

SQL Server Practice Assignments: Procedures & Triggers

Practice Assignments: Stored Procedures

Q: 1. Create a procedure to update a student's name.

Procedure should take StudentID and new name as input parameters. Update the name accordingly.

Answer:

```
CREATE PROCEDURE sp_UpdateStudentName
    @StudentID INT,
    @NewName NVARCHAR(100)
AS
BEGIN
    UPDATE Students SET Name = @NewName WHERE StudentID = @StudentID;
END;
```

Q: 2. Create a procedure to delete all students with GPA below a given threshold.

Accept a GPA threshold as input and delete all students with GPA less than this value.

Answer:

```
CREATE PROCEDURE sp_DeleteLowGPA
    @MinGPA DECIMAL(3,2)
AS
BEGIN
    DELETE FROM Students WHERE GPA < @MinGPA;
END;
```

Q: 3. Write a procedure to return students enrolled in the last 30 days.

No input needed. Return all students with EnrollmentDate in the last 30 days.

Answer:

```
CREATE PROCEDURE sp_Recent30Days
AS
BEGIN
    SELECT * FROM Students
    WHERE EnrollmentDate >= DATEADD(DAY, -30, GETDATE());
END;
```

Practice Assignments: Triggers

Q: 1. Create a trigger to prevent deletion of a student with GPA above 3.8.

SQL Server Practice Assignments: Procedures & Triggers

Use an AFTER DELETE trigger. If deleted GPA > 3.8, roll back the operation.

Answer:

```
CREATE TRIGGER trg_PreventHighGPADelete
ON Students
AFTER DELETE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM deleted WHERE GPA > 3.8)
    BEGIN
        RAISERROR('Cannot delete student with GPA above 3.8', 16, 1);
        ROLLBACK;
    END
END;
```

Q: 2. Create a trigger to track GPA updates in a log table.

If a student's GPA is updated, insert a record into GPACHanges table.

Answer:

```
CREATE TRIGGER trg_LogGPACHange
ON Students
AFTER UPDATE
AS
BEGIN
    INSERT INTO GPACHanges(StudentID, OldGPA, NewGPA, ChangedOn)
    SELECT d.StudentID, d.GPA, i.GPA, GETDATE()
    FROM inserted i
    JOIN deleted d ON i.StudentID = d.StudentID
    WHERE i.GPA <> d.GPA;
END;
```

Q: 3. Write a trigger to set LastUpdated field whenever a record is updated.

Use AFTER UPDATE trigger and update LastUpdated to GETDATE().

Answer:

```
CREATE TRIGGER trg_SetLastUpdated
ON Students
AFTER UPDATE
AS
BEGIN
```

SQL Server Practice Assignments: Procedures & Triggers

```
UPDATE s
SET LastUpdated = GETDATE()
FROM Students s
JOIN inserted i ON s.StudentID = i.StudentID;
END;
```