Ex. No: 9 **Date: 10 - 11 - 2022**

Experiment 9
Triggers

Consider the following schema of a database:

emp (EID, NAME, SALARY)

Table:
emp

EID	NAME	SALARY
100	Steven	24000
101	Neena	17000
102	Lex	17000
103	Alexander	9000
104	Bruce	6000
105	David	4800
106	Valli	4800
107	Diana	4200

Write a SQL query to

1. Create a statement-type DML trigger for UPDATE command.
2. Create a row-type DML trigger for UPDATE command.
3. Create a statement-type DDL trigger for DROP command.

Query

1. create or replace trigger emptrg1 before update on emp

   begin

   dbms_output.put_line('Updated records!');

   end;

   /

2. create or replace trigger emptrg2 before update on emp for each row

   begin

   dbms_output.put_line('Updated records!');

   end;

   /

3. create or replace trigger systrg1 before drop on system.schema

   begin

```
dbms_output.put_line('Deleted table!');

end;

/
```

Code:

```
SQL> create table emp (EID number(3), NAME varchar2(20), SALARY number,
primary key(EID));

Table created.

SQL> desc emp
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
------
 EID                                       NOT NULL NUMBER(3)
 NAME                                               VARCHAR2(20)
 SALARY                                             NUMBER

SQL> insert into emp values(&EID, '&NAME', &SALARY);
Enter value for eid: 100
Enter value for name: Steven
Enter value for salary: 24000
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(100, 'Steven', 24000)

1 row created.

SQL> /
Enter value for eid: 101
Enter value for name: Neena
Enter value for salary: 17000
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(101, 'Neena', 17000)

1 row created.

SQL> /
Enter value for eid: 102
Enter value for name: Lex
Enter value for salary: 17000
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(102, 'Lex', 17000)

1 row created.

SQL> /
Enter value for eid: 103
Enter value for name: Alexander
Enter value for salary: 9000
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(103, 'Alexander', 9000)
```

```
1 row created.

SQL> /
Enter value for eid: 104
Enter value for name: Bruce
Enter value for salary: 6000
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(104, 'Bruce', 6000)

1 row created.

SQL> /
Enter value for eid: 105
Enter value for name: David
Enter value for salary: 4800
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(105, 'David', 4800)

1 row created.

SQL> /
Enter value for eid: 106
Enter value for name: Valli
Enter value for salary: 4800
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(106, 'Valli', 4800)

1 row created.

SQL> /
Enter value for eid: 107
Enter value for name: Diana
Enter value for salary: 4200
old   1: insert into emp values(&EID, '&NAME', &SALARY)
new   1: insert into emp values(107, 'Diana', 4200)

1 row created.

SQL> select * from emp;

       EID NAME                      SALARY
---------- -------------------- ----------
       100 Steven                    24000
       101 Neena                     17000
       102 Lex                       17000
       103 Alexander                  9000
       104 Bruce                      6000
       105 David                      4800
       106 Valli                      4800
       107 Diana                      4200

8 rows selected.
```

```
SQL> set serveroutput on;
SQL> create or replace trigger emptrg1 before update on emp
  2  begin
  3  dbms_output.put_line('Updated records!');
  4  end;
  5  /

Trigger created.

SQL> update emp set salary = salary * 1.2 where salary < 10000;
Updated records!

5 rows updated.

SQL> update emp set salary = salary * 1.2 where eid = 107;
Updated records!

1 row updated.

SQL> create or replace trigger emptrg2 before update on emp for each row
  2  begin
  3  dbms_output.put_line('Updated records!');
  4  end;
  5  /

Trigger created.

SQL> update emp set salary = salary * 1.2 where salary < 10000;
Updated records!
Updated records!
Updated records!
Updated records!
Updated records!

4 rows updated.

SQL> create or replace trigger systrg1 before drop on system.schema
  2  begin
  3  dbms_output.put_line('Deleted table!');
  4  end;
  5  /

Trigger created.

SQL> drop table emp;
Deleted table!

Table dropped.
```