Practical 8

Creation of Trigger

A trigger is a pl/sql block structure which is fired when a DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when associated DML statement is executed.

Syntax:

CREATE [OR REPLACE] TRIGGER trigger\_name

BEFORE|AFTER|INSTEAD OF

INSERT [OR] | UPDATE [OR] | DELETE

[OF col\_name]

ON table\_name

[Referenceing OLD AS o NEW AS n]

[FOR EACH ROW]

WHEN (condition)

BEGIN

--- sql statements

END;

Create the following schemas:

CREATE TABLE PROJECT(PNO INT PRIMARY KEY, PNAME VARCHAR(30), THRS INT);

CREATE TABLE EMPS(ENO INT PRIMARY KEY, NAME VARCHAR(30), HRS INT, SALARY INT, PNO REFERENCES PROJECT(PNO))

insert into project values(1,'AA',0);

insert into project values(2,'BB',0);

insert into project values(3,'CC',0);

insert into project values(4,'DD',0);

insert into project values(5,'EE',0);

INSERT INTO EMPS VALUES(1111,’ABC’,4,2000,1);

INSERT INTO EMPS VALUES(2222,’DEF’,6,5000,3);

1. FOR ROW LEVEL TRIGGER (Create a trigger to delete the employees wherever corresponding project is deleted from project table)

CREATE OR REPLACE TRIGGER TRG\_DEL

AFTER DELETE

ON PROJECT

FOR EACH ROW

BEGIN

DELETE FROM EMPS

WHERE PNO = :OLD.PNO;

END;

DELETE FROM PROJECT

WHERE PNO=1;

2. CREATE STATEMENT LEVEL TRIGGER

#By default, the statement CREATE TRIGGER creates a statement-level trigger when you omit the FOR EACH ROW clause.

CREATE OR REPLACE TRIGGER customers\_credit\_trg

BEFORE UPDATE OF HRS

ON EMPS

DECLARE

l\_day\_of\_month NUMBER;

BEGIN

-- determine the transaction type

l\_day\_of\_month := EXTRACT(DAY FROM sysdate);

IF l\_day\_of\_month BETWEEN 28 AND 31 THEN

raise\_application\_error(-20100,'Cannot update customer credit from 28th to 31st');

END IF;

END;

UPDATE EMPS SET HRS=HRS\*2;