

In the name of God
Database Lab
Spring 2016

Sajad AZAMI

May 20, 2016

Date Performed: May 20, 2016

1 Question 1:

Tables were created using queries shown below and have resulted the following ERD.

- SQL codes :

```
CREATE DATABASE DBLab1;

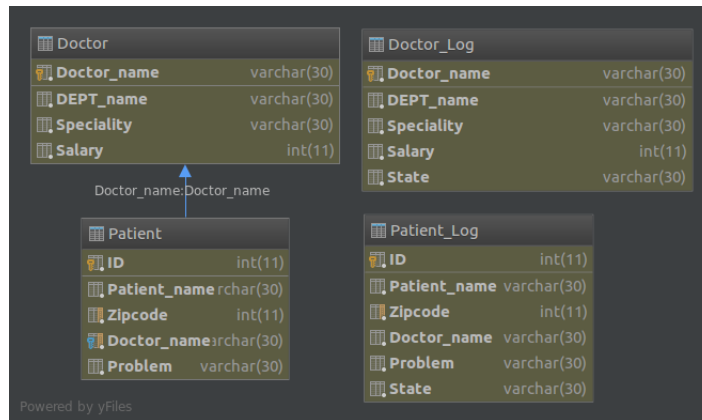
USE DBLab1;

CREATE TABLE `Patient`
(
  `ID` INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  `Patient_name` VARCHAR(30) NOT NULL,
  `Zipcode` INT UNIQUE NOT NULL,
  `Doctor_name` VARCHAR(30) NOT NULL,
  `Problem` VARCHAR(30) NOT NULL,
  FOREIGN KEY (Doctor_name)
  REFERENCES Doctor (Doctor_name)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);

CREATE TABLE `Doctor`
(
  `Doctor_name` VARCHAR(30) PRIMARY KEY NOT NULL,
  `DEPT_name` VARCHAR(30) NOT NULL,
  `Speciality` VARCHAR(30) NOT NULL,
  `Salary` INT NOT NULL
);

ALTER TABLE Doctor
ADD CONSTRAINT cnst1 CHECK (Salary > 5000)
```

- ERD:



2 Question 2:

- The procedure requested in question is written and code is shown below.

```
USE DBLab1;
CREATE PROCEDURE q1(IN id INT, name VARCHAR(30))
BEGIN
    DECLARE flag INT;
    IF EXISTS(SELECT *
              FROM Patient
              WHERE Patient.ID = id AND Doctor_name = name)
    THEN
        SET flag = 1;
    ELSE SET flag = 0;
    END IF;
    IF flag = 1
    THEN
        SELECT
            Doctor_name,
            Speciality
        FROM Doctor
        NATURAL JOIN Patient
        WHERE Patient.ID = id AND Patient_name = @name;
    ELSE
        INSERT INTO Patient VALUES (@id, @name, 12345, 'RandomDoctor', 'RandomProblem');
    END IF;
END
```

3 Question 3:

A function always returns a value but a procedure may return one or more value or may not return any value at all. The second difference is stored procedure returns always integer value by default zero. Whereas function returns type could be scalar or table or table values. Stored procedure is precompiled execution plan where as functions are not. Stored procedure has the security and reduces the network traffic and also we can call stored procedure in any no. of applications at a time. A Function can be used in the SQL Queries while a procedure cannot be used in SQL queries .that cause a major difference b/w

function and procedures.

- The function requested in question is written and code is shown below.

```
CREATE FUNCTION q3(patient_id INT)
  RETURNS VARCHAR(10)
  DETERMINISTIC
  BEGIN
    DECLARE spec VARCHAR(30);

    SET spec = (SELECT Speciality
                FROM Doctor
                NATURAL JOIN Patient
                WHERE ID = patient_id);

    RETURN (spec);
  END
```

4 Question 4:

- For this purpose, we need to add two more tables, so that we can insert logged data to them.

```
USE DBLab1;

CREATE TABLE `Patient_Log`
(
  `ID` INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  `Patient_name` VARCHAR(30) NOT NULL,
  `Zipcode` INT UNIQUE NOT NULL,
  `Doctor_name` VARCHAR(30) NOT NULL,
  `Problem` VARCHAR(30) NOT NULL,
  `State` VARCHAR(30) NOT NULL
);

CREATE TABLE `Doctor_Log`
(
  `Doctor_name` VARCHAR(30) PRIMARY KEY NOT NULL,
  `DEPT_name` VARCHAR(30) NOT NULL,
  `Speciality` VARCHAR(30) NOT NULL,
  `Salary` INT NOT NULL,
  `State` VARCHAR(30) NOT NULL
);
```

- Then triggers were created by code shown below.

```

CREATE TRIGGER logger BEFORE DELETE ON DBLab1.Patient
FOR EACH ROW
BEGIN
    INSERT INTO Patient_Log VALUES (Id, Patient_name, Zipcode, Doctor_name, Problem, 'Deleted');
END;

CREATE TRIGGER logger2 BEFORE DELETE ON DBLab1.Doctor
FOR EACH ROW
BEGIN
    INSERT INTO Doctor_Log VALUES (Doctor_name, DEPT_name, Speciality, Salary, 'Deleted');
END;

CREATE TRIGGER logger3 BEFORE INSERT ON DBLab1.Patient
FOR EACH ROW
BEGIN
    INSERT INTO Patient_Log VALUES (Id, Patient_name, Zipcode, Doctor_name, Problem, 'Inserted');
END;

CREATE TRIGGER logger4 BEFORE INSERT ON DBLab1.Doctor
FOR EACH ROW
BEGIN
    INSERT INTO Doctor_Log VALUES (Doctor_name, DEPT_name, Speciality, Salary, 'Inserted');
END;

```
