Venkata Satya Sri Ram Giduthuri SUID: 972124521

CSE 581

Lab 11: Stored Procedures

Purpose:

Create a stored procedure **in database CSE581labs**. Execute it.

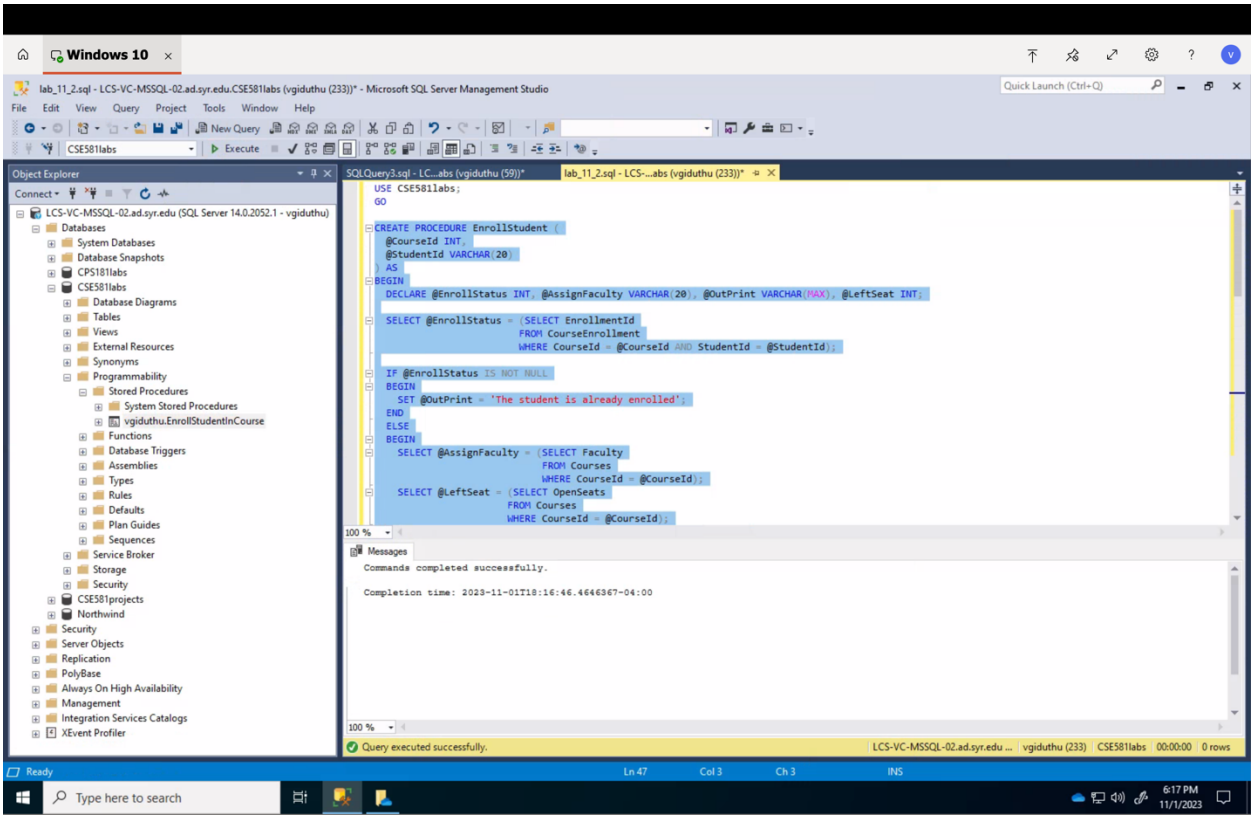
Deliverables:

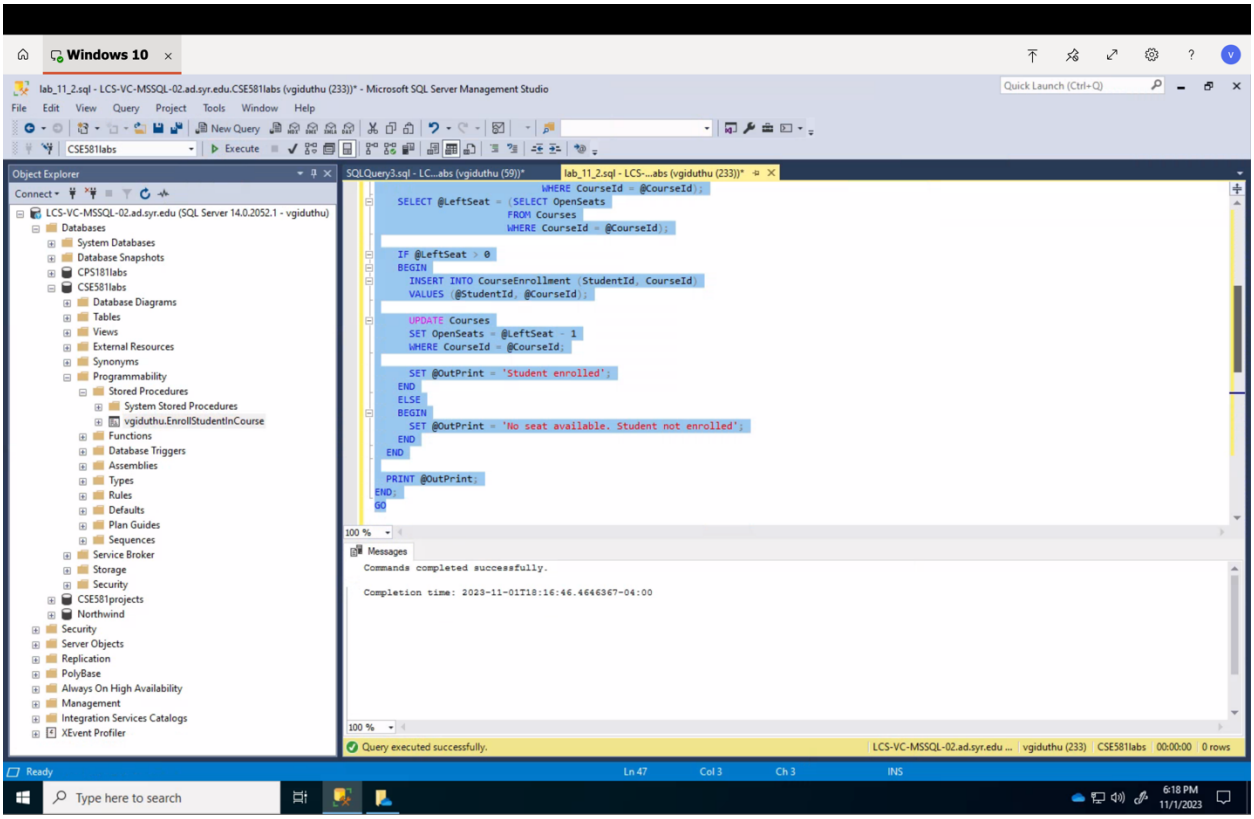
Multiple screenshots, as described through the lab.
Scripts that you used to carry out the actions.

Steps:

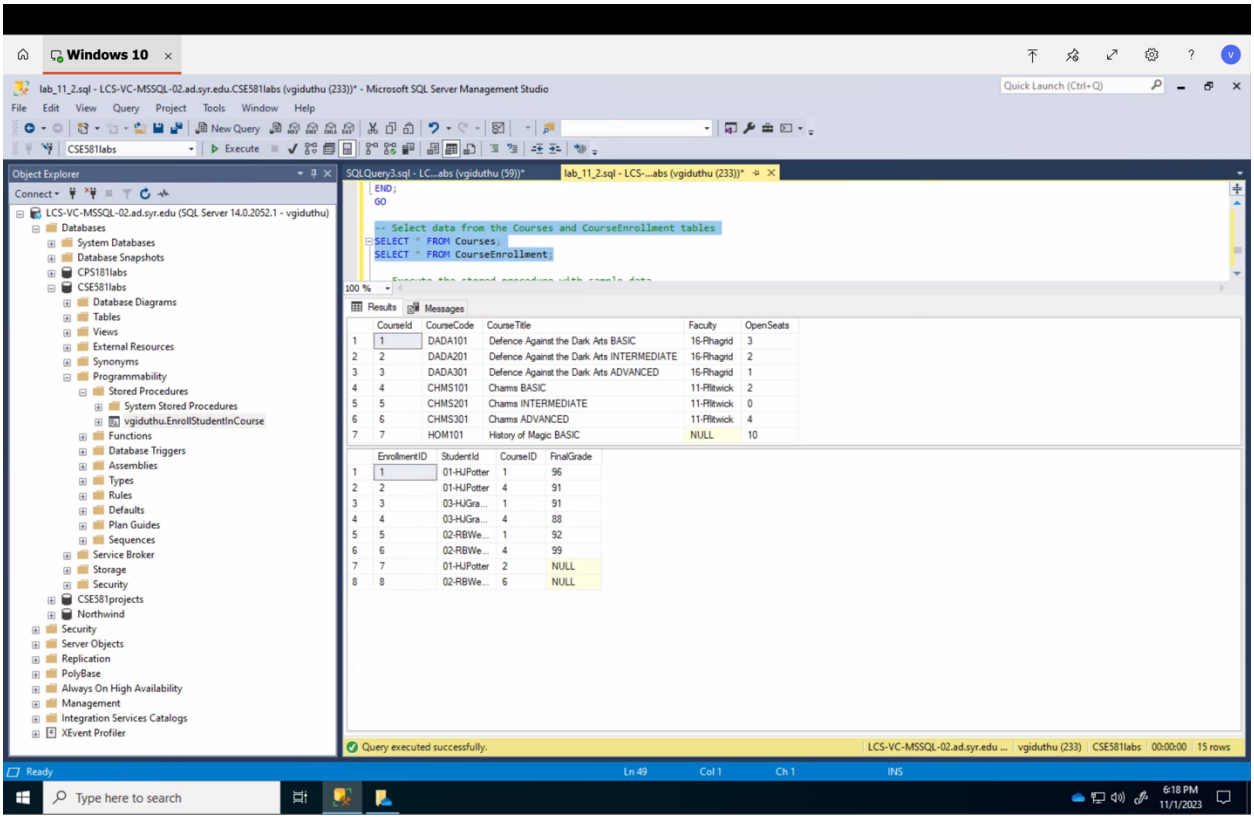
1. Create a stored procedure. Input for the stored procedure will be Course Id and Student Id. The stored procedure will check if the student is enrolled in the class yet.
    a) If the student has already been enrolled, display a message that says “The student is already enrolled”.
    b) If the student 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”.
    c) 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.

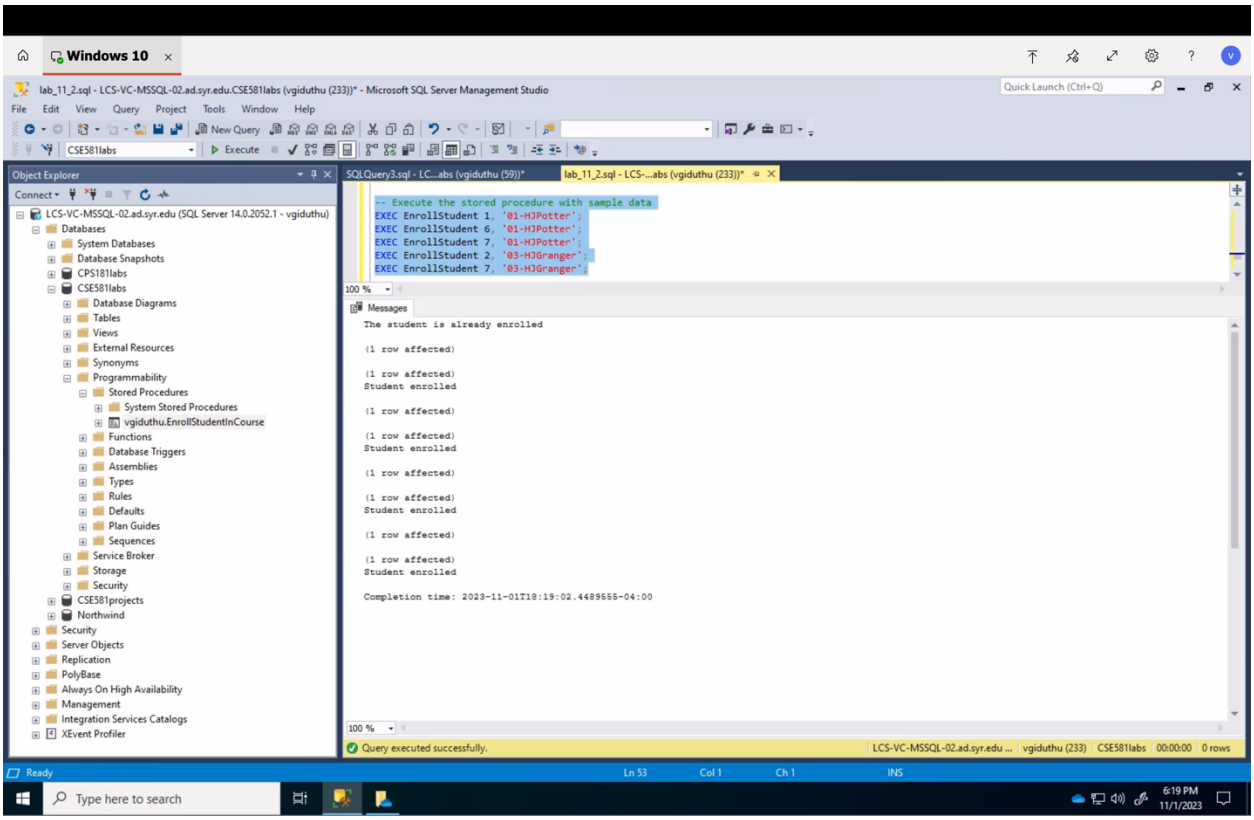

Windows 10
lab_11_2.sql - LCS-VC-MSSQL-02.ad.syr.edu.CSE581labs (vgiduthu (233))* - Microsoft SQL Server Management Studio
Object Explorer
SQLQuery3.sql - LC...abs (vgiduthu (59))*
lab_11_2.sql - LCS-...abs (vgiduthu (233))*
USE CSE581labs;
GO
CREATE PROCEDURE EnrollStudent (
@CourseId INT,
@StudentId VARCHAR(20)
) AS
BEGIN
DECLARE @EnrollStatus INT, @AssignFaculty VARCHAR(20), @OutPrint VARCHAR(MAX), @LeftSeat INT;
SELECT @EnrollStatus = (SELECT EnrollmentId
FROM CourseEnrollment
WHERE CourseId = @CourseId AND StudentId = @StudentId);
IF @EnrollStatus IS NOT NULL
BEGIN
SET @OutPrint = 'The student is already enrolled';
END
ELSE
BEGIN
SELECT @AssignFaculty = (SELECT Faculty
FROM Courses
WHERE CourseId = @CourseId);
SELECT @LeftSeat = (SELECT OpenSeats
FROM Courses
WHERE CourseId = @CourseId);
Messages
Commands completed successfully.
Completion time: 2023-11-01T18:16:46.4646367-04:00
Query executed successfully.
Ready
Ln 47
Col 3
Ch 3
INS
Type here to search
6:17 PM
11/1/2023

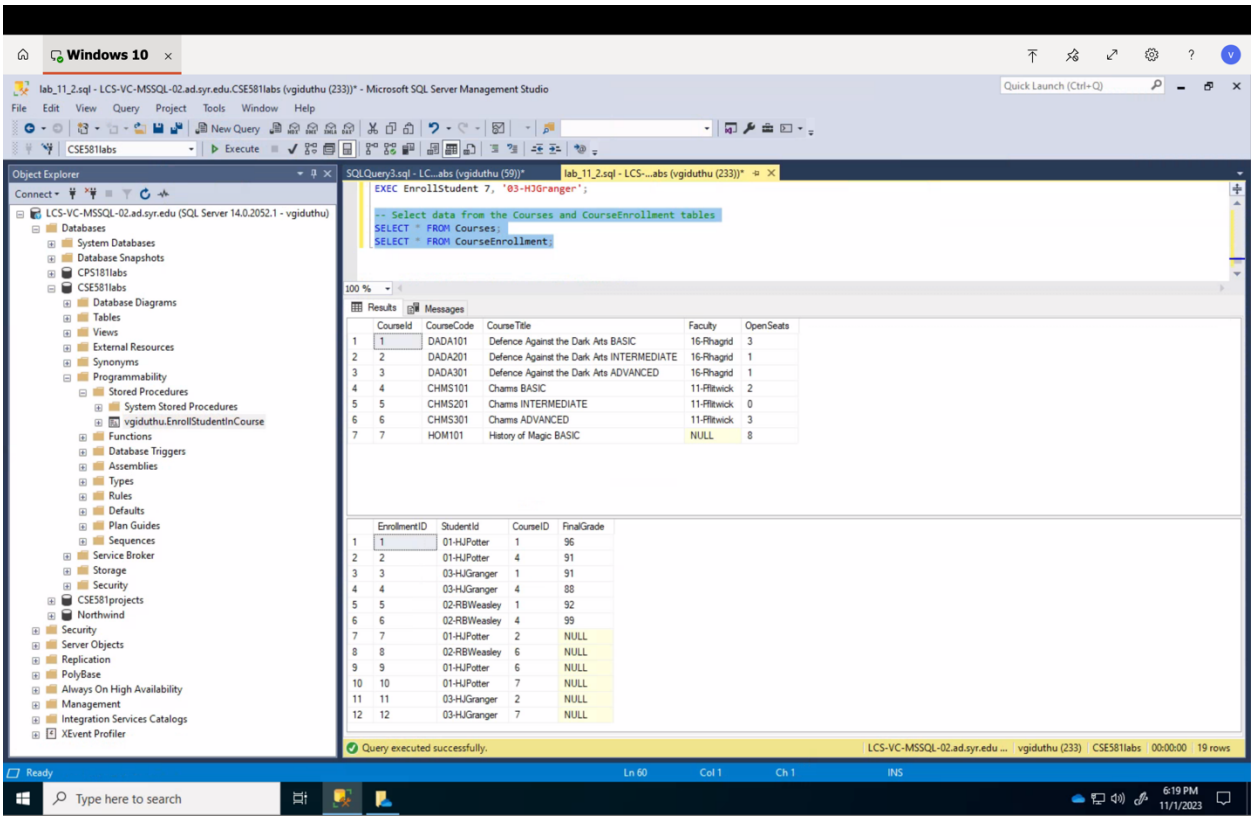

Windows 10
lab_11_2.sql - LCS-VC-MSSQL-02.ad.syr.edu.CSE581labs (vgiduthu (233))* - Microsoft SQL Server Management Studio
Object Explorer
SQLQuery3.sql - LC...abs (vgiduthu (59))*
lab_11_2.sql - LCS-...abs (vgiduthu (233))*
WHERE CourseId = @CourseId);
SELECT @LeftSeat = (SELECT OpenSeats
FROM Courses
WHERE CourseId = @CourseId);
IF @LeftSeat > 0
BEGIN
INSERT INTO CourseEnrollment (StudentId, CourseId)
VALUES (@StudentId, @CourseId);
UPDATE Courses
SET OpenSeats = @LeftSeat - 1
WHERE CourseId = @CourseId;
SET @OutPrint = 'Student enrolled';
END
ELSE
BEGIN
SET @OutPrint = 'No seat available. Student not enrolled';
END
END
PRINT @OutPrint;
END;
GO
Messages
Commands completed successfully.
Completion time: 2023-11-01T18:16:46.4646367-04:00
Query executed successfully.
Ready
Ln 47
Col 3
Ch 3
INS
Type here to search
6:18 PM
11/1/2023

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



3. 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.




Windows 10
lab_11_2.sql - LCS-VC-MSSQL-02.ad.syr.edu.CSE581labs (vgiduthu (233))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
CSE581labs
Object Explorer
Connect
LCS-VC-MSSQL-02.ad.syr.edu (SQL Server 14.0.2052.1 - vgiduthu)
Databases
System Databases
Database Snapshots
CPS181labs
CSE581labs
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Stored Procedures
System Stored Procedures
vgiduthu.EnrollStudentInCourse
Functions
Database Triggers
Assemblies
Types
Rules
Defaults
Plan Guides
Sequences
Service Broker
Storage
Security
CSE581projects
Northwind
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
XEvent Profiler
SQLQuery3.sql - LC...abs (vgiduthu (59))*
lab_11_2.sql - LCS-...abs (vgiduthu (233))*
EXEC EnrollStudent 7, '03-HJGranger';
-- Select data from the Courses and CourseEnrollment tables
SELECT * FROM Courses;
SELECT * FROM CourseEnrollment;
100 %
Results
Messages
CourseId CourseCode CourseTitle Faculty OpenSeats
1 1 DADA101 Defence Against the Dark Arts BASIC 16-Rhagrid 3
2 2 DADA201 Defence Against the Dark Arts INTERMEDIATE 16-Rhagrid 1
3 3 DADA301 Defence Against the Dark Arts ADVANCED 16-Rhagrid 1
4 4 CHMS101 Charms BASIC 11-Flitwick 2
5 5 CHMS201 Charms INTERMEDIATE 11-Flitwick 0
6 6 CHMS301 Charms ADVANCED 11-Flitwick 3
7 7 HOM101 History of Magic BASIC NULL 8
EnrollmentID StudentId CourseID FinalGrade
1 1 01-HJPotter 1 96
2 2 01-HJPotter 4 91
3 3 03-HJGranger 1 91
4 4 03-HJGranger 4 88
5 5 02-RBWeasley 1 92
6 6 02-RBWeasley 4 99
7 7 01-HJPotter 2 NULL
8 8 02-RBWeasley 6 NULL
9 9 01-HJPotter 6 NULL
10 10 01-HJPotter 7 NULL
11 11 03-HJGranger 2 NULL
12 12 03-HJGranger 7 NULL
Query executed successfully.
LCS-VC-MSSQL-02.ad.syr.edu ... vgiduthu (233) CSE581labs 00:00:00 19 rows
Ready
Ln 60 Col 1 Ch 1 INS
Type here to search
6:19 PM
11/1/2023