**DML**

1. **Database level**

create table orders(

orderId int primary key identity,

customerId int,

orderdate date,

Amount int

)

-- Insert 10 records into the Orders table

INSERT INTO Orders (CustomerID, OrderDate, Amount)

VALUES

(1, '2022-01-01', 100),

(1, '2022-01-02', 200),

(2, '2022-01-03', 150),

(3, '2022-01-04', 75),

(4, '2022-01-05', 250),

(5, '2022-01-06', 300),

(5, '2022-01-07', 175),

(5, '2022-01-08', 225),

(6, '2022-01-09', 125),

(6, '2022-01-10', 200);

CREATE TABLE OrderAudit

(

AuditID int PRIMARY KEY IDENTITY,

OrderID int NOT NULL,

OldOrderDate date NULL,

NewOrderDate date NULL,

ActionType varchar(10) NOT NULL,

AuditDate datetime NOT NULL

);

-- \_\_\_\_\_\_\_\_\_\_\_\_ Create the DML trigger on the Orders table \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE TRIGGER OrderChange

ON Orders

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

-- Insert the changed order record and the current timestamp into the OrderAudit table

IF EXISTS (SELECT \* FROM inserted) -- INSERT

BEGIN

INSERT INTO OrderAudit (OrderID, OldOrderDate, NewOrderDate, ActionType, AuditDate)

SELECT i.OrderID, NULL, i.OrderDate, 'INSERT', GETDATE()

FROM inserted i;

END

IF EXISTS (SELECT \* FROM deleted) -- DELETE

BEGIN

INSERT INTO OrderAudit (OrderID, OldOrderDate, NewOrderDate, ActionType, AuditDate)

SELECT d.OrderID, d.OrderDate, NULL, 'DELETE', GETDATE()

FROM deleted d;

END

IF EXISTS (SELECT \* FROM inserted i INNER JOIN deleted d ON i.OrderID = d.OrderID WHERE i.OrderDate <> d.OrderDate) -- UPDATE

BEGIN

INSERT INTO OrderAudit (OrderID, OldOrderDate, NewOrderDate, ActionType, AuditDate)

SELECT i.OrderID, d.OrderDate, i.OrderDate, 'UPDATE', GETDATE()

FROM inserted i

INNER JOIN deleted d ON i.OrderID = d.OrderID

WHERE i.OrderDate <> d.OrderDate;

END

END;

--\_\_\_\_\_\_\_\_ Test Trigger \_\_\_\_\_\_\_\_

select \* from orders

insert into Orders values (1,'2021-2-24',4000)

--- check audit ---

select \* from OrderAudit

1. **Server level Trigger**
2. **Server level Trigger** 🡪 **Work With** 🡪 **master database**

use master

CREATE TABLE MyAuditTable

(

AuditID INT PRIMARY KEY IDENTITY(1,1),

Event NVARCHAR(100) NOT NULL,

EventDate DATETIME NOT NULL

);

CREATE TRIGGER MyServerTrigger

ON ALL SERVER

AFTER CREATE\_LOGIN, ALTER\_LOGIN, DROP\_LOGIN

AS

BEGIN

-- Perform some action when a login is created, altered, or dropped on the server

-- For example, you could log the event to an audit table:

IF EVENTDATA().value('(/EVENT\_INSTANCE/EventType)[1]', 'nvarchar(100)') = 'CREATE\_LOGIN'

BEGIN

INSERT INTO MyAuditTable (Event, EventDate)

VALUES ('Login created', GETDATE())

END

ELSE IF EVENTDATA().value('(/EVENT\_INSTANCE/EventType)[1]', 'nvarchar(100)') = 'ALTER\_LOGIN'

BEGIN

INSERT INTO MyAuditTable (Event, EventDate)

VALUES ('Login altered', GETDATE())

END

ELSE IF EVENTDATA().value('(/EVENT\_INSTANCE/EventType)[1]', 'nvarchar(100)') = 'DROP\_LOGIN'

BEGIN

INSERT INTO MyAuditTable (Event, EventDate)

VALUES ('Login dropped', GETDATE())

END

ELSE

BEGIN

-- This should never happen, but just in case, log an error

INSERT INTO MyAuditTable (Event, EventDate)

VALUES ('Unknown event type', GETDATE())

END

END

--\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Testing Trigger \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- 1. create user/login

CREATE LOGIN [Boota] WITH PASSWORD = '12345'

-- 2. alter user/login

ALTER LOGIN [Boota] WITH PASSWORD = '123'

-- 3. drop user/logn

drop login [Boota]

--\_\_\_ see Results \_\_\_

select \* from MyAuditTable

1. **Trash Table after delete record from customer**

CREATE TRIGGER myTrigger

ON customer

AFTER DELETE

AS

BEGIN

SET NOCOUNT ON;

--if table not exist then 1st create table

if not exists (select \* from sys.tables where name = 'customerTrashTable')

begin

declare @sql Nvarchar(Max);

--\_\_\_\_\_\_\_\_ 1. customer create kr lain \_\_\_\_\_\_\_\_\_\_\_\_

--set @sql = N'

-- create table myTrashTable(

-- id int primary key,

-- name varchar(50),

-- ..,

-- ..

-- )

--'

--\_\_\_\_\_\_\_\_ 2. jis tables sa delete kr rhaa han usaa copy kr lains \_\_\_\_\_\_\_\_\_\_\_\_

set @sql = N'select Top 0 \* into customerTrashTable from customer'

exec sp\_executesql @sql

end

-- Insert the delete record int customerTrashTable

INSERT INTO customerTrashTable

SELECT \* FROM deleted; -- deleted is build in function

END

select \* from customer

--\_\_\_\_\_\_\_\_\_\_\_ test Trigere on delet customer \_\_

delete from customer where cId = 6

select \* from customerTrashTable

1. **We Can Create so many trigger but -🡪 its enough for biggener**