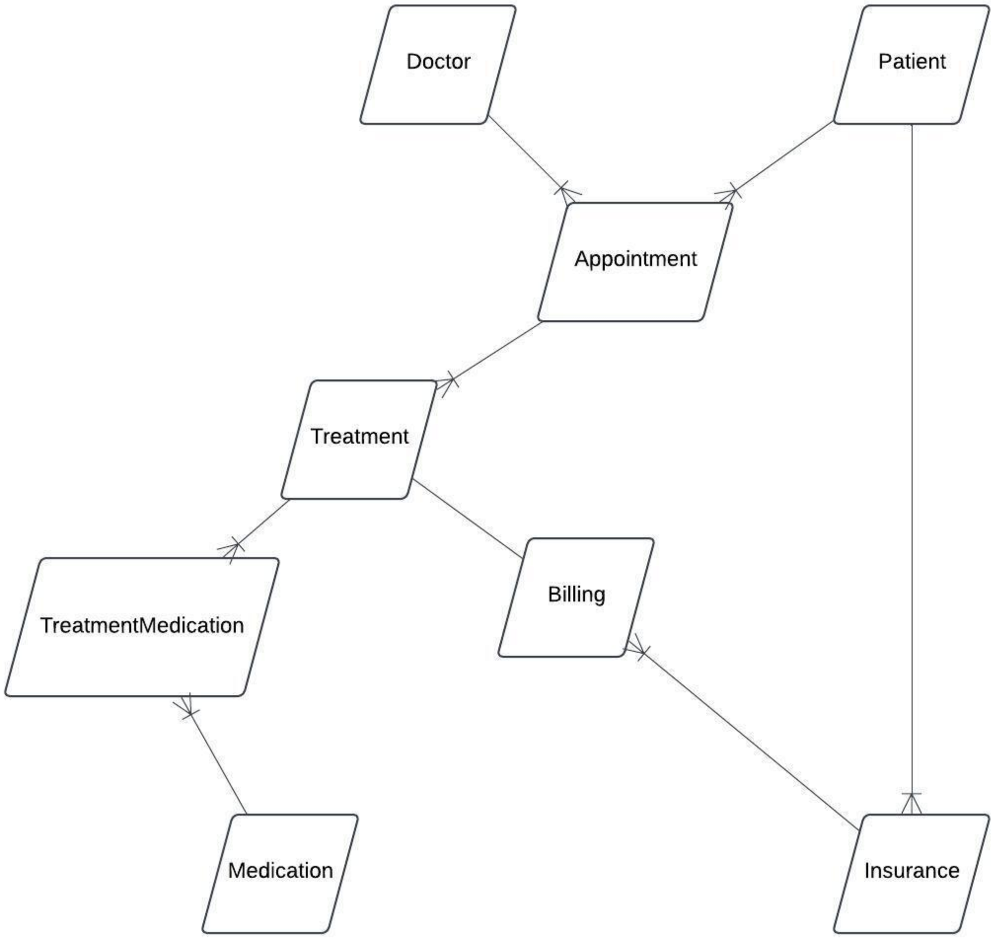
Patients like Mr. Daniel Timmis, who has ongoing heart and skin conditions, rely on regular follow-ups with multiple doctors to monitor their recovery and manage their health effectively. Similarly, Ms. Alice Monroe, diagnosed with diabetes and hypertension, requires consultations with Dr. Patel and Dr. Boateng to receive tailored care. Young patients, such as Ella Baker, depend on pediatrician Dr. Johnson for conditions like asthma, which necessitate ongoing treatment and medication management. Mark Johnson recently consulted Dr. Taylor for severe migraines, receiving an MRI and a prescription for Sumatriptan to alleviate his condition.

Each patient schedules multiple appointments with their doctors, often requiring treatments like allergy tests, blood sugar monitoring, MRI scans, or vaccinations. Alongside these treatments, medications play a crucial role in patient recovery. Patients are prescribed drugs such as Metformin for diabetes, Fluticasone for asthma, and Propranolol for hypertension. The hospital's system ensures that every medication is correctly linked to the associated treatment, minimizing risks of prescription overlaps or contradictions.

Billing and insurance claims are another vital aspect of Tyrell Health Centre’s operations. Insurance providers like HealthPlus, MediSafe, and KidsCare assist patients in covering substantial portions of their medical expenses. For instance, Mr. Timmis relies on HealthPlus to cover 80% of his cardiology-related expenses, while Ella Baker benefits from comprehensive pediatric coverage under KidsCare. Accurate billing and coordination with insurance policies are critical to ensuring that patients are charged correctly and that claims are processed efficiently.

The hospital's operations are further supported by its robust database system, which securely integrates appointment scheduling, treatment tracking, medication management, billing, and insurance handling. This system enables doctors and staff to access relevant patient information seamlessly while maintaining data security through role-based restrictions. By streamlining administrative processes, the system allows Tyrell Health Centre to focus on delivering high-quality care and improving patient outcomes.

ENTITY – RELATION DIAGRAM



PROBLEM REQUIREMENTS

To address the problem:

1. Doctor – Patient Coordination:
    - A way to schedule and manage appointments between doctors and patients.
    - Each doctor should be able to access their appointment schedules and patient histories to provide targeted care.
2. Treatment and Follow-Up Tracking:
    - Treatments should be documented per appointment, detailing the procedures and care provided during each visit.
    - A link between treatments and appointments so doctors can assess patient history effectively.
3. Medication Management:
    - For treatments that require medication, there should be a system to record prescriptions and link them to specific treatments.
    - A comprehensive list of medications to prevent any overlap or contradictions.
4. Billing and Insurance:
    - Billing should be tied to treatment, allowing for clear and accurate charges.
    - Insurance information should be linked to billing to facilitate claim management and ensure that patients like Mr. Timmis are billed accurately.
5. Data Access and Security:
    - Doctors and clinic staff should have access to relevant patient records, but there should be restrictions based on roles to protect sensitive data.

FUNCTIONAL DEPENDENCIES:

1. Doctors Table

- **Functional Dependencies**:
  - DoctorID → Name, Specialization, ContactInfo
  - DoctorID uniquely identifies each doctor, determining attributes such as Name, Specialization, and ContactInfo.

2. Patients Table

- **Functional Dependencies**:
  - PatientID → Name, DateOfBirth, ContactInfo
  - PatientID uniquely identifies each patient and determines attributes such as Name, DateOfBirth, and ContactInfo.

3. Appointments Table

- **Functional Dependencies**:
  - AppointmentID → PatientID, DoctorID, Date, Time
  - PatientID, DoctorID, Date, Time → AppointmentID
  - AppointmentID uniquely determines details of each appointment, while the combination of PatientID, DoctorID, Date, and Time determines the AppointmentID.

4. Treatments Table

- **Functional Dependencies**:
  - TreatmentID → PatientID, DoctorID, Diagnosis, Procedure, Date
  - PatientID, DoctorID, Date → TreatmentID
  - TreatmentID uniquely determines the treatment details, and a combination of PatientID, DoctorID, and Date can also uniquely identify a treatment record.

5. Billing Table

- **Functional Dependencies**:
  - BillID → PatientID, Amount, Date, Status
  - PatientID, Date → BillID
  - BillID uniquely identifies a billing record and determines attributes like Amount, Date, and Status. Additionally, a patient's bill for a specific date could uniquely identify a bill record.

6. Insurance Table

- **Functional Dependencies**:
  - InsuranceID → PatientID, Provider, PolicyNumber, CoverageDetails
  - PatientID, PolicyNumber → InsuranceID
  - InsuranceID uniquely determines insurance details, while a combination of PatientID and PolicyNumber can uniquely identify an insurance record.

7. Medications Table

- **Functional Dependencies**:
    - MedicationID → Name, Dosage, Instructions
    - MedicationID uniquely determines the medication details, including its Name, Dosage, and Instructions.

8. TreatmentMedications Table (Junction Table)

- **Functional Dependencies**:
    - TreatmentID, MedicationID → Dosage, Instructions
    - This table links TreatmentID and MedicationID, which together determine the prescribed dosage and instructions for that specific treatment-medication combination.

RELATIONS USING NORMALIZATION BY DECOMPOSITION

Normalization by Decomposition Process

1. **Doctors Table**
    - DoctorID → Name, Specialization, ContactInfo
2. **Patients Table**

- PatientID → Name, DateOfBirth, ContactInfo

3. **Appointments Table**
   - AppointmentID → PatientID, DoctorID, Date, Time
4. **Treatments Table**
   - TreatmentID→PatientID, DoctorID, Diagnosis, Procedure, Date
5. **Billing Table**
   - BillID → PatientID, Amount, Date, Status
6. **Insurance Table**
   - InsuranceID → PatientID, Provider, PolicyNumber, CoverageDetails
7. **Medications Table**
   - MedicationID → Name, Dosage, Instructions
8. **TreatmentMedications Table**
   - TreatmentID, MedicationID → Dosage, Instructions

Decomposition for Each Table

1. Doctors Table

- **Original Table**: DOCTOR (DoctorID, Name, Specialization, ContactInfo)
- No partial dependencies.
- **Resulting 2NF Table**: DOCTOR (DoctorID, Name, Specialization, ContactInfo)

2. Patients Table

- **Original Table**: PATIENT (PatientID, Name, DateOfBirth, ContactInfo
- No partial dependencies.
- **Resulting 2NF Table**: PATIENT (PatientID, Name, DateOfBirth, ContactInfo

3. Appointments Table

- **Original Table**: APPOINTMENT (AppointmentID, PatientID, DoctorID, Date, Time
- No partial dependencies.
- **Resulting 2NF Table**: APPOINTMENT (AppointmentID, PatientID, DoctorID, Date, Time

4. Treatments Table

- **Original Table**: TREATMENT (TreatmentID, PatientID, DoctorID, Diagnosis, Procedure, Date
- No partial dependencies.
- **Resulting 2NF Table**: TREATMENT (TreatmentID, PatientID, DoctorID, Diagnosis, Procedure, Date

5. Billing Table

- **Original Table**: BILLING (BillID, PatientID, Amount, Date, Status
- No partial dependencies.
- **Resulting 2NF Table**: BILLING (BillID, PatientID, Amount, Date, Status

6. Insurance Table

- **Original Table**: INSURANCE (InsuranceID, PatientID, Provider, PolicyNumber, CoverageDetails
- No partial dependencies.
- **Resulting 2NF Table**: INSURANCE (InsuranceID, PatientID, Provider, PolicyNumber, CoverageDetails

7. Medications Table

- **Original Table**: MEDICATION (MedicationID, Name, Dosage, Instructions
- No partial dependencies.
- **Resulting 2NF Table**: MEDICATION (MedicationID, Name, Dosage, Instructions

8. TreatmentMedications Table

- **Original Table**: TREATMENT_MEDICATIONS (TreatmentID, MedicationID, Dosage, Instructions)
- No partial dependencies.
- **Resulting 2NF Table**: TREATMENT_MEDICATIONS (TreatmentID, MedicationID, Dosage, Instructions)

Final 3NF Relations:

- **DOCTOR (DoctorID, Name, Specialization, ContactInfo)** No transitive dependencies.
- **PATIENT (PatientID, Name, DateOfBirth, ContactInfo)** No transitive dependencies.
- **APPOINTMENT (AppointmentID, PatientID, DoctorID, Date, Time)** No transitive dependencies.
- **TREATMENT (TreatmentID, PatientID, DoctorID, Diagnosis, Procedure, Date)** No transitive dependencies.
- **BILLING (BillID, PatientID, Amount, Date, Status)** No transitive dependencies.
- **INSURANCE (InsuranceID, PatientID, Provider, PolicyNumber, CoverageDetails)** No transitive dependencies.
- **MEDICATION (MedicationID, Name, Dosage, Instructions)** No transitive dependencies.
- **TREATMENT_MEDICATIONS (TreatmentID, MedicationID, Dosage, Instructions)** No transitive dependencies.

RELATIONAL ALGEBRA TO SHOW QUERIES

1. Doctor-Patient Coordination

Query to Find All Appointments for a Specific Doctor:

- Relational Algebra:

  πPatientID, Date, Time(σDoctorID=′D001′(APPOINTMENT))

QUERY IN SQL

```
SELECT PatientID, Date, Time

FROM APPOINTMENT

WHERE DoctorID = 'D001';
```

2. Treatment and Follow-Up Tracking

Query to Find All Treatments Associated with a Specific Patient:

- Relational Algebra:

  πDiagnosis, Procedure, Date(σPatientID=′P001′(TREATMENT))

QUERY IN SQL

```
SELECT Diagnosis, Procedure, Date

FROM TREATMENT

WHERE PatientID = 'P001';
```

3. Medication Management

Query to List All Medications Prescribed for a Specific Treatment:

- Relational Algebra:

  πName, Dosage, Instructions(MEDICATION⋈MedicationID(σTreatmentID=′T001′(TREATMENT_MEDICATIONS)))

QUERY IN SQL

```
SELECT MEDICATION.Name, MEDICATION.Dosage, MEDICATION.Instructions

FROM MEDICATION

JOIN TREATMENT_MEDICATIONS ON MEDICATION.MedicationID = TREATMENT_MEDICATIONS.MedicationID

WHERE TREATMENT_MEDICATIONS.TreatmentID = 'T001';
```

4. Billing and Insurance

Query to Retrieve Billing Details for a Patient with Insurance Information:

- Relational Algebra:

  πBillID, Amount, Date, Status, Provider, PolicyNumber(BILLING⋈PatientIDINSURANCE)

QUERY IN SQL

```
SELECT BILLING.BillID, BILLING.Amount, BILLING.Date, BILLING.Status, INSURANCE.Provider, INSURANCE.PolicyNumber

FROM BILLING

JOIN INSURANCE ON BILLING.PatientID = INSURANCE.PatientID;
```

5. Data Access and Security

Query to Show Appointment History of a Patient While Restricting Access Based on Roles:

- Relational Algebra:

  πDate, Time, DoctorID(σPatientID=′P001′(APPOINTMENT))

QUERY IN SQL

```
SELECT Date, Time, DoctorID

FROM APPOINTMENT

WHERE PatientID = 'P001';
```

6. Comprehensive Query: Patient Treatment and Medication

Query to Retrieve a Patient's Full Treatment History Including Medications:

- Relational Algebra:

  πPatientID, Diagnosis, Procedure, Date, Name, Dosage, Instructions((σPatientID=′P001′(TREATMENT))⋈TreatmentID TREATMENT_MEDICATIONS)⋈MedicationIDMEDICATION

QUERY IN SQL

```
SELECT TREATMENT.PatientID, TREATMENT.Diagnosis, TREATMENT.Procedure, TREATMENT.Date, MEDICATION.Name, MEDICATION.Dosage, MEDICATION.Instructions

FROM TREATMENT

JOIN TREATMENT_MEDICATIONS ON TREATMENT.TreatmentID = TREATMENT_MEDICATIONS.TreatmentID

JOIN MEDICATION ON TREATMENT_MEDICATIONS.MedicationID = MEDICATION.MedicationID

WHERE TREATMENT.PatientID = 'P001';
```

CREATION, POPULATION, WRITING AND EXECUTING QUERIES TO PROVIDE SOLUTIONS USING MYSQL

```
mysql> use finalproject;
Database changed
mysql> CREATE TABLE doctor (
    ->     doctorid VARCHAR(10) PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     specialization VARCHAR(255) NOT NULL,
    ->     contactinfo VARCHAR(15) NOT NULL
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> CREATE TABLE patient (
    ->     patientid VARCHAR(10) PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     dateofbirth DATE NOT NULL,
    ->     contactinfo VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE appointment (
    ->     appointmentid VARCHAR(10) PRIMARY KEY,
    ->     patientid VARCHAR(10) NOT NULL,
    ->     doctorid VARCHAR(10) NOT NULL,
    ->     appointment_date DATE NOT NULL,
    ->     appointment_time TIME NOT NULL,
    ->     FOREIGN KEY (patientid) REFERENCES patient(patientid) ON DELETE CASCADE,
    ->     FOREIGN KEY (doctorid) REFERENCES doctor(doctorid) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE TABLE treatment (
    ->     treatmentid VARCHAR(10) PRIMARY KEY,
    ->     patientid VARCHAR(10) NOT NULL,
    ->     doctorid VARCHAR(10) NOT NULL,
    ->     diagnosis VARCHAR(255) NOT NULL,
    ->     procedure_details VARCHAR(255) NOT NULL,
    ->     treatment_date DATE NOT NULL,
    ->     FOREIGN KEY (patientid) REFERENCES patient(patientid) ON DELETE CASCADE,
    ->     FOREIGN KEY (doctorid) REFERENCES doctor(doctorid) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> CREATE TABLE billing (
    ->     billid VARCHAR(10) PRIMARY KEY,
    ->     patientid VARCHAR(10) NOT NULL,
    ->     amount DECIMAL(10, 2) NOT NULL,
    ->     billing_date DATE NOT NULL,
    ->     status VARCHAR(50) NOT NULL,
    ->     FOREIGN KEY (patientid) REFERENCES patient(patientid) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE TABLE insurance (
    ->     insuranceid VARCHAR(10) PRIMARY KEY,
    ->     patientid VARCHAR(10) NOT NULL,
    ->     provider VARCHAR(255) NOT NULL,
    ->     policynumber VARCHAR(50) NOT NULL,
    ->     coveragedetails VARCHAR(255) NOT NULL,
    ->     FOREIGN KEY (patientid) REFERENCES patient(patientid) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE medication (
    ->     medicationid VARCHAR(10) PRIMARY KEY,
    ->     name VARCHAR(255) NOT NULL,
    ->     dosage VARCHAR(50) NOT NULL,
    ->     instructions VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> CREATE TABLE treatment_medication (
    ->     treatmentid VARCHAR(10) NOT NULL,
    ->     medicationid VARCHAR(10) NOT NULL,
    ->     dosage VARCHAR(50) NOT NULL,
    ->     instructions VARCHAR(255) NOT NULL,
    ->     PRIMARY KEY (treatmentid, medicationid),
    ->     FOREIGN KEY (treatmentid) REFERENCES treatment(treatmentid) ON DELETE CASCADE,
    ->     FOREIGN KEY (medicationid) REFERENCES medication(medicationid) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO doctor (doctorid, name, specialization, contactinfo) VALUES
    -> ('D001', 'Dr. Boateng', 'Cardiologist', '847-556-7794'),
    -> ('D002', 'Dr. Armstrong', 'General Practitioner', '469-885-1257'),
    -> ('D003', 'Dr. Smith', 'Dermatologist', '557-648-2963'),
    -> ('D004', 'Dr. Johnson', 'Pediatrician', '446-778-8791'),
    -> ('D005', 'Dr. Patel', 'Endocrinologist', '458-231-0104'),
    -> ('D006', 'Dr. Taylor', 'Neurologist', '569-678-0009');
Query OK, 6 rows affected (0.05 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> INSERT INTO patient (patientid, name, dateofbirth, contactinfo) VALUES
    -> ('P001', 'Daniel Timmis', '1985-06-15', 'daniel.timmis@gmail.com'),
    -> ('P002', 'Sophia Wells', '1992-09-23', 'sophia.wells@gmail.com'),
    -> ('P003', 'Lucas Harper', '1978-12-05', 'lucas.harper@gmail.com'),
    -> ('P004', 'Emily Carter', '2003-03-18', 'emily.carter@gmail.com'),
    -> ('P005', 'Alice Monroe', '1975-07-20', 'alice.monroe@gmail.com'),
    -> ('P006', 'Mark Johnson', '1989-04-13', 'mark.johnson@gmail.com'),
    -> ('P007', 'Ella Baker', '2010-11-05', 'ella.baker@gmail.com');
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> INSERT INTO appointment (appointmentid, patientid, doctorid, appointment_date, appointment_time) VALUES
    -> ('A001', 'P001', 'D001', '2024-11-01', '09:00:00'),
    -> ('A002', 'P001', 'D002', '2024-11-08', '10:30:00'),
    -> ('A003', 'P002', 'D002', '2024-11-15', '11:00:00'),
    -> ('A004', 'P003', 'D003', '2024-11-10', '14:00:00'),
    -> ('A005', 'P004', 'D004', '2024-11-20', '10:00:00'),
    -> ('A006', 'P001', 'D003', '2024-11-18', '16:00:00'),
    -> ('A007', 'P003', 'D001', '2024-11-25', '09:30:00'),
    -> ('A008', 'P004', 'D002', '2024-12-01', '11:00:00'),
    -> ('A009', 'P005', 'D005', '2024-11-22', '08:30:00'),
    -> ('A010', 'P006', 'D006', '2024-11-23', '10:00:00'),
    -> ('A011', 'P007', 'D004', '2024-12-05', '14:00:00'),
    -> ('A012', 'P005', 'D001', '2024-11-25', '13:30:00'),
    -> ('A013', 'P006', 'D003', '2024-12-01', '09:00:00');
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> INSERT INTO treatment (treatmentid, patientid, doctorid, diagnosis, procedure_details, treatment_date) VALUES
    -> ('T001', 'P001', 'D001', 'Heart Condition', 'Electrocardiogram', '2024-11-01'),
    -> ('T002', 'P001', 'D002', 'General Checkup', 'Physical Examination', '2024-11-08'),
    -> ('T003', 'P002', 'D002', 'Flu Symptoms', 'Flu Test and Medication', '2024-11-15'),
    -> ('T004', 'P003', 'D003', 'Skin Allergy', 'Allergy Test and Cream', '2024-11-10'),
    -> ('T005', 'P004', 'D004', 'Childhood Vaccination', 'MMR Vaccine', '2024-11-20'),
    -> ('T006', 'P001', 'D003', 'Eczema', 'Skin Examination and Prescription', '2024-11-18'),
    -> ('T007', 'P005', 'D005', 'Diabetes Mellitus', 'Blood Sugar Test and Diet Plan', '2024-11-22'),
    -> ('T008', 'P006', 'D006', 'Migraine', 'MRI and Prescription', '2024-11-23'),
    -> ('T009', 'P007', 'D004', 'Childhood Asthma', 'Pulmonary Function Test', '2024-12-05'),
    -> ('T010', 'P005', 'D001', 'Hypertension', 'Blood Pressure Monitoring', '2024-11-25'),
    -> ('T011', 'P006', 'D003', 'Skin Infection', 'Antibiotic Prescription', '2024-12-01');
Query OK, 11 rows affected (0.01 sec)
```

```
Records: 11  Duplicates: 0  Warnings: 0

mysql> INSERT INTO billing (billid, patientid, amount, billing_date, status) VALUES
    -> ('B001', 'P001', 150.00, '2024-11-01', 'Paid'),
    -> ('B002', 'P001', 75.00, '2024-11-08', 'Pending'),
    -> ('B003', 'P002', 100.00, '2024-11-15', 'Paid'),
    -> ('B004', 'P003', 200.00, '2024-11-10', 'Pending'),
    -> ('B005', 'P004', 150.00, '2024-11-20', 'Paid'),
    -> ('B006', 'P001', 100.00, '2024-11-18', 'Paid'),
    -> ('B007', 'P005', 300.00, '2024-11-22', 'Paid'),
    -> ('B008', 'P006', 250.00, '2024-11-23', 'Pending'),
    -> ('B009', 'P007', 150.00, '2024-12-05', 'Paid'),
    -> ('B010', 'P005', 100.00, '2024-11-25', 'Pending'),
    -> ('B011', 'P006', 200.00, '2024-12-01', 'Paid');
Query OK, 11 rows affected (0.01 sec)
Records: 11  Duplicates: 0  Warnings: 0

mysql> INSERT INTO insurance (insuranceid, patientid, provider, policynumber, coveragedetails) VALUES
    -> ('I001', 'P001', 'HealthPlus', 'HP12345', 'Covers 80% of general and specialist consultations'),
    -> ('I002', 'P002', 'MediCare', 'MC67890', 'Covers 70% of general consultations and tests'),
    -> ('I003', 'P003', 'WellCare', 'WC54321', 'Covers 90% of dermatologist consultations'),
    -> ('I004', 'P004', 'ChildHealth', 'CH09876', 'Covers 100% of vaccinations for minors'),
    -> ('I005', 'P005', 'HealthFirst', 'HF56789', 'Covers 75% of endocrinology-related consultations'),
    -> ('I006', 'P006', 'MediSafe', 'MS45678', 'Covers 80% of neurological treatments'),
    -> ('I007', 'P007', 'KidsCare', 'KC12345', 'Covers 90% of pediatric treatments and diagnostics');
Query OK, 7 rows affected (0.01 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> INSERT INTO medication (medicationid, name, dosage, instructions) VALUES
    -> ('M001', 'Aspirin', '75mg', 'Take one daily after meals'),
    -> ('M002', 'Paracetamol', '500mg', 'Take two every 6 hours as needed for pain or fever'),
    -> ('M003', 'Antihistamine Cream', 'Apply daily', 'Apply on affected areas after cleaning'),
    -> ('M004', 'Ibuprofen', '200mg', 'Take one every 8 hours as needed for pain'),
    -> ('M005', 'Vitamin D Drops', '1 drop daily', 'Administer orally to the child once a day'),
    -> ('M006', 'Metformin', '500mg', 'Take twice daily with meals'),
    -> ('M007', 'Propranolol', '10mg', 'Take once daily in the morning'),
    -> ('M008', 'Fluticasone Inhaler', '1 puff', 'Use twice daily for maintenance'),
    -> ('M009', 'Amoxicillin', '250mg', 'Take three times daily for 7 days'),
    -> ('M010', 'Sumatriptan', '50mg', 'Take one tablet during migraine onset');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO treatment_medication (treatmentid, medicationid, dosage, instructions) VALUES
    -> ('T001', 'M001', '75mg', 'Take one daily after meals'),
    -> ('T003', 'M002', '500mg', 'Take two every 6 hours as needed for pain or fever'),
    -> ('T004', 'M003', 'Apply daily', 'Apply on affected areas after cleaning'),
    -> ('T005', 'M005', '1 drop daily', 'Administer orally to the child once a day'),
    -> ('T006', 'M004', '200mg', 'Take one every 8 hours as needed for pain'),
    -> ('T007', 'M006', '500mg', 'Take twice daily with meals'),
    -> ('T007', 'M007', '10mg', 'Take once daily in the morning'),
    -> ('T008', 'M010', '50mg', 'Take one tablet during migraine onset'),
    -> ('T009', 'M008', '1 puff', 'Use twice daily for maintenance'),
    -> ('T011', 'M009', '250mg', 'Take three times daily for 7 days');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> show tables;
+------------------------+
| Tables_in_finalproject |
+------------------------+
| appointment            |
| billing                |
| doctor                 |
| insurance              |
| medication             |
| patient                |
| treatment              |
| treatment_medication   |
+------------------------+
8 rows in set (0.06 sec)

mysql> SELECT * FROM doctor;
+----------+---------------+----------------------+--------------+
| doctorid | name          | specialization       | contactinfo  |
+----------+---------------+----------------------+--------------+
| D001     | Dr. Boateng   | Cardiologist         | 847-556-7794 |
| D002     | Dr. Armstrong | General Practitioner | 469-885-1257 |
| D003     | Dr. Smith     | Dermatologist        | 557-648-2963 |
| D004     | Dr. Johnson   | Pediatrician         | 446-778-8791 |
| D005     | Dr. Patel     | Endocrinologist      | 458-231-0104 |
| D006     | Dr. Taylor    | Neurologist          | 569-678-0009 |
+----------+---------------+----------------------+--------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM patient;
+-----------+---------------+-------------+-------------------------+
| patientid | name          | dateofbirth | contactinfo             |
+-----------+---------------+-------------+-------------------------+
| P001      | Daniel Timmis | 1985-06-15  | daniel.timmis@gmail.com |
| P002      | Sophia Wells  | 1992-09-23  | sophia.wells@gmail.com  |
| P003      | Lucas Harper  | 1978-12-05  | lucas.harper@gmail.com   |
| P004      | Emily Carter  | 2003-03-18  | emily.carter@gmail.com   |
| P005      | Alice Monroe  | 1975-07-20  | alice.monroe@gmail.com   |
| P006      | Mark Johnson  | 1989-04-13  | mark.johnson@gmail.com   |
| P007      | Ella Baker    | 2010-11-05  | ella.baker@gmail.com     |
+-----------+---------------+-------------+--------------------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM appointment;
+---------------+-----------+----------+------------------+------------------+
| appointmentid | patientid | doctorid | appointment_date | appointment_time |
+---------------+-----------+----------+------------------+------------------+
| A001          | P001      | D001     | 2024-11-01       | 09:00:00         |
| A002          | P001      | D002     | 2024-11-08       | 10:30:00         |
| A003          | P002      | D002     | 2024-11-15       | 11:00:00         |
| A004          | P003      | D003     | 2024-11-10       | 14:00:00         |
| A005          | P004      | D004     | 2024-11-20       | 10:00:00         |
| A006          | P001      | D003     | 2024-11-18       | 16:00:00         |
| A007          | P003      | D001     | 2024-11-25       | 09:30:00         |
| A008          | P004      | D002     | 2024-12-01       | 11:00:00         |
| A009          | P005      | D005     | 2024-11-22       | 08:30:00         |
| A010          | P006      | D006     | 2024-11-23       | 10:00:00         |
| A011          | P007      | D004     | 2024-12-05       | 14:00:00         |
| A012          | P005      | D001     | 2024-11-25       | 13:30:00         |
| A013          | P006      | D003     | 2024-12-01       | 09:00:00         |
+---------------+-----------+----------+------------------+------------------+
13 rows in set (0.00 sec)

mysql> SELECT * FROM treatment;
+-------------+-----------+----------+-----------------------+-----------------------------------+----------------+
| treatmentid | patientid | doctorid | diagnosis             | procedure_details                 | treatment_date |
+-------------+-----------+----------+-----------------------+-----------------------------------+----------------+
| T001        | P001      | D001     | Heart Condition       | Electrocardiogram                 | 2024-11-01     |
| T002        | P001      | D002     | General Checkup       | Physical Examination              | 2024-11-08     |
| T003        | P002      | D002     | Flu Symptoms          | Flu Test and Medication           | 2024-11-15     |
| T004        | P003      | D003     | Skin Allergy          | Allergy Test and Cream            | 2024-11-10     |
| T005        | P004      | D004     | Childhood Vaccination | MMR Vaccine                       | 2024-11-20     |
| T006        | P001      | D003     | Eczema                | Skin Examination and Prescription | 2024-11-18     |
| T007        | P005      | D005     | Diabetes Mellitus     | Blood Sugar Test and Diet Plan    | 2024-11-22     |
| T008        | P006      | D006     | Migraine              | MRI and Prescription              | 2024-11-23     |
| T009        | P007      | D004     | Childhood Asthma      | Pulmonary Function Test           | 2024-12-05     |
| T010        | P005      | D001     | Hypertension          | Blood Pressure Monitoring         | 2024-11-25     |
| T011        | P006      | D003     | Skin Infection        | Antibiotic Prescription           | 2024-12-01     |
+-------------+-----------+----------+-----------------------+-----------------------------------+----------------+
11 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM billing;
+--------+-----------+--------+--------------+---------+
| billid | patientid | amount | billing_date | status  |
+--------+-----------+--------+--------------+---------+
| B001   | P001      | 150.00 | 2024-11-01   | Paid    |
| B002   | P001      |  75.00 | 2024-11-08   | Pending |
| B003   | P002      | 100.00 | 2024-11-15   | Paid    |
| B004   | P003      | 200.00 | 2024-11-10   | Pending |
| B005   | P004      | 150.00 | 2024-11-20   | Paid    |
| B006   | P001      | 100.00 | 2024-11-18   | Paid    |
| B007   | P005      | 300.00 | 2024-11-22   | Paid    |
| B008   | P006      | 250.00 | 2024-11-23   | Pending |
| B009   | P007      | 150.00 | 2024-12-05   | Paid    |
| B010   | P005      | 100.00 | 2024-11-25   | Pending |
| B011   | P006      | 200.00 | 2024-12-01   | Paid    |
+--------+-----------+--------+--------------+---------+
11 rows in set (0.00 sec)

mysql> SELECT * FROM insurance;
+-------------+-----------+-------------+--------------+----------------------------------------------------+
| insuranceid | patientid | provider    | policynumber | coveragedetails                                    |
+-------------+-----------+-------------+--------------+----------------------------------------------------+
| I001        | P001      | HealthPlus  | HP12345      | Covers 80% of general and specialist consultations |
| I002        | P002      | MediCare    | MC67890      | Covers 70% of general consultations and tests      |
| I003        | P003      | WellCare    | WC54321      | Covers 90% of dermatologist consultations          |
| I004        | P004      | ChildHealth | CH09876      | Covers 100% of vaccinations for minors             |
| I005        | P005      | HealthFirst | HF56789      | Covers 75% of endocrinology-related consultations  |
| I006        | P006      | MediSafe    | MS45678      | Covers 80% of neurological treatments              |
| I007        | P007      | KidsCare    | KC12345      | Covers 90% of pediatric treatments and diagnostics |
+-------------+-----------+-------------+--------------+----------------------------------------------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM medication;
+--------------+---------------------+--------------+----------------------------------------------------+
| medicationid | name                | dosage       | instructions                                       |
+--------------+---------------------+--------------+----------------------------------------------------+
| M001         | Aspirin             | 75mg         | Take one daily after meals                         |
| M002         | Paracetamol         | 500mg        | Take two every 6 hours as needed for pain or fever |
| M003         | Antihistamine Cream | Apply daily  | Apply on affected areas after cleaning             |
| M004         | Ibuprofen           | 200mg        | Take one every 8 hours as needed for pain          |
| M005         | Vitamin D Drops     | 1 drop daily | Administer orally to the child once a day          |
| M006         | Metformin           | 500mg        | Take twice daily with meals                        |
| M007         | Propranolol         | 10mg         | Take once daily in the morning                     |
| M008         | Fluticasone Inhaler | 1 puff       | Use twice daily for maintenance                    |
| M009         | Amoxicillin          | 250mg         | Take three times daily for 7 days                     |
| M010         | Sumatriptan          | 50mg          | Take one tablet during migraine onset                 |
+--------------+----------------------+---------------+-------------------------------------------------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM treatment_medication;
+-------------+--------------+--------------+-----------------------------------------------------+
| treatmentid | medicationid | dosage       | instructions                                        |
+-------------+--------------+--------------+-----------------------------------------------------+
| T001        | M001         | 75mg         | Take one daily after meals                          |
| T003        | M002         | 500mg        | Take two every 6 hours as needed for pain or fever |
| T004        | M003         | Apply daily  | Apply on affected areas after cleaning              |
| T005        | M005         | 1 drop daily | Administer orally to the child once a day           |
| T006        | M004         | 200mg        | Take one every 8 hours as needed for pain           |
| T007        | M006         | 500mg        | Take twice daily with meals                         |
| T007        | M007         | 10mg         | Take once daily in the morning                      |
| T008        | M010         | 50mg         | Take one tablet during migraine onset               |
| T009        | M008         | 1 puff       | Use twice daily for maintenance                     |
| T011        | M009         | 250mg        | Take three times daily for 7 days                   |
+-------------+--------------+--------------+-----------------------------------------------------+
10 rows in set (0.00 sec)

mysql> -- Doctor - Patient Coordination
mysql> /* Query: Retrieve all appointments for Dr. Boateng (Cardiologist)
          This helps Dr. Boateng plan their schedule effectively
Explanation: This query filters the appointment table by doctorid, ensuring Dr. Boateng can see all his upcoming
appointments.
     */

mysql> SELECT appointmentid, patientid, appointment_date, appointment_time
    -> FROM appointment
    -> WHERE doctorid = 'D001';
+---------------+-----------+------------------+------------------+
| appointmentid | patientid | appointment_date | appointment_time |
+---------------+-----------+------------------+------------------+
| A001          | P001      | 2024-11-01       | 09:00:00         |
| A007          | P003      | 2024-11-25       | 09:30:00         |
| A012          | P005      | 2024-11-25       | 13:30:00         |
+---------------+-----------+------------------+------------------+
3 rows in set (0.02 sec)

mysql> -- Treatment and Follow-Up Tracking
mysql> /* Query: List all treatments for Daniel Timmis (PatientID: P001). This gives an overview of his ongoing care
for heart and skin conditions.
```

```
Explanation: Displays all treatments for a specific patient, including diagnoses, procedures, and treatment dates,
which helps doctors assess the patient's history.
       */

mysql> SELECT treatmentid, diagnosis, procedure_details, treatment_date
    -> FROM treatment
    -> WHERE patientid = 'P001';
+-------------+-----------------+-----------------------------------+----------------+
| treatmentid | diagnosis       | procedure_details                 | treatment_date |
+-------------+-----------------+-----------------------------------+----------------+
| T001        | Heart Condition | Electrocardiogram                 | 2024-11-01     |
| T002        | General Checkup | Physical Examination              | 2024-11-08     |
| T006        | Eczema          | Skin Examination and Prescription | 2024-11-18     |
+-------------+-----------------+-----------------------------------+----------------+
3 rows in set (0.02 sec)

mysql> -- Medication Management
mysql> /* Query: Retrieve medications prescribed for Treatment T001 (Heart Condition). This helps track the drugs
associated with specific treatments.
Explanation: This query uses a join between the medication and treatment_medication tables to fetch all drugs
prescribed for a specific treatment.
 */
mysql> SELECT M.name AS medication_name, M.dosage, M.instructions
    -> FROM medication M
    -> JOIN treatment_medication TM ON M.medicationid = TM.medicationid
    -> WHERE TM.treatmentid = 'T001';
+-----------------+--------+---------------------------+
| medication_name | dosage | instructions              |
+-----------------+--------+---------------------------+
| Aspirin         | 75mg   | Take one daily after meals |
+-----------------+--------+---------------------------+
1 row in set (0.00 sec)

mysql> -- Billing and Insurance Integration
mysql> /* Query: Combine billing and insurance details for Daniel Timmis. This ensures accurate billing and insurance
claim management
Explanation: This query joins the billing and insurance tables to provide a complete picture of the patient's bills and
insurance coverage.
*/
mysql> SELECT B.billid, B.amount, B.billing_date, B.status, I.provider, I.policynumber
    -> FROM billing B
    -> JOIN insurance I ON B.patientid = I.patientid
    -> WHERE B.patientid = 'P001';
+--------+--------+--------------+---------+------------+--------------+
| billid | amount | billing_date | status  | provider   | policynumber |
+--------+--------+--------------+---------+------------+---------------+
| B001   | 150.00 | 2024-11-01   | Paid    | HealthPlus | HP12345       |
| B002   |  75.00 | 2024-11-08   | Pending | HealthPlus | HP12345       |
| B006   | 100.00 | 2024-11-18   | Paid    | HealthPlus | HP12345       |
+--------+--------+--------------+---------+------------+---------------+
3 rows in set (0.02 sec)

mysql> -- Most Common Diagnoses
mysql> /* Query: Find the most frequently treated conditions at the hospital. This can help identify trends and
allocate resources.
Explanation: Groups the treatment table by diagnosis to count the number of cases for each condition, helping the
hospital prioritize resources.
*/
mysql> SELECT diagnosis, COUNT(treatmentid) AS total_cases
    -> FROM treatment
    -> GROUP BY diagnosis
    -> ORDER BY total_cases DESC;
+-----------------------+-------------+
| diagnosis             | total_cases |
+-----------------------+-------------+
| Heart Condition       |           1 |
| General Checkup       |           1 |
| Flu Symptoms          |           1 |
| Skin Allergy          |           1 |
| Childhood Vaccination |           1 |
| Eczema                |           1 |
| Diabetes Mellitus     |           1 |
| Migraine              |           1 |
| Childhood Asthma      |           1 |
| Hypertension          |           1 |
| Skin Infection        |           1 |
+-----------------------+-------------+
11 rows in set (0.02 sec)

mysql> -- Patients Without Insurance
mysql> /* Query: List patients who do not have any insurance information in the system. This helps the hospital
identify patients who may need financial assistance.
Explanation: Uses a LEFT JOIN to find patients in the patient table who do not have an entry in the insurance table.
*/
mysql> SELECT P.patientid, P.name, P.contactinfo
    -> FROM patient P
    -> LEFT JOIN insurance I ON P.patientid = I.patientid
    -> WHERE I.insuranceid IS NULL;
Empty set (0.00 sec)
```

```
mysql> -- Appointments Scheduled This Week
mysql> /* Query: Retrieve appointments happening within the current week. This helps the hospital prepare for upcoming
patient visits.
Explanation: Filters appointments using YEARWEEK to retrieve only those scheduled in the current week.
*/
mysql> SELECT appointmentid, patientid, doctorid, appointment_date, appointment_time
    -> FROM appointment
    -> WHERE YEARWEEK(appointment_date, 1) = YEARWEEK(CURDATE(), 1);
+---------------+-----------+----------+------------------+------------------+
| appointmentid | patientid | doctorid | appointment_date | appointment_time |
+---------------+-----------+----------+------------------+------------------+
| A011          | P007      | D004     | 2024-12-05       | 14:00:00         |
+---------------+-----------+----------+------------------+------------------+
1 row in set (0.00 sec)

mysql> -- Total Billing for Each Patient
mysql> /* Query: Calculate the total amount billed for each patient. This provides a financial overview for hospital
management.
Explanation: Groups the billing table by patientid and calculates the total billed amount for each patient.
*/
mysql> SELECT patientid, SUM(amount) AS total_billed
    -> FROM billing
    -> GROUP BY patientid;
+-----------+--------------+
| patientid | total_billed |
+-----------+--------------+
| P001      |       325.00 |
| P002      |       100.00 |
| P003      |       200.00 |
| P004      |       150.00 |
| P005      |       400.00 |
| P006      |       450.00 |
| P007      |       150.00 |
+-----------+--------------+
7 rows in set (0.02 sec)

mysql> -- Full Patient Report
mysql> /* Query: Retrieve a complete report for Daniel Timmis, including appointments, treatments, and medications.
This offers a comprehensive view of his care history.
Explanation: Combines data from multiple tables to create a detailed report for a specific patient.
*/
mysql> SELECT
    ->     P.name AS patient_name,
    ->     A.appointment_date,
    ->     A.appointment_time,
    ->     D.name AS doctor_name,
    ->     D.specialization,
    ->     T.diagnosis,
    ->     T.procedure_details,
    ->     M.name AS medication_name,
    ->     M.dosage,
    ->     M.instructions,
    ->     (SELECT SUM(B.amount) FROM billing B WHERE B.patientid = P.patientid) AS total_billed
    -> FROM patient P
    -> JOIN appointment A ON P.patientid = A.patientid
    -> JOIN doctor D ON A.doctorid = D.doctorid
    -> LEFT JOIN treatment T ON P.patientid = T.patientid AND A.appointment_date = T.treatment_date
    -> LEFT JOIN treatment_medication TM ON T.treatmentid = TM.treatmentid
    -> LEFT JOIN medication M ON TM.medicationid = M.medicationid
    -> WHERE P.patientid = 'P001';
+---------------+------------------+------------------+---------------+----------------------+-----------------+-------
----------------------------+-----------------+--------+-------------------------------------------+--------------+
| patient_name  | appointment_date | appointment_time | doctor_name   | specialization       | diagnosis       |
procedure_details                 | medication_name | dosage | instructions                              | total_billed
|
+---------------+------------------+------------------+---------------+----------------------+-----------------+-------
----------------------------+-----------------+--------+-------------------------------------------+--------------+
| Daniel Timmis | 2024-11-01       | 09:00:00         | Dr. Boateng   | Cardiologist         | Heart Condition |
Electrocardiogram                 | Aspirin         | 75mg   | Take one daily after meals                |       325.00
|
| Daniel Timmis | 2024-11-08       | 10:30:00         | Dr. Armstrong | General Practitioner | General Checkup |
Physical Examination              | NULL            | NULL   | NULL                                      |       325.00
|
| Daniel Timmis | 2024-11-18       | 16:00:00         | Dr. Smith     | Dermatologist        | Eczema          | Skin
Examination and Prescription | Ibuprofen       | 200mg  | Take one every 8 hours as needed for pain |       325.00 |
+---------------+------------------+------------------+---------------+----------------------+-----------------+-------
----------------------------+-----------------+--------+-------------------------------------------+--------------+
3 rows in set (0.01 sec)

mysql> -- Analyze Insurance Coverage
mysql> /* Show how much of a patient's total bills are covered by their insurance.
Explanation: Calculates the insurance-covered portion of bills assuming 80% coverage.
*/
mysql> SELECT
    ->     B.patientid,
    ->     SUM(B.amount) AS total_billed,
    ->     SUM(B.amount * 0.8) AS insurance_coverage -- Assuming 80% coverage
    -> FROM billing B
    -> JOIN insurance I ON B.patientid = I.patientid
    -> GROUP BY B.patientid;
```

```
+-----------+--------------+--------------------+
| patientid | total_billed | insurance_coverage |
+-----------+--------------+--------------------+
| P001      |       325.00 |            260.000 |
| P002      |       100.00 |             80.000 |
| P003      |       200.00 |            160.000 |
| P004      |       150.00 |            120.000 |
| P005      |       400.00 |            320.000 |
| P006      |       450.00 |            360.000 |
| P007      |       150.00 |            120.000 |
+-----------+--------------+--------------------+
7 rows in set (0.13 sec)

mysql> -- Patients Treated by Multiple Doctors
mysql> /* Query: Identify patients who have been treated by more than one doctor
Explanation: Uses COUNT(DISTINCT doctorid) to find patients who have seen multiple doctors.
*/
mysql> SELECT T.patientid, COUNT(DISTINCT T.doctorid) AS doctor_count
    -> FROM treatment T
    -> GROUP BY T.patientid
    -> HAVING doctor_count > 1;
+-----------+--------------+
| patientid | doctor_count |
+-----------+--------------+
| P001      |            3 |
| P005      |            2 |
| P006      |            2 |
+-----------+--------------+
3 rows in set (0.04 sec)

mysql> -- Most frequent Appointment Times
mysql> -- Query: Find the most common appointment times across all doctors
mysql> SELECT appointment_time, COUNT(*) AS appointment_count
    -> FROM appointment
    -> GROUP BY appointment_time
    -> ORDER BY appointment_count DESC
    -> LIMIT 5;
+------------------+-------------------+
| appointment_time | appointment_count |
+------------------+-------------------+
| 09:00:00         |                 2 |
| 11:00:00         |                 2 |
| 14:00:00         |                 2 |
| 10:00:00         |                 2 |
| 10:30:00         |                 1 |
+------------------+-------------------+
5 rows in set (0.03 sec)

mysql> -- Revenue Per Doctor
mysql> -- Calculate the total revenue generated by each doctor based on treatments
mysql> SELECT
    ->     T.doctorid,
    ->     D.name AS doctor_name,
    ->     SUM(B.amount) AS total_revenue
    -> FROM treatment T
    -> JOIN billing B ON T.patientid = B.patientid
    -> JOIN doctor D ON T.doctorid = D.doctorid
    -> GROUP BY T.doctorid, D.name
    -> ORDER BY total_revenue DESC;
+----------+---------------+---------------+
| doctorid | doctor_name   | total_revenue |
+----------+---------------+---------------+
| D003     | Dr. Smith     |        975.00 |
| D001     | Dr. Boateng   |        725.00 |
| D006     | Dr. Taylor    |        450.00 |
| D002     | Dr. Armstrong |        425.00 |
| D005     | Dr. Patel     |        400.00 |
| D004     | Dr. Johnson   |        300.00 |
+----------+---------------+---------------+
6 rows in set (0.01 sec)

mysql> -- Automatic Appointment Logging
mysql> -- Create a trigger to log new appointments into a logs table
mysql> CREATE TABLE logs (
    ->     log_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     log_message VARCHAR(255),
    ->     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.21 sec)

mysql> -- Billing Summary
mysql> -- Query: Create a stored procedure to retrieve a billing summary for a specific patient.
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE GetBillingSummary(IN patient_id VARCHAR(10))
    -> BEGIN
    ->     SELECT
    ->         B.billid,
    ->         B.amount,
    ->         B.billing_date,
    ->         B.status
    ->     FROM billing B
    ->     WHERE B.patientid = patient_id;
    -> END //
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER ;
mysql> -- Now I will call the procedure.
mysql> CALL GetBillingSummary('P001');
+--------+--------+--------------+---------+
| billid | amount | billing_date | status  |
+--------+--------+--------------+---------+
| B001   | 150.00 | 2024-11-01   | Paid    |
| B002   |  75.00 | 2024-11-08   | Pending |
| B006   | 100.00 | 2024-11-18   | Paid    |
+--------+--------+--------------+---------+
3 rows in set (0.03 sec)

Query OK, 0 rows affected (0.04 sec)

mysql> -- Diagnosis Distribution
mysql> -- Query: Get a count of each diagnosis for visualization purposes
mysql> SELECT diagnosis, COUNT(*) AS diagnosis_count
    -> FROM treatment
    -> GROUP BY diagnosis;
+-----------------------+-----------------+
| diagnosis             | diagnosis_count |
+-----------------------+-----------------+
| Heart Condition       |               1 |
| General Checkup       |               1 |
| Flu Symptoms          |               1 |
| Skin Allergy          |               1 |
| Childhood Vaccination |               1 |
| Eczema                |               1 |
| Diabetes Mellitus     |               1 |
| Migraine              |               1 |
| Childhood Asthma      |               1 |
| Hypertension          |               1 |
| Skin Infection        |               1 |
+-----------------------+-----------------+
11 rows in set (0.00 sec)

mysql> -- Revenue Over Time
mysql> -- Query: Calculate the monthly revenue for the hospital.
mysql> SELECT
```

```
    ->     DATE_FORMAT(billing_date, '%Y-%m') AS billing_month,
    ->     SUM(amount) AS total_revenue
    -> FROM billing
    -> GROUP BY billing_month
    -> ORDER BY billing_month;
+---------------+---------------+
| billing_month | total_revenue |
+---------------+---------------+
| 2024-11       |       1425.00 |
| 2024-12       |        350.00 |
+---------------+---------------+
2 rows in set (0.01 sec)
```