

## Experiment No. 10

**Environment:** Microsoft Windows

**Tools/ Language:** Oracle

**Objective:** To implement the concept of Triggers.

### **THEORY:**

#### **Database Triggers:-**

Database triggers are procedures that are stored in the database and are implicitly executed (fired) when the contents of a table are changed.

#### **Use of Database Triggers:-**

Database triggers support Oracle to provide a highly customized database management system. Some of the uses to which the database triggers can be put to customize management information in Oracle are as follows:-

- A Trigger can permit DML statements against a table only if they are issued, during regular business hours or on predetermined weekdays.
- A trigger can also be used to keep an audit trail of a table along with the operation performed and the time on which the operation was performed.
- It can be used to prevent invalid transactions.
- Enforce complex security authorizations.

#### **How to apply DataBase Triggers:-**

A trigger has three basic parts:-

1. A triggering event or statement.
2. A trigger restriction
3. A trigger action.

#### **Types of Triggers:-**

Using the various options , four types of triggers can be created:-

1. **Before Statement Trigger** :- Before executing the triggering statement, the trigger action is executed.
2. **Before Row Trigger** :- Before modifying the each row affected by the triggering statement and before appropriate integrity constraints, the trigger is executed if the trigger restriction either evaluated to TRUE or was not included.'
3. **After Statement Trigger** :- After executing the triggering statement and applying any deferred integrity constraints, the trigger action is executed.
4. **After row Trigger** :- After modifying each row affected by the triggering statement and possibly applying appropriate integrity constraints, the trigger action is executed for the current row if the trigger restriction either evaluates to TRUE or was not included.

### **Syntax For Creating Trigger :-**

The syntax for Creating the Trigger is as follows:-

```
Create or replace Trigger<Trigger name>
{Before,After} {Delete, Insert, Update}
On <Tablename>
For each row
when Condition
Declare
<Variable declarations>;
<Constant Declarations>;
Begin
<PL/SQL> Subprogram Body;
Exception
Exception Pl/SQL block;
End;
```

### **How to Delete a Trigger:-**

The syntax for Deleting the Trigger is as follows:-

Drop Trigger <Trigger name>;

### **Examples:**

1. Create trigger cascade  
After Insert on Student  
Insert into Msg values('Student inserted')  
  
insert into student values(128,'Raj',3.8,15234);  
  
select \* from msg;  
  
MSSG  
-----  
Student inserted
2. Create or Replace trigger cascade  
After Insert on Student  
Referencing new as nr  
For Each Row  
Insert into Msg values('Student inserted with sid = '||:nr.sid)  
  
insert into student values (238,'Pranky',3.4,12098);  
  
select \* from msg  
MSSG  
-----  
Student inserted  
Student inserted with sid = 238

## Practical Assignment - 10

**Department:** Computer Engineering & Applications

**Course:** B.Tech. (CSE)

**Subject:** Database Management System Lab (CSE3083)

**Year:** 2<sup>nd</sup>

**Semester: 3<sup>rd</sup>**

## SQL Script for this Experiment

[illegible]

```

drop table Apply;
drop table Student;
drop table College;
create table College(cName varchar2(10) primary key, state
varchar2(10), enrollment int);
create table Student(sID int primary key, sName varchar2(10), GPA
real, sizeHS int, DoB date);
create table Apply(sID int, cName varchar2(10), major
varchar2(20), decision char(1), primary key(sID, major, cName),
constraint sID_fk Foreign key(sID) references Student, constraint
cName_fk Foreign key(cName) references College);

delete from Student;
delete from College;
delete from Apply;

insert into Student values (123, 'Amy', 3.9, 1000, '26-JUN-96');
insert into Student values (234, 'Bob', 3.6, 1500, '7-Apr-95');
insert into Student values (345, 'Craig', 3.5, 500, '4-Feb-95');
insert into Student values (456, 'Doris', 3.9, 1000, '24-Jul-97');
insert into Student values (567, 'Edward', 2.9, 2000, '21-Dec-96');
insert into Student values (678, 'Fay', 3.8, 200, '27-Aug-96');
insert into Student values (789, 'Gary', 3.4, 800, '8-Oct-96');
insert into Student values (987, 'Helen', 3.7, 800, '27-Mar-97');
insert into Student values (876, 'Irene', 3.9, 400, '7-Mar-96');
insert into Student values (765, 'Jay', 2.9, 1500, '8-Aug-98');
insert into Student values (654, 'Amy', 3.9, 1000, '26-May-96');
insert into Student values (543, 'Craig', 3.4, 2000, '27-Aug-05');
insert into College values ('Stanford', 'CA', 15000);
insert into College values ('Berkeley', 'CA', 36000);
insert into College values ('MIT', 'MA', 10000);
insert into College values ('Cornell', 'NY', 21000);
insert into College values ('Harvard', 'MA', 50040);
insert into Apply values (123, 'Stanford', 'CS', 'Y');
insert into Apply values (123, 'Stanford', 'EE', 'N');
insert into Apply values (123, 'Berkeley', 'CS', 'Y');
insert into Apply values (123, 'Cornell', 'EE', 'Y');
insert into Apply values (234, 'Berkeley', 'biology', 'N');
insert into Apply values (345, 'MIT', 'bioengineering', 'Y');
insert into Apply values (345, 'Cornell', 'bioengineering', 'N');
insert into Apply values (345, 'Cornell', 'CS', 'Y');
insert into Apply values (345, 'Cornell', 'EE', 'N');
insert into Apply values (678, 'Stanford', 'history', 'Y');
insert into Apply values (987, 'Stanford', 'CS', 'Y');
insert into Apply values (987, 'Berkeley', 'CS', 'Y');
insert into Apply values (876, 'Stanford', 'CS', 'N');
insert into Apply values (876, 'MIT', 'biology', 'Y');
insert into Apply values (876, 'MIT', 'marine biology', 'N');
insert into Apply values (765, 'Stanford', 'history', 'Y');
insert into Apply values (765, 'Cornell', 'history', 'N');
insert into Apply values (765, 'Cornell', 'psychology', 'Y');
insert into Apply values (543, 'MIT', 'CS', 'N');
commit;

```

Student					Apply			
sID	sName	GPA	sizeHS	DoB	sID	cName	major	decision
123	Amy	3.9	1000	26-JUN-96	123	Stanford	CS	Y
234	Bob	3.6	1500	7-Apr-95	123	Stanford	EE	N
345	Craig	3.5	500	4-Feb-95	123	Berkeley	CS	Y
456	Doris	3.9	1000	24-Jul-97	123	Cornell	EE	Y
567	Edward	2.9	2000	21-Dec-96	234	Berkeley	biology	N
678	Fay	3.8	200	27-Aug-96	345	MIT	bioengineering	Y
789	Gary	3.4	800	8-Oct-96	345	Cornell	bioengineering	N
987	Helen	3.7	800	27-Mar-97	345	Cornell	CS	Y
876	Irene	3.9	400	7-Mar-96	345	Cornell	EE	N
765	Jay	2.9	1500	8-Aug-98	678	Stanford	history	Y
654	Amy	3.9	1000	26-May-96	987	Stanford	CS	Y
543	Craig	3.4	2000	27-Aug-98	987	Berkeley	CS	Y
					876	Stanford	CS	N
					876	MIT	biology	Y
					876	MIT	marine biology	N
					765	Stanford	history	Y
					765	Cornell	history	N
					765	Cornell	psychology	Y
					543	MIT	CS	N

  

College		
cName	state	enrollment
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040

Write SQL queries for the following:

1. Create trigger TASK that will display which DML operation we are performing on Apply Table i.e. if we are running Update on Apply then display ‘You are Updating APPLY Table’, on deletion display 'You are Deleting APPLY Table', on inserting 'You are Inserting a new row in APPLY Table'. **Hint:** create trigger TASK on all three DML statements use IF inserting
2. Create a trigger that will delete the student data from apply table upon deletion from student table.  
*Hint:* We want if you delete any student from STUDENT Table then applications with same sID in APPLY Table will be deleted. i.e. if we run query *delete from student where sID=123* then all application of sID 123 should be deleted from APPLY Table.
3. Create a trigger that will insert all deleted rows from table student to table backup student.  
*Hint use BEFORE DELETE*
4. Create a table STU\_MARKS as given below:

COLUMN NAMES	DESCRIPTION
STUD_NAME	Name of Student
PERCENTAGE	Percentage Marks

Write a trigger for a table student which will perform the following:

- a. Create table FAIL having two columns STUD\_NAME and cent for percentage.
- b. Create Trigger TrigFail such that If percentage is less than 40 it will copy name and percentage to FAIL table.

- c. Create Trigger TrigDEL When deletion is performed in Stu\_marks table then check whether the students name exists in FAIL table or not, if exists it will delete corresponding record from the table FAIL as well.

**Pre-Experiment Question:**

1. What are advantages of using Triggers?
2. What are types of Triggers?

**Post-Experiment Question:**

1. What are triggers type?
2. On what event get triggered?
3. Can we use trigger to implement concepts of ON DELETE CASCADE?
4. Can we made use of triggers for notification purposes in applications like EMAIL or Facebook?