**EX.NO:4**

**Question 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)**

Solution

1. Create the CUSTOMERS Table

First, create the CUSTOMERS table with the specified columns:

mysql> CREATE DATABASE COMPANY04;

Query OK, 1 row affected (0.14 sec)

mysql> USE COMPANY04;

Database changed

mysql> CREATE TABLE CUSTOMERS ( ID INT PRIMARY KEY AUTO\_INCREMENT, NAME VARCHAR(255),AGE INT, ADDRESS VARCHAR(255),SALARY DECIMAL(10, 2) );

Query OK, 0 rows affected (0.49 sec)

To achieve the desired functionality of capturing changes on INSERT, UPDATE, or DELETE operations and displaying the salary difference in MySQL, you’ll need to create separate row-level triggers for each operation (INSERT, UPDATE, DELETE). These triggers will capture the OLD and NEW values of the SALARY column and display the salary difference when an INSERT, UPDATE, or DELETE operation occurs.Here’s how you can do it:

2. Create Trigger for INSERT Operation

-- INSERT TRIGGER

DELIMITER //

CREATE TRIGGER after\_insert\_salary\_difference

AFTER INSERT ON CUSTOMERS

FOR EACH ROW

BEGIN

SET @my\_sal\_diff = CONCAT('salary inserted is ', NEW.SALARY);

END;//

DELIMITER ;

3. Create Trigger for UPDATE Operation

-- UPDATE TRIGGER

DELIMITER //

CREATE TRIGGER after\_update\_salary\_difference

AFTER UPDATE ON CUSTOMERS

FOR EACH ROW

BEGIN

DECLARE old\_salary DECIMAL(10, 2);

DECLARE new\_salary DECIMAL(10, 2);

SET old\_salary = OLD.SALARY;

SET new\_salary = NEW.SALARY;

SET @my\_sal\_diff = CONCAT('salary difference after update is ', NEW.SALARY - OLD.SALARY);

END;//

DELIMITER ;

4. Create Trigger for DELETE Operation

-- DELETE TRIGGER

DELIMITER //

CREATE TRIGGER after\_delete\_salary\_difference

AFTER DELETE ON CUSTOMERS

FOR EACH ROW

BEGIN

SET @my\_sal\_diff = CONCAT('salary deleted is ', OLD.SALARY);

END;//

DELIMITER ;

5. Testing the Trigger:

Once the triggers are created, you can perform INSERT, UPDATE, or DELETE operations on the CUSTOMERS table to observe the salary difference messages generated by the triggers.

For example:

mysql> -- test INSERT TRIGGER

mysql> INSERT INTO CUSTOMERS (NAME, AGE, ADDRESS, SALARY)VALUES ('Shankara', 35, '123 Main St', 50000.00);

Query OK, 1 row affected (0.14 sec)

mysql>

mysql> SELECT @my\_sal\_diff AS SAL\_DIFF;

+-----------------------------+

| SAL\_DIFF |

+-----------------------------+

| salary inserted is 50000.00 |

+-----------------------------+

1 row in set (0.00 sec)

mysql> -- test UPDATE TRIGGER

mysql> UPDATE CUSTOMERS SET SALARY = 55000.00 WHERE ID = 1;

Query OK, 1 row affected (0.13 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT @my\_sal\_diff AS SAL\_DIFF;

+-------------------------------------------+

| SAL\_DIFF |

+-------------------------------------------+

| salary difference after update is 5000.00 |

+-------------------------------------------+

1 row in set (0.00 sec)

mysql> -- test DELETE TRIGGER

mysql> DELETE FROM CUSTOMERS WHERE ID = 1;

Query OK, 1 row affected (0.13 sec)

mysql>

mysql> SELECT @my\_sal\_diff AS SAL\_DIFF;

+----------------------------+

| SAL\_DIFF |

+----------------------------+

| salary deleted is 55000.00 |

+----------------------------+

1 row in set (0.00 sec)

Each operation (INSERT, UPDATE, DELETE) will trigger the respective trigger (after\_insert\_salary\_difference, after\_update\_salary\_difference, after\_delete\_salary\_difference), which will display the salary change or difference associated with that operation.

By using separate triggers for each operation and utilizing the OLD and NEW keywords appropriately within the trigger bodies, you can effectively capture and handle changes to the SALARY column in the CUSTOMERS table in MySQL. You can adjust the trigger logic and message formatting as needed based on your specific requirements.