Tatiana Ensslin

CSE 581

**Lab 15: Triggers**

**Steps:**

1. Create a trigger[[1]](#footnote-1) on insert/delete/update of the Enrollment table:
   1. when inserting a new record, update the Courses table’s OpenSeats for the course into which the student is being enrolled.

Create Trigger InsertEnrollment

ON CourseEnrollment AFTER INSERT AS

Declare @InsertOpenSeat INT

Declare @InsertCourseID INT

IF EXISTS (SELECT \* FROM INSERTED I

INNER JOIN Courses C

ON I.CourseID = C.CourseID)

BEGIN

SELECT @InsertCourseID = (SELECT CourseID

FROM INSERTED)

SELECT @InsertOpenSeat = (SELECT OpenSeats

FROM Courses

WHERE @InsertCourseID = CourseID)

SELECT @InsertOpenSeat = @InsertOpenSeat - 1

UPDATE Courses

SET OpenSeats = @InsertOpenSeat

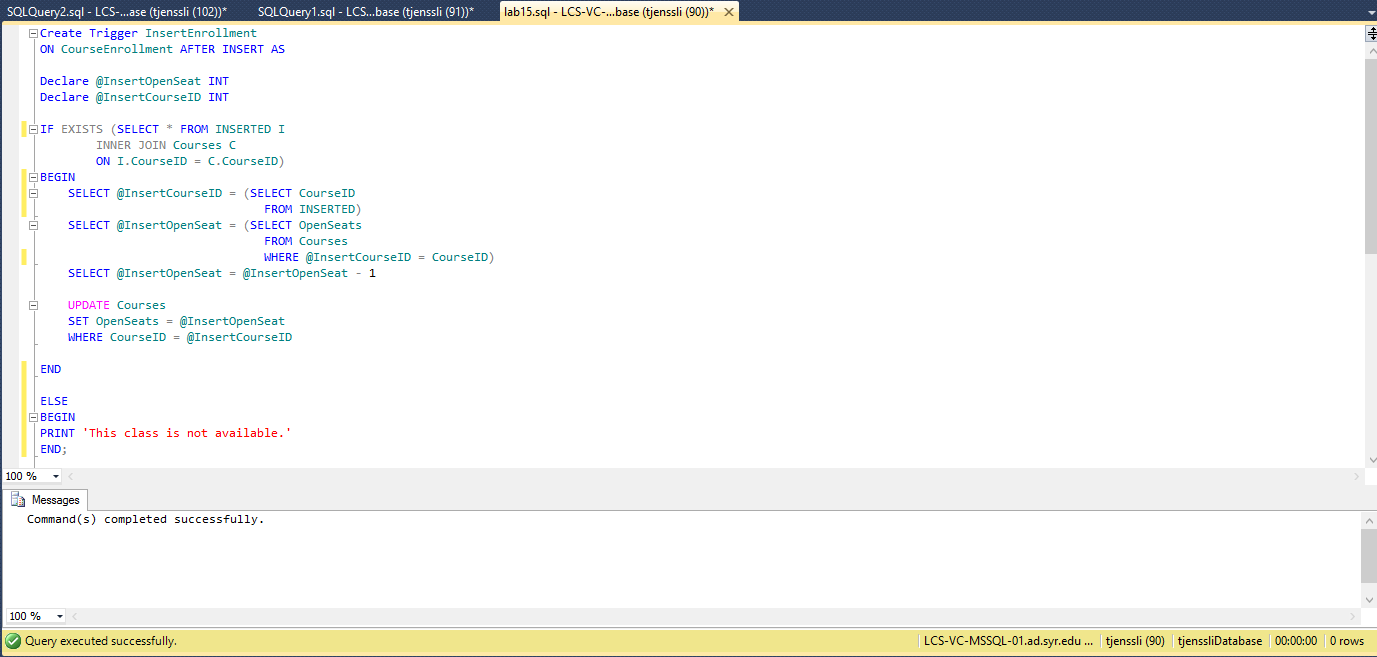
WHERE CourseID = @InsertCourseID

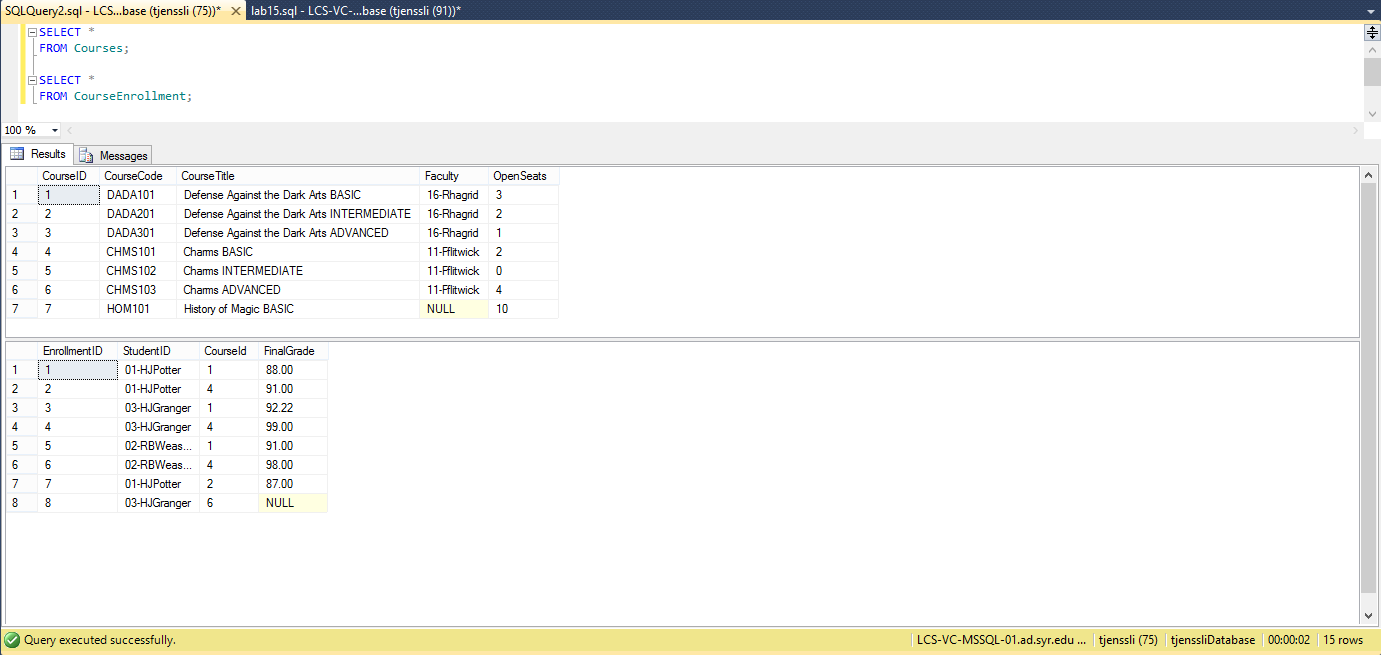
END

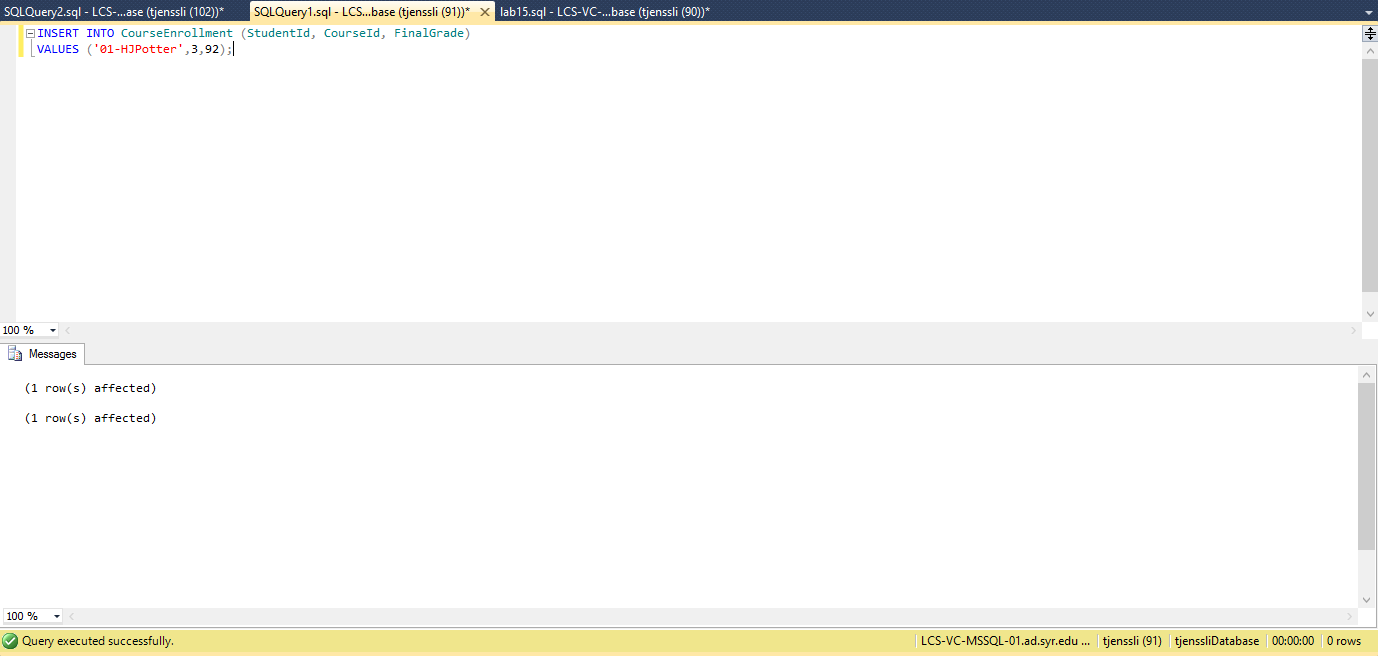
ELSE

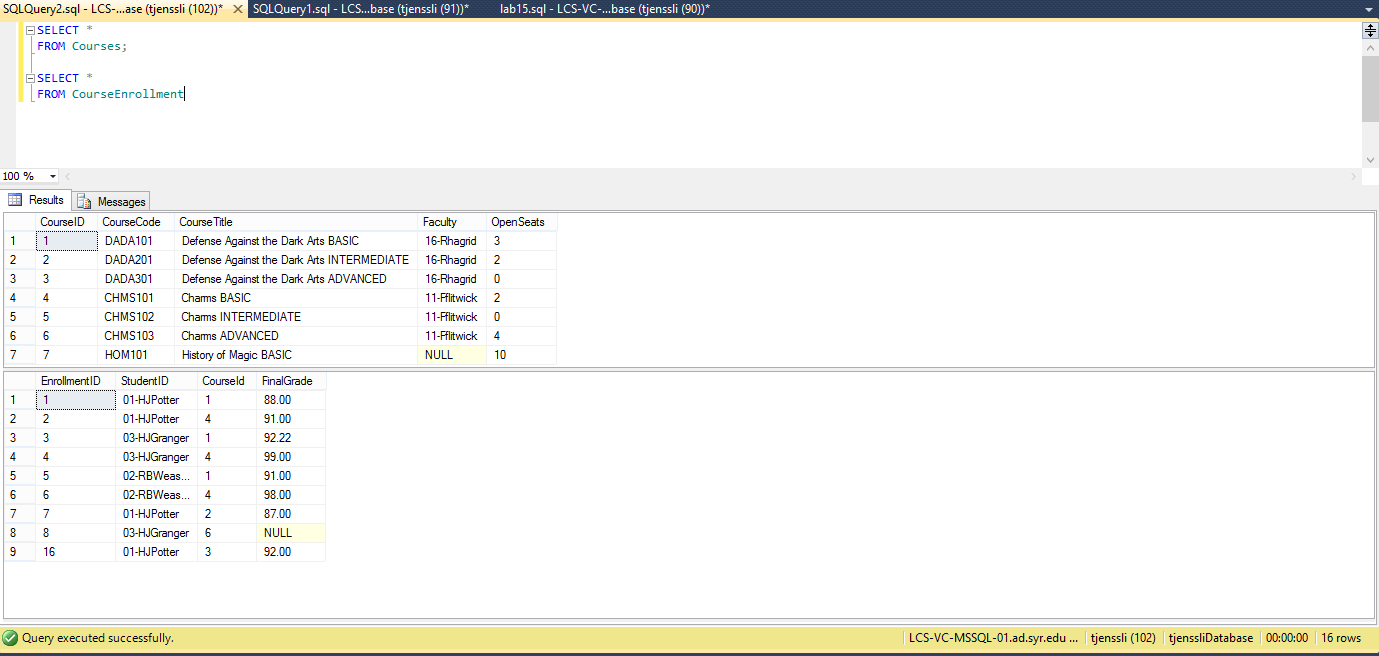
BEGIN

PRINT 'This class is not available.'

END;







b. when deleting a record, update the Courses table’s OpenSeats for the course from

which the student was deleted.

Create Trigger DeleteEnrollment

ON CourseEnrollment AFTER DELETE AS

Declare @DeletedOpenSeat INT

Declare @DeletedCourseID INT

BEGIN

SELECT @DeletedCourseID = (SELECT CourseID

FROM DELETED)

SELECT @DeletedOpenSeat = (SELECT OpenSeats

FROM Courses

WHERE CourseID = @DeletedCourseID)

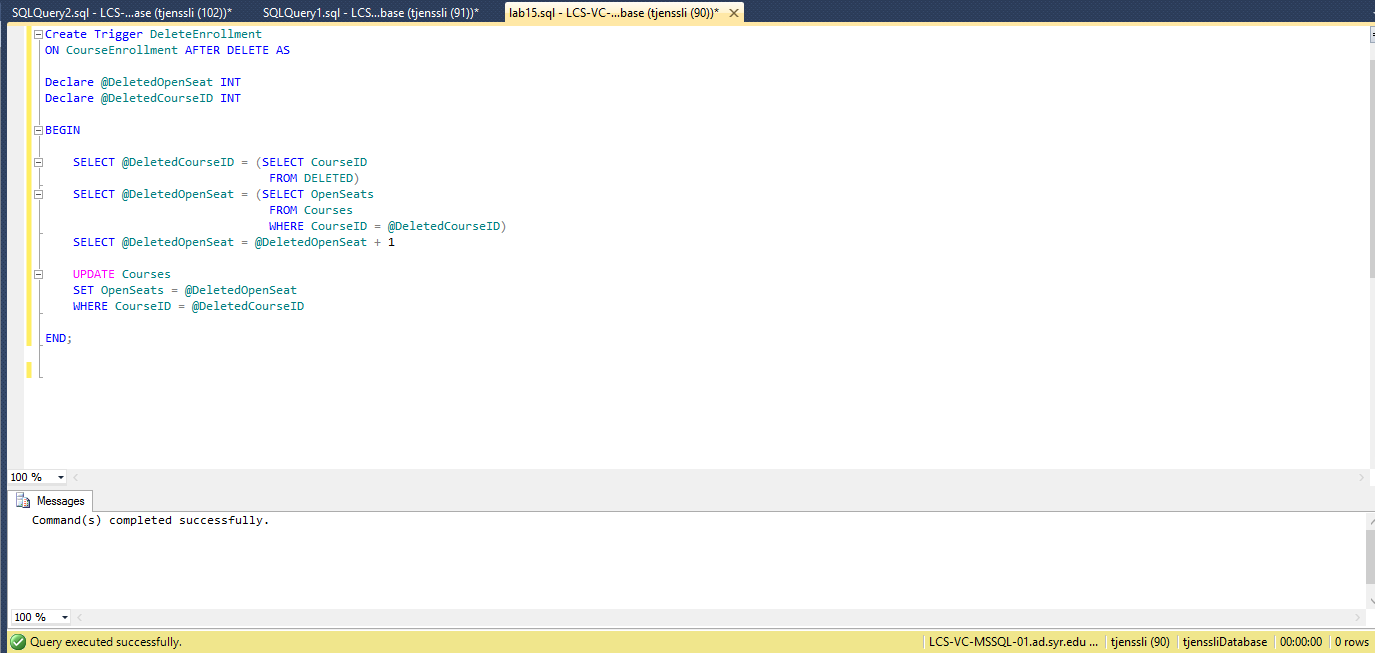
SELECT @DeletedOpenSeat = @DeletedOpenSeat + 1

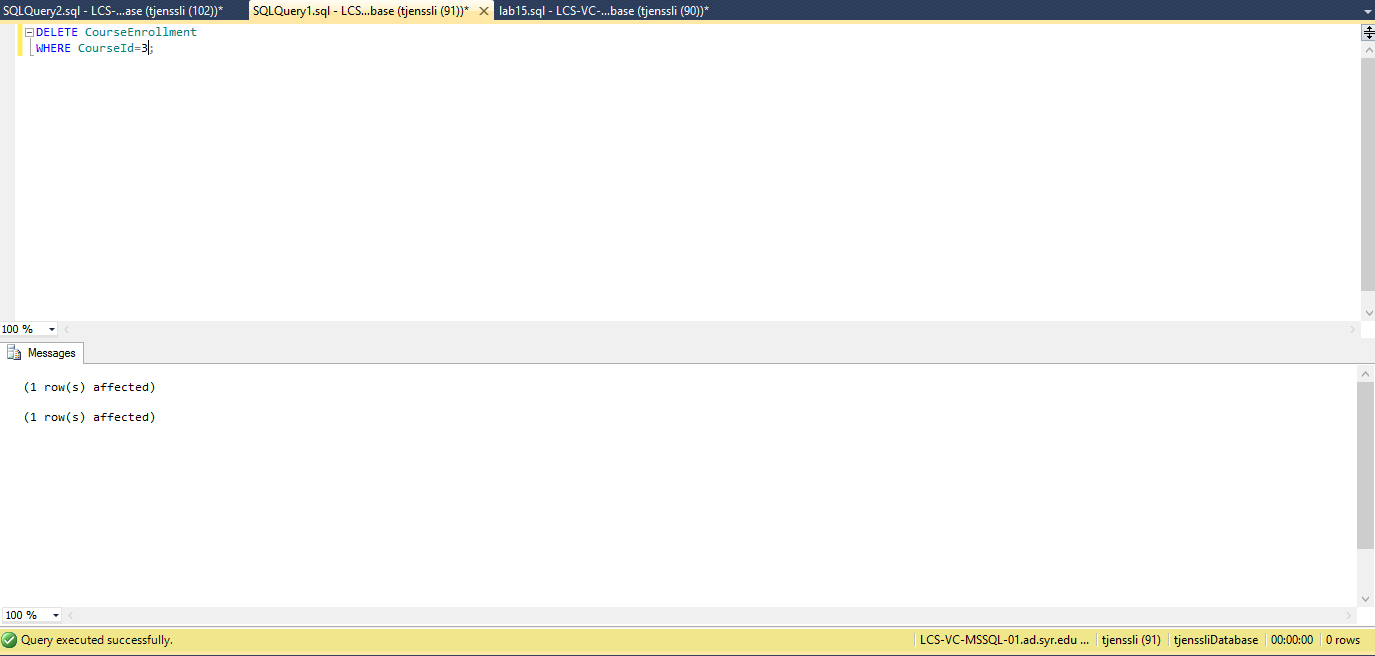
UPDATE Courses

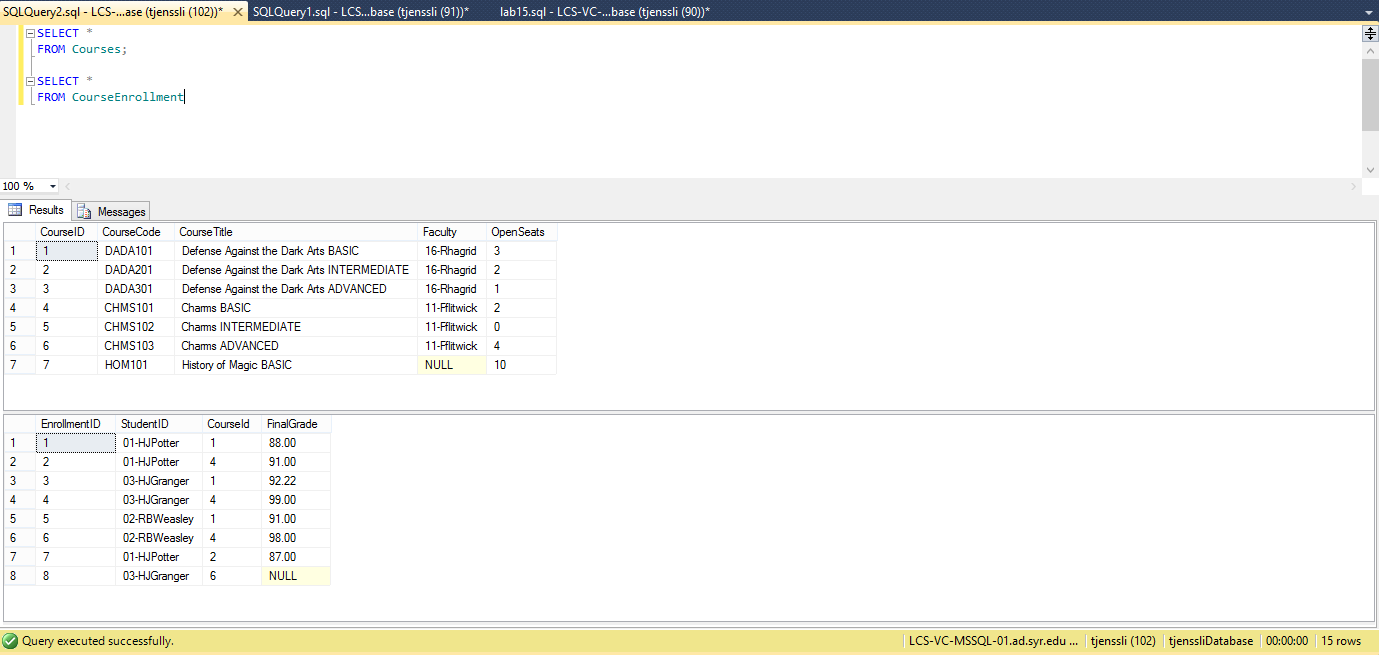
SET OpenSeats = @DeletedOpenSeat

WHERE CourseID = @DeletedCourseID

END;







C. if an update happened on the CourseId (student switched courses), adjust the Courses

table’s OpenSeats accordingly.

Create Trigger UpdateEnrollment

ON CourseEnrollment AFTER UPDATE AS

DECLARE @InsertCourseID INT

DECLARE @InsertOpenSeat INT

DECLARE @DeletedCourseID INT

DECLARE @DeletedOpenSeat INT

IF EXISTS (SELECT \*

FROM INSERTED I

INNER JOIN Courses C

ON C.CourseID = I.CourseID)

IF EXISTS (SELECT \*

FROM DELETED D

INNER JOIN Courses C

ON D.CourseID = C.CourseID)

BEGIN

SELECT @InsertCourseID = (SELECT CourseID

FROM INSERTED)

SELECT @InsertOpenSeat = (SELECT OpenSeats

FROM Courses

WHERE CourseID = @InsertCourseID)

SELECT @InsertOpenSeat = @InsertOpenSeat - 1

UPDATE Courses

SET OpenSeats = @InsertOpenSeat

WHERE CourseID = @InsertCourseId

SELECT @DeletedCourseID = (SELECT CourseID

FROM DELETED)

SELECT @DeletedOpenSeat = (SELECT OpenSeats

FROM Courses

WHERE CourseID = @DeletedCourseID)

SELECT @DeletedOpenSeat = @DeletedOpenSeat + 1

UPDATE Courses

SET OpenSeats = @DeletedOpenSeat

WHERE CourseID = @DeletedCourseID

END

ELSE

BEGIN

SELECT @InsertCourseID = (SELECT CourseID

FROM INSERTED)

SELECT @InsertOpenSeat = (SELECT OpenSeats

FROM Courses

WHERE CourseID = @InsertCourseID)

SELECT @InsertOpenSeat= @InsertOpenSeat - 1

UPDATE Courses

SET OpenSeats = @InsertOpenSeat

END

ELSE

BEGIN

SELECT @DeletedCourseID = (SELECT CourseId

FROM DELETED)

SELECT @DeletedOpenSeat = (SELECT OpenSeats

FROM Courses

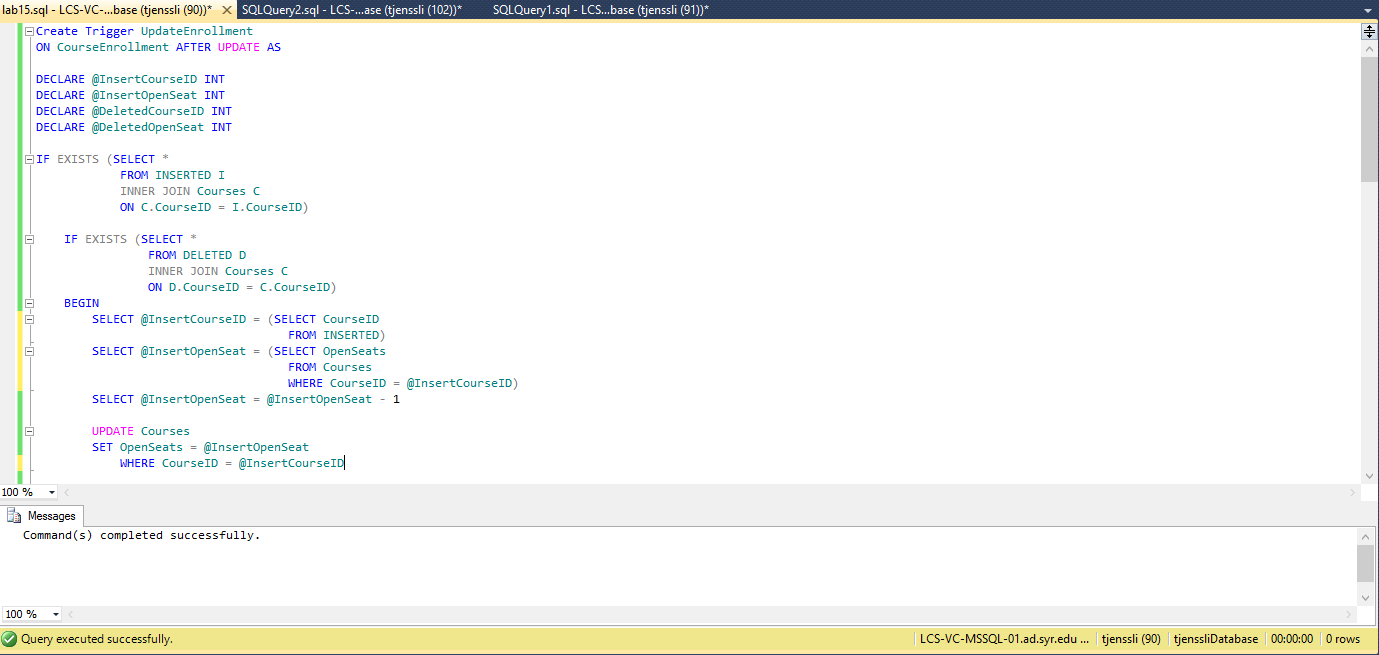
WHERE CourseId = @DeletedCourseID)

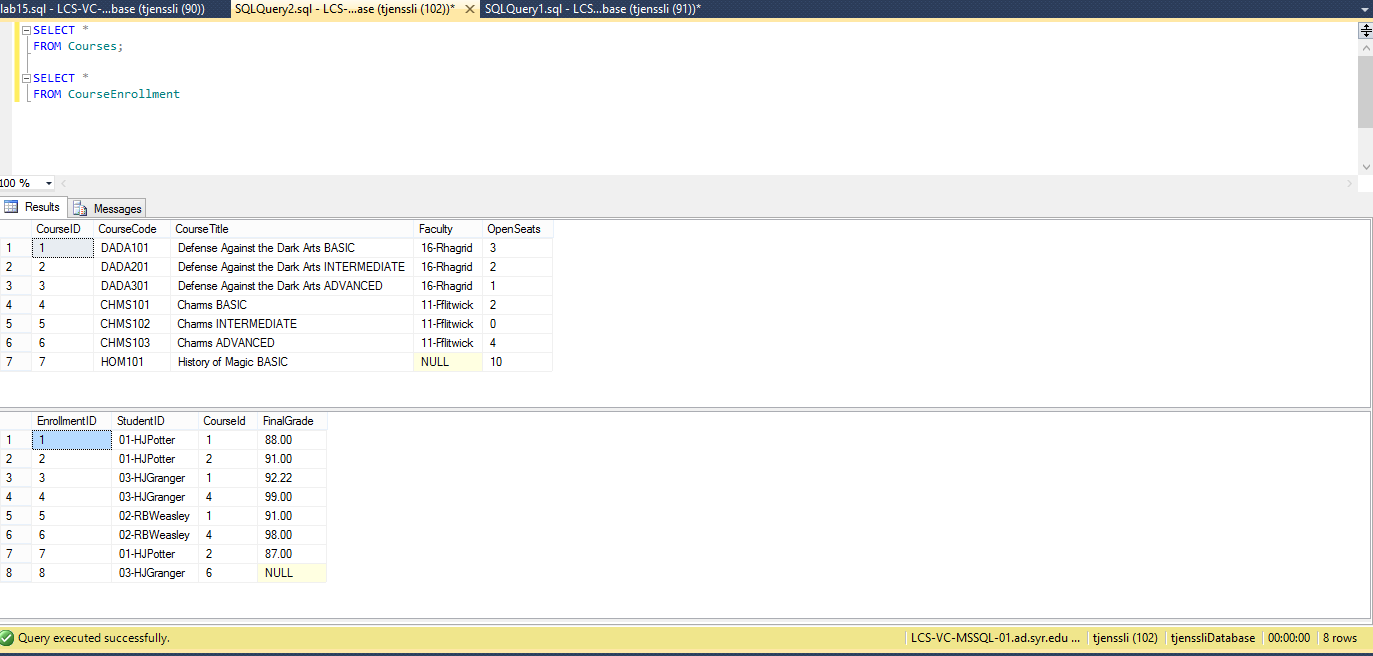
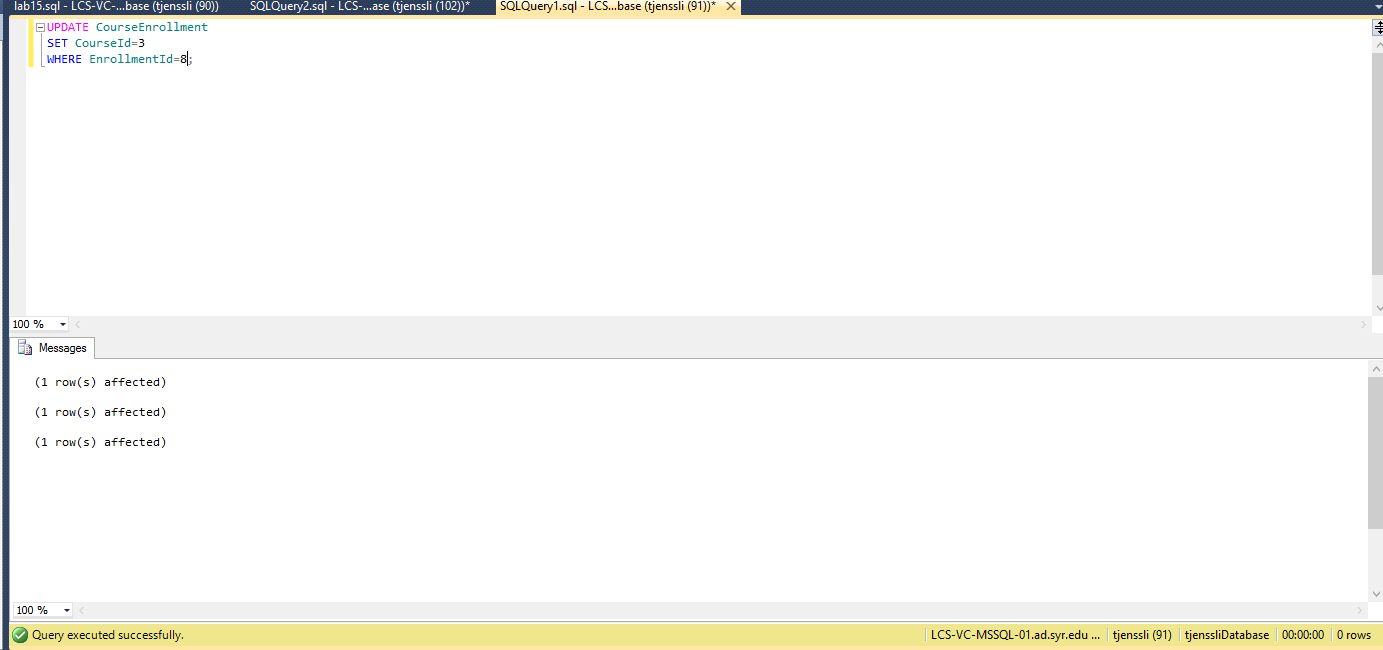
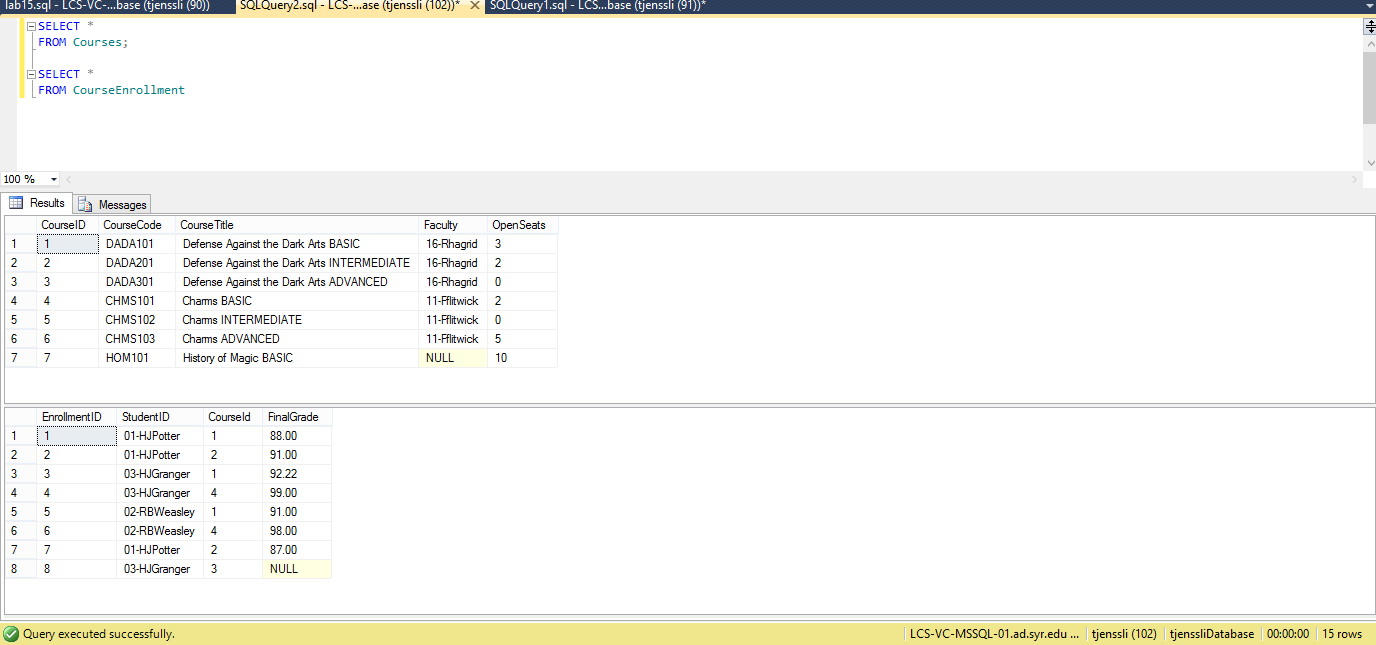
SELECT @DeletedOpenSeat = @DeletedOpenSeat + 1

UPDATE Courses

SET OpenSeats = @DeletedOpenSeat

WHERE CourseID = @DeletedCourseID

END;



1. You can create 3 triggers, one per action. You can also create a single trigger that can do all 3 – this is more difficult to do – you need to think about how you can decide which action is happening based on the data you would see during the trigger execution. Doing it via a single trigger would give you 5% bonus. [↑](#footnote-ref-1)