**TRIGGERS**

It’s a procedure that gets invoked automatically when certain specified event occurs.

BEFORE and AFTER keyword is used to specify weather action to be taken before the event or after the event.

Trigger structure

Event – which events should invoke the trigger?

Condition – condition to be satisfied to perform the action

Action – action to be taken

Example:

**CREATE TRIGGER** SALARY\_VIOLATION

**BEFORE INSERT OR UPDATE OF** SALARY, //(event) When to trigger?

SUPERVISOR\_SSN **ON** EMPLOYEE

**FOR EACH ROW //**(condition)Condition to be checked

**WHEN** ( **NEW**.SALARY >

( **SELECT** SALARY **FROM** EMPLOYEE

**WHERE** SSN = **NEW**.SUPERVISOR\_SSN ) )

INFORM\_SUPERVISOR(**NEW**.Supervisor\_ssn,

**NEW**.Ssn ); //(action) Calling a function which is the action to be taken here

**VIEWS**

Instead of doing multiple join operations to retrieve data from different table at once,

We can alternatively create a virtual table holding values from other tables

Views are for viewing only.

Update operation won’t work here.

View table gets automatically updated when base table gets updated.

We can derive view from base tables as well as other views.

View table exist till we drop it.

Sample:

**CREATE VIEW** WORKS\_ON1 //view name

**AS SELECT** Fname, Lname, Pname, Hours

**FROM** EMPLOYEE, PROJECT, WORKS\_ON //selection of rows from base table

**WHERE** Ssn=Essn **AND** Pno=Pnumber; //condition

**SQL PROGRAM**

use sys;

CREATE TABLE Student(

studentID INT NOT NULL AUTO\_INCREMENT,

FName VARCHAR(20),

LName VARCHAR(20),

Address VARCHAR(30),

City VARCHAR(15),

Marks INT,

PRIMARY KEY(studentID)

);

create TABLE teacher(

teacherID INT NOT NULL AUTO\_INCREMENT,

FName VARCHAR(20),

LName VARCHAR(20),

Address VARCHAR(30),

PRIMARY KEY(teacherID)

);

CREATE TRIGGER calculate

before INSERT

ON student

FOR EACH ROW

SET new.marks = new.marks+100;

insert into Student (studentID, FName, LName, Address, City, Marks) values(4, "vimal" , "jose" , "kottayam", "kottayam", 5);

insert into teacher (teacherID, fName, lName, Address) values(4, "james" , "mathew" , "kottayam");

select \* from Student;

select \* from teacher;

CREATE VIEW newrecord AS

SELECT Student.Fname, teacher.LName

FROM Student, teacher;

select\* from newrecord;