SQL Project: Student Course Management System

1. Introduction

The "Student Course Management System" project is designed to develop a database that manages student information, course details, enrollments, and instructors. This report documents the entire process, from database creation to executing SQL queries that handle data manipulation and retrieval. The report also includes visual outputs for each query to demonstrate the results.

2. Project Overview

Project Overview:
This project involves creating a database for managing student courses. The database includes tables for students, courses, enrollments, and instructors, and covers SQL topics such as selection, filtering, aggregation, joins, and subqueries.

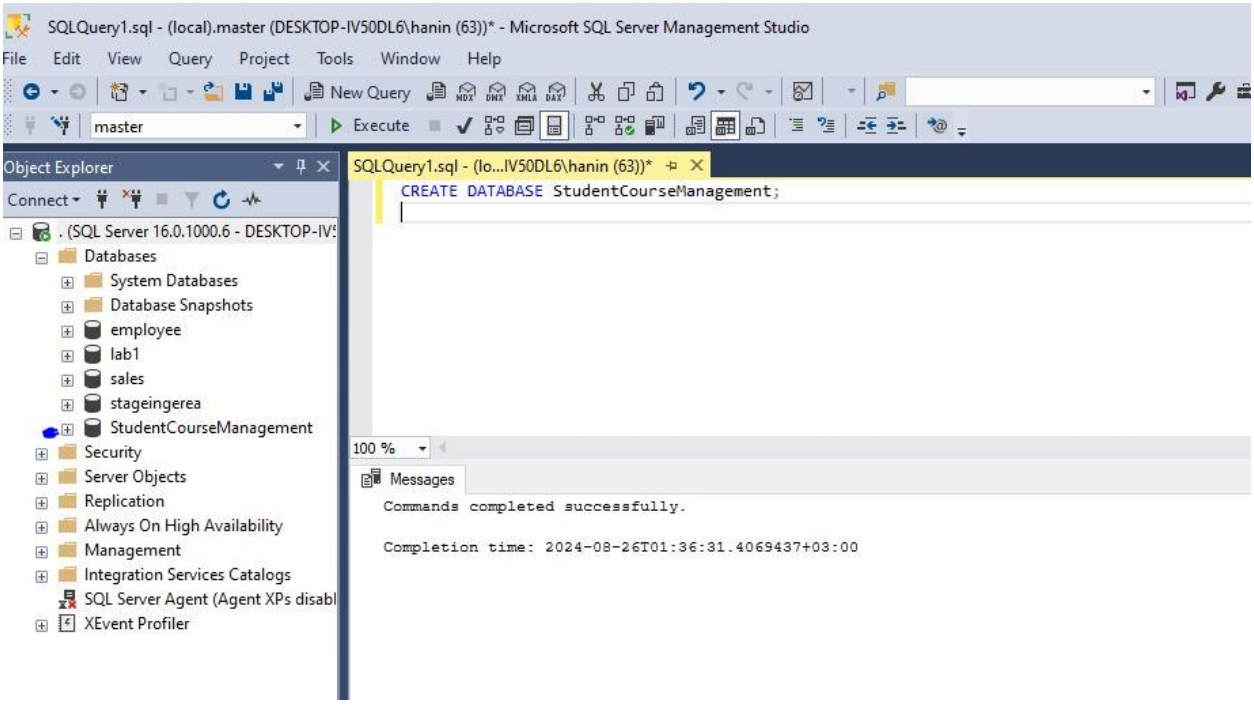
Project Requirements:

- **Database Setup:**
  Create a database named Student Course Management.
- **Table Creation:**
  Create the following tables with appropriate data types and constraints:
  - **Students:** student_id, first_name, last_name, email, date_of_birth
  - **Courses:** course_id, course_name, course_description
  - **Instructors:** instructor_id, first_name, last_name, email
  - **Enrollments:** enrollment_id, student_id, course_id, enrollment_date
- **Insert Sample Data:**
  Insert at least 10 students, 5 courses, 3 instructors, and 15 enrollments.
- **Basic Queries:**
  Select all students, courses, and enrollments with student and course names.

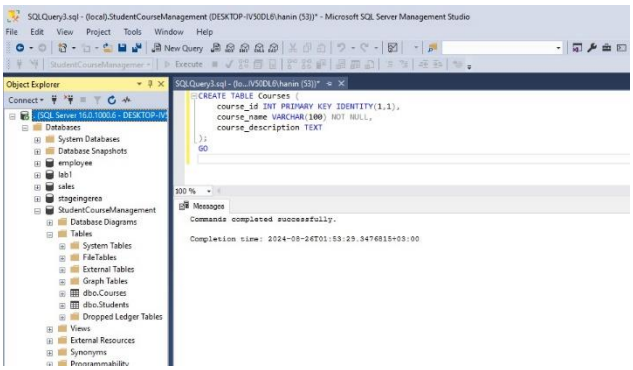
- **Advanced Queries:**
  Perform specific selections, updates, and deletions, calculate averages, and group results.
- **Join Queries:**
  Execute joins to retrieve related data from multiple tables.
- **Subqueries and Set Operations:**
  Perform subqueries, find specific data using conditions, and combine results.
- **Functions and Stored Procedures:**
  Create stored procedures and functions to manage database operations.
- **Aggregate Functions and Grouping:**
  Calculate totals, averages, and other aggregations.
- **Additional Tasks:**
  Use aliases, CASE, EXISTS, and comments to enhance SQL readability and functionality.

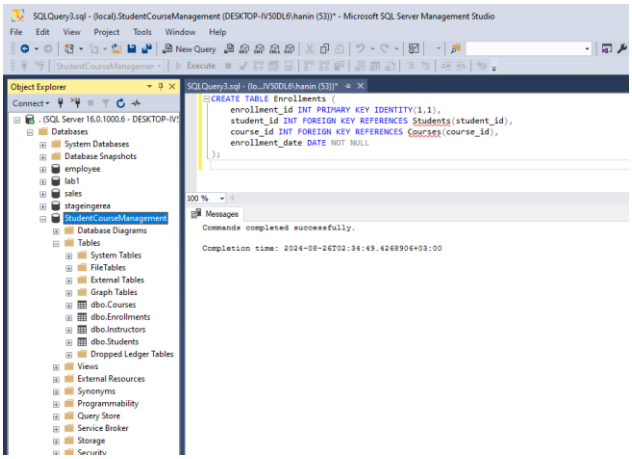
* Database Setup and Table Creation

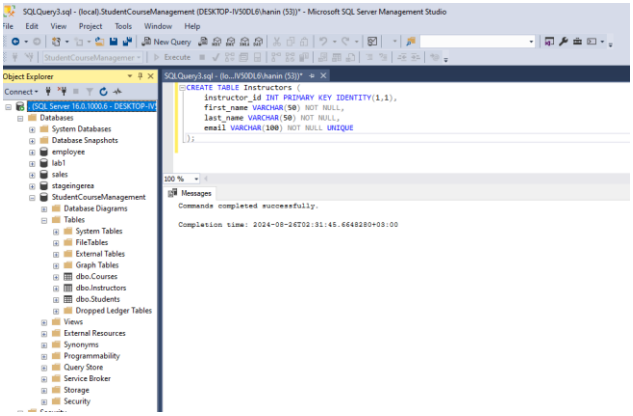
Database Setup

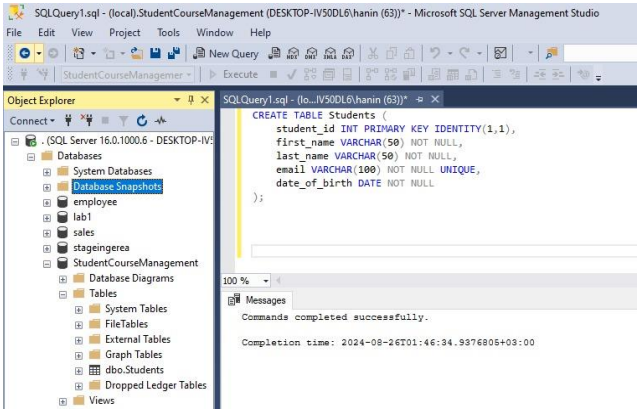


Tables Creation

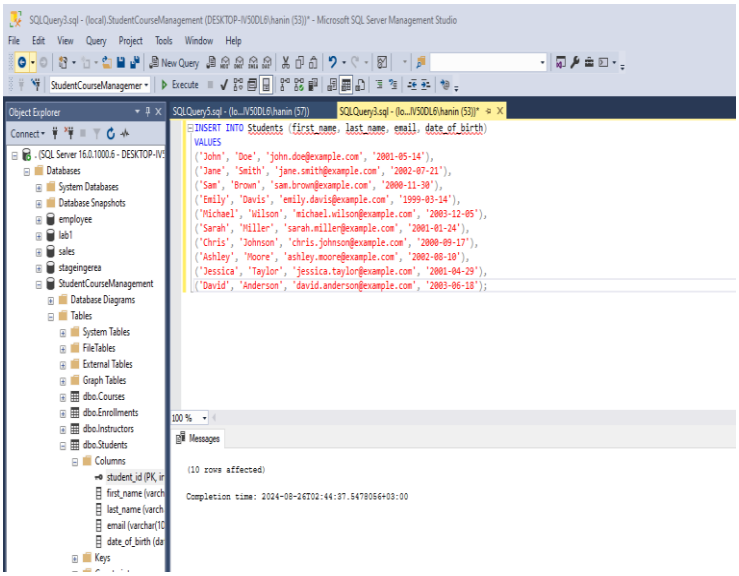
- 

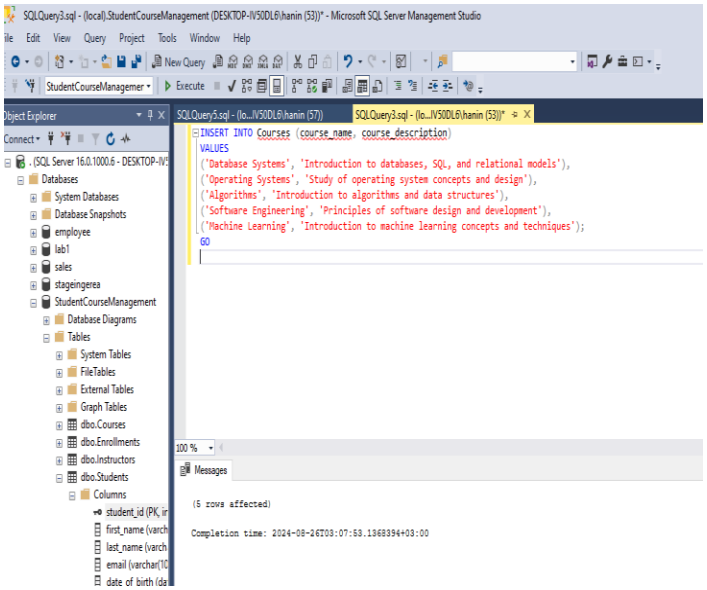


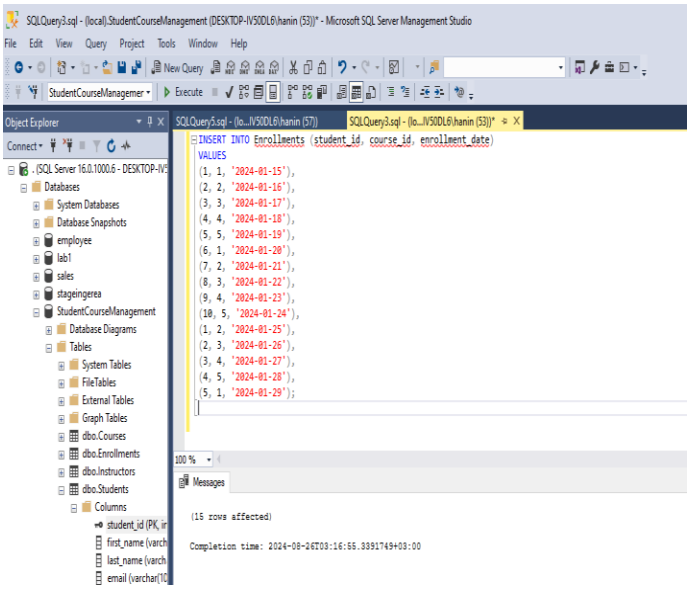


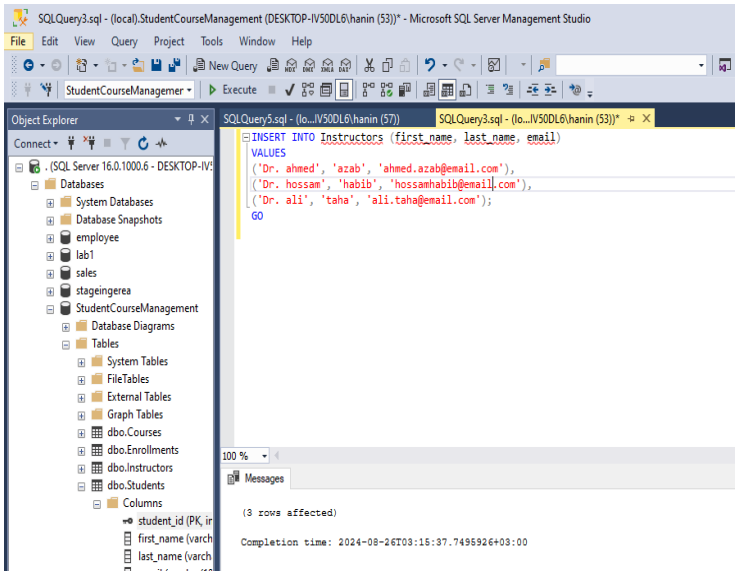


*Inserting Data to tables



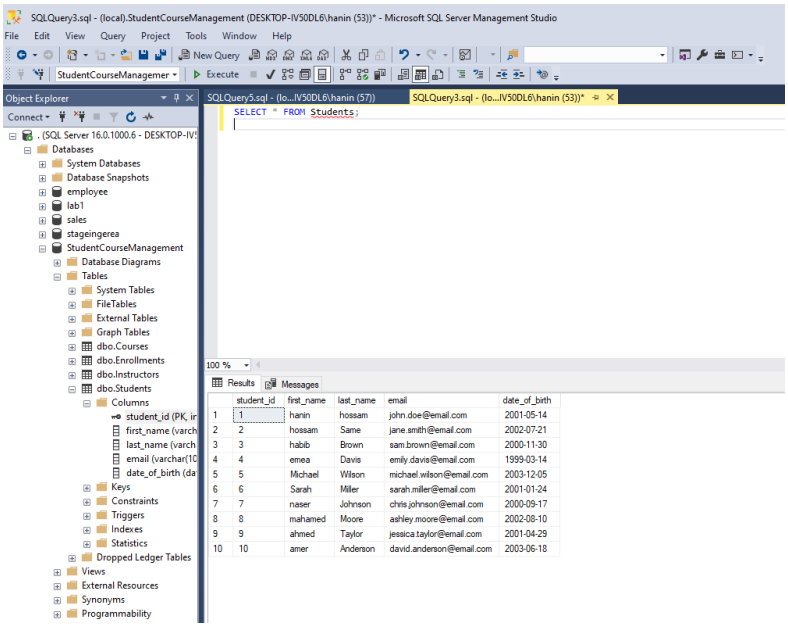




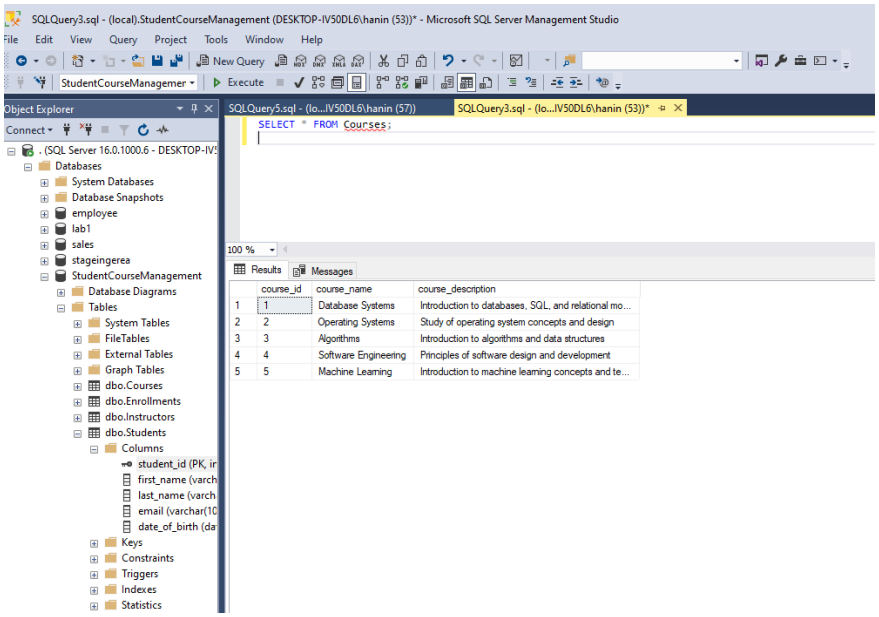


* Executing Basic Queries

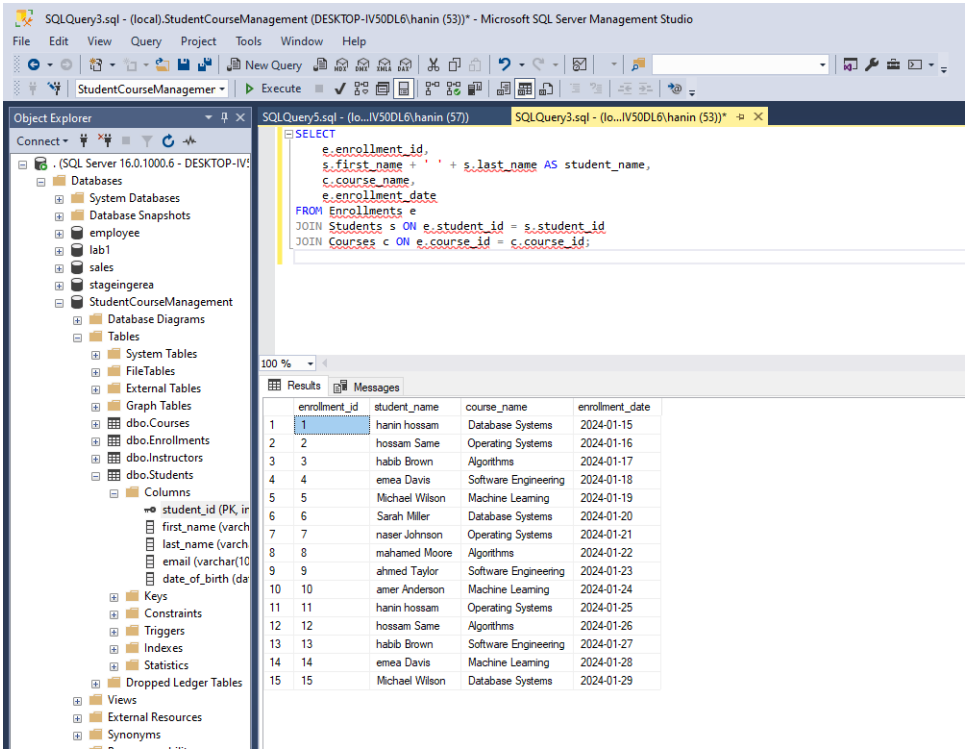
- **Select all students.**



- **Select all courses.**

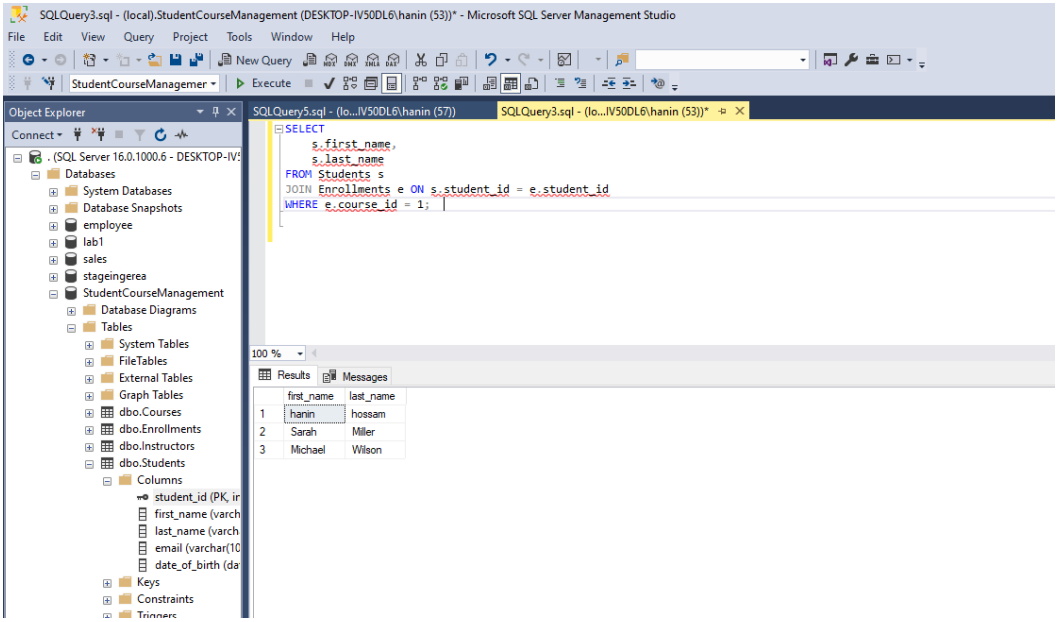


- **Select all enrollments with student names and course names.**

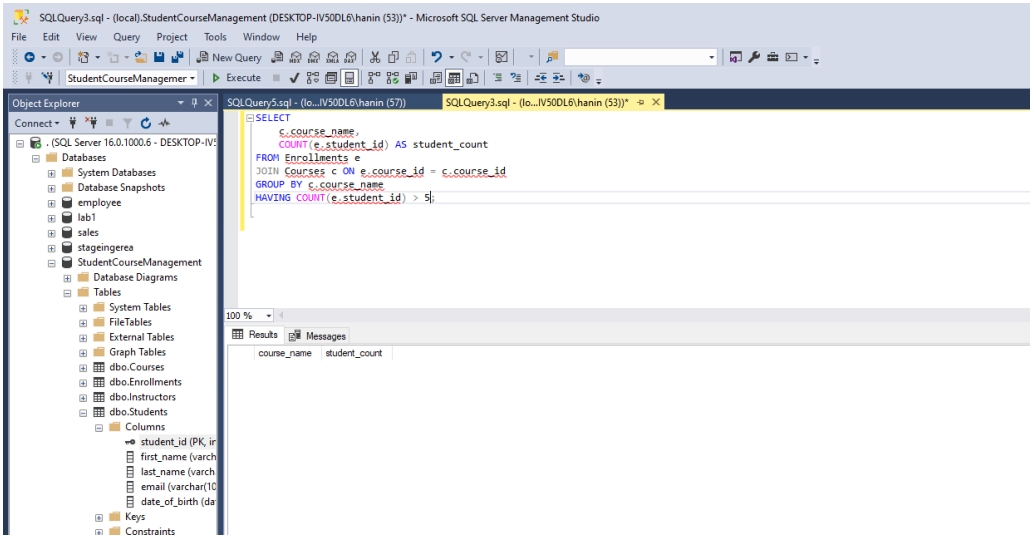


*Executing Advanced Queries

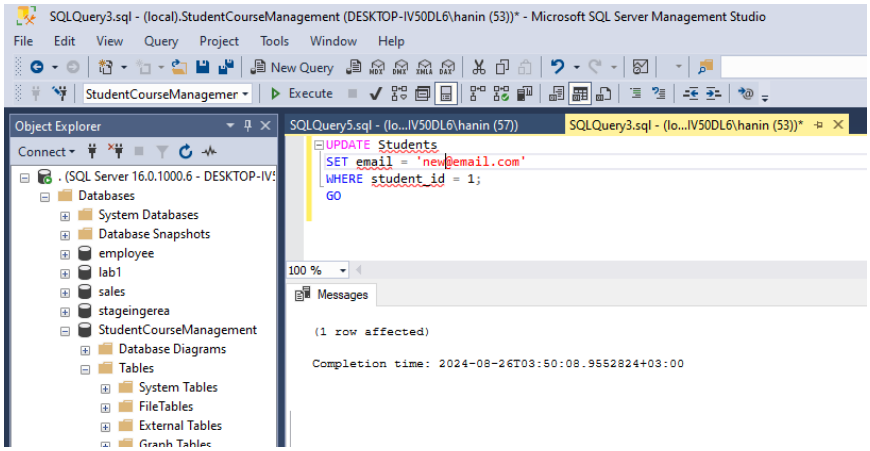
- **Select students who enrolled in a specific course.**



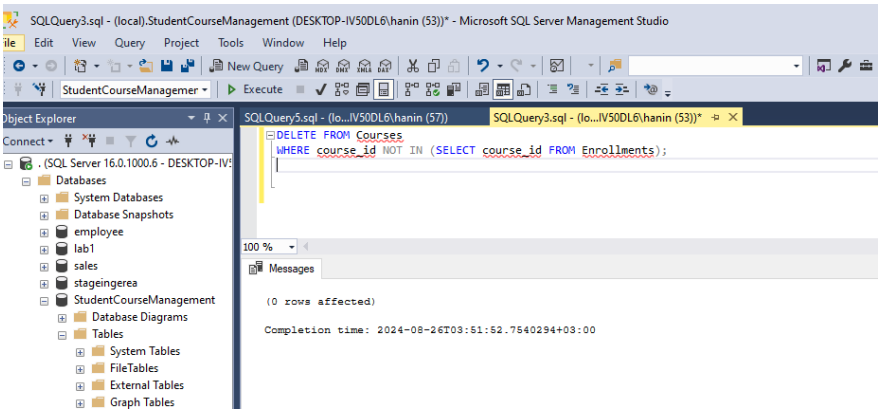
- **Select courses with more than 5 students.**



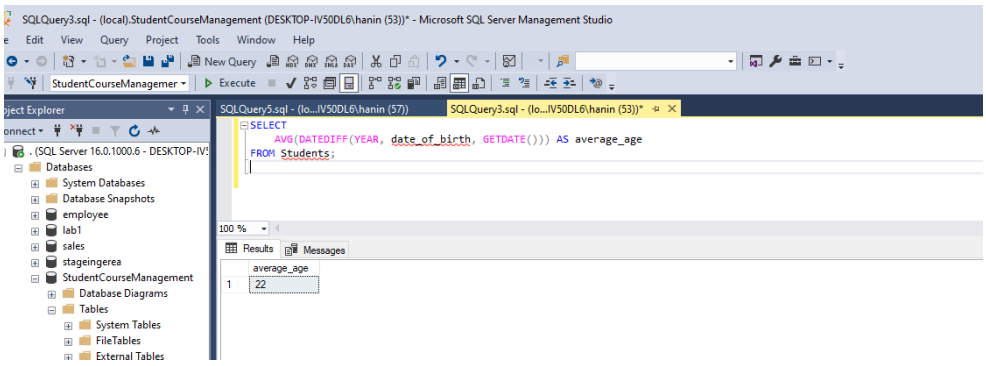
- **Update a student's email.**



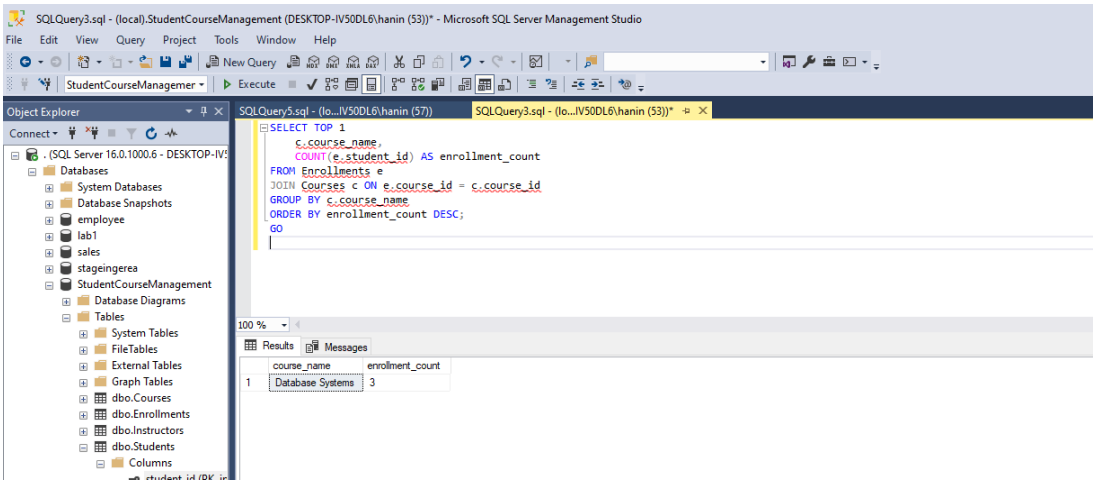
- **Delete a course that no students are enrolled in.**



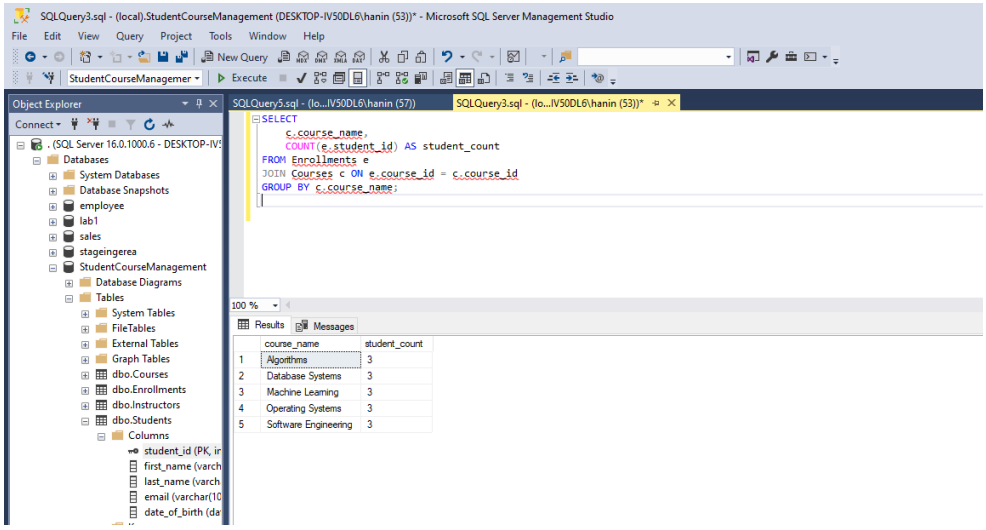
- **Calculate the average age of students.**



- **Find the course with the maximum enrollments.**

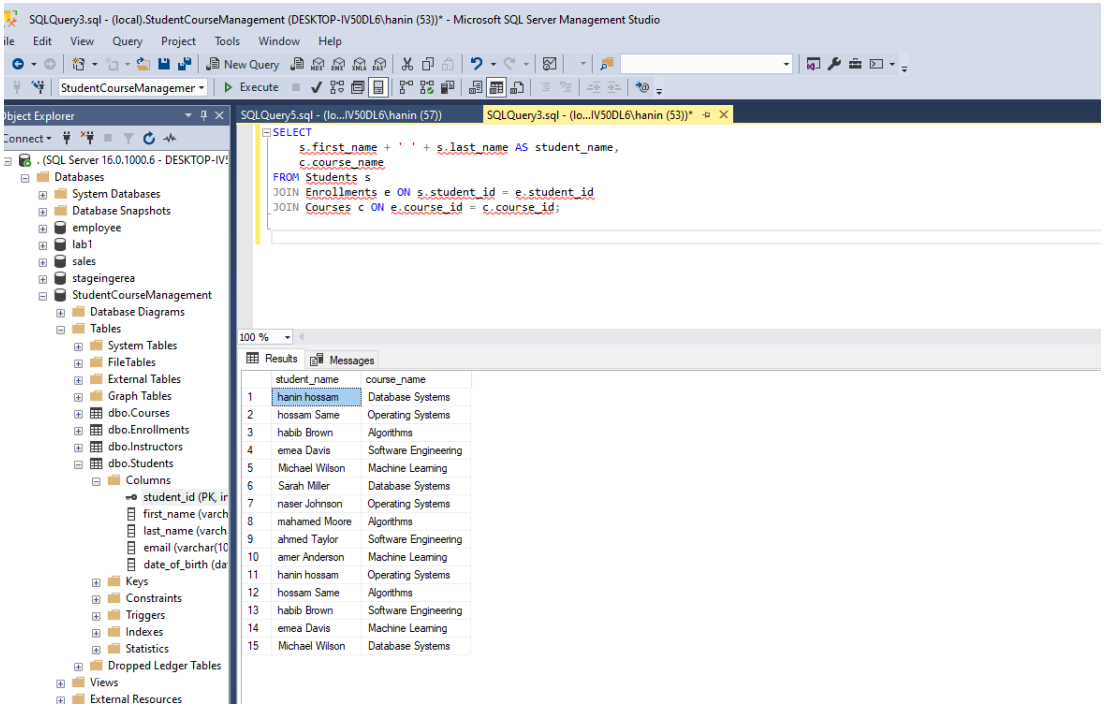


- **List courses along with the number of students enrolled (use GROUP BY).**

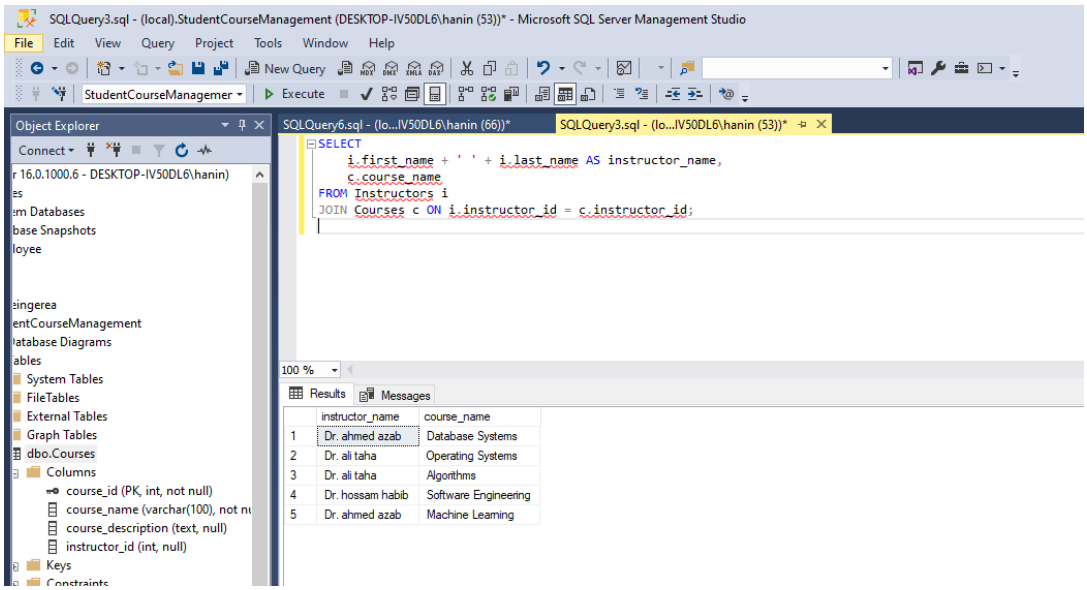


*Join Queries

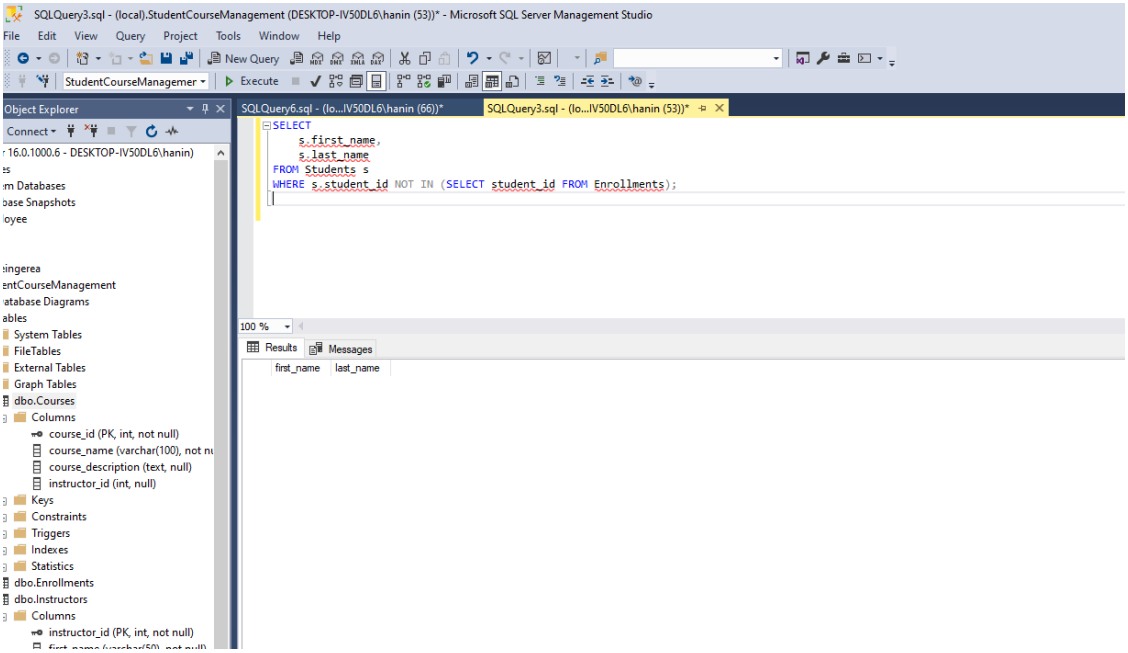
- **Select all students with their enrolled courses (use JOIN).**



- **List all instructors and their courses.**

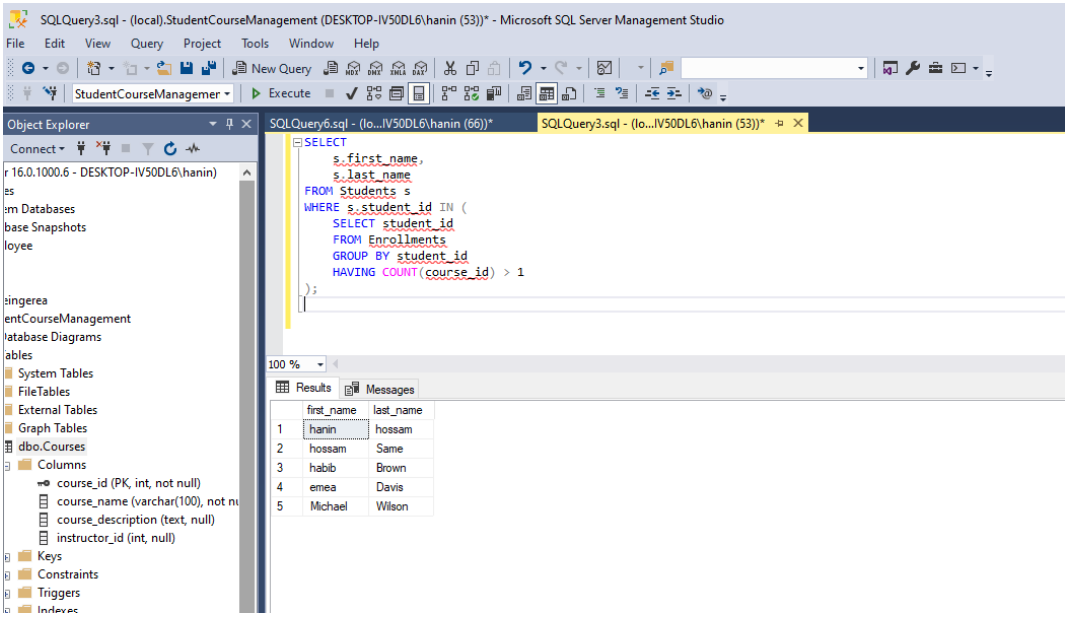


- **Find students who are not enrolled in any course.**

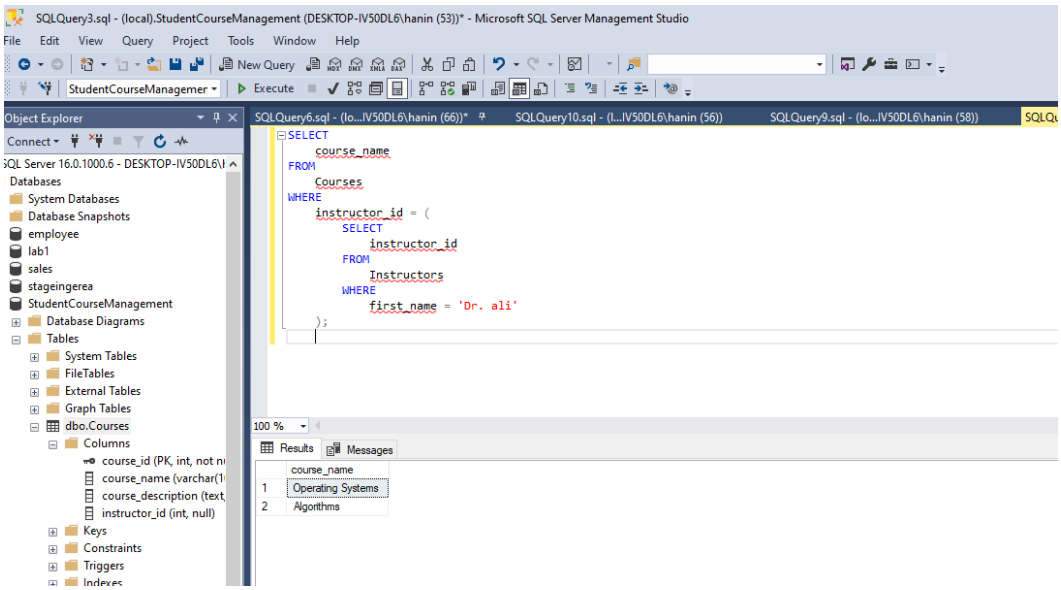


Subqueries and Set Operations

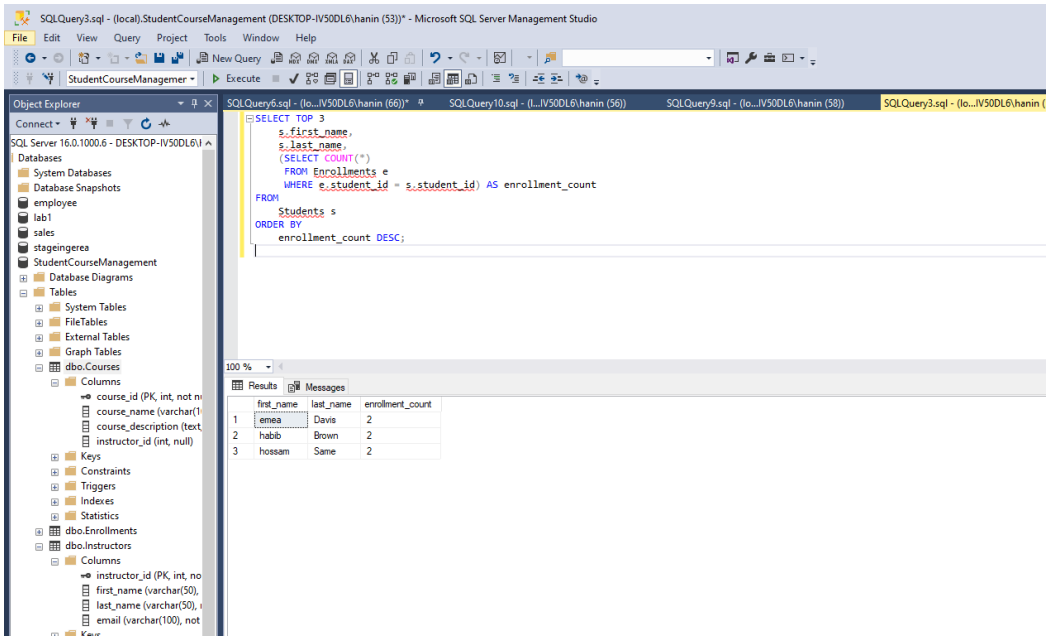
- **Select students enrolled in more than one course.**



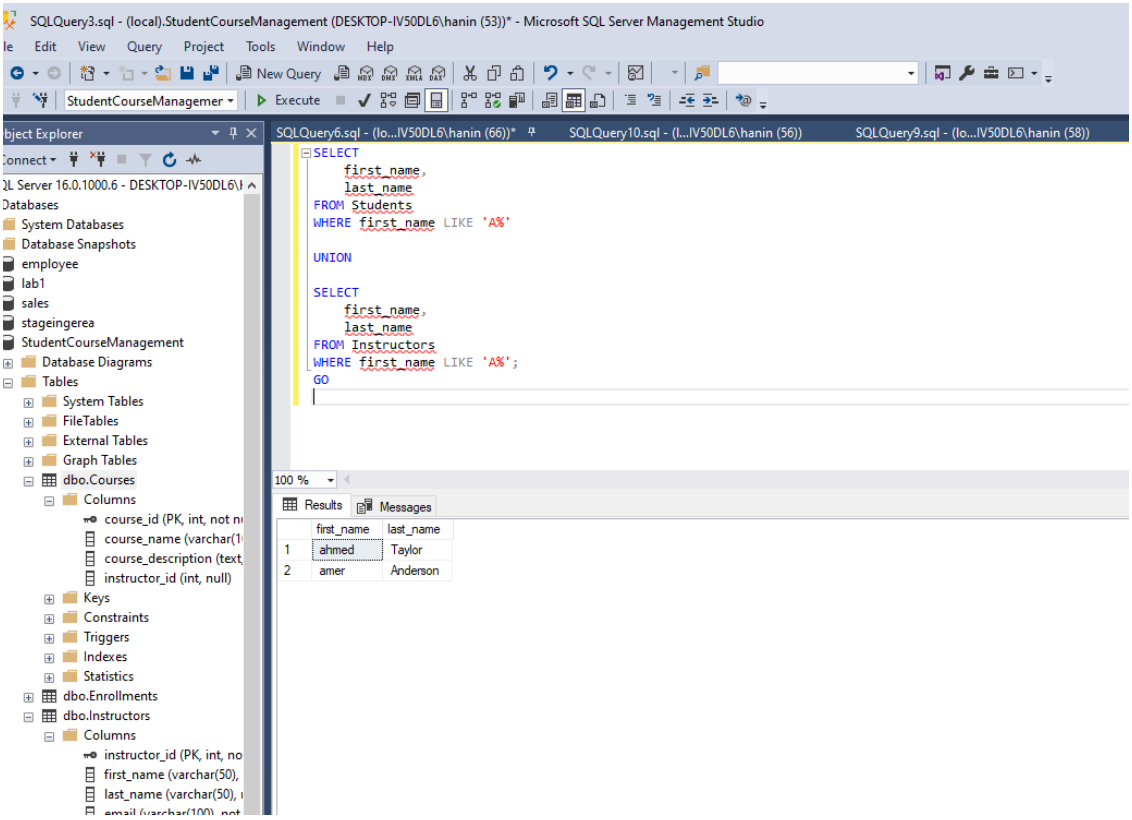
- **Find courses taught by a specific instructor.**



- **Select the top 3 students with the most enrollments.**

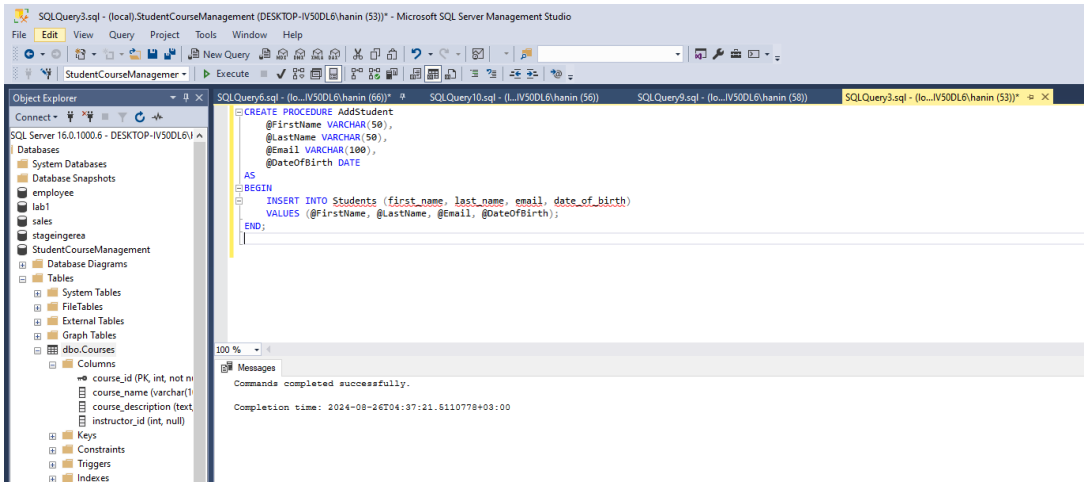


- **Use UNION to combine results of two different SELECT queries.**

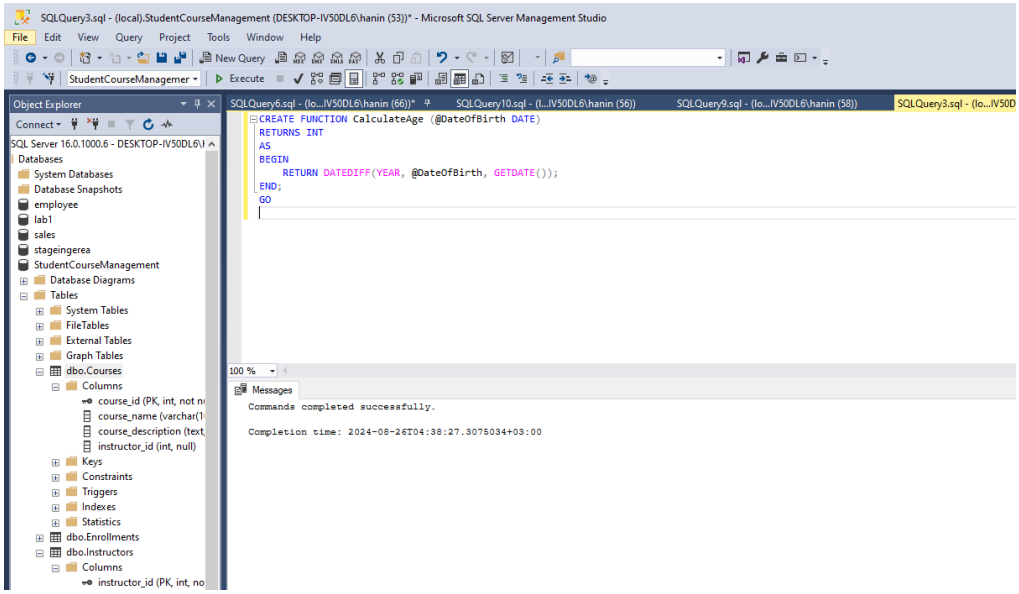


*Functions and Stored Procedures

- **Create a stored procedure to add a new student.**

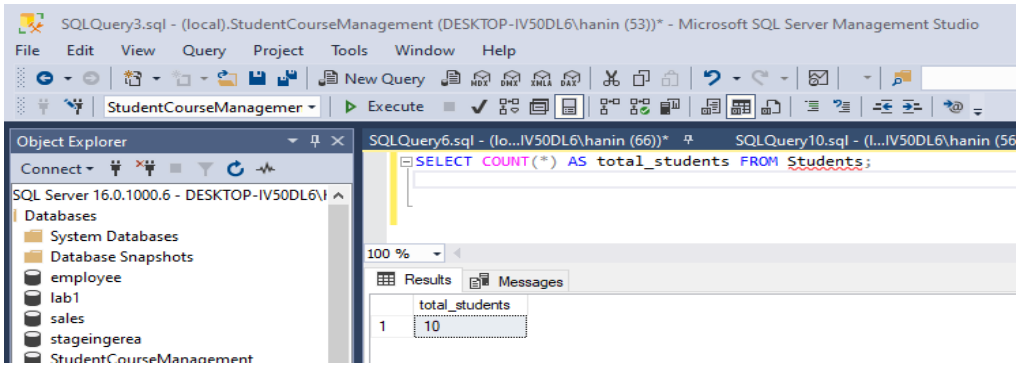


- **Create a function to calculate the age of a student based on their date of birth.**

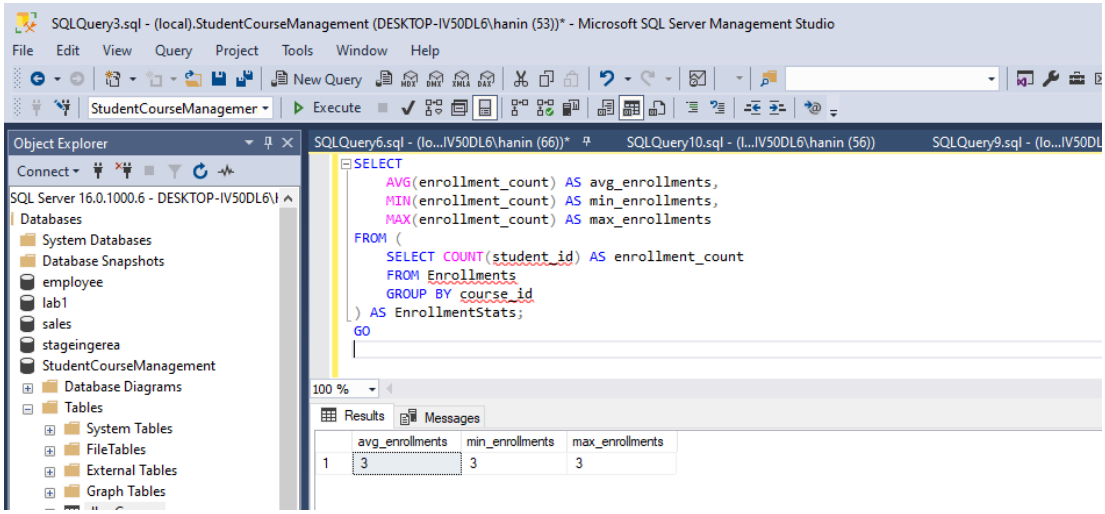


*Aggregate Functions and Grouping

- **Calculate the total number of students.**

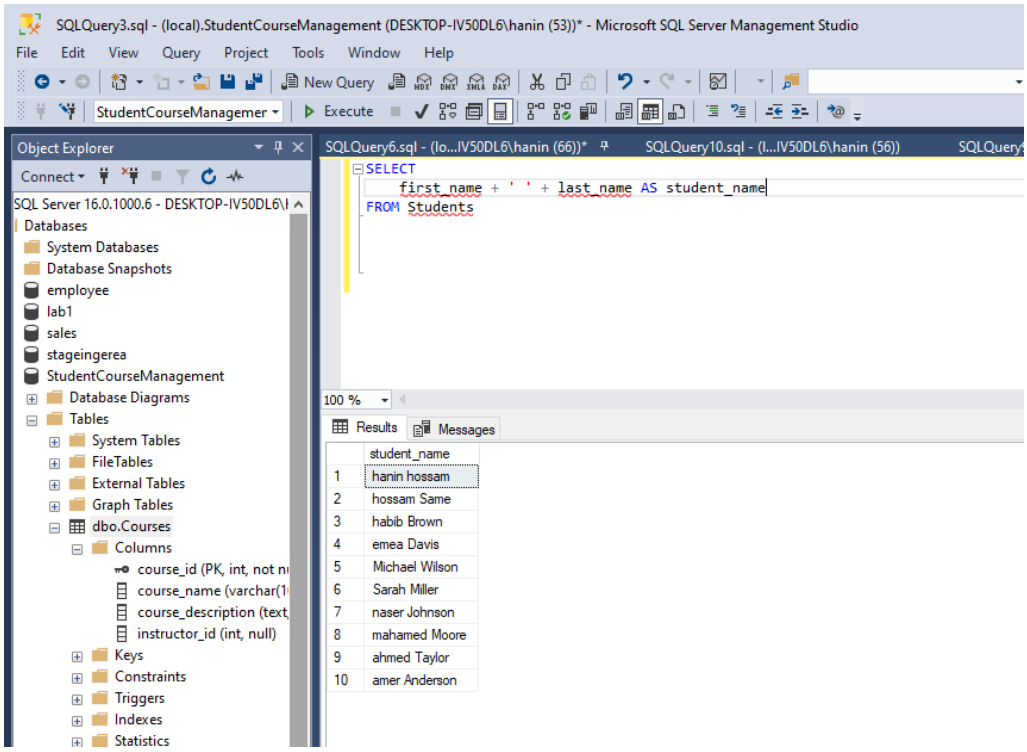


- **Calculate the average, minimum, and maximum number of enrollments per course.**

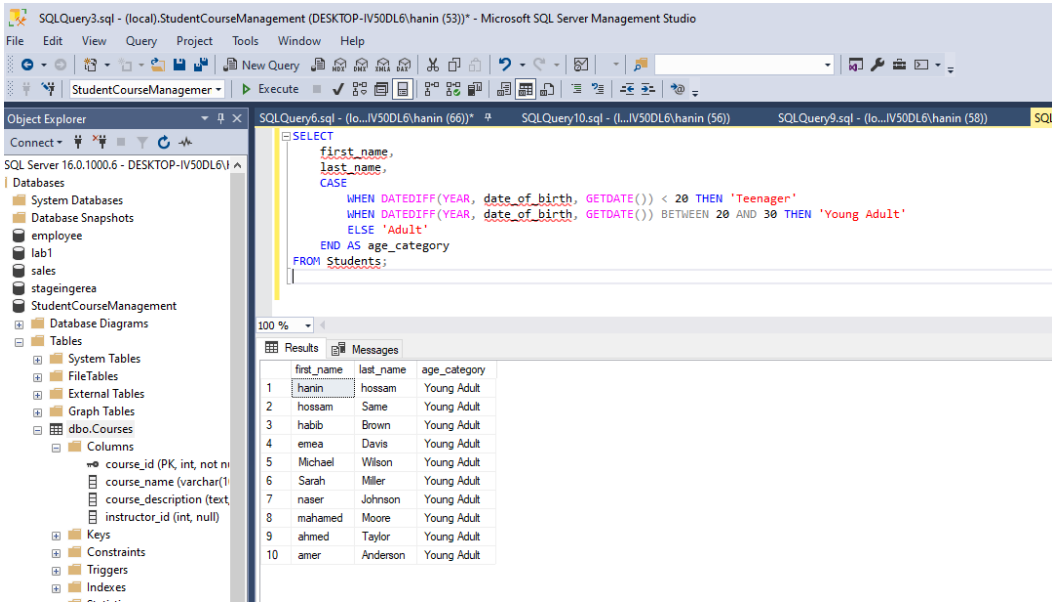


*Additional Tasks

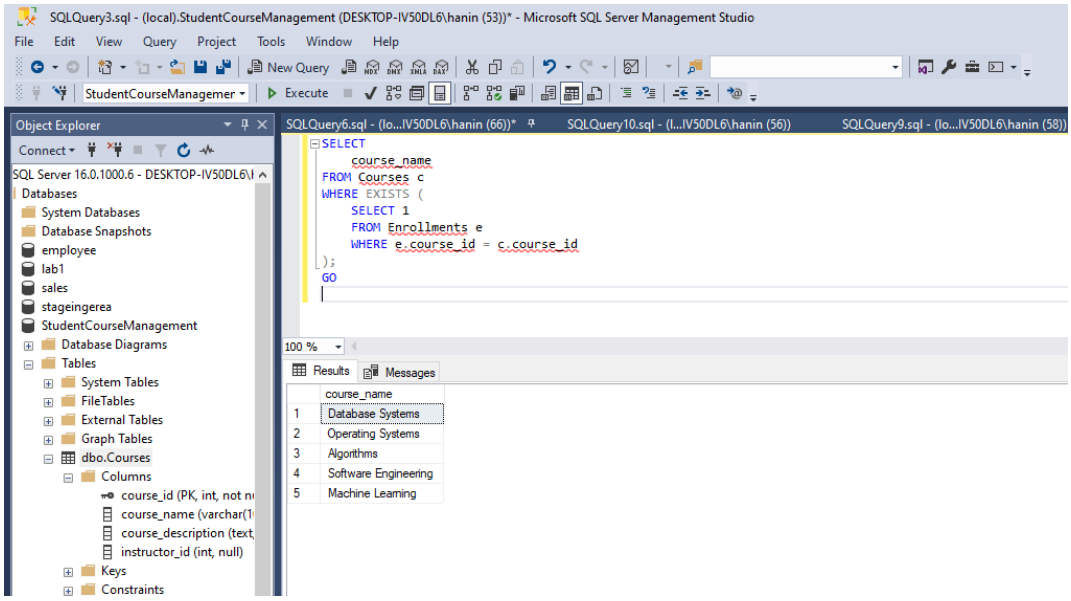
- **Create aliases for complex column names.**



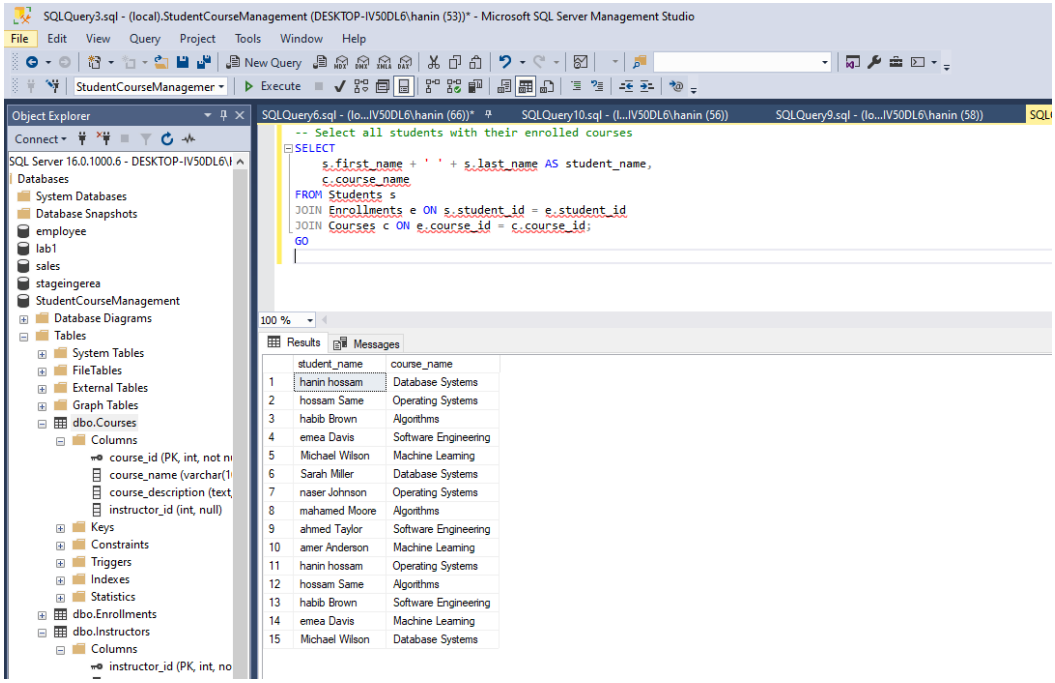
- **Use CASE to categorize students based on their age.**



- **Use EXISTS to find courses with at least one enrolled student.**



- **Create comments in SQL for clarity.**



4. Conclusion

This project provided a practical application of SQL in managing a student course database. By designing the database, creating tables, and writing various SQL queries