

```
-- Create a new database
```

```
CREATE DATABASE CollegeManagement;  
GO
```

```
-- Use the newly created database
```

```
USE CollegeManagement;  
GO
```

```
-- Create the SubjectAllotments table
```

```
CREATE TABLE SubjectAllotments (  
    StudentID varchar(50),  
    SubjectID varchar(50),  
    Is_Valid bit  
);  
GO
```

```
-- Create the SubjectRequest table
```

```
CREATE TABLE SubjectRequest (  
    StudentID varchar(50),  
    SubjectID varchar(50)  
);  
GO
```

```
-- Insert sample data into SubjectAllotments
```

```
INSERT INTO SubjectAllotments (StudentID, SubjectID, Is_Valid)  
VALUES  
( '159103036', 'P01491', 1),  
( '159103036', 'P01492', 0),  
( '159103036', 'P01493', 0),  
( '159103036', 'P01494', 0),  
( '159103036', 'P01495', 0);  
GO
```

```
-- Insert a sample request into SubjectRequest
```

```
INSERT INTO SubjectRequest (StudentID, SubjectID)  
VALUES  
( '159103036', 'P01496');  
GO
```

```
-- Create the stored procedure
```

```
CREATE PROCEDURE UpdateSubjectAllotments  
    @StudentID varchar(50),  
    @RequestedSubjectID varchar(50)  
AS  
BEGIN  
    SET NOCOUNT ON;
```

```
-- Check if the student exists in the SubjectAllotments table
```

```
IF EXISTS (SELECT 1 FROM SubjectAllotments WHERE StudentID = @StudentID)  
BEGIN
```

```
-- Check if the current subject is different from the requested subject
IF EXISTS (SELECT 1 FROM SubjectAllotments WHERE StudentID = @StudentID AND
Is_Valid = 1 AND SubjectID != @RequestedSubjectID)
BEGIN
    -- Invalidate the current subject
    UPDATE SubjectAllotments
    SET Is_Valid = 0
    WHERE StudentID = @StudentID AND Is_Valid = 1;

    -- Insert the new subject as valid
    INSERT INTO SubjectAllotments (StudentID, SubjectID, Is_Valid)
    VALUES (@StudentID, @RequestedSubjectID, 1);
END
ELSE IF NOT EXISTS (SELECT 1 FROM SubjectAllotments WHERE StudentID =
@StudentID AND Is_Valid = 1)
BEGIN
    -- If no current subject is valid, insert the new subject as valid
    --Rakshit Gupta
    INSERT INTO SubjectAllotments (StudentID, SubjectID, Is_Valid)
    VALUES (@StudentID, @RequestedSubjectID, 1);
END
END
ELSE
BEGIN
    -- If the student does not exist in the SubjectAllotments table, insert the
    new subject as valid
    INSERT INTO SubjectAllotments (StudentID, SubjectID, Is_Valid)
    VALUES (@StudentID, @RequestedSubjectID, 1);
END
END;
GO

-- Execute the stored procedure
EXEC UpdateSubjectAllotments @StudentID = '159103036', @RequestedSubjectID = 'P01496';
GO

-- Check the updated SubjectAllotments table
SELECT * FROM SubjectAllotments;
GO
```