**Milestone Project 1**

Question 1

* Create a DB Schema for Hospital Management System.
* Define the schema along with the constraints indicating the relationships between the entities.
* Be sure to make use of the database concepts like Views, Relationships, Indexing, Stored Procedure and triggers. Indicate the Normalization form being used in the schema defined and why you chose to keep it that particular normal form.
* Once your schema is well defined, choose any Relational Database system (MySQL, MariaDB, etc.) and practically implement the schema so that you are able to perform at least the following operations.
* HMS should be capable to recognize already registered patients and user roles.
* Creating Database and Tables:

CREATE SCHEMA `hospitalmanagementsystem` ;

USE `hospitalmanagementsystem` ;

* Table Patient:

CREATE TABLE `hospitalmanagementsystem`.`patient` (

`patientID` INT NOT NULL,

`firstName` VARCHAR(10) NULL,

`lastName` VARCHAR(10) NULL,

`dateOfBirt` DATE NULL,

`gender` VARCHAR(6) NULL,

`contNumber` VARCHAR(10) NULL,

`email` VARCHAR(50) NULL,

`address` VARCHAR(45) NULL,

PRIMARY KEY (`patientID`));

* Table userroles:

CREATE TABLE `hospitalmanagementsystem`.`userroles` (

`roleID` INT NOT NULL,

`roleName` VARCHAR(10) NULL,

PRIMARY KEY (`roleID`));

* Table diagnosis:

CREATE TABLE `hospitalmanagementsystem`.`diagnosis` (

`diagnosisID` INT NOT NULL,

`patientID` INT NULL,

`diagnosisDate` DATE NULL,

`diagnosisDetails` VARCHAR(45) NULL,

PRIMARY KEY (`diagnosisID`));

* Table bill:

CREATE TABLE `hospitalmanagementsystem`.`bill` (

`billID` INT NOT NULL,

`patientID` INT NULL,

`totalAmt` INT NULL,

`paymentStatus` VARCHAR(20) NULL,

PRIMARY KEY (`billID`));

* Table insurance:

CREATE TABLE `hospitalmanagementsystem`.`insurance` (

`insuranceID` INT NOT NULL,

`patientID` INT NULL,

`insuranceLimit` INT NULL,

`expiryDate` DATE NULL,

PRIMARY KEY (`insuranceID`));

* Table service:

CREATE TABLE `hospitalmanagementsystem`.`service` (

`ServiceID` INT NOT NULL,

`PatientID` INT NULL,

`ServiceDescription` VARCHAR(45) NULL,

`ServiceCost` DECIMAL(10,2) NULL,

PRIMARY KEY (`ServiceID`));

* Inserting data into insurance Table:

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('1', '1', '230405', '2030-05-01');

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('2', '5', '89512612', '2050-09-01');

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('3', '2', '9999999', '2030-07-30');

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('4', '3', '78909086', '2027-12-27');

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('5', '4', '90887644', '2025-12-12');

1. **Write necessary queries to register new user roles and patients.**

* Inserting data into userroles Table:

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('1', 'Doctor');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('2', 'PHarmacist');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('3', 'Dietitian');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('4', 'Nurse');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('5', 'Therapist');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('6', 'Surgeon');

INSERT INTO `hospitalmanagementsystem`.`userroles` (`roleID`, `roleName`) VALUES ('7', 'Assistant');

* Inserting data into patient Table:

INSERT INTO `hospitalmanagementsystem`.`patient` (`patientID`, `firstName`, `lastName`, `dateOfBirt`, `gender`, `contNumber`, `email`, `address`) VALUES ('1', 'Sayali', 'Thakare', '2000-09-02', 'Female', '7057879081', 'thakaresayalir@gmail.com', 'Nashik');

INSERT INTO `hospitalmanagementsystem`.`patient` (`patientID`, `firstName`, `lastName`, `dateOfBirt`, `gender`, `contNumber`, `email`, `address`) VALUES ('2', 'Rushikesh', 'Thakare', '2004-07-27', 'Male', '9359254181', 'thakarerushi@gmail.com', 'Pune');

INSERT INTO `hospitalmanagementsystem`.`patient` (`patientID`, `firstName`, `lastName`, `dateOfBirt`, `gender`, `contNumber`, `email`, `address`) VALUES ('3', 'Saloni', 'Thakare', '1998-02-08', 'Female', '7055869052', 'thakaresaloni@gmail.com', 'Pune');

INSERT INTO `hospitalmanagementsystem`.`patient` (`patientID`, `firstName`, `lastName`, `dateOfBirt`, `gender`, `contNumber`, `email`, `address`) VALUES ('4', 'Urmila', 'Gawali', '1999-06-25', 'Female', '7586246868', 'gawaliurmi@gmail.com', 'Pune');

INSERT INTO `hospitalmanagementsystem`.`patient` (`patientID`, `firstName`, `lastName`, `dateOfBirt`, `gender`, `contNumber`, `email`, `address`) VALUES ('5', 'Viraj', 'Pawar', '2001-03-11', 'Male', '9384923430', 'pawarviraj@gmail.com', 'Belapur');

* Inserting data into service Table:

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `PatientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1001', '1', 'X-Ray', '60');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `PatientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1002', '1', 'Blood Test', '150');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `PatientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1003', '3', 'Temperature test', '30');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `patientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1005', '2', 'Advanced Cardiac & Pulmonary', '2344');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `patientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1006', '4', 'Heart Monitoring', '6453');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `patientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1007', '5', 'CT Scan', '6543');

INSERT INTO `hospitalmanagementsystem`.`service` (`ServiceID`, `patientID`, `ServiceDescription`, `ServiceCost`) VALUES ('1008', '2', 'Foot & Ankle Surgery', '45443');

1. **Write necessary queries to add to the list of diagnosis of the patient tagged by date.**

* Inserting data into diagnosis Table:

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('101', '1', '2024-01-01', 'Fever');

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('102', '5', '2024-01-05', 'Malaria');

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('103', '2', '2024-01-02', 'Cold');

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('104', '1', '2024-01-05', 'Cancer');

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('105', '5', '2024-02-01', 'Fever');

INSERT INTO `hospitalmanagementsystem`.`diagnosis` (`diagnosisID`, `patientID`, `diagnosisDate`, `diagnosisDetails`) VALUES ('106', '3', '2024-02-05', 'Typhoid');

1. **Write necessary queries to fetch required details of a particular patient.**

SELECT \* FROM Patient WHERE PatientID = '1';

SELECT firstName, lastName FROM Patient WHERE contNumber = '7057879081';

SELECT gender, firstName, lastName FROM Patient WHERE dateOfBirt = '2001-03-11';

SELECT contNumber FROM Patient WHERE firstName = 'Saloni' AND lastName='Thakare';

1. **Write necessary queries to prepare bill for the patient at the end of checkout.**

SELECT SUM(s.ServiceCost) AS TotalCost

FROM Patient p

JOIN Service s ON p.patientID = s.ServiceID

WHERE p.patientID = '1';

1. **Write necessary queries to fetch and show data from various related tables (Joins)**

SELECT

P.patientID,

P.FirstName AS PatientFirstName,

D.roleName AS DoctorFirstName

FROM patient P

JOIN service s ON P.patientID = s.patientID

LEFT JOIN userroles D ON s.patientID = P.patientID;

1. **Optimize repeated read operations using views/materialized views.**

CREATE VIEW PatientDetailsView AS

SELECT

p.patientID,

p.firstName,

p.lastName,

p.dateOfBirt,

p.gender,

p.contNumber,

p.email,

p.address,

d.diagnosisID,

d.diagnosisDate,

d.diagnosisDetails,

b.totalAmt,

b.paymentStatus,

i.insuranceLimit,

i.expiryDate

FROM patient p

LEFT JOIN diagnosis d ON p.patientID = d.patientID

LEFT JOIN bill b ON p.patientID = b.patientID

LEFT JOIN insurance i ON p.patientID = i.patientID;

SELECT \* FROM PatientDetailsView;

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

-- Create an index

CREATE INDEX idx\_Diagnosis\_Patient\_Date ON diagnosis (patientID, diagnosisDate);

-- Query without index

EXPLAIN SELECT \* FROM diagnosis WHERE diagnosisDate = '2024-01-02';

-- Query with index

EXPLAIN SELECT \* FROM diagnosis USE INDEX (idx\_Diagnosis\_Patient\_Date) WHERE diagnosisDate = '2024-01-01';

1. **Try optimizing bill generation using stored procedures.**

DELIMITER //

CREATE PROCEDURE GenerateBillProcedure(IN p\_PatientID INT)

BEGIN

DECLARE totalAmount DECIMAL(10, 2);

-- Calculate total amount based on services, treatments, etc.

SELECT SUM(ServiceCost) INTO totalAmount

FROM Service

WHERE PatientID = p\_PatientID;

-- Inserting the calculated total amount into the Bill table

INSERT INTO Bill (patientID, totalAmt, paymentStatus)

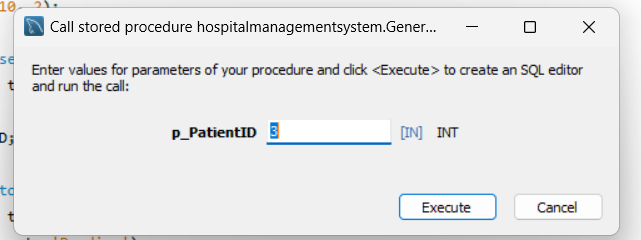
VALUES (p\_PatientID, totalAmount, 'Pending');

END //

DELIMITER ;

call hospitalmanagementsystem.GenerateBillProcedure(1);

call hospitalmanagementsystem.GenerateBillProcedure(3);

****

1. **Add necessary triggers to indicate when patient’s medical insurance limit has expired.**

DELIMITER //

CREATE TRIGGER InsuranceExpiryTrigger

BEFORE INSERT ON insurance

FOR EACH ROW

BEGIN

DECLARE today DATE;

DECLARE insuranceExpiryDate DATE;

-- Get the current date

SET today = CURDATE();

-- Get the insurance expiry date for the current record

SET insuranceExpiryDate = NEW.ExpiryDate;

-- Check if the insurance has expired

IF insuranceExpiryDate < today THEN

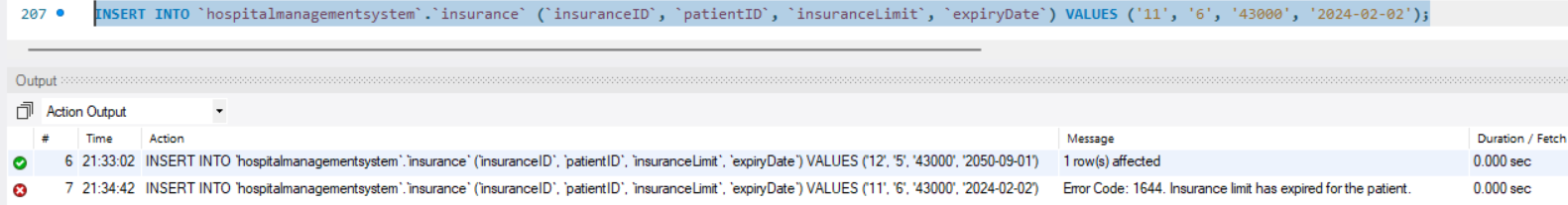
SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Insurance limit has expired for the patient.';

END IF;

END //

DELIMITER ;

INSERT INTO `hospitalmanagementsystem`.`insurance` (`insuranceID`, `patientID`, `insuranceLimit`, `expiryDate`) VALUES ('11', '6', '43000', '2024-02-02');

Question 2

1. **Write a report on your understanding of Rendering and Design Patterns, Mention and elaborate where a particular Rendering pattern is applicable and is well suited for which use case.**

Combination of rendering patterns like MVC and design patterns can enhance the system's performance and user experience. The MVC pattern helps structure the application, while rendering patterns like Template View and Observer to specific needs in displaying dynamic content(example – insurance expiry trigger) and handling real-time updates, respectively. These patterns contribute to an efficient and robust HMS that meets the demands of the healthcare environment.