

--User-defined Functions

```

/*
This function generates the grade report for a student.Only a faculty can generate the report.
*/

CREATE FUNCTION vamseepr2.UFGradeReport
(
    --Parameters
    @facultyId INTEGER = NULL,
    @studentId INTEGER = NULL
)
RETURNS @gradeTable TABLE(StudentName NVARCHAR(100), Course NVARCHAR(100), Grade CHAR(1))
AS
BEGIN
    -- Variable
    DECLARE @isFaculty INTEGER, @isStudent INTEGER

    -- @isFaculty is 0 if no record is found
    SELECT @isFaculty = ( SELECT COUNT(E.EmployeeId)
                        FROM vamseepr2.Employees E
                        INNER JOIN vamseepr2.JobInformation JI
                        ON E.JobId = JI.JobId
                        INNER JOIN vamseepr2.JobPositionType JP
                        ON JI.JobPositionTypeId = JP.JobPositionTypeId
                        WHERE JP.PostionType = 'FACULTY' AND E.EmployeeId = @facultyId)

    -- @isStudent is 0 if no record is found
    SELECT @isStudent = ( SELECT COUNT(S.StudentId)
                        FROM vamseepr2.Students S
                        WHERE S.StudentId = @studentId)

    IF @isFaculty = 0
    BEGIN
        INSERT INTO @gradeTable(StudentName) VALUES('Error! Cannot generate grade report, not a valid faculty id')
    END

    ELSE
    BEGIN
        IF @isStudent = 0
        BEGIN
            INSERT INTO @gradeTable(StudentName) VALUES('Error: Cannot generate grade report, not a valid student id')
        END

        ELSE
        BEGIN
            --Generate the grade report of the student
            INSERT INTO @gradeTable
            SELECT S.FirstName+' '+S.LastName, C.CourseCode+''+CAST(C.CourseNumber AS NVARCHAR), CG.Grade
            FROM vamseepr2.Students S
            INNER JOIN vamseepr2.Enrollment E
            ON S.StudentId = E.StudentId
            INNER JOIN vamseepr2.CourseSchedule CS
            ON E.CourseScheduleId = CS.CourseScheduleId
            INNER JOIN vamseepr2.Courses C
            ON CS.CourseCode = C.CourseCode AND CS.CourseNumber = C.CourseNumber
            INNER JOIN vamseepr2.CourseGrade CG
            ON E.GradeId = CG.GradeId
            WHERE S.StudentId = @studentId;

        END
    END
    RETURN
END

```

-----

/\*This function returns a list of suitable classrooms and the building for a course\*/

```
CREATE FUNCTION vamseepr2.UFClassroomReport
(
    --Parameters
    @courseCode    NVARCHAR(3) = NULL,
    @courseNumber  INTEGER      = NULL
)
RETURNS @classroomTable TABLE(RoomNumber NUMERIC(3,0), BuildingName NVARCHAR(50))
AS
BEGIN
    -- Variable
    DECLARE @courseSeats INTEGER

    SELECT @courseSeats = CS.CourseSeats
        FROM vamseepr2.CourseSchedule CS
        WHERE CS.CourseCode = @courseCode AND CS.CourseNumber = @courseNumber;

    INSERT INTO @classroomTable
        SELECT C.RoomNumber, B.BuildingName FROM vamseepr2.Classroom C
            LEFT OUTER JOIN vamseepr2.Buildings B
                ON C.BuildingId = B.BuildingId
            WHERE C.MaxSeating > @courseSeats;

    RETURN
END
```