TASK – 9

1. Create a table named teachers with fields id, name, subject, experience and salary and insert 8 rows.

```sql
CREATE DATABASE school;
USE school;

 CREATE TABLE teachers (
  Id INT,
  Name VARCHAR(50),
  Subject VARCHAR(40),
  experience INT,
  salary BIGINT );

INSERT INTO teachers VALUES
(100, 'RESHMA', 'MATHEMATICS', 10, 100000),
(101, 'ARATHI', 'COMPUTER', 9, 90000),
(102, 'ABID', 'BIOLOGY', 10, 100000),
(103, 'SONA', 'CHEMISTRY', 11, 110000),
(104, 'JISHA', 'HINDI', 12, 120000),
(105, 'NIAM', 'MALAYALAM', 8, 80000),
(106, 'CHRISTINA', 'GEOLOGY', 6, 60000),
(107, 'JOSEPH', 'PHYSICS', 4, 40000);

SELECT * FROM teachers ;
```

2. Create a before insert trigger named before_insert_teacher that will raise an error "salary cannot be negative" if the salary inserted to the table is less than zero.

```
CREATE TABLE message (
    id INT AUTO_INCREMENT PRIMARY KEY,
    message_id INT,
    message VARCHAR(300));

DELIMITER //

CREATE TRIGGER before_insert_teacher
BEFORE INSERT ON teachers
FOR EACH ROW
BEGIN
    IF NEW.salary IS NULL THEN
        INSERT INTO message VALUES
        (NEW.id, CONCAT('Hi ', NEW.Name, ' SALARY CANNOT BE NEGATIVE'));
    END IF;
END //

DELIMITER ;
```

3. Create an after insert trigger named after_insert_teacher that inserts a row with teacher_id,action, timestamp to a table called teacher_log when a new entry gets inserted to the teacher table. tecaher_id -> column of teacher table, action -> the trigger action, timestamp -> time at which the new row has got inserted.

```
42  CREATE TABLE teacher_log (
43      teacher_id INT,
44      action VARCHAR(50),
45      timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
46  );
47
48  DELIMITER //
49
50  CREATE TRIGGER after_insert_teacher
51  AFTER INSERT ON teachers
52  FOR EACH ROW
53  BEGIN
54      INSERT INTO teacher_log VALUES
55      (NEW.Id, 'INSERT',NOW());
56  END;
57  END //
58
59  DELIMITER ;
```

4. Create a before delete trigger that will raise an error when you try to delete a row that has experience greater than 10 years.

```
CREATE TRIGGER before_delete_teacher
BEFORE DELETE ON teachers
FOR EACH ROW
BEGIN
    IF OLD.experience > 10 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'THIS TEACHER HAVE MORE THAN 10 YEAR OF EXP';
    END IF;
END;
END //

DELIMITER ;
```

5. Create an after delete trigger that will insert a row to teacher_log table when that row is deleted from teacher table.

```
DELIMITER //

CREATE TRIGGER after_delete_teacher
AFTER DELETE ON teachers
FOR EACH ROW
BEGIN
    INSERT INTO teacher_log  VALUES
    (OLD.Id, 'DELETE', NOW());
END;
END //

DELIMITER ;
```