CSE 581

**Lab 12: Stored Procedures**

*Do this against YOUR own database.*

***For the entire lab, provide all of your SQL as text, inserted at the beginning of the document.***

***For each step also provide a screenshot of execution, showing the SQL and the results.***

***Please make sure both the SQL and the screenshots are marked w/ the question number that they are answering.***

**Steps:**

1. Create a stored procedure. Input for the stored procedure will be FacultyId, Student Id, Course Id and a Numerical Grade.

The stored procedure will attempt to assign a grade to the student, for the class that they are enrolled in, assuming it is legal to do so. The stored procedure will do the following:

1. If the user attempting to assign the grade (FacultyId) is not the faculty teaching the course, **print a message** that says “Error: You are not allowed to assign grades for this course.”, **and quit**.
2. Assuming A passed, and if the student is not enrolled in the class, **print a message** that says “Error: The student is not taking the course you specified.”, **and quit**.
3. Assuming B passed, and if the student already has a grade for that class, **change it** **to the new grade**, and **print a message** that says “Success, with a warning - Student’s existing grade OLD\_GRADE was changed to NEW\_GRADE.” Please replace the OLD\_GRADE with the existing grade, and the NEW\_GRADE with the new grade. **Then quit**.
4. Assuming B passed and C did not execute (the student does not have a grade for the course yet), **insert the new grade** and **print a message** “Success.”, **then quit**.

CREATE PROCEDURE dbo.Grading\_ForLab12(@FacultyId as VARCHAR(20), @StudentId as VARCHAR(20),

@CourseId as INTEGER, @NumericalGrade as DECIMAL(5, 2))

AS

DECLARE @Faculty VARCHAR(20)

SELECT @Faculty = (SELECT Faculty from Courses where CourseId = @CourseId)

IF (@Faculty < @FacultyId)

begin

print 'ERROR: You are not allowed to assign grade for this course'

end

else IF (@Faculty > @FacultyId)

begin

print 'ERROR: You are not allowed to assign grade for this course'

return

end

DECLARE @Enroll VARCHAR(1)

SELECT @Enroll = (SELECT 'R' from CourseEnrollment

where CourseId = @CourseId AND StudentId = @StudentId)

If @Enroll is NUll

begin

print 'Error: The student is not taking the course you specified'

return

end

DECLARE @GradeIsPresent VARCHAR(1)

SELECT @GradeIsPresent = (SELECT 'Y' from CourseEnrollment

where (StudentId = @StudentId) AND (CourseId = @CourseId) AND FinalGrade IS NOT NULL)

IF @GradeIsPresent IS NOT NULL

begin

print 'Success, with a warning - Student’s existing grade OLD\_GRADE was changed to NEW\_GRADE'

update CourseEnrollment

set FinalGrade=@NumericalGrade

where (CourseId=@CourseId) AND (StudentId=@StudentId)

return

end

IF @GradeIsPresent IS NULL

begin

update CourseEnrollment

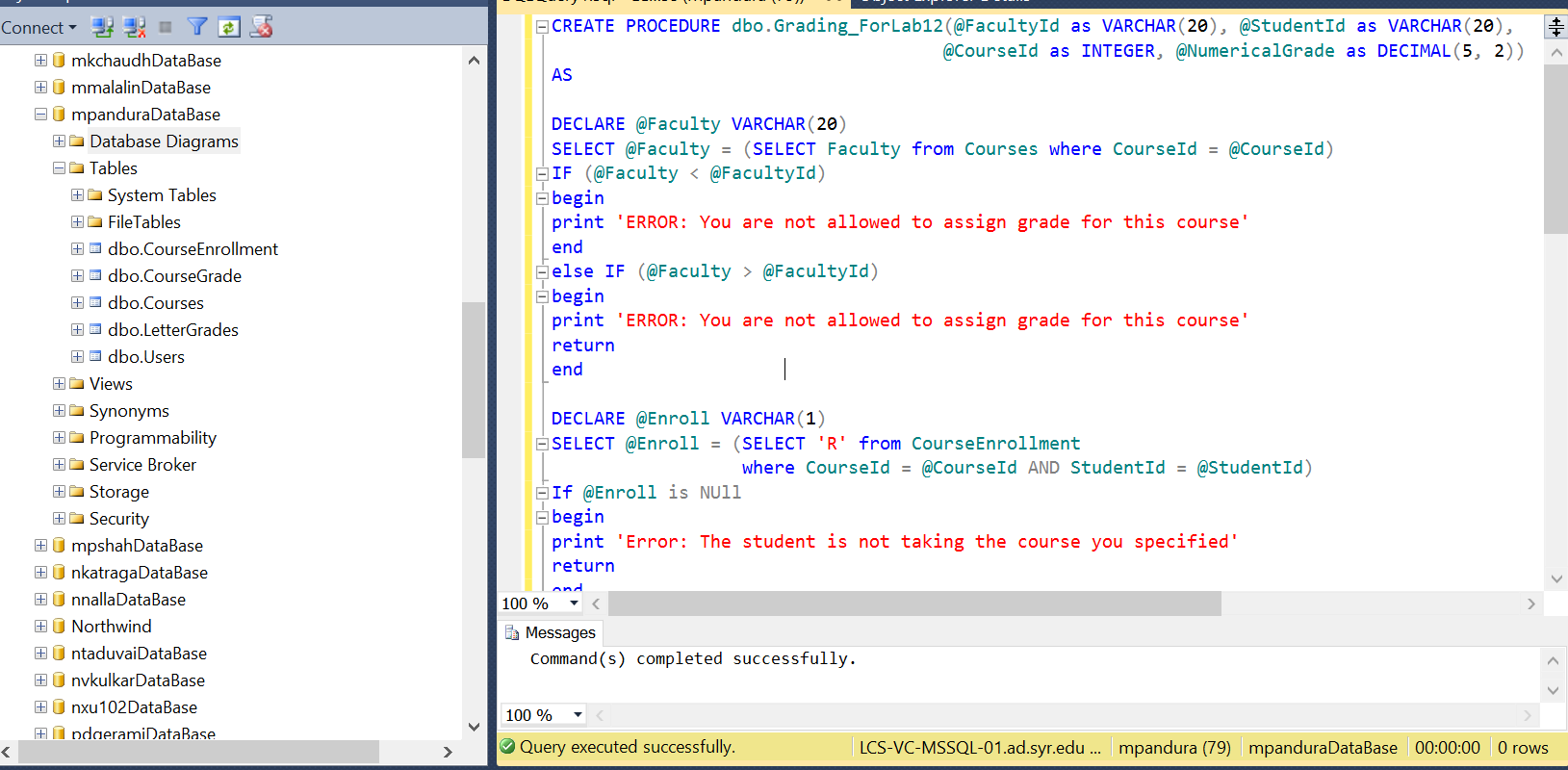
set FinalGrade=@NumericalGrade

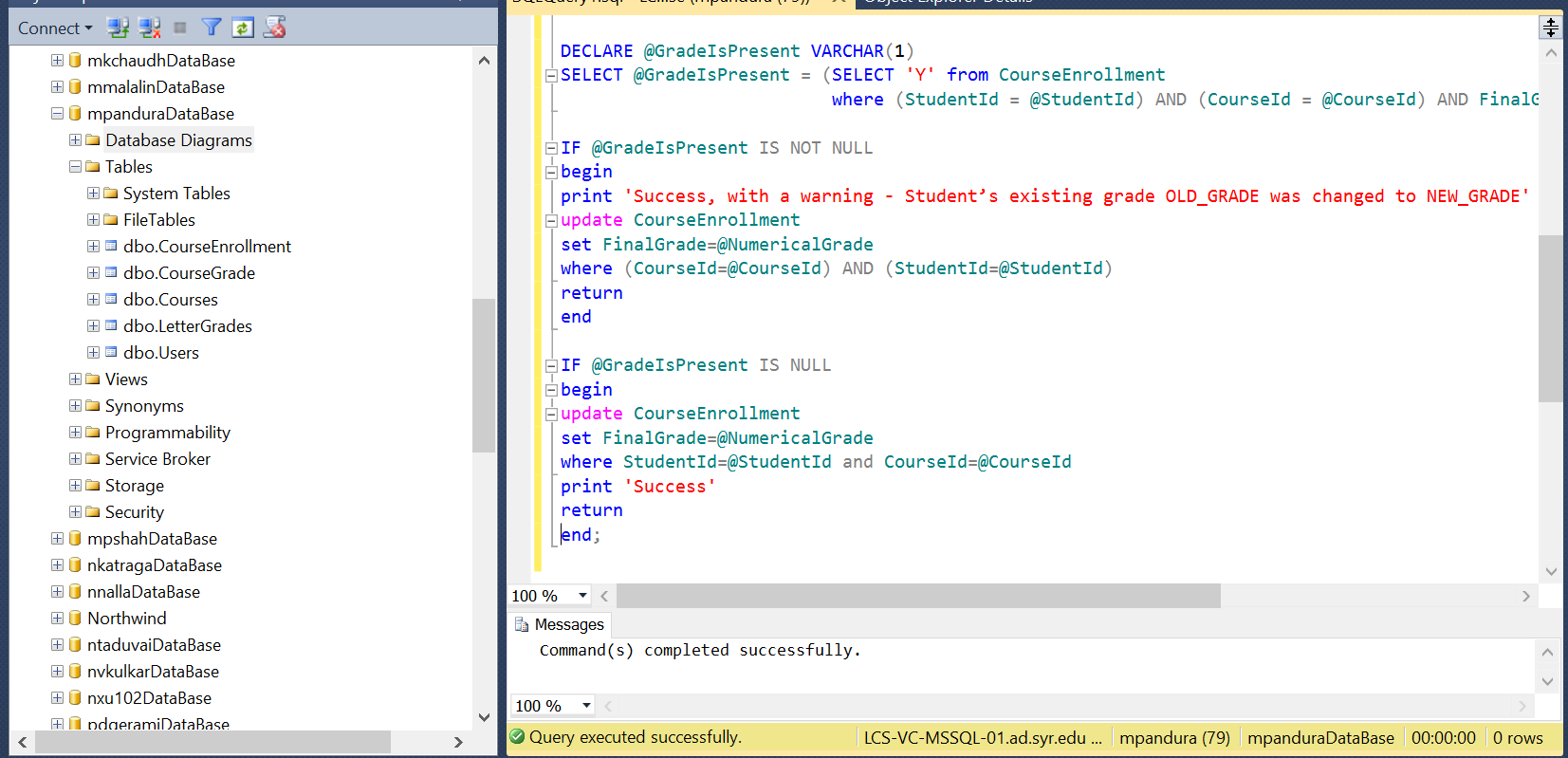
where StudentId=@StudentId and CourseId=@CourseId

print 'Success'

return

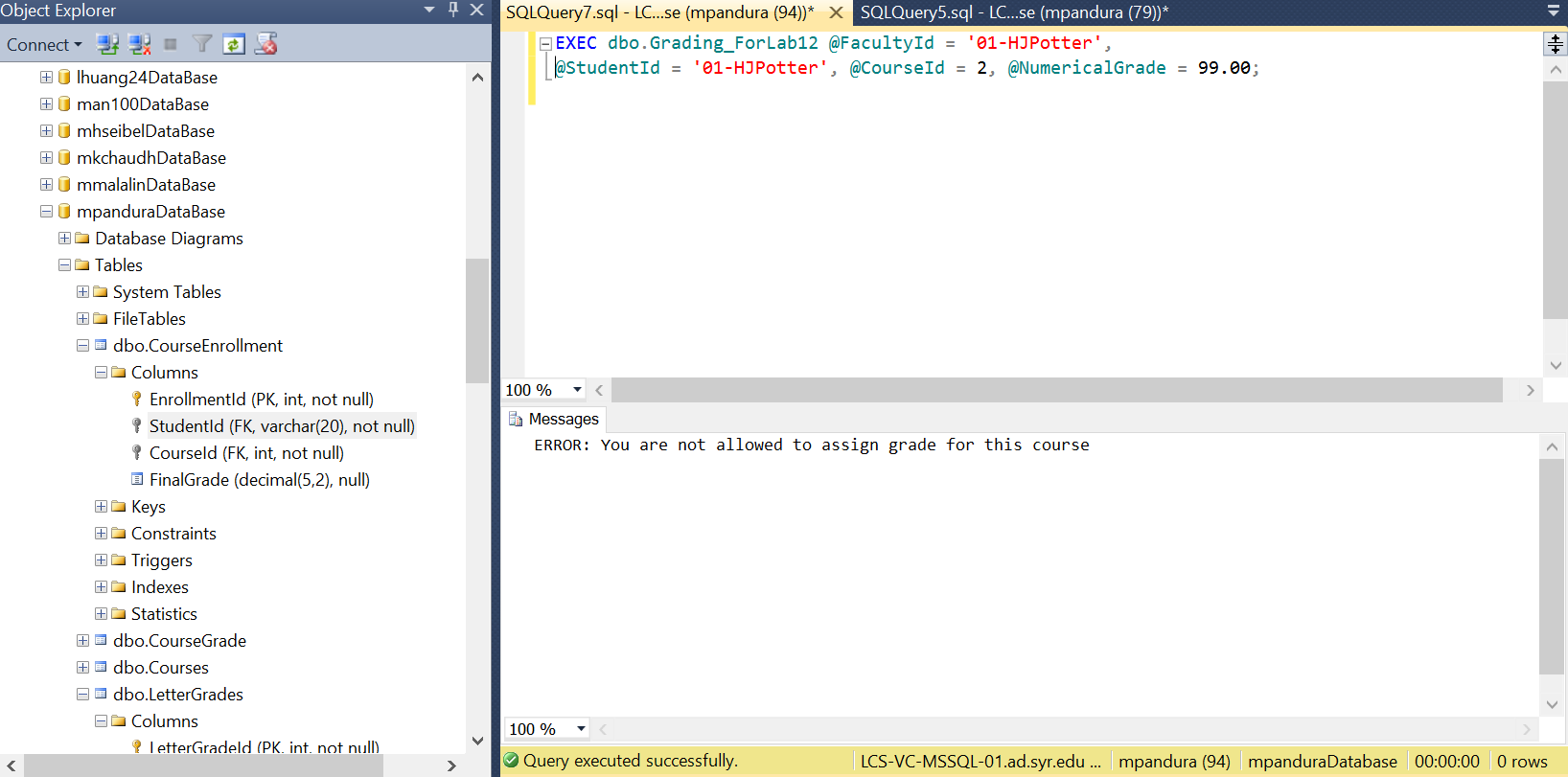
end;





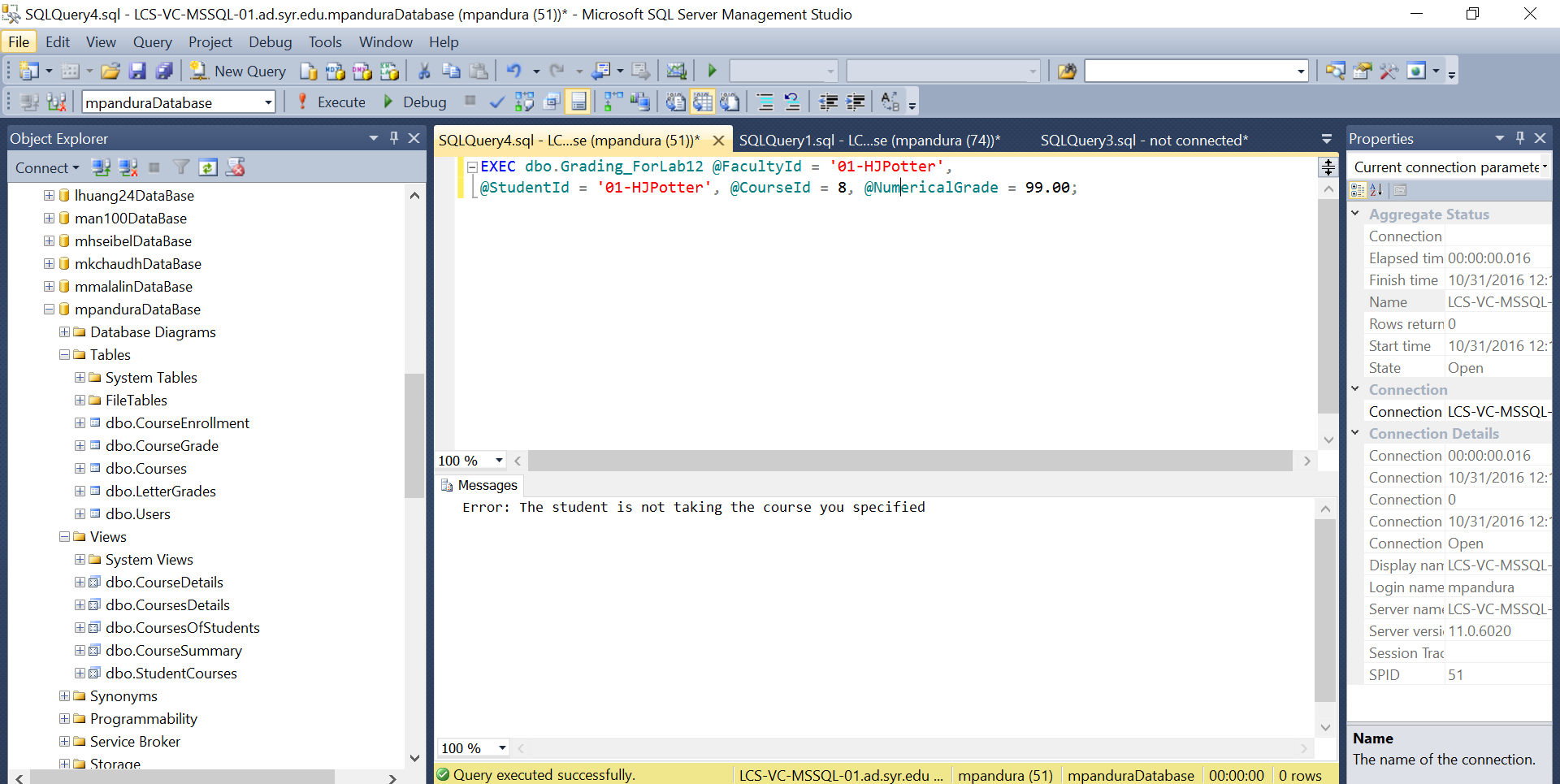
1. Run the stored procedure to prove[[1]](#footnote-1) that all 4 cases work[[2]](#footnote-2).
2. EXEC dbo.Grading\_ForLab12 @FacultyId = '01-HJPotter',

@StudentId = '01-HJPotter', @CourseId = 2, @NumericalGrade = 99.00;



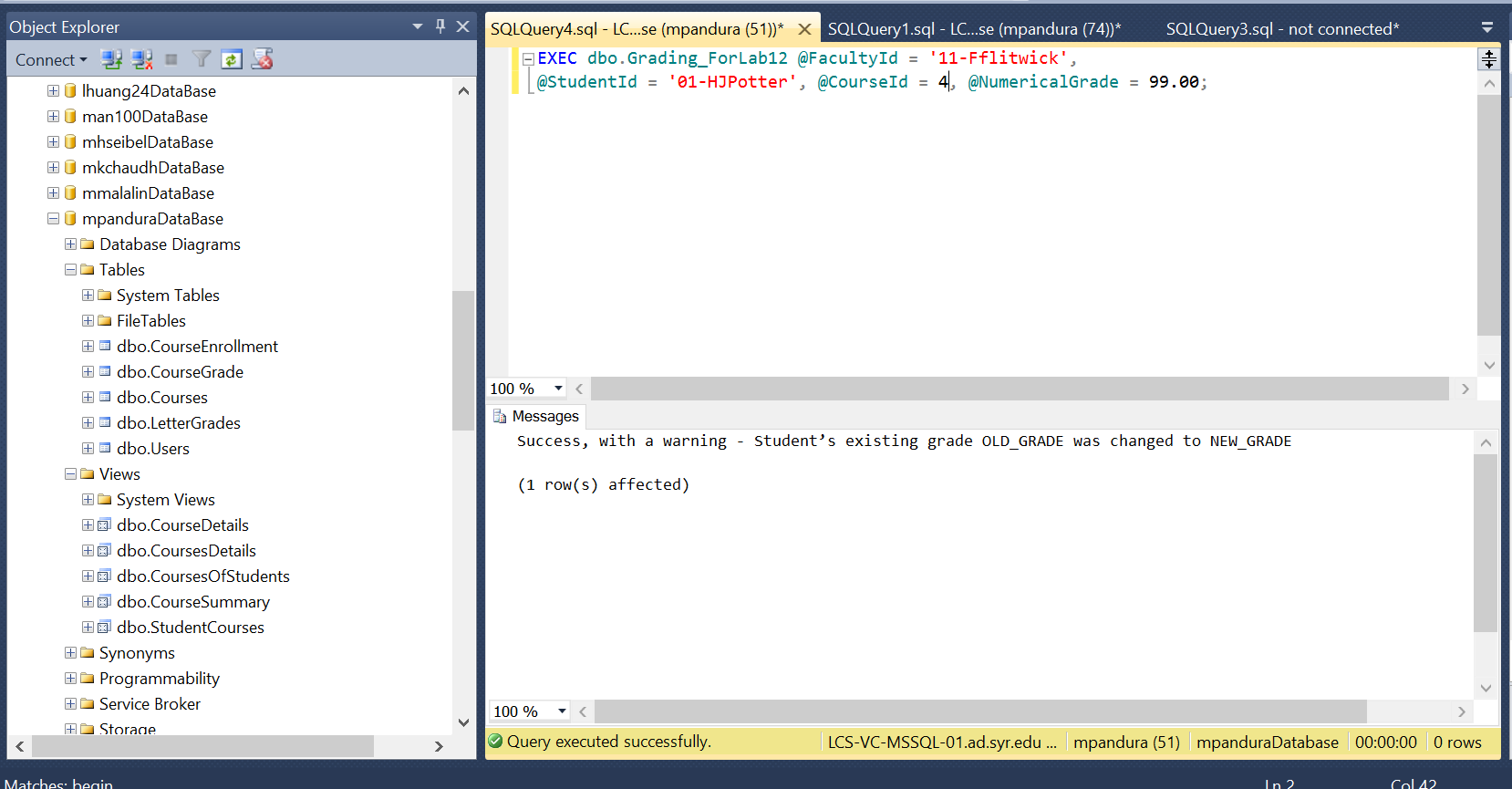
b. EXEC dbo.Grading\_ForLab12 @FacultyId = '01-HJPotter',

@StudentId = '01-HJPotter', @CourseId = 8, @NumericalGrade = 99.00;



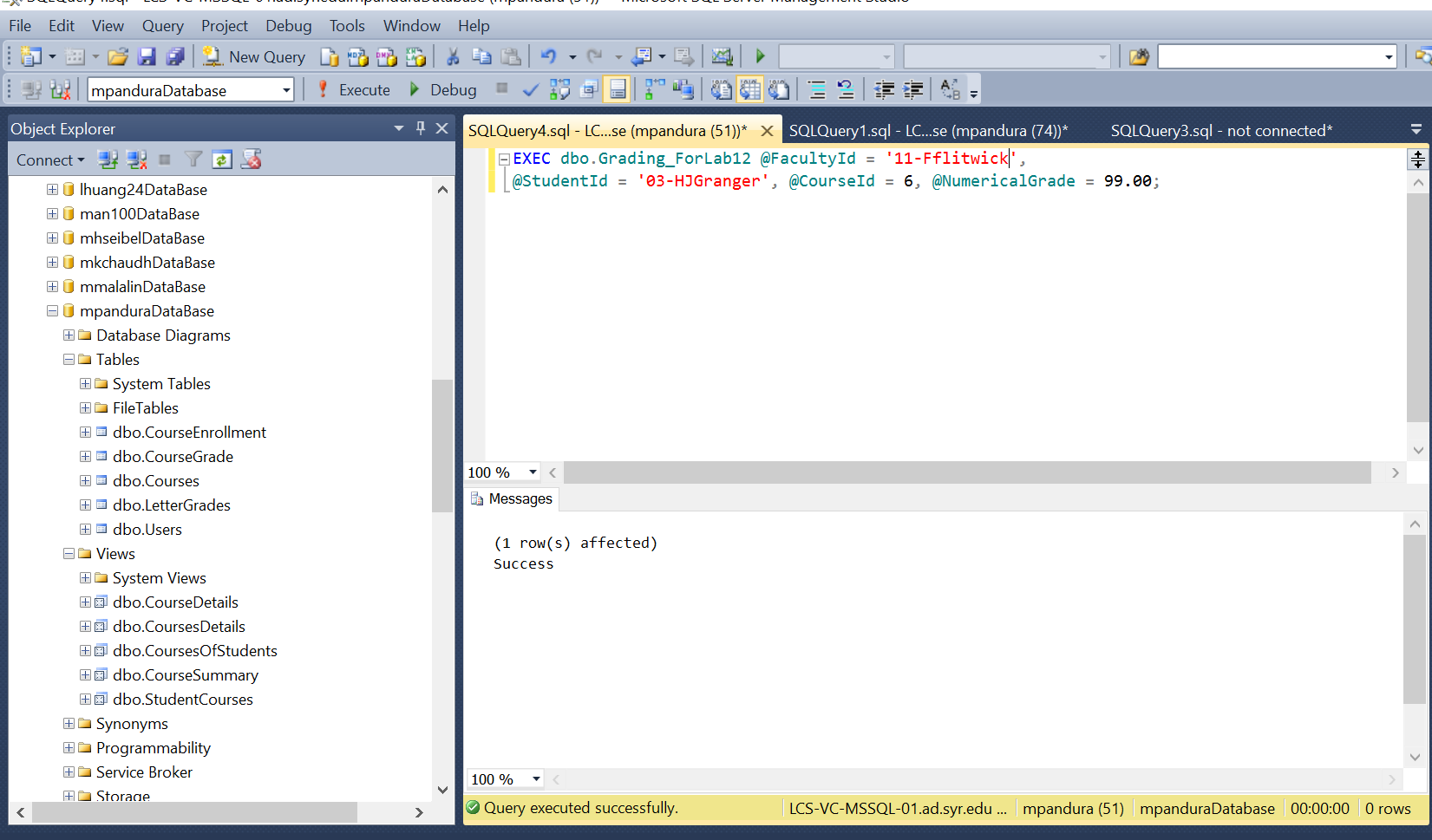
c. EXEC dbo.Grading\_ForLab12 @FacultyId = '11-Fflitwick',

@StudentId = '01-HJPotter', @CourseId = 4, @NumericalGrade = 99.00;



d. EXEC dbo.Grading\_ForLab12 @FacultyId = '11-Fflitwick',

@StudentId = '03-HJGranger', @CourseId = 6, @NumericalGrade = 99.00;



1. Create a SP that will return back 2 joined tables. I’ll leave which tables and which join type up to you.

CREATE PROCEDURE dbo.joining\_tables as

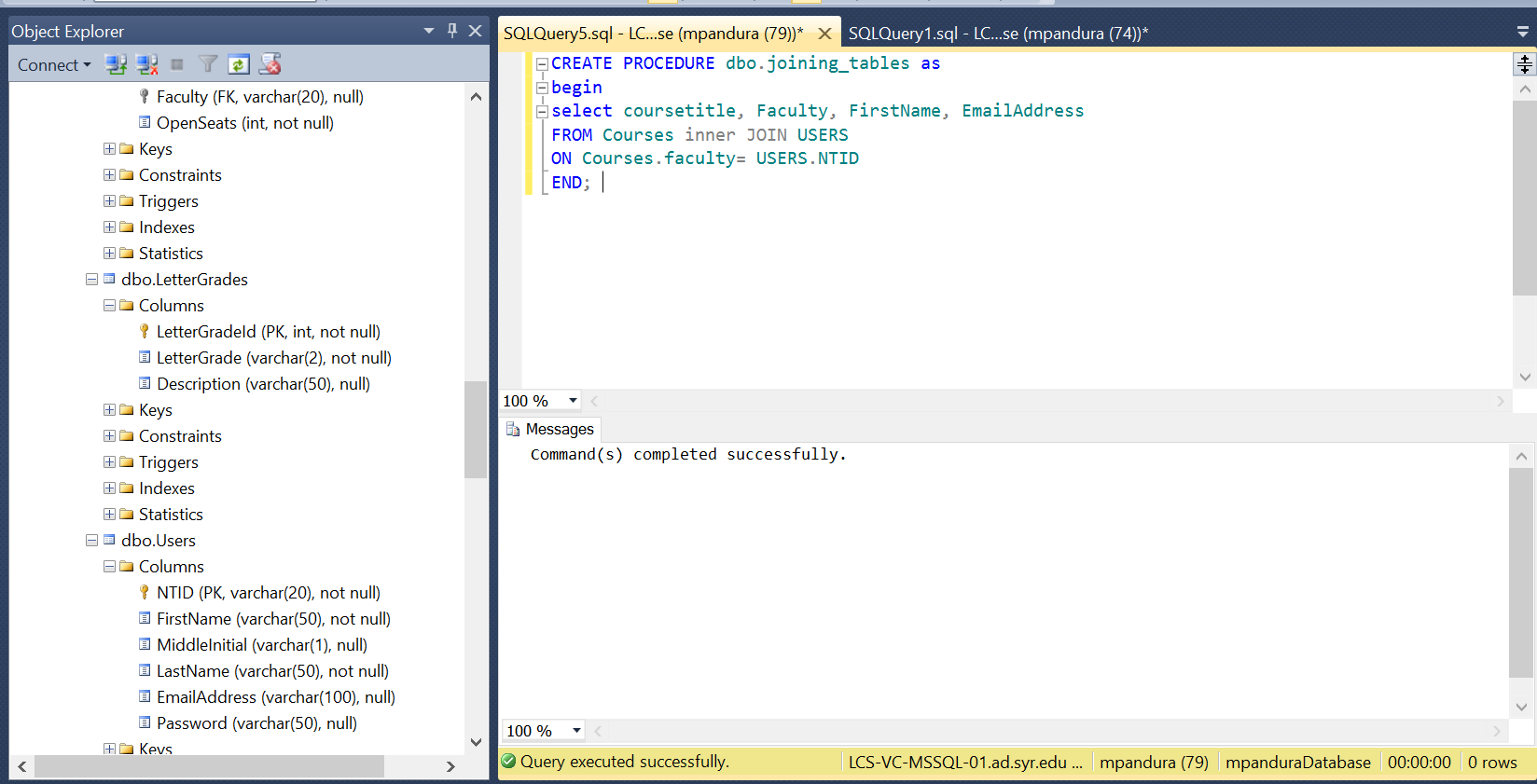
begin

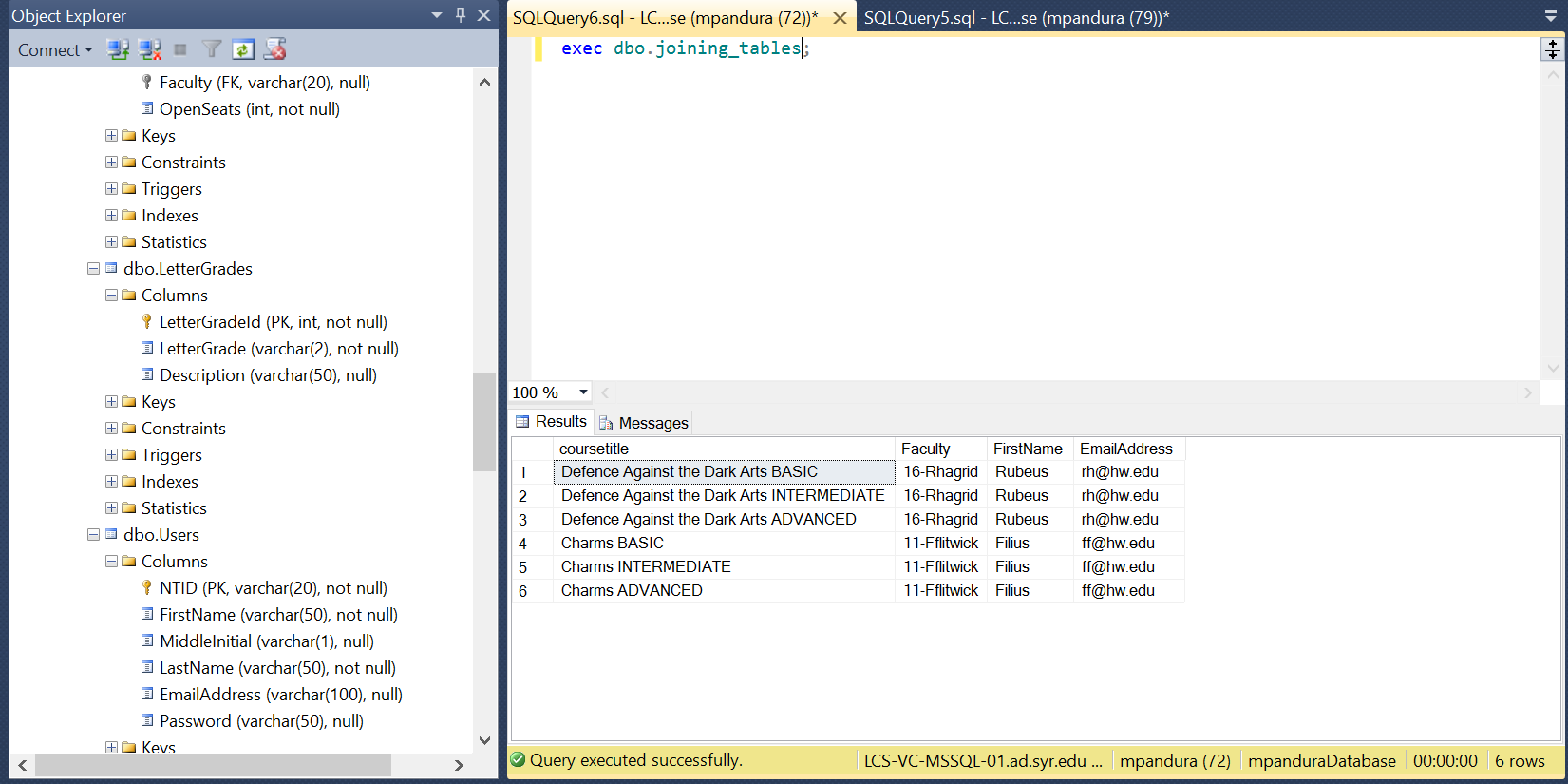
select coursetitle, Faculty, FirstName, EmailAddress

FROM Courses inner JOIN USERS

ON Courses.faculty= USERS.NTID

END;





1. You have to prove to me that it works exactly as expected. That means you need to prove that the right messages were printed, and that the data was or was not changed, depending on the case. The only way you can prove this to me is via screenshots. [↑](#footnote-ref-1)
2. If you need to modify the data in order to be able to execute all of the cases, feel free to do so. I do not need to see these, just make sure that the data prior to the SP’s execution is showing the valid input. [↑](#footnote-ref-2)