**1. Creating the Table**

SQL

CREATE TABLE Student (

RollNo INT PRIMARY KEY,

Name VARCHAR(50),

Address VARCHAR(100),

Marks INT

);

**2. Creating Triggers**

a. After Insertion Trigger:

SQL

CREATE TRIGGER after\_insert\_student

AFTER INSERT ON Student

FOR EACH ROW

BEGIN

INSERT INTO Audit\_Log (Action, Table\_Name, Timestamp, RollNo, Name, Address, Marks)

VALUES ('INSERT', 'Student', NOW(), NEW.RollNo, NEW.Name, NEW.Address, NEW.Marks);

END;

b. After Update Trigger:

SQL

CREATE TRIGGER after\_update\_student

AFTER UPDATE ON Student

FOR EACH ROW

BEGIN

INSERT INTO Audit\_Log (Action, Table\_Name, Timestamp, RollNo, Old\_Name, Old\_Address, Old\_Marks, New\_Name, New\_Address, New\_Marks)

VALUES ('UPDATE', 'Student', NOW(), OLD.RollNo, OLD.Name, OLD.Address, OLD.Marks, NEW.Name, NEW.Address, NEW.Marks);

END;

c. After Deletion Trigger:

SQL

CREATE TRIGGER after\_delete\_student

AFTER DELETE ON Student

FOR EACH ROW

BEGIN

INSERT INTO Audit\_Log (Action, Table\_Name, Timestamp, RollNo, Name, Address, Marks)

VALUES ('DELETE', 'Student', NOW(), OLD.RollNo, OLD.Name, OLD.Address, OLD.Marks);

END;

d. Before Insertion Trigger:

SQL

CREATE TRIGGER before\_insert\_student

BEFORE INSERT ON Student

FOR EACH ROW

BEGIN

IF NEW.Marks < 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Marks cannot be negative';

END IF;

END;

**3. Creating Views**

a. View of Students with Marks above 80:

SQL

CREATE VIEW High\_Scoring\_Students AS

SELECT \*

FROM Student

WHERE Marks > 80;

b. View of Students and their Addresses:

SQL

CREATE VIEW Student\_Addresses AS

SELECT Name, Address

FROM Student;

To display the views:

SQL

SELECT \* FROM High\_Scoring\_Students;

SELECT \* FROM Student\_Addresses;

How to run:

1. Create table s1(roll, name), s2(roll, name), enter tuples into s1, s2 is empty.
2. Source ass7.sql
3. Select \* from s2.(run single trigger)