Hospital management system Project- 16

29th Feb 2024

Name:

Email:

**HMS Project Structure**

1. **Database Creation Script:**
   * Create a SQL script to generate the database schema and initial data.
   * Tables
   * Appointment
   * Doctor
   * Patients
   * PatientsAttendAppointments
   * Medical history
   * PatientFillHistory
   * Medicine Cost

Queries to Solve within the HMS Structure

* Find the names of patients who have attended appointments scheduled by Dr. John Doe.
* Calculate the average age of all patients.
* Create a stored procedure to get the total number of appointments for a given patient.
* Create a trigger to update the appointment status to 'Completed' when the appointment date has passed.
* Find the names of patients along with their appointment details and the corresponding doctor's name.
* Find the patients who have a medical history of diabetes and their next appointment is scheduled within the next 7 days.
* Find patients who have multiple appointments scheduled.
* Calculate the average duration of appointments for each doctor.
* Find Patients with Most Appointments
* Calculate the total cost of medication for each patient.
* Create a stored procedure named CalculatePatientBill that calculates the total bill for a patient based on their medical history and medication costs. The procedure should take the PatientID as a parameter and calculate the total cost by summing up the medication costs and applying a charge of $50 for each surgery in the patient's medical history. If the patient has no medical history, the procedure should still return a basic charge of $50.

Note: Kindly create a Docx file and paste the queries there with a short brief as a conclusion para, and also keep your SQL server open at the time of presentation.

Kindly Paste your Queries Here

--1. Find the names of patients who have attended appointments scheduled by Dr. John Doe.

SELECT P.PatientID,P.Fname AS PatientFirstName, P.Lname AS PatientLastName

FROM Patients P

JOIN Patient\_Attendence PA ON P.PatientID = PA.PatientID

JOIN Appointment A ON PA.AppointmentID = A.AppointmentID

JOIN Doctors D ON A.DoctorID = D.DoctorID

WHERE D.DoctorID = 1;

--2. Calculate the average age of all patients.

SELECT AVG(Age) AS Average\_Age

FROM Patients;

--Average age of patients is 37

--3. Create a stored procedure to get the total number of appointments for a given patient.

CREATE PROCEDURE GetTotalAppointments (@PatientID INT)

AS

BEGIN

SELECT COUNT(\*) AS Total\_Appointments

FROM Appointment

WHERE PatientID = @PatientID;

END;

EXEC GetTotalAppointments @PatientID = 10;

--4. Create a trigger to update the appointment status to 'Completed' when the appointment date has passed.

CREATE TRIGGER UpdateAppointmentStatus

ON Appointment

AFTER UPDATE

AS

BEGIN

UPDATE Appointment

SET Status = 'Completed'

WHERE Date < GETDATE() AND Status != 'Completed';

END;

--5. Find the names of patients along with their appointment details and the corresponding doctor's name.

SELECT P.Fname AS Patient\_FirstName, P.Lname AS Patient\_LastName,

A.Date AS Appointment\_Date, A.Endtime AS Appointment\_EndTime,

D.Fname AS Doctor\_FirstName, D.Lname AS Doctor\_LastName

FROM Patients P

JOIN Appointment A ON P.PatientID = A.PatientID

JOIN Doctors D ON A.DoctorID = D.DoctorID;

--6. Find the patients who have a medical history of diabetes and their next appointment is scheduled within the next 7 days.

-- Define the date range for the next 7 days

-- Query to find patients with diabetes and schedule their next appointment

SELECT P.Fname, P.Lname, MH.Date AS ExistingAppointmentDate,MH.condition,

DATEADD(DAY, 7, MH.Date) AS SuggestedNextAppointmentDate

FROM Medical\_History MH

JOIN Patients P ON MH.PatientID = P.PatientID

WHERE MH.Condition = 'Diabetes';

--7. Find patients who have multiple appointments scheduled.

SELECT P.Fname, P.Lname, COUNT(PA.AppointmentID) AS Total\_Appointments

FROM Patients P

JOIN Patient\_Attendence PA ON P.PatientID = PA.PatientID

GROUP BY P.PatientID, P.Fname, P.Lname

HAVING COUNT(PA.AppointmentID) > 1;

--8. Calculate the average duration of appointments for each doctor.

SELECT D.Fname, D.Lname, AVG(DATEDIFF(MINUTE, A.Date, A.Endtime)) AS Avg\_Appointment\_Duration

FROM Appointment A

JOIN Doctors D ON A.DoctorID = D.DoctorID

GROUP BY D.DoctorID, D.Fname, D.Lname;

UPDATE Appointment SET EndTime = '2023-11-09 13:21:00.0000000'

WHERE EndTime = '2023-11-09 01:21:00.0000000';

--9.Find Patients with Most Appointments.

SELECT P.Fname, P.Lname, COUNT(PA.AppointmentID) AS Appointment\_Count

FROM Patients P

JOIN Patient\_Attendence PA ON P.PatientID = PA.PatientID

GROUP BY P.PatientID, P.Fname, P.Lname

ORDER BY Appointment\_Count DESC;

--10. Calculate the total cost of medication for each patient.

SELECT P.Fname, P.Lname, SUM(MC.Cost\_in\_$) AS Total\_Medication\_Cost

FROM Patients P

JOIN Medical\_History MH ON P.PatientID = MH.PatientID

JOIN Medication\_Cost MC ON MH.Medication = MC.Medication

GROUP BY P.PatientID, P.Fname, P.Lname;

--11.Create a stored procedure named CalculatePatientBill that calculates the total bill for a patient.

CREATE PROCEDURE CalculatePatientBill

@PatientID INT

AS

BEGIN

-- Declare a variable to hold the total cost

DECLARE @TotalBill DECIMAL(10, 2);

-- Calculate the total cost of medications for the patient

-- and sum up the cost for each medication

DECLARE @MedicationCost DECIMAL(10, 2);

SELECT @MedicationCost = ISNULL(SUM(MC.cost\_in\_$), 0)

FROM Medical\_History MH

JOIN Medical\_History MFH ON MH.HistoryID = MFH.HistoryID

JOIN Medication\_Cost MC ON MFH.Medication = MC.Medication

WHERE MH.PatientID = @PatientID;

-- Calculate the number of surgeries and apply $50 charge per surgery

DECLARE @SurgeryCount INT;

SELECT @SurgeryCount = ISNULL(COUNT(\*), 0)

FROM Medical\_History

WHERE PatientID = @PatientID

AND Surgeries IS NOT NULL

AND Surgeries <> '';

-- Calculate total bill

SET @TotalBill = @MedicationCost + (@SurgeryCount \* 50);

-- If no medical history, ensure a minimum charge of $50

IF @TotalBill = 0

BEGIN

SET @TotalBill = 50;

END

-- Return the total bill

SELECT @TotalBill AS TotalBill;

END;

EXEC CalculatePatientBill @PatientID = 1;

EXEC CalculatePatientBill @PatientID = 2;

EXEC CalculatePatientBill @PatientID = 3;

EXEC CalculatePatientBill @PatientID = 4;

EXEC CalculatePatientBill @PatientID = 5;

EXEC CalculatePatientBill @PatientID = 6;

EXEC CalculatePatientBill @PatientID = 7;

EXEC CalculatePatientBill @PatientID = 8;

EXEC CalculatePatientBill @PatientID = 9;

EXEC CalculatePatientBill @PatientID = 10;

Conclusions (Summary):