SET C

Student – Subject database

Consider the following database :

Student (rollno integer, name varchar(30),city varchar(50),class varchar(10))

Subject(Scodevarchar(10),subject name varchar(20))

Student and subject are related with M-M relationship with attributes marks_scored.

Create the above database in PostGreSQL and insert sufficient records

a. Write a trigger before insert/update the marks_scored. Raise exception if Marks are

negative.

b. Write a trigger which is executed when insertion is made in the student-subject table. If

marks_scored is less than 0, give appropriate message and do not allow the insertion.

c. Write a trigger which will prevent deleting students from ‘Pune’ city.

```
-- Create the Student Table
CREATE TABLE Student (
    rollno INTEGER PRIMARY KEY,
    name VARCHAR(30),
    city VARCHAR(50),
    class VARCHAR(10)
);

-- Create the Subject Table
CREATE TABLE Subject (
    Scode VARCHAR(10) PRIMARY KEY,
```

```
    subject_name VARCHAR(20)
);

-- Create the Many-to-Many Relationship Table with marks_scored
CREATE TABLE Student_Subject (
    rollno INTEGER,
    Scode VARCHAR(10),
    marks_scored INTEGER,
    PRIMARY KEY (rollno, Scode),
    FOREIGN KEY (rollno) REFERENCES Student (rollno),
    FOREIGN KEY (Scode) REFERENCES Subject (Scode)
);

-- Insert data into the Student table
INSERT INTO Student VALUES (1, 'Alice', 'Mumbai', '10A');
INSERT INTO Student VALUES(2, 'Bob', 'Pune', '10B');
INSERT INTO Student VALUES(3, 'Charlie', 'Delhi', '9A');
INSERT INTO Student VALUES(4, 'David', 'Pune', '10C');

-- Insert data into the Subject table
INSERT INTO Subject VALUES ('MATH01', 'Mathematics');
INSERT INTO Subject VALUES('SCI01', 'Science');
INSERT INTO Subject VALUES('ENG01', 'English');

-- Insert data into the Student_Subject relationship table
INSERT INTO Student_Subject VALUES (1, 'MATH01', 85);
INSERT INTO Student_Subject VALUES(2, 'SCI01', 90);
```

```
INSERT INTO Student_Subject VALUES(3, 'ENG01', 75);
INSERT INTO Student_Subject VALUES(4, 'MATH01', 88);
```

--Q1.Write a trigger before insert/update the marks_scored. Raise exception if Marks are negative.

```
CREATE OR REPLACE FUNCTION validate_marks()
RETURNS TRIGGER AS ‘
BEGIN
   IF NEW.marks_scored < 0 THEN
      RAISE EXCEPTION 'Marks cannot be negative for student % in subject %.', NEW.rollno, NEW.Scode;
   END IF;
   RETURN NEW;
END;
’ LANGUAGE plpgsql;
```

-- Create a trigger to execute the function before insert or update on Student_Subject table

```
CREATE TRIGGER marks_check_trigger
BEFORE INSERT OR UPDATE ON Student_Subject
FOR EACH ROW
EXECUTE FUNCTION validate_marks();
```

-- Test Trigger

```
INSERT INTO Student_Subject VALUES (1, 'MATH01', -5);
```

--Q3.Write a trigger which will prevent deleting students from ‘Pune’ city.

```
CREATE OR REPLACE FUNCTION prevent_deletion_pune_students()
RETURNS TRIGGER AS ‘
BEGIN
   IF OLD.city = 'Pune' THEN
      RAISE EXCEPTION 'Cannot delete student from Pune: %', OLD.name;
   END IF;
   RETURN OLD;
END;
’ LANGUAGE plpgsql;
```

-- Create a trigger to execute the function before deletion on Student table

```
CREATE TRIGGER prevent_delete_pune_students_trigger
BEFORE DELETE ON Student
FOR EACH ROW
EXECUTE FUNCTION prevent_deletion_pune_students();
```

-- Test The TRIGGER

```
DELETE FROM Student WHERE city = 'Pune' AND rollno = 2;
```