Logo

Description automatically generated

Must University for Science & Technology

College of Information Technology  
computer science Department

**IS413 - Design & Implementation of Database**

**Team members**

***Amr Osama Khafagy 200031891***

***Rabea Ibrahim Ibrahim 200031640***

***Menna Tallah Ibrahim kamel 200018656  
Mayar magdy omar 200013290***

**Project name** : Dental Clinic

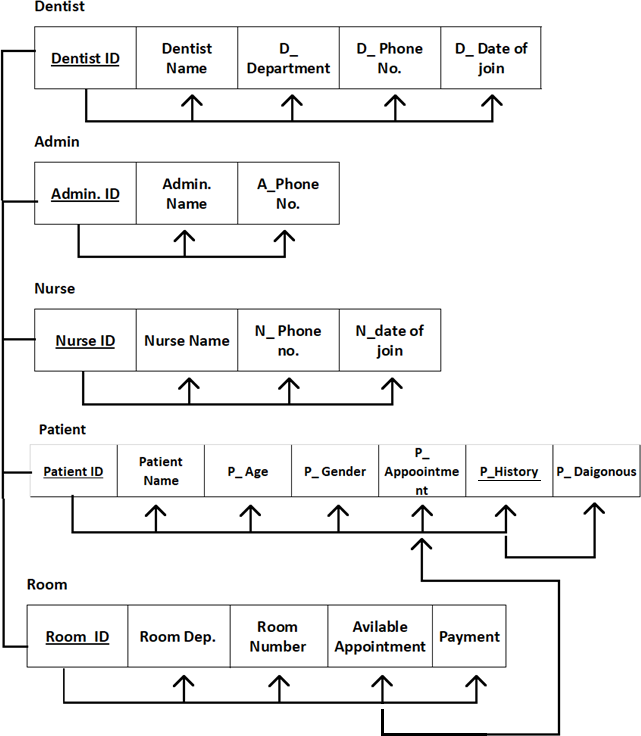
**Supervised by:**

**Dr. Mohamed Ibrahim**

**Eng. Mai Mohamed**

Spring 2025

**Database Schema (Before Normalization)**

****

**Normalization Steps**

1. **First Normal Form (1NF)**

it appears to contain atomic values in each cell, so there are no obvious repeating groups.

# Second Normal Form (2NF)

Problem in 2NF:

2NF requires that:

* + The table must first be in 1NF.
  + All non-key attributes must be fully functionally dependent on the entire primary key (not just part of it).

In our case, if we consider the table as a whole, we might be using a composite primary key (for example, a combination of Dentist ID, Nurse ID, Patient ID, and Room ID).

Partial dependencies in this table could be:

* + Columns such as Dentist Name, D\_Department, and D\_Phone No. depend only on Dentist ID.
  + Columns like Patient Name, P\_Age, and P\_History depend only on Patient ID.
  + Similarly, Room Dep. depends only on Room ID.

These columns do not depend on the full composite key but only on a part of it. This violates 2NF, as some attributes are not fully dependent on the entire primary key.

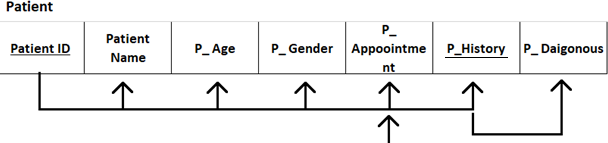
# Solution for 2NF:

To solve this, we need to separate the table into multiple tables, each focusing on a single entity and making sure each table has a primary key on which all non-key attributes are fully dependent.

After breaking down the table, we get the following separate tables:

1. Dentist Table:
   * Attributes: Dentist ID (Primary Key), Dentist Name, D\_Department, D\_Phone No., D\_Date of join
   * Purpose: Stores information specific to dentists.
2. Admin Table:
   * Attributes: Admin ID (Primary Key), Admin Name, A\_Phone No.
   * Purpose: Stores information specific to admins.
3. Nurse Table:
   * Attributes: Nurse ID (Primary Key), Nurse Name, N\_Phone no., N\_date of join
   * Purpose: Stores information specific to nurses.
4. Patient Table:
   * Attributes: Patient ID (Primary Key), Patient Name, P\_Age, P\_Gender, P\_History, P\_Diagnosis
   * Purpose: Stores information specific to patients.
5. Room Table:
   * Attributes: Room ID (Primary Key), Room Dep., Available Appointment
   * Purpose: Stores information specific to rooms.
6. Appointments Table:
   * Attributes: Appointment ID (Primary Key), Dentist ID (Foreign Key), Nurse ID (Foreign Key), Patient ID (Foreign Key), Room ID (Foreign Key), Date of Appointment
   * Purpose: Links appointments to specific dentists, nurses, patients, and rooms.

Each table now has a primary key with attributes that are fully functionally dependent on it, meaning that partial dependencies have been eliminated. We have achieved 2NF by separating the data into logically distinct entities.



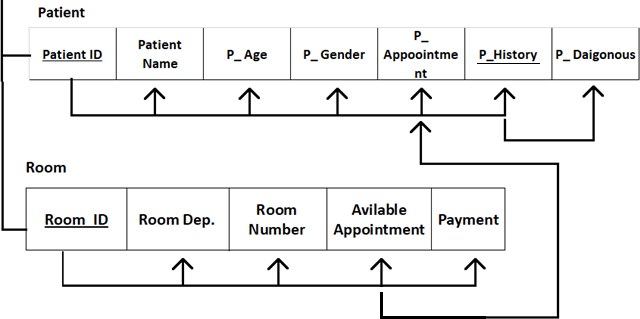
# Problem in 3NF:

3NF requires that:

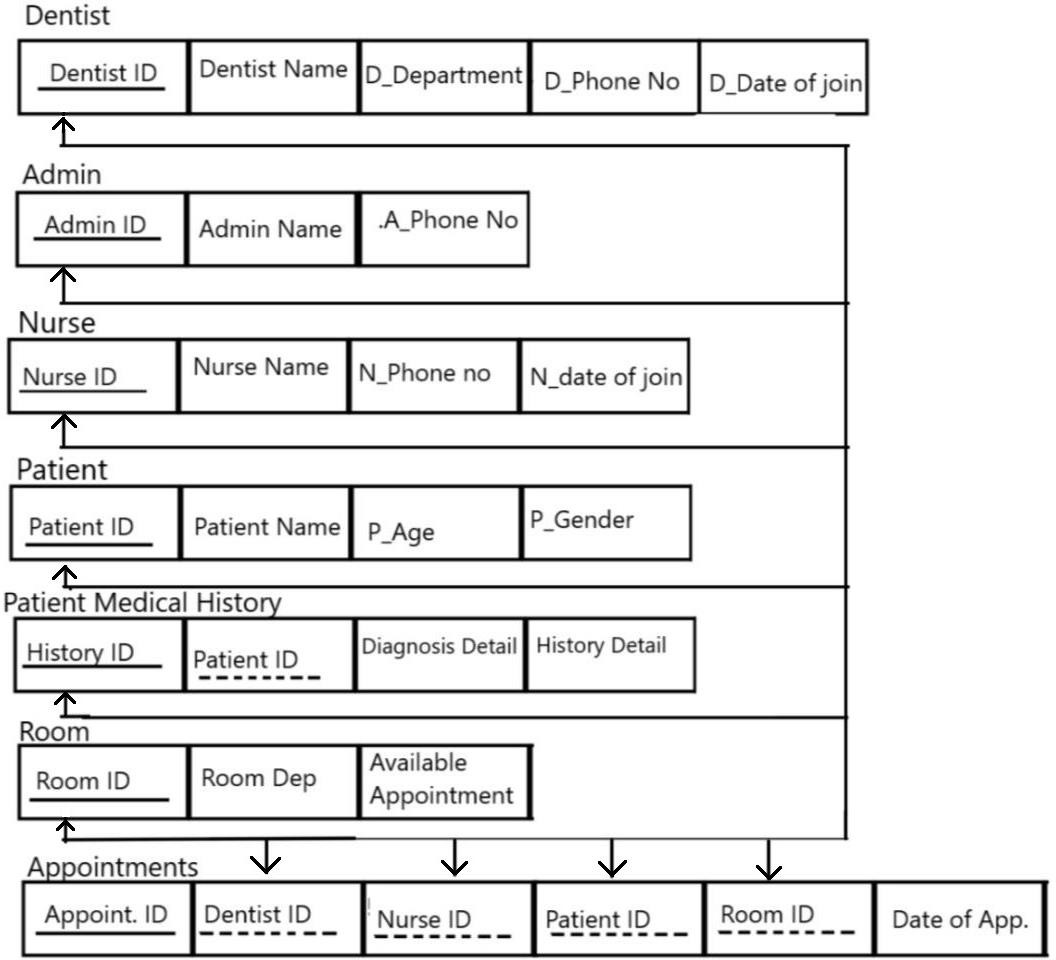
* + The table must first be in 2NF.
  + There should be no transitive dependencies. This means that non-key attributes should depend only on the primary key and not on other non-key attributes.

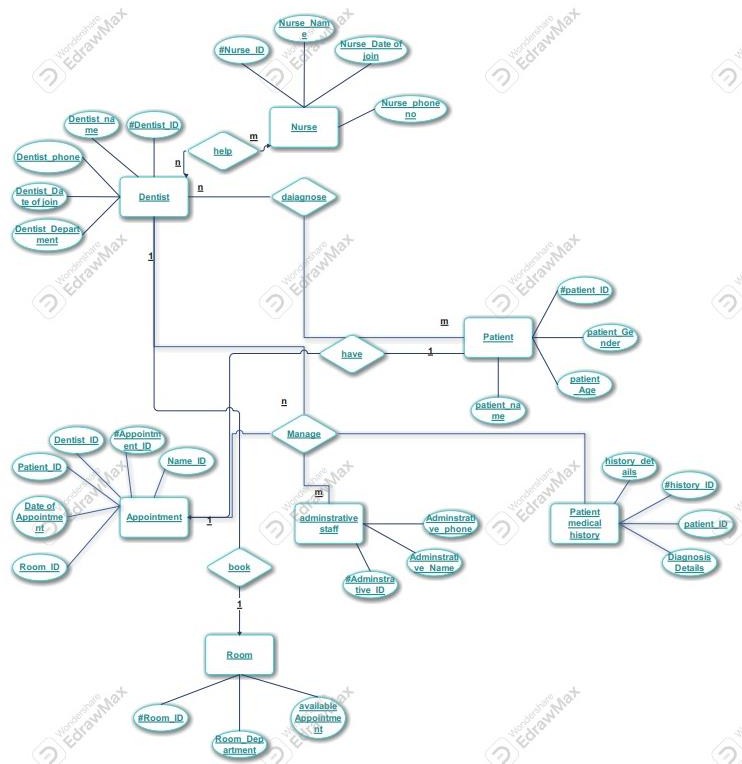
In some of the tables resulting from 2NF, there may still be transitive dependencies. For example:

* + In the Patient Table, if P\_History depends on Patient ID, but P\_Diagnosis is dependent on P\_History, then there is a transitive dependency. Here, P\_History indirectly depends on Patient ID through P\_Diagnosis.



**Database Schema (After Normalization)**

****

**ERD**

Database Creation

CREATE DATABASE *DentalClinic*;

Table Creation

# CREATE TABLE Room (

# Room\_ID INT PRIMARY KEY,

# Room\_Department VARCHAR(100) UNIQUE,

# Room\_Availability BIT

# );

# CREATE TABLE *Dentist* (

# Dentist\_ID INT *PRIMARY KEY*,

# Dentist\_Name VARCHAR(100),

# Dentist\_Phone VARCHAR(20),

# Dentist\_DateOfJoin DATE,

# Room\_Department VARCHAR(100),

# *FOREIGN KEY* (Room\_Department) *REFERENCES* Room(Room\_Department)

# );

# CREATE TABLE *Nurse* (

# Nurse\_ID INT *PRIMARY KEY*,

# Nurse\_Name VARCHAR(100),

# Nurse\_PhoneNo VARCHAR(20),

# Nurse\_DateOfJoin DATE

# );

# CREATE TABLE *Patient* (

# Patient\_ID INT *PRIMARY KEY*,

# Patient\_Name VARCHAR(100),

# Patient\_Gender CHAR(1),

# Patient\_Age INT

# );

# CREATE TABLE *Administrative\_Staff* (

# Administrative\_ID INT *PRIMARY KEY*,

# Administrative\_Name VARCHAR(100),

# Administrative\_Phone VARCHAR(20)

# );

# CREATE TABLE *Appointment* (

# Appointment\_ID INT *PRIMARY KEY*,

# Dentist\_ID INT,

# Patient\_ID INT,

# Patient\_Name VARCHAR(100),

# Room\_ID INT,

# DateOfAppointment DATE,

# *FOREIGN KEY* (Dentist\_ID) *REFERENCES* Dentist(Dentist\_ID),

# *FOREIGN KEY* (Patient\_ID) *REFERENCES* Patient(Patient\_ID),

# *FOREIGN KEY* (Room\_ID) *REFERENCES* Room(Room\_ID)

# );

# CREATE TABLE *Patient\_Medical\_History* (

# History\_ID INT *PRIMARY KEY*,

# Patient\_ID INT,

# History\_Details TEXT,

# Diagnosis\_Detail TEXT,

# *FOREIGN KEY* (Patient\_ID) *REFERENCES* Patient(Patient\_ID)

# );

Data Insertion:

INSERT INTO Room (Room\_ID, Room\_Department, Room\_Availability)

VALUES

(1, 'Orthodontics', 1),

(2, 'Surgery', 1),

(3, 'Pediatrics', 0),

(4, 'Endodontics', 1),

(5, 'Prosthodontics', 1),

(6, 'Implantology', 1),

(7, 'Radiology', 0),

(8, 'Oral Medicine', 1),

(9, 'General Dentistry', 1),

(10, 'Hygiene', 1),

(11, 'Diagnostics', 1),

(12, 'Lab', 1),

(13, 'Reception', 1),

(14, 'Consultation', 1),

(15, 'Surgical Recovery', 1),

(16, 'X-Ray', 0),

(17, 'Sterilization', 1),

(18, 'Emergency', 1),

(19, 'Waiting', 1),

(20, 'Periodontics', 1),

(21, 'Laser Treatment', 1),

(22, 'Oral Surgery', 0),

(23, 'Maxillofacial', 1

(24, 'Anesthesia', 1),

(25, 'Microbiology', 1);

INSERT INTO Dentist (Dentist\_ID, Dentist\_Name, Dentist\_Phone, Dentist\_DateOfJoin, Room\_Department)

VALUES

(1, 'Dr. Ahmed', '0100000001', '2021-01-10', 'Orthodontics'),

(2, 'Dr. Rana', '0100000002', '2021-02-11', 'Surgery'),

(3, 'Dr. Hossam', '0100000003', '2021-03-12', 'Pediatrics'),

(4, 'Dr. Mona', '0100000004', '2021-04-13', 'Endodontics'),

(5, 'Dr. Tamer', '0100000005', '2021-05-14', 'Prosthodontics'),

(6, 'Dr. Samar', '0100000006', '2021-06-15', 'Implantology'),

(7, 'Dr. Hany', '0100000007', '2021-07-16', 'Radiology'),

(8, 'Dr. Noha', '0100000008', '2021-08-17', 'Oral Medicine'),

(9, 'Dr. Fady', '0100000009', '2021-09-18', 'General Dentistry'),

(10, 'Dr. Asmaa', '0100000010', '2021-10-19', 'Hygiene'),

(11, 'Dr. Youssef', '0100000011', '2021-11-20', 'Diagnostics'),

(12, 'Dr. Farida', '0100000012', '2021-12-21', 'Lab'),

(13, 'Dr. Kareem', '0100000013', '2022-01-01', 'Reception'),

(14, 'Dr. Heba', '0100000014', '2022-01-05', 'Consultation'),

(15, 'Dr. Ibrahim', '0100000015', '2022-01-10', 'Surgical Recovery'),

(16, 'Dr. Dalia', '0100000016', '2022-01-15', 'X-Ray'),

(17, 'Dr. Walid', '0100000017', '2022-01-20', 'Sterilization'),

(18, 'Dr. Hagar', '0100000018', '2022-01-25', 'Emergency'),

(19, 'Dr. Ziad', '0100000019', '2022-01-30', 'Waiting'),

(20, 'Dr. Dina', '0100000020', '2022-02-04', 'Periodontics'),

(21, 'Dr. Omar', '0100000021', '2022-02-10', 'Laser Treatment'),

(22, 'Dr. Laila', '0100000022', '2022-02-15', 'Oral Surgery'),

(23, 'Dr. Adel', '0100000023', '2022-02-20', 'Maxillofacial'),

(24, 'Dr. Aya', '0100000024', '2022-02-25', 'Anesthesia'),

(25, 'Dr. Nour', '0100000025', '2022-03-01', 'Microbiology');

INSERT INTO Nurse (Nurse\_ID, Nurse\_Name, Nurse\_PhoneNo, Nurse\_DateOfJoin)

VALUES

(1, 'Nurse A', '0110000001', '2021-01-01'),

(2, 'Nurse B', '0110000002', '2021-01-02'),

(3, 'Nurse C', '0110000003', '2021-01-03'),

(4, 'Nurse D', '0110000004', '2021-01-04'),

(5, 'Nurse E', '0110000005', '2021-01-05'),

(6, 'Nurse F', '0110000006', '2021-01-06'),

(7, 'Nurse G', '0110000007', '2021-01-07'),

(8, 'Nurse H', '0110000008', '2021-01-08'),

(9, 'Nurse I', '0110000009', '2021-01-09'),

(10, 'Nurse J', '0110000010', '2021-01-10'),

(11, 'Nurse K', '0110000011', '2021-01-11'),

(12, 'Nurse L', '0110000012', '2021-01-12'),

(13, 'Nurse M', '0110000013', '2021-01-13'),

(14, 'Nurse N', '0110000014', '2021-01-14'),

(15, 'Nurse O', '0110000015', '2021-01-15'),

(16, 'Nurse P', '0110000016', '2021-01-16'),

(17, 'Nurse Q', '0110000017', '2021-01-17'),

(18, 'Nurse R', '0110000018', '2021-01-18'),

(19, 'Nurse S', '0110000019', '2021-01-19'),

(20, 'Nurse T', '0110000020', '2021-01-20'),

(21, 'Nurse U', '0110000021', '2021-01-21'),

(22, 'Nurse V', '0110000022', '2021-01-22'),

(23, 'Nurse W', '0110000023', '2021-01-23'),

(24, 'Nurse X', '0110000024', '2021-01-24'),

(25, 'Nurse Y', '0110000025', '2021-01-25');

INSERT INTO Administrative\_Staff (Administrative\_ID, Administrative\_Name, Administrative\_Phone)

VALUES

(1, 'Admin A', '0120000001'),

(2, 'Admin B', '0120000002'),

(3, 'Admin C', '0120000003'),

(4, 'Admin D', '0120000004'),

(5, 'Admin E', '0120000005'),

(6, 'Admin F', '0120000006'),

(7, 'Admin G', '0120000007'),

(8, 'Admin H', '0120000008'),

(9, 'Admin I', '0120000009'),

INSERT INTO Patient (Patient\_ID, Patient\_Name, Patient\_Gender, Patient\_Age)

VALUES

(1, 'Ali Hassan', 'M', 25),

(2, 'Sara Kamal', 'F', 30),

(3, 'Omar Tarek', 'M', 19),

(4, 'Nour Khaled', 'F', 23),

(5, 'Mostafa Ahmed', 'M', 35),

(6, 'Lina Mahmoud', 'F', 28),

(7, 'Tarek Hany', 'M', 40),

(8, 'Aya Ehab', 'F', 22),

(9, 'Mahmoud Zaki', 'M', 33),

(10, 'Reem Samir', 'F', 26),

(11, 'Youssef Ali', 'M', 21),

(12, 'Hana Wael', 'F', 29),

(13, 'Adel Fathy', 'M', 37),

(14, 'Dina Emad', 'F', 24),

(15, 'Amr Said', 'M', 31),

(16, 'Sally Adel', 'F', 27),

(17, 'Ibrahim Magdy', 'M', 45),

(18, 'Mariam Hossam', 'F', 20),

(19, 'Ziad Mostafa', 'M', 38),

(20, 'Fatma Sherif', 'F', 32),

(21, 'Khaled Tamer', 'M', 34),

(22, 'Yasmin Omar', 'F', 25),

(23, 'Hazem Yehia', 'M', 36),

(24, 'Layla Amr', 'F', 30),

(25, 'Hatem Osama', 'M', 39);

INSERT INTO Appointment (Appointment\_ID, Dentist\_ID, Patient\_ID, Patient\_Name, Room\_ID, DateOfAppointment)

VALUES

(1, 1, 1, 'Ali Hassan', 1, '2025-05-01'),

(2, 2, 2, 'Sara Kamal', 2, '2025-05-02'),

(3, 3, 3, 'Omar Tarek', 3, '2025-05-03'),

(4, 4, 4, 'Nour Khaled', 4, '2025-05-04'),

(5, 5, 5, 'Mostafa Ahmed', 5, '2025-05-05'),

(6, 6, 6, 'Lina Mahmoud', 6, '2025-05-06'),

(7, 7, 7, 'Tarek Hany', 7, '2025-05-07'),

(8, 8, 8, 'Aya Ehab', 8, '2025-05-08'),

(9, 9, 9, 'Mahmoud Zaki', 9, '2025-05-09'),

(10, 10, 10, 'Reem Samir', 10, '2025-05-10'),

(11, 11, 11, 'Youssef Ali', 11, '2025-05-11'),

(12, 12, 12, 'Hana Wael', 12, '2025-05-12'),

(13, 13, 13, 'Adel Fathy', 13, '2025-05-13'),

(14, 14, 14, 'Dina Emad', 14, '2025-05-14'),

(15, 15, 15, 'Amr Said', 15, '2025-05-15'),

(16, 16, 16, 'Sally Adel', 16, '2025-05-16'),

(17, 17, 17, 'Ibrahim Magdy', 17, '2025-05-17'),

(18, 18, 18, 'Mariam Hossam', 18, '2025-05-18'),

(19, 19, 19, 'Ziad Mostafa', 19, '2025-05-19'),

(20, 20, 20, 'Fatma Sherif', 20, '2025-05-20'),

(21, 21, 21, 'Khaled Tamer', 21, '2025-05-21'),

(22, 22, 22, 'Yasmin Omar', 22, '2025-05-22'),

(23, 23, 23, 'Hazem Yehia', 23, '2025-05-23'),

(24, 24, 24, 'Layla Amr', 24, '2025-05-24'),

(25, 25, 25, 'Hatem Osama', 25, '2025-05-25');

INSERT INTO Patient\_Medical\_History (History\_ID, Patient\_ID, History\_Details, Diagnosis\_Detail)

VALUES

(1, 1, 'Cavity history', 'Tooth Decay'),

(2, 2, 'Orthodontic history', 'Braces'),

(3, 3, 'Bleeding gums', 'Gingivitis'),

(4, 4, 'Tooth pain', 'Cavity'),

(5, 5, 'Surgery in 2020', 'Root Canal'),

(6, 6, 'Tooth fracture', 'Crown'),

(7, 7, 'No history', 'Normal'),

(8, 8, 'Swollen gums', 'Gum infection'),

(9, 9, 'Tooth loss', 'Implant'),

(10, 10, 'Tooth sensitivity', 'Whitening'),

(11, 11, 'Jaw pain', 'TMJ'),

(12, 12, 'Plaque buildup', 'Cleaning'),

(13, 13, 'Dry mouth', 'Medication side effect'),

(14, 14, 'Mouth ulcer', 'Viral infection'),

(15, 15, 'Bad breath', 'Bacterial infection'),

(16, 16, 'Stained teeth', 'Smoking'),

(17, 17, 'Loose tooth', 'Gum disease'),

(18, 18, 'Gum recession', 'Brushing issue'),

(19, 19, 'Tooth grinding', 'Night guard'),

(20, 20, 'Overbite', 'Orthodontics'),

(21, 21, 'Wisdom teeth issue', 'Extraction'),

(22, 22, 'Broken crown', 'Replacement'),

(23, 23, 'Filling fell out', 'New filling'),

(24, 24, 'Mouth dryness', 'Hydration needed'),

(25, 25, 'Bleeding while brushing', 'Gum sensitivity');

Stored Procedures

-First Type:

CREATE PROCEDURE GetAllDentists

AS

BEGIN

    SELECT \* FROM Dentist;

END;

EXEC GetAllDentists;

Output:

**A screenshot of a computer

AI-generated content may be incorrect.**

# -Second Type:

CREATE PROCEDURE GetAppointmentsByDentistID

 @DentistID INT

AS

BEGIN

    SELECT

        Appointment\_ID,

        Dentist\_ID,

        Patient\_ID,

        DateOfAppointment

    FROM Appointment

    WHERE Dentist\_ID = @DentistID;

END;

EXEC GetAppointmentsByDentistID 4;

Output:

**A screenshot of a computer

AI-generated content may be incorrect.**

# -Third Type:

CREATE PROCEDURE GetPatientInfo

 @minID INT,

 @maxID INT

AS

BEGIN

    SELECT Patient\_ID, Patient\_Name, Patient\_Gender, Patient\_Age

    FROM Patient

    WHERE Patient\_ID BETWEEN @minID AND @maxID

    ORDER BY Patient\_ID;

END;

EXEC GetPatientInfo 1, 11;

Output:

**A screenshot of a computer

AI-generated content may be incorrect.**

# -Fourth Type:

CREATE PROCEDURE DentalClinicProcedure

(

    @PatientCount INT OUTPUT

)

AS

BEGIN

    SELECT Patient\_ID, Patient\_Name FROM Patient;

    SELECT @PatientCount = COUNT(\*) FROM Patient;

END;

DECLARE @Count INT;

EXEC DentalClinicProcedure @PatientCount = @Count OUTPUT;

SELECT @Count AS 'Number of Patients';

A screenshot of a computer

AI-generated content may be incorrect. Output

Cursors

DECLARE @PatientName VARCHAR(50), @PatientAge INT;

DECLARE PatientCursor CURSOR FOR

SELECT

    Patient\_Name,

    Patient\_Age

FROM

    Patient;

OPEN PatientCursor;

FETCH NEXT FROM PatientCursor INTO @PatientName, @PatientAge;

WHILE @@FETCH\_STATUS = 0

BEGIN

    PRINT 'Patient Name: ' + @PatientName + ', Age: ' + CAST(@PatientAge AS VARCHAR);

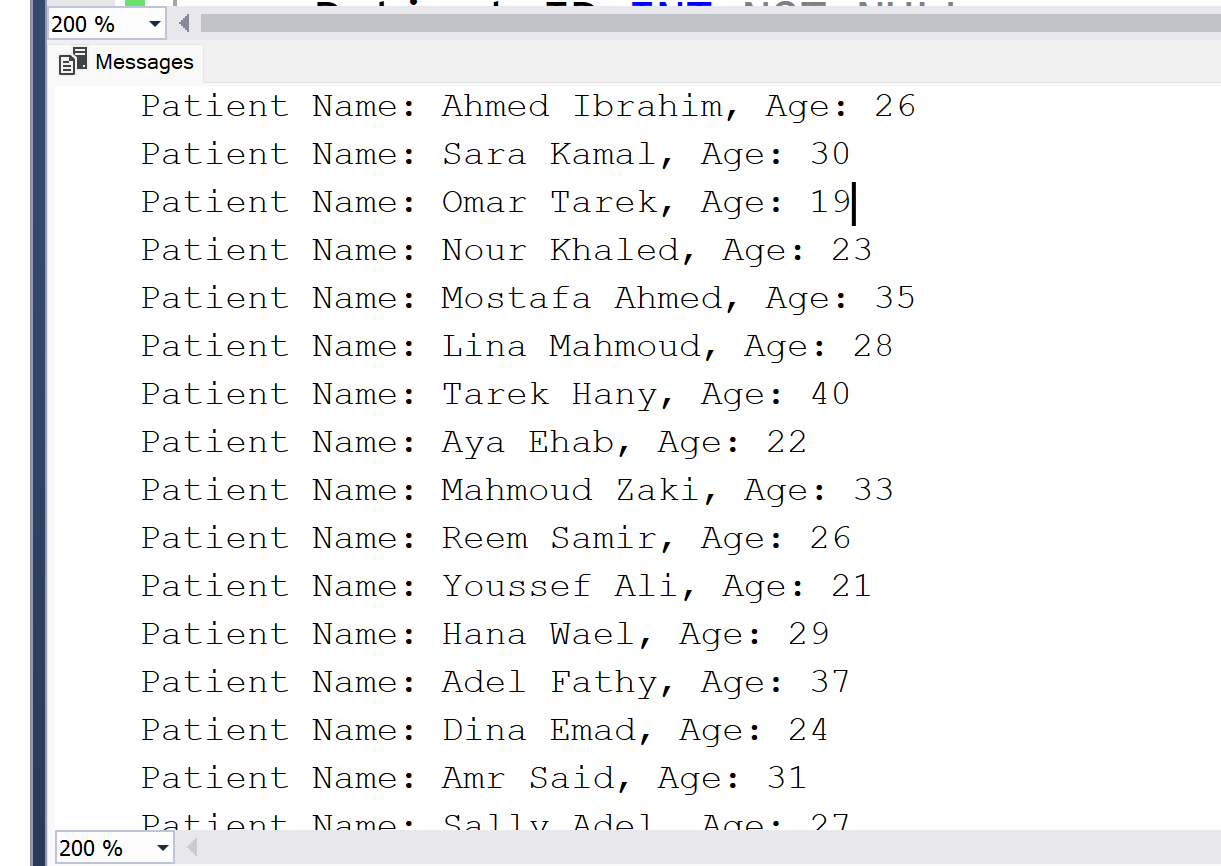
    FETCH NEXT FROM PatientCursor INTO @PatientName, @PatientAge;

END;

CLOSE PatientCursor;

DEALLOCATE PatientCursor;

Output:

******

Triggers

CREATE TABLE *DentalClinicAudits* (

    Audit\_ID INT IDENTITY(1,1) *PRIMARY KEY*,

    Patient\_ID INT NOT NULL,

    Patient\_Name VARCHAR(100) NOT NULL,

    Patient\_Gender CHAR(1) NOT NULL,

    Patient\_Age INT NOT NULL,

    Operation\_Type VARCHAR(3) NOT NULL,

*CHECK* (Operation\_Type IN ('INS', 'DEL', 'UPD'))

);

DROP TRIGGER IF EXISTS trg\_patient\_audit;

GO

CREATE TRIGGER *trg\_patient\_audit*

ON Patient

AFTER INSERT, UPDATE

AS

BEGIN

    SET NOCOUNT ON;

    INSERT INTO DentalClinicAudits (Patient\_ID, Patient\_Name, Patient\_Gender, Patient\_Age, Operation\_Type)

    SELECT i.Patient\_ID, i.Patient\_Name, i.Patient\_Gender, i.Patient\_Age, 'INS'

    FROM inserted i

    LEFT JOIN deleted d ON i.Patient\_ID = d.Patient\_ID

    WHERE d.Patient\_ID IS NULL;

    INSERT INTO DentalClinicAudits (Patient\_ID, Patient\_Name, Patient\_Gender, Patient\_Age, Operation\_Type)

    SELECT i.Patient\_ID, i.Patient\_Name, i.Patient\_Gender, i.Patient\_Age, 'UPD'

    FROM inserted i

    JOIN deleted d ON i.Patient\_ID = d.Patient\_ID;

END;

GO

SELECT \* FROM DentalClinicAudits;

Output:

***A screenshot of a computer

AI-generated content may be incorrect.***

Functions

# CREATE FUNCTION *dbo*.CalculateDentalCost

# (

# @Patient\_ID INT,

# @Patient\_Name VARCHAR(100),

# @TreatmentType NVARCHAR(50),

# @Session INT

# )

# RETURNS DECIMAL(10, 2)

# AS

# BEGIN

# DECLARE @CostPerSession DECIMAL(10, 2) = 0.00;

# DECLARE @VerifiedPatientName VARCHAR(100);

# 

# SELECT @VerifiedPatientName = Patient\_Name

# FROM Appointment

# WHERE Patient\_ID = @Patient\_ID

# AND LOWER(TRIM(Patient\_Name)) = LOWER(TRIM(@Patient\_Name));

# 

# IF @VerifiedPatientName IS NULL

# BEGIN

# RETURN NULL;

# END;

# DECLARE @CleanTreatmentType NVARCHAR(50) = UPPER(LTRIM(RTRIM(@TreatmentType)));

# 

# IF @CleanTreatmentType = 'CLEANING'

# SET @CostPerSession = 400.00;

# ELSE IF @CleanTreatmentType = 'NERVE FILLER'

# SET @CostPerSession = 800.00;

# ELSE IF @CleanTreatmentType = 'FIXED'

# SET @CostPerSession = 3800.00;

# ELSE

# RETURN NULL;

# RETURN @CostPerSession \* @Session;

# END;

# SELECT dbo.CalculateDentalCost(1, 'Ali Hassan', 'CLEANING', 3) AS Total\_Cost;

# Output:

# A screenshot of a computer AI-generated content may be incorrect.

View

CREATE VIEW *vw\_AppointmentFullDetails* AS

SELECT

    a.Appointment\_ID,

    a.DateOfAppointment,

    a.Room\_ID,

    p.Patient\_ID,

    p.Patient\_Name,

    p.Patient\_Gender,

    p.Patient\_Age,

    d.Dentist\_ID,

    d.Dentist\_Name,

    d.Dentist\_Phone

FROM

    Appointment a

INNER JOIN Patient p ON a.Patient\_ID = p.Patient\_ID

INNER JOIN Dentist d ON a.Dentist\_ID = d.Dentist\_ID;

SELECT \* FROM vw\_AppointmentFullDetails;

Output:

***A screenshot of a computer

AI-generated content may be incorrect.***

*Index*

CREATE INDEX idx\_Dentist

ON Dentist (Dentist\_ID, Dentist\_Phone, Dentist\_DateOfJoin);

CREATE INDEX idx\_Administrative\_Staff

ON Administrative\_Staff (Administrative\_ID, Administrative\_Name, Administrative\_Phone);

CREATE INDEX idx\_Appointment

ON Appointment (Appointment\_ID, Dentist\_ID, Patient\_ID, Room\_ID, DateOfAppointment);

CREATE INDEX idx\_Nurse

ON Nurse (Nurse\_ID, Nurse\_Name, Nurse\_PhoneNo, Nurse\_DateOfJoin);

CREATE INDEX idx\_Patient

ON Patient (Patient\_ID, Patient\_Name, Patient\_Gender, Patient\_Age);

CREATE INDEX idx\_Patient\_Medical\_History

ON Patient\_Medical\_History (History\_ID, Patient\_ID);

CREATE INDEX idx\_Room

ON Room (Room\_ID, Room\_Department, Room\_Availability);

Basic sql

ALTER TABLE Patient

ADD Email VARCHAR(100);

ALTER TABLE Patient

DROP COLUMN Email;

DROP TABLE Nurse;

DELETE FROM Patient

WHERE Patient\_Age < 20;

UPDATE Patient

SET Patient\_Name = 'Ahmed Ali', Patient\_Age = 35

WHERE Patient\_ID = 1;

SELECT \* FROM Patient;

SELECT \* FROM Patient

WHERE Patient\_Age > 30 AND Patient\_Gender = 'M';

SELECT \* FROM Patient

WHERE Patient\_Age < 25 OR Patient\_Gender = 'F';

SELECT \* FROM Patient

WHERE Patient\_Age BETWEEN 25 AND 40;

SELECT \* FROM Patient

WHERE Patient\_ID IN (1, 2, 5);

SELECT \* FROM Patient

WHERE Patient\_Name LIKE 'A%';