Hospital Management System

Name-Omkar Kshirsagar

In this project I have used mysql for creating database, tables and for executing the queries

Query1: - HMS should be capable to recognize already registered patients and user roles.

Patient Table-

```
INSERT INTO PatientDetails (PatientID, FirstName, LastName, DateofBirth, Gender, ContactNumber, Email, Address)
VALUES
(101, 'Aarav', 'Gupta', '1990-05-15', 'M', '9876543210', 'aarav.gupta@example.com', '123 Main Street, pune, maharashtra, Pincode'),
(102, 'Aisha', 'Sharma', '1985-08-22', 'F', '8765432109', 'aisha.sharma@example.com', '456 Park Avenue, mumbai, maharashtra, Pincode'),
(103, 'Aditya', 'Patel', '1992-02-10', 'M', '7654321098', 'aditya.patel@example.com', '789 Oak Road, nasik, maharashtra, Pincode'),
(104, 'Ananya', 'Verma', '1988-11-30', 'F', '6543210987', 'ananya.verma@example.com', '567 Pine Lane, pune, maharashtra, Pincode'),
(105, 'Arjun', 'Singh', '1995-07-18', 'M', '5432109876', 'arjun.singh@example.com', '890 Cedar Street, mumbai, maharashtra, Pincode'),
(106, 'Neha', 'Gupta', '1991-04-25', 'F', '4321098765', 'neha.gupta@example.com', '234 Birch Avenue, pune, maharashtra, Pincode'),
(107, 'Vivek', 'Yadav', '1987-09-03', 'M', '3210987654', 'vivek.yadav@example.com', '901 Maple Drive, nasik, maharashtra, Pincode'),
(108, 'Riya', 'Chopra', '1994-12-12', 'F', '2109876543', 'riya.chopra@example.com', '678 Elm Street, pune, maharashtra, Pincode'),
(109, 'Rahul', 'Mishra', '1986-06-20', 'M', '1098765432', 'rahul.mishra@example.com', '345 Walnut Lane, pune, maharashtra, Pincode'),
(110, 'Aarti', 'Shukla', '1993-03-08', 'F', '9876543210', 'aarti.shukla@example.com', '123 Pinecrest Avenue, mumbai, maharashtra,  Pincode');
```

If I tried to insert already present record then,

```
CALL InsertPatientRecord2(101, 'Aarav', 'Gupta', '1990-05-15', 'M', '9876543210', 'aarav.gupta@example.com', '123 Main Street, pune, maharashtra, Pincode');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Message
Error: Duplicate record. This patient is already in the records.

Same for User Staff record table

```sql
INSERT INTO StaffDetails (StaffID, FirstName, LastName, StaffPosition, ContactNumber, Email, DepartmentID)
VALUES
(1, 'Amit', 'Sharma', 'Nurse', '9876543210', 'amit.sharma@example.com', 1),
(2, 'Priya', 'Verma', 'Doctor', '8765432109', 'priya.verma@example.com', 2),
(3, 'Rajesh', 'Yadav', 'Receptionist', '7654321098', 'rajesh.yadav@example.com', 3),
(4, 'Neha', 'Singh', 'Technician', '6543210987', 'neha.singh@example.com', 4),
(5, 'Vikram', 'Chopra', 'Administrator', '5432109876', 'vikram.chopra@example.com', 5),
(6, 'Anjali', 'Mishra', 'Nurse', '4321098765', 'anjali.mishra@example.com', 1),
(7, 'Rahul', 'Kumar', 'Doctor', '3210987654', 'rahul.kumar@example.com', 2),
(8, 'Sunita', 'Gupta', 'Receptionist', '2109876543', 'sunita.gupta@example.com', 3),
(9, 'Amit', 'Verma', 'Technician', '1098765432', 'amit.verma@example.com', 4),
(10, 'Sneha', 'Patel', 'Administrator', '9876543210', 'sneha.patel@example.com', 5);
```

```sql
CALL InsertStaffRecord(1, 'Amit', 'Sharma', 'Nurse', '9876543210', 'amit.sharma@example.com', 1);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Message
Error: Duplicate record. This staff member is already in the records.

I used stored procedure to find and display error message if record is already present in table and If not it will add record to table and query I used is

```
    -- You can add additional handling or simply retur
END IF;

-- Check for different PatientID with the same details
IF EXISTS (
    SELECT 1
    FROM PatientDetails
    WHERE PatientID <> p_PatientID
      AND (FirstName = p_FirstName
           AND LastName = p_LastName
           AND ContactNumber = p_ContactNumber
           AND Email = p_Email
           AND Address = p_Address)
) THEN
    SELECT 'Error: PatientID is different, but details
END IF;

) //
.IMITER ;
```

```
DELIMITER //
CREATE PROCEDURE InsertPatientRecord2 (
    IN p_PatientID INT,
    IN p_FirstName VARCHAR(50),
    IN p_LastName VARCHAR(50),
    IN p_DateOfBirth DATE,
    IN p_Gender CHAR(1),
    IN p_ContactNumber VARCHAR(20),
    IN p_Email VARCHAR(50),
    IN p_Address VARCHAR(100)
)
BEGIN
    DECLARE duplicate_error BOOLEAN DEFAULT FALSE;

    -- Declared a handler for the duplicate key error
    DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'
        SET duplicate_error = TRUE;
    BEGIN

        INSERT INTO PatientDetails (PatientID, FirstName, LastName, DateofBirth, Gender, ContactNumber, Email, Address)
        VALUES (p_PatientID, p_FirstName, p_LastName, p_DateOfBirth, p_Gender, p_ContactNumber, p_Email, p_Address);

        SELECT 'Record inserted successfully.' AS Message;
    END;
    IF duplicate_error THEN
        SELECT 'Error: Duplicate record. This patient is already in the records.' AS Message;
        SET duplicate_error = FALSE; -- Reset the flag to allow further processing

    END IF;
    IF EXISTS (
        SELECT 1
        FROM PatientDetails
        WHERE PatientID <> p_PatientID
          AND (FirstName = p_FirstName
               AND LastName = p_LastName
               AND ContactNumber = p_ContactNumber
               AND Email = p_Email
               AND Address = p_Address)
    ) THEN
        SELECT 'Error: PatientID is different, but details are already associated with another record.' AS Message;
    END IF;

END //
DELIMITER ;
```

Query2: - Write necessary queries to register new user roles and personas

```sql
INSERT INTO StaffDetails (StaffID, FirstName, LastName, StaffPosition, ContactNumber, Email, DepartmentID)
VALUES
(1, 'Amit', 'Sharma', 'Nurse', '9876543210', 'amit.sharma@example.com', 1),
(2, 'Priya', 'Verma', 'Doctor', '8765432109', 'priya.verma@example.com', 2),
(3, 'Rajesh', 'Yadav', 'Receptionist', '7654321098', 'rajesh.yadav@example.com', 3),
(4, 'Neha', 'Singh', 'Technician', '6543210987', 'neha.singh@example.com', 4),
(5, 'Vikram', 'Chopra', 'Administrator', '5432109876', 'vikram.chopra@example.com', 5),
(6, 'Anjali', 'Mishra', 'Nurse', '4321098765', 'anjali.mishra@example.com', 1),
(7, 'Rahul', 'Kumar', 'Doctor', '3210987654', 'rahul.kumar@example.com', 2),
(8, 'Sunita', 'Gupta', 'Receptionist', '2109876543', 'sunita.gupta@example.com', 3),
(9, 'Amit', 'Verma', 'Technician', '1098765432', 'amit.verma@example.com', 4),
(10, 'Sneha', 'Patel', 'Administrator', '9876543210', 'sneha.patel@example.com', 5);
```

Here DepartmentID is ForeginKey from Department Table and inserted 10 records, there roles and personas successfully

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

StaffID	FirstName	LastName	StaffPosition	ContactNumber	Email	DepartmentID
1	Amit	Sharma	Nurse	9876543210	amit.sharma@example.com	1
2	Priya	Verma	Doctor	8765432109	priya.verma@example.com	2
3	Rajesh	Yadav	Receptionist	7654321098	rajesh.yadav@example.com	3
4	Neha	Singh	Technician	6543210987	neha.singh@example.com	4
5	Vikram	Chopra	Administrator	5432109876	vikram.chopra@example.com	5
6	Anjali	Mishra	Nurse	4321098765	anjali.mishra@example.com	1
7	Rahul	Kumar	Doctor	3210987654	rahul.kumar@example.com	2
8	Sunita	Gupta	Receptionist	2109876543	sunita.gupta@example.com	3
9	Amit	Verma	Technician	1098765432	amit.verma@example.com	4
10	Sneha	Patel	Administrator	9876543210	sneha.patel@example.com	5

Query3: - Write necessary queries to add to the list of diagnosis of the patient tagged by date.

These are 10 records which added to PatientVisitDetails table along with diagnosis of the patient tagged by date. Where visitID, PatientID and DoctorID is foreginKeys

```
INSERT INTO PatientVisitDetails (VisitID, PatientID, DoctorID, VisitDateTime, Diagnosis, Prescription)
VALUES
(1, 101, 11, '2024-02-10 10:30:00', 'Fever', 'Paracetamol'),
(2, 102, 12, '2024-02-15 11:45:00', 'Cough', 'Cough syrup'),
(3, 103, 13, '2024-02-20 14:00:00', 'Sprained ankle', 'Rest and pain relievers'),
(4, 104, 14, '2024-02-25 16:30:00', 'Skin rash', 'Topical cream'),
(5, 105, 15, '2024-03-01 09:45:00', 'Eye infection', 'Antibiotic eye drops'),
(6, 106, 16, '2024-03-05 12:15:00', 'Headache', 'Painkillers'),
(7, 107, 17, '2024-03-10 15:30:00', 'Digestive issues', 'Antacids'),
(8, 108, 18, '2024-03-15 18:45:00', 'Thyroid disorder', 'Medication'),
(9, 109, 19, '2024-03-20 09:15:00', 'Kidney stones', 'Fluids and pain relief'),
(10, 110, 20, '2024-03-25 11:30:00', 'Respiratory infection', 'Antibiotics');
```

The Records successfully fetched from patientdetails table along with visit date time, diagnosis of patient prescription and id by which doctor it was handled and also patientid to recognize unique patient

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

VisitID	PatientID	DoctorID	VisitDateTime	Diagnosis	Prescription
1	101	11	2024-02-10 10:30:00	Fever	Paracetamol
2	102	12	2024-02-15 11:45:00	Cough	Cough syrup
3	103	13	2024-02-20 14:00:00	Sprained ankle	Rest and pain relievers
4	104	14	2024-02-25 16:30:00	Skin rash	Topical cream
5	105	15	2024-03-01 09:45:00	Eye infection	Antibiotic eye drops
6	106	16	2024-03-05 12:15:00	Headache	Painkillers
7	107	17	2024-03-10 15:30:00	Digestive issues	Antacids
8	108	18	2024-03-15 18:45:00	Thyroid disorder	Medication
9	109	19	2024-03-20 09:15:00	Kidney stones	Fluids and pain relief
10	110	20	2024-03-25 11:30:00	Respiratory infection	Antibiotics
NULL	NULL	NULL	NULL	NULL	NULL

Query4: - Write necessary queries to fetch required details of a particular patient.

By required data of patient fetched from tables using join query all these three screen shots are from one join query

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	PatientFirstName	PatientLastName	DateofBirth	Gender	PatientContactNumber	PatientEmail	PatientAddress	DoctorID
101	Aarav	Gupta	1990-05-15	M	9876543210	aarav.gupta@example.com	123 Main Street, pune, maharashtra, Pincode	11
102	Aisha	Sharma	1985-08-22	F	8765432109	aisha.sharma@example.com	456 Park Avenue, mumbai, maharashtra, Pincode	12
103	Aditya	Patel	1992-02-10	M	7654321098	aditya.patel@example.com	789 Oak Road, nasik, maharashtra, Pincode	13
104	Ananya	Verma	1988-11-30	F	6543210987	ananya.verma@example.com	567 Pine Lane, pune, maharashtra, Pincode	14
105	Arjun	Singh	1995-07-18	M	5432109876	arjun.singh@example.com	890 Cedar Street, mumbai, maharashtra, Pincode	15
106	Neha	Gupta	1991-04-25	F	4321098765	neha.gupta@example.com	234 Birch Avenue, pune, maharashtra, Pincode	16
107	Vivek	Yadav	1987-09-03	M	3210987654	vivek.yadav@example.com	901 Maple Drive, nasik, maharashtra, Pincode	17
108	Riya	Chopra	1994-12-12	F	2109876543	riya.chopra@example.com	678 Elm Street, pune, maharashtra, Pincode	18
109	Rahul	Mishra	1986-06-20	M	1098765432	rahul.mishra@example.com	345 Walnut Lane, pune, maharashtra, Pincode	19
110	Aarti	Shukla	1993-03-08	F	9876543210	aarti.shukla@example.com	123 Pinecrest Avenue, mumbai, maharashtra, ...	20

DoctorFirstName	DoctorLastName	specialization	DoctorContactNumber	DoctorEmail	AppointmentID	AppointmentDateTime	AppointmentStatus
Rajesh	Kumar	Cardiology	123-456-7890	rajesh.kumar@example.com	1	2024-02-10 10:00:00	Scheduled
Priya	Sharma	Pediatrics	987-654-3210	priya.sharma@example.com	2	2024-02-15 11:30:00	Completed
Amit	Patel	Orthopedics	456-789-0123	amit.patel@example.com	3	2024-02-20 14:45:00	Scheduled
Anjali	Verma	Dermatology	789-012-3456	anjali.verma@example.com	4	2024-02-25 16:00:00	Cancelled
Ravi	Singh	Ophthalmology	012-345-6789	ravi.singh@example.com	5	2024-03-01 09:15:00	Scheduled
Neha	Gupta	Neurology	234-567-8901	neha.gupta@example.com	6	2024-03-05 12:30:00	Completed
Vikram	Yadav	Gastroenterology	567-890-1234	vikram.yadav@example.com	7	2024-03-10 15:45:00	Scheduled
Sunita	Chopra	Endocrinology	890-123-4567	sunita.chopra@example.com	8	2024-03-15 18:00:00	Scheduled
Rahul	Mishra	Urology	345-678-9012	rahul.mishra@example.com	9	2024-03-20 09:30:00	Scheduled
Aarti	Shukla	Pulmonology	678-901-2345	aarti.shukla@example.com	10	2024-03-25 11:45:00	Scheduled

DoctorEmail	AppointmentID	AppointmentDateTime	AppointmentStatus	TestID	TestName	TestDate	TestResult
rajesh.kumar@example.com	1	2024-02-10 10:00:00	Scheduled	1	Blood Test	2024-01-10	Normal
priy priya.sharma@example.com		2024-02-15 11:30:00	Completed	2	X-ray	2024-01-15	No abnormalities detected
amit.patel@example.com	3	2024-02-20 14:45:00	Scheduled	3	MRI Scan	2024-01-20	Minor injury detected
anjali.verma@example.com	4	2024-02-25 16:00:00	Cancelled	4	Skin Biopsy	2024-01-25	Inconclusive
ravi.singh@example.com	5	2024-03-01 09:15:00	Scheduled	5	Eye Exam	2024-02-01	Prescription needed
neha.gupta@example.com	6	2024-03-05 12:30:00	Completed	6	EEG	2024-02-05	Normal brain activity
vikram.yadav@example.com	7	2024-03-10 15:45:00	Scheduled	7	Endoscopy	2024-02-10	Digestive issues detected
sunita.chopra@example.com	8	2024-03-15 18:00:00	Scheduled	8	Thyroid Panel	2024-02-15	Abnormal thyroid levels
rahul.mishra@example.com	9	2024-03-20 09:30:00	Scheduled	9	Kidney Function Test	2024-02-20	Normal
aarti.shukla@example.com	10	2024-03-25 11:45:00	Scheduled	10	Pulmonary Function Test	2024-02-25	Normal lung function

The query through all this data I fetched

```
SELECT

    p.PatientID,
```

```
    p.FirstName AS PatientFirstName,
    p.LastName AS PatientLastName,
    p.DateofBirth,
    p.Gender,
    p.ContactNumber AS PatientContactNumber,
    p.Email AS PatientEmail,
    p.Address AS PatientAddress,
    d.DoctorID,
    d.FirstName AS DoctorFirstName,
    d.LastName AS DoctorLastName,
    d.specialization,
    d.ContactNumber AS DoctorContactNumber,
    d.Email AS DoctorEmail,
    a.AppointmentID,
    a.AppointmentDateTime,
    a.Status AS AppointmentStatus,
    lt.TestID,
    lt.TestName,
    lt.TestDate,
    lt.TestResult
FROM
    PatientDetails p
JOIN
            Appointment a ON p.PatientID = a.PatientID
JOIN
    DoctorDetails d ON a.DoctorID = d.DoctorID
LEFT JOIN
    LabTest lt ON p.PatientID = lt.PatientID AND d.DoctorID = lt.DoctorID;
```

//for fetching particular patient record we can get records

by adding where clause

// p.PatientID = 101;-> through this it will return first record

And if want to fetch particular patient record then

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	PatientFirstName	PatientLastName	DateofBirth	Gender	PatientContactNumber	PatientEmail	PatientAddress	DoctorID
101	Aarav	Gupta	1990-05-15	M	9876543210	aarav.gupta@example.com	123 Main Street, pune, maharashtra, Pincode	11

DoctorFirstName	DoctorLastName	specialization	DoctorContactNumber	DoctorEmail	AppointmentID	AppointmentDateTime	AppointmentStatus	TestID	TestName	TestDate	TestResult
Rajesh	Kumar	Cardiology	123-456-7890	rajesh.kumar@example.com	1	2024-02-10 10:00:00	Scheduled	1	Blood Test	2024-01-10	Normal

Query5: - Write necessary queries to prepare bill for the patient at the end of checkout.

This query fetches the data from patientdetails table, labtest table and patient billing table

```
SELECT
    pb.BillID,
    pb.PatientID,
    pd.FirstName AS PatientFirstName,
    pd.LastName AS PatientLastName,
    pd.DateofBirth,
    pv.VisitID,
    pv.VisitDateTime,
    pv.Diagnosis,
    pv.Prescription,
    lt.TestID,
    lt.TestName,
    lt.TestDate,
    lt.TestResult,
    pb.TotalAmount,
    pb.PaymentStatus
FROM
    PatientBilling pb
JOIN
    PatientDetails pd ON pb.PatientID = pd.PatientID
JOIN
    PatientVisitDetails pv ON pb.VisitID = pv.VisitID
LEFT JOIN
    LabTest lt ON pd.PatientID = lt.PatientID AND pv.DoctorID = lt.DoctorID
WHERE
    pb.PatientID = 101;
```

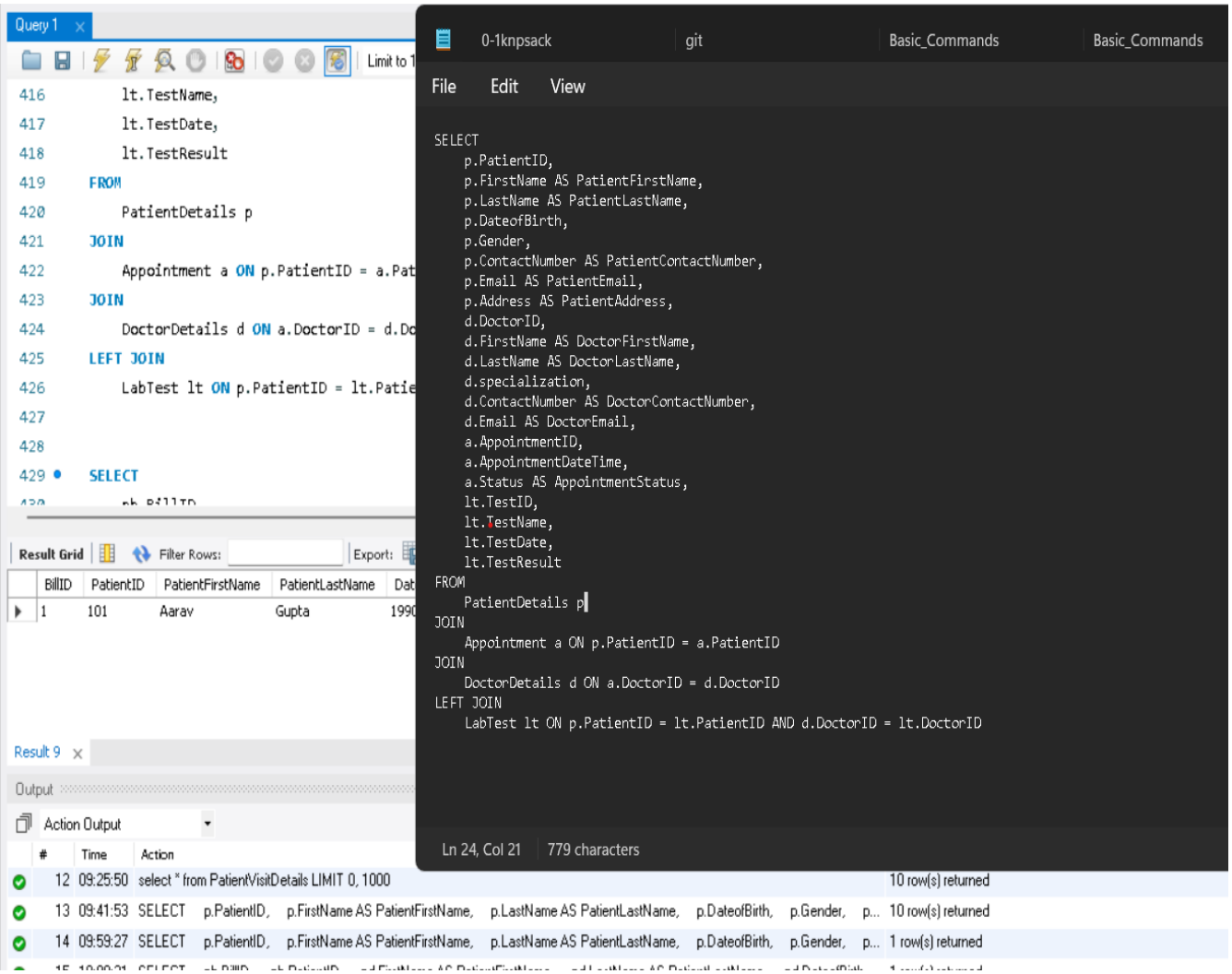
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BillID	PatientID	PatientFirstName	PatientLastName	DateofBirth	VisitID	VisitDateTime	Diagnosis
1	101	Aarav	Gupta	1990-05-15	1	2024-02-10 10:30:00	Fever

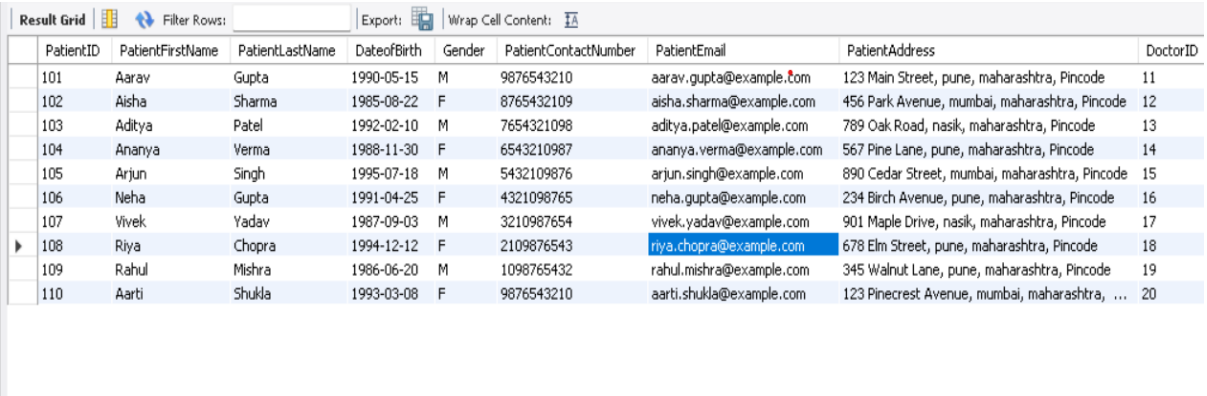
Prescription	TestID	TestName	TestDate	TestResult	TotalAmount	PaymentStatus
Paracetamol	1	Blood Test	2024-01-10	Normal	1500.00 INR	Paid

Query6: - Write necessary queries to fetch and show data from various related tables (Joins)

1.Query to show all required details of patient using left join



Outputs of left join query:-



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	PatientFirstName	PatientLastName	DateofBirth	Gender	PatientContactNumber	PatientEmail	PatientAddress	DoctorID
101	Aarav	Gupta	1990-05-15	M	9876543210	aarav.gupta@example.com	123 Main Street, pune, maharashtra, Pincode	11
102	Aisha	Sharma	1985-08-22	F	8765432109	aisha.sharma@example.com	456 Park Avenue, mumbai, maharashtra, Pincode	12
103	Aditya	Patel	1992-02-10	M	7654321098	aditya.patel@example.com	789 Oak Road, nasik, maharashtra, Pincode	13
104	Ananya	Verma	1988-11-30	F	6543210987	ananya.verma@example.com	567 Pine Lane, pune, maharashtra, Pincode	14
105	Arjun	Singh	1995-07-18	M	5432109876	arjun.singh@example.com	890 Cedar Street, mumbai, maharashtra, Pincode	15
106	Neha	Gupta	1991-04-25	F	4321098765	neha.gupta@example.com	234 Birch Avenue, pune, maharashtra, Pincode	16
107	Vivek	Yadav	1987-09-03	M	3210987654	vivek.yadav@example.com	901 Maple Drive, nasik, maharashtra, Pincode	17
108	Riya	Chopra	1994-12-12	F	2109876543	riya.chopra@example.com	678 Elm Street, pune, maharashtra, Pincode	18
109	Rahul	Mishra	1986-06-20	M	1098765432	rahul.mishra@example.com	345 Walnut Lane, pune, maharashtra, Pincode	19
110	Aarti	Shukla	1993-03-08	F	9876543210	aarti.shukla@example.com	123 Pinecrest Avenue, mumbai, maharashtra, ...	20

DoctorFirstName	DoctorLastName	specialization	DoctorContactNumber	DoctorEmail	AppointmentID	AppointmentDateTime	AppointmentStatus
Rajesh	Kumar	Cardiology	123-456-7890	rajesh.kumar@example.com	1	2024-02-10 10:00:00	Scheduled
Priya	Sharma	Pediatrics	987-654-3210	priya.sharma@example.com	2	2024-02-15 11:30:00	Completed
Amit	Patel	Orthopedics	456-789-0123	amit.patel@example.com	3	2024-02-20 14:45:00	Scheduled
Anjali	Verma	Dermatology	789-012-3456	anjali.verma@example.com	4	2024-02-25 16:00:00	Cancelled
Ravi	Singh	Ophthalmology	012-345-6789	ravi.singh@example.com	5	2024-03-01 09:15:00	Scheduled
Neha	Gupta	Neurology	234-567-8901	neha.gupta@example.com	6	2024-03-05 12:30:00	Completed
Vikram	Yadav	Gastroenterology	567-890-1234	vikram.yadav@example.com	7	2024-03-10 15:45:00	Scheduled
Sunita	Chopra	Endocrinology	890-123-4567	sunita.chopra@example.com	8	2024-03-15 18:00:00	Scheduled
Rahul	Mishra	Urology	345-678-9012	rahul.mishra@example.com	9	2024-03-20 09:30:00	Scheduled
Aarti	Shukla	Pulmonology	678-901-2345	aarti.shukla@example.com	10	2024-03-25 11:45:00	Scheduled

DoctorEmail	AppointmentID	AppointmentDateTime	AppointmentStatus	TestID	TestName	TestDate	TestResult
rajesh.kumar@example.com	1	2024-02-10 10:00:00	Scheduled	1	Blood Test	2024-01-10	Normal
priy priya.sharma@example.com		2024-02-15 11:30:00	Completed	2	X-ray	2024-01-15	No abnormalities detected
amit.patel@example.com	3	2024-02-20 14:45:00	Scheduled	3	MRI Scan	2024-01-20	Minor injury detected
anjali.verma@example.com	4	2024-02-25 16:00:00	Cancelled	4	Skin Biopsy	2024-01-25	Inconclusive
ravi.singh@example.com	5	2024-03-01 09:15:00	Scheduled	5	Eye Exam	2024-02-01	Prescription needed
neha.gupta@example.com	6	2024-03-05 12:30:00	Completed	6	EEG	2024-02-05	Normal brain activity
vikram.yadav@example.com	7	2024-03-10 15:45:00	Scheduled	7	Endoscopy	2024-02-10	Digestive issues detected
sunita.chopra@example.com	8	2024-03-15 18:00:00	Scheduled	8	Thyroid Panel	2024-02-15	Abnormal thyroid levels
rahul.mishra@example.com	9	2024-03-20 09:30:00	Scheduled	9	Kidney Function Test	2024-02-20	Normal
aarti.shukla@example.com	10	2024-03-25 11:45:00	Scheduled	10	Pulmonary Function Test	2024-02-25	Normal lung function

2. query performs right join operation, selects columns from both tables and includes all records from PatientDetails and the matching records from Appointment table

```
SELECT
    A.AppointmentID,
    A.PatientID,
    A.DoctorID,
    A.AppointmentDateTime,
    A.Status,
    P.FirstName,
    P.LastName,
    P.DateofBirth,
    P.Gender,
    P.ContactNumber,
    P.Email,
    P.Address
FROM
    Appointment A
RIGHT JOIN
    PatientDetails P ON A.PatientID = P.PatientID;
```

Outputs of this query

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AppointmentID	PatientID	DoctorID	AppointmentDateTime	Status	FirstName	LastName	DateofBirth	Gender
1	101	11	2024-02-10 10:00:00	Scheduled	Aarav	Gupta	1990-05-15	M
2	102	12	2024-02-15 11:30:00	Completed	Aisha	Sharma	1985-08-22	F
3	103	13	2024-02-20 14:45:00	Scheduled	Aditya	Patel	1992-02-10	M
4	104	14	2024-02-25 16:00:00	Cancelled	Ananya	Verma	1988-11-30	F
5	105	15	2024-03-01 09:15:00	Scheduled	Arjun	Singh	1995-07-18	M
6	106	16	2024-03-05 12:30:00	Completed	Neha	Gupta	1991-04-25	F
7	107	17	2024-03-10 15:45:00	Scheduled	Vivek	Yadav	1987-09-03	M
8	108	18	2024-03-15 18:00:00	Scheduled	Riya	Chopra	1994-12-12	F
9	109	19	2024-03-20 09:30:00	Scheduled	Rahul	Mishra	1986-06-20	M
10	110	20	2024-03-25 11:45:00	Scheduled	Aarti	Shukla	1993-03-08	F
NULL	NULL	NULL	NULL	NULL	Omkar	Kshirsagar	2002-07-23	M

ContactNumber	Email	Address
9876543210	aarav.gupta@example.com	123 Main Street, pune, maharashtra, Pincode
8765432109	aisha.sharma@example.com	456 Park Avenue, mumbai, maharashtra, Pincode
7654321098	aditya.patel@example.com	789 Oak Road, nasik, maharashtra, Pincode
6543210987	ananya.verma@example.com	567 Pine Lane, pune, maharashtra, Pincode
5432109876	arjun.singh@example.com	890 Cedar Street, mumbai, maharashtra, Pincode
4321098765	neha.gupta@example.co neha.gupta@example.com	, maharashtra, Pincode
3210987654	vivek.yadav@example.com	901 Maple Drive, nasik, maharashtra, Pincode
2109876543	riya.chopra@example.com	678 Elm Street, pune, maharashtra, Pincode
1098765432	rahul.mishra@example.com	345 Walnut Lane, pune, maharashtra, Pincode
9876543210	aarti.shukla@example.com	123 Pinecrest Avenue, mumbai, maharashtra, ...
9876543221	omkar.kshirsagarexample.com	123 suyog Street, Pune, Maharashtra, Pincode

Query7: - - Optimize repeated read operations using views/materialized views.

-Using view, the regular operation of seeing patient details from patientDetails table along with their billdata from patientbilling table I have optimized that regular task by creating a view through which it joins both table and provide sufficient information

-This query creates a view named 'PatientDetailsWithBilling' in which it combines billing table and patientdetails table and perform join operation

```
CREATE VIEW PatientDetailsWithBilling AS
SELECT
    pd.PatientID,
    pd.FirstName,
    pd.LastName,
    pd.DateofBirth,
    pd.Gender,
    pd.ContactNumber,
    pd.Email,
    pd.Address,
    pb.BillID,
    pb.TotalAmount,
    pb.PaymentStatus
FROM
    PatientDetails pd
JOIN
    PatientBilling pb ON pd.PatientID = pb.PatientID;
```

Output-

```
select * from PatientDetailsWithBilling;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PatientID	FirstName	LastName	DateofBirth	Gender	ContactNumber	Email	Address	BillID	TotalAmount	PaymentStatus
101	Aarav	Gupta	1990-05-15	M	9876543210	aarav.gupta@example.com	123 Main Street, pune, maharashtra, Pincode	1	1500.00 INR	Paid
102	Aisha	Sharma	1985-08-22	F	8765432109	aisha.sharma@example.com	456 Park Avenue, mumbai, maharashtra, Pincode	2	2000.00 INR	Unpaid
103	Aditya	Patel	1992-02-10	M	7654321098	aditya.patel@example.com	789 Oak Road, nasik, maharashtra, Pincode	3	1800.50 INR	Paid
104	Ananya	Verma	1988-11-30	F	6543210987	ananya.verma@example.com	567 Pine Lane, pune, maharashtra, Pincode	4	2500.75 INR	Unpaid
105	Arjun	Singh	1995-07-18	M	5432109876	arjun.singh@example.com	890 Cedar Street, mumbai, maharashtra, Pincode	5	1200.25 INR	Paid
106	Neha	Gupta	1991-04-25	F	4321098765	neha.gupta@example.com	234 Birch Avenue, pune, maharashtra, Pincode	6	3000.00 INR	Unpaid
107	Vivek	Yadav	1987-09-03	M	3210987654	vivek.yadav@example.com	901 Maple Drive, nasik, maharashtra, Pincode	7	1600.75 INR	Paid
108	Riya	Chopra	1994-12-12	F	2109876543	riya.chopra@example.com	678 Elm Street, pun 901 Maple Drive, nasik, maharashtra, Pincode			Unpaid
109	Rahul	Mishra	1986-06-20	M	1098765432	rahul.mishra@example.com	345 Walnut Lane, pune, maharashtra, Pincode	9	1900.25 INR	Paid
110	Aarti	Shukla	1993-03-08	F	9876543210	aarti.shukla@example.com	123 Pinecrest Avenue, mumbai, maharashtra, ...	10	2800.00 INR	Unpaid

Query8: - Optimize read operations using indexing wherever required. (Create index on at least 1 table)

-I have used indexing on frequently used columns from Doctordetials table and appointment table, firstly for foregin key as they are likely to used in joined operation.
-I have created index for doctor id from doctor details table as it will be the foregin key for appointment table through which read operation and data fetching operations can be optimized

output

```
    -- Adding indexes to foreign key columns
CREATE INDEX idx_PatientDetails_PatientID ON PatientDetails(PatientID);
CREATE INDEX idx_DoctorDetails_DoctorID ON DoctorDetails(DoctorID);

CREATE INDEX idx_Appointment_PatientID ON Appointment(PatientID);
CREATE INDEX idx_Appointment_DoctorID ON Appointment(DoctorID);

-- Adding index to AppointmentDateTime for optimizing search conditions
CREATE INDEX idx_Appointment_AppointmentDateTime ON Appointment(AppointmentDateTime);
```

Output

Action Output

#	Time	Action	Message
19	11:17:43	CREATE INDEX idx_PatientDetails_PatientID ON PatientDetails(PatientID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
20	11:17:49	CREATE INDEX idx_DoctorDetails_DoctorID ON DoctorDetails(DoctorID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
21	11:17:55	CREATE INDEX idx_Appointment_PatientID ON Appointment(PatientID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
22	11:18:08	CREATE INDEX idx_Appointment_DoctorID ON Appointment(DoctorID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
23	11:18:13	CREATE INDEX idx_Appointment_AppointmentDateTime ON Appointment(AppointmentDateTime)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-Indexes are added to the foreign key columns PatientID and DoctorID in the Appointment table, as they are used in join operations.

- index is added to the AppointmentDateTime column in the Appointment table to optimize search conditions based on appointment date and time.

Query9: - Try optimizing bill generation using stored procedures.

-I have used visitID from patientvisittable as foregin key in patientbilling table as the input for stored procedure through which it updates the total bill and reflects in patientbilling table

-And stored procedure is

```
-- Created a stored procedure for optimizing bill generation
DELIMITER //

CREATE PROCEDURE GenerateBill(IN p_VisitID INT)
BEGIN
    DECLARE totalAmount DECIMAL(10, 2);

    -- Calculated total amount for services
    SELECT
        SUM(ServiceCost) INTO totalAmount
    FROM (
        SELECT
            CASE
                WHEN m.MedicationName = 'Painkillers' THEN 10.00
                WHEN m.MedicationName = 'Antibiotics' THEN 15.00
                ELSE 0.00
            END AS ServiceCost
        FROM
            MedicationForpatients m
        WHERE
            m.VisitID = p_VisitID
    ) AS Services;

    -- Update the PatientBilling table with the calculated total amount
    UPDATE PatientBilling
    SET TotalAmount = totalAmount
    WHERE VisitID = p_VisitID;

END //

DELIMITER ;
```

Output:-

```
-- //omkar kshirsagar
CALL GenerateBill(3);// -- it calls visitid and updates data in PatientBilling table
SELECT * FROM PatientBilling WHERE VisitID = 3;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

BillID	PatientID	VisitID	TotalAmount	PaymentStatus
3	103	3	0.00	Paid
NULL	NULL	NULL	NULL	NULL

Query10- Add necessary triggers to indicate when patients' medical insurance limit has expired.

-As I have not added facility of medical insaurnce in my database firstly I added column in my patient billing table by using alter command and then according to their visitID I have updated their insaurnce policy as 'valid' and expired

```
ALTER TABLE PatientBilling
ADD COLUMN MedicalInsuranceValidity varchar(50);

UPDATE PatientBilling
SET MedicalInsuranceValidity =
    CASE
        WHEN VisitID IN (1, 3, 5, 7, 9) THEN 'Valid'
        ELSE 'Expired'
    END;

    select * from PatientBilling;
```

Created a trigger to see if patient's medical insaurance is expired or not and id expired it will return insaurnce expired if not it will insert record in patientbilling table -

```
SET NEW.MedicalInsuranceValidity = 'Valid';

-- Set MedicalInsuranceValidity to 'Expired'
SET NEW.MedicalInsuranceValidity = 'Expired';
-- You can also signal an error if needed
-- SIGNAL SQLSTATE '45000'
-- SET MESSAGE_TEXT = 'Insurance limit exceeded.
IF;
```

Rows: | Export: | Wrap Cell Content:

Type	Null	Key	Default	Extra
int	NO	PRI	NULL	
int	YES	MUL	NULL	
int	YES	MUL	NULL	
varchar(255)	NO		NULL	
varchar(255)	NO		NULL	

```
CREATE TRIGGER BeforeInsertOrUpdatePatientBilling
BEFORE INSERT OR UPDATE ON PatientBilling
FOR EACH ROW
BEGIN
    DECLARE insuranceRemaining DECIMAL(10, 2);

    SELECT PatientDetails.InsuranceLimit - COALESCE(SUM(TotalAmount), 0) INTO insuranceRemaining
    FROM PatientDetails
    LEFT JOIN PatientBilling ON PatientDetails.PatientID = PatientBilling.PatientID
    WHERE PatientDetails.PatientID = NEW.PatientID;

    -- Checking if the insurance limit is sufficient
    IF insuranceRemaining >= NEW.TotalAmount THEN
        -- Allow the insertion or update
        SET NEW.MedicalInsuranceValidity = 'Valid';
    ELSE
        -- Set MedicalInsuranceValidity to 'Expired'
        SET NEW.MedicalInsuranceValidity = 'Expired';
        -- SIGNAL SQLSTATE '45000'
        -- SET MESSAGE_TEXT = 'Insurance expired cannot insert the record';
    END IF;
END //

DELIMITER ;
```

Output-

```
select * from PatientBilling;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

BillID	PatientID	VisitID	TotalAmount	PaymentStatus	MedicalInsuranceValidity
1	101	1	0.00	Paid	Valid
2	102	2	2000.00 INR	Unpaid	Expired
3	103	3	0.00	Paid	Valid
4	104	4	2500.75 INR	Unpaid	Expired
5	105	5	1200.25 INR	Paid	Valid
6	106	6	0.00	Unpaid	Expired
7	107	7	1600.75 INR	Paid	Valid
8	108	8	2100.50 INR	Unpaid	Expired
9	109	9	1900.25 INR	Paid	Valid
10	110	10	2800.00 INR	Unpaid	Expired
NULL	NULL	NULL	NULL	NULL	NULL

PatientBilling 18