OPEN ENDED EXPERIMENT 1

Program :

Implementation of triggers in SQL

Input:

CREATE TABLE Employee

(

)

Id INT PRIMARY KEY,

Name VARCHAR(45),

Salary INT,

Gender VARCHAR(12),

DepartmentId INT

Inserting some record in the table

INSERT INTO Employee VALUES (1,'Steffan', 82000, 'Male', 3),

(2,'Amelie', 52000, 'Female', 2),

(3,'Antonio', 25000, 'male', 1),

(4,'Marco', 47000, 'Male', 2),

(5,'Eliana', 46000, 'Female', 3)

SELECT * FROM Employee;

Id	Name	Salary	Gender	DepartmentId
1	Steffan	82000	Male	3
2	Amelie	52000	Female	2
3	Antonio	25000	male	1
4	Marco	47000	Male	2
5	Eliana	46000	Female	3
6	Peter	62000	Male	3

We will also create another table named 'Employee_Audit_Test' to automatically store transaction records of each operation, such as INSERT, UPDATE, or DELETE on the Employee table

```
CREATE TABLE Employee_Audit_Test
(
Id int IDENTITY,
Audit_Action text
)
```

Now, creating a trigger that stores transaction records of each insert operation on the Employee table into the Employee_Audit_Test table.

```
CREATE TRIGGER trInsertEmployee
ON Employee
FOR INSERT
AS
BEGIN
Declare @Id int
SELECT @Id = Id from inserted
INSERT INTO Employee_Audit_Test
VALUES ('New employee with Id = ' + CAST(@Id AS VARCHAR(10)) + ' is added at ' + C
 AST(Getdate() AS VARCHAR(22)))
END
INSERT INTO Employee VALUES (6,'Peter', 62000, 'Male', 3)
```

Id	Audit_Action
1	New employee with Id = 6 is added at Mar 24 2021 2:08PM

```
CREATE TRIGGER trDeleteEmployee
ON Employee
FOR DELETE
AS
BEGIN
Declare @Id int
SELECT @Id = Id from deleted
INSERT INTO Employee_Audit_Test
VALUES ('An existing employee with Id = ' + CAST(@Id AS VARCHAR(10)) + ' is deleted
```

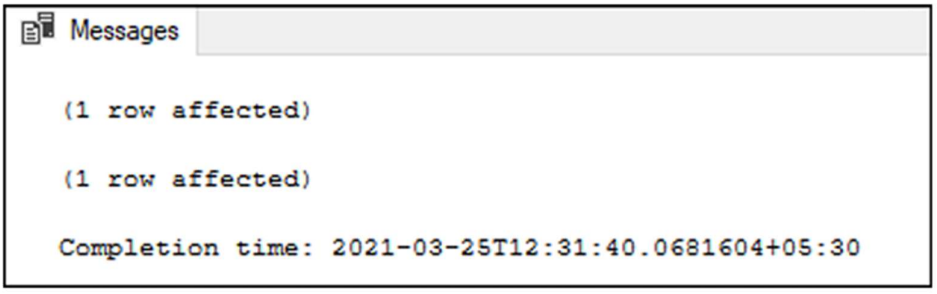
at ' + CAST(Getdate() AS VARCHAR(22)))

END

After creating a trigger, we will delete a record from the Employee table:

DELETE FROM Employee WHERE Id = 2;

If no error is found, it gives the message as below:



Finally, execute the SELECT statement to check the audit records:

Result:

Id	Audit_Action
1	New employee with Id = 6 is added at Mar 24 2021 2:08PM
2	An existing employee with Id = 2 is deleted at Mar 25 2021 12:26PM

OPEN ENDED EXPERIMENT 2

Program :

Implementation of cursor in SQL

Input:

```
CREATE TABLE customer (
id int PRIMARY KEY,
c_name nvarchar(45) NOT NULL,
email nvarchar(45) NOT NULL,
city nvarchar(25) NOT NULL
);
```

Next, we will insert values into the table

```
INSERT INTO customer (id, c_name, email, city)
VALUES (1,'Steffen', 'stephen@javatpoint.com', 'Texas'),
(2, 'Joseph', 'Joseph@javatpoint.com', 'Alaska'),
(3, 'Peter', 'Peter@javatpoint.com', 'California'),
(4,'Donald', 'donald@javatpoint.com', 'New York'),
(5, 'Kevin', 'kevin@javatpoint.com', 'Florida'),
(6, 'Marielia', 'Marielia@javatpoint.com', 'Arizona'),
(7,'Antonio', 'Antonio@javatpoint.com', 'New York'),
(8, 'Diego', 'Diego@javatpoint.com', 'California');
```

We can verify the data by executing the SELECT statement:

```
SELECT * FROM customer;
```

After executing the query,

id	c_name	email	city
1	Steffen	stephen@javatpoint.com	Texas
2	Joseph	Joseph@javatpoint.com	Alaska
3	Peter	Peter@javatpoint.com	California
4	Donald	donald@javatpoint.com	New York
5	Kevin	kevin@javatpoint.com	Florida
6	Marielia	Marielia@javatpoint.com	Arizona
7	Antonio	Antonio@javatpoint.com	New York
8	Diego	Diego@javatpoint.com	California

Now, we will create a cursor to display the customer records. --Declare the variables for holding data.

```
DECLARE @id INT, @c_name NVARCHAR(50), @city NVARCHAR(50)   --Declare and set counter.
DECLARE @Counter INT
SET @Counter = 1   --Declare a cursor
DECLARE PrintCustomers CURSOR
FOR
SELECT id, c_name, city FROM customer   --Open cursor
OPEN PrintCustomers   --Fetch the record into the variables.
FETCH NEXT FROM PrintCustomers INTO
@id, @c_name, @city   --LOOP UNTIL RECORDS ARE AVAILABLE.
WHILE @@FETCH_STATUS = 0
BEGIN
IF @Counter = 1
BEGIN
PRINT 'id' + CHAR(9) + 'c_name' + CHAR(9) + CHAR(9) + 'city'
PRINT '-------------------------'
END   --Print the current record
PRINT CAST(@id AS NVARCHAR(10)) + CHAR(9) + @c_name + CHAR(9) + CHAR(9
 ) + @city   --Increment the counter variable
SET @Counter = @Counter + 1   --Fetch the next record into the variables.
FETCH NEXT FROM PrintCustomers INTO
@id, @c_name, @city
END   --Close the cursor
CLOSE PrintCustomers   --Deallocate the cursor
DEALLOCATE PrintCustomers
```

After executing a cursor, we will get

Result: