

**Problem Statement:**

A college needs to develop a system that tracks the Open Elective Subjects of their students. During the start of the year a lot of students change their subjects and college wants to preserve that entire timeline of the data.

For example, Mohit had selected “Eco philosophy” as his Open Elective subject but later he wished to switch to “Basics of Accounting” in this scenario the college wants the visibility of Mohit’s previous choice as well instead of just the active one:

There is one table that holds the record of this change, and it is called SubjectAllotments

The below table has the subject to student mapping with the preference

Note: The details of the columns are as following:

* StudentID varchar
* SubjectID varchar
* Is\_Valid bit

|  |  |  |
| --- | --- | --- |
| StudentId | SubjectId | Is\_valid |
| 159103036 | PO1491 | 1 |
| 159103036 | PO1492 | 0 |
| 159103036 | PO1493 | 0 |
| 159103036 | PO1494 | 0 |
| 159103036 | PO1495 | 0 |

(Table Name: SubjectAllotments)

In the above example we can see the student’s active subject is “PO1491” but at some point the student was allotted other subjects.

When a student requests an allotment the request details are stored in a table called “SubjectRequest”

|  |  |
| --- | --- |
| StudentId | SubjectId |
| 159103036 | PO1496 |

(Table Name: SubjectRequest)

We can see that the student has requested a change.

Here we have to check the current subject of the student (where is\_valid=1) is different from the request or not, if the subject is different we insert another record in the table making the new record valid and changing the previously valid record to invalid, the output is indicated in the below table:

|  |  |  |
| --- | --- | --- |
| StudentId | SubjectId | Is\_valid |
| 159103036 | PO1496 | 1 |
| 159103036 | PO1491 | 0 |
| 159103036 | PO1492 | 0 |
| 159103036 | PO1493 | 0 |
| 159103036 | PO1494 | 0 |
| 159103036 | PO1495 | 0 |

(Table Name: SubjectAlottments)

If the student id that is present in the SubjectRequest Table does not exist in the SubjectAlottments table then we simply insert the requested subject as a valid record in the SubjectAllotments Table

Your Task is to write a Stored Procedure to implement the above stated workflow.

CREATE TABLE SubjectAllotments (StudentId varchar(10), SubjectId varchar(10), Is\_valid bit)

INSERT INTO SubjectAllotments VALUES(159103036 ,'PO1491', 1)

INSERT INTO SubjectAllotments VALUES(159103036 ,'PO1492', 0)

INSERT INTO SubjectAllotments VALUES(159103036 ,'PO1493', 0)

INSERT INTO SubjectAllotments VALUES(159103036 ,'PO1494', 0)

INSERT INTO SubjectAllotments VALUES(159103036 ,'PO1495', 0)

GO

CREATE PROCEDURE UpdateSubjectAllotments

(

@StudentId VARCHAR(50),

@SubjectId VARCHAR(50)

)

AS

BEGIN

-- Check if the student exists in SubjectAllotments

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

-- Check if the requested subject is different from the current subjectIF EXISTS (SELECT 1 FROM SubjectAllotments WHERE StudentId = @StudentId AND SubjectId = @SubjectId AND Is\_Valid = 1)

BEGIN

-- Student's current subject is the same as the requested subject, no update needed. RETURN;

END

ELSE

BEGIN

-- Deactivate the current valid 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, @SubjectId, 1);

END

END

ELSE

BEGIN

-- If the student doesn't exist in SubjectAllotments, insert the requested subject as valid. INSERT INTO SubjectAllotments (StudentId, SubjectId, Is\_Valid)

VALUES (@StudentId, @SubjectId, 1);

END

END

GO

EXEC UpdateSubjectAllotments @StudentId = '159103036', @SubjectId = 'PO1496';

SELECT \* from SubjectAllotments;

A screenshot of a computer

Description automatically generated