**Assessment 1: 5M**

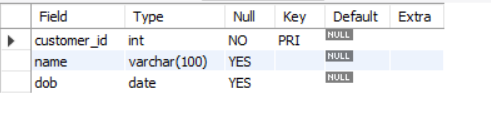
**Domain:** Insurance

**Note:** Please create the database insurance\_db

**Q1. Create the Customers table with the following columns: ----- 1M**

* customer\_id (INT, Primary Key)
* name (VARCHAR)
* dob (DATE)

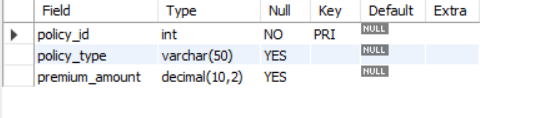
**Expected Output:**



**Q2. Create the Policies table with: ----- 1M**

* policy\_id (INT, Primary Key)
* policy\_type (VARCHAR)
* premium\_amount (DECIMAL)

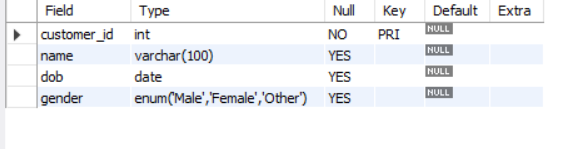
**Expected Output:**



### ****Q3. Add a new column**** gender ****to the**** Customers ****table.****

### **Data type:** ENUM('Male', 'Female', 'Other') ------ 1M

**Expected Output:**

****

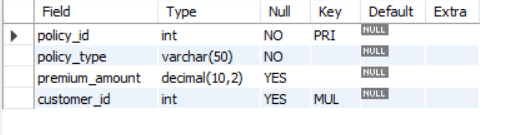
### ****Q4. Add a foreign key**** customer\_id ****to**** Policies ****referencing**** customers(customer\_id). ------ 1M

**Expected Output:**

### 

### ****Q5. Add a NOT NULL constraint to the**** policy\_type ****column in**** Policies****. 1M****

**Expected Output:**

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Assessment 2: 25M**

**Domain:** Hospital

Please execute **Step 1** and **Step 2** for the setting up of database **hospital\_db** with tables and data.

**Step 1: Table Creation Queries**

-- Patients table

CREATE TABLE Patients (

patient\_id INT PRIMARY KEY,

name VARCHAR(100),

dob DATE,

gender VARCHAR(10)

);

-- Doctors table

CREATE TABLE Doctors (

doctor\_id INT PRIMARY KEY,

name VARCHAR(100),

specialization VARCHAR(50)

);

-- Appointments table

CREATE TABLE Appointments (

appointment\_id INT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

appointment\_date DATE,

status VARCHAR(20),

fee DECIMAL(10, 2),

FOREIGN KEY (patient\_id) REFERENCES Patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES Doctors(doctor\_id)

);

-- Treatments table

CREATE TABLE Treatments (

treatment\_id INT PRIMARY KEY,

appointment\_id INT,

treatment\_details VARCHAR(200),

cost DECIMAL(10, 2),

FOREIGN KEY (appointment\_id) REFERENCES Appointments(appointment\_id)

);

**Step 2: Sample Insert Queries**

-- Patients

INSERT INTO Patients VALUES

(1, 'John Doe', '1990-05-20', 'Male'),

(2, 'Jane Smith', '1985-11-30', 'Female'),

(3, 'Alice Brown', '2000-08-15', 'Female');

-- Doctors

INSERT INTO Doctors VALUES

(101, 'Dr. Shah', 'Cardiologist'),

(102, 'Dr. Meera', 'Neurologist');

-- Appointments

INSERT INTO Appointments VALUES

(1001, 1, 101, '2024-04-01', 'Completed', 500.00),

(1002, 2, 102, '2024-04-02', 'Cancelled', 0.00),

(1003, 3, 101, '2024-04-03', 'Completed', 700.00);

-- Treatments

INSERT INTO Treatments VALUES

(201, 1001, 'ECG and medication', 300.00),

(202, 1003, 'Angioplasty', 1200.00);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INSERT INTO Patients (patient\_id, name, dob, gender)

VALUES (5, 'Sam Thomas', '1990-09-12', 'Male');

INSERT INTO Doctors (doctor\_id, name, specialization)

VALUES (105, 'Dr. Kapoor', 'Orthopedics');

INSERT INTO Appointments (appointment\_id, patient\_id, doctor\_id, appointment\_date, status, fee)

VALUES

(2001, 5, 105, '2025-04-01', 'Completed', 400.00),

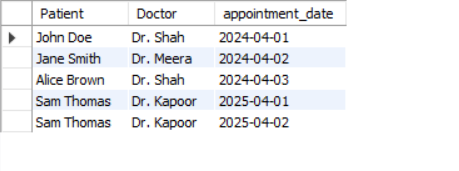
(2002, 5, 105, '2025-04-02', 'Completed', 450.00);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Questions:**

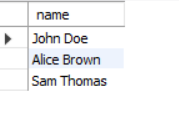
1. Retrieve the names of patients along with the names of doctors they had appointments with, and the corresponding appointment dates. **----- 2M**

**Expected Output:**



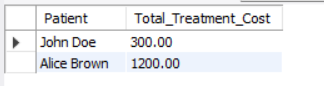
1. Show the names of patients who paid more than the average consultation fee for appointments. **----- 2M**

**Expected Output:**

****

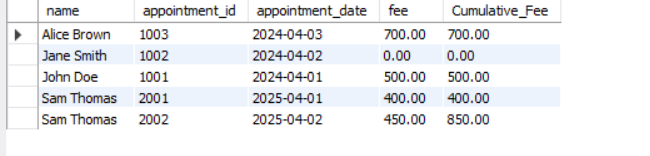
1. Display the total cost of treatments given to each patient. **----- 2M**

**Expected Output:**

****

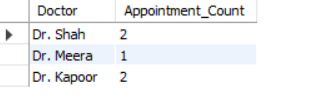
1. List the appointment ID, patient name, and cumulative fee paid by the patient over time, in order of appointment date. **----- 2M**

**Expected Output:**

****

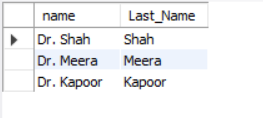
1. Show each doctor’s name and the number of appointments they have. **----- 2M**

**Expected Output:**

****

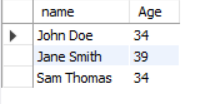
1. List doctors’ last names. **----- 1M**

**Expected Output:**

****

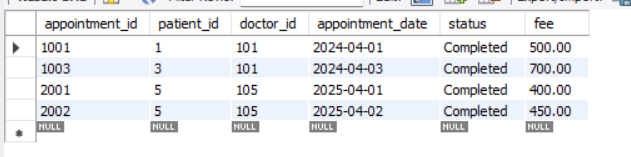
1. Show the name and age of all patients based on their date of birth whose age is more than 30 years. **----- 2M**

**Expected Output:**

****

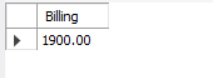
1. Display all appointments that are either marked as completed or had a non-zero fee. **----- 2M**

**Expected Output:**

****

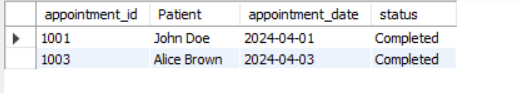
1. Create a function that returns the total amount (fee + treatment cost) for an appointment. Then, test it for appointment ID 1003. **----- 3M**

**Expected Output:**

****

1. Create a procedure that returns appointments for a doctor based on a given doctor ID. Test it with doctor ID 101. **----- 3M**

**Expected Output:**

****

1. Find the doctor who generated the highest revenue (fee + treatment cost) from appointments. **----- 2M**

**Expected Output:**

****

1. List patients who had back-to-back appointments (i.e., appointments on consecutive days). **----- 2M**

**Expected Output:**

****