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

CREATE DATABASE Health\_management\_system;

use Health\_management\_system;

CREATE TABLE patient\_table(PatientID varchar(max),Fname varchar(max),Lname varchar(max),

Contact varchar(max),Age varchar(max));

DROP TABLE patient\_table

CREATE TABLE patient(PatientID varchar(max),Fname varchar(max),Lname varchar(max),

Contact varchar(max),Age varchar(max));

BULK INSERT patient FROM 'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\PATIENT\_TABLE.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* from patient;

CREATE TABLE DOCTOR(DoctorID varchar(max),Fname varchar(max),Lname varchar(max),

Speciality varchar(max),ContactEmail varchar(max));

BULK INSERT DOCTOR FROM 'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\doctor.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* from DOCTOR;

DROP TABLE DOCTOR;

CREATE TABLE Appointment(AppointmentID varchar(max),PatientID varchar(max), DoctorID varchar(max),Date varchar(max),

EndTime varchar(max),Status varchar(max));

BULK INSERT Appointment FROM 'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\Appointment.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* from Appointment;

CREATE TABLE PatientsAttendAppointments(PatientID varchar(max), AppointmentID varchar(max));

BULK INSERT PatientsAttendAppointments FROM

'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\PatientsAttendAppointments Table.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* from PatientsAttendAppointments;

CREATE TABLE MedicalHistory(HistoryID varchar(max), PatientID varchar(max), Date varchar(max),

Condition varchar(max), Surgeries varchar(max), Medication varchar(max));

BULK INSERT MedicalHistory FROM

'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\MedicalHistory Table.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* from MedicalHistory;

CREATE TABLE PatientsFillHistory(PatientID varchar(max),HistoryID varchar(max), DateFilled varchar(max));

BULK INSERT PatientsFillHistory FROM

'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\PatientsFillHistory Table.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

CREATE TABLE Medication\_Cost(Medication varchar(max), Cost\_in$ varchar(max));

BULK INSERT Medication\_Cost FROM

'C:\Users\mamta\OneDrive\Desktop\PPTPROJECT\reproject\SQL\_PROJECT\_DATA\_CSV\Medication\_Cost.csv'

with ( fieldterminator=',' ,

rowterminator ='\n',

firstrow= 2);

select \* FROM patient

SELECT \* FROM DOCTOR;

SELECT \* FROM Appointment;

SELECT \* FROM PatientsAttendAppointments;

SELECT \* FROM MedicalHistory;

SELECT \* FROM PatientsFillHistory;

SELECT \* FROM Medication\_Cost;

select column\_name, data\_type

from INFORMATION\_SCHEMA.columns;

ALTER TABLE patient

ALTER COLUMN Age INT

**/\*in contact there are -(hyphen) between numbers so we have to update column to remove hyphen\*/**

UPDATE patient

SET Contact = REPLACE(Contact, '-', '');

**/\*in appointment table there is date and time in same column so we have to seperate this**

**------------------------------------------------------------**

**---------------------------------**

**=------------**

**------\*/**

select date,right(date,len(date)-charindex(' ',date))

as 'Start\_Time' from Appointment;

**/\*create new column of start timing of appointment\*/**

ALTER TABLE Appointment

ADD Start\_time varchar(20)

update Appointment set Start\_Time=right(date,len(date)-charindex(' ',date));

;

SELECT \* FROM Appointment;

**/\*trim data from date and seprate date and time\*/**

select Date,len(Date),len(date)-charindex(' ', Date)

from Appointment;

select Date,left(date,10)

from Appointment;

UPDATE Appointment SET Date=left(Date,10)

/\*seprate end time

----------------

-----------

\*/

Select EndTime,right(EndTime,len(EndTime)-charindex(' ',EndTime))

as 'End\_Time' from Appointment;

**/\*create new column of end timing of appointment\*/**

ALTER TABLE Appointment

ADD End\_Time varchar(20)

update Appointment set End\_Time=right(EndTime,len(EndTime)-charindex(' ',EndTime));

;

SELECT \* FROM Appointment;

/\*remove time data from endtime\*/

select EndTime,len(EndTime),len(EndTime)-charindex(' ', EndTime)

from Appointment;

select Date,left(EndTime,10) as 'End\_Date'

from Appointment;

ALTER TABLE Appointment

DROP COLUMN EndTime;

UPDATE Appointment SET EndTime=left(EndTime,10)

/\*seperate date time from medical history

------------------------------------

----------------------------

\*/

select date,right(date,len(date)-charindex(' ',date))

as 'Appointment\_Time' from MedicalHistory;

/\*add new column in medical historyy table\*/

ALTER TABLE MedicalHistory

ADD Appointment\_Time varchar(20)

/\*to fill timimg in appointment time\*/

update MedicalHistory set Appointment\_Time=right(date,len(date)-charindex(' ',date));

/\* remove time from date column\*/

select Date,len(Date),len(Date)-charindex(' ', Date)

from MedicalHistory;

select Date,left(date,10) as Appointment\_Date

from MedicalHistory;

UPDATE MedicalHistory SET Date=left(Date,10)

select Date,right(Date,4)

select \* from MedicalHistory;

/\*edit patienthistory

---------------

\*/

SELECT \* FROM PatientsFillHistory;

Select DateFilled,right(DateFilled,len(DateFilled)-charindex(' ',DateFilled))

as 'Time' from PatientsFillHistory ;

/\*add new column in medical historyy table\*/

ALTER TABLE PatientsFillHistory

ADD Time varchar(20)

/\*to fill timimg in appointment time\*/

update PatientsFillHistory set Time=right(DateFilled,len(DateFilled)-charindex(' ',DateFilled));

/\* remove time from date column\*/

select DateFilled,len(DateFilled),len(DateFilled)-charindex(' ', DateFilled)

from PatientsFillHistory;

select DateFilled,left(DateFilled,10)

from PatientsFillHistory;

UPDATE PatientsFillHistory SET DateFilled=left(DateFilled,10)

select DateFilled,right(DateFilled,4) from PatientsFillHistory

select \* from MedicalHistory;

/\*Update Medication\_Cost\*/

SELECT \* FROM Medication\_Cost

/\*QUERY----------------------------

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

-------------------------------

\*/

SELECT DISTINCT P.Fname, P.Lname

FROM Patient P

JOIN PatientsAttendAppointments PAA ON P.PatientID = PAA.PatientID

JOIN Appointment A ON PAA.AppointmentID = A.AppointmentID

WHERE A.DoctorID = 'D0001';

/\* ● Calculate the average age of all patients.\*/

SELECT AVG(AGE) AS Average\_age\_OF\_ALL\_PATIENT

FROM PATIENT;

/\*● Create a stored procedure to get the total number of appointments for a given patient.\*/

CREATE PROCEDURE GetTotalAppointmentsForPatient

@PatientIDInput VARCHAR(10)

AS

BEGIN

SELECT COUNT(\*) AS Total\_Appointments

FROM PatientsAttendAppointments

WHERE PatientID = @PatientIDInput;

END;

EXEC GetTotalAppointmentsForPatient 'P0001';

EXEC GetTotalAppointmentsForPatient 'P0004';

/\*● 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'

FROM Appointment A

INNER JOIN inserted i ON A.AppointmentID = i.AppointmentID

WHERE i.Date < GETDATE();

END;

/\*● Find the names of patients along with their

appointment details and the corresponding doctor's name.\*/

SELECT CONCAT(p.Fname,' ',p.Lname) as patient\_name,

a.Date as Apointment\_date,a.Status as appointment\_status,

CONCAT(d.Fname,' ',d.Lname) as Doctor\_name

FROM PATIENT p

join Appointment a

ON a.patientID = p.patientID

join DOCTOR d

on d.DoctorID =a.DoctorID

/\*● Find the patients who have a medical history of diabetes and their

next appointment is scheduled within the next 7 days.\*/

SELECT CONCAT(p.Fname,' ',p.Lname) as patient\_name,

MH.Condition,A.Date as appointment\_Date

FROM PATIENT P

JOIN MedicalHistory MH

ON P.PatientID = MH.PatientID

JOIN Appointment A

ON P.PatientID = A.PatientID

WHERE MH.Condition= 'diabetes'

AND A.Status= 'schedule'

AND A.Date BETWEEN GETDATE() AND DATEADD(DAY, 7, GETDATE())

SELECT

CONCAT(p.Fname, ' ', p.Lname) AS Patient\_FullName,

a.Date AS Next\_Appointment\_Date

FROM

Patient p

JOIN

MedicalHistory mh ON p.PatientID = mh.PatientID

JOIN

Appointment a ON p.PatientID = a.PatientID

WHERE

mh.Condition = 'Diabetes'

AND a.Status = 'Scheduled'

AND a.Date BETWEEN GETDATE() AND DATEADD(DAY, 7, GETDATE());/\*not running ,error shown datatype varchar\*/

ALTER TABLE Appointment

ADD Converted\_Date DATETIME;

/\*Find patients who have multiple appointments scheduled.\*/

SELECT

CONCAT(p.Fname, ' ',p.Lname) AS Patient\_FullName,

COUNT(a.AppointmentID) AS Appointment\_Count

FROM

Patient p

JOIN

Appointment a ON p.PatientID = a.PatientID

WHERE

a.Status = 'Scheduled'

GROUP BY

p.PatientID, p.Fname, p.Lname

HAVING

COUNT(a.AppointmentID) > 1;

/\*

● Calculate the average duration of appointments for each doctor.

\*/

SELECT

DoctorID,

AVG(DATEDIFF(MINUTE, Start\_Time, End\_Time)) AS AvgDuration\_IN\_MIN

FROM

Appointment

WHERE

Status != 'Cancelled'

GROUP BY

DoctorID;

SELECT \* FROM Appointment;

/\*● Find Patients with Most Appointments\*/

SELECT

PatientID,

COUNT(\*) AS TotalAppointments

FROM

Appointment

GROUP BY

PatientID

ORDER BY

TotalAppointments DESC;

/\*● Calculate the total cost of medication for each patient.\*/

SELECT

p.PatientID,

SUM(CAST(m.Cost\_in$ AS DECIMAL(10, 2))) AS TotalCost

FROM

PatientsFillHistory pfh

JOIN

MedicalHistory mh ON pfh.HistoryID = mh.HistoryID

JOIN

Medication\_Cost m ON mh.Medication = m.Medication -- Assuming 'Medication' column is in MedicalHistory

JOIN

Patient p ON p.PatientID = mh.PatientID

GROUP BY

p.PatientID;

/\* -----● 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.-\*/

CREATE PROCEDURE CalculatePatientBill

@PatientID NVARCHAR(10)

AS

BEGIN

DECLARE @TotalCost DECIMAL(10, 2) = 0;

DECLARE @SurgeryCount INT = 0;

DECLARE @MedicationCost DECIMAL(10, 2) = 0;

-- Calculate the number of surgeries

SELECT @SurgeryCount = COUNT(Surgeries)

FROM MedicalHistory

WHERE PatientID = @PatientID

AND Surgeries IS NOT NULL;

-- Calculate the total medication cost, converting the cost to DECIMAL

SELECT @MedicationCost = SUM(CAST(MC.Cost\_in$ AS DECIMAL(10, 2)))

FROM MedicalHistory MH

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

WHERE MH.PatientID = @PatientID;

-- Add base cost of $50 if there is no history

IF NOT EXISTS (SELECT 1 FROM MedicalHistory WHERE PatientID = @PatientID)

BEGIN

SET @TotalCost = 50;

END

ELSE

BEGIN

-- Calculate the total cost (Medication + $50 for each surgery)

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

END

-- Output the total cost

SELECT @TotalCost AS TotalBill;

END;

EXEC CalculatePatientBill 'P0001';

Conclusions (Summary):