Assignment No :- 8 Roll No :- 31384

Assignment No. 8

Title :- Database Triggee

problem Statement :- Write database triggee to keep track records on library table.

objective :-

- Understand the concept of database triggee
- Understand MySQL commands

Sw package : My-SQL

Aim : Study of database triggee.

pre-requisite : Basic knowledge of MySQL commands & database

Learning objectives :- To understand database triggee & ity types

Learning objective :- Implement & apply types of database triggees

Theory :

- What are triggers :-

A trigger defines an action that data base should take when some database - related event occurs.

Triggers are similar to procedure in that they are named PL-SQL blocks.

- Difference both procedure & trigger.

A procedure is executed explicitely from another block via procedure call with passing arguments while a trigger is executed implicitely whenever the triggering event happens & a trigger doesn't accept arguments.

When triggers are used ?

- Maintaining complex integrity constraints or buiness rules
- Auditing information in a table by recording the changes.
- Collecting / maintaining statistical datas.

```
CREATE [OR REPLACE] TRIGGER trigger-name
  { BEFORE |AFTER } {INSERT |UPDATE | DELETE }
    [of column] on table.reference
      [FOR EACH ROW [when trigger.condition ] ]

      [DECLARE]
```

Types of triggers :-

Row-level-trigger

- Row-level trigger execute once for each row in a transaction.
- It identified by for each Row clause in a create trigger command.

- Statement -level- trigger

Statement-level -trigger execute once for each transaction
for e.g. if a single transaction inserted 500 Rows into customer table then Statement level trigger on that table would only be executed once.

- Before- After triggers:

Since trigger occur because of events, they may be set to occur immediately before or after that events.

The event that execute trigger are database transaction.

Valid trigger types

Statement (insert, update, delete), Timing (Before, after), Level (Row-level, statement level)

Create or replace trigger updateMajorSels
After insert or Delete or update on Students

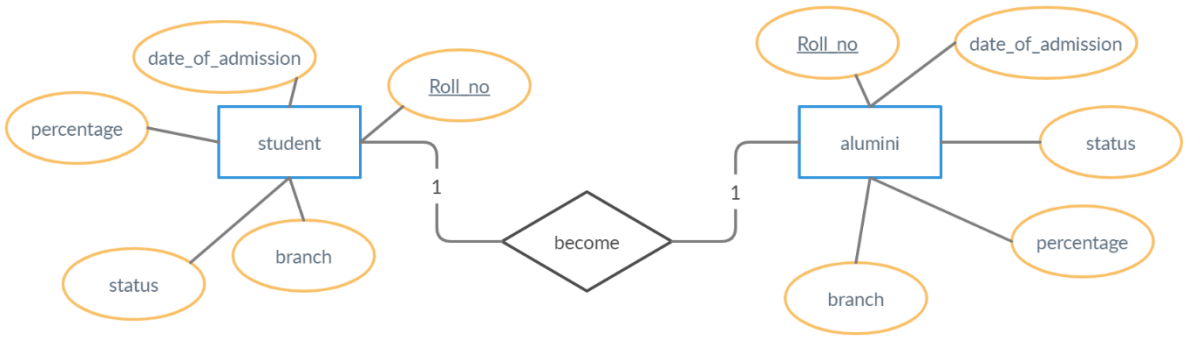
e.g.

```
Begin
    for v_stats in c_stats loop
      UPDATE major_stats
       SET total_credits = v_statsRecord.total_credits
       total_students = v_statsRecord.total_students
       where major = v_statsRecord.major;

    If SQL%NOTFOUND then
       INSERT into major_stats (major, total_credits,
          total_students) values (v_statsRecord.total_credi
           v_statsRecord.total_students);
    END IF;

    END loop;

END updatemajor_stats;
```

Conclusion :- Thus we implement the PL-SQL trigger in Oracle.

Assignment No :- 8 Roll No :- 31384

ER Diagram :-



Code :-

```
create database assignment8;

use assignment8;

create table Student(Rollno int primary key,Name varchar(45),DateofAdmission date,branch varchar(45),percent float,Status varchar(45));

create table Alumni(Rollno int,Name varchar(45),DateofAdmission date,branch varchar(45),percent float,Status varchar(45));


insert into Student values(1,"Swapnil",STR_TO_DATE("August 29 2016", "%M %d %Y"),"comp",95.00,"pursuing");

insert into Student values(2,"Ganesh",STR_TO_DATE("August 29 2013", "%M %d %Y"),"comp",70.00,"pursuing");


insert into Student values(3,"Rahul",STR_TO_DATE("August 29 2017", "%M %d %Y"),"comp",86.00,"pursuing");

insert into Student values(4,"Abhudyan",STR_TO_DATE("August 29 2016", "%M %d %Y"),"comp",60.00,"pursuing");

insert into Student values(5,"Ranjan",STR_TO_DATE("August 29 2019", "%M %d %Y"),"comp",78.00,"pursuing");


 select * from Student;

 select * from alumni;

  update student set percent=80 where Rollno=1;

  select * from alumni;

   delete from student where rollno=3;

select * from alumni;


drop table student;
```

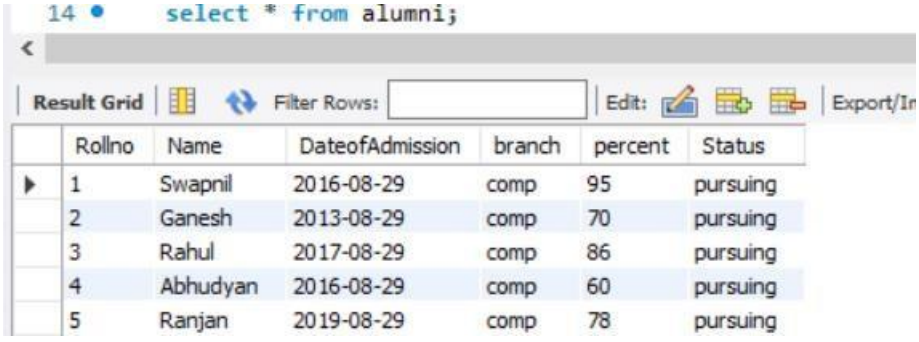
Assignment No :- 8 Roll No :- 31384

```
drop table alumni;



delimiter //
create trigger track
 after update
```

```
 on Student
FOR each row
begin
 insert into Alumni
values(OLD.Rollno,OLD.Name,OLD.DateofAdmission,OLD.branch,OLD.percent,"passout");
 end;//


create trigger track2
 after delete
 on Student FOR each row
 begin
 insert into Alumni
values(OLD.Rollno,OLD.Name,OLD.DateofAdmission,OLD.branch,OLD.percent,"passout") ;
 end;//
```

Output :-

Student table after inserting values manually and before Update or delete operation.:-

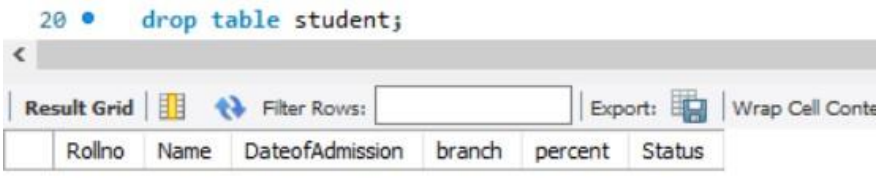


```
14 •  select * from alumni;
```

Result Grid | Filter Rows: | Edit: | Export/Ir

Rollno	Name	DateofAdmission	branch	percent	Status
1	Swapnil	2016-08-29	comp	95	pursuing
2	Ganesh	2013-08-29	comp	70	pursuing
3	Rahul	2017-08-29	comp	86	pursuing
4	Abhudyan	2016-08-29	comp	60	pursuing
5	Ranjan	2019-08-29	comp	78	pursuing

Alumni table before insert and update (empty)



```
20 •  drop table student;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

Rollno	Name	DateofAdmission	branch	percent	Status

update student set percent=80 where Rollno=1;

after updating Student table , alumni table looks like :-

```
16 •   select * from alumni;
17 •    delete from student where rollno=3;
18 •  select * from alumni;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Rollno	Name	DateofAdmission	branch	percent	Status
1	Swapnil	2016-08-29	comp	80	passout

delete from student where rollno=3;

after deleting values from Student table , alumni table looks like :-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Rollno	Name	DateofAdmission	branch	percent	Status
1	Swapnil	2016-08-29	comp	80	passout
3	Rahul	2017-08-29	comp	86	passout

\