Assignment - 8 :

- Date of Completion : 05/10/2020

- Date of Submission : 12/10/2020

- Title : Database trigger (All types: Row level and statement level triggers, before and after triggers)

- Problem Statement :

  Write a database trigger on student_table. The system should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in 'Alumni' table.
  Student (RollNo, Name, DateofAdmission, branch, percent, status)

- Learning Objective :
  - To understood row level and statement level triggers
  - To understand before and after triggers.

- Learning Outcome :

  Students will be able to :
  - Write triggers for before and after update or delete statements on a table.
  - Write row level and statement level triggers

- S/W And H/W Requirements:
  MySQL, windows 10 (64bit), i5 processer, etc.

- Theory:

- MySQL triggers:

A trigger is a set of actions that are run automatically when a specified change operation (INSERT, UPDATE or DELETE) is performed on specified table.

Syntax:

```
create trigger t_name
trigger_time    trigger_event
  on table_name  for each row
begin
   ---
end;
```

trigger_time: before / after
trigger_event: insert / ~~befo~~ up date / delete

- PL/SQL triggers:

Triggers are stored programs, which are automatically executed on or fixed when certain events occur.

Triggers are infact written to be executed in response to following events:

- DML statements
- DDL statements
- a database operation (LOGON, LOGOFF)

Triggers can be defined on table, view, schema or database with which the event is associated

- creating triggers:
  - Syntax:

```
        create [or replace] trigger t_name
   { before / after / instead of }
   { insert [OR] / update [OR] / delete }
   [ OF col_name ]
     ON table_name
   [ referencing old as o new as n ]
   [ for each row ]
   when (condition)
   declare
         declaration statements
    begin
         execution statements
    exception
         exception statements
    end;
```

Page No.
Date

- BEFORE AND AFTER TRIGGER
  - before triggers run the trigger action before the triggering statement is run
  - after triggering run the triggers action after the triggering statement is run.

- ROW LEVEL triggers:
  A row trigger is fired each time the table is affected by the triggering statement. If a statement affects multiple rows, for each row, the trigger will be fired.

- Statement level triggers:
  A statement trigger is fired once on behalf of a triggering statement, regardless of the number of rows in the table that the triggering statement affects.

- various combinations of trigger timing and event can be used

ex:

i) before insert for each row
ii) after delete for each statement
iii) before update for each row
iv) after insert, etc.

Teacher's Sign.: ____________

* CONCLUSION:

Thus, we implemented

i) All types of triggers in database

ii) row level and statement level triggers.

iii) before and after triggers.