

## EXPERIMENT-15

**AIM :** TO Write PL/SQL program to implement Trigger on table.

Trigger is invoked by Oracle engine automatically whenever a specified event occurs. Trigger is

stored into database and invoked repeatedly, when specific condition match.

Triggers are

stored programs, which are automatically executed or fired when some event occurs. Triggers

are written to be executed in response to any of the following events.

A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).

A database definition (DDL) statement (CREATE, ALTER, or DROP).

A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

```
SQL-CSE530>CREATE TABLE INSTRUCTOR
 2  (ID VARCHAR2(5),
 3  NAME VARCHAR2(20) NOT NULL,
 4  DEPT_NAME VARCHAR2(20),
 5  SALARY NUMERIC(8,2) CHECK (SALARY > 29000),
 6  PRIMARY KEY (ID),
 7  FOREIGN KEY (DEPT_NAME) REFERENCES DEPARTMENT(DEPT_NAME)
 8  ON DELETE SET NULL
 9  );
```

Table created.

```
SQL-CSE530>insert into department values ('Biology', 'Watson', '90000');
1 row created.

SQL-CSE530>insert into department values ('Comp. Sci.', 'Taylor', '100000');
1 row created.

SQL-CSE530>insert into department values ('Elec. Eng.', 'Taylor', '85000');
1 row created.

SQL-CSE530>insert into department values ('Finance', 'Painter', '120000');
1 row created.

SQL-CSE530>insert into department values ('History', 'Painter', '50000');
1 row created.

SQL-CSE530>insert into department values ('Music', 'Packard', '80000');
1 row created.
```

CREATING DEPARTMENT TABLE :

```
SQL-CSE530>CREATE TABLE DEPARTMENT
 2  (DEPT_NAME VARCHAR2(20),
 3  BUILDING VARCHAR2(15),
 4  BUDGET NUMERIC(12,2) CHECK (BUDGET > 0),
 5  PRIMARY KEY (DEPT_NAME)
 6  );
```

Table created.

### An example to create Trigger :

```
SQL-CSE530>CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE UPDATE ON instructor
  3  FOR EACH ROW
  4  WHEN (NEW.ID = OLD.ID)
  5  DECLARE
  6  sal_diff number;
  7  BEGIN
  8  sal_diff := :NEW.salary - :OLD.salary;
  9  dbms_output.put_line('Old salary: ' || :OLD.salary);
 10  dbms_output.put_line('New salary: ' || :NEW.salary);
 11  dbms_output.put_line('Salary difference: ' || sal_diff);
 12  END;
 13  /
```

Trigger created.

### A PL/SQL Procedure to execute a trigger:

```
SQL-CSE530>DECLARE
  2  total_rows number(2);
  3  BEGIN
  4  UPDATE instructor
  5  SET salary = salary + 5000;
  6  IF sql%notfound THEN
  7  dbms_output.put_line('no instructors updated');
  8  ELSIF sql%found THEN
  9  total_rows := sql%rowcount;
 10  dbms_output.put_line( total_rows || ' instructors updated ');
 11  END IF;
 12  END;
 13  /
no instructors updated

PL/SQL procedure successfully completed.
```