

```
-- Trigger: Create a row level trigger for the CUSTOMERS table that would fire
INSERT or
--UPDATE or DELETE operations performed on the
-- CUSTOMERS table. This trigger will display the salary difference between the
old values and new values.
```

```
-- CREATE TABLE CUSTOMERS (
--     ID NUMBER,
--     NAME VARCHAR2(100),
--     SALARY NUMBER
-- );
-- INSERT INTO CUSTOMERS (ID, NAME, SALARY) VALUES (1, 'TEST', 50000);
-- INSERT INTO CUSTOMERS (ID, NAME, SALARY) VALUES (2, 'TEST1', 20000);
```

```
CREATE OR REPLACE TRIGGER salary_difference_trigger
BEFORE INSERT OR UPDATE OR DELETE
ON CUSTOMERS
FOR EACH ROW
```

```
DECLARE
```

```
    v_old_salary NUMBER;
```

```
    v_new_salary NUMBER;
```

```
BEGIN
```

```
    -- For UPDATE and DELETE operations, capture the old salary value
```

```
    IF UPDATING OR DELETING THEN
```

```
        v_old_salary := :OLD.SALARY;
```

```
    END IF;
```

```
    -- For INSERT and UPDATE operations, capture the new salary value
```

```
    IF INSERTING OR UPDATING THEN
```

```
        v_new_salary := :NEW.SALARY;
```

```
    END IF;
```

```
    -- Display the salary difference
```

```
    IF v_old_salary IS NOT NULL AND v_new_salary IS NOT NULL THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Salary Difference: ' || (v_new_salary -
v_old_salary));
```

```
    ELSIF v_new_salary IS NOT NULL THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Salary Set: ' || v_new_salary);
```

```
    ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('Salary Deleted');
```

```
    END IF;
```

```
END;
```

```
/
```

```
-- Test
```

```
-- UPDATE CUSTOMERS SET SALARY = 60000 WHERE ID = 1;
```

```
-- DELETE FROM CUSTOMERS WHERE ID = 1;
```