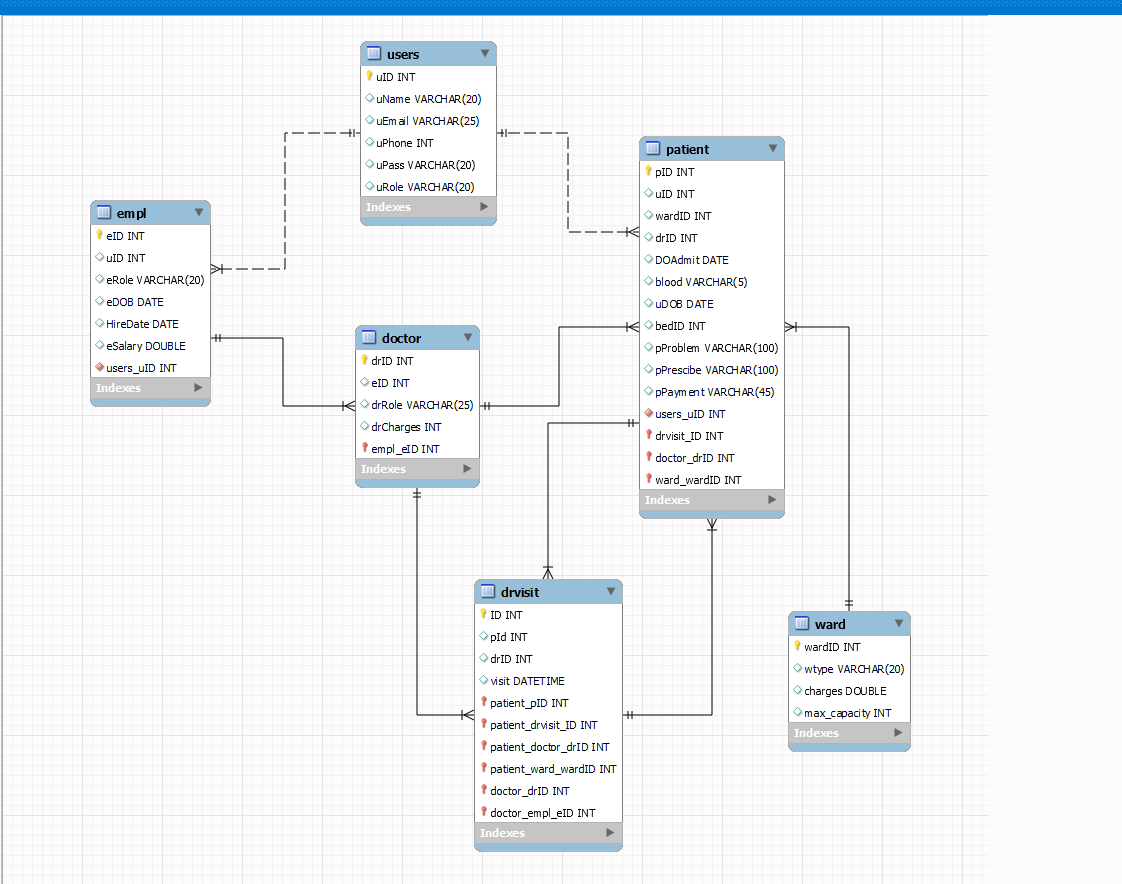
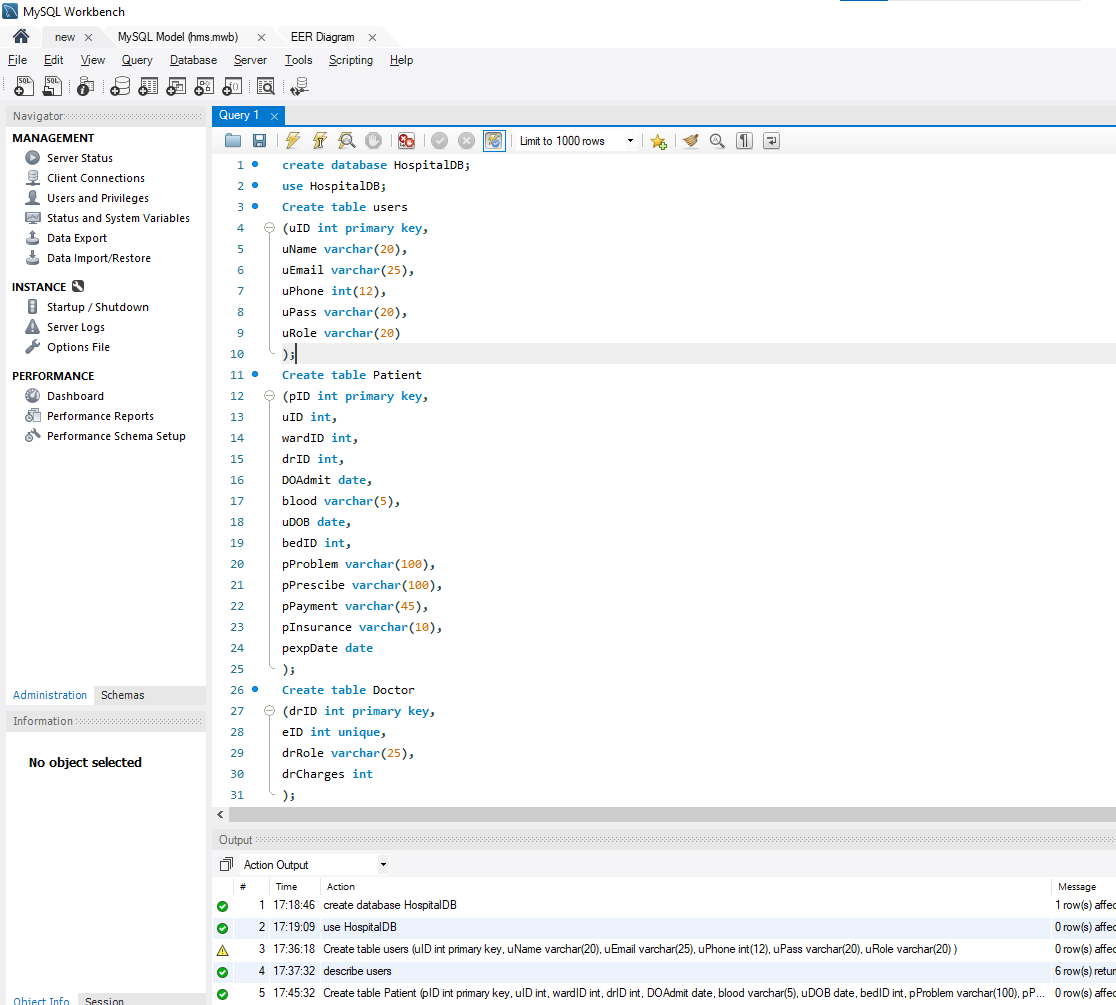
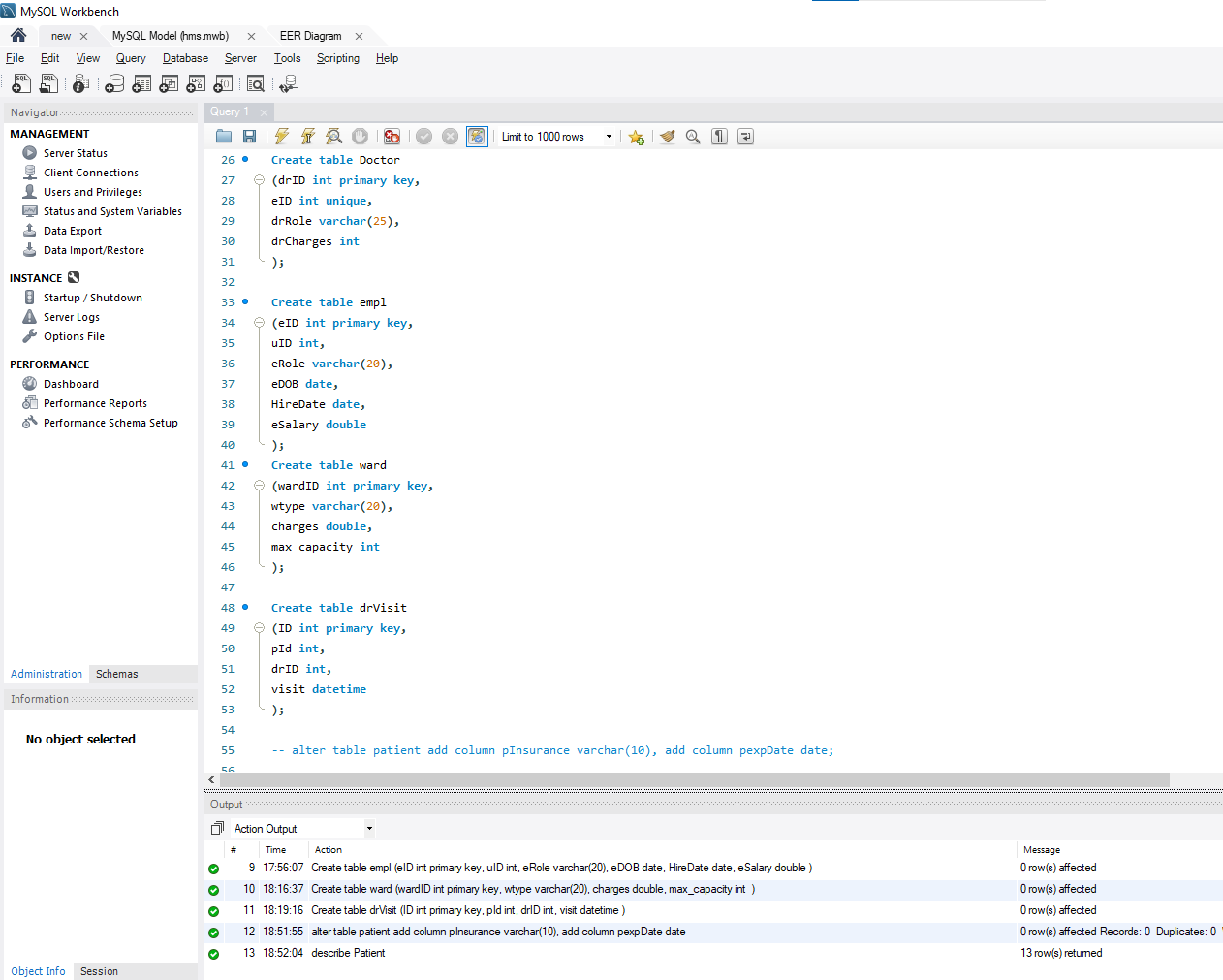
Q1. Hospital Management System Schema

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.



**Code:**

**create database HospitalDB;**

**use HospitalDB;**

**Create table users**

**( uID int primary key,**

**uName varchar(20),**

**uEmail varchar(25),**

**uPhone int(12),**

**uPass varchar(20),**

**uRole varchar(20)**

**);**

**Create table Patient**

**( pID int primary key,**

**uID int,**

**wardID int,**

**drID int,**

**DOAdmit date,**

**blood varchar(5),**

**uDOB date,**

**bedID int,**

**pProblem varchar(100),**

**pPrescibe varchar(100),**

**pPayment varchar(45),**

**pInsurance varchar(10),**

**pexpDate date**

**);**

**Create table Doctor**

**(drID int primary key,**

**eID int unique,**

**drRole varchar(25),**

**drCharges int**

**);**

**Create table empl**

**(eID int primary key,**

**uID int,**

**eRole varchar(20),**

**eDOB date,**

**HireDate date,**

**eSalary double**

**);**

**Create table ward**

**(wardID int primary key,**

**wtype varchar(20),**

**charges double,**

**max\_capacity int**

**);**

**Create table drVisit**

**(ID int primary key,**

**pId int,**

**drID int,**

**visit datetime**

**);**

- Write necessary queries to register new user roles and personas

**INSERT INTO users (uID, uName, uEmail, uPhone, uPass, uRole)**

**VALUES**

**(1, 'Admin', 'admin@example.com', 1234567890, 'adminpass', 'Administrator'),**

**(2, 'Doctor', 'doctor@example.com', 9876543210, 'doctorpass', 'Doctor'),**

**(3, 'Nurse', 'nurse@example.com', 8765432109, 'nursepass', 'Nurse'),**

**(4, 'ram', 'ram@example.com', 9876543210, 'rampass', 'Patient'),**

**(5, 'Sita', 'sita@example.com', 8765432109, 'sitapass', 'Patient')**

**;**

**INSERT INTO empl (eID, uID, eRole, eDOB, HireDate, eSalary) VALUES**

**(1, 6, 'Receptionist', '1999-05-15', '2023-01-01', 15000.00),**

**(2, 2, 'Doctor', '1985-08-20', '2013-02-15', 140000.00),**

**(3, 8, 'Doctor', '1982-03-10', '2015-03-20', 200000.00),**

**(4, 7, 'Janitor', '1980-08-20', '2012-02-15', 300000.00),**

**(5, 3, 'Nurse', '1997-03-10', '2022-03-20', 20000.00)**

**;**

**INSERT INTO Doctor (drID, eID, drRole, drCharges)**

**VALUES**

**(1, 2, 'General Practitioner', 10000),**

**(2, 3, 'Surgeon', 15000)**

**;**

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

**INSERT INTO Patient (**

**pID, uID, wardID, drID, DOAdmit, blood, uDOB, bedID, pProblem, pPrescibe, pPayment, pInsurance, pExpDate**

**) VALUES**

**(3, 8, 1, 1, '2023-01-29', 'A+', '1997-01-01', 101, 'Fever and headache', 'Paracetamol and analgesics', 'Pending', 'Yes', '2027-01-29'),**

**(4, 9, 2, 2, '2024-01-30', 'B-', '1990-05-15', 102, 'Broken leg', 'Painkillers', 'Paid', 'No', NULL),**

**(5, 5, 1, 1, '2024-01-29', 'O+', '1999-01-01', 101, 'Skin Problem ', 'Cream', 'Pending', 'Yes', '2025-01-29'),**

**(6, 6, 2, 2, '2024-01-30', 'AB-', '2000-05-15', 102, 'Acne', 'Painkillers', 'Pending', 'No', NULL)**

**;**

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

**Select \* from Patient where pID=2;**

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

**SELECT**

**pID, u.uname, d.drCharges, w.charges as Wardcharges, sum(d.drCharges + w.charges) as TotalSum**

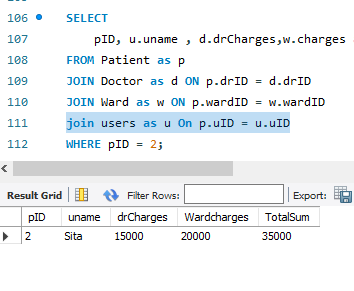
**FROM Patient as p**

**JOIN Doctor as d ON p.drID = d.drID**

**JOIN Ward as w ON p.wardID = w.wardID**

**JOIN users as u On p.uID = u.uID**

**WHERE pID = 2;**

****

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

For Payment Join Query

**SELECT**

**users.uName, Patient.pID, Doctor.drRole as DrAssigned, Ward.wtype**

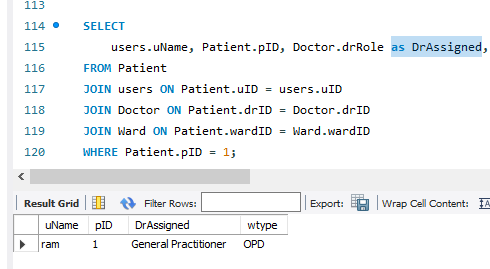
**FROM Patient**

**JOIN users ON Patient.uID = users.uID**

**JOIN Doctor ON Patient.drID = Doctor.drID**

**JOIN Ward ON Patient.wardID = Ward.wardID**

**WHERE Patient.pID = 1;**

****

- Optimize repeated read operations using views/materialized views.

-- Create a view for frequently accessed information

**CREATE VIEW PatientDetails AS**

**SELECT**

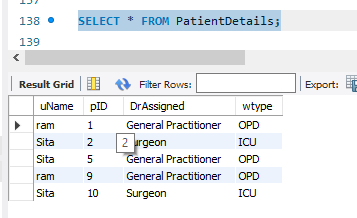
**users.uName, Patient.pID, Doctor.drRole as DrAssigned, Ward.wtype**

**FROM Patient**

**JOIN users ON Patient.uID = users.uID**

**JOIN Doctor ON Patient.drID = Doctor.drID**

**JOIN Ward ON Patient.wardID = Ward.wardID;**



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

**CREATE INDEX idx\_patient\_id ON Patient(pID);**

- Try optimizing bill generation using stored procedures.

**DELIMITER //**

**CREATE PROCEDURE GenerateBill ( IN patientID INT)**

**BEGIN**

**SELECT**

**pID, u.uname , d.drCharges,w.charges as Wardcharges, sum(d.drCharges+w.charges) as TotalSum**

**FROM Patient as p**

**JOIN Doctor as d ON p.drID = d.drID**

**JOIN Ward as w ON p.wardID = w.wardID**

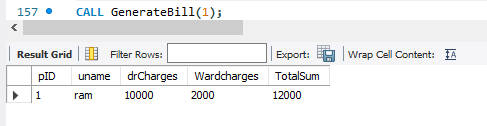
**join users as u On p.uID = u.uID**

**WHERE pID = patientID**

**GROUP BY pID, u.uname;**

**END //**

**DELIMITER ;**

****

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

**DELIMITER //**

**CREATE TRIGGER CheckInsuranceExpiry**

**BEFORE INSERT ON Patient**

**FOR EACH ROW**

**BEGIN**

**IF NEW.pExpDate < CURDATE() THEN**

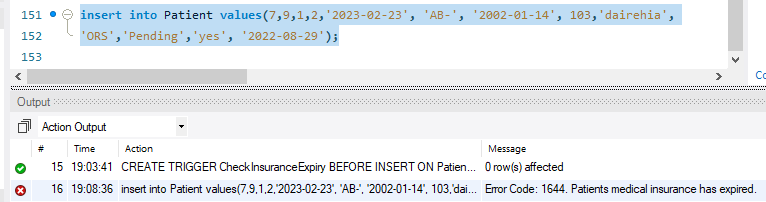
**SIGNAL SQLSTATE '45000'**

**SET MESSAGE\_TEXT = 'Patients medical insurance has expired.';**

**END IF;**

**END //**

**DELIMITER ;**

****