

--DDL trigger Sử dụng CSDL quản lý điểm trường đại học
--Tạo một bảng theo dõi các sự kiện và tạo DDL trigger cập nhật bảng theo các yêu cầu sau:

--1.Theo dõi việc tạo bảng mới: Ghi lại thông tin khi một bảng mới được tạo trong cơ sở dữ liệu.

```
CREATE TABLE SuKienBangMoi (  
    TenBang NVARCHAR(255),  
    ThoiGianTao DATETIME DEFAULT GETDATE(),  
    NguoiThucHien NVARCHAR(255)  
)  
  
CREATE TRIGGER TheoDoiBangMoi  
ON DATABASE  
FOR CREATE_TABLE  
AS  
BEGIN  
    DECLARE @TenBang NVARCHAR(255), @NguoiThucHien NVARCHAR(255);  
    -- Lấy thông tin bảng mới và người tạo  
    SELECT @TenBang = EVENTDATA().value('(/EVENT_INSTANCE/ObjectName)[1]',  
    'NVARCHAR(255)')  
    SELECT @NguoiThucHien = USER_SNAME()  
    -- Ghi lại sự kiện vào bảng SuKienBangMoi  
    INSERT INTO SuKienBangMoi (TenBang, NguoiThucHien)  
    VALUES (@TenBang, @NguoiThucHien);  
END;  
--kiểm tra  
CREATE TABLE SampleTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100)  
);  
SELECT * FROM SuKienBangMoi
```

--2.Theo dõi việc xóa bảng: Ghi log khi một bảng bị xóa khỏi cơ sở dữ liệu.

```
CREATE TABLE SuKienXoaBang (  
    TenBang NVARCHAR(255),  
    ThoiGianXoa DATETIME DEFAULT GETDATE(),  
    NguoiThucHien NVARCHAR(255)  
)  
  
CREATE TRIGGER TheoDoiXoaBang  
ON DATABASE  
FOR DROP_TABLE  
AS  
BEGIN  
    DECLARE @TenBang NVARCHAR(255), @NguoiThucHien NVARCHAR(255);
```

```

-- Lấy tên bảng bị xóa và người thực hiện
SELECT @TenBang =
EVENTDATA().value('/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(255)');
SELECT @NguoiThucHien = SUSER_SNAME();
-- Ghi lại thông tin vào bảng ghi log
INSERT INTO SuKienXoaBang (TenBang, NguoiThucHien)
VALUES (@TenBang, @NguoiThucHien);
END

```

--3. Theo dõi việc sửa đổi bảng (CREATE, ALTER, DROP TABLE): Ghi lại các thay đổi cấu trúc của bảng như thêm, sửa, hoặc xóa cột.

```

CREATE TABLE SuKienThayDoiBang (
    TenBang NVARCHAR(255),
    ThoiGianThayDoi DATETIME DEFAULT GETDATE(),
    HanhDong NVARCHAR(50),
    NguoiThucHien NVARCHAR(255)
)

CREATE TRIGGER TheoDoiThayDoiBang
ON DATABASE
FOR CREATE_TABLE, ALTER_TABLE, DROP_TABLE
AS
BEGIN
    DECLARE @TenBang NVARCHAR(255), @HanhDong NVARCHAR(50),
@NguoiThucHien NVARCHAR(255);
    -- Lấy tên bảng và hành động từ EVENTDATA()
    SELECT @TenBang =
EVENTDATA().value('/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(255)')
    SELECT @HanhDong =
EVENTDATA().value('/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(50)')
    SELECT @NguoiThucHien = SUSER_SNAME()
    -- Ghi lại thông tin vào bảng ghi log
    INSERT INTO SuKienThayDoiBang (TenBang, HanhDong, NguoiThucHien)
    VALUES (@TenBang, @HanhDong, @NguoiThucHien)
END

```

--4. Theo dõi việc tạo view (CREATE, ALTER, DROP VIEW): Ghi lại khi một view mới được tạo, được sửa, xóa trong cơ sở dữ liệu.

```

CREATE TABLE SuKienThayDoiView (
    TenView NVARCHAR(255),
    ThoiGianThayDoi DATETIME DEFAULT GETDATE(),
    HanhDong NVARCHAR(50),
    NguoiThucHien NVARCHAR(255)
)

```

```

CREATE TRIGGER TheoDoiThayDoiView
ON DATABASE
FOR CREATE_VIEW, ALTER_VIEW, DROP_VIEW
AS
BEGIN
    DECLARE @TenView NVARCHAR(255), @HanhDong NVARCHAR(50),
@NguoiThucHien NVARCHAR(255)
    -- Lấy tên view và hành động từ EVENTDATA()
    SELECT @TenView =
EVENTDATA().value('/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(255)')
    SELECT @HanhDong =
EVENTDATA().value('/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(50)')
    SELECT @NguoiThucHien = SUSER_SNAME()
    -- Ghi lại thông tin vào bảng ghi log
    INSERT INTO SuKienThayDoiView (TenView, HanhDong, NguoiThucHien)
    VALUES (@TenView, @HanhDong, @NguoiThucHien)
END

```

--5.Theo dõi việc tạo trigger mới (CREATE, ALTER, DROP TRIGGER): Ghi lại khi một trigger mới được tạo, được sửa, xóa trong hệ thống.

```

CREATE TABLE SuKienThayDoiTrigger (
    TenTrigger NVARCHAR(255),
    ThoiGianThayDoi DATETIME DEFAULT GETDATE(),
    HanhDong NVARCHAR(50),
    NguoiThucHien NVARCHAR(255)
)

```

```

CREATE TRIGGER TheoDoiThayDoiTrigger
ON DATABASE
FOR CREATE_TRIGGER, ALTER_TRIGGER, DROP_TRIGGER
AS
BEGIN
    DECLARE @TenTrigger NVARCHAR(255), @HanhDong NVARCHAR(50),
@NguoiThucHien NVARCHAR(255);
    -- Lấy tên trigger và hành động từ EVENTDATA()
    SELECT @TenTrigger =
EVENTDATA().value('/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(255)');
    SELECT @HanhDong =
EVENTDATA().value('/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(50)');
    SELECT @NguoiThucHien = SUSER_SNAME();
    -- Ghi lại thông tin vào bảng ghi log
    INSERT INTO SuKienThayDoiTrigger (TenTrigger, HanhDong,
NguoiThucHien)
    VALUES (@TenTrigger, @HanhDong, @NguoiThucHien)
END

```

--6.Theo dõi việc cấp quyền (GRANT): Ghi lại khi quyền được cấp cho người dùng hoặc vai trò.

```
CREATE TABLE GrantAudit (  
    UserName NVARCHAR(255),  
    RoleName NVARCHAR(255),  
    PermissionGranted NVARCHAR(255),  
    GrantDate DATETIME,  
    GrantedBy NVARCHAR(255)  
)
```

```
CREATE TRIGGER trg_AuditGrant  
ON sys.database_permissions  
AFTER INSERT  
AS  
BEGIN  
    INSERT INTO GrantAudit (UserName, RoleName, PermissionGranted,  
GrantDate, GrantedBy)  
    SELECT  
        USER_NAME(major_id),  
        USER_NAME(grantee_principal_id),  
        permission_name,  
        GETDATE(),  
        SYSTEM_USER  
    FROM inserted;  
END;
```

--7.Theo dõi việc thu hồi quyền (REVOKE): Ghi lại khi quyền bị thu hồi từ người dùng hoặc vai trò.

```
CREATE TABLE RevokeAudit (  
    UserName NVARCHAR(255),  
    RoleName NVARCHAR(255),  
    PermissionRevoked NVARCHAR(255),  
    RevokeDate DATETIME,  
    RevokedBy NVARCHAR(255)  
)
```

```
CREATE TRIGGER trg_AuditRevoke  
ON sys.database_permissions  
AFTER DELETE  
AS  
BEGIN  
    INSERT INTO RevokeAudit (UserName, RoleName, PermissionRevoked,  
RevokeDate, RevokedBy)  
    SELECT
```

```

        USER_NAME(major_id),
        USER_NAME(grantee_principal_id),
        permission_name,
        GETDATE(),
        SYSTEM_USER
    FROM deleted;
END;

--8.Không cho phép xóa bảng SinhVien
CREATE TABLE SuKien (
    TenSuKien NVARCHAR(255),
    ThoiGian DATETIME DEFAULT GETDATE(),
    NguoiThucHien NVARCHAR(255)
)

CREATE TRIGGER PreventDropSinhVien
ON DATABASE
FOR DROP_TABLE
AS
BEGIN
    DECLARE @TenBang NVARCHAR(255);
    SELECT @TenBang =
EVENT_DATA.value('(/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(255)');

    IF @TenBang = 'SinhVien'
    BEGIN
        ROLLBACK TRANSACTION;
        INSERT INTO SuKien (TenSuKien, NguoiThucHien)
        VALUES ('Cố gắng xóa bảng SinhVien', SYSTEM_USER);

        RAISERROR('Không thể xóa bảng SinhVien.', 16, 1);
    END
END;

```