TRIGGER QUERIES, BEFORE AND INSTEAD OF.

BEFORE UPDATE

```
CREATE TRIGGER before_employee_update
BEFORE UPDATE ON employees
FOR EACH ROW
INSERT INTO Employees_Audit
SET action = 'update',
            id = OLD.id,
```

BEFORE INSERT

```
CREATE TRIGGER before_employees_insert
BEFORE INSERT
ON Employee FOR EACH ROW
BEGIN
   INSERT INTO Employee_Audit
   SET action = 'insert',
END $$
```

BEFORE DELETE

```
CREATE TRIGGER before_employees_delete
BEFORE DELETE
ON Employee FOR EACH ROW
BEGIN
   INSERT INTO Employee_Audit
   SET action = 'delete',
```

INSTEAD OF INSERT TRIGGER

```
USE EMPLOYEES;
SELECT * FROM EMPLOYEE;
CREATE TRIGGER id_sum BEFORE INSERT ON EMPLOYEE FOR EACH ROW SET @sum = @sum +
NEW.ID;
SET @sum = 0;
INSERT INTO EMPLOYEE VALUES(19,'WB','AM',103), (20, 'JK', 'IJ', 104), (21, 'GH','EF', 101), (22,
'AB','CD',102);
SELECT @sum AS 'Total sum ID';
```

INSTEAD OF UPDATE TRIGGER

```
CREATE TRIGGER empUpdate BEFORE UPDATE ON EMPLOYEE FOR EACH ROW
INSERT INTO EMP_AUDIT
SET ACTION = "update",
DEPT_ID = OLD.DEPT_ID,LASTNAME = OLD.LASTNAME,FIRSTNAME = OLD.FIRSTNAME,ID =
OLD.ID;
```

INSTEAD OF DELETE TRIGGER

```
create table EmployeeHistory(id integer Primary Key, firstNm varchar(30), lastNm varchar(30),
Department_ID integer)
CREATE TRIGGER empDelete BEFORE DELETE ON Employee FOR EACH ROW
INSERT INTO Employee_History VALUES(OLD.id,OLD.firstNm,OLD.lastNm,OLD.Department_ID);
```