# Stored Procedure Documentation: Subject Allotment System

## 1. Problem Statement

A college tracks Open Elective Subject allotments for students. When students request subject changes, their previous and current choices must be maintained. The data is managed across two tables:  
  
1. SubjectAllotments - Contains historical and current subject allotments with a validity flag.  
2. SubjectRequest - Contains current student requests for subject changes.

## 2. Tables Involved

• SubjectAllotments(StudentId, SubjectId, Is\_Valid)  
• SubjectRequest(StudentId, SubjectId)

## 3. Objective

Develop a stored procedure to:  
- Check the current valid subject for each student.  
- If the requested subject differs and already exists, mark previous as invalid and requested as valid.  
- If the requested subject does not exist, insert it as valid and mark the previous as invalid.

## 4. Procedure Logic

1. Loop through each record in SubjectRequest.  
2. Retrieve the currently valid subject for the student.  
3. If the requested subject exists in SubjectAllotments:  
 a. If it is different from the valid one, update the validity.  
4. If the requested subject does not exist:  
 a. Insert it as valid and set existing one as invalid.

## 5. SQL Stored Procedure Summary

The stored procedure uses a cursor to iterate over SubjectRequest. It ensures only one active (Is\_Valid = 1) subject per student and updates SubjectAllotments accordingly.

## 6. Execution & Result

Run the procedure with:  
EXEC UpdateSubjectAllotment;  
  
Then view changes with:  
SELECT \* FROM SubjectAllotments WHERE StudentId = 'YourID';  
  
The result will reflect the newly requested subject as valid and the old one as invalid.