Hospital Management System

Course: Database Management System

Document Name: Phase 3 Submission

Prepared by: Meet Saraswat

Contents

0. Pre-Illumination

This report outlines the implementation phase of the database project, focusing on the creation of the database, table setup, data population, and SQL queries. Our project utilizes the MySQL database management system. Part 1 is the creation of the database, including tables, all other structures as well as constraints, data type and format, Part 2 is the query scenario design and implementation.

1. Creation of Database with SQL Statements

1.1 Table Creation

Patient:

```
CREATE TABLE Patient (
    Patient_ID INT NOT NULL,
    Patient_FName VARCHAR(20) NOT NULL,
    Patient_LName VARCHAR(20) NOT NULL,
    Phone VARCHAR(12) NOT NULL,
    Blood_Type  VARCHAR(5) NOT NULL,
    Email VARCHAR(50),
    Gender  VARCHAR(10),
    Condition_ VARCHAR(30),
    Admission_Date DATE,
    Discharge_Date DATE,
    PRIMARY KEY (Patient_ID)
    );
```

Department:

```
CREATE TABLE Department (
    Dept_ID INT NOT NULL,
    Dept_Head VARCHAR(20) NOT NULL,
    Dept_Name VARCHAR(15) NOT NULL,
    Emp_Count INT,
    PRIMARY  KEY (Dept_ID)
```

```
);
```

Staff:

```
CREATE TABLE Staff (
    Emp_ID INT  NOT NULL,
    Emp_FName  VARCHAR(20) NOT NULL,
    Emp_LName  VARCHAR(20) NOT NULL,
    Date_Joining  DATE,
    Date_Seperation DATE,
    Emp_Type VARCHAR(15) NOT NULL,
    Email  VARCHAR(50),
    Address  VARCHAR(50) NOT NULL,
    Dept_ID  INT NOT NULL,
    SSN  INT NOT NULL,
    PRIMARY KEY (Emp_ID),
    FOREIGN KEY (Dept_ID) REFERENCES Department  (Dept_ID)
);
```

Doctor:

```
CREATE TABLE Doctor (
    Doctor_ID INT NOT NULL,
    Qualifications VARCHAR(15) NOT NULL,
    Emp_ID INT NOT NULL,
    Specialization VARCHAR(20) NOT NULL,
        Dept_ID INT NOT NULL,
    PRIMARY KEY (Doctor_ID),
    FOREIGN KEY (Emp_ID) REFERENCES Staff (Emp_ID),
    FOREIGN KEY (Dept_ID) REFERENCES Department (Dept_ID)
);
```

Nurse:

```
CREATE TABLE Nurse (
    Nurse_ID INT  NOT NULL,
    Patient_ID  INT  NOT NULL,
```

```
    Emp_ID  INT NOT NULL,
    Dept_ID INT NOT NULL,
    PRIMARY KEY(Nurse_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
    FOREIGN KEY (Emp_ID) REFERENCES Staff  (Emp_ID),
    FOREIGN KEY (Dept_ID) REFERENCES Department (Dept_ID)
);
```

Emergency_Contact:

```
CREATE TABLE Emergency_Contact(
    Contact_ID INT  NOT NULL,
    Contact_Name VARCHAR(20) NOT NULL,
    Phone VARCHAR(12) NOT NULL,
    Relation VARCHAR(20) NOT NULL,
          Patient_ID  INT NOT NULL,
    PRIMARY KEY (Contact_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
);
```

Payroll:

```
  CREATE TABLE Payroll (
    Account_No VARCHAR(25) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    Bonus DECIMAL(10,2),
    Emp_ID INT NOT NULL,
    IBAN VARCHAR(25),
    PRIMARY KEY (Account_No),
    FOREIGN KEY (Emp_ID) REFERENCES Staff (Emp_ID)
   );
```

Lab_Screening:

```
 CREATE TABLE  Lab_Screening (
    Lab_ID INT NOT NULL,
    Patient_ID  INT  NOT NULL,
```

```
        Technician_ID  INT  NOT NULL,
    Doctor_ID  INT NOT NULL,
        Test_Cost  DECIMAL(10,2),
        Date  DATE  NOT NULL,
        PRIMARY KEY (Lab_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES Doctor (Doctor_ID)
);
```

Insurance:

```
CREATE TABLE Insurance (
    Policy_Number VARCHAR(20) NOT NULL,
    Patient_ID INT NOT NULL,
    Ins_Code VARCHAR(20) NOT NULL,
    End_Date VARCHAR(10),
    Provider VARCHAR(20),
    Plan VARCHAR(20),
    Co_Pay  DECIMAL(10,2),
    Coverage VARCHAR(20),
    Maternity  BOOLEAN,
    Dental  BOOLEAN,
    Optical BOOLEAN,
    PRIMARY  KEY (Policy_Number),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
);
```

Medicine:

```
CREATE TABLE Medicine (
    Medicine_ID INT  NOT NULL,
    M_Name VARCHAR(20) NOT NULL,
    M_Quantity INT NOT NULL,
    M_Cost  Decimal(10,2),
   PRIMARY KEY (Medicine_ID)
```

```
);
```

Prescription:

```
CREATE TABLE Prescription (
    Prescription_ID INT  NOT NULL,
    Patient_ID  INT  NOT NULL,
    Medicine_ID  INT  NOT NULL,
    Date  DATE,
    Dosage  INT,
    Doctor_ID INT NOT NULL,
    PRIMARY KEY (Prescription_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES Doctor (Doctor_ID),
    FOREIGN KEY (Medicine_ID) REFERENCES Medicine (Medicine_ID)
);
```

Medical_History:

```
CREATE TABLE Medical_History (
    Record_ID  INT NOT NULL,
    Patient_ID  INT NOT NULL,
        Allergies VARCHAR(50),
    Pre_Conditions VARCHAR(50),
    PRIMARY KEY (Record_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
);
```

Appointment:

```
CREATE TABLE Appointment (
    Appt_ID INT  NOT NULL,
    Scheduled_On  DATETIME NOT NULL,
    Date  DATE,
    Time TIME,
    Doctor_ID INT NOT NULL,
    Patient_ID  INT NOT NULL,
```

```
    PRIMARY KEY (Appt_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES Doctor (Doctor_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
);
```

Room:

```
 CREATE TABLE  Room (
    Room_ID INT NOT NULL,
    Room_Type VARCHAR(50) NOT NULL,
    Patient_ID  INT  NOT NULL,
    Room_Cost  DECIMAL(10,2),
    PRIMARY KEY (Room_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
    );
```

Bill:

```
CREATE TABLE Bill (
    Bill_ID INT NOT NULL,
    Date  DATE,
    Room_Cost Decimal(10,2),
    Test_Cost  DECIMAL(10,2),
    Other_Charges  DECIMAL(10,2),
          M_Cost DECIMAL(10,2),
    Total  DECIMAL(10,2),
    Patient_ID INT NOT NULL,
    Remaining_Balance DECIMAL(10,2),
    Policy_Number VARCHAR(20) NOT NULL,
    PRIMARY KEY (Bill_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
    FOREIGN KEY (Policy_Number) REFERENCES Insurance (Policy_Number)
);
```

1.2 A Database State

To ensure the database is populated for testing and development purposes, sample dummy data was inserted into each table. The following records were added to each table, maintaining data consistency and validity. Only a part of data will be shown here since there are multiple rows for each table.

Insertion of Table “Patient”

INSERT INTO Patient (Patient_ID, Patient_FName, Patient_LName, Phone, Blood_Type, Email, Gender, Condition_, Admission_Date, Discharge_Date)
VALUES
(1, 'John', 'Doe', '555-1234', 'A+', 'john.doe@email.com', 'Male', 'Injury', '2023-01-01', '2023-01-10'),
(2, 'Jane', 'Smith', '555-5678', 'O-', 'jane.smith@email.com', 'Female', 'Flu', '2023-02-05', '2023-02-15'),
(3, 'Michael', 'Johnson', '555-8765', 'B+', 'michael.johnson@email.com', 'Male', 'Allergies', '2023-03-10', '2023-03-20'), …..

Patient_ID	Patient_FName	Patient_LName	Phone	Blood_Type	Email	Gender	Condition_	Admission_Date	Discharge_Date
1	John	Doe	555-1234	A+	john.doe@email.com	Male	Injury	2023-01-01	2023-01-10
2	Jane	Smith	555-5678	O-	jane.smith@email.com	Female	Flu	2023-02-05	2023-02-15
3	Michael	Johnson	555-8765	B+	michael.johnson@email.com	Male	Allergies	2023-03-10	2023-03-20
4	Emily	Williams	555-2345	AB-	emily.williams@email.com	Female	Headache	2023-04-15	2023-04-25
5	Robert	Brown	555-5432	A-	robert.brown@email.com	Male	Fracture	2023-05-20	2023-05-30
6	Alice	Davis	555-7890	O+	alice.davis@email.com	Female	Respiratory Infection	2023-06-25	2023-07-05
7	Christopher	Miller	555-4321	B-	chris.miller@email.com	Male	Back Pain	2023-07-30	2023-08-09
8	Olivia	Jones	555-9876	AB+	olivia.jones@email.com	Female	Concussion	2023-09-04	2023-09-14
9	William	Wilson	555-8765	A+	william.wilson@email.com	Male	Appendicitis	2023-10-09	2023-10-19
10	Sophia	Moore	555-6543	O-	sophia.moore@email.com	Female	Diabetes	2023-11-14	2023-11-24

Image 1. Patient

Insertion of Table “Department”

INSERT INTO Department (Dept_ID, Dept_Head, Dept_Name, Emp_Count)
VALUES
(1, 'John Smith', 'Cardiology_1', 5),
(2, 'Isabella Fisher', 'Emergency_2', 5),
(3, 'James White', 'Diagnostic_3', 5), …..

	Dept_ID	Dept_Head	Dept_Name	Emp_Count
	1	John Smith	Cardiology_1	5
	2	Isabella Fisher	Emergency_2	5
	3	James White	Diagnostic_3	5
	4	Emily Davis	Cardiology_4	5
	5	Mia Anderson	Emergency_5	5
	6	Lily Bell	Diagnostic_6	5

Image 2. Department

Insertion of Table "Staff"

INSERT INTO Staff (Emp_ID, Emp_FName, Emp_LName, Date_Joining, Date_Separation, Emp_Type, Email, Address, Dept_ID, SSN)
VALUES
(1, 'John', 'Smith', '2022-01-01', NULL, 'Doctor', 'john.smith@email.com', '123 Main St', 1, '123456789'),
(2, 'Jane', 'Johnson', '2022-02-15', NULL, 'Nurse', 'jane.johnson@email.com', '456 Oak St', 2, '234567890'),
(3,'Michael', 'Williams', '2022-03-10', NULL, 'Lab Technician', 'michael.williams@email.com', '789 Pine St', 3, '345678901'),.....

	Emp_ID	Emp_FName	Emp_LName	Date_Joining	Date_Separation	Emp_Type	Email	Address	Dept_ID	SSN
	1	John	Smith	2022-01-01	NULL	Doctor	john.smith@email.com	123 Main St	1	123456789
	2	Jane	Johnson	2022-02-15	NULL	Nurse	jane.johnson@email.com	456 Oak St	2	234567890
	3	Michael	Williams	2022-03-10	NULL	Lab Technician	michael.williams@email.com	789 Pine St	3	345678901
	4	Emily	Davis	2022-04-25	NULL	Doctor	emily.davis@email.com	321 Cedar St	4	456789012
	5	Robert	Brown	2022-05-30	NULL	Nurse	robert.brown@email.com	654 Elm St	5	567890123
	6	Alice	Miller	2022-07-05	NULL	Lab Technician	alice.miller@email.com	987 Maple St	6	678901234

Image 3. Staff

Insertion of Table "Doctor"

INSERT INTO Doctor (Doctor_ID, Qualifications, Emp_ID, Specialization, Dept_ID)
VALUES

(101, 'MD', 1, 'General Medicine', 1),

(40, 'MD', 4, 'General Medicine', 4),

(70, 'MD', 7, 'General Medicine', 7), …..

Doctor_ID	Qualifications	Emp_ID	Specialization	Dept_ID
40	MD	4	General Medicine	4
70	MD	7	General Medicine	7
101	MD	1	General Medicine	1
102	MD	10	General Medicine	10
130	MD	13	General Medicine	2
160	MD	16	General Medicine	5

Image 4. Doctor

Insertion of Table “Nurse”

INSERT INTO Nurse (Nurse_ID, Patient_ID, Emp_ID, Dept_ID)

VALUES

(2, 1, 2, 2),

(5, 2, 5, 5),

(8, 3, 8, 8), ….

Nurse_ID	Patient_ID	Emp_ID	Dept_ID
2	1	2	2
5	2	5	5
8	3	8	8
11	4	11	1
14	5	14	4
17	6	17	7
20	7	20	10

Image 5. Nurse

Insertion of Table “Emergency_Contact”

INSERT INTO Emergency_Contact (Contact_ID, Contact_Name, Phone, Relation, Patient_ID)
VALUES
(10, 'John Doe', '555-1234', 'Parent', 1),
(20, 'Jane Smith', '555-5678', 'Sibling', 2),
(30, 'Robert Johnson', '555-8765', 'Parent', 3), …..

Contact_ID	Contact_Name	Phone	Relation	Patient_ID
10	John Doe	555-1234	Parent	1
20	Jane Smith	555-5678	Sibling	2
30	Robert Johnson	555-8765	Parent	3
40	Emily Davis	555-4321	Sibling	4
50	Michael Wilson	555-9876	Parent	5
60	Alice Taylor	555-3456	Sibling	6
70	David Brown	555-6789	Parent	7
80	Susan Miller	555-2345	Sibling	8

Image 6. Emergency Contact

Insertion of Table “Payroll”

INSERT INTO Payroll (Account_No, Salary, Bonus, Emp_ID, IBAN)
VALUES
(9078881226, 100000.00, 20000.00, 1, 'IBAN1'),
(7798155825, 80000.00, 15000.00, 2, 'IBAN2'),
(3215909832, 75000.00, 10000.00, 3, 'IBAN3'), …..

Account_No	Salary	Bonus	Emp_ID	IBAN
1088886261	75000.00	10000.00	42	IBAN42
1124642741	80000.00	15000.00	5	IBAN5
1402638832	75000.00	10000.00	21	IBAN21
1566676915	80000.00	15000.00	14	IBAN14
1943051877	100000.00	20000.00	7	IBAN7
2220438658	100000.00	20000.00	25	IBAN25
2460106576	80000.00	15000.00	44	IBAN44

Image 7. Payroll

Insertion of Table "Lab_Screening"

INSERT INTO lab_screening (Lab_ID, Patient_ID, Technician_ID, Doctor_ID, Test_Cost, Date)

VALUES

(1, 1, 6, 101, 75, '2023-12-05'),

(2, 2, 3, 101, 50, '2023-12-06'),

(3, 3, 9, 280, 80, '2023-12-07'), ….

Lab_ID	Patient_ID	Technician_ID	Doctor_ID	Test_Cost	Date
1	1	6	101	75.00	2023-12-05
2	2	3	101	50.00	2023-12-06
3	3	9	280	80.00	2023-12-07
4	4	7	220	60.00	2023-12-08
5	5	8	70	70.00	2023-12-09
6	6	2	102	65.00	2023-12-10
7	7	1	220	55.00	2023-12-11

Image 8. Lab_Screening

Insertion of Table “Insurance”

INSERT INTO insurance (Policy_Number, Patient_ID, Ins_Code, End_Date, Provider, Plan, Co_Pay, Coverage, Maternity, Dental, Optical)

VALUES
('A123456', 1, 'INS001', '2024-12-31', 'XYZ Insurance', 'Standard Plan', 20.00, 'Health Insurance', true, false, false),
('B789012', 2, 'INS002', '2024-11-30', 'ABC Insurance', 'Extended Plan', 30.00, 'Health and Dental Insurance', false, true, false),
('C345678', 3, 'INS003', '2024-10-31', 'DEF Insurance', 'Basic Plan', 15.00, 'Health Insurance', true, false, true), …..

Policy_Number	Patient_ID	Ins_Code	End_Date	Provider	Plan	Co_Pay	Coverage	Maternity	Dental	Optical
A123456	1	INS001	2024-12-31	XYZ Insurance	Standard Plan	20.00	Health Insurance	1	0	0
AA987123	27	INS027	2022-10-31	RWS Insurance	Basic Plan	15.00	Health Insurance	1	0	1
B789012	2	INS002	2024-11-30	ABC Insurance	Extended Plan	30.00	Health and Dental Insurance	0	1	0
BB654321	28	INS028	2022-09-30	XJY Insurance	Family Plan	25.00	Health and Dental Insurance	0	0	1
C345678	3	INS003	2024-10-31	DEF Insurance	Basic Plan	15.00	Health Insurance	1	0	1
CC321789	29	INS029	2022-08-31	KLN Insurance	Standard Plan	20.00	Health Insurance	1	1	0
D901234	4	INS004	2024-09-30	GHI Insurance	Family Plan	25.00	Health and Dental Insurance	1	1	1
DD901234	30	INS030	2022-07-31	OPU Insurance	Extended Plan	30.00	Health and Dental Insurance	0	1	1

Image 9. Insurance

Insertion of Table “Medicine”

INSERT INTO Medicine (Medicine_ID, M_Name, M_Quantity, M_Cost)
VALUES
(1, 'Aspirin', 100, 5.99),
(2, 'Ibuprofen', 50, 8.49),
(3, 'Acetaminophen', 75, 6.25), ….

	Medicine_ID	M_Name	M_Quantity	M_Cost
	1	Aspirin	100	5.99
	2	Ibuprofen	50	8.49
	3	Acetaminophen	75	6.25
	4	Amoxicillin	30	12.75
	5	Ciprofloxacin	20	15.99
	6	Lisinopril	40	9.75
	7	Atorvastatin	60	22.50

Image 10. Medicine

Insertion of Table “Prescription”

INSERT INTO Prescription (Prescription_ID, Patient_ID, Medicine_ID, Date, Dosage, Doctor_ID)
VALUES
(13, 1, 1, '2023-12-05', 2, 101),
(23, 2, 3, '2023-12-06', 1, 101),
(33, 32, 6, '2023-12-07', 1, 101), ….

	Prescription_ID	Patient_ID	Medicine_ID	Date	Dosage	Doctor_ID
	13	1	1	2023-12-05	2	101
	23	2	3	2023-12-06	1	101
	33	32	6	2023-12-07	1	101
	43	18	10	2023-12-08	2	220
	53	49	15	2023-12-09	1	310
	63	31	7	2023-12-10	2	400

Image 11. Prescription

Insertion of Table “Medical_History”

INSERT INTO Medical_History (Record_ID, Patient_ID, Allergies, Pre_Conditions)
VALUES
(11, 1, 'Penicillin', 'None'),
(21, 2, 'None', 'Asthma'),
(31, 3, 'Sulfa Drugs', 'High Blood Pressure'), ….

Record_ID	Patient_ID	Allergies	Pre_Conditions
11	1	Penicillin	None
21	2	None	Asthma
31	3	Sulfa Drugs	High Blood Pressure
41	4	Pollen	None
51	5	Shellfish	Diabetes
61	6	None	Heart Disease
71	7	Peanuts	High Cholesterol
81	8	None	Anemia

Image 12. Medical_History

Insertion of Table “Appointment”

INSERT INTO Appointment (Appt_ID, Scheduled_On, Date, Time, Doctor_ID, Patient_ID)
VALUES
(1, '2023-12-02 14:24:16', '2023-12-13', '21:00:00', 101, 1),
(2, '2023-12-02 14:24:16', '2023-12-15', '14:00:00', 101, 2),
(3, '2023-12-02 14:24:16', '2023-12-18', '10:00:00', 101, 32), …..

	Appt_ID	Scheduled_On	Date	Time	Doctor_ID	Patient_ID
	1	2023-12-02 14:24:16	2023-12-13	21:00:00	101	1
	2	2023-12-02 14:24:16	2023-12-15	14:00:00	101	2
	3	2023-12-02 14:24:16	2023-12-18	10:00:00	101	32
	4	2023-12-02 14:24:16	2023-12-10	16:00:00	220	18
	5	2023-12-02 14:24:16	2023-12-18	22:00:00	310	49
	6	2023-12-02 14:24:16	2023-12-08	19:00:00	400	31
	7	2023-12-02 14:24:16	2023-12-13	13:00:00	310	25

Image 13. Appointment

Insertion of Table “Room”

INSERT INTO Room (Room_ID, Room_Type, Patient_ID, Room_Cost)

VALUES

(101, 'Premium', 1, 150.00),

(205, 'Deluxe', 2, 200.00),

(307, 'Deluxe', 3, 200.00), …..

	Room_ID	Room_Type	Patient_ID	Room_Cost
	101	Premium	1	150.00
	103	Premium	10	150.00
	110	Premium	19	150.00
	204	Deluxe	11	200.00
	205	Deluxe	2	200.00
	209	Deluxe	20	200.00
	301	Executive	21	250.00
	306	Executive	12	250.00

Image 14. Room

Insertion of Table “Bill”

INSERT INTO bill (Bill_ID, Date, Room_Cost, Test_Cost, Other_Charges, M_Cost, Total, Patient_ID, Remaining_Balance, Policy_Number)

VALUES

(60, '2023-12-05', 150, 75, 20, 5.99, 250.99, 1, 150, 'A123456'),

(61, '2023-12-06', 200, 50, 25, 12.75, 287.75, 2, 200, 'B789012'),

(62, '2023-12-07', 200, 80, 15, 6.25, 301.25, 3, 175, 'C345678'), ….

Bill_ID	Date	Room_Cost	Test_Cost	Other_Charges	M_Cost	Total	Patient_ID	Remaining_Balance	Policy_Number
60	2023-12-05	150.00	75.00	20.00	5.99	250.99	1	150.00	A123456
61	2023-12-06	200.00	50.00	25.00	12.75	287.75	2	200.00	B789012
62	2023-12-07	200.00	80.00	15.00	6.25	301.25	3	175.00	C345678
63	2023-12-08	250.00	60.00	20.00	9.75	339.75	4	180.00	D901234
64	2023-12-09	200.00	70.00	25.00	14.99	309.99	5	190.00	E567890
65	2023-12-10	250.00	65.00	20.00	22.50	357.50	6	220.00	F123789
66	2023-12-11	150.00	55.00	15.00	18.75	238.75	7	160.00	G456321
67	2023-12-12	200.00	60.00	20.00	12.75	292.75	8	200.00	H987654

Image 15. Bll

---

2. Query Scenario Design

Query 01: The hospital management wants to calculate the total revenue generated by the hospital, including room charges, lab screening charges, and other miscellaneous charges, for a specific date range.

```
SELECT DATE_FORMAT(b.Date, '%Y-%m-%d') AS Billing_Date,

SUM(b.Room_Cost + b.Test_Cost + b.Other_Charges + b.M_Cost) AS
Total_Revenue

FROM bill b

WHERE       b.Date BETWEEN '2023-12-01' AND '2023-12-31'

GROUP BY   DATE_FORMAT(b.Date, '%Y-%m-%d')

ORDER BY   DATE_FORMAT(b.Date, '%Y-%m-%d');
```

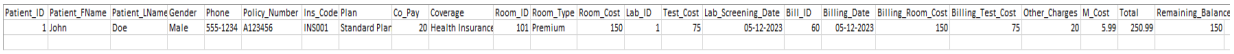
Result for Query 01:

Billing_Date	Total_Revenue
2023-12-05	250.99
2023-12-06	287.75
2023-12-07	301.25
2023-12-08	339.75
2023-12-09	309.99
2023-12-10	357.50
2023-12-11	238.75
2023-12-12	292.75
2023-12-13	364.99
2023-12-14	230.49

Query 02: A hospital administrator wants to retrieve information about a specific patient, including their personal details, insurance information, room history, lab screening details, and billing history. This comprehensive query will provide a detailed overview of the patient's interactions with the hospital. (Retrieve Patient Information and Associated Bills)

```
SELECT
    p.Patient_ID, p.Patient_FName, p.Patient_LName, p.Gender, p.Phone, i.Policy_Number, i.Ins_Code, i.Plan, i.Co_Pay, i.Coverage, r.Room_ID, r.Room_Type, r.Room_Cost, l.Lab_ID, l.Test_Cost, l.Date AS Lab_Screening_Date, b.Bill_ID, b.Date AS Billing_Date, b.Room_Cost AS Billing_Room_Cost, b.Test_Cost AS Billing_Test_Cost, b.Other_Charges, b.M_Cost, b.Total, b.Remaining_Balance
FROM patient p
JOIN insurance i ON p.Patient_ID = i.Patient_ID
LEFT JOIN room r ON p.Patient_ID = r.Patient_ID
LEFT JOIN lab_screening l ON p.Patient_ID = l.Patient_ID
LEFT JOIN bill b ON p.Patient_ID = b.Patient_ID
WHERE
    p.Patient_ID = 1;  -- Replace with the desired Patient_ID
```

Result for Query 02:



Patient_ID	Patient_FName	Patient_LName	Gender	Phone	Policy_Number	Ins_Code	Plan	Co_Pay	Coverage	Room_ID	Room_Type	Room_Cost	Lab_ID	Test_Cost	Lab_Screening_Date	Bill_ID	Billing_Date	Billing_Room_Cost	Billing_Test_Cost	Other_Charges	M_Cost	Total	Remaining_Balance
1	John	Doe	Male	555-1234	A123456	INS001	Standard Plan	20	Health Insurance	101	Premium	150	1	75	05-12-2023	60	05-12-2023	150	75	20	5.99	250.99	150

Query 03: Retrieve Patient Information for those who are currently prescribed with Aspirin.

```
SELECT DISTINCT P.Patient_ID, P.Patient_FName, P.Patient_LName, Pre.Dosage
FROM Patient P
INNER JOIN Prescription Pre ON P.Patient_ID = Pre.Patient_ID
INNER JOIN Medicine M ON Pre.Medicine_ID = M.Medicine_ID
WHERE M.M_Name = 'Aspirin';
```

Result for Query 03:

Patient_ID	Patient_FName	Patient_LName	Dosage
1	John	Doe	2
36	Liam	Long	2
32	William	Reed	1

Query 04: Retrieve doctors and their associated department.

```
SELECT D.Doctor_ID, D.Qualifications, D.Specialization, D.Dept_ID, Dep.Dept_Name
FROM Doctor D
JOIN Department Dep ON D.Dept_ID = Dep.Dept_ID;
```

Result for Query 04:

Doctor_ID	Qualifications	Specialization	Dept_ID	Dept_Name
101	MD	General Medicine	1	Cardiology_1
220	MD	General Medicine	1	Cardiology_1
430	MD	General Medicine	1	Cardiology_1
130	MD	General Medicine	2	Emergency_2
340	MD	General Medicine	2	Emergency_2
40	MD	General Medicine	4	Cardiology_4
250	MD	General Medicine	4	Cardiology_4
460	MD	General Medicine	4	Cardiology_4
160	MD	General Medicine	5	Emergency_5
370	MD	General Medicine	5	Emergency_5
70	MD	General Medicine	7	Cardiology_7
280	MD	General Medicine	7	Cardiology_7
490	MD	General Medicine	7	Cardiology_7
190	MD	General Medicine	8	Emergency_8
400	MD	General Medicine	8	Emergency_8
102	MD	General Medicine	10	Cardiology_10
310	MD	General Medicine	10	Cardiology_10

Query 05: Retrieve total number of appointments for each doctor by date.

```
SELECT A.Date, A.Doctor_ID, COUNT(*) AS TotalAppointments
FROM Appointment A
GROUP BY A.Date, A.Doctor_ID
ORDER BY A.Date, A.Doctor_ID;
```

Result for Query 05:

Date	Doctor_ID	TotalAppointme...
2023-12-07	130	1
2023-12-08	70	1
2023-12-08	130	1
2023-12-08	250	1
2023-12-08	370	1
2023-12-08	400	1

Query 06: Retrieve emergency contact details along with patient information.

```
SELECT EC.Contact_ID, EC.Contact_Name, EC.Phone, EC.Relation, P.Patient_FName,
P.Patient_LName
FROM Emergency_Contact EC
JOIN Patient P ON EC.Patient_ID = P.Patient_ID;
```

Result for Query 06:

Contact_ID	Contact_Name	Phone	Relation	Patient_FName	Patient_LName
10	John Doe	555-1234	Parent	John	Doe
20	Jane Smith	555-5678	Sibling	Jane	Smith
30	Robert Johnson	555-8765	Parent	Michael	Johnson
40	Emily Davis	555-4321	Sibling	Emily	Williams
50	Michael Wilson	555-9876	Parent	Robert	Brown
60	Alice Taylor	555-3456	Sibling	Alice	Davis
70	David Brown	555-6789	Parent	Christopher	Miller
80	Susan Miller	555-2345	Sibling	Olivia	Jones
90	Chris Anderson	555-7890	Parent	William	Wilson
100	Emma White	555-9012	Sibling	Sophia	Moore
110	Mark Thompson	555-5432	Sibling	David	Taylor
120	Laura Harris	555-8765	Parent	Emma	Anderson
130	Brian Jackson	555-2345	Sibling	James	Martin
140	Olivia Johnson	555-7890	Parent	Grace	White
150	Alex Turner	555-4321	Sibling	Benjamin	Hall
160	Grace Martin	555-5678	Parent	Lily	Young

Query 07: Retrieve the names and contact information of patients who have a medical history related to allergies.

```
SELECT Patient_FName,Patient_LName,Phone

FROM Patient

JOIN Medical_History ON Patient.Patient_ID = Medical_History.Patient_ID

WHERE Medical_History.Allergies IS NOT NULL;
```

Result for Query 07:

Patient_FName	Patient_LName	Phone
John	Doe	555-1234
Jane	Smith	555-5678
Michael	Johnson	555-8765
Emily	Williams	555-2345
Robert	Brown	555-5432

Query 08: Find the total bill amount paid by patients who have insurance coverage.

```
SELECT Patient_FName, COALESCE(SUM(Bill.Total), 0) AS TotalBillPaid

FROM Patient

LEFT JOIN Bill ON Patient.Patient_ID = Bill.Patient_ID

LEFT JOIN Insurance ON Patient.Patient_ID = Insurance.Patient_ID

GROUP BY Patient.Patient_FName;
```

Result for Query 08:

Patient_FName	TotalBillPaid
John	250.99
Jane	287.75
Michael	301.25
Emily	339.75
Robert	309.99

Query 09: Retrieve the prescription details along with patient information.

```
SELECT Prescription.Prescription_ID, Prescription.Date, Patient.Patient_ID,
Patient.Patient_FName, Patient.Patient_LName, Medicine.Medicine_ID,
Medicine.M_Name, Prescription.Dosage

FROM Prescription

JOIN Patient ON Prescription.Patient_ID = Patient.Patient_ID

JOIN Medicine ON Prescription.Medicine_ID = Medicine.Medicine_ID;
```

Result for Query 09:

Prescription_ID	Date	Patient_ID	Patient_FName	Patient_LName	Medicine_ID	M_Name	Dosage
13	2023-12-05	1	John	Doe	1	Aspirin	2
213	2023-12-25	36	Liam	Long	1	Aspirin	2
403	2024-01-13	32	William	Reed	1	Aspirin	1
123	2023-12-16	20	Chloe	Baker	2	Ibuprofen	2
323	2024-01-05	2	Jane	Smith	2	Ibuprofen	1
23	2023-12-06	2	Jane	Smith	3	Acetaminophen	1

Query 10: Calculate the total cost of prescriptions for each patient.

```
SELECT Patient.Patient_ID, Patient.Patient_FName, Patient.Patient_LName,
SUM(Medicine.M_Cost * Prescription.Dosage) AS Total_Prescription_Cost

FROM Patient

JOIN Prescription ON Patient.Patient_ID = Prescription.Patient_ID

JOIN Medicine ON Prescription.Medicine_ID = Medicine.Medicine_ID

GROUP BY Patient.Patient_ID, Patient.Patient_FName, Patient.Patient_LName;
```

Result for Query 10:

Patient_ID	Patient_FName	Patient_LName	Total_Prescription_Cost
1	John	Doe	56.45
36	Liam	Long	11.98
32	William	Reed	15.74
20	Chloe	Baker	16.98
2	Jane	Smith	33.24
45	Lily	Ross	6.25

Query 11: Retrieve the information of the patients who have an outstanding bill balance to notify them.

```
SELECT distinct pt.Patient_ID, pt.Patient_FName, pt.Patient_LName, pt.phone,
pt.email, bill.Remaining_Balance

FROM patient pt JOIN appointment apt ON pt.patient_id = apt.Patient_ID

JOIN bill ON bill.Patient_ID = pt.patient_id

WHERE bill.Remaining_Balance > 0;
```

Results for Query 11:

Patient_ID	Patient_FName	Patient_LName	phone	email	Remaining_Balance
1	John	Doe	555-1234	john.doe@email.com	150.00
2	Jane	Smith	555-5678	jane.smith@email.com	200.00
3	Michael	Johnson	555-8765	michael.johnson@email.com	175.00
4	Emily	Williams	555-2345	emily.williams@email.com	180.00
5	Robert	Brown	555-5432	robert.brown@email.com	190.00
6	Alice	Davis	555-7890	alice.davis@email.com	220.00
7	Christopher	Miller	555-4321	chris.miller@email.com	160.00
8	Olivia	Jones	555-9876	olivia.jones@email.com	200.00
9	William	Wilson	555-8765	william.wilson@email.com	180.00
10	Sophia	Moore	555-6543	sophia.moore@email.com	195.00
11	David	Taylor	555-3210	david.taylor@email.com	150.00

Query 12: Retrieve the phone number of the male patients who are O+ and are free from allergies for future blood donation.

```
SELECT pt.Patient_ID, pt.Patient_FName, pt.Patient_LName, pt.phone
FROM patient pt
JOIN medical_history mh ON pt.Patient_ID = mh.Patient_ID
WHERE Gender ='Male'
AND allergies = 'None'
AND Blood_Type = 'O+';
```

Result for Query 12:

Patient_ID	Patient_FName	Patient_LName	phone
26	Jack	Harrison	555-9876
34	Mason	Lopez	555-5432
42	Lucas	Fisher	555-8765
46	Grayson	Harrison	555-5432

3. Conclusion

In this report we describe the implementation process of our updated hospital management system. This includes steps for setting up the database, inserting dummy values into the database, and creating scenarios for data retrieval.