ASSIGNMENT – 3

CSA0593 - DBMS

NAME : S.VARSHA

REG NO : 192311425

DATE : 15:11:2024

QUESTION:

1.Hospital Management System Database with Patient Care Managemen

t -Design tables for patients, doctors, appointments, medical records, and billing.

- Implement stored procedures for patient registration, appointment scheduling, and billing.

- Develop SQL queries for reporting patient medical history, doctor schedules, and billing summaries.

- Test using various failure scenarios

ANSWER:

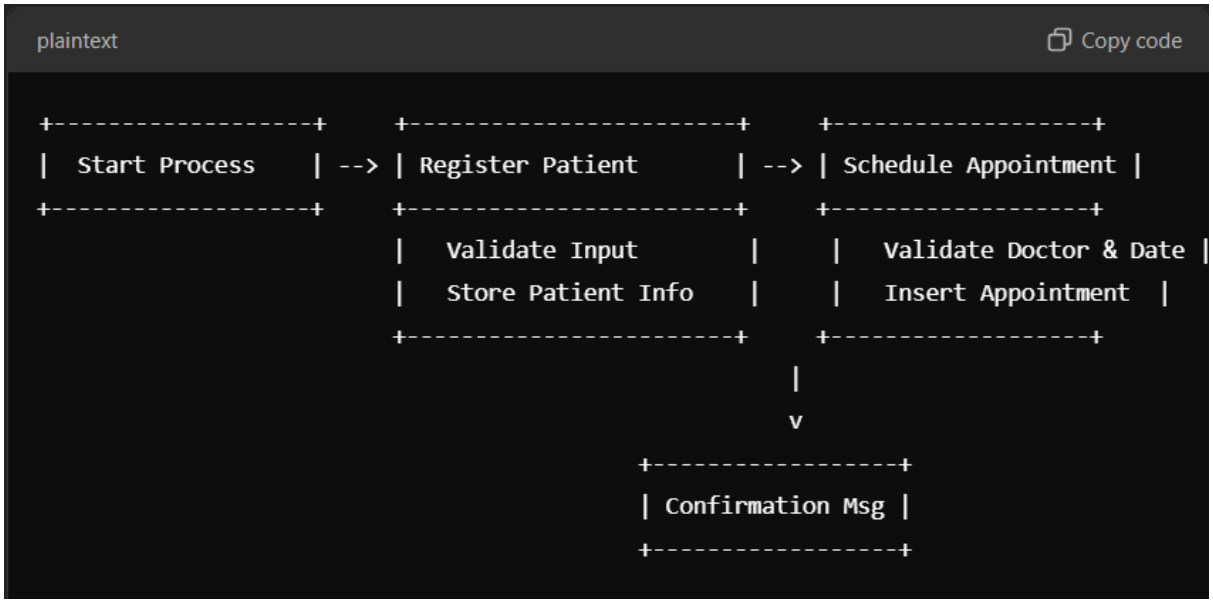
A Hospital Management System Database with Patient Care Management is designed to efficiently manage patient information, medical records, and treatment plans. The database consists of multiple tables, including Patients, Doctors, Nurses, Appointments, Medications, Lab Tests, and Medical Records. The Patients table stores demographic information, medical history, and contact details. The Doctors and Nurses tables store information about healthcare professionals, including their specialties, qualifications, and schedules.

The Appointments table manages scheduling and booking of appointments, while the Medications table tracks prescriptions, dosages, and administration schedules. The Lab Tests table stores results of laboratory tests and diagnostic procedures. The Medical

Records table stores comprehensive medical histories, including diagnoses, treatments, and outcomes. Relationships between tables are established using foreign keys, enabling efficient data retrieval and updating.

To ensure data security and integrity, the database can be designed with role-based access control, data encryption, and regular backups. By implementing a Hospital Management System Database with Patient Care Management, healthcare providers can improve patient care, streamline clinical workflows, and enhance operational efficiency.

FLOWCHART:



1.PATIENT TABLE:

Column Name	Data Type	Description
`patient_id`	INT	Primary key, Auto Increment
`first_name`	VARCHAR(100)	Patient's first name
`last_name`	VARCHAR(100)	Patient's last name
`gender`	ENUM('Male', 'Female', 'Other')	Patient's gender
`date_of_birth`	DATE	Patient's date of birth
`address`	VARCHAR(255)	Patient's address
`phone_number`	VARCHAR(15)	Patient's phone number
`email`	VARCHAR(100)	Patient's email address
`emergency_contact`	VARCHAR(100)	Emergency contact information
`date_registered`	TIMESTAMP	Date and time of patient registration

2.DOCTORS TABLE

Column Name	Data Type	Description
doctor_id	INT	Primary key, Auto Increment
first_name	VARCHAR(100)	Doctor's first name
last_name	VARCHAR(100)	Doctor's last name
specialty	VARCHAR(100)	Doctor's medical specialty
phone_number	VARCHAR(15)	Doctor's phone number
email	VARCHAR(100)	Doctor's email address
available_from	TIME	Doctor's available start time
available_to	TIME	Doctor's available end time

3.APPOINTMENTS TABLE

Column Name	Data Type	Description
appointment_id	INT	Primary key, Auto Increment
patient_id	INT	Foreign Key referencing patients
doctor_id	INT	Foreign Key referencing doctors
appointment_date	DATE	Date of the appointment
appointment_time	TIME	Time of the appointment
status	ENUM('Scheduled', 'Completed', 'Canceled')	Status of the appointment

4.MEDICAL RECORDS:

Column Name	Data Type	Description
`record_id`	INT	Primary key, Auto Increment
`patient_id`	INT	Foreign Key referencing `patients`
`doctor_id`	INT	Foreign Key referencing `doctors`
`diagnosis`	TEXT	Diagnosis provided by the doctor
`treatment`	TEXT	Treatment prescribed by the doctor
`record_date`	TIMESTAMP	Date and time the record was created

4.BILLING:

Column Name	Data Type	Description
`bill_id`	INT	Primary key, Auto Increment
`patient_id`	INT	Foreign Key referencing `patients`
`appointment_id`	INT	Foreign Key referencing `appointments`
`amount`	DECIMAL(10, 2)	Total amount billed
`bill_date`	TIMESTAMP	Date and time the bill was generated

Stored Procedures:

Patient Registration

```
DELIMITER $$

CREATE PROCEDURE register_patient(
    IN first_name VARCHAR(100),
    IN last_name VARCHAR(100),
    IN gender ENUM('Male', 'Female', 'Other'),
    IN dob DATE,
    IN address VARCHAR(255),
    IN phone_number VARCHAR(15),
    IN email VARCHAR(100),
    IN emergency_contact VARCHAR(100)
)
BEGIN
    INSERT INTO patients (first_name, last_name, gender, date_of_birth, address, phon
    VALUES (first_name, last_name, gender, dob, address, phone_number, email, emerger
END $$

DELIMITER ;
```

APPOINTMENT SHEDULING:

```
DELIMITER $$

CREATE PROCEDURE schedule_appointment(
    IN patient_id INT,
    IN doctor_id INT,
    IN appointment_date DATE,
    IN appointment_time TIME
)
BEGIN
    INSERT INTO appointments (patient_id, doctor_id, appointment_date, appointment_tim
    VALUES (patient_id, doctor_id, appointment_date, appointment_time);
END $$

DELIMITER ;
```

BILLING:

```
DELIMITER $$

CREATE PROCEDURE generate_bill(
    IN patient_id INT,
    IN appointment_id INT,
    IN amount DECIMAL(10, 2)
)
BEGIN
    INSERT INTO billing (patient_id, appointment_id, amount)
    VALUES (patient_id, appointment_id, amount);
END $$

DELIMITER ;
```

SQL Queries for Reporting:

Patient's Medical History

```
SELECT m.record_id, m.diagnosis, m.treatment, m.record_date, d.first_name AS doctor_fi
FROM medical_records m
JOIN doctors d ON m.doctor_id = d.doctor_id
WHERE m.patient_id = ?;
```

Doctor's Schedule

```
SELECT doctor_id, first_name, last_name, available_from, available_to
FROM doctors
WHERE doctor_id = ?;
```

Billing Summary for a Patient

```
SELECT b.bill_id, b.amount, b.bill_date, a.appointment_date, a.appointment_time
FROM billing b
JOIN appointments a ON b.appointment_id = a.appointment_id
WHERE b.patient_id = ?;
```

Testing Failure Scenarios:

Failure Scenario	Description	Expected Outcome
1. Patient Registration Failure	Attempting to register a patient with missing details (e.g., no phone number).	The operation fails, and no patient data is inserted.
2. Appointment Scheduling Failure	Scheduling an appointment with a doctor who is not available.	The system will raise an error and no appointment is created.
3. Billing Generation Failure	Generating a bill for a non-existent appointment.	The system will raise an error and prevent the bill from being generated.
4. Concurrency Issue (Double Booking)	Two users attempt to schedule an appointment with the same doctor at the same time.	One user will be blocked, and the second will receive an error message.