CSE581

Project 3

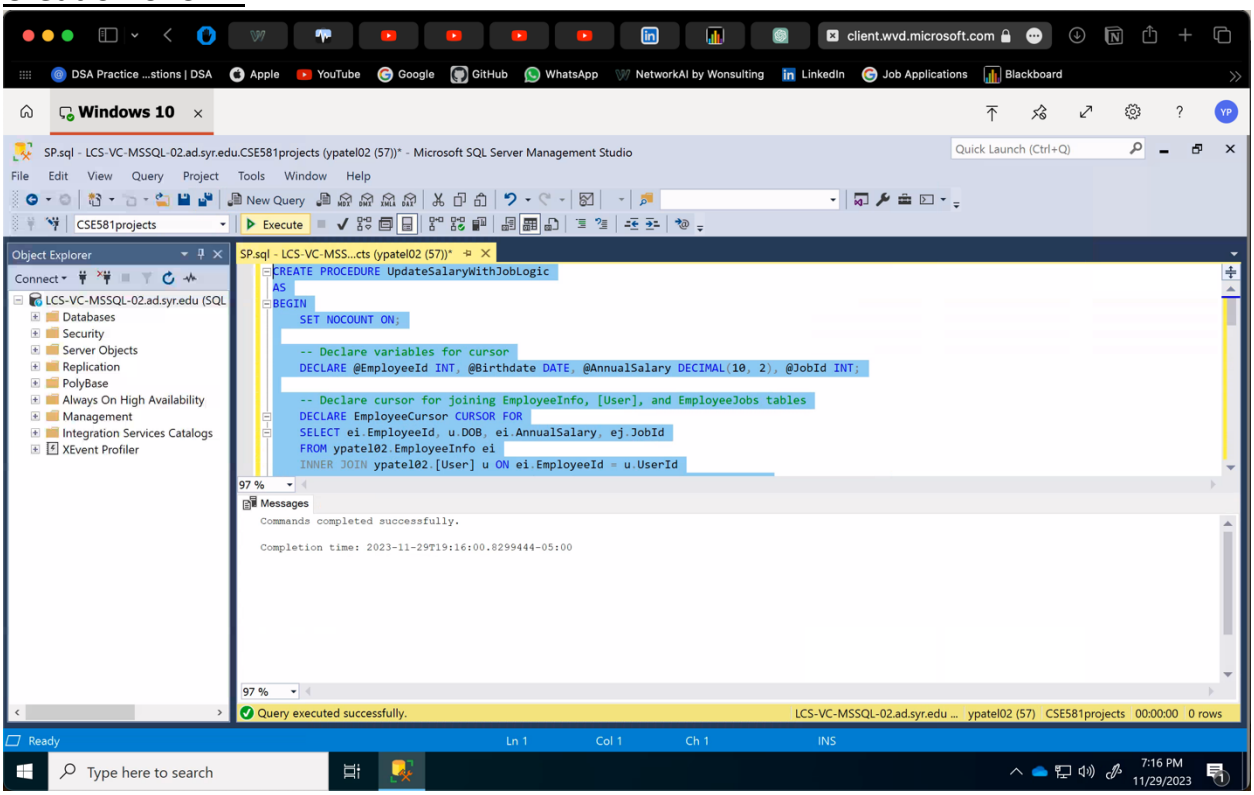
Submission By – Yash Rajeshkumar Patel | SUID - 517958851

- **(10% points) SP to use cursor(s)**

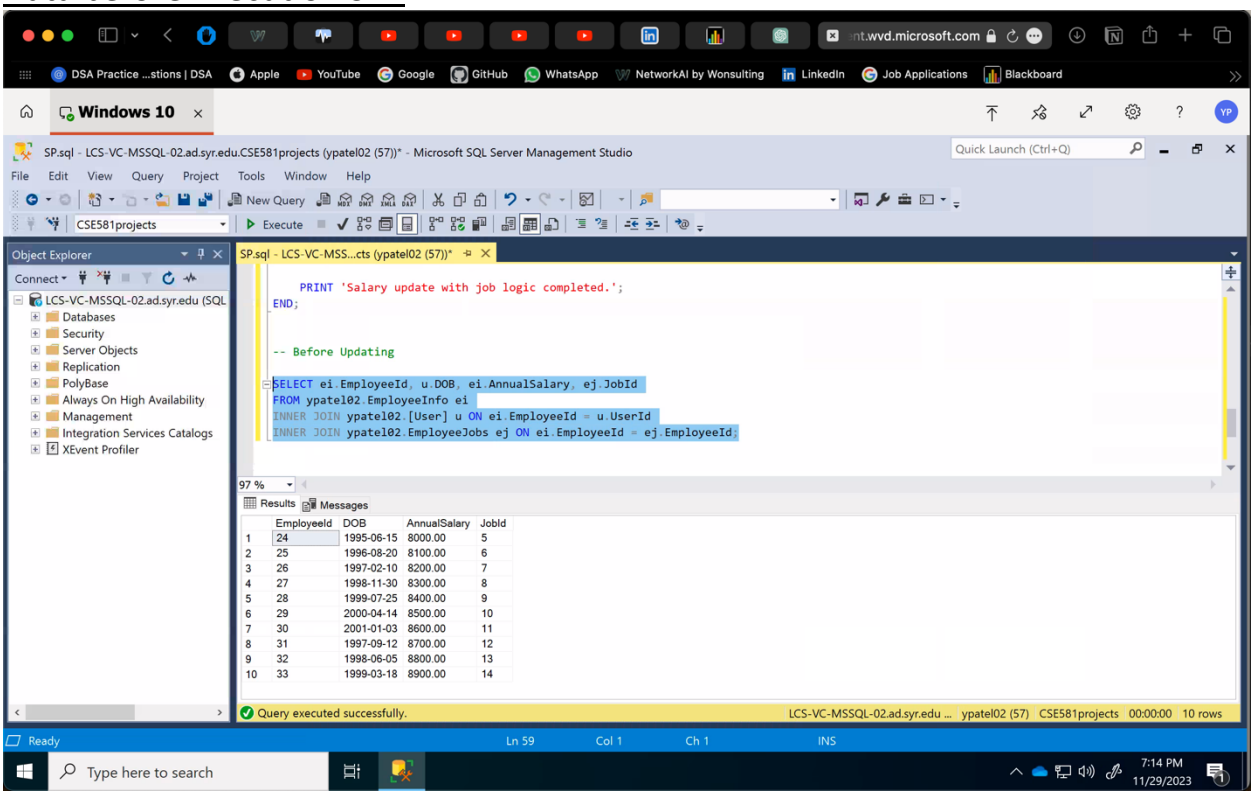
Stored Procedure Name: **UpdateSalaryWithJobLogic**

Purpose: The UpdateSalaryWithJobLogic stored procedure is designed to intelligently adjust the annual salary of employees based on a set of business rules. This stored procedure utilizes a cursor to iterate through a result set obtained by joining three tables: EmployeeInfo, User, and EmployeeJobs. The primary focus is to identify employees aged 25 or older and, for a specific set of job roles (JobIds 5, 7, 9, 11, 13), grant a 7% increase in their annual salary. For other job roles, a 5% increase is applied.

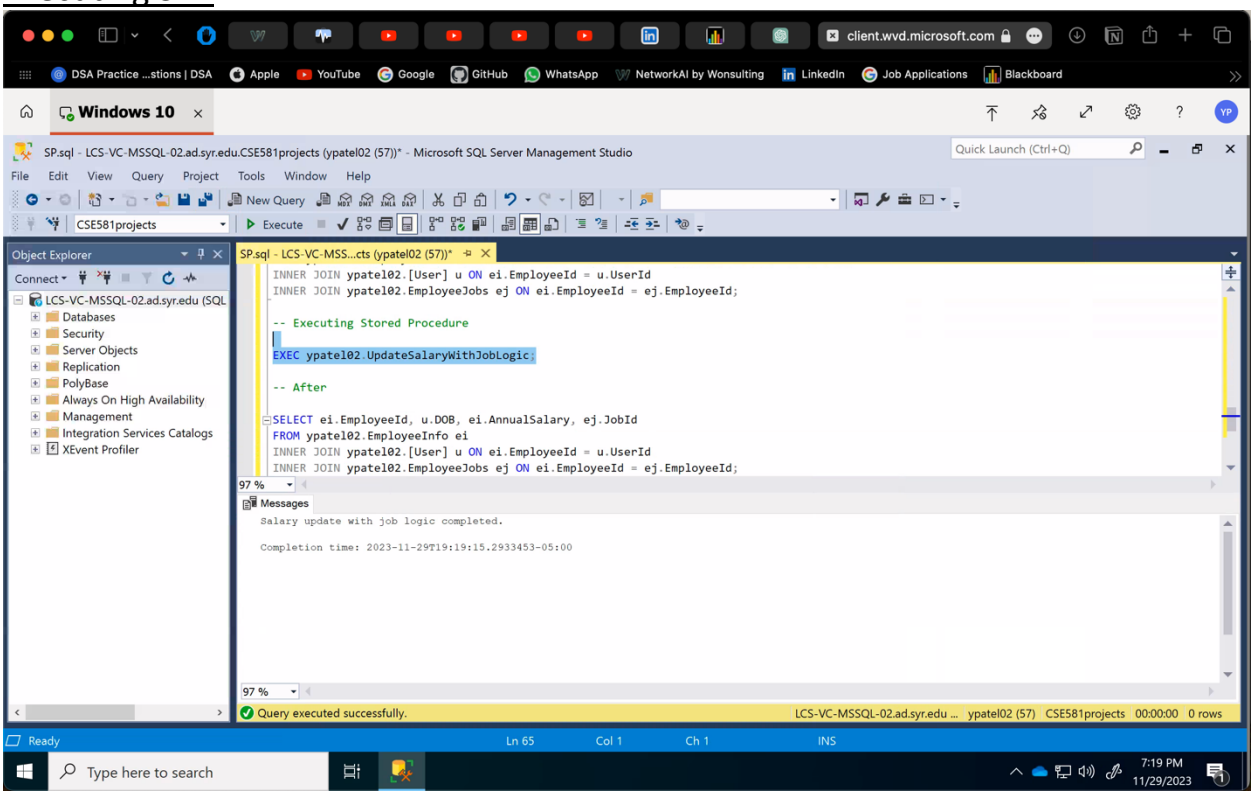
Creation of SP 1



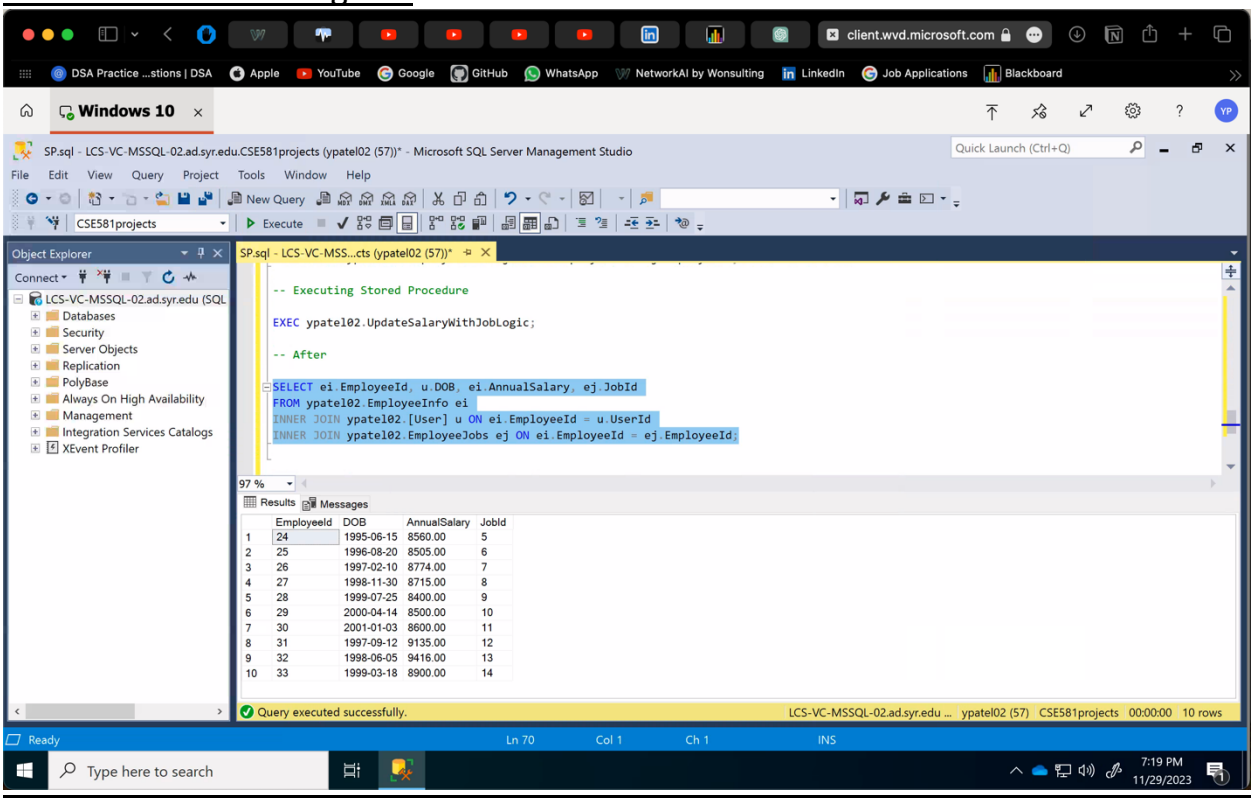
Data before Execution SP1



Executing SP1



Results after Executing SP1



- **(10% points) SP to update data in a table (perform validation)**

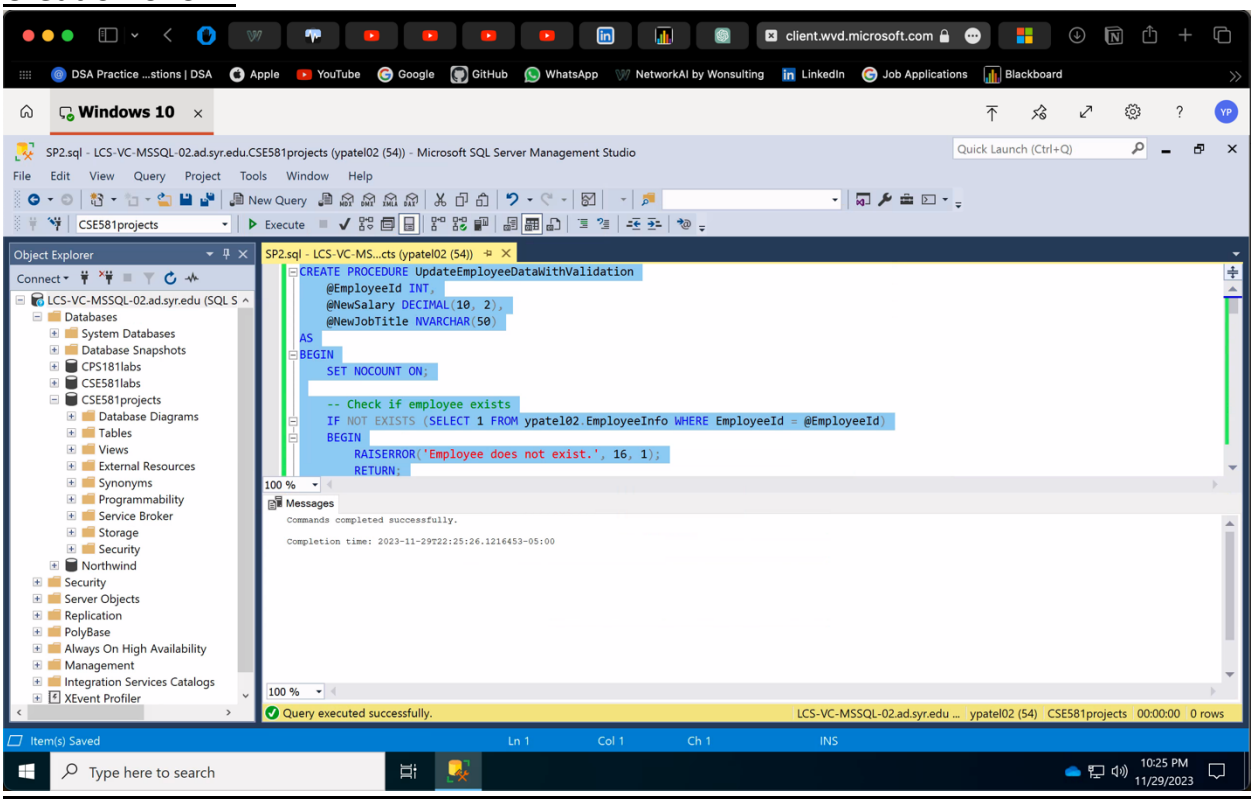
Stored Procedure Name: **UpdateEmployeeDataWithValidation**

Description : This stored procedure is designed to update employee data with validation checks. It takes input parameters such as @EmployeeId (for identifying the employee), @NewSalary (the updated annual salary), and @NewJobTitle (the updated job title).

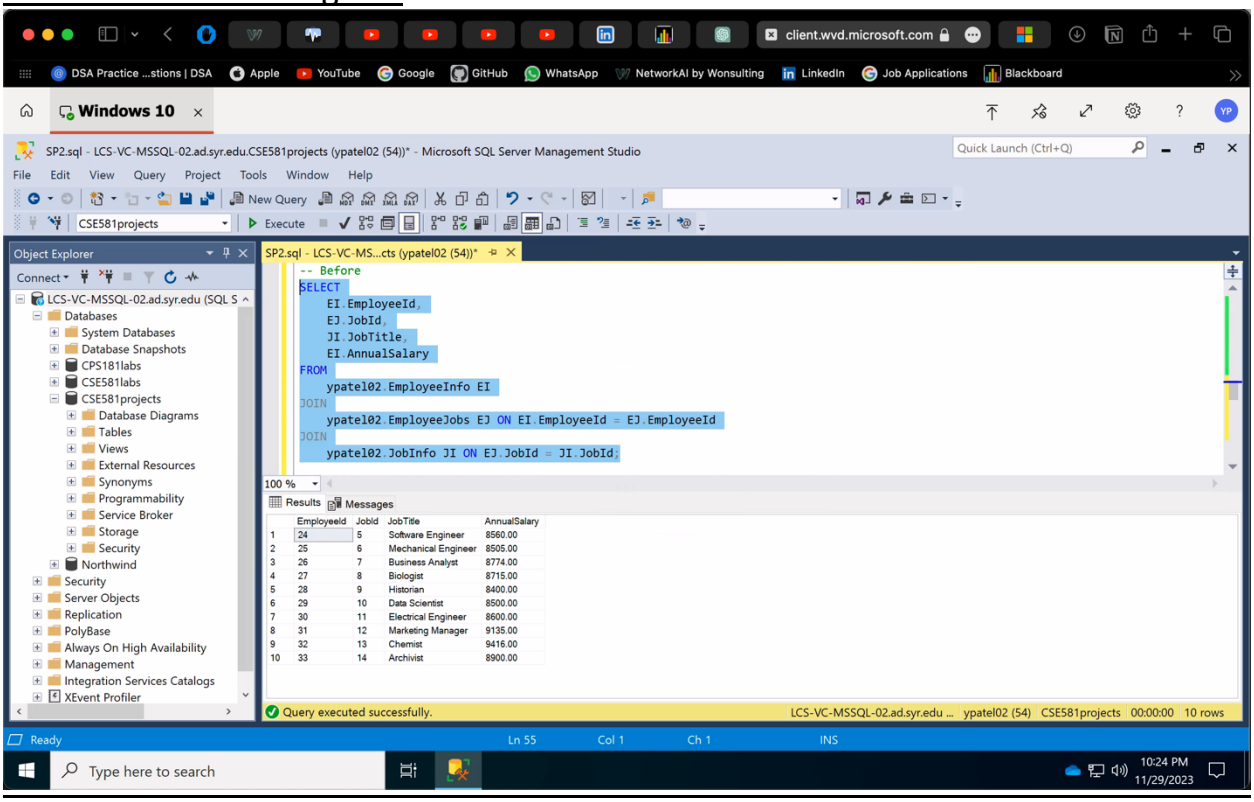
Validations :

- Verifies whether the specified employee exists in the **EmployeeInfo** table.
- Validates that the new salary provided (**@NewSalary**) is not negative.
- Validates that the new job title provided (**@NewJobTitle**) exists in the **JobInfo** table.

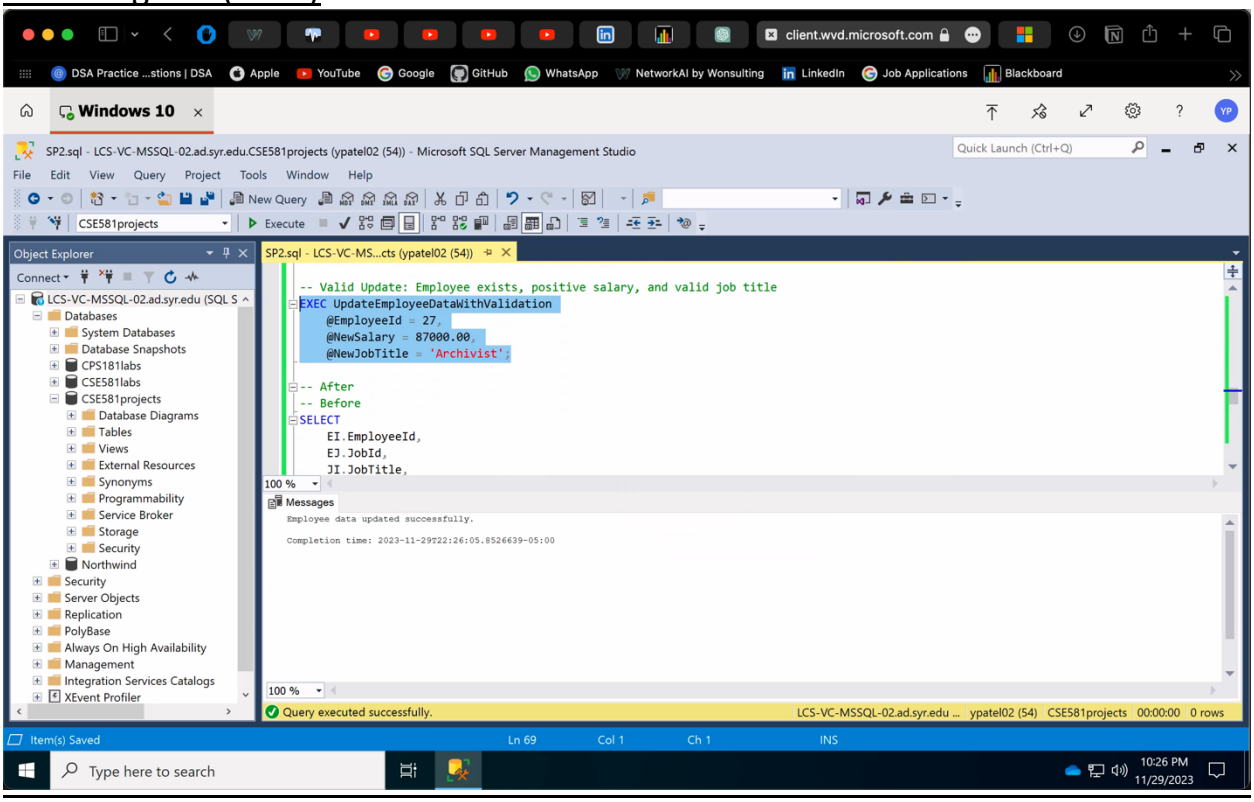
Creation of SP2



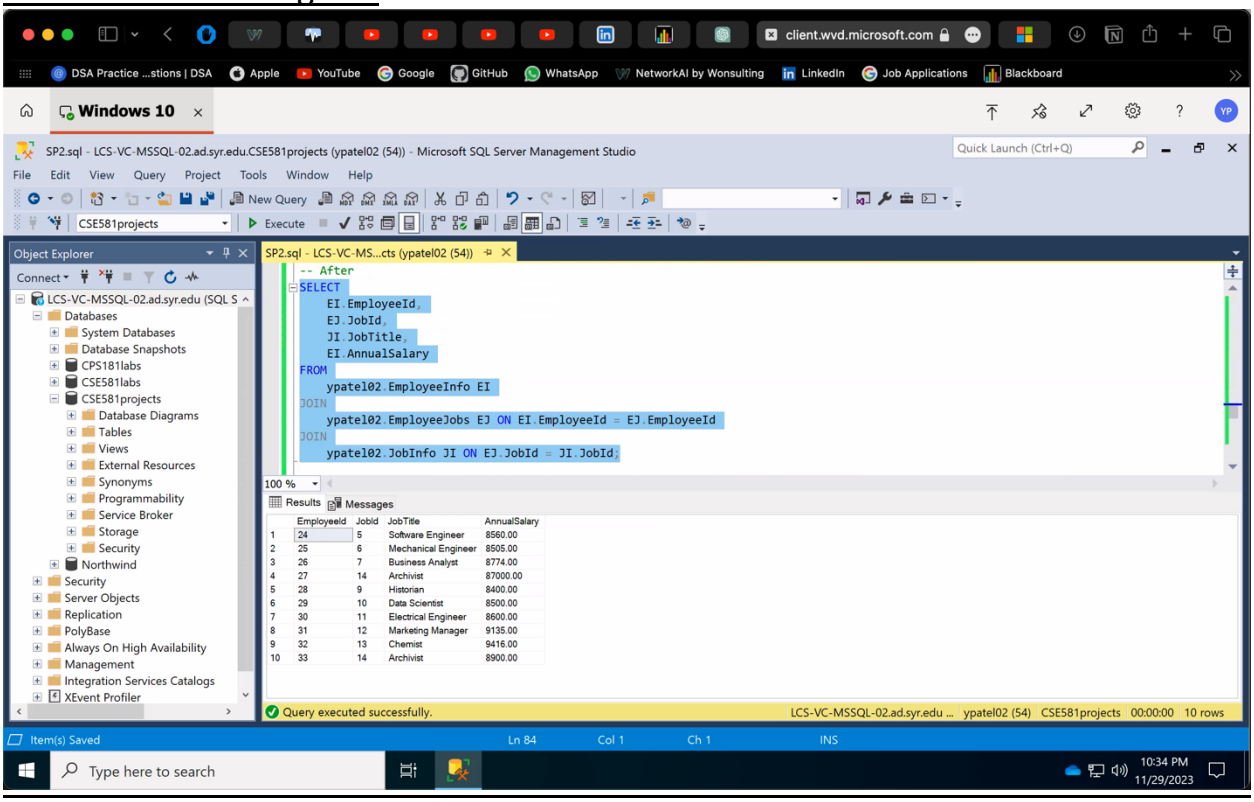
Data Before Executing SP2



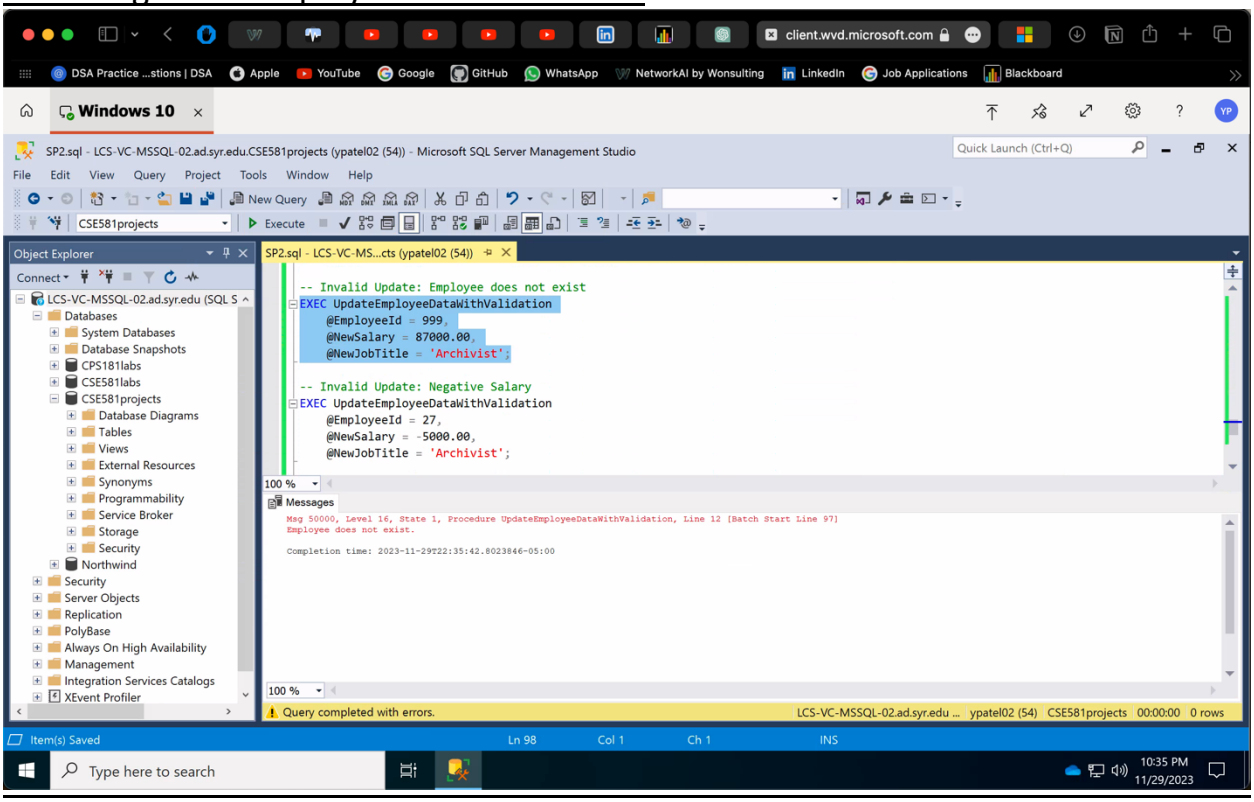
Executing SP2 (Valid)



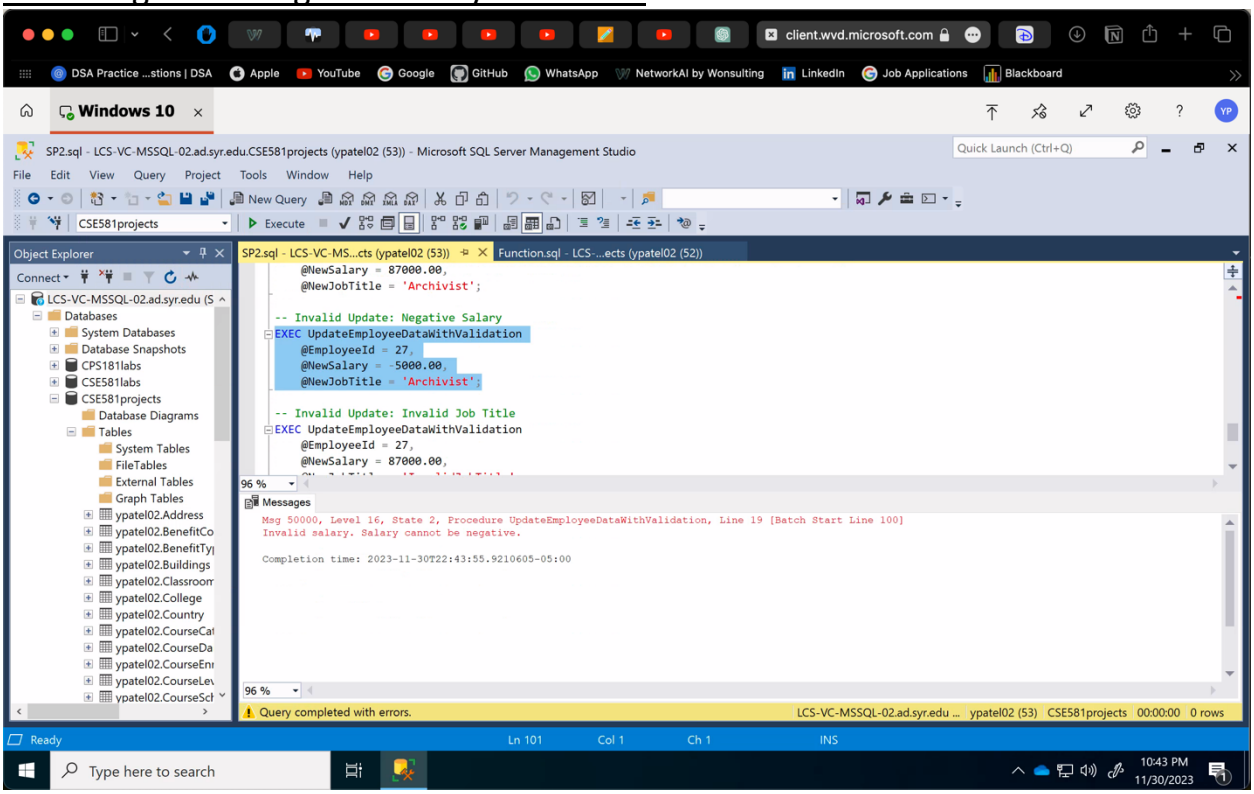
Data After Executing SP2



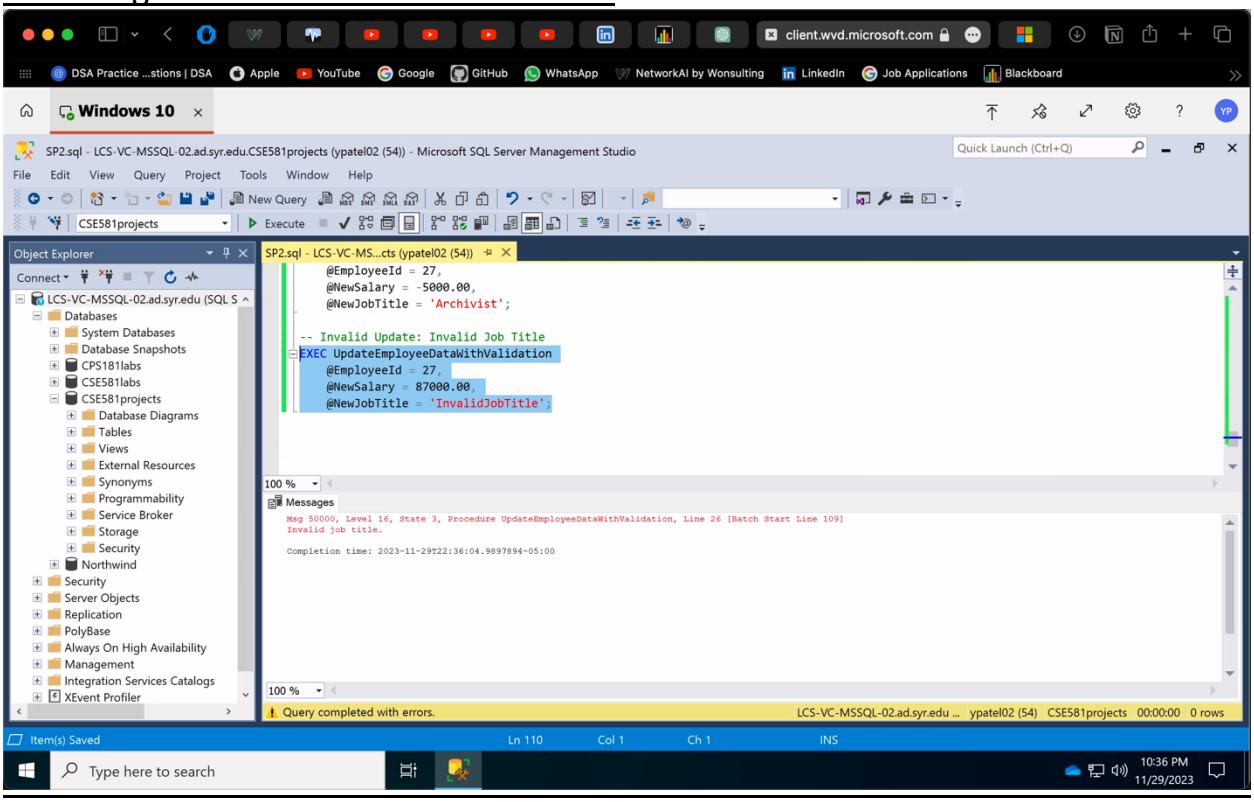
Executing when Employee does not exists.



Executing when Negative Salary is Entered



Executing when Invalid Job Title Entered

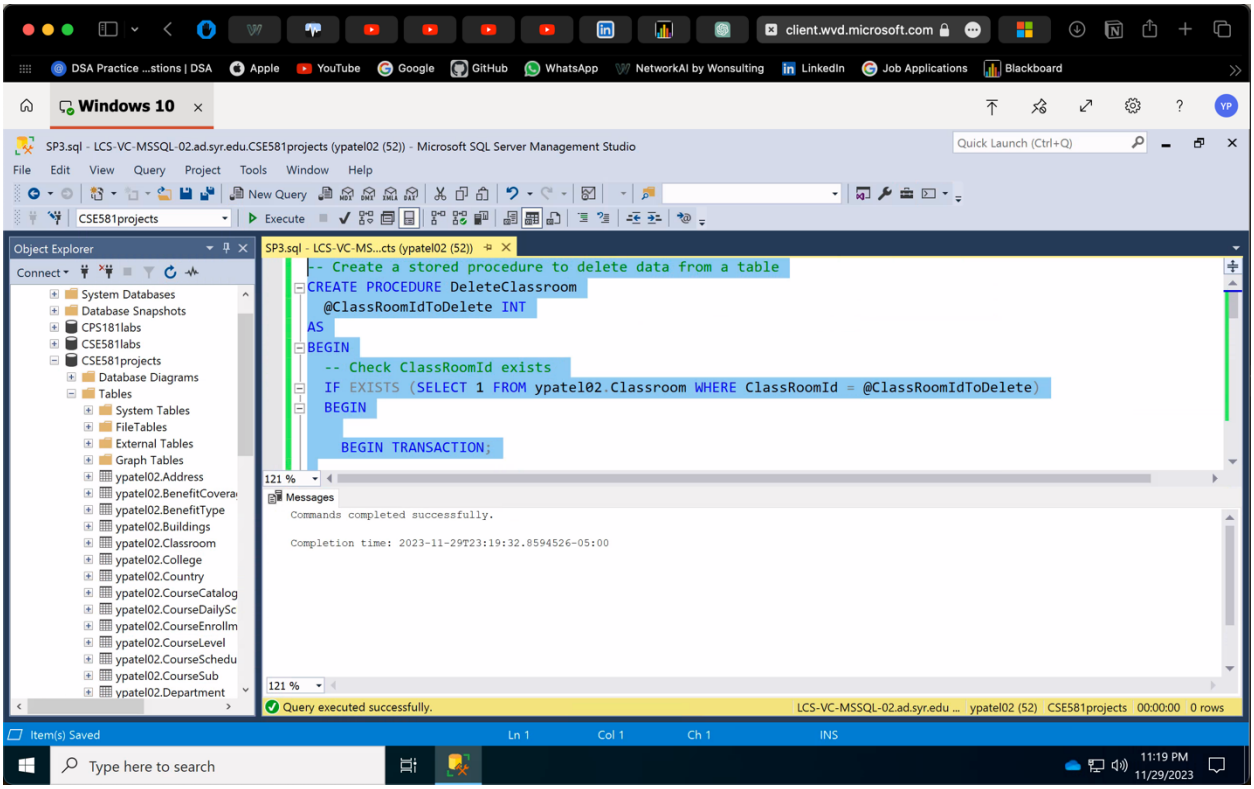


- **(10% points) SP to delete data from a table**

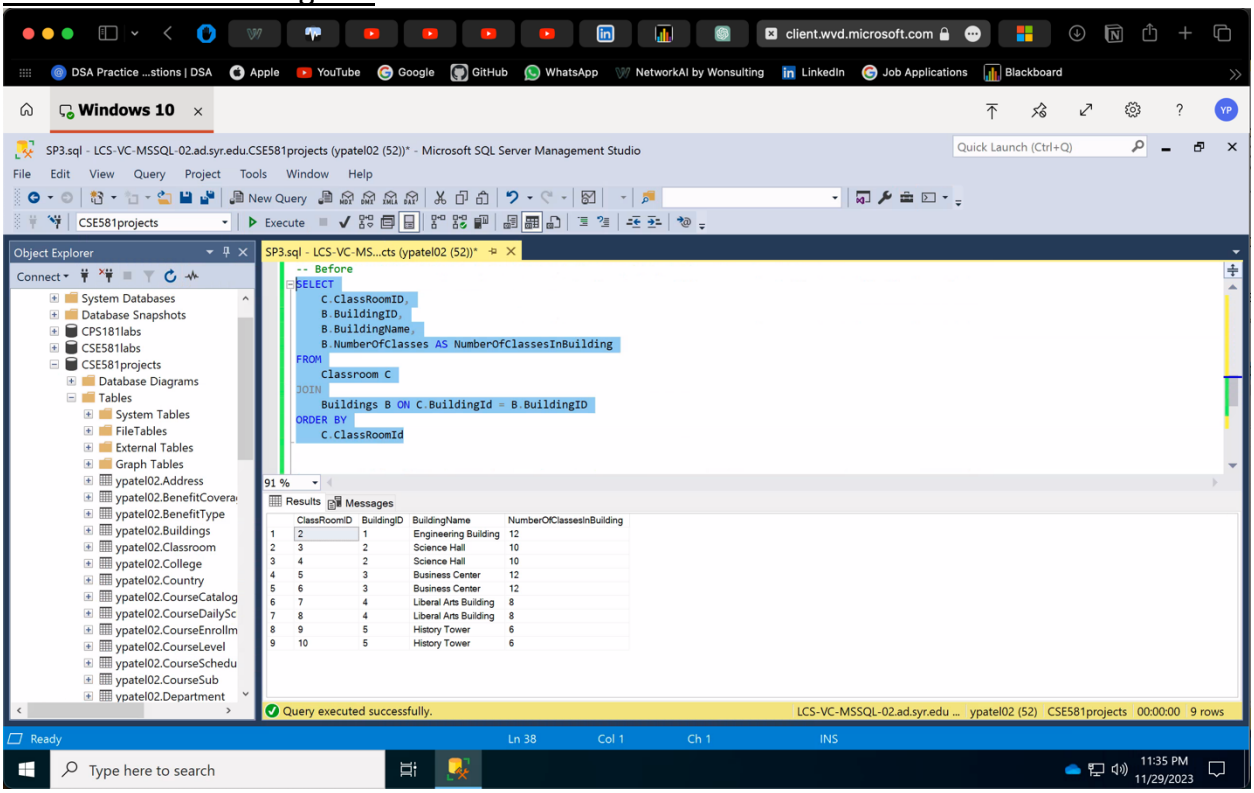
Stored Procedure Name**: DeleteClassroom**

Description: The stored procedure DeleteClassroom is designed to facilitate the secure deletion of a classroom record from the ypatel02.Classroom table in our project database. Ensure number of classrooms available in a building gets updated accordingly and sets classroom of classes to be null if that classroom was removed.

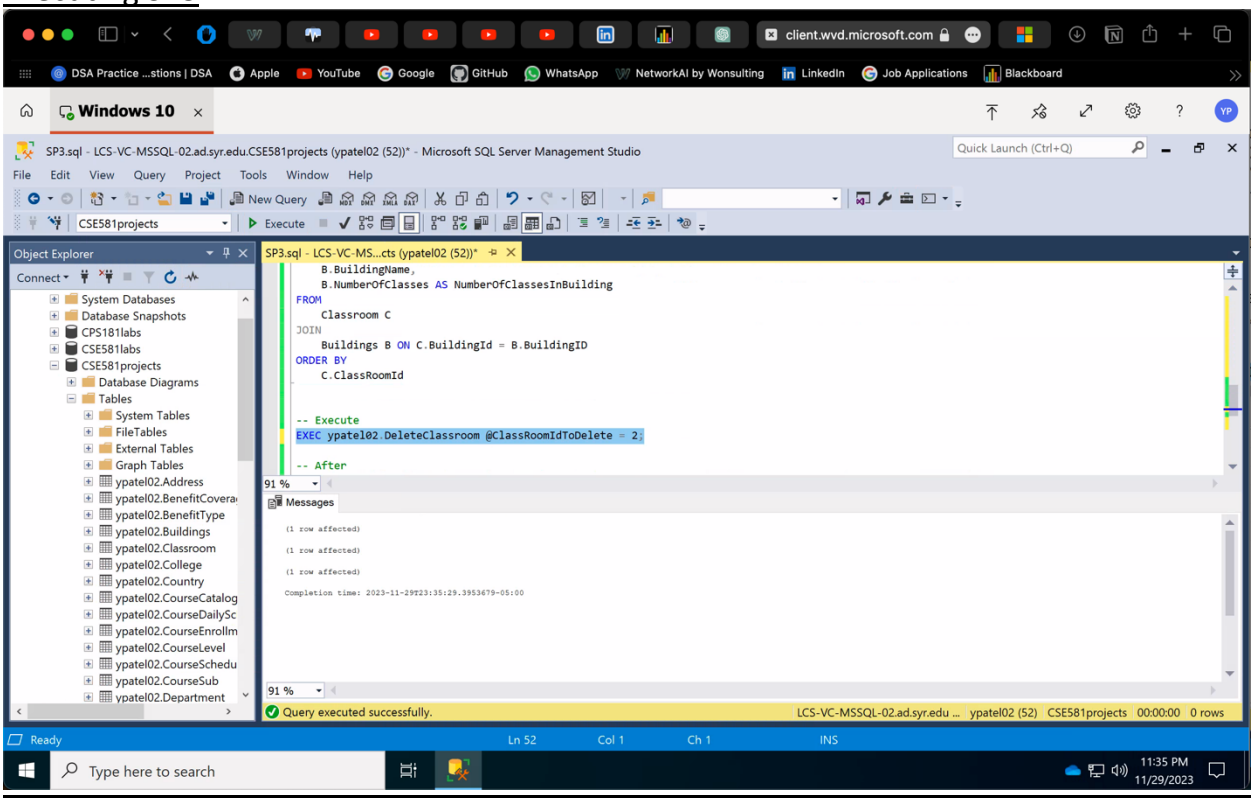
Creation of SP3



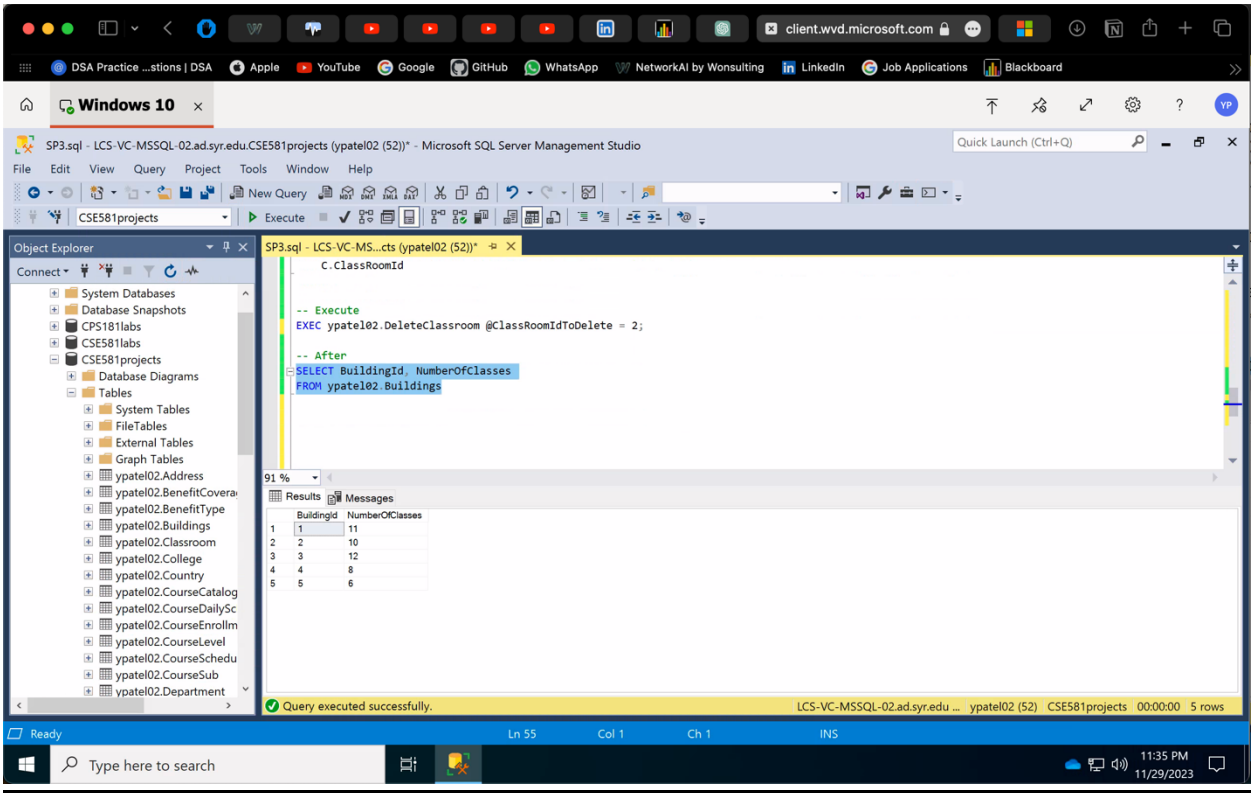
Data Before Executing SP3



Executing SP3



Data After Executing SP3



- **(10%+5% bonus points) 1 SP of your own choice, performing a business action[1].**

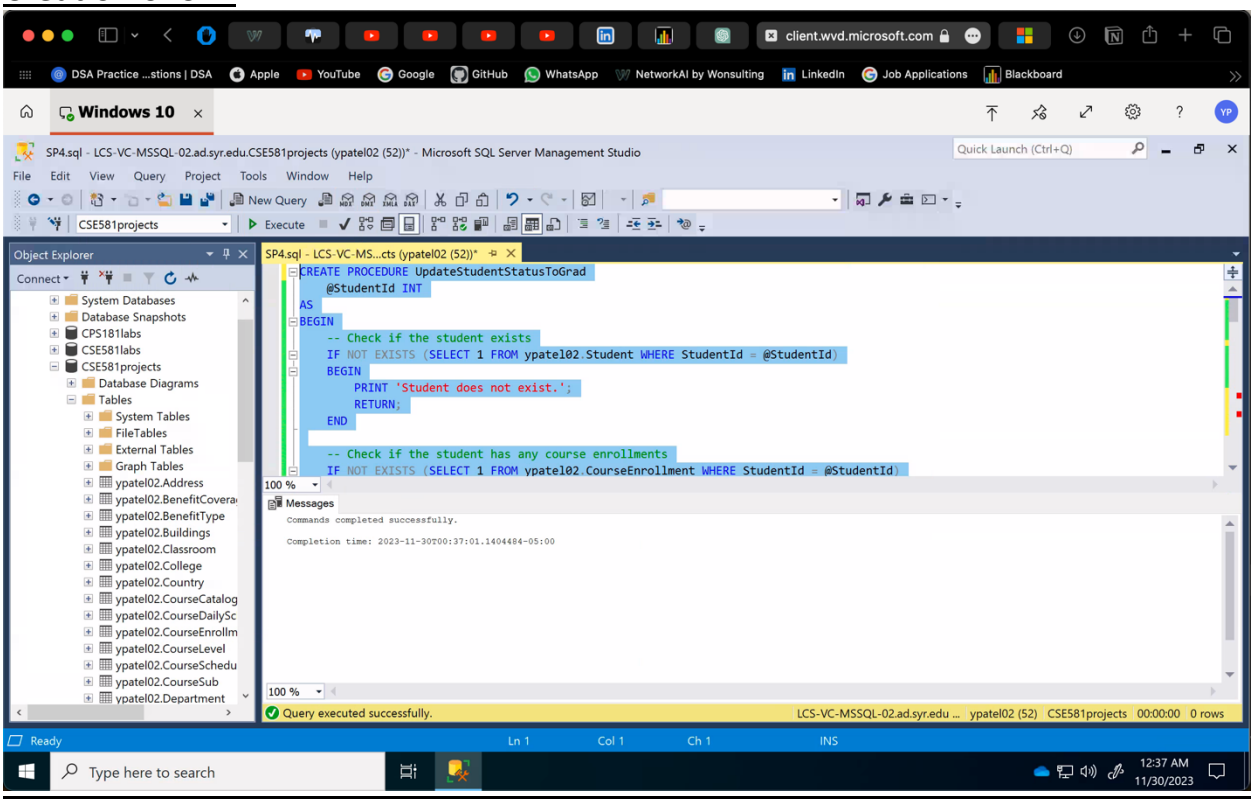
Stored Procedure Name : **UpdateStudentStatusToGrad**

Description **:** The stored procedure UpdateStudentStatusToGrad automates the process of updating a student's graduation status within the educational system. This checks Grades obtained in courses enrolled during his study and will not allow to graduate if he has failed one or more subjects.

Validations :

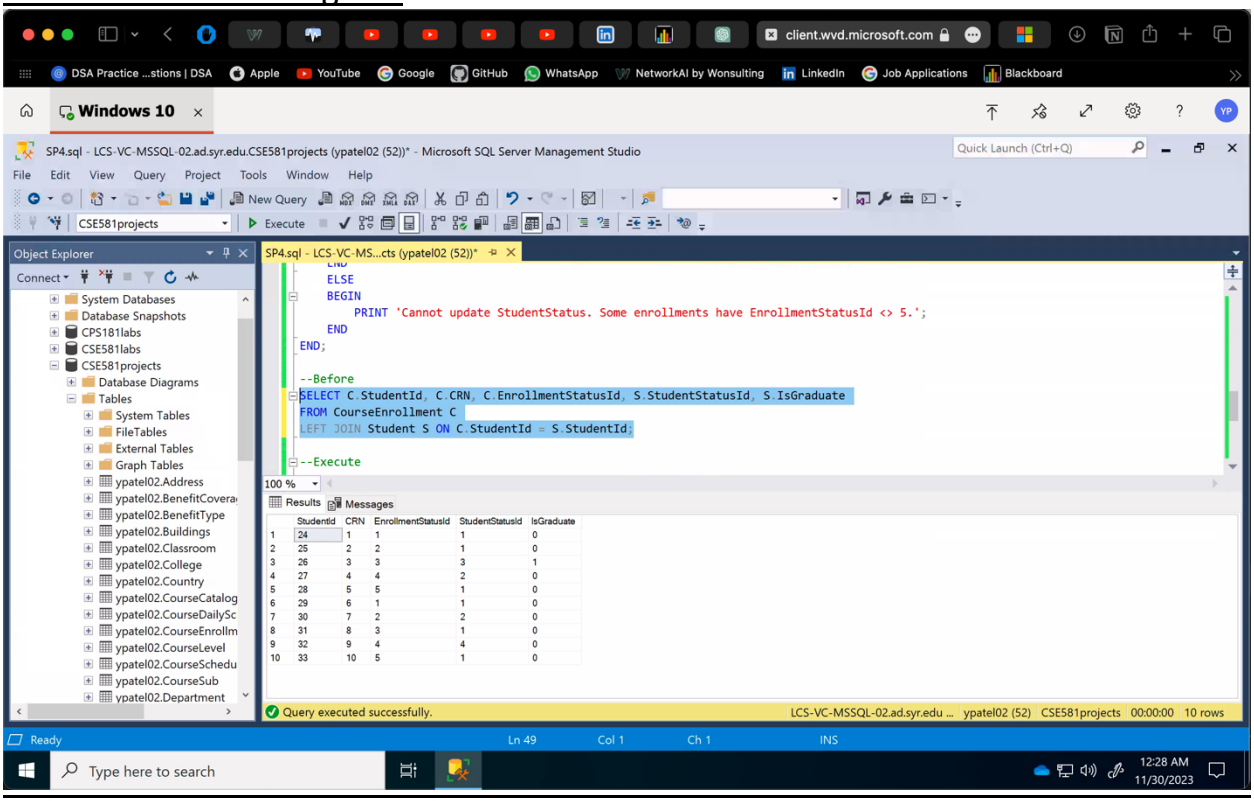
- Check if Student Exists
- Check if Student Failed one or more Courses.
- Checks if Student already Graduated.

Creation of SP4

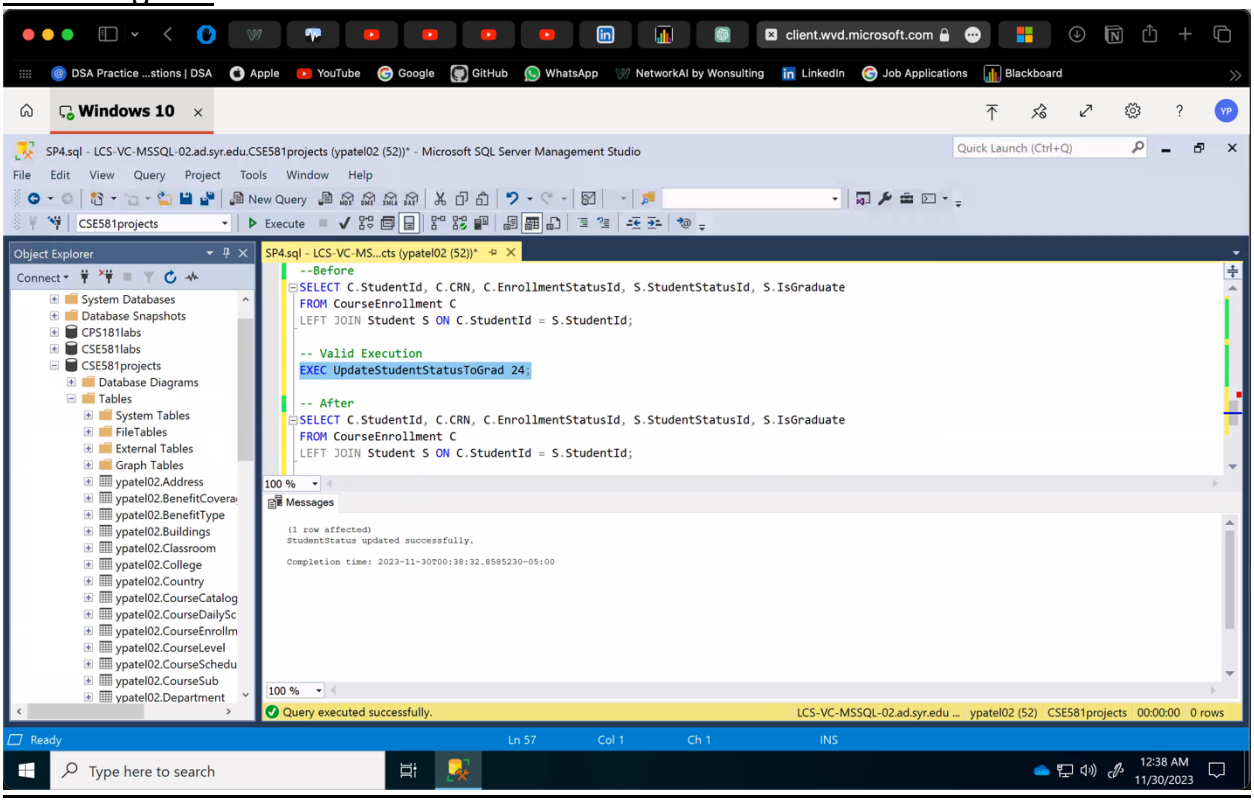


[1] You will get 5% bonus if this is a valid action within the scope of the business problem, and the SP is reasonably complex. In other words, a 4 line single-table select SP will get you nothing.

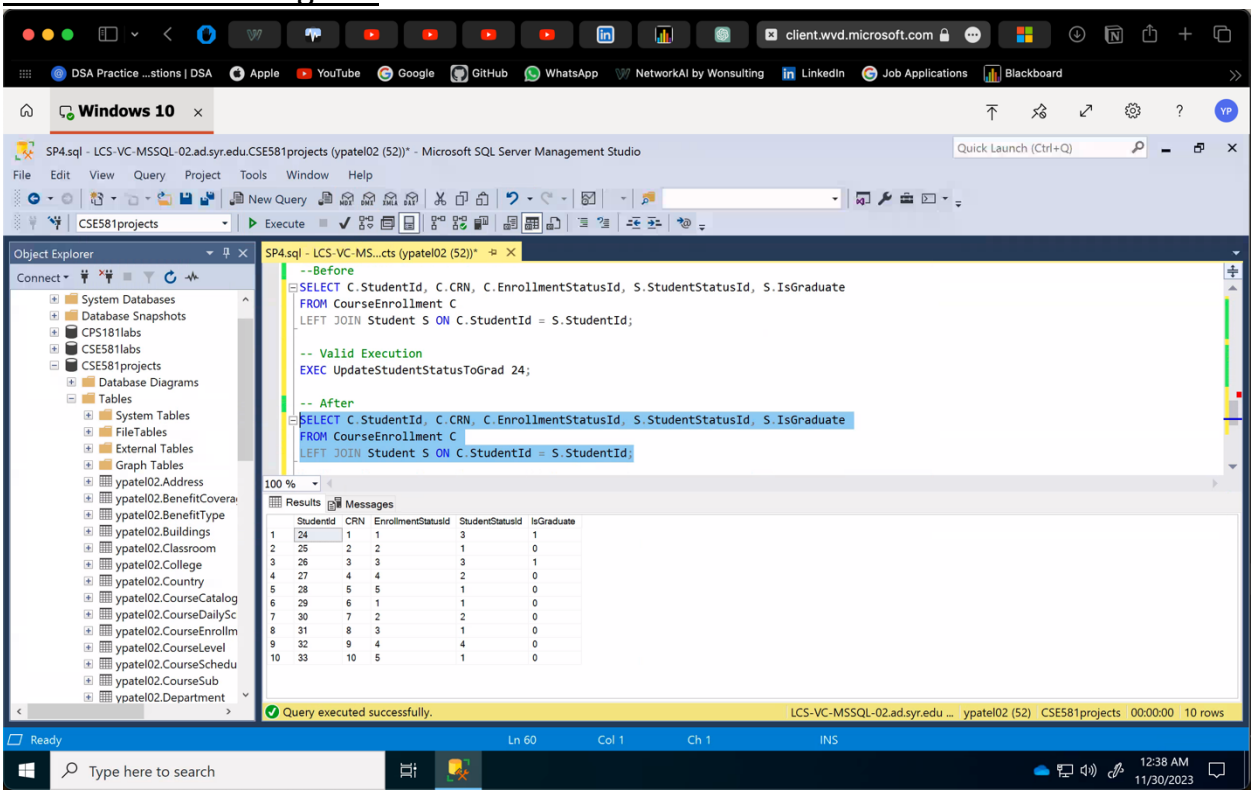
Data Before Executing SP4



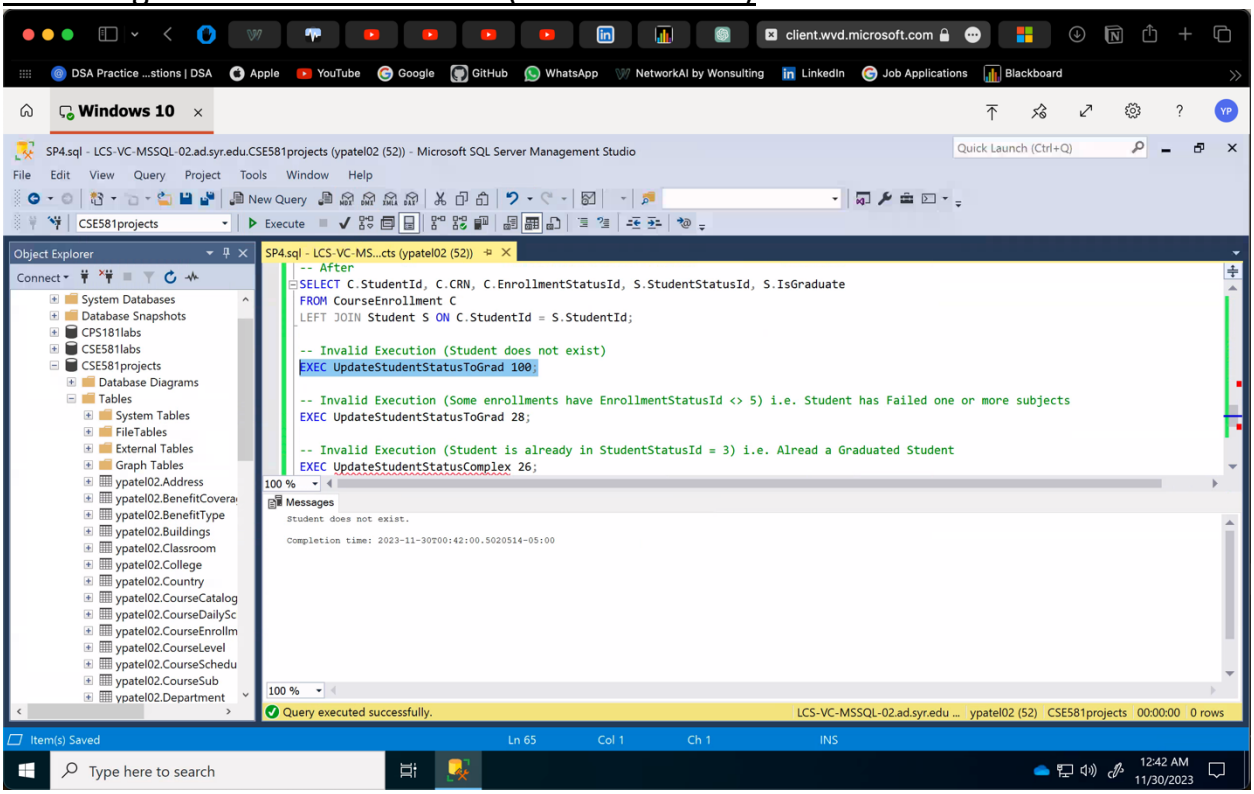
Executing SP4



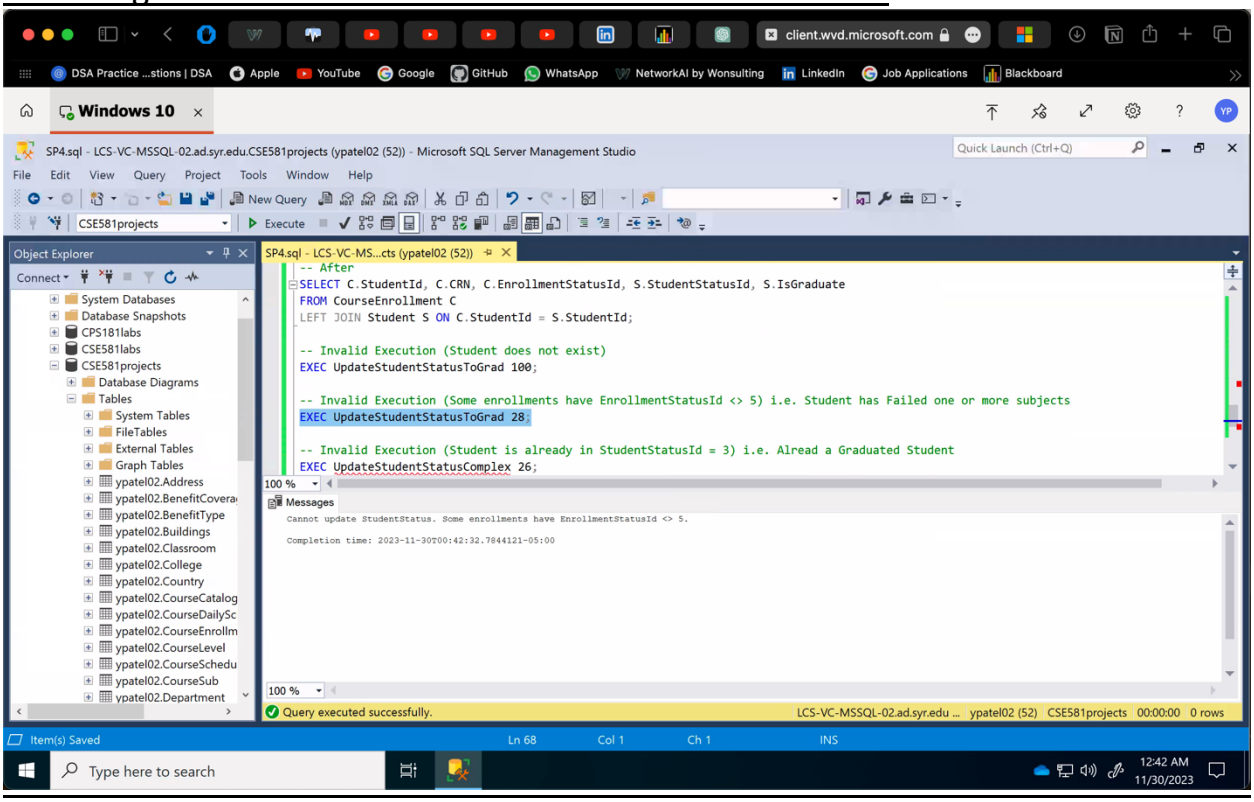
Data After Executing SP4



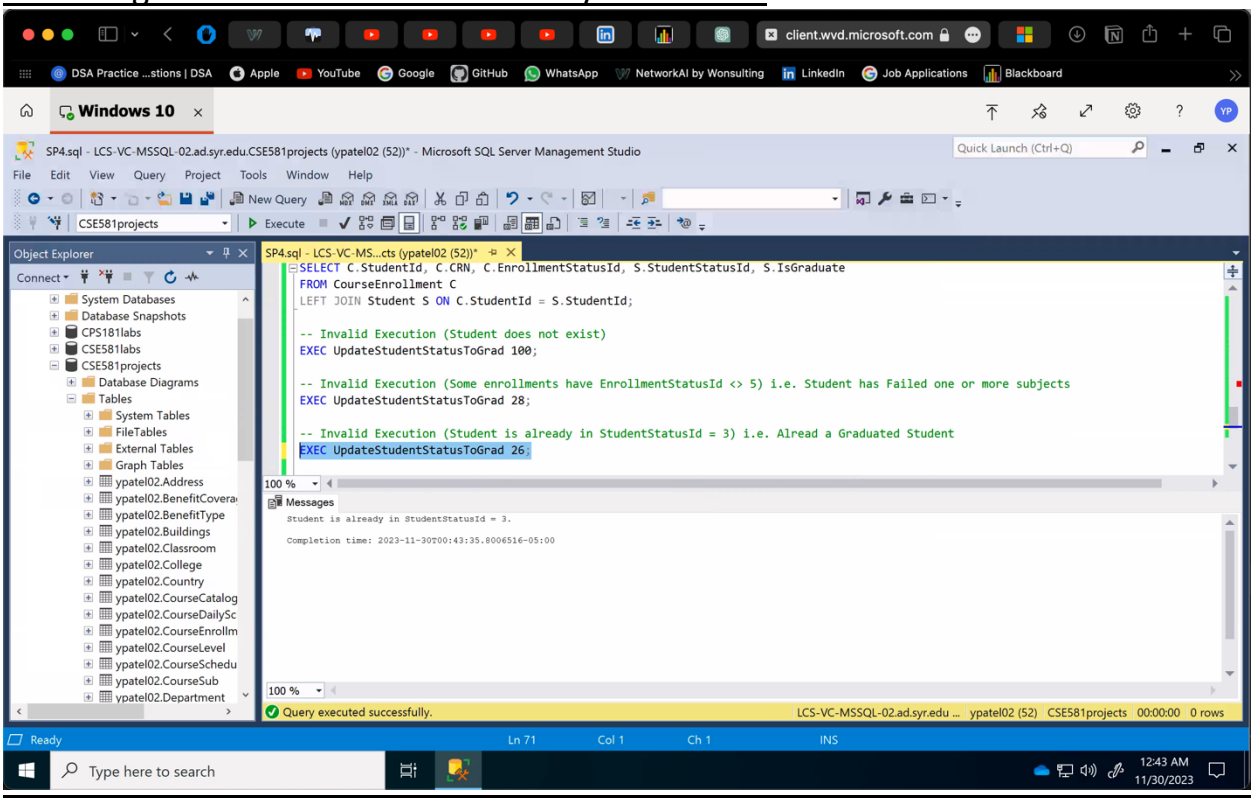
Executing SP4 on Invalid Student (does not exists)



Executing SP4 on Student who Failed one or more Courses



Executing SP4 A student who is already Graduated

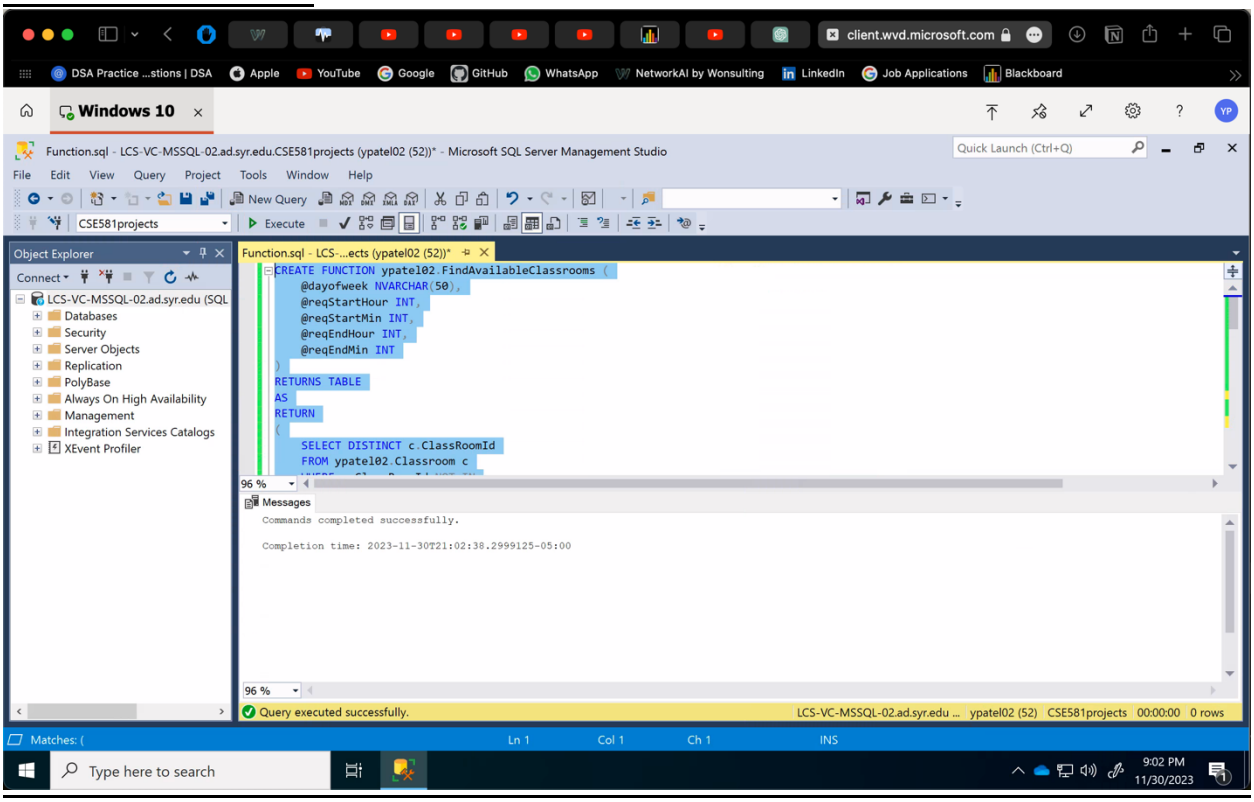


Create 1 *Function* (10% points) which can be executed by "Graders". Not the function you did for the lab.

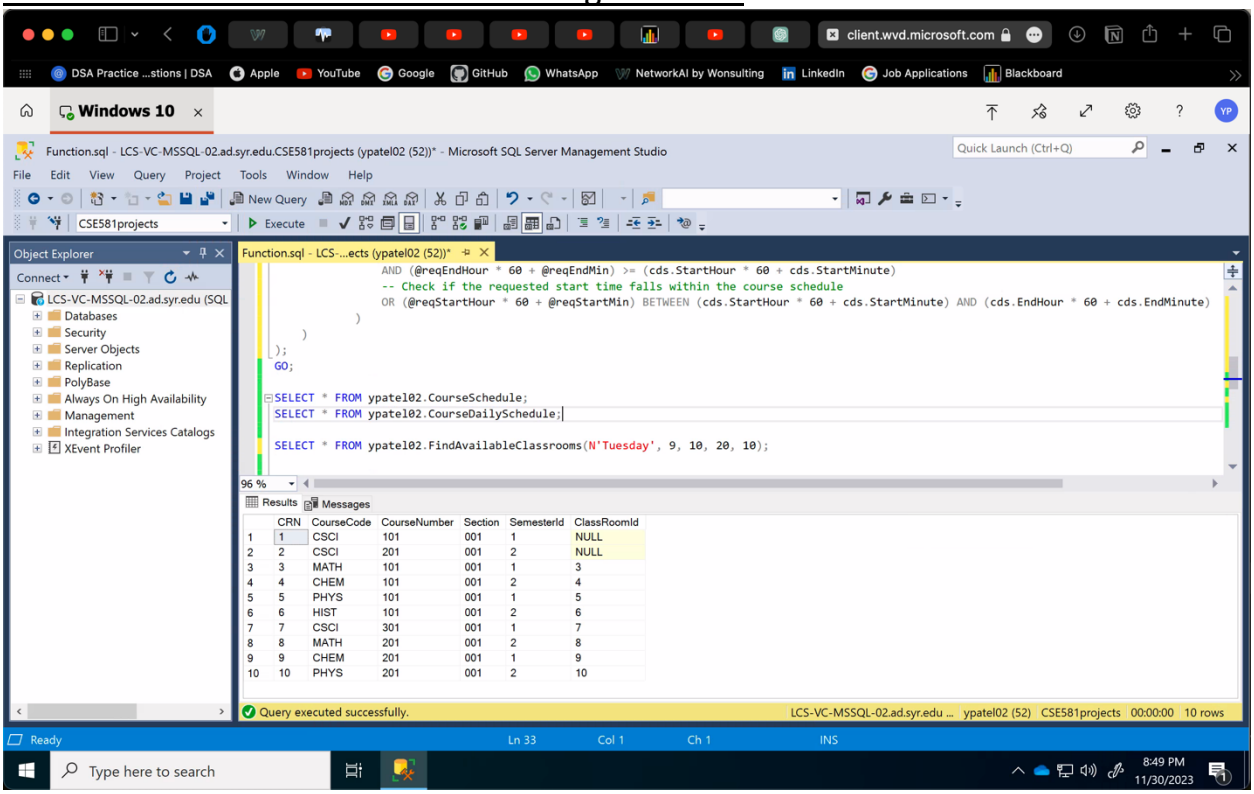
Function Name : **FindAvailableClassrooms**

Description : This function, ypatel02.FindAvailableClassrooms, is designed to find available classrooms for a specified day of the week and time range. It utilizes information from the ypatel02.Classroom, ypatel02.CourseDailySchedule, and ypatel02.CourseSchedule tables. The function identifies classrooms that are not scheduled for courses during the specified period, considering both the end and start times of the courses. Additionally, it handles cases where ClassRoomId is NULL in the CourseSchedule table. The result is a list of distinct available ClassRoomIds meeting the given criteria.

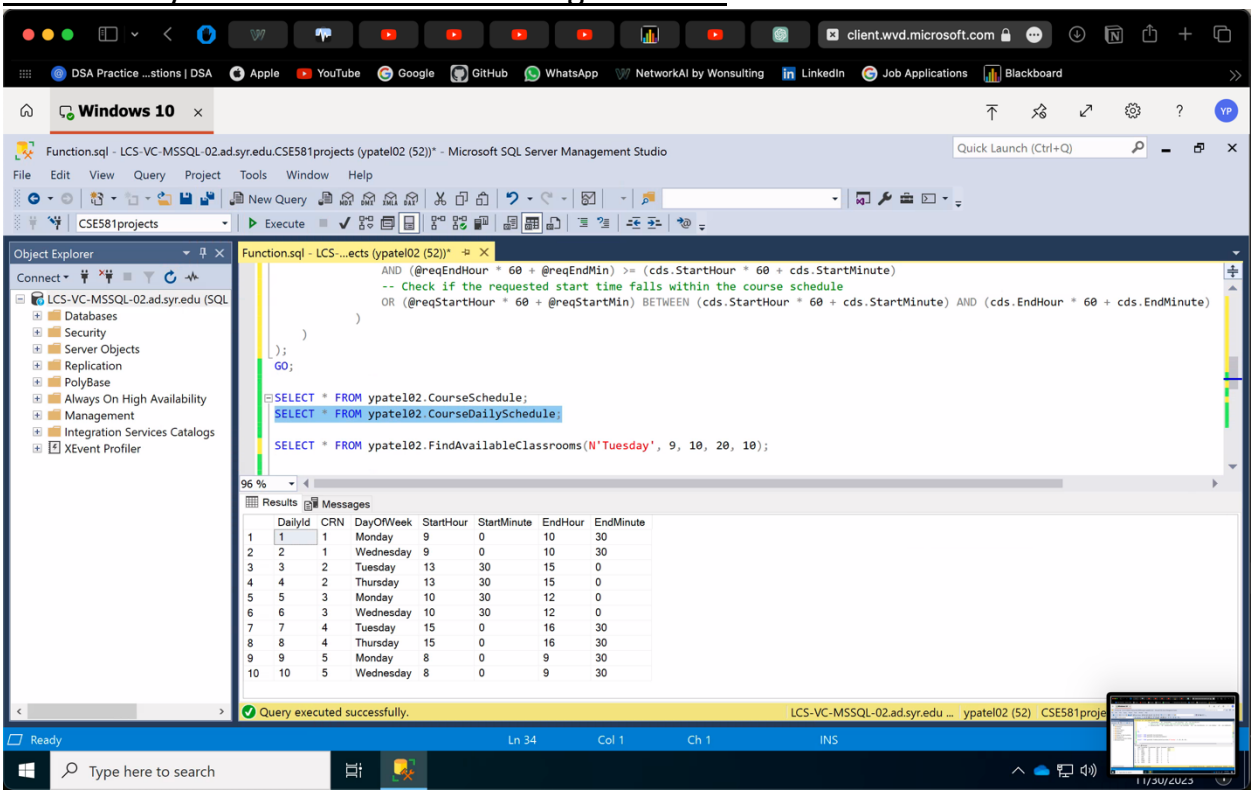
Creation of Function



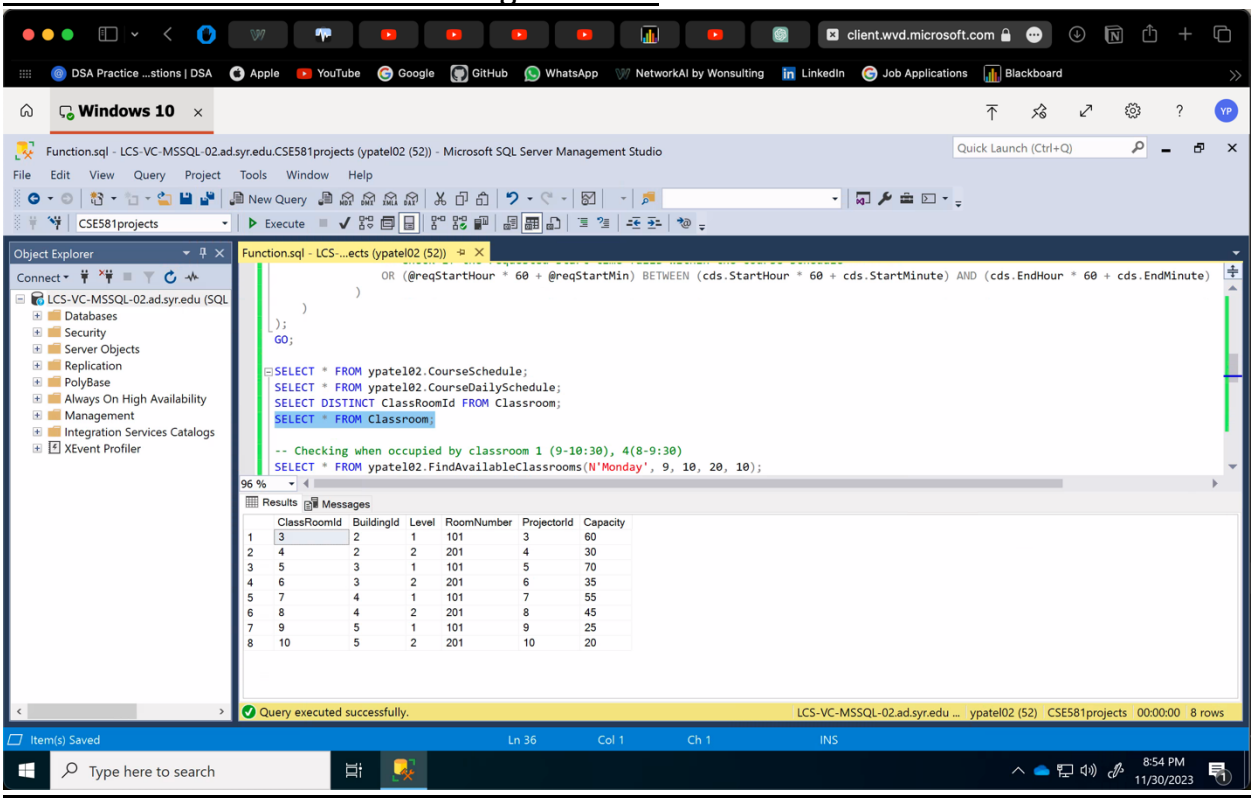
CourseSchedule Table Before Executing Function



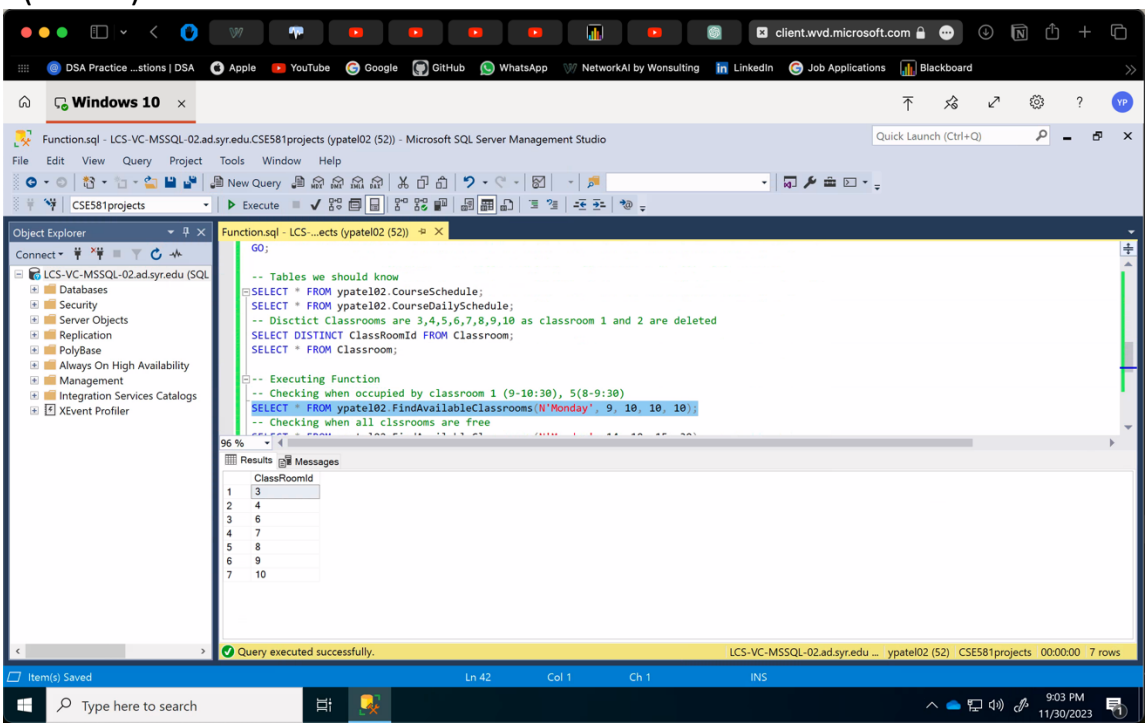
CourseDailySchedule before Executing Function



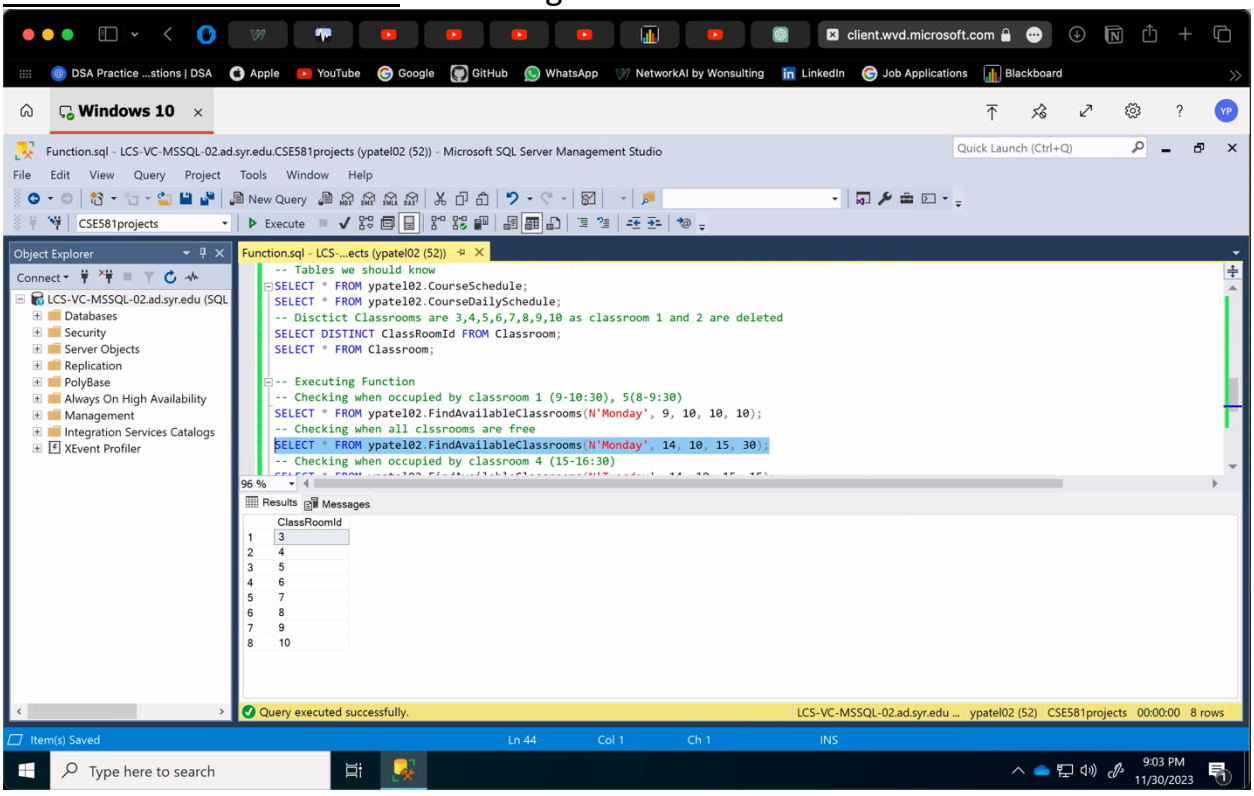
Classroom Table Before executing Function



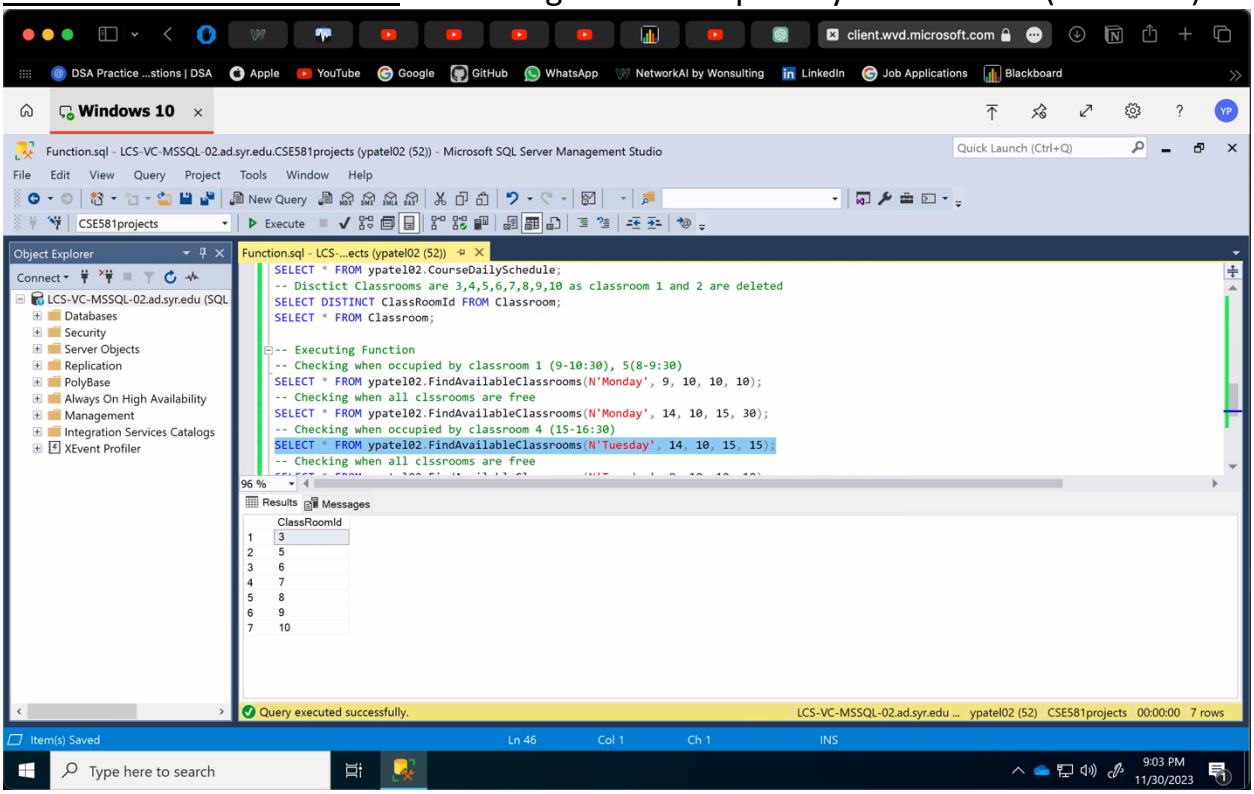
Function Execution Case 1 : Checking when occupied by classroom 1 (9-10:30), 5(8-9:30)



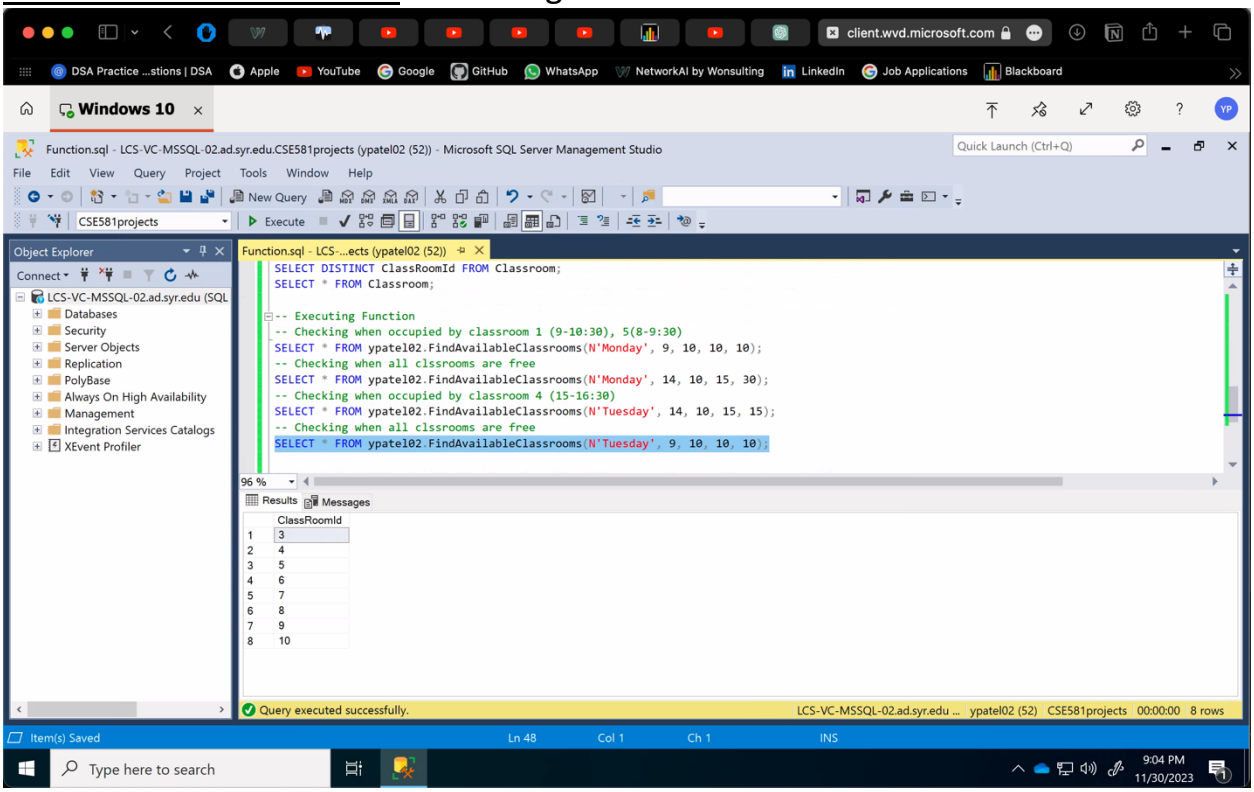
Function Execution Case 2 : Checking when all classrooms are free.



Function Execution Case 3 : Checking when occupied by classroom 4 (15-16:30)

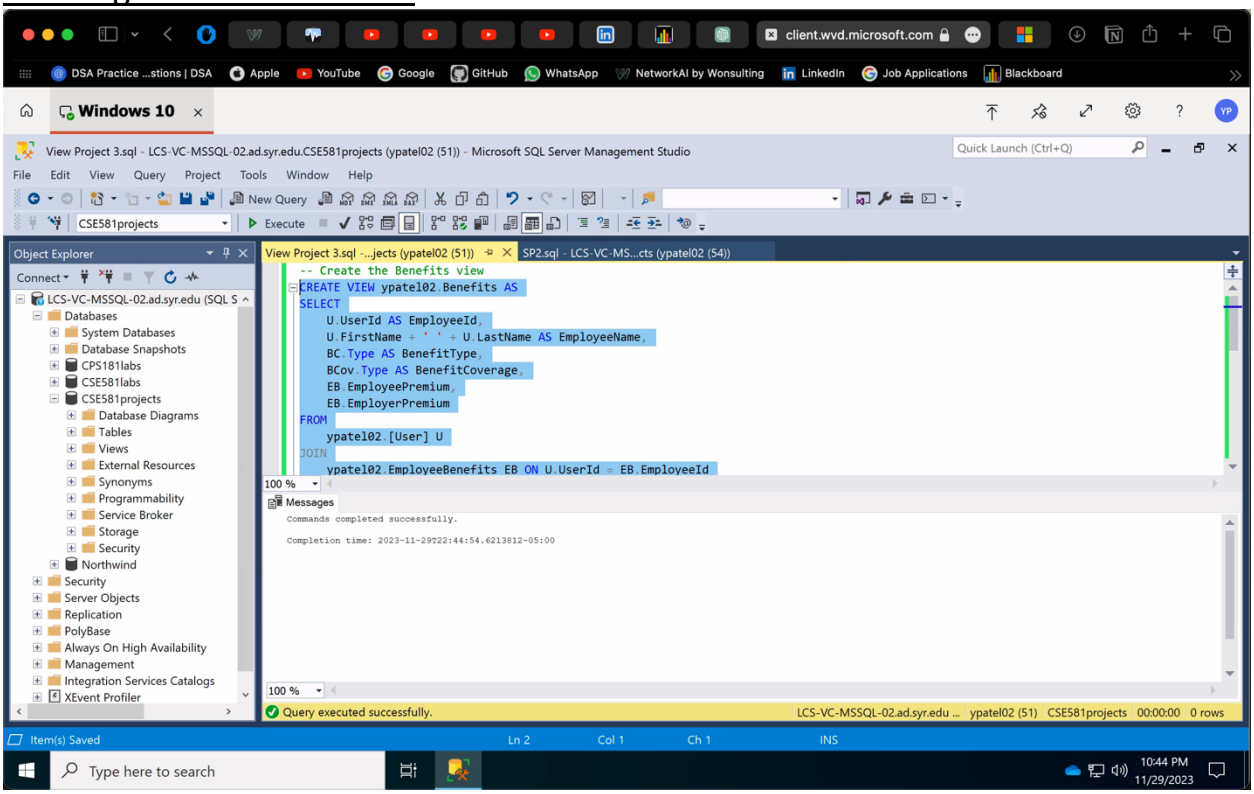


Function Execution Case 4 : Checking when all clssrooms are free.

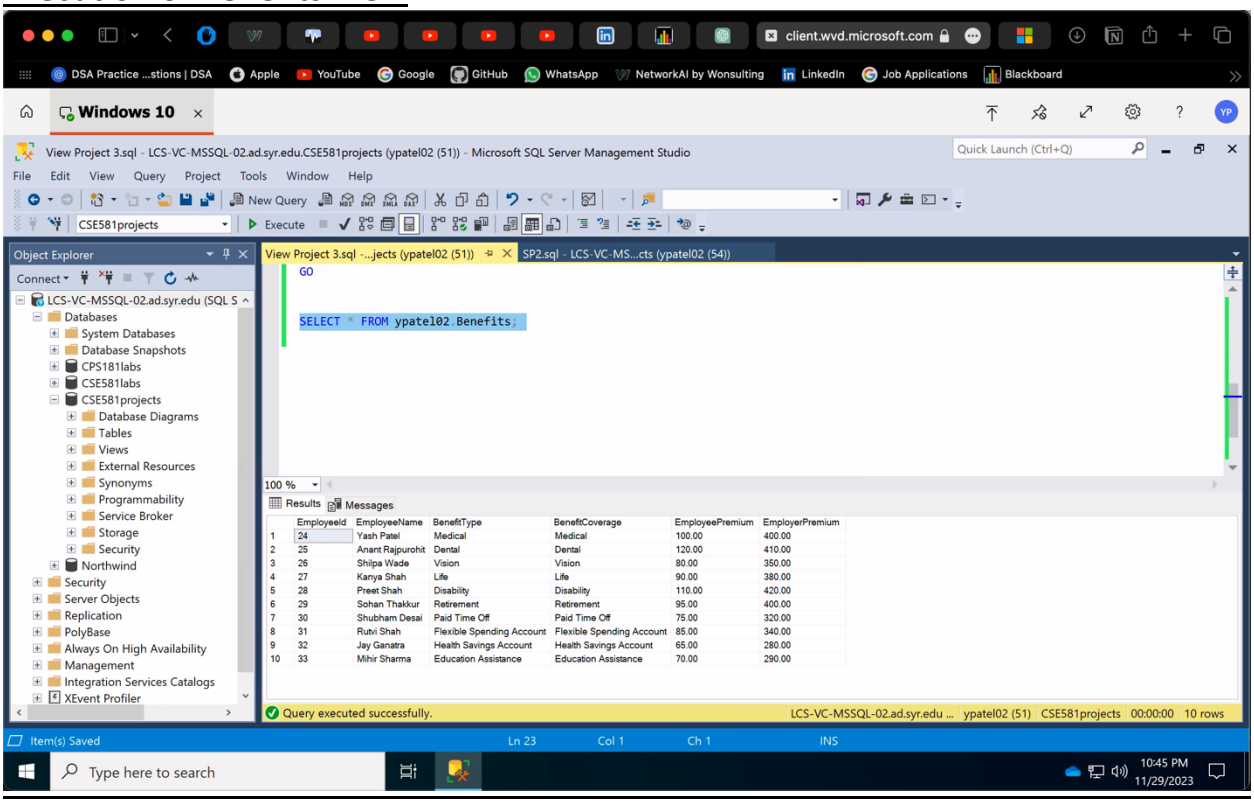


You will also *create a view* (named as "Benefits") (10% points) which can be viewed by "Graders" (GRANT SELECT ON SCHEMA::[yourSchema] TO Graders;) that shows every employee's name, ID, benefit's type, benefit coverage, employee premium and employer premium.

Creating View as Described



Execution of Benefits View



GRANTING ACCESS TO GRADERS