CSE 581

**Lab 11: Stored Procedures**

1. Create a stored procedure. Input for the stored procedure will be Course Id and Student Id.

The stored procedure will check if the student is enrolled in the class yet.

1. If the student has already been enrolled, display a message that says “The student is already enrolled”.
2. If the student has not been enrolled yet, and the course doesn’t have a faculty yet, display a message that says “Cannot enroll until faculty is selected”.
3. If the student has not been enrolled yet, and the course does have a faculty assigned, then enroll the student in the course and display a message that says “Student enrolled”.

Provide a **screenshot** of script execution that creates the stored procedure.

**Script:**

create procedure apmahaja.checkCourseEnrollment (@courseID as integer, @studentID as varchar(20), @message as varchar(256) out) as

declare @enrollmentID integer

select @enrollmentID = (select EnrollmentID from apmahaja.CourseEnrollment

where StudentId = @studentID and CourseId = @courseID);

if(@enrollmentID is not null)

begin

select @message = 'Student is already enrolled'

end

else

begin

declare @faculty varchar(20)

select @faculty = (select Faculty from apmahaja.Courses where CourseId = @courseID);

if (@faculty is null or @faculty = '')

begin

select @message = 'Cannot enroll until faculty is selected'

end

else

begin

declare @openSeats integer

select @openseats = (select Openseats from apmahaja.Courses where CourseId = @courseID);

if(@openSeats > 0)

begin

insert into apmahaja.CourseEnrollment (StudentID, CourseId) values (@studentID, @courseID);

select @message = 'Student Enrolled'

select @openSeats = @openSeats -1

update apmahaja.Courses set OpenSeats = @openSeats where CourseId = @courseID;

end

else

begin

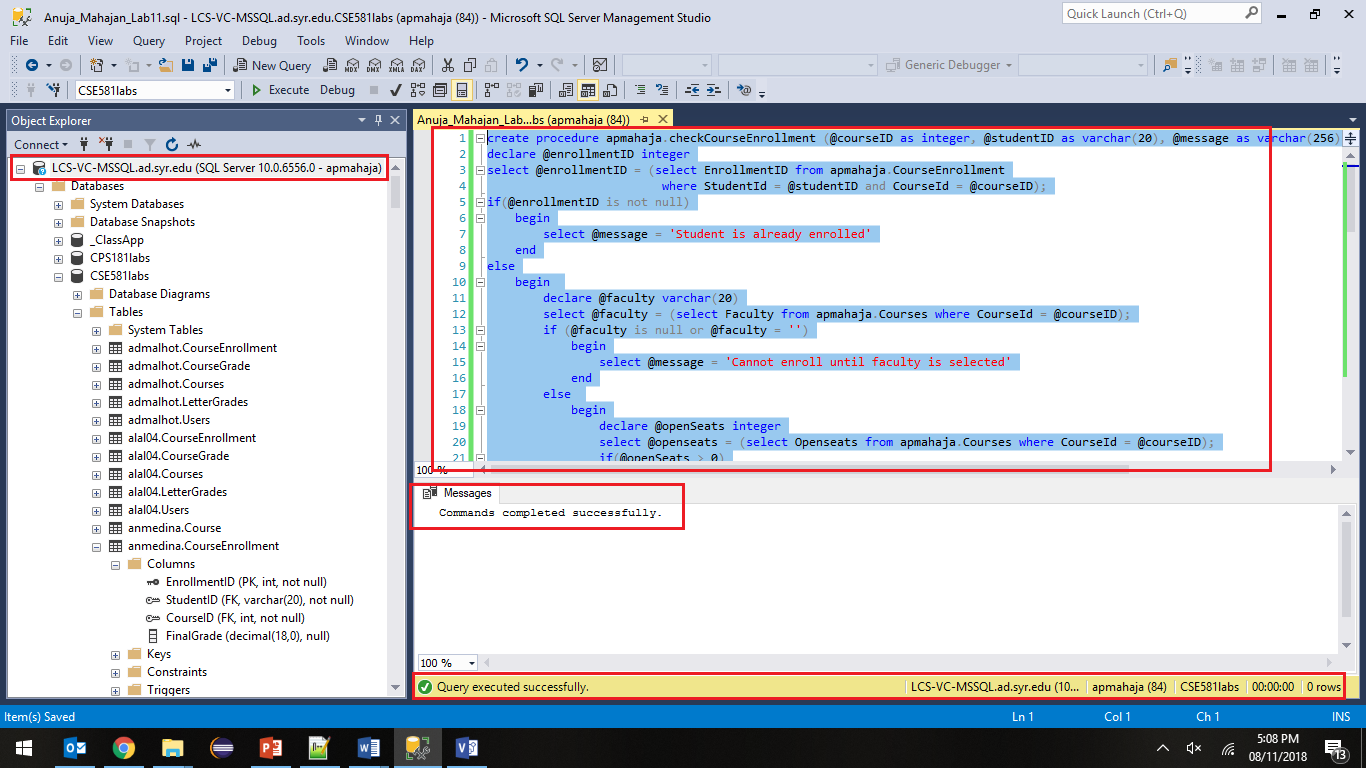
select @message = 'Student cannot enrol as seats are not available'

end

end

end

**Execution:**

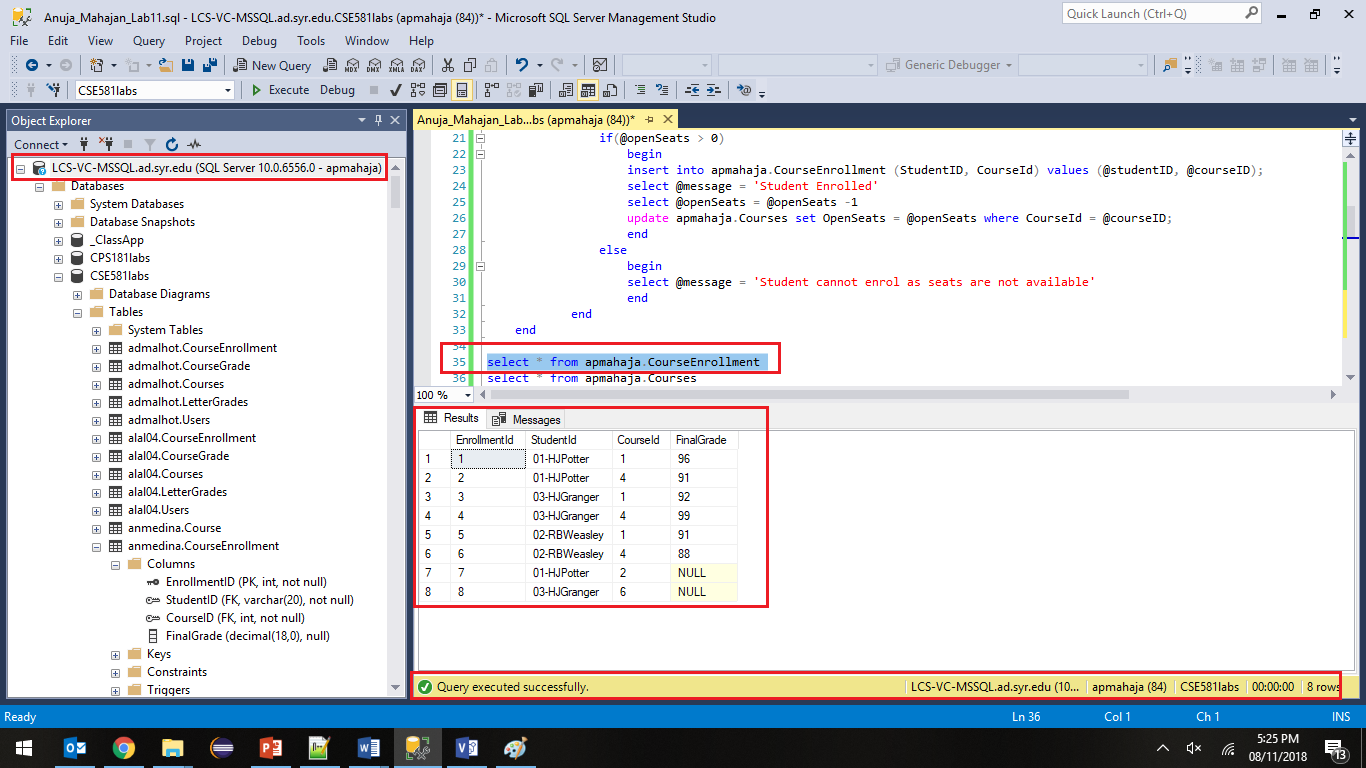


1. Select from the Enrollment table, to see data prior to execution. Provide a **screenshot**.

**Script:**

select \* from apmahaja.CourseEnrollment

**Execution:**



1. Run the stored procedure to prove that all 3 cases work. For each case, provide a **screenshot** of script execution and **screenshots** of Enrollment and Courses table after execution.

**Case 1:** Student is already enrolled in course

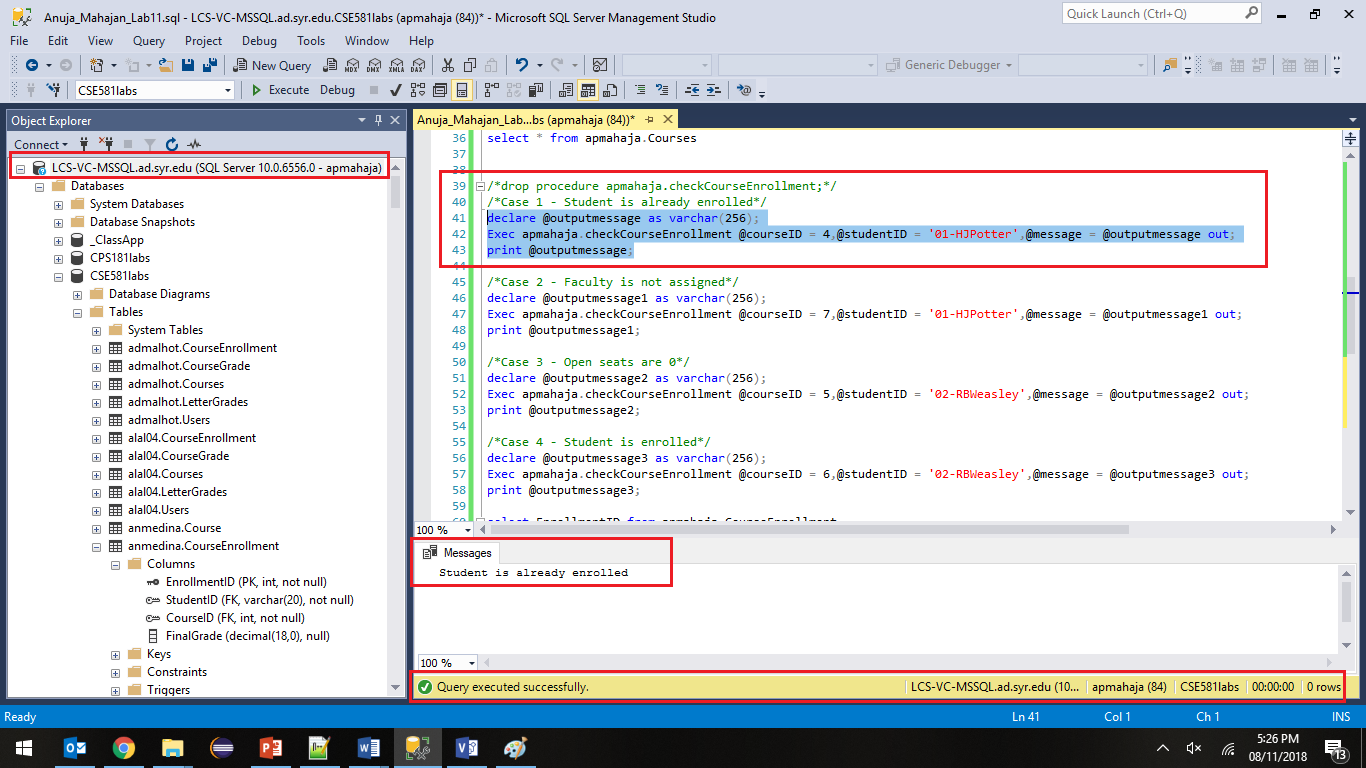
**Script:**

declare @outputmessage as varchar(256);

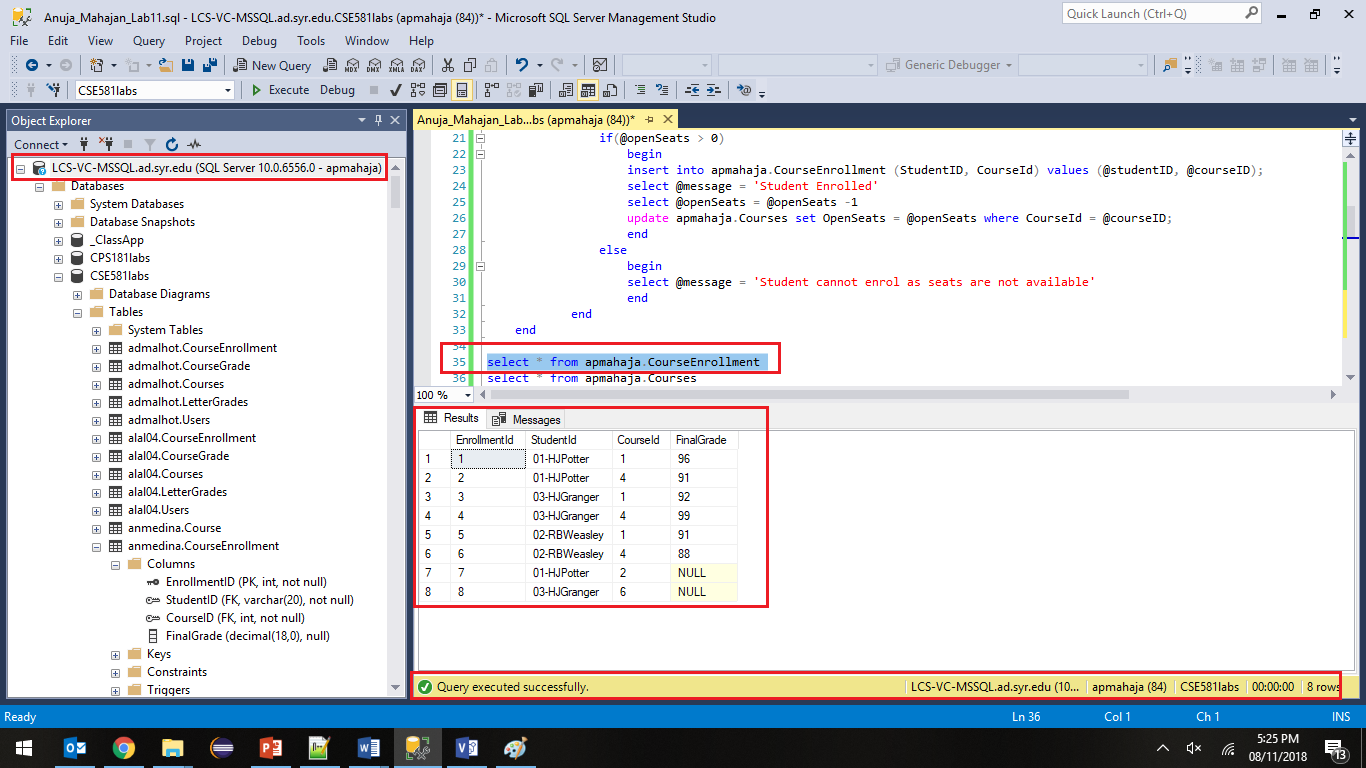
Exec apmahaja.checkCourseEnrollment @courseID = 4,@studentID = '01-HJPotter',@message = @outputmessage out;

print @outputmessage;

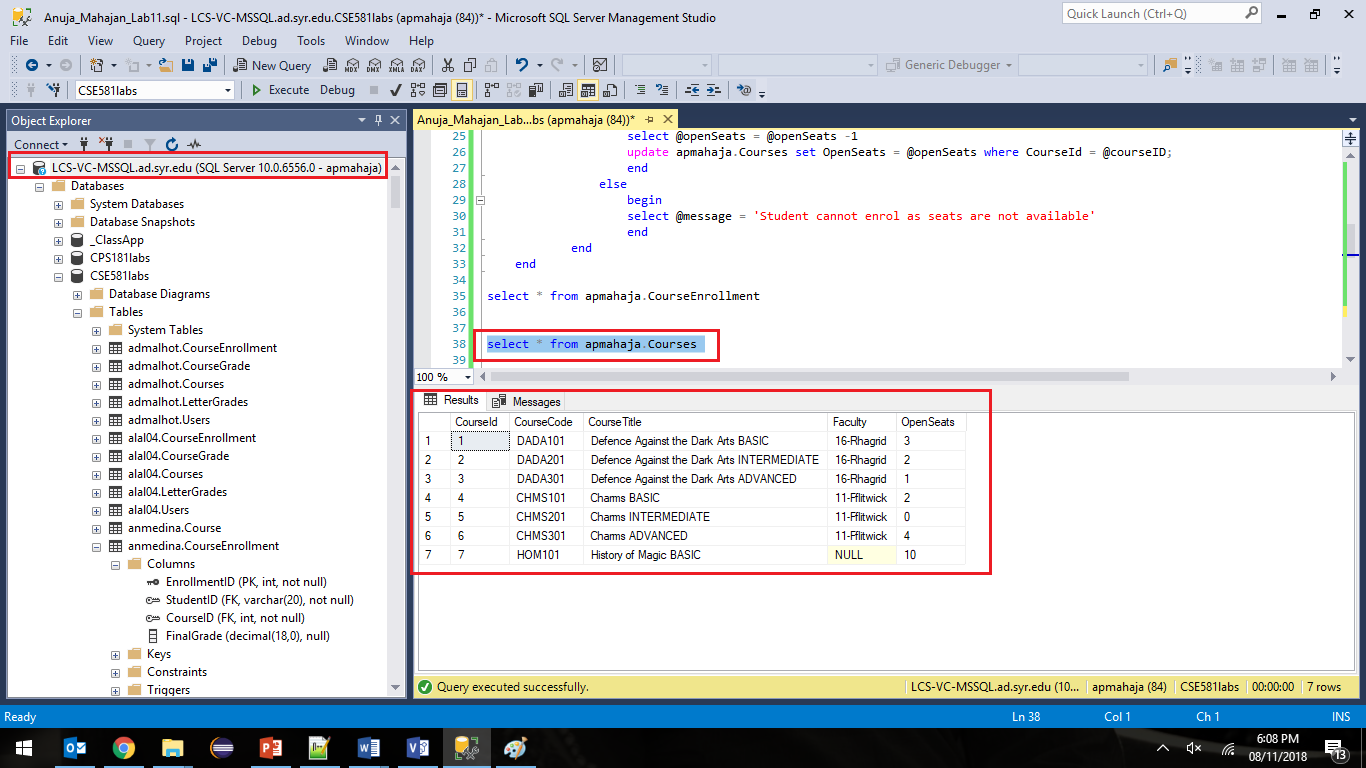
**Execution:**



**Course Enrollment Table:**



**Course Table:**



**Case 2: Student has not been enrolled, and faculty has not been assigned to Course**

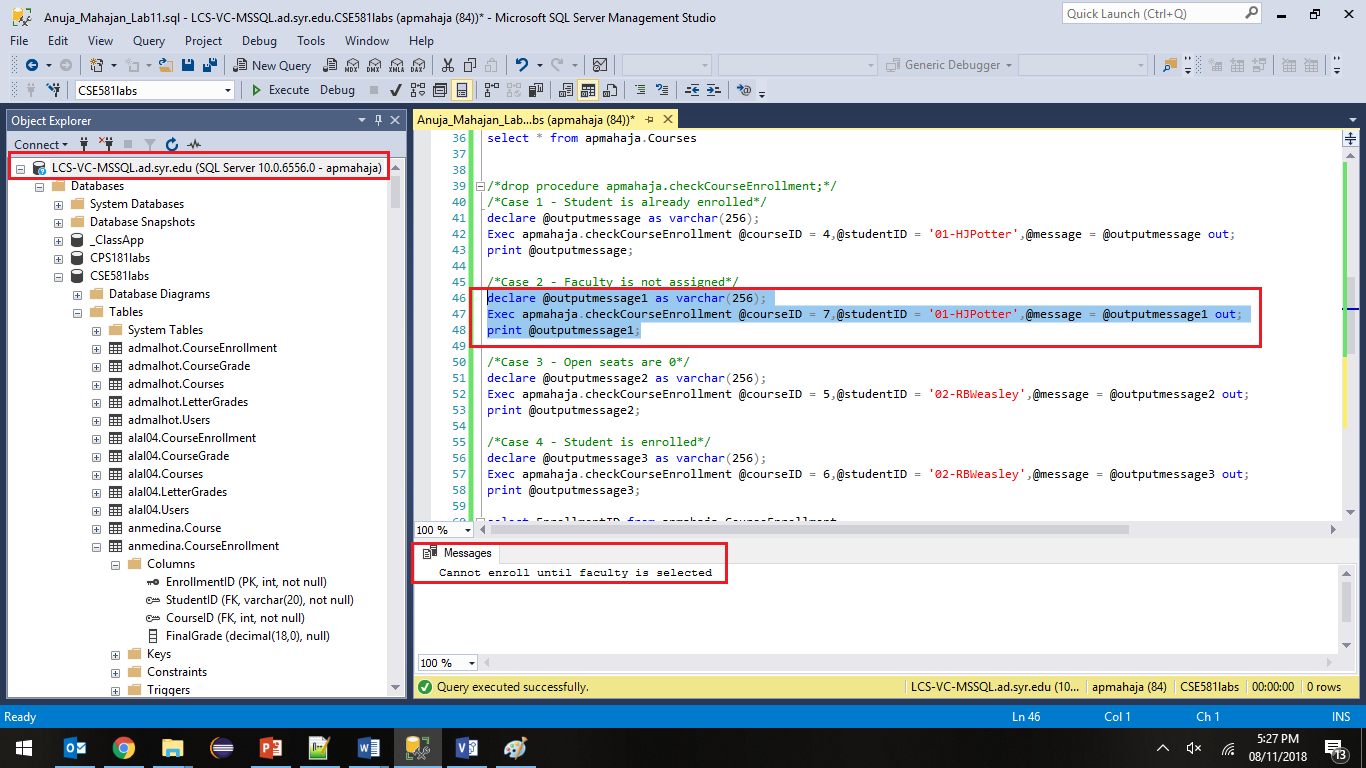
**Script:**

declare @outputmessage1 as varchar(256);

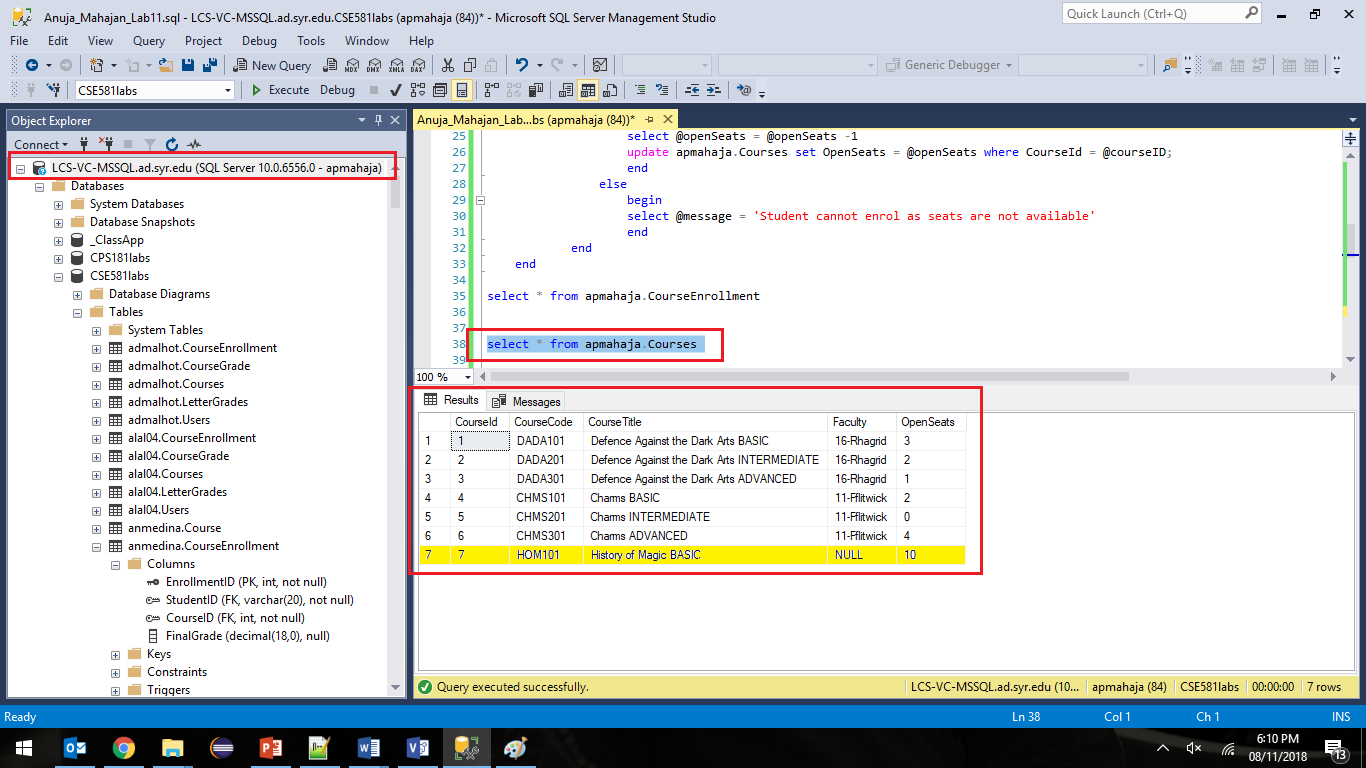
Exec apmahaja.checkCourseEnrollment @courseID = 7,@studentID = '01-HJPotter',@message = @outputmessage1 out;

print @outputmessage1;

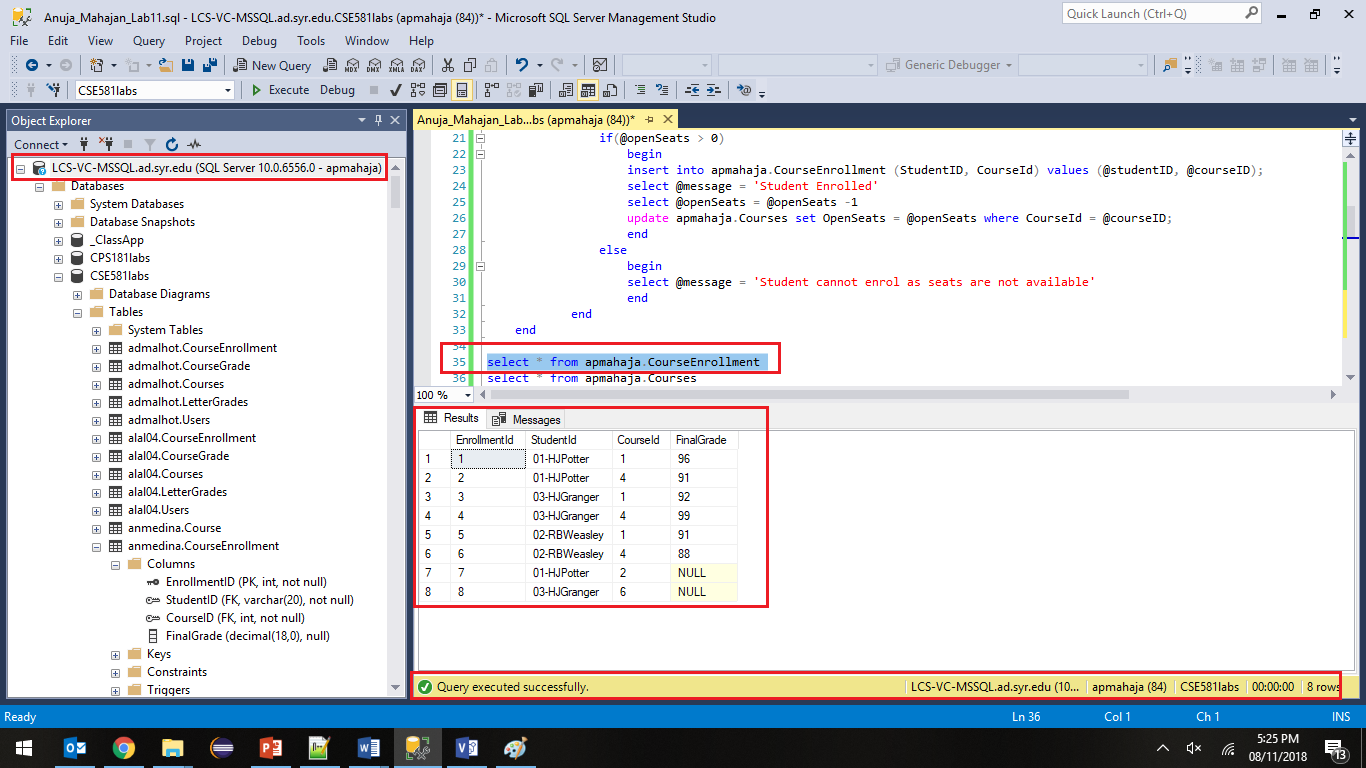
**Execution:**



**Course Table:**



**Enrollment Table:**



**Case 3: Student has not been enrolled, Faculty is assigned but Open seats are 0**

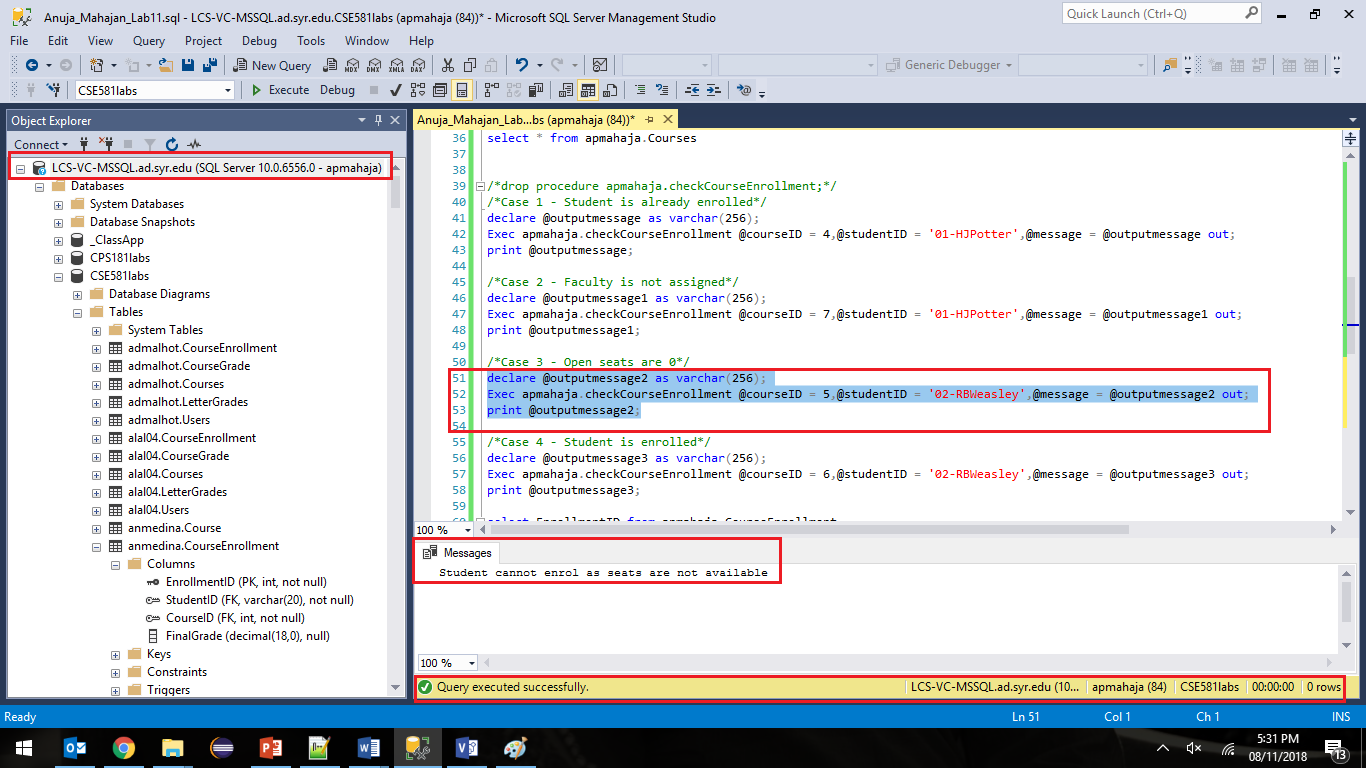
**Script:**

declare @outputmessage2 as varchar(256);

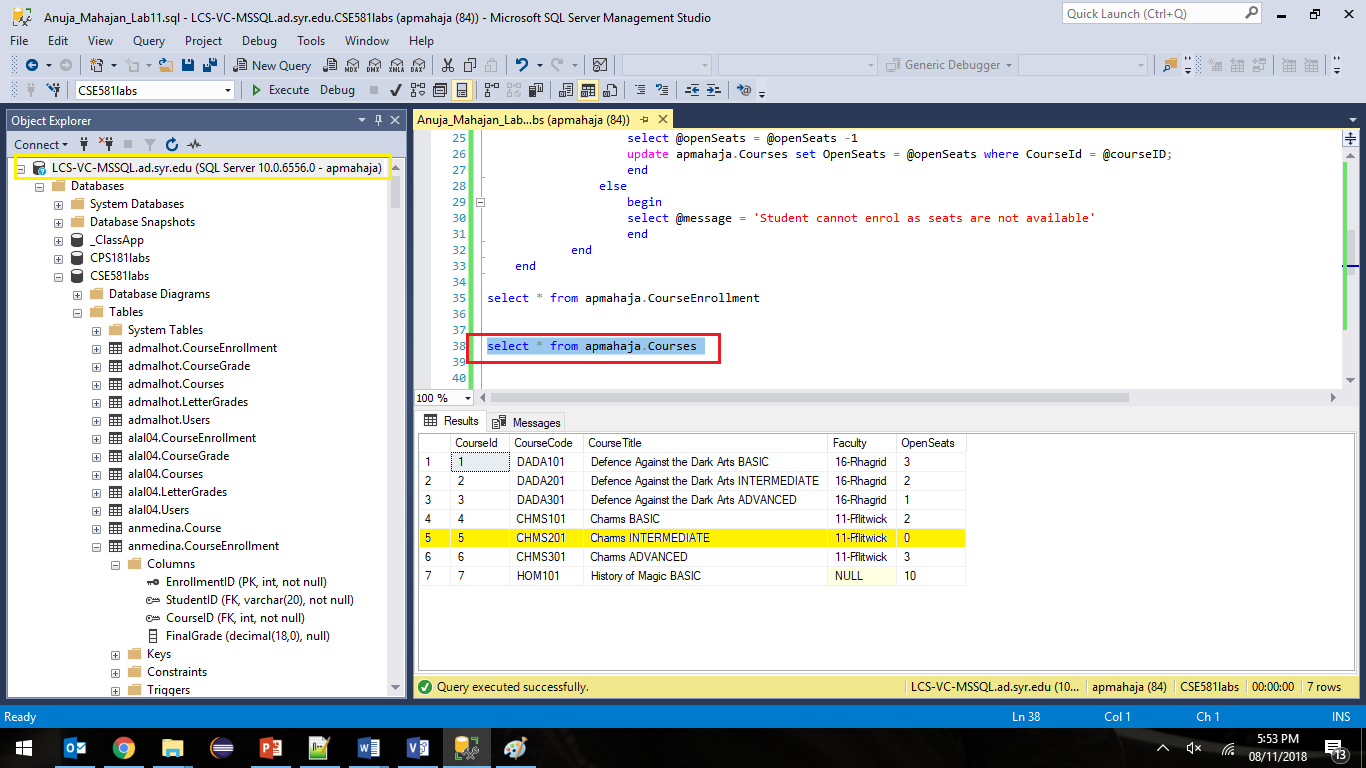
Exec apmahaja.checkCourseEnrollment @courseID = 5,@studentID = '02-RBWeasley',@message = @outputmessage2 out;

print @outputmessage2;

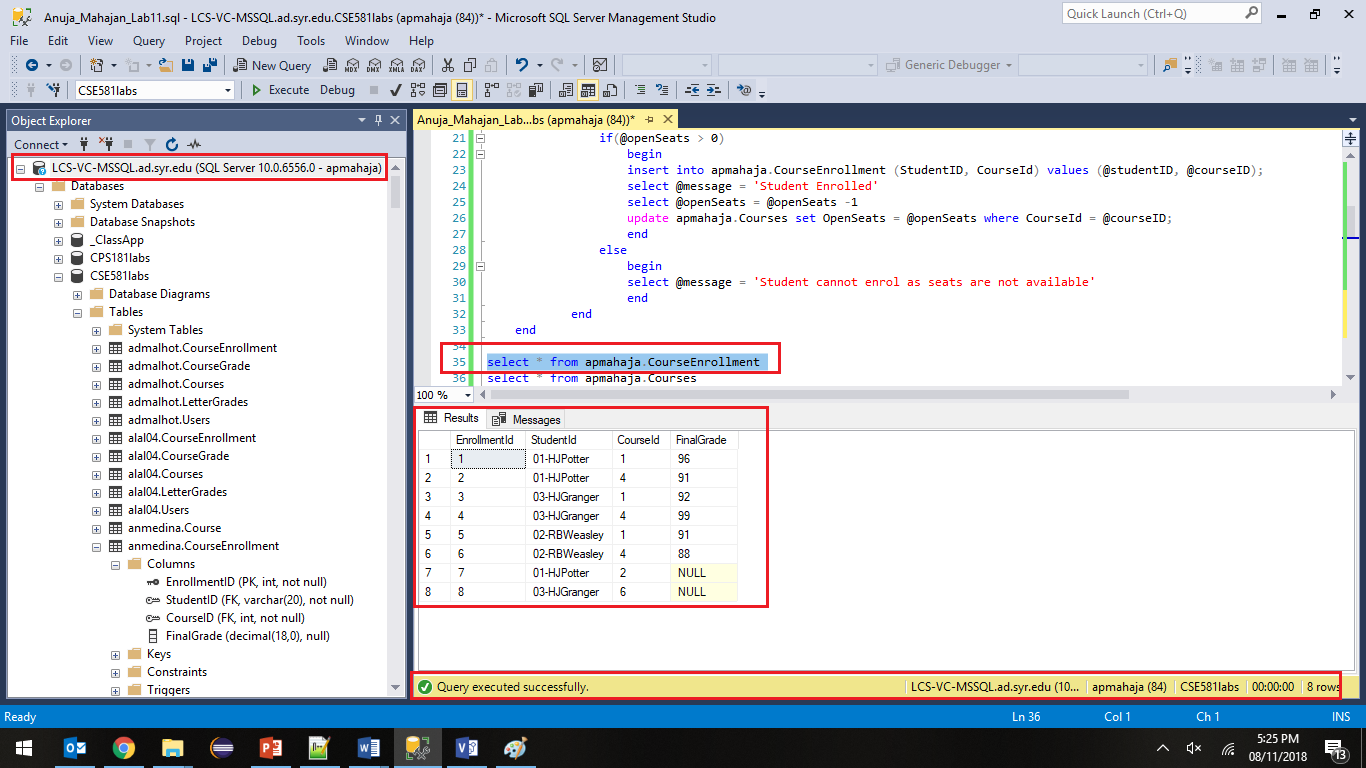
**Execution:**



**Course Table:**



**Enrollment Table:**



**Case 4: Student has not been enrolled and faculty has been assigned, open seats are available**

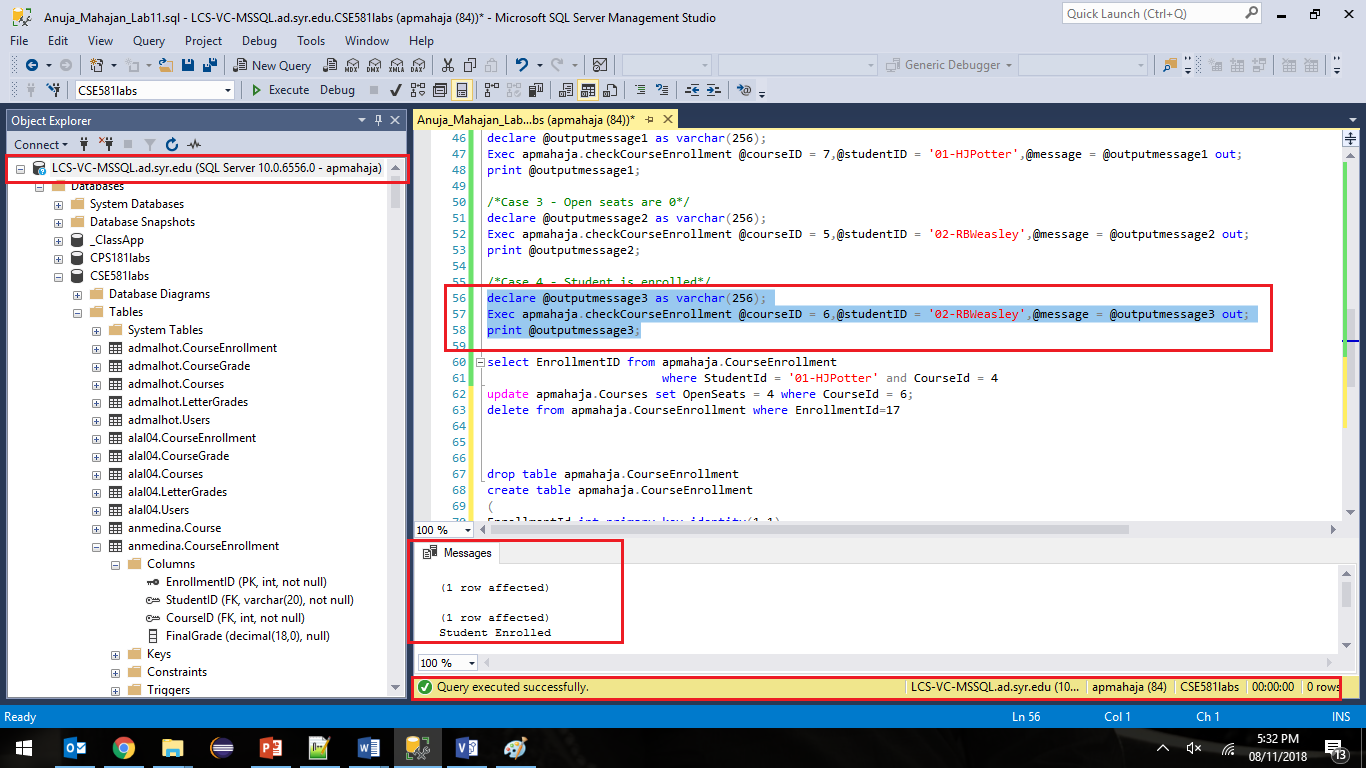
**Script:**

declare @outputmessage3 as varchar(256);

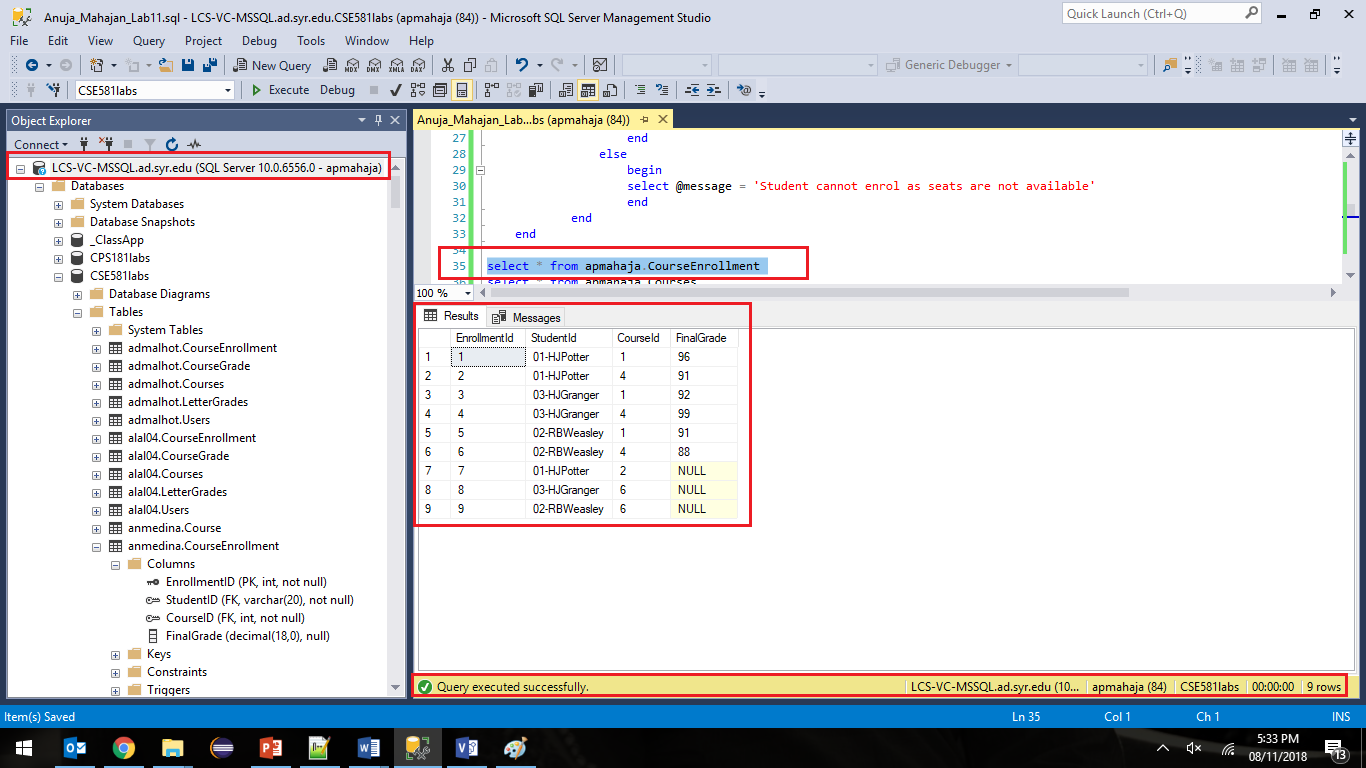
Exec apmahaja.checkCourseEnrollment @courseID = 6,@studentID = '02-RBWeasley',@message = @outputmessage3 out;

print @outputmessage3;

**Execution:**



**Course Enrollment Table:**



**Course Table:**

