***Stored Procedure with Triggers***

**Triggers :**

* A trigger is a special kind of a store procedure that executes in response to certain action on the table like insertion, deletion or updation of data. It is a database object which is bound to a table and is executed automatically.

**Common Use :**

* Creating audit trails: This means writing out records that keep track of not just the most current data, but also the actual change history for each record. This may eventually become less popular with the change-data tracking that SQL Server 2008 added, but triggers are still a pretty popular choice

**Trigger:**

CREATE TABLE Employee

1 - Create a new table or use any other table from database

(

Emp\_ID INT IDENTITY,

Emp\_name VARCHAR(30),

Emp\_sal DECIMAL(10, 2)

);

INSERT INTO Employee (Emp\_name, Emp\_sal)

VALUES

2- As we have created a new table so lets insert some records in it.

('Umair', 1000),

('Salman', 2000),

('Noman', 3000);

CREATE TABLE Employee\_Audit (

Emp\_ID INT,

3- Create a table that will be used to track changes made in Employee.

Emp\_name VARCHAR(30),

Emp\_sal DECIMAL(10, 2),

Audit\_Action VARCHAR(100),

Audit\_Timestamp DATETIME);

**Trigger For Insertion:**

CREATE TRIGGER audit\_insertion\_employees

ON Employee

FOR INSERT

AS

DECLARE @empid INT;

DECLARE @empname VARCHAR(30);

DECLARE @empsal DECIMAL(10, 2);

DECLARE @audit VARCHAR(100);

SELECT @empid = i.Emp\_ID FROM INSERTED i;

SELECT @empname = i.Emp\_name FROM INSERTED i;

SELECT @empsal = i.Emp\_sal FROM INSERTED i;

SELECT @audit = 'Insert Record = After Insert Trigger';

INSERT INTO Employee\_Audit

VALUES (

@empid,

@empname,

@empsal,

@audit,

GETDATE()

);

PRINT 'AFTER INSERT TRIGGER FIRED';

**Inserting Values After Creating a Tigger:**

INSERT INTO Employee (Emp\_name, Emp\_sal)

VALUES

('Arsalan', 4000);

**Output:**

Trigger Work

(1 row affected)

AFTER INSERT TRIGGER FIRED

(1 row affected)

Completion time: 2024-0902T09:43:40.1744809+05:00

**Trigger For Updation:**

CREATE TRIGGER trg\_after\_update

ON Employee

FOR UPDATE

AS

DECLARE @empid INT;

DECLARE @empname VARCHAR(30);

DECLARE @empsal DECIMAL(10, 2);

DECLARE @audit VARCHAR(100);

SELECT @empid = i.Emp\_ID FROM INSERTED i;

SELECT @empname = i.Emp\_name FROM INSERTED i;

SELECT @empsal = i.Emp\_sal FROM INSERTED i;

IF UPDATE (Emp\_Name)

SET @audit = 'Update Record, After Update Trigger, Name Updated';

IF UPDATE (Emp\_sal)

SET @audit = 'Update Record, After Update Trigger, Sal Updated';

INSERT INTO Employee\_Audit

VALUES (

@empid,

@empname,

@empsal,

@audit,

GETDATE()

);

PRINT 'AFTER UPDATE TRIGGER FIRED';

**Updating Values After Creating a Tigger:**

UPDATE Employee

SET Emp\_name = 'Luqman'

WHERE Emp\_ID = '2';

**Output:**

(1 row affected)

AFTER UPDATE TRIGGER FIRED

(0 rows affected)

Completion time: 2024-09-2T09:52:15.8954422+05:00

Trigger Work

**Trigger For Deletion:**

CREATE TRIGGER trg\_after\_delete

ON Employee

AFTER DELETE

AS

DECLARE @empid INT;

DECLARE @empname VARCHAR(30);

DECLARE @empsal DECIMAL(10, 2);

DECLARE @audit VARCHAR(100);

SELECT @empid = i.Emp\_ID FROM INSERTED i;

SELECT @empname = i.Emp\_name FROM INSERTED i;

SELECT @empsal = i.Emp\_sal FROM INSERTED i;

SELECT @audit = 'Deleted, After Delete Trigger';

INSERT INTO Employee\_Audit

VALUES (

@empid,

@empname,

@empsal,

@audit,

GETDATE()

);

PRINT 'AFTER DELETE TRIGGER FIRED';

**Deleting Values After Creating a Tigger:**

DELETE FROM Employee

WHERE Emp\_name = 'Luqman';

**Output:**

(1 row affected)

AFTER DELETE TRIGGER FIRED

(0 rows affected)

Completion time: 2024-0902T09:57:45.7419929+05:00

Trigger Work