DBMS Practical

Q-1) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

SUPPLIER (Sno, Sname, address,City) PARTS (Pno, Pname, Color, Weight, price) PROJECT (Jno, Mame, City) SPJ (in=q,En, JUQ, Qty)

Integrity Constraints:

- The values of any attributes should not be null.
- Legal cities are London, Paris, Rome, New York and Amsterdam.
- Supplier Number must start with 'S' followed by a decimal integer in the range of 0 to 9999.

Queries:

a) Find all the projects which are provided 3 or more parts .

b) Write a trigger on PROJECT table for update / insert such that the .jname value Should not be repeated.

c) Find full details of all projects in London.

d) Write a procedure for calculating the total sales of all the parts which are provided to projects in paris city.

Design an Input form for entering Parts data. Apply possible validations.

->

1. TABLE CREATION IN SQL SERVER

```
-- SUPPLIER table
CREATE TABLE SUPPLIER (
    Sno VARCHAR(6) PRIMARY KEY CHECK (Sno LIKE 'S%' AND TRY_CAST(SUBSTRING(Sno, 2, LEN(Sno)) AS INT) BETWEEN 0 AND 9999),
    Sname VARCHAR(50) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL CHECK (City IN ('London', 'Paris', 'Rome', 'New York', 'Amsterdam'))
);

-- PARTS table
CREATE TABLE PARTS (
    Pno VARCHAR(6) PRIMARY KEY,
    Pname VARCHAR(50) NOT NULL,
    Color VARCHAR(20) NOT NULL,
    Weight DECIMAL(5,2) NOT NULL,
    Price DECIMAL(10,2) NOT NULL
);

-- PROJECT table
```

```
CREATE TABLE PROJECT (
    Jno VARCHAR(6) PRIMARY KEY,
    Jname VARCHAR(50) NOT NULL UNIQUE,
    City VARCHAR(50) NOT NULL CHECK (City IN ('London', 'Paris', 'Rome', 'New York', 'Amsterdam'))
);


-- SPJ table
CREATE TABLE SPJ (
    Sno VARCHAR(6) NOT NULL FOREIGN KEY REFERENCES SUPPLIER(Sno),
    Pno VARCHAR(6) NOT NULL FOREIGN KEY REFERENCES PARTS(Pno),
    Jno VARCHAR(6) NOT NULL FOREIGN KEY REFERENCES PROJECT(Jno),
    Qty INT NOT NULL,
    PRIMARY KEY (Sno, Pno, Jno)
);
```

2. INSERTING SAMPLE DATA

```
-- SUPPLIER
INSERT INTO SUPPLIER VALUES
('S101', 'Alpha Ltd', '12 King St', 'London'),
('S102', 'Bravo Inc', '5 Rue de Lyon', 'Paris'),
('S103', 'Charlie LLC', '88 Roma Ave', 'Rome'),
```

('S104', 'Delta Corp', '123 Broadway', 'New York'),
('S105', 'Echo GmbH', '78 Amstel', 'Amsterdam'),
('S106', 'Foxtrot BV', '10 Canal Rd', 'Amsterdam'),
('S107', 'Golf Ltd', '456 Wall St', 'New York'),
('S108', 'Hotel PLC', '3 Champs Elysees', 'Paris'),
('S109', 'India Co', '7 Via Appia', 'Rome'),
('S110', 'Juliet Ltd', '1 Tower Bridge', 'London');

-- PARTS
INSERT INTO PARTS VALUES
('P1', 'Bolt', 'Red', 0.50, 1.00),
('P2', 'Nut', 'Blue', 0.30, 0.50),
('P3', 'Screw', 'Green', 0.40, 0.75),
('P4', 'Washer', 'Red', 0.10, 0.25),
('P5', 'Gear', 'Black', 2.00, 5.00),
('P6', 'Wheel', 'Black', 4.50, 15.00),
('P7', 'Spring', 'Silver', 1.20, 2.50),
('P8', 'Chain', 'Gray', 3.00, 6.00),
('P9', 'Lever', 'Yellow', 1.50, 3.00),
('P10', 'Axle', 'Blue', 3.50, 7.50);

-- PROJECT
INSERT INTO PROJECT VALUES
('J1', 'Apollo', 'London'),
('J2', 'Zeus', 'Paris'),
('J3', 'Hermes', 'Rome'),
('J4', 'Athena', 'New York'),
('J5', 'Poseidon', 'Amsterdam'),
('J6', 'Hera', 'London'),

```
('J7', 'Artemis', 'Paris'),
('J8', 'Ares', 'Rome'),
('J9', 'Dionysus', 'New York'),
('J10', 'Hephaestus', 'Amsterdam');

-- SPJ
INSERT INTO SPJ VALUES
('S101', 'P1', 'J1', 100),
('S101', 'P2', 'J1', 200),
('S101', 'P3', 'J1', 220),
('S102', 'P3', 'J2', 150),
('S102', 'P4', 'J2', 250),
('S103', 'P5', 'J2', 190),
('S104', 'P6', 'J4', 100),
('S105', 'P7', 'J5', 200),
('S106', 'P8', 'J6', 150),
('S107', 'P9', 'J7', 200),
('S108', 'P10', 'J8', 180),
('S109', 'P1', 'J9', 160),
('S110', 'P2', 'J10', 140);
```

3. QUERIES

a) Projects that have 3 or more parts

```
SELECT Jno, COUNT(DISTINCT Pno) AS PartCount
FROM SPJ
GROUP BY Jno
HAVING COUNT(DISTINCT Pno) >= 3;
```

b) Full details of projects in London

```
SELECT * FROM PROJECT WHERE City = 'London';
```

c) Procedure: Total Sales of Parts in Projects in Paris

```
CREATE PROCEDURE GetTotalSalesParis
AS
BEGIN
   DECLARE @total DECIMAL(12,2);

   SELECT @total = SUM(spj.Qty * p.Price)
   FROM SPJ spj
   JOIN PARTS p ON spj.Pno = p.Pno
   JOIN PROJECT j ON spj.Jno = j.Jno
   WHERE j.City = 'Paris';

   PRINT 'Total Sales for Paris Projects: $' + CAST(@total
AS VARCHAR);
END;
```

Q-2) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). PRODUCT (Maker, Modelno, Type) PC (Modelno,,Speed, RAM, HD, CD, Price) LAPTOP (Modelno, Speed, RAM, HD, Price) PRINTER (Mg_delaQ, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.

• The value for Maker in Product table can be IBM, Compaq,etc.

• PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black 8r. white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null.

• Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find PC models having a speed of at least 150 MHz.

b) Find those manufacturers that sell Laptops, but not PC's.

c) Write a trigger on LAPTOP table such that the price should not less than 30000

d) Write a procedure to find the manufacturer who has produced the most expensive laptop.

Design an input form for entering LAPTOP data. Apply possible validations.

->

1. **Create Tables with Constraints – SQL Server Syntax**

```
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

CREATE TABLE PC (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    CD VARCHAR(10) NOT NULL,
    Price INT NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE LAPTOP (
```

```
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price INT NOT NULL CHECK (Price >= 30000),
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price INT NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno
));
```

2. **Insert Sample Records**

```
-- PRODUCT
INSERT INTO PRODUCT VALUES
('IBM', 1001, 'PC'), ('IBM', 1002, 'Laptop'), ('Compaq', 1003, 'PC'),
('HP', 1004, 'Printer'), ('Dell', 1005, 'Laptop'), ('Lenovo', 1006, 'PC'),
```

('HP', 1007, 'Laptop'), ('Canon', 1008, 'Printer'), ('Asus', 1009, 'Laptop'),

('Epson', 1010, 'Printer'), ('Dell', 1011, 'PC'), ('Asus', 1012, 'Laptop');

-- PC

INSERT INTO PC VALUES

(1001, 200, 8, 500, '52X', 40000),

(1003, 150, 4, 320, '40X', 35000),

(1006, 180, 8, 256, '48X', 37000),

(1011, 170, 4, 500, '52X', 33000);

-- LAPTOP

INSERT INTO LAPTOP VALUES

(1002, 250, 8, 512, 45000),

(1005, 220, 4, 256, 40000),

(1007, 180, 4, 320, 35000),

(1009, 240, 8, 500, 48000),

(1012, 200, 8, 512, 30000);

-- PRINTER

INSERT INTO PRINTER VALUES

```
(1004, 'T', 'laser', 25000),
(1008, 'F', 'dot-matrix', 15000),
(1010, 'T', 'ink-jet', 20000);
```

3. Queries

a) PC models with speed >= 150 MHz

```
SELECT * FROM PC
WHERE Speed >= 150;
```

b) Manufacturers that make Laptops but not PCs

```
SELECT DISTINCT p.Maker
FROM PRODUCT p
WHERE p.Type = 'Laptop'
AND p.Maker NOT IN (
   SELECT DISTINCT Maker
   FROM PRODUCT
   WHERE Type = 'PC'
);
```

4. Stored Procedure: Manufacturer of Most Expensive Laptop

```
CREATE PROCEDURE
Get_Most_Expensive_Laptop_Maker
AS
BEGIN
   DECLARE @Modelno INT;
   DECLARE @Maker VARCHAR(50);

   SELECT TOP 1 @Modelno = Modelno
   FROM LAPTOP
```

```
    ORDER BY Price DESC;

    SELECT @Maker = Maker
    FROM PRODUCT
    WHERE Modelno = @Modelno;

    PRINT 'Maker of most expensive laptop: ' + @Maker;
END;
```

Q-3) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table).

'PRODUCT (Maker, Modelno, Type) PC (Modelno, Speed, RAM, HD, CD, Price) LAPTOP (Modelno, Speed, RAM, HD, Price) PRINTER (Modelno, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price.

• The value for Maker in Product table can be IBM, Compaq,etc.

• PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null.

• Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find the different types of printers produced by Epson. b) Find those hard disk sizes which occur in two or more PC's. c) Write a trigger on LAPTOP table such that the minimum speed should be 1201\4Hz. d) Demonstrate the use of cursor using PRODUCT table. Design an input form for entering LAPTOP data. Apply possible validations.

->

1. Create Tables with Integrity Constraints in SQL Server

```
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY,
    Type VARCHAR(10) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

CREATE TABLE PC (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    CD VARCHAR(10) NOT NULL,
    Price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE LAPTOP (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
```

```
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

2. Insert Sample Data (Product + 3 other tables)

```
-- PRODUCT
INSERT INTO PRODUCT VALUES
('IBM', 1001, 'PC'),
('Compaq', 1002, 'PC'),
('Dell', 1003, 'PC'),
('HP', 1004, 'Laptop'),
('Compaq', 1005, 'Laptop'),
```

```
('IBM', 1006, 'Laptop'),
('Epson', 1007, 'Printer'),
('Epson', 1008, 'Printer'),
('HP', 1009, 'Printer'),
('Dell', 1010, 'Printer');

-- PC
INSERT INTO PC VALUES
(1001, 2400, 512, 80, '52x', 500.00),
(1002, 3000, 1024, 160, '48x', 600.00),
(1003, 2200, 2048, 250, '52x', 700.00);

-- LAPTOP
INSERT INTO LAPTOP VALUES
(1004, 2500, 4096, 500, 1000.00),
(1005, 3200, 8192, 1000, 1500.00),
(1006, 2400, 2048, 256, 900.00);

-- PRINTER
INSERT INTO PRINTER VALUES
(1007, 'T', 'ink-jet', 120.00),
(1008, 'F', 'dot-matrix', 80.00),
```

(1009, 'T', 'laser', 200.00), SELECT HD

FROM PC

GROUP BY HD

HAVING COUNT(*) >= 2; (1010, 'F', 'dry', 100.00);

3. SQL Server Queries

a) Different types of printers produced by Epson

```
SELECT DISTINCT p.Type
FROM PRODUCT pr
JOIN PRINTER p ON pr.Modelno = p.Modelno
WHERE pr.Maker = 'Epson';
```

b) Hard disk sizes that occur in two or more PCs

```
SELECT HD
FROM PC
GROUP BY HD
HAVING COUNT(*) >= 2;
```

Q-4) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification) 1AT1ENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Pid) ADMITTEDPATIENT (P_code, Entry date, Discharge date, wardno, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be M male) or F(female). • Wardno should be less than 6.

Queries:

a) Find the details of doctors who are treating the patient of ward no 3. b) Write a trigger on PATIENTMASTER table such that the blood group should be A,B,AB or 0. c) Find the details of patient who are discharge within the period 03/03/12 to 25/ 03/12 d) Write a procedure on ADMIFTEDPATIENT table such as to calculate bill of all discharged patients. The charges per day for a ward is WardNo. * 100. e.g. For ward no 3 charges/day are 300Rs.

Create a data report for the details of Doctors.

Report should also include the details of patients treated by that doctor.

->

1. Create Tables with Constraints

```
-- Create the Database
CREATE DATABASE HospitalDB;
GO

USE HospitalDB;
GO

-- DOCTOR Table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY,
    Dname VARCHAR(50) NOT NULL,
    Daddress VARCHAR(100) NOT NULL,
    qualification VARCHAR(50) NOT NULL
);

-- PATIENTMASTER Table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY,
    Pname VARCHAR(50) NOT NULL,
    Padd VARCHAR(100) NOT NULL,
```

```
    age INT NOT NULL,
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F')),
    bloodgroup VARCHAR(3) NOT NULL,
    Pid INT NOT NULL
);

-- ADMITTEDPATIENT Table
CREATE TABLE ADMITTEDPATIENT (
    P_code INT NOT NULL FOREIGN KEY REFERENCES
PATIENTMASTER(Pcode),
    EntryDate DATE NOT NULL,
    DischargeDate DATE NOT NULL,
    wardno INT NOT NULL CHECK (wardno < 6),
    disease VARCHAR(100) NOT NULL,
    Did INT NOT NULL FOREIGN KEY REFERENCES DOCTOR(Did)
);
```

2. Insert Sample Data (10+ Records Each)

```
-- Insert into DOCTOR
INSERT INTO DOCTOR VALUES
(1, 'Dr. A Sharma', 'Delhi', 'MD'),
(2, 'Dr. B Kumar', 'Mumbai', 'MBBS'),
(3, 'Dr. C Rao', 'Chennai', 'MD'),
```

```
(4, 'Dr. D Iyer', 'Bangalore', 'MS'),
(5, 'Dr. E Khan', 'Hyderabad', 'MBBS'),
(6, 'Dr. F Mehta', 'Pune', 'MS'),
(7, 'Dr. G Nair', 'Kolkata', 'MD'),
(8, 'Dr. H Patel', 'Ahmedabad', 'MBBS'),
(9, 'Dr. I Yadav', 'Lucknow', 'MS'),
(10, 'Dr. J Reddy', 'Vizag', 'MD');

-- Insert into PATIENTMASTER
INSERT INTO PATIENTMASTER VALUES
(101, 'Ravi Verma', 'Delhi', 30, 'M', 'A', 1),
(102, 'Neha Singh', 'Mumbai', 25, 'F', 'B', 2),
(103, 'Karan Das', 'Chennai', 40, 'M', 'AB', 3),
(104, 'Priya Nair', 'Kolkata', 28, 'F', 'O', 4),
(105, 'Anil Gupta', 'Pune', 35, 'M', 'A', 5),
(106, 'Sonal Joshi', 'Ahmedabad', 23, 'F', 'B', 6),
(107, 'Rohit Rao', 'Bangalore', 38, 'M', 'O', 7),
(108, 'Meena Shah', 'Hyderabad', 31, 'F', 'AB', 8),
(109, 'Deepak Yadav', 'Lucknow', 29, 'M', 'B', 9),
(110, 'Divya Jain', 'Delhi', 27, 'F', 'A', 10);

-- Insert into ADMITTEDPATIENT
```

```
INSERT INTO ADMITTEDPATIENT VALUES
(101, '2012-03-01', '2012-03-10', 3, 'Fever', 1),
(102, '2012-03-05', '2012-03-15', 2, 'Malaria', 2),
(103, '2012-03-10', '2012-03-22', 3, 'Typhoid', 3),
(104, '2012-03-12', '2012-03-25', 1, 'Injury', 4),
(105, '2012-02-20', '2012-03-03', 4, 'Diabetes', 5),
(106, '2012-03-14', '2012-03-20', 5, 'Flu', 6),
(107, '2012-03-01', '2012-03-18', 3, 'Fever', 7),
(108, '2012-03-02', '2012-03-21', 1, 'Cold', 8),
(109, '2012-03-16', '2012-03-24', 2, 'Fever', 9),
(110, '2012-03-08', '2012-03-20', 3, 'Typhoid', 10);
```

3. Required SQL Queries

a) Doctors treating patients in ward no. 3

```
SELECT DISTINCT D.*
FROM DOCTOR D
JOIN ADMITTEDPATIENT A ON D.Did = A.Did
WHERE A.wardno = 3;
```

b) Patients discharged between 03/03/12 and 25/03/12

```
SELECT P.*
FROM PATIENTMASTER P
JOIN ADMITTEDPATIENT A ON P.Pcode = A.P_code
```

```
WHERE A.DischargeDate BETWEEN '2012-03-03' AND
'2012-03-25';
```

c) Procedure to calculate bill

```
CREATE PROCEDURE CalculateBills
AS
BEGIN
   SELECT
      A.P_code,
      P.Pname,
      A.EntryDate,
      A.DischargeDate,
      A.wardno,
      DATEDIFF(DAY, A.EntryDate, A.DischargeDate) AS
DaysStayed,
      (DATEDIFF(DAY, A.EntryDate, A.DischargeDate) *
A.wardno * 100) AS BillAmount
   FROM ADMITTEDPATIENT A
   JOIN PATIENTMASTER P ON A.P_code = P.Pcode
   WHERE A.DischargeDate IS NOT NULL;
END;
```

Q-5) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification)
PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) ADMITTEDPATIENT (Pcode, Entry date, Discharge_date, wardno, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be M (male) or F(female). • Wardno should be less than 6.

Queries:

a) Find the details of the doctors who are treating the patients of ward no 3 & display the result along with patient name & disease. b) Find the name of the disease by which maximum patients are suffering. c) Write a trigger on ADMITTEDPATIENT table such that the wardno value should be between 1-5. d) Write a procedure to give the details of patients who are admitted in the hospital for more than 5 days.

Create a input form for doctors. Apply all possible validations.

->

1. **Create Tables with Integrity Constraints**

```
-- Create DOCTOR table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY,
    Dname VARCHAR(100) NOT NULL,
    Daddress VARCHAR(255) NOT NULL,
    qualification VARCHAR(100) NOT NULL
);

-- Create PATIENTMASTER table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY,
    Pname VARCHAR(100) NOT NULL,
    Padd VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    gender CHAR(1) CHECK (gender IN ('M', 'F')) NOT NULL,
    bloodgroup VARCHAR(5) NOT NULL,
    Did INT NOT NULL,
    FOREIGN KEY (Did) REFERENCES DOCTOR(Did)
);
```

```
-- Create ADMITTEDPATIENT table
CREATE TABLE ADMITTEDPATIENT (
    Pcode INT NOT NULL,
    Entry_date DATE NOT NULL,
    Discharge_date DATE NOT NULL,
    wardno INT CHECK (wardno < 6) NOT NULL,
    disease VARCHAR(100) NOT NULL,
    FOREIGN KEY (Pcode) REFERENCES
PATIENTMASTER(Pcode)
);
```

2. **Insert Sample Data (Minimum 10 rows per table)**

```
-- Insert data into DOCTOR
INSERT INTO DOCTOR VALUES
(1, 'Dr. Mehta', 'Delhi', 'MD'),
(2, 'Dr. Sharma', 'Mumbai', 'MBBS'),
(3, 'Dr. Kapoor', 'Bangalore', 'MS'),
(4, 'Dr. Singh', 'Chennai', 'MD'),
(5, 'Dr. Iyer', 'Hyderabad', 'MBBS'),
(6, 'Dr. Das', 'Kolkata', 'MD'),
(7, 'Dr. Joshi', 'Pune', 'MBBS'),
(8, 'Dr. Reddy', 'Ahmedabad', 'MS'),
(9, 'Dr. Nair', 'Jaipur', 'MD'),
```

```
(10, 'Dr. Thomas', 'Cochin', 'MBBS');

-- Insert data into PATIENTMASTER
INSERT INTO PATIENTMASTER VALUES
(101, 'Amit', 'Delhi', 25, 'M', 'B+', 1),
(102, 'Riya', 'Mumbai', 30, 'F', 'O+', 2),
(103, 'Anil', 'Bangalore', 45, 'M', 'A-', 3),
(104, 'Sneha', 'Chennai', 33, 'F', 'B-', 4),
(105, 'Karan', 'Hyderabad', 50, 'M', 'O+', 5),
(106, 'Neha', 'Kolkata', 29, 'F', 'AB+', 6),
(107, 'Rakesh', 'Pune', 60, 'M', 'B+', 7),
(108, 'Priya', 'Ahmedabad', 22, 'F', 'O-', 8),
(109, 'Raj', 'Jaipur', 40, 'M', 'A+', 9),
(110, 'Tina', 'Cochin', 28, 'F', 'B+', 10);

-- Insert data into ADMITTEDPATIENT
INSERT INTO ADMITTEDPATIENT VALUES
(101, '2025-04-01', '2025-04-08', 3, 'Flu'),
(102, '2025-04-02', '2025-04-07', 2, 'Malaria'),
(103, '2025-04-05', '2025-04-15', 3, 'Covid-19'),
(104, '2025-04-07', '2025-04-10', 1, 'Typhoid'),
(105, '2025-04-10', '2025-04-16', 3, 'Flu'),
```

(106, '2025-04-12', '2025-04-18', 4, 'Dengue'),

(107, '2025-04-13', '2025-04-20', 5, 'Covid-19'),

(108, '2025-04-14', '2025-04-16', 2, 'Malaria'),

(109, '2025-04-15', '2025-04-21', 3, 'Flu'),

(110, '2025-04-17', '2025-04-19', 1, 'Allergy');

3. **SQL Queries**

a) Doctors treating patients in ward no 3

```
SELECT D.Did, D.Dname, D.Daddress, D.qualification,
P.Pname, A.disease

FROM DOCTOR D

JOIN PATIENTMASTER P ON D.Did = P.Did

JOIN ADMITTEDPATIENT A ON P.Pcode = A.Pcode

WHERE A.wardno = 3;
```

b) Name of disease with maximum patients

```
SELECT TOP 1 disease, COUNT(*) AS PatientCount
FROM ADMITTEDPATIENT
GROUP BY disease
ORDER BY PatientCount DESC;
```

4. Stored Procedure: Patients admitted > 5 days

```
CREATE PROCEDURE GetLongAdmittedPatients
AS
BEGIN
   SELECT P.Pcode, P.Pname, A.Entry_date,
A.Discharge_date,
```

```
        DATEDIFF(DAY, A.Entry_date, A.Discharge_date)
AS DaysAdmitted
   FROM PATIENTMASTER P
   JOIN ADMITTEDPATIENT A ON P.Pcode = A.Pcode
   WHERE DATEDIFF(DAY, A.Entry_date,
A.Discharge_date) > 5;
END;
```

Q-6) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least l 0 records in each table).

DOCTOR (Did, Dname, Daddress, qualification) PATIENTMASTER (Pcode, Pnarne, Padd, age, gender, bloodgroup, aid) ADMITTEDPATIENT (Pcode, Entry_date, Discharge date, wardno, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be M (male) or F(female). • Wardno should be less than 6.

Queries:

a) Find details of the patients who are treated by M.B.B.S. doctors. b) Find the details of patient who is suffered from blood cancer having age less than 50 years & blood group is A. c) write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients. (bill no of days * 600) d) Write a cursor on PATIENTMASTER table to fetch the last record & display the rows in that table.

Create a data entry for New Doctor's entry. Apply all possible validations

->

1. Create Tables with Constraints

```
CREATE DATABASE HospitalDB;
GO

USE HospitalDB;
GO

-- 1. DOCTOR Table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY,
    Dname VARCHAR(100) NOT NULL,
    Daddress VARCHAR(200) NOT NULL,
    qualification VARCHAR(50) NOT NULL
);

-- 2. PATIENTMASTER Table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY,
    Pname VARCHAR(100) NOT NULL,
    Padd VARCHAR(200) NOT NULL,
    age INT NOT NULL CHECK (age > 0),
```

```
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F')),
    bloodgroup VARCHAR(5) NOT NULL,
    aid INT NOT NULL -- FK to Doctor's Did
);

-- 3. ADMITTEDPATIENT Table
CREATE TABLE ADMITTEDPATIENT (
    Pcode INT NOT NULL,
    Entry_date DATE NOT NULL,
    Discharge_date DATE NOT NULL,
    wardno INT NOT NULL CHECK (wardno < 6),
    disease VARCHAR(100) NOT NULL,
    FOREIGN KEY (Pcode) REFERENCES
PATIENTMASTER(Pcode)
);
```

2. Insert Sample Records

```
-- Insert into DOCTOR
INSERT INTO DOCTOR VALUES
(1, 'Dr. Meera Rao', 'Delhi', 'M.B.B.S.'),
(2, 'Dr. Sanjay Kulkarni', 'Mumbai', 'MD'),
(3, 'Dr. Alok Singh', 'Pune', 'M.B.B.S.'),
(4, 'Dr. Rita Sharma', 'Lucknow', 'MS'),
```

```
(5, 'Dr. Karan Mehta', 'Jaipur', 'M.B.B.S.'),
(6, 'Dr. Neha Jain', 'Bangalore', 'MD'),
(7, 'Dr. Priya Verma', 'Chennai', 'M.B.B.S.'),
(8, 'Dr. Raj Patel', 'Ahmedabad', 'MS'),
(9, 'Dr. Tarun Gupta', 'Hyderabad', 'M.B.B.S.'),
(10, 'Dr. Anjali Desai', 'Kolkata', 'M.B.B.S.');

-- Insert into PATIENTMASTER
INSERT INTO PATIENTMASTER VALUES
(101, 'Ankit Roy', 'Delhi', 45, 'M', 'A', 1),
(102, 'Rina Das', 'Kolkata', 32, 'F', 'B', 2),
(103, 'Mohit Jain', 'Mumbai', 60, 'M', 'A+', 3),
(104, 'Sneha Paul', 'Pune', 40, 'F', 'A', 5),
(105, 'Rahul Sinha', 'Delhi', 55, 'M', 'B+', 4),
(106, 'Neha Sharma', 'Lucknow', 25, 'F', 'A', 3),
(107, 'Anuj Mehta', 'Jaipur', 48, 'M', 'O+', 1),
(108, 'Meena Kumari', 'Chennai', 35, 'F', 'A', 5),
(109, 'Ravi Patel', 'Ahmedabad', 38, 'M', 'AB+', 7),
(110, 'Pooja Singh', 'Bangalore', 28, 'F', 'A', 9);

-- Insert into ADMITTEDPATIENT
INSERT INTO ADMITTEDPATIENT VALUES
```

(101, '2024-12-01', '2024-12-10', 2, 'Fever'),

(102, '2024-12-03', '2024-12-08', 1, 'Malaria'),

(103, '2024-12-05', '2024-12-15', 3, 'Diabetes'),

(104, '2024-12-06', '2024-12-11', 4, 'Blood Cancer'),

(105, '2024-12-07', '2024-12-13', 2, 'TB'),

(106, '2024-12-08', '2024-12-12', 3, 'Blood Cancer'),

(107, '2024-12-10', '2024-12-14', 1, 'Allergy'),

(108, '2024-12-11', '2024-12-17', 2, 'Fever'),

(109, '2024-12-12', '2024-12-19', 3, 'Blood Cancer'),

(110, '2024-12-13', '2024-12-16', 2, 'Covid');

3. **Queries**

a) Patients treated by M.B.B.S. doctors:

```
SELECT P.*
FROM PATIENTMASTER P
JOIN DOCTOR D ON P.aid = D.Did
WHERE D.qualification = 'M.B.B.S.';
```

b) Patient suffering from **blood cancer**, **age < 50**, **blood group A**:

```
SELECT P.*
FROM PATIENTMASTER P
JOIN ADMITTEDPATIENT A ON P.Pcode = A.Pcode
WHERE A.disease = 'Blood Cancer'
  AND P.age < 50
  AND P.bloodgroup = 'A';
```

4. Procedure to calculate bill (days × 600)

```
CREATE PROCEDURE CalculateBills
AS
BEGIN
   SELECT
      Pcode,
      Entry_date,
      Discharge_date,
      DATEDIFF(DAY, Entry_date, Discharge_date) AS
Days_Stayed,
      DATEDIFF(DAY, Entry_date, Discharge_date) * 600
AS Bill_Amount
   FROM ADMITTEDPATIENT;
END;
GO

-- Execute the procedure
EXEC CalculateBills;
```

Q-7) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

DOCTOR (Did, Dname, Daddress, qualification) PATIENTMASTER (Pcocle, Pname, Padd, age, gender, bloodgroup, Did) ADMITTEDPATIENT (Pcode, Entry date, Discharge date. wardno, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be M (male) or F(female). • Wardno should be less than 6.

Queries:

a) Find details of the patients who are treated by M.S. doctors. b) Find the name of doctor who is treating maximum number of patients. c) Write a procedure to give the details of patients who are admitted in the hospital for more than 15 days. d) Create a view on DOCTOR & PATIENTMASTER tables. Update details of the patients who are treated by B.A.-M.S. doctors to M.B.B.S doctor.

Create a data entry form for discharging a patient. Also give information regarding his bill. (bill no_of_days * 500)

->

1. Create Tables with Integrity Constraints

```
-- DOCTOR table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY,
    Dname VARCHAR(100) NOT NULL,
    Daddress VARCHAR(200) NOT NULL,
    qualification VARCHAR(50) NOT NULL
);

-- PATIENTMASTER table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY,
    Pname VARCHAR(100) NOT NULL,
    Padd VARCHAR(200) NOT NULL,
    age INT NOT NULL,
    gender CHAR(1) CHECK (gender IN ('M', 'F')) NOT NULL,
    bloodgroup VARCHAR(5) NOT NULL,
    Did INT NOT NULL FOREIGN KEY REFERENCES DOCTOR(Did)
);

-- ADMITTEDPATIENT table
```

```
CREATE TABLE ADMITTEDPATIENT (
   Pcode INT PRIMARY KEY FOREIGN KEY REFERENCES PATIENTMASTER(Pcode),
   entry_date DATE NOT NULL,
   discharge_date DATE NOT NULL,
   wardno INT CHECK (wardno < 6) NOT NULL,
   disease VARCHAR(100) NOT NULL
);
```

2. Insert Sample Data

```
-- DOCTOR data
INSERT INTO DOCTOR VALUES
(1, 'Dr. Ramesh', 'Delhi', 'M.S.'),
(2, 'Dr. Suresh', 'Mumbai', 'M.B.B.S'),
(3, 'Dr. Meena', 'Chennai', 'M.S.'),
(4, 'Dr. Asha', 'Kolkata', 'B.A.-M.S.'),
(5, 'Dr. Vivek', 'Pune', 'M.B.B.S'),
(6, 'Dr. Neha', 'Hyderabad', 'M.D.'),
(7, 'Dr. Kamal', 'Jaipur', 'M.S.'),
(8, 'Dr. Ritu', 'Lucknow', 'B.A.-M.S.'),
(9, 'Dr. Shyam', 'Ahmedabad', 'M.B.B.S'),
(10, 'Dr. Kavita', 'Bangalore', 'M.B.B.S');
```

```
-- PATIENTMASTER data
INSERT INTO PATIENTMASTER VALUES
(101, 'Aman', 'Delhi', 25, 'M', 'A+', 1),
(102, 'Priya', 'Mumbai', 30, 'F', 'B+', 2),
(103, 'Rohit', 'Chennai', 40, 'M', 'O-', 3),
(104, 'Sneha', 'Kolkata', 22, 'F', 'AB+', 4),
(105, 'Raj', 'Pune', 50, 'M', 'A-', 5),
(106, 'Simran', 'Hyderabad', 35, 'F', 'B-', 6),
(107, 'Vikram', 'Jaipur', 45, 'M', 'O+', 7),
(108, 'Neha', 'Lucknow', 28, 'F', 'A+', 8),
(109, 'Karan', 'Ahmedabad', 31, 'M', 'AB-', 9),
(110, 'Tina', 'Bangalore', 26, 'F', 'B+', 10);

-- ADMITTEDPATIENT data
INSERT INTO ADMITTEDPATIENT VALUES
(101, '2025-03-01', '2025-03-20', 1, 'Typhoid'),
(102, '2025-04-01', '2025-04-10', 2, 'Fracture'),
(103, '2025-03-15', '2025-03-30', 3, 'Malaria'),
(104, '2025-03-10', '2025-03-28', 1, 'Dengue'),
(105, '2025-04-01', '2025-04-03', 4, 'Cough'),
(106, '2025-03-05', '2025-03-22', 2, 'Fever'),
(107, '2025-04-01', '2025-04-20', 3, 'Covid'),
```

(108, '2025-03-12', '2025-03-20', 4, 'Injury'),

(109, '2025-03-18', '2025-04-10', 1, 'Ulcer'),

(110, '2025-03-01', '2025-03-18', 2, 'Allergy');

3. SQL Queries

(a) Patients treated by M.S. Doctors

```
SELECT P.*
FROM PATIENTMASTER P
JOIN DOCTOR D ON P.Did = D.Did
WHERE D.qualification = 'M.S.';
```

(b) Doctor Treating Maximum Patients

```
SELECT TOP 1 D.Dname, COUNT(*) AS PatientCount
FROM PATIENTMASTER P
JOIN DOCTOR D ON P.Did = D.Did
GROUP BY D.Dname
ORDER BY PatientCount DESC;
```

c) Procedure: Patients Admitted > 15 Days

```
CREATE PROCEDURE GetLongAdmittedPatients
AS
BEGIN
   SELECT P.Pname, A.entry_date, A.discharge_date,
        DATEDIFF(DAY, A.entry_date, A.discharge_date) AS
DaysAdmitted
```

```
    FROM PATIENTMASTER P
    JOIN ADMITTEDPATIENT A ON P.Pcode = A.Pcode
    WHERE DATEDIFF(DAY, A.entry_date, A.discharge_date) >
15;
END;
```

Q-8) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).
ACCOUNT (accno, open date. acctype, balance)
TRANSACTION (trans id, trans date, accno, trans type, amount) CUSTOMER (cust id, name, address, Accno)

Integrity Constraints:

• The values of any attributes should not be null. • acctype value should be P(Personal) or J(Joint). • Accno should be less than 3 digits. • Trans type should be C(Credit) or D(Debit)

Queries:

a) Find the details of customers whose minimum balance is 1 lakhs. b) Find the details of amount credited within the period 25-3-2012 to 28-3- 2012 c) Write a trigger on TRANSACTION table to calculate current balance of account on which transaction is made. d) Write a cursor on ACCOUNT table of balance attribute such that if the balance is less than 10000 then print the 'loan is not provided'else 'loan is provided'.
Create a data entry for New account entry. Apply all possible validations

->

1. Create the Database & Tables with Integrity Constraints

```
CREATE DATABASE BankDB;
GO

USE BankDB;
GO

-- ACCOUNT Table
CREATE TABLE ACCOUNT (
   accno INT PRIMARY KEY CHECK (accno < 100), -- Less
than 3 digits
   opendate DATE NOT NULL,
   acctype CHAR(1) NOT NULL CHECK (acctype IN ('P', 'J')),
   balance DECIMAL(18, 2) NOT NULL CHECK (balance >=
0)
);

-- CUSTOMER Table
CREATE TABLE CUSTOMER (
   custid INT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   address VARCHAR(255) NOT NULL,
   accno INT NOT NULL,
   FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)
);

-- TRANSACTION Table
```

```
CREATE TABLE TRANSACTION (
   transid INT PRIMARY KEY,
   transdate DATE NOT NULL,
   accno INT NOT NULL,
   transtype CHAR(1) NOT NULL CHECK (transtype IN ('C',
'D')),
   amount DECIMAL(18, 2) NOT NULL CHECK (amount > 0),
   FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)
);
```

2. Insert Sample Data

```
-- ACCOUNT
INSERT INTO ACCOUNT VALUES
(1, '2010-01-10', 'P', 150000.00),
(2, '2012-03-20', 'J', 98000.00),
(3, '2011-07-19', 'P', 120000.00),
(4, '2015-12-01', 'P', 5000.00),
(5, '2013-11-25', 'J', 105000.00),
(6, '2014-08-30', 'P', 87000.00),
(7, '2016-09-10', 'J', 11000.00),
(8, '2020-02-17', 'P', 300000.00),
(9, '2018-05-15', 'P', 2000.00),
(10, '2019-06-21', 'J', 15000.00);

-- CUSTOMER
INSERT INTO CUSTOMER VALUES
(101, 'Alice', 'New York', 1),
(102, 'Bob', 'Los Angeles', 2),
(103, 'Carol', 'Chicago', 3),
(104, 'David', 'Houston', 4),
```

```
(105, 'Eve', 'Phoenix', 5),
(106, 'Frank', 'San Antonio', 6),
(107, 'Grace', 'San Diego', 7),
(108, 'Heidi', 'Dallas', 8),
(109, 'Ivan', 'San Jose', 9),
(110, 'Judy', 'Austin', 10);

-- TRANSACTION
INSERT INTO TRANSACTION VALUES
(1001, '2012-03-25', 1, 'C', 10000.00),
(1002, '2012-03-26', 2, 'C', 5000.00),
(1003, '2012-03-27', 3, 'D', 4000.00),
(1004, '2012-03-28', 4, 'C', 3000.00),
(1005, '2014-06-11', 5, 'D', 10000.00),
(1006, '2015-08-19', 6, 'C', 25000.00),
(1007, '2016-09-10', 7, 'C', 15000.00),
(1008, '2020-01-01', 8, 'D', 100000.00),
(1009, '2022-07-04', 9, 'C', 500.00),
(1010, '2023-11-21', 10, 'D', 2000.00);
```

3. Queries

a) Customers with minimum balance = 1 lakh

```
SELECT C.*
FROM CUSTOMER C
JOIN ACCOUNT A ON C.accno = A.accno
WHERE A.balance >= 100000;
```

b) Amount credited between 25-03-2012 and 28-03-2012

```
SELECT *
FROM TRANSACTION
WHERE transtype = 'C'
```

```
AND transdate BETWEEN '2012-03-25' AND '2012-03-28';
```

Q-9) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

ACCOUNT (accno, open date, acctype, balance)
TRANSACTION (trans id, trans date, cno, trans type, amount)
CUSTOMER (cust id, name, address, accno)

Integrity Constraints:

• The values of any attributes should not be null. • acctype value should be P(Personal) or J(Joint). • Accno should be less than 3 digits. • Trans_type should be C(Credit) Or D(Debit)

Queries:

a) Find the details of customers who have personal accounts & balance is less than 2 lakhs. b) Find the details of customers who have joint accounts. c) Write a trigger on ACCOUNT table such that if balance is less than 300 then customer should not withdraw the money. d) Write a procedure on ACCOUNT & TRANSACTION table such that as user enters new transaction the balance is, updated in ACCOUNT table.

Create a data entry for New customer entry. Apply all possible validations.

->

1. **Create Tables with Constraints**

```
CREATE TABLE ACCOUNT (
    accno INT PRIMARY KEY CHECK (accno < 1000),
    opendate DATE NOT NULL,
    acctype CHAR(1) NOT NULL CHECK (acctype IN ('P', 'J')),
    balance DECIMAL(15,2) NOT NULL CHECK (balance >= 0)
);

CREATE TABLE CUSTOMER (
    custid INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(255) NOT NULL,
    accno INT NOT NULL,
    FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)
);

CREATE TABLE TRANSACTION (
    transid INT PRIMARY KEY,
    transdate DATE NOT NULL,
    cno INT NOT NULL,
```

```
    transtype CHAR(1) NOT NULL CHECK (transtype IN ('C',
'D')),
    amount DECIMAL(15,2) NOT NULL CHECK (amount > 0),
    FOREIGN KEY (cno) REFERENCES CUSTOMER(custid)
);
```

2. Insert Sample Data (10 Records Each)

```
-- Insert into ACCOUNT
INSERT INTO ACCOUNT VALUES
(101, '2022-01-10', 'P', 150000),
(102, '2022-03-15', 'J', 300000),
(103, '2022-05-20', 'P', 50000),
(104, '2023-01-10', 'P', 250000),
(105, '2022-08-18', 'J', 180000),
(106, '2022-09-12', 'P', 199999),
(107, '2022-11-25', 'J', 120000),
(108, '2023-02-14', 'P', 70000),
(109, '2023-03-21', 'J', 250000),
(110, '2023-04-01', 'P', 5000);

-- Insert into CUSTOMER
INSERT INTO CUSTOMER VALUES
(1, 'John Doe', '123 Main St', 101),
```

```
(2, 'Jane Smith', '456 Park Ave', 102),
(3, 'Alice Brown', '789 Oak St', 103),
(4, 'Bob Martin', '321 Pine St', 104),
(5, 'Charlie Lee', '654 Elm St', 105),
(6, 'Daisy Ray', '147 Maple St', 106),
(7, 'Ethan Hunt', '951 Spruce St', 107),
(8, 'Fiona Cruz', '852 Birch St', 108),
(9, 'George King', '963 Cedar St', 109),
(10, 'Hannah Watts', '741 Willow St', 110);

-- Insert into TRANSACTION
INSERT INTO TRANSACTION VALUES
(1, '2024-01-01', 1, 'C', 20000),
(2, '2024-01-02', 2, 'D', 50000),
(3, '2024-01-03', 3, 'C', 10000),
(4, '2024-01-04', 4, 'D', 100000),
(5, '2024-01-05', 5, 'C', 30000),
(6, '2024-01-06', 6, 'D', 25000),
(7, '2024-01-07', 7, 'C', 40000),
(8, '2024-01-08', 8, 'D', 2000),
(9, '2024-01-09', 9, 'C', 50000),
(10, '2024-01-10', 10, 'D', 1000);
```

3. SQL Queries

a) Customers with Personal Account and balance < 2 Lakhs

```
SELECT C.*
FROM CUSTOMER C
JOIN ACCOUNT A ON C.accno = A.accno
WHERE A.acctype = 'P' AND A.balance < 200000;
```

b) Customers with Joint Account

```
SELECT C.*
FROM CUSTOMER C
JOIN ACCOUNT A ON C.accno = A.accno
WHERE A.acctype = 'J';
```

Q-10) Create database using following schema. Apply given integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). ACCOUNT (accno, open date. acctype, balance) TRANSACTION (trans id, trans date, accno, trans type, amount) CUSTOMER (cust id, name, address. accno)

Integrity Constraints:

• The values of any attributes should not be null. • acctype value should be P(Personal) or Moira): • Accno should be less than 3 digits. • Transtype should be C(Credit) or D(Debit)

Queries:

a) Find the details of all transactions performed on account number 101. Also specify the name/names of cutomers who owns that account. b) Find the details ol amount credited within the period 15 -3-2012 to 18 -3 -2012. c) Write a trigger on insert on ACCOUNT table such that the account which is having balance less than or equal to 500 should not be debited. d) Write a procedure on ACCOUNT table to calculate interest on current balance from open_date to today's date. (Take interest rate from user).

Create a data entry for New account entry. Apply all possible validations**.**

->

1: Create the Tables with Constraints

```
-- Create ACCOUNT table
CREATE TABLE ACCOUNT (
   accno INT PRIMARY KEY CHECK (accno < 1000),
   [open date] DATE NOT NULL,
   acctype CHAR(1) NOT NULL CHECK (acctype IN ('P', 'M')),
-- P = Personal, M = Moira
   balance DECIMAL(10, 2) NOT NULL CHECK (balance >= 0)
);

-- Create CUSTOMER table
CREATE TABLE CUSTOMER (
   custid INT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   address VARCHAR(200) NOT NULL,
   accno INT NOT NULL FOREIGN KEY REFERENCES
ACCOUNT(accno)
);

-- Create TRANSACTION table
CREATE TABLE TRANSACTION (
```

```
   transid INT PRIMARY KEY,
   [trans date] DATE NOT NULL,
   accno INT NOT NULL FOREIGN KEY REFERENCES ACCOUNT(accno),
   transtype CHAR(1) NOT NULL CHECK (transtype IN ('C', 'D')),  -- C = Credit, D = Debit
   amount DECIMAL(10,2) NOT NULL CHECK (amount > 0)
);
```

2: Insert Sample Data (10+ records in each table)

```
-- Insert into ACCOUNT
INSERT INTO ACCOUNT VALUES
(101, '2012-03-10', 'P', 2000),
(102, '2012-03-11', 'M', 1500),
(103, '2012-03-12', 'P', 1000),
(104, '2012-03-13', 'M', 2500),
(105, '2012-03-14', 'P', 300),
(106, '2012-03-15', 'M', 5000),
(107, '2012-03-16', 'P', 10000),
(108, '2012-03-17', 'M', 750),
(109, '2012-03-18', 'P', 850),
(110, '2012-03-19', 'M', 620);
```

```
-- Insert into CUSTOMER
INSERT INTO CUSTOMER VALUES
(1, 'John Doe', 'New York', 101),
(2, 'Jane Smith', 'California', 102),
(3, 'Mike Brown', 'Texas', 103),
(4, 'Lucy Gray', 'Nevada', 104),
(5, 'Tom Wayne', 'Arizona', 105),
(6, 'Sara Lee', 'Florida', 106),
(7, 'Tim Cook', 'Georgia', 107),
(8, 'Elon Tusk', 'New Mexico', 108),
(9, 'Mona Lisa', 'Washington', 109),
(10, 'Bruce Kent', 'Oregon', 110);

-- Insert into TRANSACTION
INSERT INTO TRANSACTION VALUES
(1, '2012-03-15', 101, 'C', 500),
(2, '2012-03-16', 101, 'D', 300),
(3, '2012-03-17', 102, 'C', 200),
(4, '2012-03-18', 103, 'C', 100),
(5, '2012-03-19', 104, 'D', 500),
(6, '2012-03-15', 105, 'C', 800),
(7, '2012-03-16', 106, 'D', 400),
```

(8, '2012-03-17', 107, 'C', 900),

(9, '2012-03-18', 108, 'D', 700),

(10, '2012-03-15', 109, 'C', 600);

3: Queries

a) Details of transactions for account number 101 and customer names

```
SELECT T.*, C.name
FROM TRANSACTION T
JOIN CUSTOMER C ON T.accno = C.accno
WHERE T.accno = 101;
```

b) Amount credited between 15-3-2012 to 18-3-2012

```
SELECT *
FROM TRANSACTION
WHERE transtype = 'C'
  AND [trans date] BETWEEN '2012-03-15' AND '2012-03-18';
```

Q-11) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). ACCOUNT (accno, open date, acctype, balance) TRANSACTION (trans id, trans date, accno, trans type, amount) CUSTOMER (cust id name, address, accno)

Integrity Constraints:

• The values of any attributes should not be null. • acctype value should be P(Personal) or Moira). • Accno should be less than 3 digits. • Trans_type should be C(Credit) or D(Debit)

Queries:

a) Find the details of customers who have opened the accounts within the period 25-3-2012 to 28-3-2012. b) Find the details of customers who have joint accounts & balance is less than 2 lakhs. c) Write a trigger TRANSACTION on table to calculate the current balance of the account on which transaction is made. (if trans_type = c then bal = bal amt else if trans_type = d then bal = bal — amt) d) write a cursor on CUSTOMER table to fetch the last row.

Create a data entry for New customer entry. Apply all possible validations.

->

1: Database Schema and Integrity Constraints

```
CREATE DATABASE BankDB;
GO

USE BankDB;
GO

-- Create ACCOUNT table
CREATE TABLE ACCOUNT (
    accno INT PRIMARY KEY CHECK (accno < 100), -- Ensuring accno is less than 3 digits
    open_date DATE NOT NULL,
    acctype CHAR(1) CHECK (acctype IN ('P', 'M')), -- Ensure acctype is 'P' or 'M'
    balance DECIMAL(15, 2) NOT NULL
);
GO

-- Create TRANSACTION table
CREATE TABLE TRANSACTION (
    trans_id INT PRIMARY KEY,
```

```
    trans_date DATE NOT NULL,
    accno INT NOT NULL,
    trans_type CHAR(1) CHECK (trans_type IN ('C', 'D')), -- 'C' for Credit, 'D' for Debit
    amount DECIMAL(15, 2) NOT NULL,
    FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)
);
GO

-- Create CUSTOMER table
CREATE TABLE CUSTOMER (
    cust_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(255) NOT NULL,
    accno INT NOT NULL,
    FOREIGN KEY (accno) REFERENCES ACCOUNT(accno)
);
GO
```

2: **Insert data into the tables (with at least 10 records for each).**

```
-- Insert records into ACCOUNT table
INSERT INTO ACCOUNT (accno, open_date, acctype, balance) VALUES
```

```
(101, '2012-03-25', 'P', 50000.00),
(102, '2012-03-26', 'M', 150000.00),
(103, '2012-03-27', 'P', 250000.00),
(104, '2012-03-28', 'M', 120000.00),
(105, '2012-04-01', 'P', 30000.00),
(106, '2012-04-02', 'M', 95000.00),
(107, '2012-04-05', 'P', 100000.00),
(108, '2012-04-10', 'P', 200000.00),
(109, '2012-04-15', 'M', 50000.00),
(110, '2012-04-20', 'P', 75000.00);
GO

-- Insert records into TRANSACTION table
INSERT INTO TRANSACTION (trans_id, trans_date, accno,
trans_type, amount) VALUES
(1, '2012-03-25', 101, 'C', 10000.00),
(2, '2012-03-26', 102, 'D', 20000.00),
(3, '2012-03-27', 103, 'C', 50000.00),
(4, '2012-03-28', 104, 'D', 10000.00),
(5, '2012-04-01', 105, 'C', 15000.00),
(6, '2012-04-02', 106, 'D', 5000.00),
(7, '2012-04-05', 107, 'C', 30000.00),
```

```
(8, '2012-04-10', 108, 'D', 10000.00),
(9, '2012-04-15', 109, 'C', 25000.00),
(10, '2012-04-20', 110, 'D', 5000.00);
GO

-- Insert records into CUSTOMER table
INSERT INTO CUSTOMER (cust_id, name, address, accno)
VALUES
(1, 'John Doe', '123 Main St, City, State', 101),
(2, 'Jane Smith', '456 Oak St, City, State', 102),
(3, 'Emily Johnson', '789 Pine St, City, State', 103),
(4, 'Michael Brown', '101 Maple St, City, State', 104),
(5, 'Sarah Davis', '202 Elm St, City, State', 105),
(6, 'David Wilson', '303 Birch St, City, State', 106),
(7, 'Mary Moore', '404 Cedar St, City, State', 107),
(8, 'James Taylor', '505 Spruce St, City, State', 108),
(9, 'Linda Anderson', '606 Willow St, City, State', 109),
(10, 'Robert Thomas', '707 Redwood St, City, State', 110);
```

3: SQL Queries

a) Find the details of customers who have opened accounts within the period 25-03-2012 to 28-03-2012.

```
SELECT c.cust_id, c.name, c.address, a.accno,
a.open_date, a.acctype, a.balance
FROM CUSTOMER c
JOIN ACCOUNT a ON c.accno = a.accno
WHERE a.open_date BETWEEN '2012-03-25' AND '2012-
03-28';
```

b) Find the details of customers who have joint accounts & balance is less than 2 lakhs.

Since there is no explicit field to define whether an account is a joint account or not, we assume that the 'M' (joint) type represents joint accounts.

```
SELECT c.cust_id, c.name, c.address, a.accno,
a.open_date, a.acctype, a.balance
FROM CUSTOMER c
JOIN ACCOUNT a ON c.accno = a.accno
WHERE a.acctype = 'M' AND a.balance < 200000;
```

Q-12) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in eae,h table). EMPLOYEE (fname, Mame, ssn, sex, salary, joindate,superssn, dno,) DEPT (dname, dnum, mgrssn, dlocation) PROJECT (pname, pno, plocation, dnumber) WORK ON (ssn, pno, hours)

Integrity Constraints:

• The values of any attributes should not be null. • The deptno should less than 4 digit • ssn -(social security no of-employee) • mgrssn(manager_social_security_no) • superssn(supervisor_social_securit)'_no)

Queries:

a) For every project located in 'jalgaon". List the pno,the controlling detptno and dept manager last name. b) For each project on which more than two employee work, Find the pno, pname & no. of employees who work on the project. c) create a view that has the deptname,manager name & manager salary for every dept. d) Express the following constraint as SQL assertions - "salary of employee must not be greater than the salary of the manager of the dept".

Create a data entry for New employee entry. Apply all possible validations.

->

1: Creating the Database and Tables with Integrity Constraints

```
-- Create the Database
CREATE DATABASE CompanyDB;
GO

USE CompanyDB;
GO

-- Create EMPLOYEE Table
CREATE TABLE EMPLOYEE (
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    ssn CHAR(9) PRIMARY KEY,  -- Social Security Number as the primary key
    sex CHAR(1) NOT NULL,
    salary DECIMAL(10, 2) NOT NULL CHECK (salary > 0),  -- Ensure salary is positive
    joindate DATE NOT NULL,
    superssn CHAR(9),  -- Supervisor's SSN (can be NULL if there's no supervisor)
    dno INT NOT NULL,  -- Department Number
```

```
    FOREIGN KEY (dno) REFERENCES DEPT(dnum),
    FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn)  -- Supervisor should exist in EMPLOYEE table
);
GO

-- Create DEPT Table
CREATE TABLE DEPT (
    dname VARCHAR(50) NOT NULL,
    dnum INT PRIMARY KEY CHECK (dnum < 1000),  -- Department number should be less than 4 digits
    mgrssn CHAR(9) NOT NULL,  -- Manager's SSN
    dlocation VARCHAR(50) NOT NULL,
    FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn)  -- Manager's SSN must exist in EMPLOYEE table
);
GO

-- Create PROJECT Table
CREATE TABLE PROJECT (
    pname VARCHAR(50) NOT NULL,
    pno INT PRIMARY KEY,  -- Project Number
    plocation VARCHAR(50) NOT NULL,
```

```
    dnumber INT NOT NULL,  -- Department number responsible for the project
    FOREIGN KEY (dnumber) REFERENCES DEPT(dnum)
);
GO

-- Create WORK_ON Table
CREATE TABLE WORK_ON (
    ssn CHAR(9),  -- Employee's SSN
    pno INT,  -- Project Number
    hours DECIMAL(5, 2) CHECK (hours >= 0),  -- Ensure hours worked are non-negative
    PRIMARY KEY (ssn, pno),  -- Composite primary key
    FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),
    FOREIGN KEY (pno) REFERENCES PROJECT(pno)
);
```

2: Insert Sample Data

```
-- Insert sample records into DEPT
INSERT INTO DEPT (dname, dnum, mgrssn, dlocation) VALUES
('HR', 1, '123456789', 'New York'),
('IT', 2, '987654321', 'San Francisco'),
('Finance', 3, '234567890', 'Chicago');
```

```
GO


-- Insert sample records into EMPLOYEE
INSERT INTO EMPLOYEE (fname, lname, ssn, sex, salary, joindate, superssn, dno) VALUES
('John', 'Doe', '123456789', 'M', 55000, '2022-01-15', NULL, 1),
('Jane', 'Smith', '234567890', 'F', 62000, '2021-08-01', '123456789', 1),
('Emily', 'Johnson', '345678901', 'F', 73000, '2020-07-11', '234567890', 2),
('Michael', 'Brown', '456789012', 'M', 80000, '2019-12-03', '234567890', 2),
('Chris', 'Davis', '567890123', 'M', 75000, '2021-11-14', '234567890', 3),
('Laura', 'Miller', '678901234', 'F', 54000, '2022-02-10', '567890123', 3),
('George', 'Taylor', '789012345', 'M', 65000, '2018-03-15', '567890123', 1),
('Sarah', 'Anderson', '890123456', 'F', 70000, '2021-06-20', '123456789', 2),
('David', 'Thomas', '901234567', 'M', 48000, '2020-09-28', '234567890', 1),
('Anna', 'Jackson', '123678901', 'F', 85000, '2017-04-12', '123456789', 3);
```

```
GO

-- Insert sample records into PROJECT
INSERT INTO PROJECT (pname, pno, plocation, dnumber) VALUES
('Project A', 101, 'New York', 1),
('Project B', 102, 'San Francisco', 2),
('Project C', 103, 'Chicago', 3),
('Project D', 104, 'Jalgaon', 2),
('Project E', 105, 'Jalgaon', 3);
GO

-- Insert sample records into WORK_ON
INSERT INTO WORK_ON (ssn, pno, hours) VALUES
('123456789', 101, 40),
('234567890', 101, 35),
('345678901', 102, 45),
('456789012', 102, 38),
('567890123', 103, 40),
('678901234', 103, 42),
('789012345', 104, 50),
('890123456', 104, 30),
```

('901234567', 105, 40),

('123678901', 105, 45);

3: Queries

1. **Query (a)**: For every project located in 'Jalgaon', list the project number, the controlling department number, and the department manager's last name.

```
SELECT
   p.pno,
   p.dnumber,
   d.dname,
   e.lname AS manager_lname
FROM
   PROJECT p
JOIN
   DEPT d ON p.dnumber = d.dnum
JOIN
   EMPLOYEE e ON d.mgrssn = e.ssn
WHERE
   p.plocation = 'Jalgaon';
```

2. **Query (b)**: For each project on which more than two employees work, find the project number, project name, and the number of employees who work on the project.

```
SELECT
```

```
    p.pno,
    p.pname,
    COUNT(w.ssn) AS num_employees
FROM
    PROJECT p
JOIN
    WORK_ON w ON p.pno = w.pno
GROUP BY
    p.pno, p.pname
HAVING
    COUNT(w.ssn) > 2;
```

4. **Query (d)**: Express the following constraint as SQL assertions:
    - "The salary of an employee must not be greater than the salary of the manager of the department."

```
CREATE ASSERTION salary_check
CHECK (
    NOT EXISTS (
        SELECT 1
        FROM EMPLOYEE e
        JOIN DEPT d ON e.dno = d.dnum
        JOIN EMPLOYEE mgr ON d.mgrssn = mgr.ssn
```

```
        WHERE e.salary > mgr.salary
    )
);
```

Q-13) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). EMPLOYEE (fname, lname, ssn, sex, salary, joindate,superssn, dno,) DEPT (dname, dnum, En_grssn, (llocation) PROJECT (pname, pno, plocation, dnumber) WORK ON (5sn, pno, hours)

Integrity Constraints:

• The values of any attributes should not be null. • The deptno should less than 4 digit • Ssn (social security_no of employee) • Mgrssn(managersocial_security_no) a Superssn(supervisor_social_security_no)

Queries:

a) For each employee, Find the employee first & last name & the first & last name of his or her immediate supervisor. b) For each dept, Find the deptno, the no. of employees in the dept & their average salary. c) Create a view that has pname,controlling dept name, no of employees, & total hours worked per week on the project for each project with more than one employee working on it. d) Create a procedure on EMPLOYEE table to determine the employees who will get promotion. (An employee will get promotion after working on 5 projects.)

Create a data report showing the information of all female employees working in "Research" department.

->

1. **Database Schema and Integrity Constraints**

```
-- Create DEPT table
CREATE TABLE DEPT (
    dname VARCHAR(50) NOT NULL,
    dnum INT NOT NULL CHECK (dnum < 1000), -- dnum should be less than 4 digits
    En_grssn VARCHAR(9) NOT NULL, -- Manager's SSN
    llocation VARCHAR(100) NOT NULL
);

-- Create EMPLOYEE table
CREATE TABLE EMPLOYEE (
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    ssn VARCHAR(9) NOT NULL PRIMARY KEY, -- ssn should be unique and not null
    sex CHAR(1) NOT NULL, -- Male (M) or Female (F)
    salary DECIMAL(10, 2) NOT NULL,
    joindate DATE NOT NULL,
    superssn VARCHAR(9), -- Supervisor's SSN (nullable)
    dno INT NOT NULL,
```

```
    FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn), -- Self-referencing for supervisor
    FOREIGN KEY (dno) REFERENCES DEPT(dnum) -- Department number references DEPT
);

-- Create PROJECT table
CREATE TABLE PROJECT (
    pname VARCHAR(50) NOT NULL,
    pno INT NOT NULL PRIMARY KEY,
    plocation VARCHAR(100) NOT NULL,
    dnumber INT NOT NULL,
    FOREIGN KEY (dnumber) REFERENCES DEPT(dnum) -- Department controlling project
);

-- Create WORK_ON table
CREATE TABLE WORK_ON (
    ssn VARCHAR(9) NOT NULL,
    pno INT NOT NULL,
    hours INT NOT NULL,
    PRIMARY KEY (ssn, pno), -- Composite primary key (employee SSN, project number)
```

```
    FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn), -- Employee SSN references EMPLOYEE
    FOREIGN KEY (pno) REFERENCES PROJECT(pno) -- Project number references PROJECT
);
```

2. Sample Data Insertion

```
-- Insert data into DEPT
INSERT INTO DEPT (dname, dnum, En_grssn, llocation)
VALUES
('Research', 1, '123456789', 'Building A'),
('HR', 2, '987654321', 'Building B'),
('Engineering', 3, '135792468', 'Building C'),
('Marketing', 4, '246813579', 'Building D');

-- Insert data into EMPLOYEE
INSERT INTO EMPLOYEE (fname, lname, ssn, sex, salary, joindate, superssn, dno)
VALUES
('John', 'Doe', '123456789', 'M', 50000, '2020-01-01', NULL, 1),
('Jane', 'Smith', '987654321', 'F', 60000, '2019-01-01', '123456789', 2),
('Alice', 'Johnson', '192837465', 'F', 55000, '2021-06-01', '123456789', 1),
('Bob', 'Brown', '564738291', 'M', 70000, '2018-03-15', '987654321', 3),
('Charlie', 'Davis', '847362514', 'M', 45000, '2022-07-01', '192837465', 2),
```

```
('David', 'Martinez', '672839405', 'M', 80000, '2017-11-11', '987654321', 3),
('Emma', 'Clark', '135792468', 'F', 65000, '2016-02-25', '847362514', 4),
('Grace', 'Lewis', '246813579', 'F', 56000, '2020-10-10', '135792468', 1),
('Henry', 'Walker', '314159265', 'M', 53000, '2019-07-21', '564738291', 2),
('Ivy', 'Harris', '271828182', 'F', 60000, '2021-04-20', '672839405', 3);

-- Insert data into PROJECT
INSERT INTO PROJECT (pname, pno, plocation, dnumber)
VALUES
('Project Alpha', 101, 'Location A', 1),
('Project Beta', 102, 'Location B', 1),
('Project Gamma', 103, 'Location C', 2),
('Project Delta', 104, 'Location D', 3),
('Project Epsilon', 105, 'Location E', 4);

-- Insert data into WORK_ON
INSERT INTO WORK_ON (ssn, pno, hours)
VALUES
('123456789', 101, 30),
('987654321', 102, 40),
('192837465', 101, 35),
('564738291', 103, 25),
('847362514', 104, 20),
```

```
('135792468', 105, 50),
('246813579', 101, 45),
('672839405', 103, 28),
('135792468', 104, 38),
('271828182', 105, 32);
```

3. Queries

a) For each employee, find the employee first & last name & the first & last name of his or her immediate supervisor.

```
SELECT
    e.fname AS EmployeeFirstName,
    e.lname AS EmployeeLastName,
    s.fname AS SupervisorFirstName,
    s.lname AS SupervisorLastName
FROM
    EMPLOYEE e
LEFT JOIN
    EMPLOYEE s ON e.superssn = s.ssn;
```

b) For each dept, find the deptno, the no. of employees in the dept & their average salary.

```
SELECT
    d.dnum AS DepartmentNo,
    COUNT(e.ssn) AS NumberOfEmployees,
    AVG(e.salary) AS AverageSalary
FROM
    DEPT d
LEFT JOIN
    EMPLOYEE e ON d.dnum = e.dno
GROUP BY
```

```
        d.dnum;
```

d) Create a procedure on EMPLOYEE table to determine the employees who will get promotion.

```
CREATE PROCEDURE GetPromotionEligibleEmployees AS
BEGIN
  SELECT
    e.fname,
    e.lname,
    COUNT(w.pno) AS ProjectsWorkedOn
  FROM
    EMPLOYEE e
  JOIN
    WORK_ON w ON e.ssn = w.ssn
  GROUP BY
    e.ssn
  HAVING
    COUNT(w.pno) >= 5;
END;
```

d) Create a data report showing the information of all female employees working in "Research" department.

```
SELECT
  e.fname,
  e.lname,
```

```
    e.ssn,
    e.salary,
    e.joindate,
    e.dno
FROM
    EMPLOYEE e
JOIN
    DEPT d ON e.dno = d.dnum
WHERE
    e.sex = 'F'
    AND d.dname = 'Research'
```

Q-14) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in eaCh table).

EMPLOYEE (fna.me, Mame, ssn, sex, salary, joindate,superssn, dno4) DEPT (dname, dnum, mgrssn, dlocation) PROJECT (pname, pno, plocation, dnumber) WORK_ ON (ssn, pno, hours)

Integrity Constraints:

• The values of any attributes should not be null. • The deptno should less than 4 digit • Ssn (social_security_no of employee) • Mgrssn(manager_social_security_no) • Superssn(supervisor_social_security_no)

Queries:

a) Find the ssn of all employees who work on pno 101, 102 or 103. b) Make a list of all pno for project that involve an employee whose last name is 'sonar' either as a worker or as a manager of the dept that control the project. c) Write a trigger on insert on WORK_ON table such that if total work hours of employee in company is less than 20 hours then his salary is deducted. d) Write a cursor on PROJECT table to fetch the first row from the table & display the total number of rows present in the table.

Create a data report showing the informatio.n of all the projects and names of employees working on individual projects**.**

->

1. Create the Database Schema and Integrity Constraints--Create EMPLOYEE Table

```
CREATE TABLE EMPLOYEE (
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    ssn CHAR(9) PRIMARY KEY, -- Assuming SSN is a unique 9-digit number
    sex CHAR(1) CHECK (sex IN ('M', 'F')) NOT NULL,
    salary DECIMAL(10, 2) NOT NULL,
    joindate DATE NOT NULL,
    superssn CHAR(9), -- This will be the SSN of the supervisor
    dno INT NOT NULL, -- Department number (FK reference)
    FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn), -- Self-referencing for supervisor
    FOREIGN KEY (dno) REFERENCES DEPT(dnum) -- Reference to Department
);

-- Create DEPT Table
CREATE TABLE DEPT (
```

```
    dname VARCHAR(50) NOT NULL,
    dnum INT PRIMARY KEY CHECK (dnum < 1000), -- Dept number should be less than 4 digits
    mgrssn CHAR(9) NOT NULL, -- Manager SSN
    dlocation VARCHAR(100) NOT NULL,
    FOREIGN KEY (mgrssn) REFERENCES EMPLOYEE(ssn) -- Reference to Employee table
);

-- Create PROJECT Table
CREATE TABLE PROJECT (
    pname VARCHAR(50) NOT NULL,
    pno INT PRIMARY KEY, -- Project number
    plocation VARCHAR(100) NOT NULL,
    dnumber INT NOT NULL, -- Department number controlling the project
    FOREIGN KEY (dnumber) REFERENCES DEPT(dnum) -- Reference to Department
);

-- Create WORK_ON Table
CREATE TABLE WORK_ON (
    ssn CHAR(9), -- Employee SSN
```

```
    pno INT, -- Project number
    hours DECIMAL(5, 2) NOT NULL,
    PRIMARY KEY (ssn, pno), -- Composite primary key (employee, project)
    FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn), -- Reference to Employee table
    FOREIGN KEY (pno) REFERENCES PROJECT(pno) -- Reference to Project table
);
```

2. **Insert Sample Data into the Tables**

```
-- Insert sample data into DEPT table
INSERT INTO DEPT (dname, dnum, mgrssn, dlocation) VALUES
('HR', 1, '123456789', 'New York'),
('IT', 2, '987654321', 'San Francisco'),
('Sales', 3, '112233445', 'Chicago'),
('Finance', 4, '998877665', 'Los Angeles');

-- Insert sample data into EMPLOYEE table
INSERT INTO EMPLOYEE (fname, lname, ssn, sex, salary, joindate, superssn, dno) VALUES
('John', 'Doe', '123456789', 'M', 60000, '2020-01-10', NULL, 1),
```

('Jane', 'Smith', '987654321', 'F', 65000, '2019-03-15', '123456789', 2),

('Mary', 'Johnson', '112233445', 'F', 50000, '2021-06-25', '987654321', 3),

('Mike', 'Brown', '556677889', 'M', 70000, '2018-02-20', NULL, 4),

('Tom', 'Davis', '223344556', 'M', 75000, '2017-07-10', '556677889', 1),

('Nancy', 'Wilson', '334455667', 'F', 55000, '2020-08-30', '223344556', 2),

('Emily', 'Taylor', '445566778', 'F', 72000, '2019-04-15', '334455667', 3),

('David', 'Anderson', '556677990', 'M', 80000, '2016-11-05', '445566778', 4),

('Laura', 'Thomas', '667788991', 'F', 54000, '2022-09-18', '556677990', 1),

('James', 'Jackson', '778899002', 'M', 60000, '2021-02-05', '667788991', 2);

-- Insert sample data into PROJECT table

INSERT INTO PROJECT (pname, pno, plocation, dnumber) VALUES

('Project Alpha', 101, 'New York', 1),

('Project Beta', 102, 'San Francisco', 2),

```
('Project Gamma', 103, 'Chicago', 3),
('Project Delta', 104, 'Los Angeles', 4),
('Project Epsilon', 105, 'New York', 1),
('Project Zeta', 106, 'San Francisco', 2),
('Project Eta', 107, 'Chicago', 3),
('Project Theta', 108, 'Los Angeles', 4),
('Project Iota', 109, 'New York', 1),
('Project Kappa', 110, 'San Francisco', 2);

-- Insert sample data into WORK_ON table
INSERT INTO WORK_ON (ssn, pno, hours) VALUES
('123456789', 101, 15.5),
('987654321', 102, 22.0),
('112233445', 103, 18.0),
('556677889', 101, 19.0),
('223344556', 102, 25.0),
('334455667', 103, 10.0),
('445566778', 104, 30.0),
('556677990', 105, 15.5),
('667788991', 106, 40.0),
('778899002', 107, 12.0);
```

3. SQL Queries

a) Find the ssn of all employees who work on pno 101, 102, or 103.

```
SELECT DISTINCT ssn
FROM WORK_ON
WHERE pno IN (101, 102, 103);
```

b) Make a list of all pno for projects that involve an employee whose last name is 'sonar' either as a worker or as a manager of the dept that controls the project.

```
SELECT DISTINCT pno
FROM PROJECT
WHERE dnumber IN (
   SELECT dno
   FROM EMPLOYEE
   WHERE lname = 'sonar'
   UNION
   SELECT p.dnumber
   FROM PROJECT p
   JOIN WORK_ON w ON p.pno = w.pno
   JOIN EMPLOYEE e ON w.ssn = e.ssn
   WHERE e.lname = 'sonar'
);
```

4. Data Report Showing Project Information and Employee Names

```
SELECT p.pname, e.fname, e.lname
FROM PROJECT p
JOIN WORK_ON w ON p.pno = w.pno
JOIN EMPLOYEE e ON w.ssn = e.ssn
ORDER BY p.pname, e.lname;
```

Q-15) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). BOOKMASTER (bid, title, author, price) STUDENTMASTER (stud enrollno, sname,class,dept) ACCESSIONTABLE (bid, accession_no,avail) ISSUETABLE(issueid,accession no,stud enrollno,issuedate,cluedate, ret_date,bid)

Integrity Constraints:

• The values of any attributes should not be null. • Avail should be T (if book is not issue) or F (if book is issue)

Queries:

a) Find the name of books which is issued maximum times. b) Find the detail information of books that are issued by computer department students. c) Write a procedure to calculate the fines for the books which are not return on or before due date. no.of days = (ret_date - due_date) fine = no.of days (ret_date - due_date) * 10 d) Write a trigger on insert of ISSUETABLE such that duedate = issuedate + 7

Create a data report that display the information of all books available in the library**.**

->

1. Creating the Database Schema and Tables

```
-- Create Bookmaster table
CREATE TABLE BOOKMASTER (
    bid INT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL
);

-- Create Studentmaster table
CREATE TABLE STUDENTMASTER (
    enrollno INT PRIMARY KEY,
    sname VARCHAR(100) NOT NULL,
    class VARCHAR(50) NOT NULL,
    dept VARCHAR(50) NOT NULL
);

-- Create Accessiontable table
CREATE TABLE ACCESSIONTABLE (
    bid INT,
    accession_no INT PRIMARY KEY,
```

```
    avail CHAR(1) CHECK (avail IN ('T', 'F')) NOT NULL,
    FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)
);

-- Create Issuetable table
CREATE TABLE ISSUETABLE (
    issueid INT PRIMARY KEY,
    accession_no INT,
    enrollno INT,
    issuedate DATE NOT NULL,
    cluedate DATE NOT NULL,
    ret_date DATE,
    bid INT,
    FOREIGN KEY (accession_no) REFERENCES
ACCESSIONTABLE(accession_no),
    FOREIGN KEY (enrollno) REFERENCES
STUDENTMASTER(enrollno),
    FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)
);
```

2. Inserting Data into the Tables

```
-- Inserting into BOOKMASTER
INSERT INTO BOOKMASTER (bid, title, author, price) VALUES
```

(1, 'Book A', 'Author A', 200.00),

(2, 'Book B', 'Author B', 300.00),

(3, 'Book C', 'Author C', 150.00),

(4, 'Book D', 'Author D', 250.00),

(5, 'Book E', 'Author E', 180.00),

(6, 'Book F', 'Author F', 220.00),

(7, 'Book G', 'Author G', 210.00),

(8, 'Book H', 'Author H', 190.00),

(9, 'Book I', 'Author I', 230.00),

(10, 'Book J', 'Author J', 270.00);

-- Inserting into STUDENTMASTER

INSERT INTO STUDENTMASTER (enrollno, sname, class, dept) VALUES

(1, 'Alice', '1st Year', 'Computer'),

(2, 'Bob', '2nd Year', 'Electronics'),

(3, 'Charlie', '3rd Year', 'Computer'),

(4, 'David', '4th Year', 'Mechanical'),

(5, 'Eva', '1st Year', 'Electrical'),

(6, 'Frank', '2nd Year', 'Civil'),

(7, 'Grace', '3rd Year', 'Computer'),

(8, 'Hannah', '4th Year', 'Biotech'),

(9, 'Ivy', '1st Year', 'Computer'),

(10, 'Jack', '2nd Year', 'Electrical');

-- Inserting into ACCESSIONTABLE

INSERT INTO ACCESSIONTABLE (bid, accession_no, avail) VALUES

(1, 101, 'T'),

(2, 102, 'T'),

(3, 103, 'T'),

(4, 104, 'T'),

(5, 105, 'T'),

(6, 106, 'F'),

(7, 107, 'T'),

(8, 108, 'F'),

(9, 109, 'T'),

(10, 110, 'T');

-- Inserting into ISSUETABLE

INSERT INTO ISSUETABLE (issueid, accession_no, enrollno, issuedate, cluedate, ret_date, bid) VALUES

(1, 101, 1, '2025-04-01', '2025-04-08', '2025-04-10', 1),

(2, 102, 3, '2025-04-02', '2025-04-09', '2025-04-12', 2),

(3, 103, 5, '2025-04-03', '2025-04-10', '2025-04-13', 3),

(4, 104, 2, '2025-04-04', '2025-04-11', '2025-04-14', 4),

(5, 105, 7, '2025-04-05', '2025-04-12', '2025-04-15', 5),

(6, 106, 9, '2025-04-06', '2025-04-13', '2025-04-16', 6),

(7, 107, 8, '2025-04-07', '2025-04-14', '2025-04-17', 7),

(8, 108, 4, '2025-04-08', '2025-04-15', '2025-04-18', 8),

(9, 109, 6, '2025-04-09', '2025-04-16', '2025-04-19', 9),

(10, 110, 10, '2025-04-10', '2025-04-17', '2025-04-20', 10);

3. Queries

a) Find the name of books which are issued the maximum times.

```
SELECT b.title, COUNT(i.issueid) AS issue_count

FROM BOOKMASTER b

JOIN ISSUETABLE i ON b.bid = i.bid

GROUP BY b.title

ORDER BY issue_count DESC

LIMIT 1;
```

b) Find the detailed information of books that are issued by computer department students.

```
SELECT b.*, s.sname, s.class, s.dept, i.issuedate, i.cluedate, i.ret_date

FROM BOOKMASTER b

JOIN ISSUETABLE i ON b.bid = i.bid
```

```
JOIN STUDENTMASTER s ON i.enrollno = s.enrollno
WHERE s.dept = 'Computer';
```

c) Write a procedure to calculate the fines for the books which are not returned on or before the due date

```
CREATE PROCEDURE CalculateFines()
AS
BEGIN
    DECLARE @fine INT, @no_of_days INT;

    -- Declare a cursor to fetch records
    DECLARE book_cursor CURSOR FOR
    SELECT issueid, ret_date, cluedate
    FROM ISSUETABLE
    WHERE ret_date > cluedate;

    OPEN book_cursor;

    FETCH NEXT FROM book_cursor INTO @issueid,
@ret_date, @cluedate;

    -- Loop through each record and calculate fine
    WHILE @@FETCH_STATUS = 0
```

```
    BEGIN
        SET @no_of_days = DATEDIFF(DAY, @cluedate,
@ret_date);
        SET @fine = @no_of_days * 10;


        PRINT 'Issue ID: ' + CAST(@issueid AS VARCHAR) + ' -
Fine: ' + CAST(@fine AS VARCHAR);


        FETCH NEXT FROM book_cursor INTO @issueid,
@ret_date, @cluedate;
    END;


    CLOSE book_cursor;
    DEALLOCATE book_cursor;
END;
```

4. Data Report: Display all books available in the library.

```
SELECT b.title, b.author, a.avail
FROM BOOKMASTER b
JOIN ACCESSIONTABLE a ON b.bid = a.bid
WHERE a.avail = 'T';
```

Q-16) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table).

BOOKMASTER (bid, title, author, price) STUDENTMASTER (stud enrollno, snaine,class,dept) ACCESSIONTABLE (accession no,avail) fiMUETABLE(issueid,as_ussion no,stud earalno,issuedate,duedate, ret_date,bi,d)

Integrity Constraints:

• The values of any attributes should not be null. • Avail should be T (if book is not issue) or 1' (if book is issue)

Queries:

a) Find the detail information of the students who have issued books Between two given dates. b) Create a view that display all the accession infortnatiOn for a book having bid = 100 c) Write a cursor to fetch last record from view in (b). d) Find the information of books issued by MCA students.

Create a input form for new book issue. Apply all possible validations.

->

1: Create Tables with Integrity Constraints

```
-- BOOKMASTER
CREATE TABLE BOOKMASTER (
    bid INT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(100) NOT NULL,
    price DECIMAL(10,2) NOT NULL
);

-- STUDENTMASTER
CREATE TABLE STUDENTMASTER (
    stud_enrollno VARCHAR(20) PRIMARY KEY,
    sname VARCHAR(100) NOT NULL,
    class VARCHAR(10) NOT NULL,
    dept VARCHAR(50) NOT NULL
);

-- ACCESSIONTABLE
CREATE TABLE ACCESSIONTABLE (
    accession_no INT PRIMARY KEY,
    avail CHAR(1) CHECK (avail IN ('T', 'F')) NOT NULL
```

```
);

-- ISSUETABLE
CREATE TABLE ISSUETABLE (
    issueid INT PRIMARY KEY,
    accession_no INT NOT NULL,
    stud_enrollno VARCHAR(20) NOT NULL,
    issuedate DATE NOT NULL,
    duedate DATE NOT NULL,
    ret_date DATE NOT NULL,
    bid INT NOT NULL,
    FOREIGN KEY (accession_no) REFERENCES
ACCESSIONTABLE(accession_no),
    FOREIGN KEY (stud_enrollno) REFERENCES
STUDENTMASTER(stud_enrollno),
    FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)
);
```

2: Insert Sample Data (10 records each)

```
-- BOOKMASTER
INSERT INTO BOOKMASTER VALUES
(100, 'Data Structures', 'Mark Weiss', 450.50),
(101, 'DBMS Concepts', 'Silberschatz', 499.00),
```

```
(102, 'Operating Systems', 'Galvin', 599.00),
(103, 'Computer Networks', 'Tanenbaum', 550.00),
(104, 'AI Basics', 'Stuart Russell', 670.00),
(105, 'Machine Learning', 'Tom Mitchell', 720.00),
(106, 'Python Programming', 'Zed Shaw', 300.00),
(107, 'Java Complete Ref', 'Herbert Schildt', 650.00),
(108, 'Web Dev', 'Jon Duckett', 400.00),
(109, 'Cyber Security', 'William Stallings', 800.00);

-- STUDENTMASTER
INSERT INTO STUDENTMASTER VALUES
('S001', 'Alice Smith', 'FY', 'MCA'),
('S002', 'Bob Brown', 'SY', 'MCA'),
('S003', 'Charlie Day', 'TY', 'MBA'),
('S004', 'David Johnson', 'FY', 'MCA'),
('S005', 'Eva Green', 'TY', 'MSc'),
('S006', 'Frank Wright', 'SY', 'MBA'),
('S007', 'Grace Hopper', 'TY', 'MCA'),
('S008', 'Henry Ford', 'FY', 'MSc'),
('S009', 'Irene Adler', 'SY', 'MCA'),
('S010', 'Jack Ryan', 'TY', 'MBA');
```

```
-- ACCESSIONTABLE
INSERT INTO ACCESSIONTABLE VALUES
(1, 'T'), (2, 'F'), (3, 'T'), (4, 'F'), (5, 'T'),
(6, 'T'), (7, 'F'), (8, 'T'), (9, 'T'), (10, 'F');

-- ISSUETABLE
INSERT INTO ISSUETABLE VALUES
(201, 2, 'S001', '2024-04-01', '2024-04-15', '2024-04-10', 100),
(202, 4, 'S002', '2024-03-01', '2024-03-15', '2024-03-14', 101),
(203, 7, 'S004', '2024-04-02', '2024-04-12', '2024-04-11', 100),
(204, 10, 'S007', '2024-04-03', '2024-04-17', '2024-04-16',
104),
(205, 1, 'S003', '2024-04-04', '2024-04-18', '2024-04-15', 102),
(206, 3, 'S006', '2024-04-05', '2024-04-20', '2024-04-18', 103),
(207, 5, 'S009', '2024-04-06', '2024-04-21', '2024-04-20', 100),
(208, 6, 'S010', '2024-04-07', '2024-04-22', '2024-04-21', 101),
(209, 8, 'S008', '2024-04-08', '2024-04-23', '2024-04-22', 109),
(210, 9, 'S005', '2024-04-09', '2024-04-24', '2024-04-23', 108);
```

3: Required Queries

a) Students who have issued books between two given dates

```
DECLARE @startDate DATE = '2024-04-01';
```

```
DECLARE @endDate DATE = '2024-04-10';


SELECT s.*
FROM STUDENTMASTER s
JOIN ISSUETABLE i ON s.stud_enrollno = i.stud_enrollno
WHERE i.issuedate BETWEEN @startDate AND @endDate;
```

b) Create a view for accession info for bid = 100

```
CREATE VIEW Book100Accessions AS
SELECT a.accession_no, a.avail, i.bid
FROM ACCESSIONTABLE a
JOIN ISSUETABLE i ON a.accession_no = i.accession_no
WHERE i.bid = 100;
```

d) Information of books issued by MCA students

```
SELECT b.*
FROM BOOKMASTER b
JOIN ISSUETABLE i ON b.bid = i.bid
JOIN STUDENTMASTER s ON s.stud_enrollno =
i.stud_enrollno
WHERE s.dept = 'MCA';
```

Q-17) Create database using following schema. Apply given Integrity Constraints and answer the followingqueries using SQL. (Fill up database with at least 10 records ir+ eaoh,tab4e).

BOOKMASTER (bid, title, author, price) STUDENTMASTER (stud enroll no, sname,class,dept) ACCESSIONTABLE (bid, accession no,avail) SUETABLE(issueid,accessipn no.stl.KLmrsillag,issuedate,cluedate, ret_date,hisp

Integrity Constraints:

• The values of any attributes should not be null. • Avail should be T (if book is not issue) or F (if book i8 issue)

Queries:

a) Write a procedure for giving the detail information of books available in the library. b) Find the number of books issued by each student. c) Write a trigger such that the return date should not exceed today's date. d) Find the number of books available in the library & written by "Henry Korth".

Create a class wise issue report of books.

->

1. **Create Tables with Integrity Constraints**

```
-- BOOKMASTER table
CREATE TABLE BOOKMASTER (
    bid INT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(100) NOT NULL,
    price DECIMAL(8, 2) NOT NULL
);

-- STUDENTMASTER table
CREATE TABLE STUDENTMASTER (
    stud_enroll_no INT PRIMARY KEY,
    sname VARCHAR(100) NOT NULL,
    class VARCHAR(50) NOT NULL,
    dept VARCHAR(50) NOT NULL
);

-- ACCESSIONTABLE
CREATE TABLE ACCESSIONTABLE (
    bid INT NOT NULL,
    accession_no INT PRIMARY KEY,
```

```
    avail CHAR(1) CHECK (avail IN ('T', 'F')) NOT NULL,
    FOREIGN KEY (bid) REFERENCES BOOKMASTER(bid)
);

-- SUETABLE (assumed to mean "ISSUETABLE")
CREATE TABLE ISSUETABLE (
    issueid INT PRIMARY KEY,
    accession_no INT NOT NULL,
    stud_enroll_no INT NOT NULL,
    issuedate DATE NOT NULL,
    duedate DATE NOT NULL,
    ret_date DATE NOT NULL,
    FOREIGN KEY (accession_no) REFERENCES
ACCESSIONTABLE(accession_no),
    FOREIGN KEY (stud_enroll_no) REFERENCES
STUDENTMASTER(stud_enroll_no)
);
```

2. **Insert Sample Records (10 each)**

```
-- BOOKMASTER
INSERT INTO BOOKMASTER VALUES
(1, 'Database Systems', 'Henry Korth', 450.00),
(2, 'Operating Systems', 'Galvin', 500.00),
```

```
(3, 'Networking Concepts', 'Tanenbaum', 550.00),
(4, 'DBMS Concepts', 'Navathe', 480.00),
(5, 'Software Engineering', 'Sommerville', 600.00),
(6, 'Algorithms', 'Cormen', 700.00),
(7, 'Computer Architecture', 'Morris Mano', 650.00),
(8, 'AI Basics', 'Russell', 520.00),
(9, 'Java Programming', 'Herbert Schildt', 400.00),
(10, 'Database Design', 'Henry Korth', 470.00);

-- STUDENTMASTER
INSERT INTO STUDENTMASTER VALUES
(101, 'Alice', 'FY', 'CS'),
(102, 'Bob', 'FY', 'CS'),
(103, 'Charlie', 'SY', 'IT'),
(104, 'David', 'TY', 'IT'),
(105, 'Eve', 'SY', 'CS'),
(106, 'Frank', 'TY', 'CS'),
(107, 'Grace', 'FY', 'IT'),
(108, 'Hank', 'SY', 'CS'),
(109, 'Ivy', 'TY', 'CS'),
(110, 'Jack', 'FY', 'IT');
```

```
-- ACCESSIONTABLE
INSERT INTO ACCESSIONTABLE VALUES
(1, 1001, 'T'),
(1, 1002, 'F'),
(2, 1003, 'T'),
(3, 1004, 'F'),
(4, 1005, 'T'),
(5, 1006, 'T'),
(6, 1007, 'F'),
(7, 1008, 'T'),
(10, 1009, 'F'),
(10, 1010, 'T');

-- ISSUETABLE
INSERT INTO ISSUETABLE VALUES
(1, 1002, 101, '2025-04-01', '2025-04-10', '2025-04-09'),
(2, 1004, 102, '2025-04-03', '2025-04-12', '2025-04-11'),
(3, 1007, 103, '2025-04-05', '2025-04-15', '2025-04-14'),
(4, 1009, 104, '2025-04-06', '2025-04-16', '2025-04-15');
```

3. Queries

a) Procedure to get available books

```
CREATE PROCEDURE GetAvailableBooks
```

```
AS
BEGIN
    SELECT B.bid, B.title, B.author, B.price, A.accession_no
    FROM BOOKMASTER B
    JOIN ACCESSIONTABLE A ON B.bid = A.bid
    WHERE A.avail = 'T';
END;
```

b) Number of books issued by each student

```
SELECT S.sname, S.stud_enroll_no, COUNT(I.issueid) AS
books_issued
FROM STUDENTMASTER S
LEFT JOIN ISSUETABLE I ON S.stud_enroll_no =
I.stud_enroll_no
GROUP BY S.sname, S.stud_enroll_no;
```

c) Count of available books by "Henry Korth"

```
SELECT COUNT(*) AS AvailableBooksByHenryKorth
FROM BOOKMASTER B
JOIN ACCESSIONTABLE A ON B.bid = A.bid
WHERE B.author = 'Henry Korth' AND A.avail = 'T';
```

d) Class-wise issue report of books

```
SELECT SM.class, COUNT(I.issueid) AS total_issues
FROM STUDENTMASTER SM
JOIN ISSUETABLE I ON SM.stud_enroll_no =
I.stud_enroll_no
GROUP BY SM.class;
```

Q-18) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in •aa table). EMPLOYEE (fname, lname, ssn, sex, salary, joindate,superssn, dno,) DEPT (dname, dnum, mgrssn, dlocation) PROJECT (pname, pno, plocation, &lumber) WORK ON (ssn, pno, hours)

Integrity Constraints:

• The values of any attributes should not be null. • The deptno should less than 4 digit • ssn (social security_no of employee) • mgrssn(manager_social_security_no) • superssn(supervisor_social_security_no)

Queries:

a) For every project located in `jalgaon'. List the pno,the controlling detptno and dept manager last name. b) For each project on which more than two employee work, Find the pno, pnarne & no. of employees who work on the project. c) create a view that has the deptna.me,manager name 8z manager salary for every dept. d) Express the following constraint as SQL assertions - "salary of employee must not be greater than the salary of the manager of the dept".

Create a data entry for New employee entry. Apply all possible validations.

->

1. **Create Tables with Integrity Constraints**

```
-- DEPT Table
CREATE TABLE DEPT (
    dname VARCHAR(50) NOT NULL,
    dnum INT NOT NULL CHECK (dnum < 10000),
    mgrssn CHAR(9) NOT NULL,
    dlocation VARCHAR(50) NOT NULL,
    PRIMARY KEY (dnum)
);

-- EMPLOYEE Table
CREATE TABLE EMPLOYEE (
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    ssn CHAR(9) NOT NULL PRIMARY KEY,
    sex CHAR(1) NOT NULL CHECK (sex IN ('M', 'F')),
    salary DECIMAL(10,2) NOT NULL,
    joindate DATE NOT NULL,
    superssn CHAR(9) NOT NULL,
    dno INT NOT NULL,
    FOREIGN KEY (dno) REFERENCES DEPT(dnum),
```

```
    FOREIGN KEY (superssn) REFERENCES EMPLOYEE(ssn)
);

-- PROJECT Table
CREATE TABLE PROJECT (
    pname VARCHAR(50) NOT NULL,
    pno INT NOT NULL PRIMARY KEY,
    plocation VARCHAR(50) NOT NULL,
    dnum INT NOT NULL,
    FOREIGN KEY (dnum) REFERENCES DEPT(dnum)
);

-- WORKS_ON Table
CREATE TABLE WORKS_ON (
    ssn CHAR(9) NOT NULL,
    pno INT NOT NULL,
    hours DECIMAL(5,2) NOT NULL,
    PRIMARY KEY (ssn, pno),
    FOREIGN KEY (ssn) REFERENCES EMPLOYEE(ssn),
    FOREIGN KEY (pno) REFERENCES PROJECT(pno)
);
```

2. **Insert Sample Data (with at least 10 employees)**

```
-- DEPT Records
INSERT INTO DEPT VALUES
('HR', 101, '123456789', 'New York'),
('IT', 102, '987654321', 'Pune'),
('Finance', 103, '456123789', 'Mumbai');

-- EMPLOYEE Records
INSERT INTO EMPLOYEE VALUES
('John', 'Doe', '123456789', 'M', 80000, '2020-01-10',
'123456789', 101),
('Jane', 'Smith', '234567891', 'F', 90000, '2019-05-12',
'123456789', 101),
('Mike', 'Brown', '345678912', 'M', 75000, '2021-03-15',
'123456789', 101),
('Sara', 'Wilson', '456789123', 'F', 85000, '2020-07-20',
'987654321', 102),
('Tom', 'Taylor', '567891234', 'M', 72000, '2021-11-01',
'987654321', 102),
('Lucy', 'Moore', '678912345', 'F', 70000, '2022-01-10',
'987654321', 102),
('David', 'Clark', '789123456', 'M', 78000, '2020-09-23',
'456123789', 103),
('Emily', 'Davis', '891234567', 'F', 77000, '2022-05-30',
'456123789', 103),
```

('Chris', 'Evans', '912345678', 'M', 76000, '2023-04-15', '456123789', 103),

('Nina', 'Hall', '135792468', 'F', 80000, '2021-08-09', '456123789', 103);

-- PROJECT Records

INSERT INTO PROJECT VALUES

('HR System', 1, 'Jalgaon', 101),

('Payroll', 2, 'Pune', 103),

('Recruitment', 3, 'Jalgaon', 102),

('Audit', 4, 'Mumbai', 103);

-- WORKS_ON Records

INSERT INTO WORKS_ON VALUES

('123456789', 1, 20),

('234567891', 1, 25),

('345678912', 1, 10),

('456789123', 2, 30),

('567891234', 3, 15),

('678912345', 3, 18),

('789123456', 3, 20),

('891234567', 4, 22),

('912345678', 4, 25),

('135792468', 4, 24);

3. SQL Queries

(a) Projects in Jalgaon: Show pno, dnum, manager last name

```
SELECT P.pno, P.dnum, E.lname AS manager_last_name
FROM PROJECT P
JOIN DEPT D ON P.dnum = D.dnum
JOIN EMPLOYEE E ON D.mgrssn = E.ssn
WHERE P.plocation = 'Jalgaon';
```

(b) Projects with more than 2 employees: Show pno, pname, count

```
SELECT P.pno, P.pname, COUNT(W.ssn) AS num_employees
FROM WORKS_ON W
JOIN PROJECT P ON W.pno = P.pno
GROUP BY P.pno, P.pname
HAVING COUNT(W.ssn) > 2;
```

5. Insert New Employee (with Validations)

```
-- Assuming supervisor and manager already exist with ssn
= '123456789'


INSERT INTO EMPLOYEE (fname, lname, ssn, sex, salary,
joindate, superssn, dno)
```

VALUES ('Alice', 'Walker', '246813579', 'F', 78000, '2024-12-01', '123456789', 101);

Q-19) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). ACCOUNT (accno, open_date, acctype, balance) TRANSACTION (trans id, trans date, accno. trans_type, amount) CUSTOMER (cust id name, address, accno)

Integrity Constraints:

• The values of any attributes should not be null. • acctype value should be P(Personal) or J(Joint). • Accno should be less than 3 digits. • Trans_type should be C(Credit) or D(Debit)

Queries:

a) Find the details of customers who have opened the accounts within the period 25-3 2006 to 28-3-2006. b) Find the details of customers who have joint accounts & balance is less than 2 lakhs. c) Write a trigger TRANSACTION on table to calculate the current balance of the account on which transaction is made. (if trans_type = c then bal = bal + amt else if trans_type = d then bal bal — amt) d) write a cursor on CUSTOMER table to fetch the last row.

Create a data entry for New customer entry. Apply all possible validations**.**

->

1. Create Tables with Integrity Constraints

```
CREATE TABLE ACCOUNT (
   accno INT PRIMARY KEY CHECK (accno < 100),
   open_date DATE NOT NULL,
   acctype CHAR(1) NOT NULL CHECK (acctype IN ('P', 'J')),
   balance DECIMAL(18,2) NOT NULL
);

CREATE TABLE CUSTOMER (
   cust_id INT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   address VARCHAR(255) NOT NULL,
   accno INT NOT NULL FOREIGN KEY REFERENCES
ACCOUNT(accno)
);

CREATE TABLE TRANSACTION (
   trans_id INT PRIMARY KEY,
   trans_date DATE NOT NULL,
   accno INT NOT NULL FOREIGN KEY REFERENCES
ACCOUNT(accno),
```

```
    trans_type CHAR(1) NOT NULL CHECK (trans_type IN ('C',
'D')),
    amount DECIMAL(18,2) NOT NULL
);
```

2. Insert Sample Data (10+ rows)

```
-- ACCOUNT Table
INSERT INTO ACCOUNT VALUES
(1, '2006-03-25', 'P', 150000),
(2, '2006-03-26', 'J', 190000),
(3, '2006-03-27', 'J', 210000),
(4, '2006-03-28', 'P', 50000),
(5, '2006-03-20', 'J', 180000),
(6, '2006-03-29', 'P', 120000),
(7, '2006-03-25', 'P', 100000),
(8, '2006-03-24', 'J', 90000),
(9, '2006-03-23', 'P', 300000),
(10, '2006-03-27', 'J', 199999);

-- CUSTOMER Table
INSERT INTO CUSTOMER VALUES
(101, 'Alice', 'New York', 1),
(102, 'Bob', 'Los Angeles', 2),
```

```
(103, 'Carol', 'Chicago', 3),
(104, 'Dave', 'Houston', 4),
(105, 'Eve', 'Phoenix', 5),
(106, 'Frank', 'Philadelphia', 6),
(107, 'Grace', 'San Antonio', 7),
(108, 'Heidi', 'San Diego', 8),
(109, 'Ivan', 'Dallas', 9),
(110, 'Judy', 'San Jose', 10);

-- TRANSACTION Table
INSERT INTO TRANSACTION VALUES
(1, '2024-04-01', 1, 'C', 10000),
(2, '2024-04-02', 2, 'D', 5000),
(3, '2024-04-03', 3, 'C', 2000),
(4, '2024-04-04', 4, 'D', 10000),
(5, '2024-04-05', 5, 'C', 15000),
(6, '2024-04-06', 6, 'D', 8000),
(7, '2024-04-07', 7, 'C', 5000),
(8, '2024-04-08', 8, 'D', 2000),
(9, '2024-04-09', 9, 'C', 3000),
(10, '2024-04-10', 10, 'D', 4000);
```

3. Queries

a) Customers with accounts opened between 25-3-2006 and 28-3-2006

```
SELECT C.*
FROM CUSTOMER C
JOIN ACCOUNT A ON C.accno = A.accno
WHERE A.open_date BETWEEN '2006-03-25' AND '2006-03-28';
```

b) Joint account customers with balance < 200000

```
SELECT C.*
FROM CUSTOMER C
JOIN ACCOUNT A ON C.accno = A.accno
WHERE A.acctype = 'J' AND A.balance < 200000;
```

Q-20) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). PRODUCT (Maker, .Modelno, Type) PC(4.oc jeki,o, Speed, RAM, HD, CD, Price) LAPTOP (Modelno, Speed, RAM, HD, Price) PRINTER (Mosklm, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price. • The value for Maker in Product table can be IBM, Compaq,etc. • PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null. • Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find the manufacturers of color printers. b) Find the laptops whose speed is slower than that of any PC. c) Express the following constraint as SQL assertions - "No black & white printer should have price greater than color printers." d) write a trigger on PC & LAPTOP table such that the hard disk size should be greater than 20 GB

Design an input form for entering PRINTER data.

Apply possible validations.

->

1. **Create Database and Tables**

```
CREATE DATABASE ComputerStore;
GO

USE ComputerStore;
GO

-- Product Table
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY NOT NULL,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

-- PC Table
CREATE TABLE PC (
    Modelno INT PRIMARY KEY NOT NULL,
    Speed INT NOT NULL,       -- MHz
    RAM INT NOT NULL,         -- MB
    HD INT NOT NULL,          -- GB
```

```
    CD VARCHAR(10) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- LAPTOP Table
CREATE TABLE LAPTOP (
    Modelno INT PRIMARY KEY NOT NULL,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- PRINTER Table
CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY NOT NULL,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price DECIMAL(10, 2) NOT NULL,
```

```
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

2. **Insert Sample Records**

```
-- PRODUCT Table
INSERT INTO PRODUCT VALUES
('IBM', 1001, 'PC'),
('Compaq', 1002, 'Laptop'),
('HP', 1003, 'Printer'),
('Dell', 1004, 'PC'),
('Acer', 1005, 'Laptop'),
('Canon', 1006, 'Printer'),
('Lenovo', 1007, 'PC'),
('Sony', 1008, 'Laptop'),
('Samsung', 1009, 'Printer'),
('Asus', 1010, 'Laptop');

-- PC Table
INSERT INTO PC VALUES
(1001, 3000, 8, 500, '48x', 40000),
(1004, 3200, 16, 1000, '52x', 55000),
(1007, 2800, 4, 250, '40x', 30000);
```

```
-- LAPTOP Table
INSERT INTO LAPTOP VALUES
(1002, 2500, 8, 256, 50000),
(1005, 2400, 4, 128, 45000),
(1008, 2000, 4, 64, 35000),
(1010, 3000, 16, 512, 65000);

-- PRINTER Table
INSERT INTO PRINTER VALUES
(1003, 'T', 'laser', 15000),
(1006, 'F', 'ink-jet', 12000),
(1009, 'T', 'dot-matrix', 10000);
```

3. SQL Queries

a) Find the manufacturers of color printers.

```
SELECT DISTINCT p.Maker
FROM PRODUCT p
JOIN PRINTER pr ON p.Modelno = pr.Modelno
WHERE pr.Color = 'T';
```

b) Find the laptops whose speed is slower than that of any PC.

```
SELECT *
FROM LAPTOP
```

```
WHERE Speed < ALL (SELECT Speed FROM PC);
```

c) SQL Assertion: No black & white printer should have price greater than any color printer.

```
-- Check for invalid printers
SELECT *
FROM PRINTER bw
WHERE Color = 'F' AND Price > ANY (
   SELECT Price FROM PRINTER WHERE Color = 'T'
);
```

Q-21) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). PRODUCT (Maker, Modelno, Type) PC (IvIodeln, Speed, RAM, HD, CD, Price) LAPTOP (Rh, Speed, RAM, HD, Price) PRINTER (Model, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price. • The value for Maker in Product table can be IBM, Compaq,etc. • PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null. • Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find the manufacturers of color printers. b) Find the laptops whose speed is slower than that of any PC. C) Express the following constraint as SQL assertions - No black & white printer should have price greater than color printers." d) write a trigger on PC & LAPTOP table such that the hard disk size should be greater than 20 GB

Design an input form for entering PC data. Apply possible validations.

->

1. Create Tables with Integrity Constraints

```
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY NOT NULL,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

CREATE TABLE PC (
    Modeln INT PRIMARY KEY NOT NULL,
    Speed INT NOT NULL,          -- in MHz
    RAM INT NOT NULL,            -- in MB
    HD INT NOT NULL CHECK (HD > 20),  -- in GB, constraint for Trigger logic
    CD VARCHAR(10) NOT NULL,     -- e.g. '52x'
    Price MONEY NOT NULL,
    FOREIGN KEY (Modeln) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE LAPTOP (
    Rh INT PRIMARY KEY NOT NULL,
```

```
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL CHECK (HD > 20),
    Price MONEY NOT NULL,
    FOREIGN KEY (Rh) REFERENCES PRODUCT(Modelno)
);

CREATE TABLE PRINTER (
    Model INT PRIMARY KEY NOT NULL,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price MONEY NOT NULL,
    FOREIGN KEY (Model) REFERENCES PRODUCT(Modelno)
);
```

2. **Insert Sample Data (At least 10 records each)**

```
-- PRODUCT Table
INSERT INTO PRODUCT VALUES
('IBM', 101, 'PC'), ('Dell', 102, 'Laptop'), ('HP', 103, 'Printer'),
('Lenovo', 104, 'PC'), ('Asus', 105, 'Laptop'), ('Canon', 106, 'Printer'),
('Acer', 107, 'PC'), ('Toshiba', 108, 'Laptop'), ('Brother', 109, 'Printer'),
```

('Samsung', 110, 'PC'), ('Sony', 111, 'Laptop'), ('Epson', 112, 'Printer');

-- PC Table

INSERT INTO PC VALUES

(101, 2400, 2048, 120, '52x', 45000),

(104, 1800, 1024, 80, '48x', 32000),

(107, 2200, 2048, 250, '52x', 40000),

(110, 2000, 1024, 60, '40x', 30000),

(113, 3000, 4096, 500, '52x', 60000),

(114, 2600, 2048, 320, '48x', 50000),

(115, 2700, 4096, 500, '52x', 58000),

(116, 2800, 2048, 350, '52x', 55000),

(117, 2900, 2048, 400, '52x', 57000),

(118, 3100, 8192, 1000, '52x', 65000);

-- LAPTOP Table

INSERT INTO LAPTOP VALUES

(102, 1700, 1024, 120, 35000),

(105, 1600, 512, 60, 30000),

(108, 1400, 2048, 100, 32000),

(111, 2000, 4096, 250, 42000),

```
(119, 1900, 2048, 300, 40000),
(120, 1500, 1024, 80, 31000),
(121, 1350, 1024, 60, 29000),
(122, 2100, 2048, 120, 39000),
(123, 1800, 2048, 100, 35000),
(124, 2200, 4096, 500, 48000);

-- PRINTER Table
INSERT INTO PRINTER VALUES
(103, 'T', 'laser', 15000),
(106, 'F', 'dot-matrix', 8000),
(109, 'T', 'ink-jet', 12000),
(112, 'F', 'laser', 9000),
(125, 'T', 'ink-jet', 16000),
(126, 'F', 'dot-matrix', 7000),
(127, 'T', 'dry', 17000),
(128, 'F', 'ink-jet', 8500),
(129, 'F', 'laser', 9500),
(130, 'T', 'laser', 18000);
```

3. Queries

a) Find the manufacturers of color printers

```
SELECT DISTINCT p.Maker
```

```
FROM PRODUCT p
JOIN PRINTER pr ON p.Modelno = pr.Model
WHERE pr.Color = 'T';
```

b) Find the laptops whose speed is slower than that of any PC

```
SELECT *
FROM LAPTOP
WHERE Speed < ALL (SELECT Speed FROM PC);
```

c) SQL Assertion: No black & white printer should have price greater than any color printer

```
CREATE ASSERTION NoBWExpensive
CHECK (
   NOT EXISTS (
      SELECT *
      FROM PRINTER bw, PRINTER clr
      WHERE bw.Color = 'F' AND clr.Color = 'T'
      AND bw.Price > clr.Price
   )
);
```

Q-22) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). PRODUCT (Maker, Modelno, Type) PC (Modelno, Speed, RAM, HD, CD, Price) LAPTOP (Modelno, Speed, RAM, HD, Price) PRINTER (Modelno, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price. • The value for Maker in Product table can be IBM, Compaq,etc. • PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null. • Product Type should one of these (PC, Laptop or Printer).

Queries:

a) Find the different types of printers produced by Epson. b) Find those hard disk sizes which occur in two or more PC's. c) Write a trigger on LAPTOP table such that the minimum speed should be 120MHz. d) Demonstrate the use of cursor using PRODUCT table.

Design an input form for entering LAPTOP data. Apply possible validations.

->

1. Create Database and Tables (with Constraints)

```
-- Create the database
CREATE DATABASE ProductDB;
GO

USE ProductDB;
GO

-- PRODUCT table
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY NOT NULL,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

-- PC table
CREATE TABLE PC (
    Modelno INT PRIMARY KEY NOT NULL,
    Speed INT NOT NULL,       -- MHz
    RAM INT NOT NULL,         -- MB
```

```
    HD INT NOT NULL,          -- GB
    CD VARCHAR(10) NOT NULL,  -- e.g., '52x', '16x'
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- LAPTOP table
CREATE TABLE LAPTOP (
    Modelno INT PRIMARY KEY NOT NULL,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- PRINTER table
CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY NOT NULL,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
```

```
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

2. Insert Sample Data (10 records per table)

```
-- PRODUCT
INSERT INTO PRODUCT VALUES
('IBM', 101, 'PC'), ('Compaq', 102, 'PC'), ('HP', 103, 'Laptop'),
('Dell', 104, 'Printer'), ('Lenovo', 105, 'Laptop'), ('Epson', 106, 'Printer'),
('HP', 107, 'PC'), ('IBM', 108, 'Laptop'), ('Epson', 109, 'Printer'),
('Dell', 110, 'PC');

-- PC
INSERT INTO PC VALUES
(101, 2200, 4096, 500, '52x', 450.00),
(102, 2400, 2048, 320, '48x', 400.00),
(107, 2000, 2048, 500, '52x', 420.00),
(110, 2600, 8192, 1000, '16x', 700.00),
(111, 2100, 1024, 320, '48x', 350.00),
(112, 2200, 2048, 500, '52x', 440.00),
(113, 2300, 4096, 500, '52x', 460.00),
(114, 2500, 4096, 1000, '24x', 600.00),
```

```
(115, 2600, 8192, 1000, '24x', 650.00),
(116, 2400, 4096, 750, '52x', 620.00);

-- LAPTOP
INSERT INTO LAPTOP VALUES
(103, 1800, 4096, 500, 800.00),
(105, 1600, 2048, 320, 650.00),
(108, 2000, 8192, 1000, 1200.00),
(117, 2200, 4096, 500, 850.00),
(118, 2300, 4096, 750, 870.00),
(119, 2500, 8192, 1000, 950.00),
(120, 2400, 6144, 750, 900.00),
(121, 2600, 8192, 1000, 1100.00),
(122, 2800, 16384, 2000, 1400.00),
(123, 3000, 16384, 2000, 1500.00);

-- PRINTER
INSERT INTO PRINTER VALUES
(104, 'T', 'laser', 250.00),
(106, 'F', 'dot-matrix', 100.00),
(109, 'T', 'ink-jet', 150.00),
(124, 'F', 'laser', 180.00),
```

(125, 'T', 'dry', 200.00),

(126, 'F', 'dot-matrix', 120.00),

(127, 'T', 'ink-jet', 140.00),

(128, 'F', 'laser', 130.00),

(129, 'T', 'dry', 210.00),

(130, 'F', 'dot-matrix', 110.00);

3. Queries

(a) Find the different types of printers produced by Epson:

```
SELECT DISTINCT P.Type
FROM PRINTER PR
JOIN PRODUCT P ON PR.Modelno = P.Modelno
WHERE P.Maker = 'Epson';
```

(b) Find those hard disk sizes which occur in two or more PCs:

```
SELECT HD
FROM PC
GROUP BY HD
HAVING COUNT(*) >= 2;
```

Q-23) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). *PRODUCT (Maker, Modelno, Type) PC (Modelno, Speed, RAM, HD, CD, Price) LAPTOP (Modelno, Speed, RAM, HD, Price) PRINTER (1Viodelno, Color, Type, Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price. • The value for Maker in Product table can be IBM, Compacketc. • PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null. • Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find the different types of printers produced by Epson. b) Find those hard disk sizes which occur in two or more PC's. c) Write a trigger on LAPTOP table such that the minimum speed should be 1201v1Hz. d) Demonstrate the use of cursor using PRODUCT table.

Design an input form for entering LAPTOP data. Apply possible validations.

->

1. Creating the Database and Tables with Constraints

```
CREATE DATABASE ElectronicsDB;
GO

USE ElectronicsDB;
GO

-- PRODUCT Table
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY,
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

-- PC Table
CREATE TABLE PC (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL, -- in MHz
    RAM INT NOT NULL,   -- in MB
    HD INT NOT NULL,    -- in GB
```

```
    CD VARCHAR(20) NOT NULL, -- e.g., '24x', '48x'
    Price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- LAPTOP Table
CREATE TABLE LAPTOP (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- PRINTER Table
CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price DECIMAL(10,2) NOT NULL,
```

```
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

2. Inserting Sample Data (At Least 10 Records per Table)

```
-- PRODUCT Records
INSERT INTO PRODUCT VALUES
('IBM', 101, 'PC'),
('HP', 102, 'PC'),
('Dell', 103, 'Laptop'),
('Lenovo', 104, 'Laptop'),
('Epson', 105, 'Printer'),
('Epson', 106, 'Printer'),
('Asus', 107, 'Laptop'),
('HP', 108, 'PC'),
('Canon', 109, 'Printer'),
('IBM', 110, 'PC'),
('Acer', 111, 'Laptop'),
('Epson', 112, 'Printer');

-- PC Records
INSERT INTO PC VALUES
(101, 2400, 4096, 500, '48x', 400.00),
(102, 3000, 8192, 1000, '52x', 600.00),
```

```
(108, 2600, 4096, 500, '48x', 450.00),
(110, 2800, 8192, 1000, '52x', 700.00),
(113, 3200, 16384, 2000, '52x', 850.00),
(114, 2200, 2048, 500, '24x', 300.00),
(115, 2400, 4096, 1000, '48x', 550.00),
(116, 2600, 4096, 500, '48x', 450.00),
(117, 2800, 8192, 500, '52x', 500.00),
(118, 3000, 8192, 1000, '52x', 620.00);

-- LAPTOP Records
INSERT INTO LAPTOP VALUES
(103, 2500, 4096, 256, 600.00),
(104, 2400, 8192, 512, 750.00),
(107, 2600, 8192, 256, 700.00),
(111, 2200, 4096, 512, 550.00),
(119, 2400, 4096, 512, 580.00),
(120, 2600, 8192, 512, 620.00),
(121, 2800, 16384, 1000, 1000.00),
(122, 2300, 4096, 256, 490.00),
(123, 2400, 8192, 512, 610.00),
(124, 2500, 8192, 512, 630.00);
```

```
-- PRINTER Records
INSERT INTO PRINTER VALUES
(105, 'T', 'ink-jet', 150.00),
(106, 'F', 'laser', 120.00),
(109, 'T', 'ink-jet', 200.00),
(112, 'F', 'dot-matrix', 100.00),
(125, 'F', 'laser', 110.00),
(126, 'T', 'ink-jet', 130.00),
(127, 'T', 'dry', 250.00),
(128, 'F', 'dot-matrix', 90.00),
(129, 'F', 'laser', 105.00),
(130, 'T', 'laser', 300.00);
```

3. Queries

a) Find the different types of printers produced by Epson.

```
SELECT DISTINCT PR.Type
FROM PRODUCT P
JOIN PRINTER PR ON P.Modelno = PR.Modelno
WHERE P.Maker = 'Epson';
```

b) Find those hard disk sizes which occur in two or more PCs.

```
SELECT HD
FROM PC
GROUP BY HD
```

```
HAVING COUNT(*) >= 2;
```

Q-24) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL.(Fill up database with at least 10 records in each table). PRODUCT(Maker, Modelno, Type) PC (Modelno, Speed, RAM, HD, CD, Price) LAPTOP (Modetno, Speed, RAM, HD, Price) PRINTER (ModelaQ, Color, Type. Price)

Details regarding Schemas

• PC relation contains model no. of PC, its speed in MHz, RAM in MB, HD size in GB, Speed of CD reader, and price. • The value for Maker in Product table can be 113M, Compaq,etc. • PRINTER relation contains model no., value of Color should be T(if printer is color) or F (if printer is black & white), type(laser, ink-jet, dot- matrix or dry), and price.

Integrity Constraints:

• The values of any attributes should not be null. • Product Type should one of these (PC, Laptop or Printer)

Queries:

a) Find PC models having a speed of at least 150 MHz. b) Find those manufacturers that sell Laptops, but not PC's. c) Write a trigger on LAPTOP table such that the price should not less than 30000 d) Write a procedure to find the manufacturer who has produced the most expensive laptop.

Design an input form for entering LAPTOP data. Apply possible validations.

->

1. Create Tables with Integrity Constraints (SQL Server Syntax)

```
-- PRODUCT table
CREATE TABLE PRODUCT (
    Maker VARCHAR(50) NOT NULL,
    Modelno INT PRIMARY KEY,
    Type VARCHAR(10) NOT NULL CHECK (Type IN ('PC', 'Laptop', 'Printer'))
);

-- PC table
CREATE TABLE PC (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    CD VARCHAR(10) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

```
-- LAPTOP table
CREATE TABLE LAPTOP (
    Modelno INT PRIMARY KEY,
    Speed INT NOT NULL,
    RAM INT NOT NULL,
    HD INT NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);

-- PRINTER table
CREATE TABLE PRINTER (
    Modelno INT PRIMARY KEY,
    Color CHAR(1) NOT NULL CHECK (Color IN ('T', 'F')),
    Type VARCHAR(20) NOT NULL CHECK (Type IN ('laser', 'ink-jet', 'dot-matrix', 'dry')),
    Price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (Modelno) REFERENCES PRODUCT(Modelno)
);
```

2. Insert Sample Data (10 Records Each)

```
-- PRODUCT
INSERT INTO PRODUCT VALUES
```

('HP', 1001, 'PC'), ('HP', 1002, 'Laptop'), ('Dell', 1003, 'Printer'),

('Lenovo', 1004, 'PC'), ('Lenovo', 1005, 'Laptop'), ('Canon', 1006, 'Printer'),

('HP', 1007, 'PC'), ('Asus', 1008, 'Laptop'), ('Samsung', 1009, 'Printer'),

('Apple', 1010, 'Laptop');

-- PC

INSERT INTO PC VALUES

(1001, 200, 4, 500, '52X', 40000),

(1004, 150, 8, 256, '48X', 35000),

(1007, 220, 16, 1024, '52X', 60000),

(1011, 180, 4, 320, '50X', 42000),

(1012, 160, 8, 500, '48X', 45000),

(1013, 170, 4, 250, '52X', 33000),

(1014, 155, 2, 120, '40X', 30000),

(1015, 210, 8, 1000, '52X', 50000),

(1016, 190, 16, 512, '48X', 38000),

(1017, 160, 4, 400, '52X', 34000);

-- LAPTOP

INSERT INTO LAPTOP VALUES

```
(1002, 250, 8, 512, 50000),
(1005, 200, 4, 256, 35000),
(1008, 180, 8, 500, 38000),
(1010, 300, 16, 1024, 90000),
(1018, 220, 8, 512, 45000),
(1019, 240, 4, 500, 43000),
(1020, 260, 16, 1000, 87000),
(1021, 200, 4, 256, 36000),
(1022, 280, 16, 2048, 95000),
(1023, 300, 32, 2048, 120000);

-- PRINTER
INSERT INTO PRINTER VALUES
(1003, 'T', 'laser', 15000),
(1006, 'F', 'dot-matrix', 8000),
(1009, 'T', 'ink-jet', 12000),
(1024, 'F', 'dry', 9000),
(1025, 'T', 'laser', 18000),
(1026, 'F', 'dot-matrix', 7000),
(1027, 'T', 'ink-jet', 13000),
(1028, 'F', 'laser', 14000),
(1029, 'T', 'dry', 16000),
```

(1030, 'F', 'dot-matrix', 7500);

3. Queries

a) PC models having a speed of at least 150 MHz

```
SELECT Modelno, Speed FROM PC WHERE Speed >= 150;
```

b) Manufacturers that sell laptops but not PCs

```
SELECT DISTINCT Maker
FROM PRODUCT
WHERE Type = 'Laptop'
AND Maker NOT IN (
   SELECT DISTINCT Maker FROM PRODUCT WHERE Type = 'PC'
);
```

Q-25) Create database using following schema. Apply given Integrity Constraints and answer the following queries using SQL. (Fill up database with at least 10 records in each table). DOCTOR (Did, Dname, Daddress, qualification) PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) ADMITTEDPATIENT (Pcode, Entry date, Discharge_clate, wardno, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be M (male) or F(feinale). • Wardno should be less than 6.

Queries:

a) Find the details of patient who are admitted within the period 03/03/08 to 25/ 03/08. b) Find the names of doctors who are treating Jalf_zaon patients. c) write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients currently admitted in the hospital. (bill = no _ of_ days * 500) d) Write a trigger on Doctor table such that the specialization should be :- M.B.B.S./B.A.M.S/M.S.

Create a data entry form for new DOCTOR. Apply all possible validations.

->

1: Create the Tables with Integrity Constraints

```
-- Create Doctor Table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY, -- Doctor ID
    Dname VARCHAR(100) NOT NULL, -- Doctor Name
    Daddress VARCHAR(255) NOT NULL, -- Doctor Address
    qualification VARCHAR(50) NOT NULL CHECK (qualification IN ('M.B.B.S.', 'B.A.M.S', 'M.S.')) -- Qualification (with validation)
);

-- Create Patient Master Table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY, -- Patient Code
    Pname VARCHAR(100) NOT NULL, -- Patient Name
    Padd VARCHAR(255) NOT NULL, -- Patient Address
    age INT NOT NULL CHECK (age > 0), -- Patient Age (must be greater than 0)
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F')), -- Gender must be 'M' or 'F'
    bloodgroup VARCHAR(3) NOT NULL, -- Blood group
    Did INT NOT NULL, -- Doctor ID (foreign key)
```

```
    FOREIGN KEY (Did) REFERENCES DOCTOR(Did)
);


-- Create Admitted Patient Table
CREATE TABLE ADMITTEDPATIENT (
    Pcode INT, -- Patient Code (foreign key)
    Entry_date DATE NOT NULL, -- Entry Date
    Discharge_date DATE NOT NULL, -- Discharge Date
    wardno INT NOT NULL CHECK (wardno < 6), -- Ward Number (less than 6)
    disease VARCHAR(100) NOT NULL, -- Disease
    PRIMARY KEY (Pcode, Entry_date), -- Composite primary key
    FOREIGN KEY (Pcode) REFERENCES PATIENTMASTER(Pcode)
);
```

2: Insert Sample Data (at least 10 records in each table)

```
-- Insert sample data into DOCTOR table
INSERT INTO DOCTOR (Did, Dname, Daddress, qualification) VALUES
(1, 'Dr. John Smith', '123 Main St, City', 'M.B.B.S.'),
(2, 'Dr. Emily Johnson', '456 Oak Rd, City', 'M.S.'),
(3, 'Dr. Robert Brown', '789 Pine Ave, City', 'B.A.M.S'),
```

(4, 'Dr. Sarah Davis', '321 Maple Blvd, City', 'M.B.B.S.'),

(5, 'Dr. Michael Wilson', '654 Cedar Ln, City', 'M.S.'),

(6, 'Dr. Jessica Moore', '987 Birch St, City', 'B.A.M.S'),

(7, 'Dr. David Clark', '135 Elm Dr, City', 'M.B.B.S.'),

(8, 'Dr. Patricia Lewis', '246 Oak St, City', 'M.S.'),

(9, 'Dr. Brian Walker', '579 Pine Rd, City', 'B.A.M.S'),

(10, 'Dr. Lisa Adams', '864 Willow Ln, City', 'M.B.B.S.');

-- Insert sample data into PATIENTMASTER table

INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) VALUES

(101, 'Alice Green', '1234 Rose St', 29, 'F', 'O+', 1),

(102, 'Bob White', '5678 Lily Ave', 35, 'M', 'A-', 2),

(103, 'Charlie Black', '9101 Tulip Rd', 41, 'M', 'B+', 3),

(104, 'David Grey', '1122 Orchid Blvd', 25, 'M', 'AB+', 4),

(105, 'Eva Blue', '3344 Magnolia Ln', 33, 'F', 'O-', 5),

(106, 'Grace Yellow', '5566 Sunflower St', 27, 'F', 'A+', 6),

(107, 'Hannah Red', '7788 Daisy Dr', 31, 'F', 'B-', 7),

(108, 'Ian Silver', '9900 Maple St', 28, 'M', 'O+', 8),

(109, 'Jack Gold', '1122 Violet Ave', 39, 'M', 'AB-', 9),

(110, 'Kathy Brown', '4455 Rosewood Ln', 43, 'F', 'A-', 10);

-- Insert sample data into ADMITTEDPATIENT table

INSERT INTO ADMITTEDPATIENT (Pcode, Entry_date, Discharge_date, wardno, disease) VALUES

(101, '2008-03-03', '2008-03-10', 1, 'Flu'),

(102, '2008-03-05', '2008-03-12', 2, 'Malaria'),

(103, '2008-03-07', '2008-03-15', 3, 'Dengue'),

(104, '2008-03-10', '2008-03-17', 1, 'Pneumonia'),

(105, '2008-03-12', '2008-03-20', 4, 'Typhoid'),

(106, '2008-03-14', '2008-03-21', 5, 'Chickenpox'),

(107, '2008-03-18', '2008-03-22', 2, 'Tuberculosis'),

(108, '2008-03-20', '2008-03-25', 3, 'Bronchitis'),

(109, '2008-03-22', '2008-03-28', 4, 'Diabetes'),

(110, '2008-03-23', '2008-03-30', 5, 'Asthma');

3: Write the SQL Queries

Query (a): Find the details of patients admitted between '03/03/08' and '25/03/08'

SELECT *

FROM ADMITTEDPATIENT ap

JOIN PATIENTMASTER pm ON ap.Pcode = pm.Pcode

WHERE ap.Entry_date BETWEEN '2008-03-03' AND '2008-03-25';

Query (b): Find the names of doctors treating 'Jalf_zaon' patients

```
SELECT DISTINCT d.Dname
FROM DOCTOR d
JOIN PATIENTMASTER pm ON d.Did = pm.Did
JOIN ADMITTEDPATIENT ap ON pm.Pcode = ap.Pcode
WHERE ap.disease = 'Jaundice';
```

Query (c): Procedure to calculate the bill for currently admitted patients

```
CREATE PROCEDURE CalculateBillForAdmittedPatients
AS
BEGIN
    DECLARE @Pcode INT;
    DECLARE @Entry_date DATE;
    DECLARE @Discharge_date DATE;
    DECLARE @NoOfDays INT;
    DECLARE @BillAmount INT;

    DECLARE patient_cursor CURSOR FOR
    SELECT Pcode, Entry_date, Discharge_date
    FROM ADMITTEDPATIENT
    WHERE Discharge_date IS NULL;

    OPEN patient_cursor;
```

```
    FETCH NEXT FROM patient_cursor INTO @Pcode,
@Entry_date, @Discharge_date;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @NoOfDays = DATEDIFF(DAY, @Entry_date,
GETDATE());
        SET @BillAmount = @NoOfDays * 500;

        PRINT 'Patient ' + CAST(@Pcode AS VARCHAR) + ' Bill: ' +
CAST(@BillAmount AS VARCHAR);

        FETCH NEXT FROM patient_cursor INTO @Pcode,
@Entry_date, @Discharge_date;
    END

    CLOSE patient_cursor;
    DEALLOCATE patient_cursor;
END;
```

Q-26) answer the following queries using SQL. (Fill up database with at least 10 records in each table). DOCTOR (Did, Dname, Daddress, qualification) PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did) ADMITTEDPATIENT (Pcode, Entry date, Discharge_date, warcino, disease)

Integrity Constraints:

• The values of any attributes should not be null. • Gender value should be .M (male) or F(female). • Wardno should be less than 6.

Queries:

a) Find details of the patients who are treated by M.B.B.S. doctors. b) Find name of the doctor who are treating the male patients suffering from disease brain tumor & having age less than 40 years. c) write a procedure on ADMITTEDPATIENT table such as to calculate the bill of all patients who are discharged on 30-3-2008 . (bill no of days * 500) d) write a cursor on DOCTOR table to fetch the first row & display the number of rows present in the table.

Create a data entry form for new patient. Apply all possible validations.

->

1. Creating Tables with Integrity Constraints

```
-- Create DOCTOR table
CREATE TABLE DOCTOR (
    Did INT PRIMARY KEY,
    Dname VARCHAR(100) NOT NULL,
    Daddress VARCHAR(255) NOT NULL,
    qualification VARCHAR(100) NOT NULL
);

-- Create PATIENTMASTER table
CREATE TABLE PATIENTMASTER (
    Pcode INT PRIMARY KEY,
    Pname VARCHAR(100) NOT NULL,
    Padd VARCHAR(255) NOT NULL,
    age INT NOT NULL CHECK (age > 0),
    gender CHAR(1) NOT NULL CHECK (gender IN ('M', 'F')),
    bloodgroup VARCHAR(5) NOT NULL,
    Did INT,
    FOREIGN KEY (Did) REFERENCES DOCTOR(Did)
);
```

```
-- Create ADMITTEDPATIENT table
CREATE TABLE ADMITTEDPATIENT (
    Pcode INT,
    Entry_date DATE NOT NULL,
    Discharge_date DATE NOT NULL,
    warcino INT CHECK (warcino < 6),
    disease VARCHAR(100) NOT NULL,
    PRIMARY KEY (Pcode, Entry_date),
    FOREIGN KEY (Pcode) REFERENCES
PATIENTMASTER(Pcode)
);
```

2. Inserting Sample Data

```
-- Insert data into DOCTOR table
INSERT INTO DOCTOR (Did, Dname, Daddress, qualification)
VALUES
(1, 'Dr. John Doe', '123 Main St, City', 'MBBS'),
(2, 'Dr. Jane Smith', '456 Oak St, City', 'MBBS'),
(3, 'Dr. Sam Wilson', '789 Pine St, City', 'MD'),
(4, 'Dr. Mary Green', '101 Elm St, City', 'MBBS'),
(5, 'Dr. Alex Brown', '202 Maple St, City', 'MBBS'),
(6, 'Dr. Emily White', '303 Birch St, City', 'MD'),
(7, 'Dr. Peter Black', '404 Cedar St, City', 'MBBS'),
```

```
(8, 'Dr. Lucy Gray', '505 Willow St, City', 'MBBS'),
(9, 'Dr. Daniel Blue', '606 Cherry St, City', 'MD'),
(10, 'Dr. Sophia Red', '707 Pine St, City', 'MBBS');

-- Insert data into PATIENTMASTER table
INSERT INTO PATIENTMASTER (Pcode, Pname, Padd, age, gender, bloodgroup, Did)
VALUES
(1, 'Alice Brown', '123 Park Ave, City', 30, 'F', 'A+', 1),
(2, 'Bob White', '456 Oak St, City', 35, 'M', 'O-', 2),
(3, 'Charlie Green', '789 Pine St, City', 60, 'M', 'B+', 3),
(4, 'David Gray', '101 Elm St, City', 25, 'M', 'A+', 4),
(5, 'Eva Black', '202 Maple St, City', 40, 'F', 'AB-', 5),
(6, 'Frank Red', '303 Birch St, City', 45, 'M', 'O+', 6),
(7, 'Grace Blue', '404 Cedar St, City', 29, 'F', 'A-', 7),
(8, 'Hank White', '505 Willow St, City', 50, 'M', 'B-', 8),
(9, 'Ivy Yellow', '606 Cherry St, City', 60, 'F', 'O-', 9),
(10, 'Jack Purple', '707 Pine St, City', 28, 'M', 'AB+', 10);

-- Insert data into ADMITTEDPATIENT table
INSERT INTO ADMITTEDPATIENT (Pcode, Entry_date, Discharge_date, warcino, disease)
```

```
VALUES
(1, '2008-03-01', '2008-03-10', 1, 'Brain Tumor'),
(2, '2008-03-05', '2008-03-15', 2, 'Pneumonia'),
(3, '2008-03-10', '2008-03-20', 3, 'Heart Disease'),
(4, '2008-03-02', '2008-03-12', 4, 'Brain Tumor'),
(5, '2008-03-01', '2008-03-11', 5, 'Cancer'),
(6, '2008-03-07', '2008-03-17', 1, 'Brain Tumor'),
(7, '2008-03-01', '2008-03-10', 2, 'Asthma'),
(8, '2008-03-02', '2008-03-12', 3, 'Brain Tumor'),
(9, '2008-03-03', '2008-03-13', 4, 'Diabetes'),
(10, '2008-03-05', '2008-03-15', 5, 'Brain Tumor');
```

3. Queries

a) Find details of the patients who are treated by M.B.B.S. doctors.

```
SELECT PM.Pcode, PM.Pname, PM.Padd, PM.age, PM.gender,
PM.bloodgroup
FROM PATIENTMASTER PM
JOIN DOCTOR D ON PM.Did = D.Did
WHERE D.qualification = 'MBBS';
```

b) Find the name of the doctor who is treating male patients suffering from the disease "brain tumor" and having age less than 40 years.

```
SELECT D.Dname
```

```
FROM DOCTOR D
JOIN PATIENTMASTER PM ON D.Did = PM.Did
JOIN ADMITTEDPATIENT AP ON PM.Pcode = AP.Pcode
WHERE AP.disease = 'Brain Tumor'
AND PM.gender = 'M'
AND PM.age < 40;
```

c) Write a procedure on ADMITTEDPATIENT table to calculate the bill of all patients who are discharged on 30-3-2008. (Bill no of days * 500)

```
CREATE PROCEDURE
Calculate_Bill_For_Discharged_Patients
AS
BEGIN
   DECLARE @Pcode INT, @Entry_date DATE,
@Discharge_date DATE, @Bill INT;

   DECLARE patient_cursor CURSOR FOR
   SELECT Pcode, Entry_date, Discharge_date
   FROM ADMITTEDPATIENT
   WHERE Discharge_date = '2008-03-30';

   OPEN patient_cursor;
   FETCH NEXT FROM patient_cursor INTO @Pcode,
@Entry_date, @Discharge_date;

   WHILE @@FETCH_STATUS = 0
   BEGIN
      SET @Bill = DATEDIFF(DAY, @Entry_date,
@Discharge_date) * 500;
```

```
        PRINT 'Patient Code: ' + CAST(@Pcode AS
VARCHAR) + ' Bill: ' + CAST(@Bill AS VARCHAR);
        FETCH NEXT FROM patient_cursor INTO @Pcode,
@Entry_date, @Discharge_date;
    END;

    CLOSE patient_cursor;
    DEALLOCATE patient_cursor;
END;
```