EXPERIMENT -4

Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the Salary difference between the old & new Salary. CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)

1. Create the CUSTOMERS Table

```
SQL> connect system
Enter password:
Connected.
SQL> create user a1 identified by a1;

User created.

SQL> Grant resource,connect to a1;

Grant succeeded.

SQL> connect a1
Enter password:
Connected.
SQL> SET SERVEROUTPUT ON;
SQL> CREATE TABLE CUSTOMERS12 (
  2  ID INT PRIMARY KEY,
  3  NAME VARCHAR(100),
  4  AGE INT,
  5  ADDRESS VARCHAR(255),
  6  SALARY DECIMAL(10, 2)
  7  );

Table created.
```

2. Create Trigger for INSERT Operation
3. Create Trigger for UPDATE Operation
4. Create Trigger for DELETE Operation

```
SQL> CREATE OR REPLACE TRIGGER salary_difference_trigger12
  2  BEFORE INSERT OR UPDATE OR DELETE ON CUSTOMERS12
  3  FOR EACH ROW
  4  DECLARE
  5  old_salary NUMBER;
  6  new_salary NUMBER;
  7  BEGIN
  8  IF INSERTING OR UPDATING THEN
  9  old_salary := NVL(:OLD.SALARY, 0);
 10  new_salary := NVL(:NEW.SALARY, 0);
 11  DBMS_OUTPUT.PUT_LINE('Salary difference: ' || (new_salary - old_salary));
 12  ELSIF DELETING THEN
 13  old_salary := NVL(:OLD.SALARY, 0);
 14  DBMS_OUTPUT.PUT_LINE('Salary before deletion: ' || old_salary);
 15  END IF;
 16  END;
 17  /

Trigger created.
```

```
SQL> INSERT INTO CUSTOMERS12 (ID, NAME, AGE, ADDRESS, SALARY)
  2  VALUES (1, 'John Doe', 30, '123 Main St', 50000);
Salary difference: 50000

1 row created.
```

```
SQL> COMMIT;

Commit complete.

SQL> UPDATE CUSTOMERS12
  2  SET salary = 65000
  3  WHERE ID= 1;
Salary difference: 15000

1 row updated.
```

```
SQL> DELETE FROM CUSTOMERS12
  2  WHERE ID = 1;
Salary before deletion: 65000

1 row deleted.
```

```
SQL> CREATE OR REPLACE TRIGGER salary_difference_trigger12
  2  AFTER INSERT OR UPDATE OR DELETE ON CUSTOMERS12
  3  FOR EACH ROW
  4  DECLARE
  5   old_salary customers12.salary%TYPE;
  6  new_salary customers12.salary%TYPE;
  7      difference NUMBER;
  8  BEGIN
  9      IF INSERTING THEN
 10          DBMS_OUTPUT.PUT_LINE('New record inserted.');
 11          DBMS_OUTPUT.PUT_LINE('ID: ' || :NEW.ID || ', Name: ' || :NEW.NAME || ', Age: ' || :NEW.AGE || ', Address: ' || :NEW.ADDRESS || ', Salary: ' ||
:NEW.SALARY);
 12      ELSIF UPDATING THEN
 13          old_salary := :OLD.SALARY;
 14          new_salary := :NEW.SALARY;
 15          difference := new_salary - old_salary;
 16          DBMS_OUTPUT.PUT_LINE('Salary updated for ID: ' || :NEW.ID || '. Old Salary: ' || old_salary || ', New Salary: ' || new_salary || ', Salary Diff
erence: ' || difference);
 17      ELSIF DELETING THEN
 18          DBMS_OUTPUT.PUT_LINE('Record deleted for ID: ' || :OLD.ID || ', Name: ' || :OLD.NAME || ', Age: ' || :OLD.AGE || ', Address: ' || :OLD.ADDRESS
|| ', Salary: ' || :OLD.SALARY);
 19      END IF;
 20  END;
 21  /

Trigger created.
```

5. Testing the Trigger

```
SQL> INSERT INTO CUSTOMERS12 (ID, NAME, AGE, ADDRESS, SALARY)
  2  VALUES (1, 'Alice Smith', 30, '123 Maple St', 60000);
New record inserted.
ID: 1, Name: Alice Smith, Age: 30, Address: 123 Maple St, Salary: 60000

1 row created.

SQL> UPDATE CUSTOMERS12
  2  SET SALARY = 65000
  3  WHERE ID = 1;
Salary updated for ID: 1. Old Salary: 60000, New Salary: 65000, Salary
Difference: 5000

1 row updated.

SQL> DELETE FROM CUSTOMERs12
  2  WHERE ID = 1;
Record deleted for ID: 1, Name: Alice Smith, Age: 30, Address: 123 Maple St,
Salary: 65000

1 row deleted.
```