

# HOSPITAL DATABASE PROJECT REPORT

---

Muhammed Abdel Hamid Shelleh

July 27, 2024

## 1 Assumptions

- Each patient can only be assigned to one room at a time.
- Each physician and nurse have unique IDs and certification numbers.
- The status of health records can be either 'ongoing' or 'resolved'.
- Each payable is linked to an invoice.
- The patient balance is tracked by subtracting payments from total invoices.

## 2 Enhanced Entity-Relationship Diagram (EERD)

## 3 Relations and Keys

- **Physician(physician\_id, name, certification\_number, field\_of\_expertise, address, phone)**
  - Primary Key: {physician\_id}
- **Nurse(nurse\_id, name, certification\_number, address, phone)**
  - Primary Key: {nurse\_id}
- **Room(room\_number, capacity, nightly\_fee)**
  - Primary Key: {room\_number}
- **Patient(patient\_id, name, address, phone, balance)**
  - Primary Key: {patient\_id}
- **HealthRecord(record\_id, patient\_id, disease, diagnosis\_date, status, description)**
  - Primary Key: {record\_id}
  - Foreign Key: {patient\_id} references **Patient(patient\_id)**
- **PatientRoom(assignment\_id, patient\_id, room\_number, check\_in\_date, check\_out\_date)**
  - Primary Key: {assignment\_id}

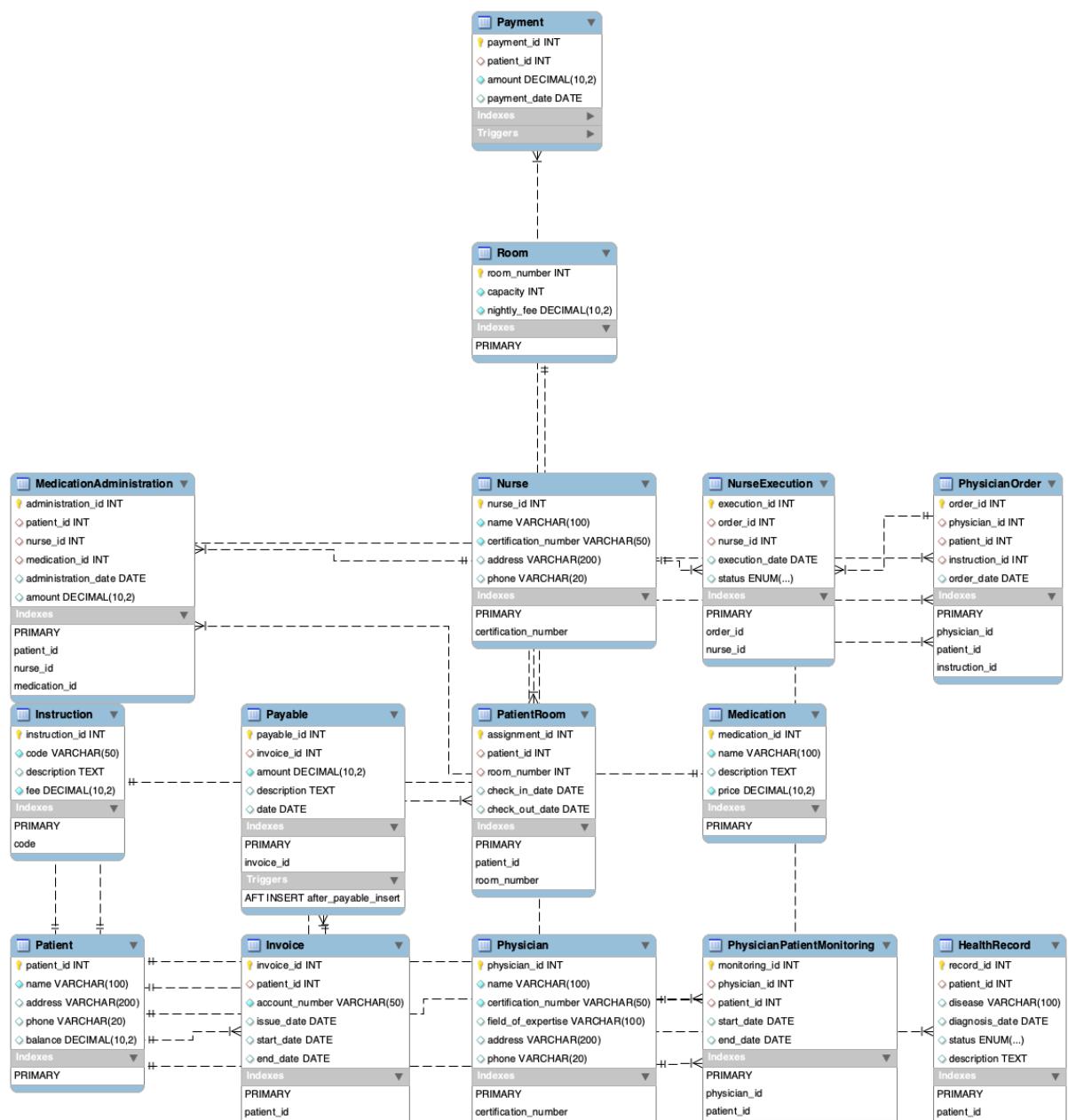


Figure 1: Enhanced Entity-Relationship Diagram for the Hospital Database

- Foreign Keys: {patient\_id} references **Patient(patient\_id)**, {room\_number} references **Room(room\_number)**
- **PhysicianPatientMonitoring(monitored\_id, physician\_id, patient\_id, start\_date, end\_date)**
  - Primary Key: {monitored\_id}
  - Foreign Keys: {physician\_id} references **Physician(physician\_id)**, {patient\_id} references **Patient(patient\_id)**
- **Medication(medication\_id, name, description, price)**
  - Primary Key: {medication\_id}
- **MedicationAdministration(administration\_id, patient\_id, nurse\_id, medication\_id, administration\_date, amount)**
  - Primary Key: {administration\_id}
  - Foreign Keys: {patient\_id} references **Patient(patient\_id)**, {nurse\_id} references **Nurse(nurse\_id)**, {medication\_id} references **Medication(medication\_id)**
- **Instruction(instruction\_id, code, description, fee)**
  - Primary Key: {instruction\_id}
- **PhysicianOrder(order\_id, physician\_id, patient\_id, instruction\_id, order\_date)**
  - Primary Key: {order\_id}
  - Foreign Keys: {physician\_id} references **Physician(physician\_id)**, {patient\_id} references **Patient(patient\_id)**, {instruction\_id} references **Instruction(instruction\_id)**
- **Payable(payable\_id, invoice\_id, amount, description, date)**
  - Primary Key: {payable\_id}
  - Foreign Key: {invoice\_id} references **Invoice(invoice\_id)**
- **Invoice(invoice\_id, patient\_id, account\_number, issue\_date, start\_date, end\_date)**
  - Primary Key: {invoice\_id}
  - Foreign Key: {patient\_id} references **Patient(patient\_id)**
- **Payment(payment\_id, patient\_id, amount, payment\_date)**
  - Primary Key: {payment\_id}
  - Foreign Key: {patient\_id} references **Patient(patient\_id)**

## 4 Views and Descriptions

### 4.1 View: Patient Billing

---

```
1 CREATE VIEW PatientBillingOverview AS
2 SELECT
3     p.patient_id,
4     p.name,
5     SUM(pa.amount) AS total_billed,
6     SUM(pay.amount) AS total_paid,
7     p.balance AS current_balance
8 FROM
9     Patient p
10 LEFT JOIN Invoice i ON p.patient_id = i.patient_id
11 LEFT JOIN Payable pa ON i.invoice_id = pa.invoice_id
12 LEFT JOIN Payment pay ON p.patient_id = pay.patient_id
13 GROUP BY p.patient_id;
```

---

**Description:** This view provides a comprehensive billing summary for each patient, including invoice details and total payments made.

### 4.2 View: Physician Performance

---

```
1 CREATE VIEW PhysicianPerformance AS
2 SELECT
3     ph.physician_id,
4     ph.name,
5     COUNT(DISTINCT po.patient_id) AS patients_treated,
6     COUNT(po.order_id) AS orders_given,
7     AVG(i.fee) AS avg_instruction_fee
8 FROM
9     Physician ph
10 LEFT JOIN PhysicianOrder po ON ph.physician_id = po.physician_id
11 LEFT JOIN Instruction i ON po.instruction_id = i.instruction_id
12 GROUP BY ph.physician_id;
```

---

**Description:** This view evaluates physician performance by counting the number of patients monitored and calculating the total revenue generated.

### 4.3 View: Room Occupancy Rate

---

```
1 CREATE VIEW RoomOccupancyRate AS
2 SELECT
```

```

3     r.room_number ,
4     r.capacity ,
5     COUNT(DISTINCT pr.patient_id) AS occupied_beds ,
6     (COUNT(DISTINCT pr.patient_id) / r.capacity) * 100 AS ←
      occupancy_rate
7 FROM
8     Room r
9 LEFT JOIN PatientRoom pr ON r.room_number = pr.room_number
10 WHERE
11     pr.check_out_date IS NULL OR pr.check_out_date >= CURDATE()
12 GROUP BY r.room_number;

```

---

**Description:** This view calculates the occupancy rate for each room, helping to monitor room utilization.

## 5 Triggers and Descriptions

### 5.1 Trigger: After Payable Insert

---

```

1 -- Triggers to update patient balance
2
3 DELIMITER //
4
5 CREATE TRIGGER after_payable_insert
6 AFTER INSERT ON Payable
7 FOR EACH ROW
8 BEGIN
9     UPDATE Patient p
10    SET p.balance = p.balance + NEW.amount
11   WHERE p.patient_id = (SELECT patient_id FROM Invoice WHERE ←
12     invoice_id = NEW.invoice_id);
13 END;//
14
15 DELIMITER ;

```

---

**Description:** This trigger updates the patient balance after a new payable is added, ensuring the balance reflects the latest charges.

### 5.2 Trigger: After Payment Insert

---

```

1 DELIMITER //
2
3 CREATE TRIGGER after_payment_insert

```

---

```

4 AFTER INSERT ON Payment
5 FOR EACH ROW
6 BEGIN
7     UPDATE Patient p
8     SET p.balance = p.balance - NEW.amount
9     WHERE p.patient_id = NEW.patient_id;
10 END;//
11
12 DELIMITER ;

```

---

**Description:** This trigger updates the patient balance after a payment is made, ensuring the balance reflects the latest payments.

### 5.3 Trigger: Prevent Room Overbooking

```

1 DELIMITER //
2 CREATE TRIGGER PreventOverBooking
3 BEFORE INSERT ON PatientRoom
4 FOR EACH ROW
5 BEGIN
6     DECLARE current_occupancy INT;
7     DECLARE room_capacity INT;
8
9     SELECT capacity INTO room_capacity
10    FROM Room
11   WHERE room_number = NEW.room_number;
12
13    SELECT COUNT(*) INTO current_occupancy
14    FROM PatientRoom
15   WHERE room_number = NEW.room_number
16   AND check_out_date >= NEW.check_in_date;
17
18    IF current_occupancy >= room_capacity THEN
19        SIGNAL SQLSTATE '45000'
20        SET MESSAGE_TEXT = 'Room is at full capacity for the specified←
21                           dates';
22    END IF;
23 END //
24 DELIMITER ;

```

---

**Description:** This trigger prevents overbooking of rooms by checking the current occupancy before a new assignment.

## 6 Queries, Descriptions, and Results

### 6.1 Query 1: Patient's Medication History

---

```
1 SELECT
2     p.name AS patient_name ,
3     m.name AS medication_name ,
4     ma.administration_date ,
5     n.name AS nurse_name
6 FROM
7     MedicationAdministration ma
8 JOIN Patient p ON ma.patient_id = p.patient_id
9 JOIN Medication m ON ma.medication_id = m.medication_id
10 JOIN Nurse n ON ma.nurse_id = n.nurse_id
11 ORDER BY ma.administration_date;
```

---

**Description:** Retrieves the medication history for each patient, including the patient's name, medication name, administration date, and the nurse who administered the medication.

### 6.2 Query 2: Total Medications Administered by Each Nurse

---

```
1 SELECT
2     n.name AS nurse_name ,
3     COUNT(ma.administration_id) AS total_medications_administered
4 FROM
5     MedicationAdministration ma
6 JOIN Nurse n ON ma.nurse_id = n.nurse_id
7 GROUP BY n.name
8 ORDER BY total_medications_administered DESC;
```

---

**Description:** Counts the total number of medications administered by each nurse and orders the results from highest to lowest.

### 6.3 Query 3: Patients with Above Average Payments

---

```
1 SELECT
2     p.name ,
3     SUM(pay.amount) AS total_payments
4 FROM
5     Patient p
6 JOIN Payment pay ON p.patient_id = pay.patient_id
7 GROUP BY p.name
```

---

```
8 HAVING total_payments > (SELECT AVG(total_payments) FROM (SELECT SUM(←
    amount) AS total_payments FROM Payment GROUP BY patient_id) AS ←
    subquery)
9 ORDER BY total_payments DESC;
```

---

**Description:** Identifies patients whose total payments are above the average payment amount.

## 6.4 Query 4: Physicians with Most Patients

```
1 SELECT
2     ph.name AS physician_name ,
3     COUNT(ppm.patient_id) AS total_patients
4 FROM
5     PhysicianPatientMonitoring ppm
6 JOIN Physician ph ON ppm.physician_id = ph.physician_id
7 GROUP BY ph.name
8 ORDER BY total_patients DESC;
```

---

**Description:** Finds the physicians who monitor the most patients, sorted by the number of patients in descending order.

## 6.5 Query 5: Total Revenue by Medication

```
1 SELECT
2     m.name AS medication_name ,
3     SUM(m.price) AS total_revenue
4 FROM
5     MedicationAdministration ma
6 JOIN Medication m ON ma.medication_id = m.medication_id
7 GROUP BY m.name
8 ORDER BY total_revenue DESC;
```

---

**Description:** Calculates the total revenue generated by each medication and orders the results by total revenue in descending order.

## 6.6 Query 6: Rooms with Above Average Occupancy Rate

```
1 SELECT
2     r.room_number ,
3     COUNT(pr.assignment_id) AS total_occupancy
4 FROM
5     PatientRoom pr
```

---

```
6 JOIN Room r ON pr.room_number = r.room_number
7 GROUP BY r.room_number
8 HAVING total_occupancy > (SELECT AVG(total_occupancy) FROM (SELECT ←
    COUNT(assignment_id) AS total_occupancy FROM PatientRoom GROUP BY ←
    room_number) AS subquery)
9 ORDER BY total_occupancy DESC;
```

---

**Description:** Identifies rooms that have an occupancy rate above the average occupancy rate.

## 6.7 Query 7: Patients and Their Total Expenses

```
1 SELECT
2     p.name AS patient_name,
3     SUM(pay.amount) AS total_expenses
4 FROM
5     Payment pay
6 JOIN Patient p ON pay.patient_id = p.patient_id
7 GROUP BY p.name
8 ORDER BY total_expenses DESC;
```

---

**Description:** Retrieves each patient and their total expenses incurred, sorted by the highest total expenses.

## 6.8 Query 8: Average Length of Stay per Room

```
1 SELECT
2     r.room_number,
3     AVG(DATEDIFF(pr.check_out_date, pr.check_in_date)) AS ←
        avg_length_of_stay
4 FROM
5     PatientRoom pr
6 JOIN Room r ON pr.room_number = r.room_number
7 GROUP BY r.room_number
8 ORDER BY avg_length_of_stay DESC;
```

---

**Description:** Calculates the average length of stay for each room and orders the results from longest to shortest average stay.

## 6.9 Query 9: Physicians Treating Most Chronic Cases

```
1 SELECT
2     ph.name AS physician_name,
```

```
3     COUNT(hr.record_id) AS chronic_cases
4 FROM
5     HealthRecord hr
6 JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id
7 JOIN Physician ph ON ppm.physician_id = ph.physician_id
8 WHERE hr.status = 'ongoing'
9 GROUP BY ph.name
10 ORDER BY chronic_cases DESC;
```

---

**Description:** Identifies physicians who are treating the most chronic cases (ongoing health records).

## 6.10 Query 10: Total Revenue by Physician

```
1 SELECT
2     ph.name AS physician_name ,
3     SUM(i.fee) AS total_revenue
4 FROM
5     PhysicianOrder po
6 JOIN Physician ph ON po.physician_id = ph.physician_id
7 JOIN Instruction i ON po.instruction_id = i.instruction_id
8 GROUP BY ph.name
9 ORDER BY total_revenue DESC;
```

---

**Description:** Calculates the total revenue generated by each physician based on the fees of the instructions they have given.

## 6.11 Query 11: Total Number of Treatments per Department

```
1 SELECT
2     ph.field_of_expertise AS department ,
3     COUNT(po.order_id) AS total_treatments
4 FROM
5     PhysicianOrder po
6 JOIN Physician ph ON po.physician_id = ph.physician_id
7 GROUP BY ph.field_of_expertise
8 ORDER BY total_treatments DESC;
```

---

**Description:** Finds the total number of treatments ordered per department, sorted by the highest number of treatments.

## 6.12 Query 12: Patients with Highest Number of Medications

---

```
1 SELECT
2     p.name ,
3     COUNT(ma.administration_id) AS num_medications
4 FROM
5     MedicationAdministration ma
6 JOIN Patient p ON ma.patient_id = p.patient_id
7 GROUP BY p.name
8 ORDER BY num_medications DESC
9 LIMIT 5;
```

---

**Description:** Retrieves the top 5 patients who have received the highest number of medications.

### 6.13 Query 13: Total Revenue by Department

---

```
1 SELECT
2     ph.field_of_expertise AS department ,
3     SUM(i.fee) AS total_revenue
4 FROM
5     PhysicianOrder po
6 JOIN Physician ph ON po.physician_id = ph.physician_id
7 JOIN Instruction i ON po.instruction_id = i.instruction_id
8 GROUP BY ph.field_of_expertise
9 ORDER BY total_revenue DESC;
```

---

**Description:** Calculates the total revenue generated by each department based on the fees of the treatments they have given.

### 6.14 Query 14: Patient Treatment Summary

---

```
1 SELECT
2     p.name AS patient_name ,
3     ph.name AS physician_name ,
4     i.description AS treatment ,
5     po.order_date
6 FROM
7     Patient p
8 JOIN PhysicianOrder po ON p.patient_id = po.patient_id
9 JOIN Physician ph ON po.physician_id = ph.physician_id
10 JOIN Instruction i ON po.instruction_id = i.instruction_id
11 ORDER BY p.name , po.order_date;
```

---

**Description:** Provides a summary of the different treatments and medications a patient had

been given.

## 6.15 Query 15: Patients with Above Average Stays

---

```
1 SELECT
2     p.name ,
3     DATEDIFF(pr.check_out_date , pr.check_in_date) AS length_of_stay
4 FROM
5     Patient p
6 JOIN PatientRoom pr ON p.patient_id = pr.patient_id
7 WHERE
8     DATEDIFF(pr.check_out_date , pr.check_in_date) >
9     (SELECT AVG(DATEDIFF(check_out_date , check_in_date))
10    FROM PatientRoom)
11 ORDER BY length_of_stay DESC;
```

---

**Description:** Retrieves the patients who have stayed longer than the average nights at the hospital in descending order.

## 7 Transactions and Descriptions

### 7.1 Transaction: Patient Admission

---

```
1 START TRANSACTION;
2
3 INSERT INTO Patient (patient_id, name, address, phone)
4 VALUES (6, 'Frank Williams', '600 New St, Newtown', '555-6666');
5
6 INSERT INTO HealthRecord (record_id, patient_id, disease, ←
7     diagnosis_date, status, description)
8 VALUES (6, 6, 'Influenza', CURDATE(), 'ongoing', 'Patient admitted ←
9     with severe flu symptoms');
10
11 INSERT INTO PatientRoom (assignment_id, patient_id, room_number, ←
12     check_in_date, check_out_date)
13 VALUES (6, 6, 101, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 5 DAY));
14
15 COMMIT;
```

---

**Description:** This transaction inserts a new patient and their initial health record, ensuring atomicity and consistency.

## 7.2 Transaction: Patient Discharge

---

```
1 START TRANSACTION;
2
3 UPDATE PatientRoom
4 SET check_out_date = CURDATE()
5 WHERE patient_id = 1 AND check_out_date IS NULL;
6
7 INSERT INTO Invoice (invoice_id, patient_id, account_number, ←
     issue_date, start_date, end_date)
8 VALUES (6, 1, 'ACCO06', CURDATE(),
9          (SELECT check_in_date FROM PatientRoom WHERE patient_id = 1 ←
               ORDER BY check_in_date DESC LIMIT 1),
10         CURDATE());
11
12 INSERT INTO Payable (payable_id, invoice_id, amount, description, date←
     )
13 SELECT
14     (SELECT COALESCE(MAX(payable_id), 0) FROM Payable) + ROW_NUMBER() ←
         OVER () AS payable_id,
15     6 AS invoice_id,
16     CASE
17         WHEN p.type = 'room' THEN r.nightly_fee * DATEDIFF(pr.←
             check_out_date, pr.check_in_date)
18         WHEN p.type = 'medication' THEN m.price * ma.amount
19         WHEN p.type = 'instruction' THEN i.fee
20     END AS amount,
21     p.description,
22     p.date
23 FROM (
24     SELECT 'room' AS type, pr.check_in_date AS date, CONCAT('Room ←
          charge: ', pr.room_number) AS description
25     FROM PatientRoom pr WHERE pr.patient_id = 1
26     UNION ALL
27     SELECT 'medication' AS type, ma.administration_date AS date, ←
          CONCAT('Medication: ', m.name) AS description
28     FROM MedicationAdministration ma JOIN Medication m ON ma.←
          medication_id = m.medication_id
29     WHERE ma.patient_id = 1
30     UNION ALL
31     SELECT 'instruction' AS type, po.order_date AS date, i.description
32     FROM PhysicianOrder po JOIN Instruction i ON po.instruction_id = i.←
          instruction_id
33     WHERE po.patient_id = 1
34 ) p
```

```

35 LEFT JOIN PatientRoom pr ON p.type = 'room' AND p.date = pr.←
    check_in_date AND pr.patient_id = 1
36 LEFT JOIN Room r ON pr.room_number = r.room_number
37 LEFT JOIN MedicationAdministration ma ON p.type = 'medication' AND p.←
    date = ma.administration_date AND ma.patient_id = 1
38 LEFT JOIN Medication m ON ma.medication_id = m.medication_id
39 LEFT JOIN PhysicianOrder po ON p.type = 'instruction' AND p.date = po.←
    order_date AND po.patient_id = 1
40 LEFT JOIN Instruction i ON po.instruction_id = i.instruction_id;
41
42 COMMIT;

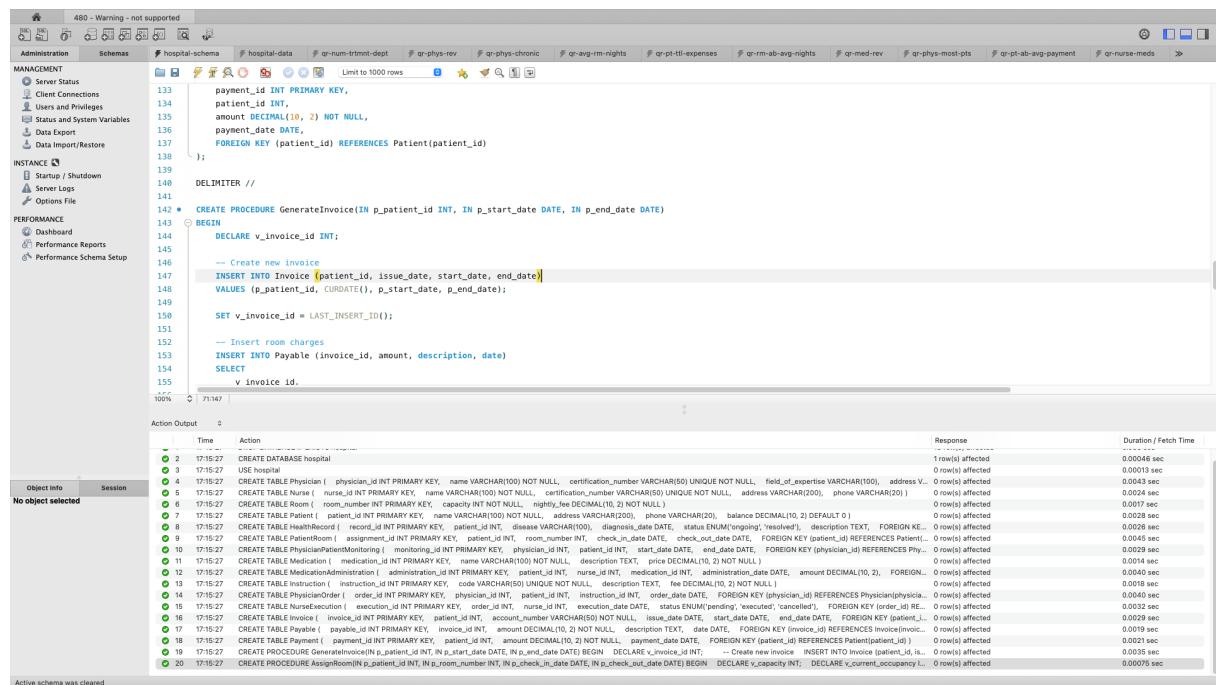
```

---

**Description:** This transaction updates the check-out date for a patient and issues an invoice for the hospitalization period.

## A Appendix

### A.1 SQL File: hospital-schema



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

- SQL Editor:** Displays the SQL code for creating the hospital schema, including tables like Payment, Invoice, Room, Patient, Physician, Nurse, MedicationAdministration, and PhysicianOrder, along with their respective columns and constraints.
- Results Pane:** Shows the execution log with 20 rows affected, indicating successful creation of all schema objects.
- Object Info:** Shows the session information for the current connection.
- Action Output:** Shows the execution log with 20 rows affected.

```

480 - Warning - not supported
MANAGEMENT
    ○ Server Status
    ○ Client Connections
    ○ Users and Privileges
    ○ Status and System Variables
    ○ Data Export
    ○ Data Import/Restore
INSTANCE
    ○ Startup / Shutdown
    ○ Server Logs
    ○ Options File
PERFORMANCE
    ○ Dashboard
    ○ Performance Reports
    ○ Performance Schema Setup

```

Time	Action	Response	Duration / Fetch Time
2 17:15:27	CREATE DATABASE hospital	1 row(s) affected	0.00006 sec
3 17:15:27	USE hospital	0 row(s) affected	0.00013 sec
4 17:15:27	CREATE TABLE Physician ( physician_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, certification_number VARCHAR(50) UNIQUE NOT NULL, field_of_expertise VARCHAR(100), address VARCHAR(200), phone VARCHAR(20))	0 row(s) affected	0.0045 sec
5 17:15:27	CREATE TABLE Nurse ( nurse_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, certification_number VARCHAR(50) UNIQUE NOT NULL, address VARCHAR(200), phone VARCHAR(20))	0 row(s) affected	0.0024 sec
6 17:15:27	CREATE TABLE Room ( room_number INT PRIMARY KEY, capacity INT NOT NULL, nightly_fee DECIMAL(10, 2) NOT NULL)	0 row(s) affected	0.0017 sec
7 17:15:27	CREATE TABLE Patient ( patient_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, address VARCHAR(200), phone VARCHAR(20), balance DECIMAL(10, 2) DEFAULT 0 )	0 row(s) affected	0.0028 sec
8 17:15:27	CREATE TABLE MedicationAdministration ( administration_id INT PRIMARY KEY, patient_id INT, medication_id INT, administration_date DATE, FOREIGN KEY (patient_id) REFERENCES Patient(patient_id), FOREIGN KEY (medication_id) REFERENCES Medication(medication_id))	0 row(s) affected	0.0026 sec
9 17:15:27	CREATE TABLE PhysicianPatientMonitoring ( monitoring_id INT PRIMARY KEY, physician_id INT, patient_id INT, start_date DATE, end_date DATE, FOREIGN KEY (physician_id) REFERENCES Physician(physician_id), FOREIGN KEY (patient_id) REFERENCES Patient(patient_id))	0 row(s) affected	0.0045 sec
10 17:15:27	CREATE TABLE Medication ( medication_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, description TEXT, price DECIMAL(10, 2) NOT NULL )	0 row(s) affected	0.0014 sec
11 17:15:27	CREATE TABLE RoomAssignment ( assignment_id INT PRIMARY KEY, room_id INT, room_number INT, check_in_date DATE, check_out_date DATE)	0 row(s) affected	0.0029 sec
12 17:15:27	CREATE TABLE MedicationAdministration ( administration_id INT PRIMARY KEY, patient_id INT, nurse_id INT, medication_id INT, administration_date DATE, amount DECIMAL(10, 2), FOREIGN KEY (patient_id) REFERENCES Patient(patient_id), FOREIGN KEY (nurse_id) REFERENCES Nurse(nurse_id), FOREIGN KEY (medication_id) REFERENCES Medication(medication_id))	0 row(s) affected	0.0040 sec
13 17:15:27	CREATE TABLE PhysicianPatientMonitoring ( monitoring_id INT PRIMARY KEY, physician_id INT, patient_id INT, start_date DATE, end_date DATE, fee DECIMAL(10, 2) NOT NULL, FOREIGN KEY (physician_id) REFERENCES Physician(physician_id), FOREIGN KEY (patient_id) REFERENCES Patient(patient_id))	0 row(s) affected	0.0018 sec
14 17:15:27	CREATE TABLE Payment ( payment_id INT PRIMARY KEY, account_number VARCHAR(50) NOT NULL, issue_date DATE, start_date DATE, end_date DATE, FOREIGN KEY (patient_id) REFERENCES Patient(patient_id), FOREIGN KEY (nurse_id) REFERENCES Nurse(nurse_id), FOREIGN KEY (medication_id) REFERENCES Medication(medication_id))	0 row(s) affected	0.0040 sec
15 17:15:27	CREATE TABLE Invoice ( invoice_id INT PRIMARY KEY, patient_id INT, account_number VARCHAR(50) NOT NULL, issue_date DATE, start_date DATE, end_date DATE, FOREIGN KEY (patient_id) REFERENCES Patient(patient_id), FOREIGN KEY (account_number) REFERENCES Payment(payment_id))	0 row(s) affected	0.0033 sec
16 17:15:27	CREATE TABLE NurseExecution ( execution_id INT PRIMARY KEY, order_id INT, nurse_id INT, execution_date DATE, status ENUM('pending', 'executed'))	0 row(s) affected	0.0040 sec
17 17:15:27	CREATE PROCEDURE GenerateInvoice(IN p_patient_id INT, IN p_start_date DATE, IN p_end_date DATE) BEGIN DECLARE v_invoice_id INT; -- Create new invoice INSERT INTO Invoice (patient_id, issue_date, start_date, end_date) VALUES (p_patient_id, CURDATE(), p_start_date, p_end_date); SET v_invoice_id = LAST_INSERT_ID();	0 row(s) affected	0.0035 sec
18 17:15:27	CREATE TABLE Payment ( payment_id INT PRIMARY KEY, patient_id INT, amount DECIMAL(10, 2) NOT NULL, payment_date DATE, FOREIGN KEY (patient_id) REFERENCES Patient(patient_id))	0 row(s) affected	0.0021 sec
19 17:15:27	CREATE PROCEDURE GenerateInvoice(IN p_patient_id INT, IN p_start_date DATE, IN p_end_date DATE) BEGIN DECLARE v_invoice_id INT; -- Create new invoice INSERT INTO Invoice (patient_id, issue_date, start_date, end_date) VALUES (p_patient_id, CURDATE(), p_start_date, p_end_date); SET v_invoice_id = LAST_INSERT_ID();	0 row(s) affected	0.00079 sec

## A.2 SQL File: hospital-data

```

480 - Warning - not supported

Administration Schemas # hospital-schema # hospital-data # qr-num-trmt-dept # qr-phys-rev # qr-phs-chronic # qr-avg-rr-nights # qr-pt-tl-expenses # qr-rm-ab-avg-nights # qr-med-rev # qr-phys-most-pts # qr-pt-ab-avg-payment # qr-nurse-meds >

MANAGEMENT
  Server Status
  Cloud Connections
  Users and Privileges
  Status and System Variables
    Data Export
    Data Import/Restore
INSTANCE
  Startup / Shutdown
  Server Logs
  Options File
PERFORMANCE
  Dashboard
  Performance Reports
  Performance Schema Setup

100% 19:109

98 | (4, 4, 4, 4, '2024-07-05'),  

99 | (5, 5, 5, '2024-07-06');  

100  

101 -- Insert Nurse Executions  

102 • INSERT INTO NurseExecution VALUES  

103 | (1, 1, 1, '2024-07-02', 'executed'),  

104 | (2, 2, 2, '2024-07-03', 'executed'),  

105 | (3, 3, 3, '2024-07-04', 'pending'),  

106 | (4, 4, 4, '2024-07-05', 'executed'),  

107 | (5, 5, 5, '2024-07-06', 'pending');  

108  

109 -- Insert Invoices  

110 • INSERT INTO Invoice VALUES  

111 | (1, 1, 400.00, 'Room charge: 101 for 4 nights', '2024-07-05'),  

112 | (2, 1, 50.00, 'Medication: Lisinopril', '2024-07-02'),  

113  

114  

115  

116  

117  

118  

119  

120  

121  

122  

123  

124  

125  

126  

127  

128  

129  

130  

131  

132  

133  

134  

135  

136  

137  

138  

139  

140  

141  

142  

143  

144  

145  

146  

147  

148  

149  

150  

151  

152  

153  

154  

155  

156  

157  

158  

159  

160  

161  

162  

163  

164  

165  

166  

167  

168  

169  

170  

171  

172  

173  

174  

175  

176  

177  

178  

179  

180  

181  

182  

183  

184  

185  

186  

187  

188  

189  

190  

191  

192  

193  

194  

195  

196  

197  

198  

199  

200  

201  

202  

203  

204  

205  

206  

207  

208  

209  

210  

211  

212  

213  

214  

215  

216  

217  

218  

219  

220  

221  

222  

223  

224  

225  

226  

227  

228  

229  

230  

231  

232  

233  

234  

235  

236  

237  

238  

239  

240  

241  

242  

243  

244  

245  

246  

247  

248  

249  

250  

251  

252  

253  

254  

255  

256  

257  

258  

259  

260  

261  

262  

263  

264  

265  

266  

267  

268  

269  

270  

271  

272  

273  

274  

275  

276  

277  

278  

279  

280  

281  

282  

283  

284  

285  

286  

287  

288  

289  

290  

291  

292  

293  

294  

295  

296  

297  

298  

299  

300  

301  

302  

303  

304  

305  

306  

307  

308  

309  

310  

311  

312  

313  

314  

315  

316  

317  

318  

319  

320  

321  

322  

323  

324  

325  

326  

327  

328  

329  

330  

331  

332  

333  

334  

335  

336  

337  

338  

339  

340  

341  

342  

343  

344  

345  

346  

347  

348  

349  

350  

351  

352  

353  

354  

355  

356  

357  

358  

359  

360  

361  

362  

363  

364  

365  

366  

367  

368  

369  

370  

371  

372  

373  

374  

375  

376  

377  

378  

379  

380  

381  

382  

383  

384  

385  

386  

387  

388  

389  

390  

391  

392  

393  

394  

395  

396  

397  

398  

399  

400  

401  

402  

403  

404  

405  

406  

407  

408  

409  

410  

411  

412  

413  

414  

415  

416  

417  

418  

419  

420  

421  

422  

423  

424  

425  

426  

427  

428  

429  

430  

431  

432  

433  

434  

435  

436  

437  

438  

439  

440  

441  

442  

443  

444  

445  

446  

447  

448  

449  

450  

451  

452  

453  

454  

455  

456  

457  

458  

459  

460  

461  

462  

463  

464  

465  

466  

467  

468  

469  

470  

471  

472  

473  

474  

475  

476  

477  

478  

479  

480  

481  

482  

483  

484  

485  

486  

487  

488  

489  

490  

491  

492  

493  

494  

495  

496  

497  

498  

499  

500  

501  

502  

503  

504  

505  

506  

507  

508  

509  

510  

511  

512  

513  

514  

515  

516  

517  

518  

519  

520  

521  

522  

523  

524  

525  

526  

527  

528  

529  

530  

531  

532  

533  

534  

535  

536  

537  

538  

539  

540  

541  

542  

543  

544  

545  

546  

547  

548  

549  

550  

551  

552  

553  

554  

555  

556  

557  

558  

559  

559  

560  

561  

562  

563  

564  

565  

566  

567  

568  

569  

570  

571  

572  

573  

574  

575  

576  

577  

578  

579  

580  

581  

582  

583  

584  

585  

586  

587  

588  

589  

589  

590  

591  

592  

593  

594  

595  

596  

597  

598  

599  

600  

601  

602  

603  

604  

605  

606  

607  

608  

609  

610  

611  

612  

613  

614  

615  

616  

617  

618  

619  

620  

621  

622  

623  

624  

625  

626  

627  

628  

629  

630  

631  

632  

633  

634  

635  

636  

637  

638  

639  

640  

641  

642  

643  

644  

645  

646  

647  

648  

649  

650  

651  

652  

653  

654  

655  

656  

657  

658  

659  

659  

660  

661  

662  

663  

664  

665  

666  

667  

668  

669  

669  

670  

671  

672  

673  

674  

675  

676  

677  

678  

679  

680  

681  

682  

683  

684  

685  

686  

687  

688  

689  

689  

690  

691  

692  

693  

694  

695  

696  

697  

698  

699  

700  

701  

702  

703  

704  

705  

706  

707  

708  

709  

709  

710  

711  

712  

713  

714  

715  

716  

717  

718  

719  

720  

721  

722  

723  

724  

725  

726  

727  

728  

729  

729  

730  

731  

732  

733  

734  

735  

736  

737  

738  

739  

739  

740  

741  

742  

743  

744  

745  

746  

747  

748  

749  

749  

750  

751  

752  

753  

754  

755  

756  

757  

758  

759  

759  

760  

761  

762  

763  

764  

765  

766  

767  

768  

769  

769  

770  

771  

772  

773  

774  

775  

776  

777  

778  

779  

779  

780  

781  

782  

783  

784  

785  

786  

787  

788  

789  

789  

790  

791  

792  

793  

794  

795  

796  

797  

798  

799  

800  

801  

802  

803  

804  

805  

806  

807  

808  

809  

809  

810  

811  

812  

813  

814  

815  

816  

817  

818  

819  

819  

820  

821  

822  

823  

824  

825  

826  

827  

828  

829  

829  

830  

831  

832  

833  

834  

835  

836  

837  

838  

839  

839  

840  

841  

842  

843  

844  

845  

846  

847  

848  

849  

849  

850  

851  

852  

853  

854  

855  

856  

857  

858  

859  

859  

860  

861  

862  

863  

864  

865  

866  

867  

868  

869  

869  

870  

871  

872  

873  

874  

875  

876  

877  

878  

879  

879  

880  

881  

882  

883  

884  

885  

886  

887  

888  

889  

889  

890  

891  

892  

893  

894  

895  

896  

897  

898  

899  

900  

901  

902  

903  

904  

905  

906  

907  

908  

909  

910  

911  

912  

913  

914  

915  

916  

917  

918  

919  

920  

921  

922  

923  

924  

925  

926  

927  

928  

929  

930  

931  

932  

933  

934  

935  

936  

937  

938  

939  

940  

941  

942  

943  

944  

945  

946  

947  

948  

949  

950  

951  

952  

953  

954  

955  

956  

957  

958  

959  

959  

960  

961  

962  

963  

964  

965  

966  

967  

968  

969  

969  

970  

971  

972  

973  

974  

975  

976  

977  

978  

979  

979  

980  

981  

982  

983  

984  

985  

986  

987  

988  

989  

989  

990  

991  

992  

993  

994  

995  

996  

997  

998  

999  

1000  

1001  

1002  

1003  

1004  

1005  

1006  

1007  

1008  

1009  

1009  

1010  

1011  

1012  

1013  

1014  

1015  

1016  

1017  

1018  

1019  

1019  

1020  

1021  

1022  

1023  

1024  

1025  

1026  

1027  

1028  

1029  

1029  

1030  

1031  

1032  

1033  

1034  

1035  

1036  

1037  

1038  

1039  

1039  

1040  

1041  

1042  

1043  

1044  

1045  

1046  

1047  

1048  

1049  

1049  

1050  

1051  

1052  

1053  

1054  

1055  

1056  

1057  

1058  

1059  

1059  

1060  

1061  

1062  

1063  

1064  

1065  

1066  

1067  

1068  

1069  

1069  

1070  

1071  

1072  

1073  

1074  

1075  

1076  

1077  

1078  

1079  

1079  

1080  

1081  

1082  

1083  

1084  

1085  

1086  

1087  

1088  

1089  

1089  

1090  

1091  

1092  

1093  

1094  

1095  

1096  

1097  

1098  

1099  

1100  

1101  

1102  

1103  

1104  

1105  

1106  

1107  

1108  

1109  

1109  

1110  

1111  

1112  

1113  

1114  

1115  

1116  

1117  

1118  

1119  

1119  

1120  

1121  

1122  

1123  

1124  

1125  

1126  

1127  

1128  

1129  

1129  

1130  

1131  

1132  

1133  

1134  

1135  

1136  

1137  

1138  

1139  

1139  

1140  

1141  

1142  

1143  

1144  

1145  

1146  

1147  

1148  

1149  

1149  

1150  

1151  

1152  

1153  

1154  

1155  

1156  

1157  

1158  

1159  

1159  

1160  

1161  

1162  

1163  

1164  

1165  

1166  

1167  

1168  

1169  

1169  

1170  

1171  

1172  

1173  

1174  

1175  

1176  

1177  

1178  

1179  

1179  

1180  

1181  

1182  

1183  

1184  

1185  

1186  

1187  

1188  

1189  

1189  

1190  

1191  

1192  

1193  

1194  

1195  

1196  

1197  

1198  

1199  

1199  

1200  

1201  

1202  

1203  

1204  

1205  

1206  

1207  

1208  

1209  

1209  

1210  

1211  

1212  

1213  

1214  

1215  

1216  

1217  

1218  

1219  

1219  

1220  

1221  

1222  

1223  

1224  

1225  

1226  

1227  

1228  

1229  

1229  

1230  

1231  

1232  

1233  

1234  

1235  

1236  

1237  

1238  

1239  

1239  

1240  

1241  

1242  

1243  

1244  

1245  

1246  

1247  

1248  

1249  

1249  

1250  

1251  

1252  

1253  

1254  

1255  

1256  

1257  

1258  

1259  

1259  

1260  

1261  

1262  

1263  

1264  

1265  

1266  

1267  

1268  

1269  

1269  

1270  

1271  

1272  

1273  

1274  

1275  

1276  

1277  

1278  

1279  

1279  

1280  

1281  

1282  

1283  

1284  

1285  

1286  

1287  

1288  

1289  

1289  

1290  

1291  

1292  

1293  

1294  

1295  

1296  

1297  

1298  

1299  

1299  

1300  

1301  

1302  

1303  

1304  

1305  

1306  

1307  

1308  

1309  

1309  

1310  

1311  

1312  

1313  

1314  

1315  

1316  

1317  

1318  

1319  

1319  

1320  

1321  

1322  

1323  

1324  

1325  

1326  

1327  

1328  

1329  

1329  

1330  

1331  

1332  

1333  

1334  

1335  

1336  

1337  

1338  

1339  

1339  

1340  

1341  

1342  

1343  

1344  

1345  

1346  

1347  

1348  

1349  

1349  

1350  

1351  

1352  

1353  

1354  

1355  

1356  

1357  

1358  

1359  

1359  

1360  

1361  

1362  

1363  

1364  

1365  

1366  

1367  

1368  

1369  

1369  

1370  

1371  

1372  

1373  

1374  

1375  

1376  

1377  

1378  

1379  

1379  

1380  

1381  

1382  

1383  

1384  

1385  

1386  

1387  

1388  

1389  

1389  

1390  

1391  

1392  

1393  

1394  

1395  

1396  

1397  

1398  

1399  

1399  

1400  

1401  

1402  

1403  

1404  

1405  

1406  

1407  

1408  

1409  

1410  

1411  

1412  

1413  

1414  

1415  

1416  

1417  

1418  

1419  

1419  

1420  

1421  

1422  

1423  

1424  

1425  

1426  

1427  

1428  

1429  

1429  

1430  

1431  

1432  

1433  

1434  

1435  

1436  

1437  

1438  

1439  

1439  

1440  

1441  

1442  

1443  

1444  

1445  

1446  

1447  

1448  

1449  

1449  

1450  

1451  

1452  

1453  

1454  

1455  

1456  

1457  

1458  

1459  

1459  

1460  

1461  

1462  

1463  

1464  

1465  

1466  

1467  

1468  

1469  

1469  

1470  

1471  

1472  

1473  

1474  

1475  

1476  

1477  

1478  

1479  

1479  

1480  

1481  

1482  

1483  

1484  

1485  

1486  

1487  

1488  

1489  

1489  

1490  

1491  

1492  

1493  

1494  

1495  

1496  

1497  

1498  

1499  

1499  

1500  

1501  

1502  

1503  

1504  

1505  

1506  

1507  

1508  

1509  

1509  

1510  

1511  

1512  

1513  

1514  

1515  

1516  

1517  

1518  

1519  

1519  

1520  

1521  

1522  

1523  

1524  

1525  

1526  

1527  

1528  

1529  

1529  

1530  

1531  

1532  

1533  

1534  

1535  

1536  

1537  

1538  

1539  

1539  

1540  

1541  

1542  

1543  

1544  

1545  

1546  

1547  

1548  

1549  

1549  

1550  

1551  

1552  

1553  

1554  

1555  

1556  

1557  

1558  

1559  

1559  

1560  

1561  

1562  

1563  

1564  

1565  

1566  

1567  

1568  

1569  

1569  

1570  

1571  

1572  

1573  

1574  

1575  

1576  

1577  

1578  

1579  

1579  

1580  

1581  

1582  

1583  

1584  

1585  

1586  

1587  

1588  

1589  

1589  

1590  

1591  

1592  

1593  

1594  

1595  

1596  

1597  

1598  

1599  

1599  

1600  

1601  

1602  

1603  

1604  

1605  

1606  

1607  

1608  

1609  

1609  

1610  

1611  

1612  

1613  

1614  

1615  

1616  

1617  

1618  

1619  

1619  

1620  

1621  

1622  

1623  

1624  

1625  

1626  

1627  

1628  

1629  

1629  

1630  

1631  

1632  

1633  

1634  

1635  

1636  

1637  

1638  

1639  

1639  

1640  

1641  

1642  

1643  

1644  

1645  

1646  

1647  

1648  

1649  

1649  

1650  

1651  

1652  

1653  

1654  

1655  

1656  

1657  

1658  

1659  

1659  

1660  

1661  

1662  

1663  

1664  

1665  

1666  

1667  

1668  

1669  

1669  

1670  

1671  

1672  

1673  

1674  

1675  

1676  

1677  

1678  

1679  

1679  

1680  

1681  

1682  

1683  

1684  

1685  

1686  

1687  

1688  

1689  

1689  

1690  

1691  

1692  

1693  

1694  

1695  

1696  

1697  

1698  

1699  

1699  

1700  

1701  

1702  

1703  

1704  

1705  

1706  

1707  

1708  

1709  

1709  

1710  

1711  

1712  

1713  

1714  

1715  

1716  

1717  

1718  

1719  

1719  

1720  

1721  

1722  

1723  

1724  

1725  

1726  

1727  

1728  

1729  

1729  

1730  

1731  

1732  

1733  

1734  

1735  

1736  

1737  

1738  

1739  

1739  

1740  

1741  

1742  

1743  

1744  

1745  

1746  

1747  

1748  

1749  

1749  

1750  

1751  

1752  

1753  

1754  

1755  

1756  

1757  

1758  

1759  

1759  

1760  

1761  

1762  

1763  

1764  

1765  

1766  

1767  

1768  

1769  

1769  

1770  

1771  

1772  

1773  

1774  

1775  

1776  

1777  

1778  

1779  

1779  

1780  

1781  

1782  

1783  

1784  

1785  

1786  

1787  

1788  

1789  

1789  

1790  

1791  

1792  

1793  

1794  

1795  

1796  

1797  

1798  

1799  

1799  

1800  

1801  

1802  

1803  

1804  

1805  

1806  

1807  

1808  

1809  

1809  

1810  

1811  

1812  

1813  

1814  

1815  

1816  

1817  

1818  

1819  

1819  

1820  

1821  

1822  

1823  

1824  

1825  

1826  

1827  

1828  

1829  

1829  

1830  

1831  

1832  

1833  

1834  

1835  

1836  

1837  

1838  

1839  

1839  

1840  

1841  

1842  

1843  

1844  

1845  

1846  

1847  

1848  

1849  

1849  

1850  

1851  

1852  

1853  

1854  

1855  

1856  

1857  

1858  

1859  

1859  

1860  

1861  

1862  

1863  

1864  

1865  

1866  

1867  

1868  

1869  

1869  

1870  

1871  

1872  

1873  

1874  

1875  

1876  

1877  

1878  

1879  

1879  

1880  

1881  

1882  

1883  

1884  

1885  

1886  

1887  

1888  

1889  

1889  

1890  

1891  

1892  

1893  

1894  

1895  

1896  

1897  

1898  

1899  

1899  

1900  

1901  

1902  

1903  

1904  

1905  

1906  

1907  

1908
```

## A.4 SQL View: view-patient-billing

480 - Warning - not supported

Administration Schemas view-patient-billing view-physician-performance view-room-occupancy-rate trig-stop-overbooking trig-after-payable trig-pt-treatment-summary qr-dept-rev qr-pt-ab-avg-stay trans-pt-admit

Schemas Filter objects Limit to 1000 rows

**View: patientbillingoverv...**

```

1 • CREATE VIEW PatientBillingOverview AS
2
3     SELECT
4         p.patient_id,
5         p.name,
6         SUM(pa.amount) AS total_billed,
7         SUM(pa.amount) AS total_paid,
8         p.balance AS current_balance
9
10    FROM
11        Patient p
12    LEFT JOIN Invoice i ON p.patient_id = i.patient_id
13    LEFT JOIN Payable pa ON i.invoice_id = pa.invoice_id
14    LEFT JOIN Payment pay ON p.patient_id = pay.patient_id
15    GROUP BY p.patient_id;

```

Action Output 0

Object Info	Session	Time	Action	Response	Duration / Fetch Time
View: patientbillingoverv...		30 17:15:56	INSERT INTO MedicationAdministration VALUES (1, 1, 5, '2024-07-02', 1, 2, 2, 5, '2024-07-09', 2), (3, 3, 3, '2024-07-04', 1), (4, 4, 4, 4, 2024-07-05), (5, 5, 5, 1, '2024-07-06', 2)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0018 sec
Comments:		31 17:15:56	INSERT INTO Instruction VALUES (1, 'CB001', 'Complete blood count; 50,000, (2, 'XRAY001', 'X-ray; 100,00, (3, 'MRI001', 'MRI scan; 500,00, (4, 'PT001', 'Physical therapy session; 75,00, (5, 'NUTR001', 'Nutrition consultation; 100,00, (6, 'ACCD001', '2024-07-07', 2, 2, 2, 2, 2024-07-09), (3, 3, 3, 2024-07-04), (4, 4, 4, 2024-07-05), (5, 5, 5, 2024-07-06), 1)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
Columns:		32 17:15:56	INSERT INTO PhysicianOrder VALUES (1, 1, 1, 2024-07-02), (2, 2, 2, 2, 2024-07-03), (3, 3, 3, 2024-07-04), (4, 4, 4, 2024-07-05), (5, 5, 5, 2024-07-06)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0014 sec
Object Info	Session	33 17:15:56	INSERT INTO NurseExecution VALUES (1, 1, 1, 2024-07-02, 'executed'), (2, 2, 2024-07-03, 'executed'), (3, 3, 2024-07-04, 'pending'), (4, 4, 4, 2024-07-05, 'executed'), (5, 5, 5, 2024-07-06, 'pend...')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
Comments:		34 17:15:56	INSERT INTO Invoice VALUES (1, 1, 'ACCD001', '2024-07-05', '2024-07-01', '2024-07-05', (2, 2, 'ACCD001', '2024-07-07', 2024-07-06, '2024-07-07'), (3, 3, 'ACCD001', '2024-07-06', '2024-07-07', 2024-07-07), (4, 4, 'ACCD001', '2024-07-07', 2024-07-07, '2024-07-07'), (5, 5, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (6, 6, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (7, 7, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (8, 8, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (9, 9, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (10, 10, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (11, 11, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (12, 12, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (13, 13, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (14, 14, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (15, 15, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (16, 16, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (17, 17, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (18, 18, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (19, 19, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (20, 20, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (21, 21, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (22, 22, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (23, 23, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (24, 24, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (25, 25, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (26, 26, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (27, 27, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (28, 28, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (29, 29, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (30, 30, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (31, 31, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (32, 32, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (33, 33, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (34, 34, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (35, 35, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (36, 36, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (37, 37, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (38, 38, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (39, 39, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (40, 40, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (41, 41, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (42, 42, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (43, 43, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (44, 44, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (45, 45, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (46, 46, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (47, 47, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (48, 48, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (49, 49, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (50, 50, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (51, 51, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (52, 52, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (53, 53, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (54, 54, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (55, 55, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (56, 56, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (57, 57, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (58, 58, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (59, 59, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (60, 60, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (61, 61, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (62, 62, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (63, 63, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (64, 64, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (65, 65, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (66, 66, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (67, 67, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (68, 68, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (69, 69, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (70, 70, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (71, 71, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (72, 72, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (73, 73, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (74, 74, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (75, 75, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (76, 76, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (77, 77, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (78, 78, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (79, 79, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (80, 80, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (81, 81, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (82, 82, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (83, 83, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (84, 84, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (85, 85, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (86, 86, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (87, 87, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (88, 88, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (89, 89, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (90, 90, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (91, 91, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (92, 92, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (93, 93, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (94, 94, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (95, 95, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (96, 96, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (97, 97, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (98, 98, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (99, 99, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (100, 100, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (101, 101, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (102, 102, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (103, 103, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (104, 104, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (105, 105, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (106, 106, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (107, 107, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (108, 108, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (109, 109, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (110, 110, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (111, 111, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (112, 112, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (113, 113, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (114, 114, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (115, 115, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (116, 116, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (117, 117, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (118, 118, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (119, 119, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (120, 120, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (121, 121, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (122, 122, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (123, 123, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (124, 124, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (125, 125, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (126, 126, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (127, 127, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (128, 128, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (129, 129, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (130, 130, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (131, 131, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (132, 132, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (133, 133, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (134, 134, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (135, 135, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (136, 136, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (137, 137, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (138, 138, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (139, 139, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (140, 140, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (141, 141, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (142, 142, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (143, 143, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (144, 144, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (145, 145, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (146, 146, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (147, 147, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (148, 148, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (149, 149, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (150, 150, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (151, 151, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (152, 152, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (153, 153, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (154, 154, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (155, 155, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (156, 156, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (157, 157, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (158, 158, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (159, 159, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (160, 160, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (161, 161, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (162, 162, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (163, 163, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (164, 164, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (165, 165, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (166, 166, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (167, 167, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (168, 168, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (169, 169, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (170, 170, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (171, 171, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (172, 172, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (173, 173, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (174, 174, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (175, 175, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (176, 176, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (177, 177, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (178, 178, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (179, 179, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (180, 180, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (181, 181, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (182, 182, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (183, 183, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (184, 184, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (185, 185, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (186, 186, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (187, 187, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (188, 188, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (189, 189, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (190, 190, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (191, 191, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (192, 192, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (193, 193, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (194, 194, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (195, 195, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (196, 196, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (197, 197, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (198, 198, 300, 2024-07-01, 2024-07-01, '2024-07-01'), (199, 199,		

## A.6 SQL Trigger: trig-after-payable

## A.7 SQL Trigger: trig-stop-overbooking

## A.8 SQL Trigger: trig-after-payment

```

DELIMITER //
CREATE TRIGGER after_payment_insert
AFTER INSERT ON Payment
FOR EACH ROW
BEGIN
    UPDATE Patient p
    SET p.balance = p.balance - NEW.amount
    WHERE p.patient_id = NEW.patient_id;
END;//
DELIMITER ;

```

Action Output 0

Object Info	Session	Time	Action	Response	Duration / Fetch Time
View: roomoccupancyrate		36	17:18:58 INSERT INTO Payable VALUES(1, 1, 400.00, 'Room charge 101 for 4 nights', '2024-07-07'), (2, 1, 50.00, 'Medication: Lainepril', '2024-07-07'), (3, 2, 500.00, 'Room charge 102 for 5 nights', '2024-07-07'), ... 10 rows(s) affected Records: 10 Duplicates: 0 Warnings: 0	0 rows(s) returned	0.0018 sec
Column: room_number		37	17:18:59 SELECT ph.field_of_expense AS department, COUNT(po.order_id) AS total_treatments FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id GROUP BY ph.field_of_expense	0 rows(s) returned	0.0023 sec / 0.0000...
Column: capacity		38	17:18:19 SELECT ph.name AS physician_name, SUM(po.fee) AS total_revenue FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id GROUP BY ph.name	0 rows(s) returned	0.0009 sec / 0.0000...
Column: occupied_beds		39	17:18:28 SELECT ph.name AS physician_name, COUNT(hr.visit_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Smith' AND hr.visit_date > '2024-01-01' AND hr.visit_date < '2024-02-01'	0 rows(s) returned	0.0009 sec / 0.0000...
Column: occupancy_rate		40	17:18:30 SELECT ph.name AS physician_name, COUNT(hr.visit_id) AS acute_cases FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Smith' AND hr.visit_date > '2024-01-01' AND hr.visit_date < '2024-02-01'	0 rows(s) returned	0.0008 sec / 0.0000...
Column: name		41	17:18:39 SELECT r.room_number, COUNT(pr.assignment_id) AS total_occupancy FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number HAVING total_occupancy < 5	0 rows(s) returned	0.0008 sec / 0.0000...
Column: capacity		42	17:18:58 SELECT r.room_number, COUNT(pr.assignment_id) AS total_occupancy FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number ORDER BY total_occupancy DESC LIMIT 1	0 rows(s) returned	0.0008 sec / 0.0000...
Column: occupied_beds		43	17:18:16 SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration ma JOIN Medication m ON ma.medication_id = m.medication_id GROUP BY m.name	0 rows(s) returned	0.0009 sec / 0.0000...
Column: capacity		44	17:18:36 SELECT p.name, SUM(po.amount) AS total_payments FROM Patient p JOIN Payment po ON p.patient_id = po.patient_id GROUP BY p.name HAVING total_payments > (SELECT AVG(total_payments) FROM Patient)	2 rows(s) returned	0.0012 sec / 0.0000...
Column: occupied_beds		45	17:18:37 SELECT p.name AS patient_name, m.name AS medication_name, ma.administration_date, n.name AS nurse_name, ppm.MedicationAdministration ma JOIN Patient p ON ma.patient_id = p.patient_id WHERE p.name = 'Frank Williams' AND m.name = 'Lainepril' AND ma.administration_date = '2024-07-07' AND n.name = 'Sarah' AND ppm.MedicationAdministration ma JOIN Nurse n ON ma.nurse_id = n.nurse_id	1 row(s) affected	0.0009 sec / 0.0000...
Column: capacity		46	17:18:56 CREATE VIEW PatientBillingOverview AS SELECT p.patient_id, p.name, SUM(po.amount) AS total_bill, p.balance AS current_balance FROM Patient p JOIN Payment po ON p.patient_id = po.patient_id GROUP BY p.name	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		47	17:22:19 CREATE VIEW PatientBillingOverview AS SELECT p.patient_id, p.name, SUM(po.amount) AS total_bill, p.balance AS current_balance FROM Patient p JOIN Payment po ON p.patient_id = po.patient_id GROUP BY p.name	0 rows(s) returned	0.0039 sec
Column: capacity		48	17:22:19 CREATE VIEW PatientBillingOverview AS SELECT p.patient_id, p.name, SUM(po.amount) AS total_bill, p.balance AS current_balance FROM Patient p JOIN Payment po ON p.patient_id = po.patient_id GROUP BY p.name	0 rows(s) affected	0.0039 sec
Column: capacity		49	17:23:11 CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS occupied_Beds, COUNT(DISTINCT pr.patient_id) / r.capacity AS occupancy FROM Room r JOIN PatientRoom pr ON r.room_number = pr.room_number	0 rows(s) affected	0.0030 sec
Column: capacity		50	17:23:26 CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS occupied_Beds, COUNT(DISTINCT pr.patient_id) / r.capacity AS occupancy FROM Room r JOIN PatientRoom pr ON r.room_number = pr.room_number	0 rows(s) affected	0.0031 sec
Column: capacity		51	17:23:43 CREATE TRIGGER PreventOverBooking BEFORE INSERT ON PatientRoom FOR EACH ROW BEGIN DECLARE current_capacity INT; DECLARE room_capacity INT;	0 rows(s) affected	0.0082 sec
Column: capacity		52	17:23:44 CREATE TRIGGER after_payment_insert AFTER INSERT ON Payable FOR EACH ROW BEGIN UPDATE Patient p SET p.balance = p.balance + NEW.amount WHERE p.patient_id = NEW.patient_id END;	0 rows(s) affected	0.0034 sec
Column: capacity		53	17:24:44 SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_id, po.date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician ph ON p.physician_id = po.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		54	17:24:44 SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_id, po.date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician ph ON p.physician_id = po.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		55	17:24:47 SELECT ph.field_of_expense AS department, SUM(po.fee) AS total_revenue FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		56	17:25:05 SELECT p.name, DATEDIFF(pr.check_in_date, pr.check_out_date) AS length_of_stay FROM Patient p JOIN PatientRoom pr ON p.patient_id = pr.patient_id WHERE DATEDIFF(pr.check_in_date, pr.	0 rows(s) returned	0.0021 sec / 0.0000...
Column: capacity		57	17:25:10 START TRANSACTION;	0 rows(s) affected	0.0023 sec
Column: capacity		58	17:25:19 INSERT INTO Patient (patient_id, name, address, phone) VALUES (6, 'Frank Williams', '600 New St, Newtown', '555-6666')	1 row(s) affected	0.0009 sec
Column: capacity		59	17:25:19 INSERT INTO HealthRecord (record_id, patient_id, disease, diagnosis_date, status, description) VALUES (6, 6, 'Influenza', CURDATE(), 'ongoing', 'Patient admitted with severe flu symptoms')	1 row(s) affected	0.0031 sec
Column: capacity		60	17:25:15 INSERT INTO PatientRoom (assignment_id, patient_id, room_number, check_in_date, check_out_date) VALUES (6, 6, 101, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 5 DAY))	1 row(s) affected	0.0020 sec
Column: capacity		61	17:25:15 COMMIT;	0 rows(s) affected	0.0008 sec

Query Completed

## A.9 SQL Transaction: trans-pt-admit

```

START TRANSACTION;
INSERT INTO Patient (patient_id, name, address, phone)
VALUES (6, 'Frank Williams', '600 New St, Newtown', '555-6666');
INSERT INTO HealthRecord (record_id, patient_id, disease, diagnosis_date, status, description)
VALUES (6, 6, 'Influenza', CURDATE(), 'ongoing', 'Patient admitted with severe flu symptoms');
INSERT INTO PatientRoom (assignment_id, patient_id, room_number, check_in_date, check_out_date)
VALUES (6, 6, 101, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 5 DAY));
COMMIT;

```

Action Output 0

Object Info	Session	Time	Action	Response	Duration / Fetch Time
View: roomoccupancyrate		43	17:18:16 SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration ma JOIN Medication m ON ma.medication_id = m.medication_id GROUP BY m.name	0 rows(s) returned	0.0009 sec / 0.0000...
Column: room_number		44	17:18:26 SELECT ph.name AS physician_name, COUNT(ppm.patient_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	0 rows(s) returned	0.0011 sec / 0.0000...
Column: capacity		45	17:18:36 SELECT p.name AS nurse_name, COUNT(ma.administration_id) AS total_medications FROM MedicationAdministration ma JOIN Medication m ON ma.medication_id = m.medication_id JOIN Nurse n ON ma.nurse_id = n.nurse_id GROUP BY p.name	0 rows(s) returned	0.0012 sec / 0.0000...
Column: capacity		46	17:18:44 SELECT r.name AS room_name, COUNT(pr.assignment_id) AS total_occupied FROM Room r JOIN PatientRoom pr ON r.room_number = pr.room_number GROUP BY r.name	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		47	17:18:54 CREATE VIEW PatientBillingOverview AS SELECT p.patient_id, p.name, SUM(po.amount) AS total_bill, p.balance AS current_balance FROM Patient p JOIN Payment po ON p.patient_id = po.patient_id GROUP BY p.name	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		48	17:21:29 CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS occupied_Beds, COUNT(DISTINCT pr.patient_id) / r.capacity AS occupancy FROM Room r JOIN PatientRoom pr ON r.room_number = pr.room_number	0 rows(s) returned	0.0039 sec
Column: capacity		49	17:23:11 CREATE VIEW PhysicianPerformance AS SELECT ph.physician_id, ph.name, COUNT(DISTINCT pr.patient_id) AS patients_treated, COUNT(DISTINCT pr.patient_id) / r.capacity AS occupancy FROM Physician ph JOIN PhysicianPatientMonitoring ppm ON ph.physician_id = ppm.physician_id JOIN Room r ON ppm.room_id = r.room_id	0 rows(s) returned	0.0030 sec
Column: capacity		50	17:23:26 CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS occupied_Beds, COUNT(DISTINCT pr.patient_id) / r.capacity AS occupancy FROM Room r JOIN PatientRoom pr ON r.room_number = pr.room_number	0 rows(s) returned	0.0031 sec
Column: capacity		51	17:23:43 CREATE TRIGGER PreventOverBooking BEFORE INSERT ON PatientRoom FOR EACH ROW BEGIN DECLARE current_capacity INT; DECLARE room_capacity INT;	0 rows(s) affected	0.0082 sec
Column: capacity		52	17:23:44 CREATE TRIGGER after_payment_insert AFTER INSERT ON Payable FOR EACH ROW BEGIN UPDATE Patient p SET p.balance = p.balance + NEW.amount WHERE p.patient_id = NEW.patient_id END;	0 rows(s) affected	0.0034 sec
Column: capacity		53	17:24:44 SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_id, po.date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician ph ON p.physician_id = po.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		54	17:24:44 SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_id, po.date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician ph ON p.physician_id = po.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		55	17:24:47 SELECT ph.field_of_expense AS department, SUM(po.fee) AS total_revenue FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id	0 rows(s) returned	0.0010 sec / 0.0000...
Column: capacity		56	17:25:05 SELECT p.name, DATEDIFF(pr.check_in_date, pr.check_out_date) AS length_of_stay FROM Patient p JOIN PatientRoom pr ON p.patient_id = pr.patient_id WHERE DATEDIFF(pr.check_in_date, pr.	0 rows(s) returned	0.0021 sec / 0.0000...
Column: capacity		57	17:25:10 START TRANSACTION;	0 rows(s) affected	0.0023 sec
Column: capacity		58	17:25:19 INSERT INTO Patient (patient_id, name, address, phone) VALUES (6, 'Frank Williams', '600 New St, Newtown', '555-6666')	1 row(s) affected	0.0009 sec
Column: capacity		59	17:25:19 INSERT INTO HealthRecord (record_id, patient_id, disease, diagnosis_date, status, description) VALUES (6, 6, 'Influenza', CURDATE(), 'ongoing', 'Patient admitted with severe flu symptoms')	1 row(s) affected	0.0031 sec
Column: capacity		60	17:25:15 INSERT INTO PatientRoom (assignment_id, patient_id, room_number, check_in_date, check_out_date) VALUES (6, 6, 101, CURDATE(), DATE_ADD(CURDATE(), INTERVAL 5 DAY))	1 row(s) affected	0.0020 sec
Column: capacity		61	17:25:15 COMMIT;	0 rows(s) affected	0.0008 sec

Query Completed

## A.10 SQL Transaction: trans-pt-discharge

```

1 • START TRANSACTION;
2
3 • UPDATE PatientRoom
4   SET check_out_date = CURDATE()
5   WHERE patient_id = 1 AND check_out_date IS NULL;
6
7 • INSERT INTO Invoice (invoice_id, patient_id, account_number, issue_date, start_date, end_date)
8   VALUES (6, 1, 'ACCO05', CURDATE());
9   (SELECT check_in_date FROM PatientRoom WHERE patient_id = 1 ORDER BY check_in_date DESC LIMIT 1),
10   CURDATE());
11
12 • INSERT INTO Payable (payable_id, invoice_id, amount, description, date)
13   (SELECT COALESCE(MAX(payable_id), 0) FROM Payable) + ROW_NUMBER() OVER () AS payable_id,
14   6 AS invoice_id,
15   CASE
16     WHEN p.type = 'room' THEN r.nightly_fee * DATEDIFF(pr.check_out_date, pr.check_in_date)
17     WHEN p.type = 'medication' THEN m.price * m.amount
18     WHEN p.type = 'instruction' THEN i.fee
19   END AS amount,
20   p.description,
21   p.date
22   FROM (
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
686
687
688
689
689
690
691
692
693
694
695
696
696
697
698
699
699
700
701
702
703
704
705
705
706
707
708
709
709
710
711
712
713
714
715
715
716
717
718
718
719
719
720
721
722
723
723
724
725
726
726
727
728
728
729
729
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
157
```

## A.12 SQL Query: qr-pt-ab-avg-payment

```

5      Patient p
6      JOIN Payment pay ON p.patient_id = pay.patient_id
7      GROUP BY p.name
8      HAVING total_payments > (SELECT AVG(total_payments) FROM (SELECT SUM(amount) AS total_payments FROM Payment GROUP BY patient_id) AS subquery)
9      ORDER BY total_payments DESC;
10

```

Result Grid

name	total_payments
Carol Brown	500.00
David Wilson	400.00

Action Output

Time	Action	Response	Duration / Fetch Time
27 17:15:56	INSERT INTO PatientRoom VALUES (1, 1, 101, '2024-07-01', '2024-07-01', 2, 1, 102, '2024-07-02', '2024-07-02', 1, 3, 103, '2024-07-03', '2024-07-03'), (4, 2, 201, '2024-07-04', '2024-07-04', 15, 5, 202, 5, 104, '2024-07-05', '2024-07-05')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
28 17:15:56	INSERT INTO Medication VALUES (1, 'Aspirin', 'Pain reliever', 5000, (2, 'Ampicillin', 'Antibiotic', 10000), (3, 'Lisinopril', 'Blood pressure medication', 15000), (4, 'Metformin', 'Diabetes medication', 12000), (5, 'Ibuprofen', 'Anti-inflammatory', 8000), (6, 'MRI001', 'MRI scan', 50000), (7, 'PT001', 'Physical therapy session', 7500), (8, 'NUTR001', 'Nutrition consultation', 10000), (9, 'Sedative', 'Sleep aid', 3000), (10, 'COPD001', 'COPD treatment', 40000), (11, 'Hypertension001', 'Hypertension medication', 10000), (12, 'Insulin', 'Insulin injection', 10000), (13, 'Oxygen', 'Oxygen therapy', 5000), (14, 'PPI', 'Proton pump inhibitor', 10000), (15, 'NSAID', 'Nonsteroidal anti-inflammatory drug', 10000), (16, 'Anticoagulant', 'Blood thinner', 10000), (17, 'Antidepressant', 'Antidepressant medication', 10000), (18, 'Antihistamine', 'Antihistamine', 10000), (19, 'Antidiabetic', 'Antidiabetic medication', 10000), (20, 'Antifungal', 'Antifungal medication', 10000), (21, 'Antiviral', 'Antiviral medication', 10000), (22, 'Antimicrobial', 'Antimicrobial medication', 10000), (23, 'Anticonvulsant', 'Anticonvulsant medication', 10000), (24, 'Antipsychotic', 'Antipsychotic medication', 10000), (25, 'Antihypertensive', 'Antihypertensive medication', 10000), (26, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (27, 'Antidiuretic', 'Antidiuretic medication', 10000), (28, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (29, 'Antineoplastic', 'Antineoplastic medication', 10000), (30, 'Anticoagulant', 'Anticoagulant medication', 10000), (31, 'Antidiabetic', 'Antidiabetic medication', 10000), (32, 'Antihypertensive', 'Antihypertensive medication', 10000), (33, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (34, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (35, 'Antineoplastic', 'Antineoplastic medication', 10000), (36, 'Anticoagulant', 'Anticoagulant medication', 10000), (37, 'Antidiabetic', 'Antidiabetic medication', 10000), (38, 'Antihypertensive', 'Antihypertensive medication', 10000), (39, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (40, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (41, 'Antineoplastic', 'Antineoplastic medication', 10000), (42, 'Anticoagulant', 'Anticoagulant medication', 10000), (43, 'Antidiabetic', 'Antidiabetic medication', 10000), (44, 'Antihypertensive', 'Antihypertensive medication', 10000), (45, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000))	0 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
29 17:15:56	INSERT INTO Medication VALUES (1, 'Aspirin', 'Pain reliever', 5000, (2, 'Ampicillin', 'Antibiotic', 10000), (3, 'Lisinopril', 'Blood pressure medication', 15000), (4, 'Metformin', 'Diabetes medication', 12000), (5, 'Ibuprofen', 'Anti-inflammatory', 8000), (6, 'MRI001', 'MRI scan', 50000), (7, 'PT001', 'Physical therapy session', 7500), (8, 'NUTR001', 'Nutrition consultation', 10000), (9, 'Sedative', 'Sleep aid', 3000), (10, 'COPD001', 'COPD treatment', 40000), (11, 'Hypertension001', 'Hypertension medication', 10000), (12, 'Insulin', 'Insulin injection', 10000), (13, 'Oxygen', 'Oxygen therapy', 5000), (14, 'PPI', 'Proton pump inhibitor', 10000), (15, 'NSAID', 'Nonsteroidal anti-inflammatory drug', 10000), (16, 'Anticoagulant', 'Blood thinner', 10000), (17, 'Antidepressant', 'Antidepressant medication', 10000), (18, 'Antihistamine', 'Antihistamine', 10000), (19, 'Antidiabetic', 'Antidiabetic medication', 10000), (20, 'Antifungal', 'Antifungal medication', 10000), (21, 'Antiviral', 'Antiviral medication', 10000), (22, 'Antimicrobial', 'Antimicrobial medication', 10000), (23, 'Anticonvulsant', 'Anticonvulsant medication', 10000), (24, 'Antipsychotic', 'Antipsychotic medication', 10000), (25, 'Antihypertensive', 'Antihypertensive medication', 10000), (26, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (27, 'Antidiuretic', 'Antidiuretic medication', 10000), (28, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (29, 'Antineoplastic', 'Antineoplastic medication', 10000), (30, 'Anticoagulant', 'Anticoagulant medication', 10000), (31, 'Antidiabetic', 'Antidiabetic medication', 10000), (32, 'Antihypertensive', 'Antihypertensive medication', 10000), (33, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (34, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (35, 'Antineoplastic', 'Antineoplastic medication', 10000), (36, 'Anticoagulant', 'Anticoagulant medication', 10000), (37, 'Antidiabetic', 'Antidiabetic medication', 10000), (38, 'Antihypertensive', 'Antihypertensive medication', 10000), (39, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000), (40, 'Anticholinesterase', 'Anticholinesterase medication', 10000), (41, 'Antineoplastic', 'Antineoplastic medication', 10000), (42, 'Anticoagulant', 'Anticoagulant medication', 10000), (43, 'Antidiabetic', 'Antidiabetic medication', 10000), (44, 'Antihypertensive', 'Antihypertensive medication', 10000), (45, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 10000))	0 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
30 17:15:56	INSERT INTO MedicationAdministration VALUES (1, 1, 3, '2024-07-02'), (2, 2, 5, '2024-07-03'), (3, 3, 2, '2024-07-04'), (4, 4, 4, '2024-07-05'), (5, 5, 1, '2024-07-06')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
31 17:15:56	INSERT INTO Instruction VALUES (1, 'CB001', 'Complete blood count', 50000), (2, 'XRAY001', 'X-ray', 100000), (3, 'MR001', 'MRI scan', 500000), (4, 'PT001', 'Physical therapy session', 75000), (5, 'NUTR001', 'Nutrition consultation', 100000), (6, 'Sedative', 'Sleep aid', 30000), (7, 'COPD001', 'COPD treatment', 400000), (8, 'Hypertension001', 'Hypertension medication', 100000), (9, 'Insulin', 'Insulin injection', 100000), (10, 'Oxygen', 'Oxygen therapy', 50000), (11, 'PPI', 'Proton pump inhibitor', 100000), (12, 'NSAID', 'Nonsteroidal anti-inflammatory drug', 100000), (13, 'Anticoagulant', 'Blood thinner', 100000), (14, 'Antidepressant', 'Antidepressant medication', 100000), (15, 'Antihistamine', 'Antihistamine', 100000), (16, 'Antidiabetic', 'Antidiabetic medication', 100000), (17, 'Antifungal', 'Antifungal medication', 100000), (18, 'Antiviral', 'Antiviral medication', 100000), (19, 'Antimicrobial', 'Antimicrobial medication', 100000), (20, 'Anticonvulsant', 'Anticonvulsant medication', 100000), (21, 'Antipsychotic', 'Antipsychotic medication', 100000), (22, 'Antihypertensive', 'Antihypertensive medication', 100000), (23, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 100000), (24, 'Antidiuretic', 'Antidiuretic medication', 100000), (25, 'Anticholinesterase', 'Anticholinesterase medication', 100000), (26, 'Antineoplastic', 'Antineoplastic medication', 100000), (27, 'Anticoagulant', 'Anticoagulant medication', 100000), (28, 'Antidiabetic', 'Antidiabetic medication', 100000), (29, 'Antihypertensive', 'Antihypertensive medication', 100000), (30, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 100000), (31, 'Anticholinesterase', 'Anticholinesterase medication', 100000), (32, 'Antineoplastic', 'Antineoplastic medication', 100000), (33, 'Anticoagulant', 'Anticoagulant medication', 100000), (34, 'Antidiabetic', 'Antidiabetic medication', 100000), (35, 'Antihypertensive', 'Antihypertensive medication', 100000), (36, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 100000), (37, 'Anticholinesterase', 'Anticholinesterase medication', 100000), (38, 'Antineoplastic', 'Antineoplastic medication', 100000), (39, 'Anticoagulant', 'Anticoagulant medication', 100000), (40, 'Antidiabetic', 'Antidiabetic medication', 100000), (41, 'Antihypertensive', 'Antihypertensive medication', 100000), (42, 'Antihyperlipidemic', 'Antihyperlipidemic medication', 100000), (43, 'Anticholinesterase', 'Anticholinesterase medication', 100000), (44, 'Antineoplastic', 'Antineoplastic medication', 100000), (45, 'Anticoagulant', 'Anticoagulant medication', 100000))	0 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
32 17:15:56	INSERT INTO PhysicianMonitoring VALUES (1, 1, '2024-07-01', 2, 2, 3, '2024-07-02'), (3, 3, 5, '2024-07-03'), (4, 4, 4, '2024-07-04'), (5, 5, 1, '2024-07-05')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
33 17:15:56	INSERT INTO RoomOccupancy VALUES (1, 1, '2024-07-01', 2, 2, 3, '2024-07-02'), (3, 3, 5, '2024-07-03'), (4, 4, 4, '2024-07-04'), (5, 5, 1, '2024-07-05')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
34 17:15:56	INSERT INTO Invoice VALUES (1, 'ACC001', '2024-07-05', 2, 2, 'ACCO02', '2024-07-06', 3, 3, 'ACCO03', '2024-07-07', 4, 4, 'ACCO04', '2024-07-08', 5, 5, 'ACCO05', '2024-07-11'), (6, 6, 'ACCO06', '2024-07-12'), (7, 7, 'ACCO07', '2024-07-13'), (8, 8, 'ACCO08', '2024-07-14')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
35 17:15:56	INSERT INTO Payable VALUES (1, 1, 14000, 'Room charge: 101 for 4 nights', '2024-07-05'), (2, 2, 50000, '2024-07-10'), (3, 3, 60000, '2024-07-12'), (4, 4, 40000, '2024-07-13'), (5, 5, 30000, '2024-07-14')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
36 17:15:56	INSERT INTO Payment VALUES (1, 1, 20000, '2024-07-10'), (2, 2, 30000, '2024-07-11'), (3, 3, 40000, '2024-07-12'), (4, 4, 50000, '2024-07-13'), (5, 5, 60000, '2024-07-14')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0039 sec
37 17:15:56	SELECT ph.field_of_expertise AS department, COUNT(po.order_id) AS total_treatments FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.field_of_expertise	5 rows(s) returned	0.0022 sec
38 17:15:56	SELECT ph.name AS physician_name, COUNT(hr.record_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianMonitoring pm ON hr.patient_id = pm.patient_id JOIN Physician ph ON ph.physician_id = pm.physician_id WHERE hr.date BETWEEN '2024-07-01' AND '2024-07-05' GROUP BY ph.name	5 rows(s) returned	0.00097 sec
39 17:15:56	SELECT r.room_number, AVG(DATEDIFF(pr.check_out_date, pr.check_in_date)) AS avg_length_of_stay FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.0018 sec
40 17:15:56	SELECT r.room_number, COUNT(pr.admission_date) AS total_admissions FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.00097 sec
41 17:15:56	SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS total_patients FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.00098 sec
42 17:15:56	SELECT r.room_number, COUNT(DISTINCT pr.visit_id) AS total_visits FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.00099 sec
43 17:15:56	SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration ma JOIN Medication m ON ma.medication_id = m.medication_id GROUP BY m.name	5 rows(s) returned	0.001 sec
44 17:15:56	SELECT ph.name AS physician_name, COUNT(ppn.patient_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 rows(s) returned	0.0011 sec
45 17:15:56	SELECT ph.name, SUM(pay.amount) AS total_payments FROM Patient p JOIN Payment pay ON p.patient_id = pay.patient_id GROUP BY ph.name HAVING total_payments > (SELECT AVG(total_payments) FROM Patient GROUP BY ph.name)	2 rows(s) returned	0.0012 sec

Object info Session

Schema: hospital

Query Completed

## A.13 SQL Query: qr-dept-rev

```

1 • SELECT
2   ph.field_of_expertise AS department,
3   SUM(l.fee) AS total_revenue
4   FROM
5   Physician ph
6   JOIN Instruction l ON po.instruction_id = l.instruction_id
7   GROUP BY ph.field_of_expertise
8   ORDER BY total_revenue DESC;
9

```

Result Grid

department	total_revenue
Orthopedics	500
Pediatrics	75
Neurology	65
Otolaryngology	60
Cardiology	60

Action Output

Time	Action	Response	Duration / Fetch Time
37 17:17:28	SELECT ph.field_of_expertise AS department, COUNT(po.order_id) AS total_treatments FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.field_of_expertise	5 rows(s) returned	0.0022 sec / 0.00000...
38 17:17:28	SELECT ph.name AS physician_name, SUM(l.fee) AS total_revenue FROM Physician ph ON po.instruction_id = l.instruction_id GROUP BY ph.name	5 rows(s) returned	0.001 sec / 0.00000...
39 17:17:28	SELECT ph.name AS physician_name, COUNT(hr.record_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianMonitoring pm ON hr.patient_id = pm.patient_id JOIN Physician ph ON ph.physician_id = pm.physician_id WHERE hr.date BETWEEN '2024-07-01' AND '2024-07-05' GROUP BY ph.name	5 rows(s) returned	0.00097 sec / 0.00000...
40 17:17:37	SELECT r.room_number, AVG(DATEDIFF(pr.check_out_date, pr.check_in_date)) AS avg_length_of_stay FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.0018 sec
41 17:17:48	SELECT r.room_number, COUNT(pr.admission_date) AS total_admissions FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.00097 sec / 0.00000...
42 17:17:58	SELECT r.room_number, COUNT(DISTINCT pr.patient_id) AS total_patients FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number	5 rows(s) returned	0.00098 sec / 0.00000...
43 17:18:08	SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration ma JOIN Medication m ON ma.medication_id = m.medication_id GROUP BY m.name	5 rows(s) returned	0.0009 sec
44 17:18:26	SELECT ph.name AS physician_name, COUNT(ppn.patient_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.00000...
45 17:18:36	SELECT ph.name, SUM(pay.amount) AS total_payments FROM Patient p JOIN Payment pay ON p.patient_id = pay.patient_id GROUP BY ph.name HAVING total_payments > (SELECT AVG(total_payments) FROM Patient GROUP BY ph.name)	2 rows(s) returned	0.0012 sec / 0.00000...
46 17:18:46	SELECT n.name AS nurse_name, COUNT(po.order_id) AS total_orders FROM PhysicianOrder po JOIN Nurse n ON po.nurse_id = n.nurse_id GROUP BY n.name	5 rows(s) returned	0.001 sec / 0.00001...
47 17:18:46	SELECT n.name AS nurse_name, COUNT(po.order_id) AS total_orders FROM PhysicianOrder po JOIN Nurse n ON po.nurse_id = n.nurse_id GROUP BY n.name	5 rows(s) returned	0.00099 sec / 0.00001...
48 17:19:20	CREATE VIEW PatientBillingReview AS SELECT p.patient_id, ph.name, SUM(l.fee) AS total_bill, SUM(pay.amount) AS total_paid, (AVG(l.fee)) AS avg_bill	0 rows(s) affected	0.0039 sec
49 17:23:11	CREATE VIEW PhysicianPerformance AS SELECT ph.physician_id, ph.name, COUNT(DISTINCT po.patient_id) AS patients_treated, COUNT(po.order_id) AS orders_given, COUNT(po.order_id) AS orders_treated, COUNT(po.order_id) AS orders_left	0 rows(s) affected	0.0030 sec
50 17:23:26	CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, r.capacity, COUNT(DISTINCT pr.patient_id) AS occupied_beds, (COUNT(DISTINCT pr.patient_id) / capacity) * 100 AS occupancy_percent	0 rows(s) affected	0.0031 sec
51 17:23:43	CREATE TRIGGER PreDeleteOverBooking BEFORE DELETE ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; SELECT COUNT(*) INTO current_occupancy; END IF current_occupancy >= capacity THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Overbooking detected'; END IF;	0 rows(s) affected	0.0082 sec
52 17:23:43	CREATE TRIGGER InsertOverBooking BEFORE INSERT ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; SELECT COUNT(*) INTO current_occupancy; END IF current_occupancy >= capacity THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Overbooking detected'; END IF;	0 rows(s) affected	0.0082 sec
53 17:23:44	CREATE TRIGGER after_payment_insert AFTER INSERT ON Payment FOR EACH ROW BEGIN UPDATE Patient p SET balance = balance - amount WHERE p.patient_id = NEW.patient_id; END IF;	0 rows(s) affected	0.0034 sec
54 17:24:44	SELECT ph.name AS physician_name, ph.name AS physician_name, l.description AS treatment, po.order_id, date FROM Patient p JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction l ON po.instruction_id = l.instruction_id	5 rows(s) returned	0.0010 sec / 0.00001...
55 17:24:57	SELECT ph.field_of_expertise AS department, COUNT(po.order_id) AS total_revenue FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction l ON po.instruction_id = l.instruction_id	5 rows(s) returned	0.0010 sec / 0.00000...

Object info Session

Schema: hospital

Query Completed

## A.14 SQL Query: qr-pt-ttl-expenses

```

1 • SELECT
2   p.name AS patient_name,
3   SUM(pay.amount) AS total_expenses
4   FROM
5   Payment|pay
6   JOIN Patient p ON pay.patient_id = p.patient_id
7   GROUP BY p.patient_id
8
9 Result Grid Filter Rows: Search Export: 
patient_name total_expenses...
Carol Brown 500.00
Jane Doe 400.00
Bob Smith 200.00
Eve Davis 300.00
Alice Johnson 200.00

```

Object Info Session No object selected

Action Output	Time	Action	Response	Duration / Fetch Time
23	17:15:56	INSERT INTO Nurse VALUES ('1', Jane Doe, 'NCERT001', '111 First St. Anytown, 555-2348'), ('2', 'Mark Johnson', 'NCERT002', '222 Second Ave, Somewhere', '555-6789'), ('3', 'Lisa Brown', 'NCERT003', '333 Third St., Anywhere', '555-1234')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0003 sec
24	17:15:56	INSERT INTO Room VALUES ('101', 2, 100.00), ('102', 2, 100.00), ('103', 1, 100.00), ('104', 3, 100.00), ('105', 1, 100.00)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
25	17:15:56	INSERT INTO Patient (patient_id, name, address, phone) VALUES ('1', Alice Johnson, '100 Patient St, Sickville', '555-1111'), ('2', Bob Smith, '200 Ill Ave, Unwell City', '555-2222'), ('3', Carol Brown, '300 Health St., Anywhere', '555-3333')	3 rows(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.0015 sec
26	17:15:56	INSERT INTO HealthRecord VALUES ('1', 1, 1, 'Hypertension', '2024-01-15', 'ongoing'), ('Patient experiencing high blood pressure'), ('2', 1, 2, 102, 2024-07-09, '2024-07-07'), ('3', 1, 103, 2024-07-09, '2024-07-06'), ('4', 1, 102, 2024-07-09, '2024-07-04'), ('5', 1, 20, 2024-07-09, '2024-07-09')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0007 sec
27	17:15:56	INSERT INTO Medication VALUES ('1', 1, 1, 'Aspirin', 'Pan relief', 5.00), ('2', Amoxicillin, 'Antibiotic', 10.00), ('3', Lamictal, 'Depression medication', 12.00), ('4', Metformin, 'Diabetes medication', 12.00), ('5', Ibuprofen, 'Pain reliever', 5.00)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0019 sec
28	17:15:56	INSERT INTO MedicationAdministration VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 5, 2024-07-03), ('3', 3, 3, 2, 2024-07-04), ('4', 4, 4, 4, 2024-07-05), ('5', 5, 5, 1, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0018 sec
29	17:15:56	INSERT INTO Instruction VALUES ('1', 1, 1, 2024-07-02, 'executed'), ('2', 2, 2, 2024-07-03, 'executed'), ('3', 3, 3, 2024-07-04, 'pending'), ('4', 4, 4, 2024-07-05, 'executed'), ('5', 5, 5, 2024-07-06, 'pending')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
30	17:15:56	INSERT INTO PhysicianOrder VALUES ('1', 1, 1, 2024-07-03), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0014 sec
31	17:15:56	INSERT INTO NurseFee VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0012 sec
32	17:15:56	INSERT INTO RoomCharge VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0011 sec
33	17:15:56	INSERT INTO Payment VALUES ('1', 1, 200.00, '2024-07-10'), ('2', 2, 300.00, '2024-07-11'), ('3', 3, 600.00, '2024-07-12'), ('4', 4, 500.00, '2024-07-13'), ('5', 5, 500.00, '2024-07-14')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0010 sec
34	17:15:56	SELECT ph.field_of_expense AS department, COUNT(po.patient_id) AS total_patients FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.field_of_expense	5 rows(s) returned	0.0022 sec / 0.0006...
35	17:15:56	SELECT ph.name AS physician_name, SUM(po.amount) AS total_fee FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.0006...
36	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS chronic_cases FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0016 sec / 0.0006...
37	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_rooms FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0009 sec / 0.0006...
38	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_patients FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.0006...
39	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0012 sec / 0.0006...
40	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0012 sec / 0.0006...
41	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0010 sec / 0.0006...

Object Info Session No object selected

Query Completed

## A.15 SQL Query: qr-nurse-meds

```

1 • SELECT
2   n.name AS nurse_name,
3   SUM(ma.amount) AS total_medications_administered
4   FROM
5   MedicationAdministration ma
6   JOIN Nurse n ON ma.nurse_id = n.nurse_id
7   GROUP BY n.name
8   ORDER BY total_medications_administered DESC;
9
10 Result Grid Filter Rows: Search Export: 
nurse_name total_medications_administered...
Jane Doe 1
Mark Johnson 1
Lisa Brown 1
Tom Davis 1
Amy Wilson 1

```

Object Info Session No object selected

Action Output	Time	Action	Response	Duration / Fetch Time
28	17:15:56	INSERT INTO PhysicianPatientMonitoring VALUES ('1', 1, 2024-07-01, '2024-07-01'), ('2', 2, 2, 2024-07-02, '2024-07-02'), ('3', 3, 3, 2024-07-03, '2024-07-03'), ('4', 4, 4, 2024-07-04, '2024-07-04'), ('5', 5, 5, 2024-07-05, '2024-07-05')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0019 sec
29	17:15:56	INSERT INTO Medication VALUES ('1', Aspirin, 'Pain reliever', 5.00), ('2', Amoxicillin, 'Antibiotic', 10.00), ('3', LinsingerP, 'Blood pressure medication', 15.00), ('4', Metformin, 'Diabetes medication', 12.00), ('5', Ibuprofen, 'Pain reliever', 5.00)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0018 sec
30	17:15:56	INSERT INTO MedicationAdministration VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 5, 2024-07-03), ('3', 3, 3, 2, 2024-07-04), ('4', 4, 4, 4, 2024-07-05), ('5', 5, 5, 1, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0018 sec
31	17:15:56	INSERT INTO Instruction VALUES ('1', 1, 1, 2024-07-02, 'executed'), ('2', 2, 2, 2024-07-03, 'executed'), ('3', 3, 3, 2024-07-04, 'pending'), ('4', 4, 4, 2024-07-05, 'executed'), ('5', 5, 5, 2024-07-06, 'pending')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
32	17:15:56	INSERT INTO PhysicianOrder VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0013 sec
33	17:15:56	INSERT INTO NurseFee VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0012 sec
34	17:15:56	INSERT INTO RoomCharge VALUES ('1', 1, 1, 2024-07-02), ('2', 2, 2, 2024-07-03), ('3', 3, 3, 2024-07-04), ('4', 4, 4, 2024-07-05), ('5', 5, 5, 2024-07-06)	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0012 sec
35	17:15:56	INSERT INTO Payment VALUES ('1', 1, 200.00, '2024-07-10'), ('2', 2, 300.00, '2024-07-11'), ('3', 3, 600.00, '2024-07-12'), ('4', 4, 500.00, '2024-07-13'), ('5', 5, 500.00, '2024-07-14')	5 rows(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0010 sec
36	17:15:56	SELECT ph.field_of_expense AS department, COUNT(po.patient_id) AS total_patients FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.field_of_expense	5 rows(s) returned	0.0022 sec / 0.0006...
37	17:15:56	SELECT ph.name AS physician_name, SUM(po.amount) AS total_fee FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.0006...
38	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS chronic_cases FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0016 sec / 0.0006...
39	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_rooms FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0009 sec / 0.0006...
40	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_patients FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.0006...
41	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0012 sec / 0.0006...
42	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0009 sec / 0.0006...
43	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0009 sec / 0.0006...
44	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0011 sec / 0.0006...
45	17:15:56	SELECT ph.name AS physician_name, COUNT(pr.room_number) AS total_payments FROM PhysicianRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY ph.name	5 rows(s) returned	0.0012 sec / 0.0006...
46	17:15:56	SELECT n.name AS nurse_name, COUNT(ma.administration_id) AS total_medications_administered FROM MedicationAdministration ma JOIN Nurse n ON ma.nurse_id = n.nurse_id GROUP BY n.name	5 rows(s) returned	0.0010 sec / 0.0006...

Object Info Session No object selected

Query Completed

## A.16 SQL Query: qr-pt-ab-avg-stay

```

1 • SELECT
2   p.name,
3   DATEDIFF(pr.check_out_date, pr.check_in_date) AS length_of_stay
4   FROM
5   Patient p
6   JOIN PatientRoom pr ON p.patient_id = pr.patient_id
7   WHERE
8     DATEDIFF(pr.check_out_date, pr.check_in_date) >
9       (SELECT AVG(DATEDIFF(check_out_date, check_in_date))
10      FROM PatientRoom)
11   ORDER BY length_of_stay DESC;

```

The screenshot shows the MySQL Workbench interface with the query results. The results table has one row:

name	length_of_stay
Bob Smith	5
David Wilson	5

Below the results, the 'Object Info' and 'Session' panes are visible. The 'Object Info' pane shows the query's execution details, including the time taken for each step (e.g., 0.0001 sec / 0.00000... for the first select). The 'Session' pane shows the schema is 'hospital'.

## A.17 SQL Query: qr-num-trtmnt-dept

```

3   COUNT(po.order_id) AS total_treatments
4   FROM
5   Physician ph ON po.physician_id = ph.physician_id
6   GROUP BY ph.field_of_expertise
7
8   ORDER BY total_treatments DESC;

```

The screenshot shows the MySQL Workbench interface with the query results. The results table has one row:

department	total_treatments
Cardiology	1
Pediatrics	1
Orthopedics	1
Neurology	1
Oncology	1

Below the results, the 'Object Info' and 'Session' panes are visible. The 'Object Info' pane shows the query's execution details, including the time taken for each step (e.g., 0.0038 sec / 0.00000... for the first select). The 'Session' pane shows the schema is 'hospital-schema'.

## A.18 SQL Query: qr-rm-ab-avg-nights

## A.19 SQL Query: qr-pt-med-hist

## A.20 SQL Query: qr-phys-rev

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

- Administration** tab selected.
- Schemas** tab selected.
- hospital-schema** selected.
- Limit to 1000 rows** checkbox checked.
- Result Grid** pane showing the results of the query:

physician_name	total_revenue
Dr. Michael Brown	500.00
Dr. Sarah Johnson	100.00
Dr. Emily Davis	75.00
Dr. Robert Wilson	60.00
Dr. John Smith	50.00

- Performance** tab selected.
- Dashboard** button.
- Performance Reports** button.
- Performance Schema Setup** button.
- INSTANCE** tab selected.
- Startup / Shutdown** button.
- Server Logs** button.
- Options File** button.
- MANAGEMENT** tab selected.
- Server Status** button.
- User Connections** button.
- Users and Privileges** button.
- Status and System Variables** button.
- Data Export** button.
- Data Import/Restore** button.
- Result Grid** toolbar with icons for Refresh, Filter Rows, Search, and Export.
- Result Grid** status bar showing 100% completion and 1:10 ratio.
- Result Grid** title bar with "Result Grid" and "Read Only" checkboxes.
- Action Output** section with a green checkmark icon and "Result 1" label.
- Object Info** and **Session** sections both show "No object selected".
- Response** section showing execution statistics: 5 rows(s) affected, 5 duplicates, 0 warnings, 0 errors, 0 rows(s) returned, 0.0005 sec duration, 0.0005 sec fetch time.
- Duration / Fetch Time** section showing execution statistics for each row.
- Query Completed** status at the bottom.

## A.21 SQL Query: qr-med-rev

## A.22 SQL Query: qr-phys-chronic

```

5   HealthRecord hr
6   JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id
7   JOIN Physician ph ON ppm.physician_id = ph.physician_id
8   WHERE hr.status = 'ongoing'
9   GROUP BY ph.name
10  ORDER BY chronic_cases DESC;
11

```

Result Grid

physician_name	chronic_cases
Dr. John Smith	1
Dr. Sarah Johnson	1
Dr. Emily Davis	1

Action Output

Time	Action	Response	Duration / Fetch Time
21 17:15:56	USE hospital	0 rows(s) affected	0.00003 sec
22 17:15:56	INSERT INTO Physician VALUES ('Dr. John Smith', 'CERT001', 'Cardiology', '123 Main St, Anytown', '555-1234'), ('Dr. Sarah Johnson', 'CERT002', 'Pediatrics', '456 Oak Ave, Somewhere', '555-5678'), ('Dr. Emily Davis', 'CERT003', 'Hypertension', '222 Second Ave, Somewhere', '555-6789), ('Dr. Michael Brown', 'CERT004', 'Neurology', '333 Elm St, Anywhere', '555-1111')	5 row(s) affected	0.00020 sec
23 17:15:56	INSERT INTO Nurse VALUES ('Jane Doe', 'NCERT001', '111 First St, Anytown', '555-2345), ('Mark Johnson', 'NCERT002', '222 Second Ave, Somewhere', '555-6789), ('Lisa Brown', 'NCERT003', '333 Elm St, Anywhere', '555-1111')	3 row(s) affected	0.00015 sec
24 17:15:56	INSERT INTO Patient (patient_id, name, address, phone) VALUES ('Alice Johnson', '100 Patient St, Sicksville, 12345-1234'), ('Bob Smith', '200 Main St, Anytown, 12345-1234'), ('Carol Brown', '300 Elm St, Anywhere, 12345-1234'), ('David Green', '400 Oak Ave, Somewhere, 12345-1234'), ('Eve White', '500 Pine St, Anywhere, 12345-1234'), ('Frank Black', '600 Elm St, Anywhere, 12345-1234'), ('Grace Grey', '700 Pine St, Anywhere, 12345-1234'), ('Hank Blue', '800 Elm St, Anywhere, 12345-1234'), ('Ivy Green', '900 Pine St, Anywhere, 12345-1234'), ('Jack Red', '100 Elm St, Anywhere, 12345-1234'), ('Kathy Purple', '110 Elm St, Anywhere, 12345-1234'), ('Liam Blue', '120 Elm St, Anywhere, 12345-1234'), ('Mia Red', '130 Elm St, Anywhere, 12345-1234'), ('Natalie Green', '140 Elm St, Anywhere, 12345-1234'), ('Oscar Purple', '150 Elm St, Anywhere, 12345-1234'), ('Parker Blue', '160 Elm St, Anywhere, 12345-1234'), ('Quinn Red', '170 Elm St, Anywhere, 12345-1234'), ('Riley Green', '180 Elm St, Anywhere, 12345-1234'), ('Samantha Purple', '190 Elm St, Anywhere, 12345-1234'), ('Tucker Blue', '200 Elm St, Anywhere, 12345-1234'), ('Ulysses Red', '210 Elm St, Anywhere, 12345-1234'), ('Vivian Green', '220 Elm St, Anywhere, 12345-1234'), ('Wade Purple', '230 Elm St, Anywhere, 12345-1234'), ('Xavier Blue', '240 Elm St, Anywhere, 12345-1234'), ('Yara Red', '250 Elm St, Anywhere, 12345-1234'), ('Zane Green', '260 Elm St, Anywhere, 12345-1234')	5 row(s) affected	0.00015 sec
25 17:15:56	INSERT INTO PatientRoom VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
26 17:15:56	INSERT INTO Room VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
27 17:15:56	INSERT INTO PhysicianRoom VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
28 17:15:56	INSERT INTO PhysicianPatientMonitoring VALUES (1, 1, 1, 2024-07-01, 2024-07-05), (2, 2, 2, 2024-07-01, 2024-07-05), (3, 3, 2, 2024-07-01, 2024-07-05), (4, 4, 4, 2024-07-01, 2024-07-05), (5, 5, 5, 2024-07-01, 2024-07-05)	5 row(s) affected	0.00019 sec
29 17:15:56	INSERT INTO Medication VALUES ('Aspirin', 5.00), ('Amoxicillin', 10.00), ('Ibuprofen', 15.00), ('Metformin', 12.00), ('Nutmeg', 1.00)	5 row(s) affected	0.00015 sec
30 17:15:56	INSERT INTO MedicationAdministration VALUES (1, 1, 1, 2024-07-01, 2024-07-05), (2, 2, 2, 2024-07-01, 2024-07-05), (3, 3, 3, 2024-07-01, 2024-07-05), (4, 4, 4, 2024-07-01, 2024-07-05), (5, 5, 5, 2024-07-01, 2024-07-05)	5 row(s) affected	0.00018 sec
31 17:15:56	INSERT INTO Instruction VALUES ('Complete bowel movement', 5.00), ('X-ray', 100.00), ('MRI scan', 500.00), ('PT', 50.00), ('Physical therapy session', 50.00), ('NUTRO', 1.00)	5 row(s) affected	0.00013 sec
32 17:15:56	INSERT INTO RoomCharge VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
33 17:15:56	INSERT INTO NurseExecution VALUES (1, 1, 1, 2024-07-01, 2024-07-05), (2, 2, 2, 2024-07-01, 2024-07-05), (3, 3, 3, 2024-07-01, 2024-07-05), (4, 4, 4, 2024-07-01, 2024-07-05), (5, 5, 5, 2024-07-01, 2024-07-05)	5 row(s) affected	0.00013 sec
34 17:15:56	INSERT INTO Invoice VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
35 17:15:56	INSERT INTO Payment VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00013 sec
36 17:15:56	SELECT ph.field_id, ph.expense AS total_expense FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.id WHERE ph.name = 'Dr. Michael Brown' AND po.order_id = 101 AND po.expense > 0	0 rows(s) returned	0.00022 sec / 0.00000...
37 17:15:56	SELECT COUNT(po.order_id) AS total_treatment FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.id WHERE ph.name = 'Dr. Michael Brown' AND po.order_id = 101 AND po.expense > 0	10 row(s) affected	0.00019 sec
38 17:15:56	SELECT COUNT(hr.record_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
39 17:15:56	SELECT COUNT(hr.record_id) AS total_accommodation FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
40 17:15:56	SELECT COUNT(hr.record_id) AS total_payment FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
41 17:15:56	SELECT COUNT(hr.record_id) AS total_expenditure FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
42 17:15:56	SELECT COUNT(pr.assignment_id) AS total_assignment FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number HAVING total_assignment > 0	0 rows(s) returned	0.00008 sec / 0.00000...
43 17:15:56	SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration m ON m.medication_id = m.medication_id GROUP BY m.name	0 rows(s) returned	0.00009 sec / 0.00000...
44 17:15:56	SELECT COUNT(ppm.record_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 row(s) returned	0.00011 sec / 0.00000...

Query Completed

## A.23 SQL Query: qr-phys-most-pts

```

4   FROM
5   PhysicianPatientMonitoring ppm
6   JOIN Physician ph ON ppm.physician_id = ph.physician_id
7   GROUP BY ph.name
8   ORDER BY total_patients DESC;
9

```

Result Grid

physician_name	total_patients
Dr. John Smith	1
Dr. Sarah Johnson	1
Dr. Michael Brown	1
Dr. Emily Davis	1
Dr. Robert Wilson	1

Action Output

Time	Action	Response	Duration / Fetch Time
26 17:15:56	INSERT INTO HealthRecord VALUES (1, 1, 'Hypertension', '2024-01-15', 'ongoing', 'Patient experiencing high blood pressure'), (2, 2, 'Fractured arm', '2024-02-20', 'resolved', 'Left arm fracture, cast applied'), (3, 3, 3, 2024-02-20, 'resolved', 'Left arm fracture, cast applied')	3 row(s) affected	0.00007 sec
27 17:15:56	INSERT INTO PatientRoom VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00030 sec
28 17:15:56	INSERT INTO Room VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00019 sec
29 17:15:56	INSERT INTO Medication VALUES ('Acetaminophen', 5.00), ('Ibuprofen', 10.00), ('Antibiotic', 15.00), ('Metformin', 12.00), ('Nutmeg', 1.00)	5 row(s) affected	0.00015 sec
30 17:15:56	INSERT INTO MedicationAdministration VALUES (1, 1, 1, 2024-07-01, 2024-07-05), (2, 2, 2, 2024-07-01, 2024-07-05), (3, 3, 3, 2024-07-01, 2024-07-05), (4, 4, 4, 2024-07-01, 2024-07-05), (5, 5, 5, 2024-07-01, 2024-07-05)	5 row(s) affected	0.00019 sec
31 17:15:56	INSERT INTO Instruction VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00015 sec
32 17:15:56	INSERT INTO RoomCharge VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00015 sec
33 17:15:56	INSERT INTO NurseExecution VALUES (1, 1, 1, 2024-07-01, 2024-07-05), (2, 2, 2, 2024-07-01, 2024-07-05), (3, 3, 3, 2024-07-01, 2024-07-05), (4, 4, 4, 2024-07-01, 2024-07-05), (5, 5, 5, 2024-07-01, 2024-07-05)	5 row(s) affected	0.00015 sec
34 17:15:56	INSERT INTO Invoice VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00015 sec
35 17:15:56	INSERT INTO Payment VALUES (1, 1, 101, 2, 100,00), (102, 2, 100,00), (103, 1, 100,00), (201, 2, 100,00), (202, 1, 100,00)	5 row(s) affected	0.00015 sec
36 17:15:56	SELECT COUNT(po.order_id) AS total_treatment FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.id WHERE ph.name = 'Dr. Michael Brown' AND po.order_id = 101 AND po.expense > 0	10 row(s) affected	0.00019 sec
37 17:15:56	SELECT COUNT(hr.record_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
38 17:15:56	SELECT COUNT(hr.record_id) AS total_accommodation FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
39 17:15:56	SELECT COUNT(hr.record_id) AS total_payment FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
40 17:15:56	SELECT COUNT(hr.record_id) AS total_expenditure FROM HealthRecord hr JOIN PhysicianPatientMonitoring ppm ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ppm.physician_id = ph.physician_id WHERE ph.name = 'Dr. Michael Brown' AND ppm.patient_id = 101	5 row(s) affected	0.00019 sec
41 17:15:56	SELECT COUNT(pr.assignment_id) AS total_assignment FROM PatientRoom pr JOIN Room r ON pr.room_number = r.room_number GROUP BY r.room_number HAVING total_assignment > 0	0 rows(s) returned	0.00008 sec / 0.00000...
42 17:15:56	SELECT m.name AS medication_name, SUM(m.price) AS total_revenue FROM MedicationAdministration m ON m.medication_id = m.medication_id GROUP BY m.name	0 rows(s) returned	0.00009 sec / 0.00000...
43 17:15:56	SELECT COUNT(ppm.record_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 row(s) returned	0.00011 sec / 0.00000...

Query Completed

## A.24 SQL Query: qr-pt-treatment-summary

```

1 • SELECT
2   p.name AS patient_name,
3   ph.name AS physician_name,
4   i.description AS treatment,
5   po.order_date
6   FROM
7   Patient p
8   JOIN PhysicianOrder po ON po.patient_id = p.patient_id
9   JOIN Physician ph ON po.physician_id = ph.physician_id
10  JOIN Instruction i ON po.instruction_id = i.instruction_id
11  ORDER BY p.name, po.order_date;

```

Result Grid

patient_name	physician_name	treatment	order_date
Alia Johnson	Dr. John Smith	Complete blood count	2024-07-02
Bob Smith	Dr. Sarah Johnson	X-ray	2024-07-03
Carol Brown	Dr. Michael Brown	MRI scan	2024-07-04
David Wilson	Dr. Emily Davis	Physical therapy session	2024-07-05
Eve Davis	Dr. Robert Wilson	Healthcare counseling	2024-07-06

Action Output 0

Time	Action	Response	Duration / Fetch Time
06:36 17-18-59	INSERT INTO Payment VALUES (1, 200.00, '2024-07-11'), (2, 300.00, '2024-07-11'), (3, 500.00, '2024-07-11'), (4, 400.00, '2024-07-11'), (5, 300.00, '2024-07-11')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.0010 sec
06:37 17-18-59	SELECT ph.field_of_expertise AS department, COUNT(po.order_id) AS total_treatments FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id GROUP BY ph.field_of_expertise	5 row(s) returned	0.0033 sec / 0.0000...
06:38 17-17-19	SELECT ph.name AS physician_name, COUNT(hr.record_id) AS chronic_cases FROM HealthRecord hr JOIN PhysicianPatientMonitoring ON hr.patient_id = ppm.patient_id JOIN Physician ph ON ...	5 row(s) returned	0.0011 sec / 0.0000...
06:39 17-17-28	SELECT r.room_number, AVG(DATEDIFF(pr.check_in_date, pr.check_out_date)) AS avg_length_of_stay FROM PatientRoom pr JOIN Room r ON pr.room_id = r.room_id GROUP BY r.room_number HAVING COUNT(r.room_number) > 1	5 row(s) returned	0.0016 sec / 0.0000...
06:40 17-17-37	SELECT p.name AS patient_name, COUNT(pr.visit_id) AS total_visits FROM PatientVisit pr JOIN Patient p ON pr.patient_id = p.patient_id GROUP BY p.name ORDER BY p.visit_id	5 row(s) returned	0.0009 sec / 0.0000...
06:41 17-17-58	SELECT COUNT(pr.visit_id) AS total_accommodation FROM PatientRoom pr JOIN Room r ON pr.room_id = r.room_id HAVING COUNT(pr.visit_id) > 1	5 row(s) returned	0.0009 sec / 0.0000...
06:42 17-17-58	SELECT m.name AS medication_name, SUM(pr.price) AS total_revenue FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 row(s) returned	0.0009 sec / 0.0000...
06:43 17-18-26	SELECT m.name AS medication_name, COUNT(ppm.visit_id) AS total_patients FROM PhysicianPatientMonitoring ppm JOIN Physician ph ON ppm.physician_id = ph.physician_id GROUP BY ph.name	5 row(s) returned	0.0011 sec / 0.0000...
06:44 17-18-36	SELECT p.name, SUM(pr.amount) AS total_payments FROM Patient p JOIN Payment pay ON p.patient_id = pay.patient_id GROUP BY p.name HAVING total_payments > (SELECT AVG(pr.payment))	2 row(s) returned	0.0012 sec / 0.0000...
06:45 17-18-44	SELECT n.name AS nurse_name, COUNT(m.administration_id) AS total_medications_administered FROM MedicationAdministration m JOIN Nurse n ON m.nurse_id = n.nurse_id GROUP BY n.name	5 row(s) returned	0.0015 sec / 0.0001...
06:46 17-18-44	SELECT n.name AS nurse_name, COUNT(m.administration_id) AS total_medications_administered FROM MedicationAdministration m JOIN Nurse n ON m.nurse_id = n.nurse_id GROUP BY n.name	5 row(s) returned	0.0015 sec / 0.0001...
06:47 17-22-19	CREATE VIEW PatientBillingOverview AS SELECT p.patient_id, p.name, SUM(pr.amount) AS total_bill, SUM(pr.payment) AS total_paid, p.balance AS current_balance FROM Patient p LEFT JOIN Payment pr ON p.patient_id = pr.patient_id	0 row(s) affected	0.039 sec
06:48 17-22-19	CREATE VIEW PhysicianPerformance AS SELECT ph.physician_id, ph.name, COUNT(DISTINCT pp.visit_id) AS patients_treated, COUNT(pp.visit_id) AS orders_given, AVG(pr.fee) AS avg_instru...	0 row(s) affected	0.039 sec
06:49 17-23-11	CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, r.capacity, COUNT(DISTINCT pr.visit_id) AS occupied_beds, (COUNT(DISTINCT pr.visit_id) / capacity) * 100 AS occupancy_r...	0 row(s) affected	0.030 sec
06:50 17-23-26	CREATE TRIGGER PreventOverBooking BEFORE INSERT ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; DECLARE room_capacity INT; SELECT capacity INTO room_capa...	0 row(s) affected	0.0031 sec
06:51 17-23-43	CREATE TRIGGER PreventOverBooking BEFORE UPDATE ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; DECLARE room_capacity INT; IF new.occupancy > room_capacity THEN SET new.occupancy = room_capacity; END IF; END	0 row(s) affected	0.0082 sec
06:52 17-23-43	CREATE TRIGGER PreventOverBooking AFTER INSERT ON PatientRoom FOR EACH ROW BEGIN UPDATE Patient p SET balance = balance + new.amount WHERE p.patient_id = new.patient_id; END	0 row(s) affected	0.0033 sec
06:53 17-24-04	CREATE TRIGGER after_payment_insert AFTER INSERT ON Payment FOR EACH ROW BEGIN UPDATE Patient p SET balance = p.balance + new.amount WHERE p.patient_id = new.patient_id; END;	0 row(s) affected	0.0034 sec
06:54 17-24-44	SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician...	5 row(s) returned	0.0010 sec / 0.0000...

## A.25 SQL Query: qr-pt-high-med

```

1 • SELECT
2   p.name,
3   COUNT(ma.administration_id) AS num_medications
4   FROM
5   MedicationAdministration ma
6   JOIN Patient p ON ma.patient_id = p.patient_id
7   GROUP BY p.name
8   ORDER BY num_medications DESC
9   LIMIT 5;

```

Result Grid

name	num_medications
Alia Johnson	1
Bob Smith	1
Carol Brown	1
David Wilson	1
Eve Davis	1

Action Output 0

Time	Action	Response	Duration / Fetch Time
06:49 17-23-11	CREATE VIEW PhysicianPerformance AS SELECT ph.physician_id, ph.name, COUNT(DISTINCT pp.visit_id) AS patients_treated, COUNT(pp.visit_id) AS orders_given, AVG(pr.fee) AS avg_instru...	0 row(s) affected	0.0030 sec
06:50 17-23-26	CREATE VIEW RoomOccupancyRate AS SELECT r.room_number, r.capacity, COUNT(DISTINCT pr.visit_id) AS occupied_beds, (COUNT(DISTINCT pr.visit_id) / capacity) * 100 AS occupancy_r...	0 row(s) affected	0.0031 sec
06:51 17-23-43	CREATE TRIGGER PreventOverBooking BEFORE INSERT ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; DECLARE room_capacity INT; SELECT capacity INTO room_capa...	0 row(s) affected	0.0082 sec
06:52 17-23-43	CREATE TRIGGER PreventOverBooking BEFORE UPDATE ON PatientRoom FOR EACH ROW BEGIN DECLARE current_occupancy INT; DECLARE room_capacity INT; IF new.occupancy > room_capacity THEN SET new.occupancy = room_capacity; END IF; END	0 row(s) affected	0.0033 sec
06:53 17-24-04	CREATE TRIGGER after_payment_insert AFTER INSERT ON Payment FOR EACH ROW BEGIN UPDATE Patient p SET balance = balance + new.amount WHERE p.patient_id = new.patient_id; END	0 row(s) affected	0.0034 sec
06:54 17-24-44	SELECT p.name AS patient_name, ph.name AS physician_name, i.description AS treatment, po.order_date FROM Patient p JOIN PhysicianOrder po ON p.patient_id = po.patient_id JOIN Physician...	0 row(s) returned	0.0010 sec / 0.0000...
06:55 17-25-05	SELECT ph.field_of_expertise AS department, SUM(pr.fee) AS total_revenue FROM PhysicianOrder po JOIN Physician ph ON po.physician_id = ph.physician_id JOIN Instruction i ON po.instruction_id = i.instruction_id WHERE DATEDIFF(pr.check_out_date, pr.check_in_date) AS length_of_stay	0 row(s) returned	0.0010 sec / 0.0000...
06:56 17-25-05	SELECT p.name, COUNT(ma.administration_id) AS num_medications FROM MedicationAdministration ma JOIN Patient p ON ma.patient_id = p.patient_id GROUP BY p.name ORDER BY num_medicati...	0 row(s) returned	0.0021 sec / 0.0001...
06:57 17-25-15	START TRANSACTION	0 row(s) affected	0.00023 sec
06:58 17-25-15	INSERT INTO Patient (patient_id, name, address, phone) VALUES (6, 'Frank Williams', '600 New St., Novem', '555-6667')	1 row(s) affected	0.0001 sec
06:59 17-25-15	INSERT INTO HealthRecord (record_id, patient_id, disease, diagnosis_date, status) VALUES (6, 6, 'Influenza', 'CURDATE()', 'ongoing', 'Patient admitted with severe flu symptoms')	1 row(s) affected	0.0001 sec
06:60 17-25-15	INSERT INTO PatientRoom (assignment_id, patient_id, room_number, check_in_date, check_out_date) VALUES (6, 6, 101, CURDATE()), DATE_ADD(CURDATE(), INTERVAL 5 DAY)	1 row(s) affected	0.0020 sec
06:61 17-25-15	COMMIT	0 row(s) affected	0.00085 sec
06:62 17-25-28	START TRANSACTION	0 row(s) affected	0.00005 sec
06:63 17-25-28	INSERT INTO Payable (payable_id, patient_id, account_number, issue_date, start_date, end_date) VALUES (6, 1, 'ACCO006', CURDATE(), (SELECT check_in_date FROM PatientRoom WHERE patient_id = 6), (SELECT check_out_date FROM PatientRoom WHERE patient_id = 6))	1 row(s) affected Rows matched: 0 Changed: 0 W...	0.0001 sec
06:64 17-25-28	INSERT INTO Payable (payable_id, patient_id, account_number, description, date) SELECT (SELECT COALESCE(MAX(payable_id), 0) FROM Payable) + ROW_NUMBER() OVER (PARTITION BY patient_id ORDER BY payable_id), 6 AS invoice_id, ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.0003 sec
06:65 17-25-28	INSERT INTO Payable (payable_id, patient_id, account_number, description, date) SELECT (SELECT COALESCE(MAX(payable_id), 0) FROM Payable) + ROW_NUMBER() OVER (PARTITION BY patient_id ORDER BY payable_id), 6 AS invoice_id, ...	0 row(s) affected	0.00034 sec
06:66 17-25-28	COMMIT	0 row(s) affected	0.00009 sec / 0.0000...