

Advanced SQL Server Procedures and Triggers

Advanced Stored Procedures

Procedure 1 - Transfer GPA Between Students

```
CREATE PROCEDURE sp_TransferGPA
    @FromStudentID INT,
    @ToStudentID INT,
    @Amount DECIMAL(3,2)
AS
BEGIN
    BEGIN TRANSACTION;
    UPDATE Students SET GPA = GPA - @Amount WHERE StudentID = @FromStudentID;
    UPDATE Students SET GPA = GPA + @Amount WHERE StudentID = @ToStudentID;

    IF @@ERROR <> 0
    BEGIN
        ROLLBACK TRANSACTION;
        RETURN;
    END

    COMMIT TRANSACTION;
END;
```

Procedure 2 - Generate GPA Report

```
CREATE PROCEDURE sp_GenerateGPAReport
AS
BEGIN
    SELECT
        CASE
            WHEN GPA >= 3.5 THEN 'High Honors'
            WHEN GPA >= 3.0 THEN 'Honors'
            ELSE 'Regular'
        END AS Category,
        COUNT(*) AS StudentCount
    FROM Students
    GROUP BY
        CASE
            WHEN GPA >= 3.5 THEN 'High Honors'
            WHEN GPA >= 3.0 THEN 'Honors'
            ELSE 'Regular'
        END;
END;
```

Advanced Triggers

Advanced SQL Server Procedures and Triggers

Trigger 1 - Prevent Multiple Updates in One Day

```
CREATE TRIGGER trg_LimitDailyUpdate
ON Students
AFTER UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted i
        JOIN GPACHanges g ON i.StudentID = g.StudentID
        WHERE CONVERT(DATE, g.ChangedOn) = CONVERT(DATE, GETDATE())
    )
    BEGIN
        RAISERROR('A student's GPA can only be updated once per day.', 16, 1);
        ROLLBACK;
    END
END;
```

Trigger 2 - Dynamic GPA Category Logging

```
CREATE TABLE GPACategoryLog (
    StudentID INT,
    OldCategory NVARCHAR(50),
    NewCategory NVARCHAR(50),
    ChangedAt DATETIME
);

CREATE TRIGGER trg_LogGPACategoryChange
ON Students
AFTER UPDATE
AS
BEGIN
    INSERT INTO GPACategoryLog (StudentID, OldCategory, NewCategory, ChangedAt)
    SELECT
        d.StudentID,
        CASE WHEN d.GPA >= 3.5 THEN 'High Honors'
             WHEN d.GPA >= 3.0 THEN 'Honors'
             ELSE 'Regular' END,
        CASE WHEN i.GPA >= 3.5 THEN 'High Honors'
             WHEN i.GPA >= 3.0 THEN 'Honors'
             ELSE 'Regular' END,
        GETDATE()
    FROM inserted i
    JOIN deleted d ON i.StudentID = d.StudentID
    WHERE
        CASE WHEN d.GPA >= 3.5 THEN 'High Honors'
```

Advanced SQL Server Procedures and Triggers

```
        WHEN d.GPA >= 3.0 THEN 'Honors'
        ELSE 'Regular' END <>
CASE WHEN i.GPA >= 3.5 THEN 'High Honors'
      WHEN i.GPA >= 3.0 THEN 'Honors'
      ELSE 'Regular' END;

END;
```