CREATE TABLE Employee

(

EmpId int Primary Key,

EmpName nvarchar(30),

Salary money,

Gender nvarchar(10),

DepartmentId nvarchar(20)

)

GO

-- Insert data into Employee table

INSERT INTO Employee VALUES (1,'Pranaya', 5000, 'Male', 'MST')

INSERT INTO Employee VALUES (2,'Priyanka', 5400, 'Female', 'FET')

INSERT INTO Employee VALUES (3,'Anurag', 6500, 'male', 'QA')

INSERT INTO Employee VALUES (4,'sambit', 4700, 'Male', 'MST')

INSERT INTO Employee VALUES (5,'Hina', 6600, 'Female', 'QA')

CREATE TRIGGER trInsertEmployee

ON Employee

FOR INSERT

AS

BEGIN

PRINT 'YOU CANNOT PERFORM INSERT OPERATION'

ROLLBACK TRANSACTION

END

INSERT INTO Employee VALUES (6, 'Saroj', 7600, 'Male', 'QA')

CREATE TRIGGER trUpdateEmployee

ON Employee

FOR UPDATE

AS

BEGIn

PRINT 'YOU CANNOT PERFORM UPDATE OPERATION'

ROLLBACK TRANSACTION

END

UPDATE Employee SET Salary = 90000 WHERE EmpId = 1;

CREATE TRIGGER trDeleteEmployee

ON Employee

FOR DELETE

AS

BEGIN

PRINT 'YOU CANNOT PERFORM DELETE OPERATION'

ROLLBACK TRANSACTION

END

DELETE FROM Employee WHERE EmpId = 1;

DROP TRIGGER trDeleteEmployee

DROP TRIGGER trInsertEmployee

DROP TRIGGER trUpdateEmployee

CREATE TRIGGER trAllDMLOperationsOnEmployee

ON Employee

FOR INSERT, UPDATE, DELETE

AS

BEGIN

PRINT 'YOU CANNOT PERFORM DML OPERATION'

ROLLBACK TRANSACTION

END

drop trigger trAllDMLOperationsOnEmployee

drop trigger trAllDMLOperationsOnEmployee

Create or ALTER TRIGGER trAllDMLOperationsOnEmployee

ON Employee

FOR INSERT, UPDATE, DELETE

AS

BEGIN

IF DATEPART(DW,GETDATE())= 3

BEGIN

PRINT 'DML OPERATIONS ARE RESTRICTED ON MONDAY'

ROLLBACK TRANSACTION

END

END

Create or ALTER TRIGGER trAllDMLOperationsOnEmployee

ON Employee

FOR INSERT, UPDATE, DELETE

AS

BEGIN

IF DATEPART(HH,GETDATE()) < 11

BEGIN

PRINT 'INVALID TIME'

ROLLBACK TRANSACTION

END

END

select \* from Employee;

Create database trg;

use trg

go

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

TotalAmount money

);

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

VALUES (1, 1001, '2022-01-01', 1000),

(2, 1002, '2022-02-01', 2000),

(3, 1003, '2022-03-01', 3000),

(4, 1004, '2022-04-01', 4000);

SELECT \* from orders;

SELECT \* FROM AuditOrders;

-- Create TRIGGER

CREATE TRIGGER tr\_AuditOrders

ON Orders

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

DECLARE @EventType VARCHAR(20)

SET @EventType =

CASE

WHEN EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted) THEN 'UPDATE'

WHEN EXISTS (SELECT \* FROM inserted) THEN 'INSERT'

WHEN EXISTS (SELECT \* FROM deleted) THEN 'DELETE'

END

IF @EventType = 'DELETE'

BEGIN

INSERT INTO AuditOrders (EventType, EventTime, OrderID, CustomerID, OrderDate, TotalAmount)

SELECT @EventType, GETDATE(), OrderID, CustomerID, OrderDate, TotalAmount

FROM deleted;

END

ELSE

BEGIN

INSERT INTO AuditOrders (EventType, EventTime, OrderID, CustomerID, OrderDate, TotalAmount)

SELECT @EventType, GETDATE(), OrderID, CustomerID, OrderDate, TotalAmount

FROM inserted;

END

END;

CREATE TABLE AuditOrders (

EventType VARCHAR(20),

EventTime DATETIME,

OrderID INT,

CustomerID INT,

OrderDate DATE,

TotalAmount money

);

-- Insert new record

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

VALUES (5, 1005, '2022-05-01', 5000);

-- Update record

UPDATE Orders

SET TotalAmount = TotalAmount + 1000

WHERE OrderID = 5;

-- Delete record

DELETE FROM Orders

WHERE OrderID = 5;