CS 4347.005 - Database Systems

Final Report Deliverable 1

Student Record Keeping System

Mia Sorola Yoshida, Abbas Khawaja, Vaishnavi Josyula, Anveetha Suresh, Justin Lu, Ahimsa Yukta, Abhishek Madhavan, Shubham Patel, Paribesh Upreti

0. Description:

- *Title*: Student Record Keeping System
- *Github Link*: https://github.com/justinlu200014/CS-4347-Group-6-project
- *Group Members:*
  - Paribesh Upreti - pxu200001@utdallas.edu
  - Abhishek Madhavan - axm200159@utdallas.edu
  - Anveetha Suresh - axs220399@utdallas.edu
  - Mia Sorola Yoshida - mas220015@utdallas.edu
  - Vaishnavi Josyula - vxj210016@utdallas.edu
  - Abbas Khawaja - aik210000@utdallas.edu
  - Justin Lu - jjl200014@utdallas.edu
  - Shubham Patel - shp200002@utdallas.edu
  - Ahimsa Yukta - axy210017@utdallas.edu
- *Delegation of tasks for the overall project*:
  - Paribesh: Frontend
    - UI/UX design: Work on the wireframes and prototypes using Figma and create user flows for authentication and form submissions.
    - Develop landing pages, home pages, and static content pages.
    - Implement student registration and data entry forms with appropriate input fields. Add search and filter components.
  - Justin: Frontend
    - Define the design system: typography, color palette, button styles, form elements, etc.
    - Implement global styles using CSS/Tailwind CSS to ensure UI consistency.
    - Develop header, sidebar, and footer components for reuse across multiple pages.
  - Shubham: Frontend
    - Implement client form validation for login services.
    - Collaborate with the backend team to fetch data dynamically.
    - Create a navigation menu for quick access to student records, courses, and reports.
  - Vaishnavi: Frontend
    - Design a clean dashboard interface to display key student statistics and actions.
    - Develop client-side routing logic for pages.
    - Build the student profile page to display personal details and course enrollments.
  - Abhishek: Backend
    - Implement logic to manage student records, ensuring data validation and consistency.
    - Set up role-based access control to manage permissions for different user types.
    - Collaborate with the frontend team to ensure smooth data integration and consistent workflows.
  - Anveetha: Backend
    - Write scripts to import/export bulk student data.
    - Ensure data consistency by setting up constraints like foreign keys and unique IDs.

        - Create the relational model for the record keeping system database.
    - Mia: Backend
        - Implement stored procedures and triggers to automate actions (e.g., log changes).
        - Set up backup strategies and write scripts to automate database backups.
        - Create the EER diagram for the record keeping system database.
    - Abbas: Backend
        - Create SQL queries to handle student record creation, updates, and deletions.
        - Set up database indexing to optimize search and filter operations.
        - Create views and work on DDL components of the database.
    - Ahimsa: Backend
        - Develop reports generation scripts to export data summaries (e.g., student grades).
        - Configure transaction handling to ensure atomic operations on student records.
        - Monitor query performance and optimize for better execution times.
- *Delegation of tasks for this report:*
    - Paribesh: worked on section 3.3
    - Justin: worked on section 4 and the Github repository
    - Shubham: worked on section 3.1
    - Vaishnavi: worked on sections 0 and 1
    - Abhishek: worked on section 0
    - Anveetha: worked on section 3.2
    - Mia: worked on section 3.1
    - Abbas: worked on section 3.4
    - Ahimsa: worked on section 2
- *Motivation:*
    - Designing a system to help college students manage various aspects of their lives can significantly improve their efficiency and reduce stress.
    - Our design will be used at U.S. universities to simplify administrative processes, benefiting students and institutions by providing student record management solutions.
    - This project has a personal connection to us, as we are college students who deal with the same information on a regular basis. Class schedules, club meetings, tuition, housing, and meal plans are all topics that are familiar to us.
- *Project timeline:*
    - 1. Motivation and Planning (9/14 - 9/17):
        - We identified the need for a comprehensive Student Record Keeping System to help students manage various aspects of their lives (schedules, finances, housing, etc.). The initial planning phase involved discussions around the project's scope and setting clear goals for how the system would be implemented and used.
    - 2. Task Delegation (9/18 - 9/21):
        - The team divided responsibilities across frontend and backend tasks, ensuring a clear focus for each member:
            - Frontend design and development (UI/UX, form validation, routing).
            - Backend database design (data validation, access control, query optimization, stored procedures).

        - Report writing was also delegated among the team members, as each person took responsibility for specific sections.
    - 3. Design Phase (9/22 - 9/26):
        - The EER Conceptual Data Model Design was completed, defining entities such as Student, Course, Assignment, Housing, and Finances.
        - The relationships between entities were mapped out (e.g., a student managing finances or attending events).
    - 4. Relational Data Model Design (9/27 - 10/4):
        - The EER model was mapped to a relational database model, utilizing techniques such as disjoint subclassing for handling different assignment types (exam, homework, quiz).
    - 5. Database Creation and Population (10/5 - 10/11):
        - We created the database and populated it with sample data for testing purposes. This ensured proper population and performed operations like queries, insertions, deletions, and updates to validate the database structure.
    - 6. Query Execution and Testing (10/12 - 10/15):
        - SQL queries were executed for various operations (listing, inserting, deleting, updating data) to ensure that the database functions as expected.
    - Report Finalization and Submission Preparation (10/16–10/19):
        - We wrote in our sections of the final deliverable 1 and reviewed them, ensuring all sections were complete and accurate in order to submit.

1. Introduction:

The project we are developing focuses on creating a comprehensive system to help college students manage multiple aspects of their campus life, including but not limited to academic schedules, finances, dining, housing, and extracurricular activities. As college students, we personally understand the challenges of organizing these essential components of student life. With multiple systems to navigate, from academic portals to dining and housing platforms, the process can become overwhelming.

Our goal is to develop a solution that mitigates this complexity by centralizing everything into a single, user-friendly platform. While many existing systems focus primarily on academic performance and grades, our platform goes beyond that. It aims to encompass the full spectrum of student life by including features that address extracurricular involvement, campus events, club participation, meal plans, housing information, and financial responsibilities.

Additionally, by implementing this program, we expect to contribute to the field of security by ensuring that sensitive information such as academic records, financial data, and housing agreements are protected through secure access control mechanisms, encryption, and privacy-focused design. Our contribution also includes enhancing data security in systems that manage multiple types of personal data, ensuring confidentiality, integrity, and availability.

Overall, we aim to provide students with a holistic tool that not only helps them manage their coursework but also promotes a balanced college experience. Through this platform, we aim to allow students to better organize their academic and non-academic lives, helping them stay on top of deadlines, manage extracurricular commitments, track dining and housing plans, and monitor financial obligations—all from one cohesive system.

2. Background and Related Work:

Background:

The Student Record Keeping System that we are developing is unique as it seeks to integrate various aspects of campus life—academic schedules, finances, dining, and housing—into a single cohesive platform. While universities and colleges offer systems that manage each of these components, they are typically separated into different portals. For instance, academic scheduling is handled independently through platforms like university portals, while financial accounts are often managed by bursars. Dining services and housing have their own systems, adding another layer of separation. These separate systems force students to juggle multiple logins and interfaces, dividing and complicating the overall management of their campus life.

For example, at The University of Texas at Dallas (UTD), students manage their academic responsibilities through the Galaxy portal, which handles class registration, schedules, and grades, while financial responsibilities such as tuition payments and fee management are processed through bursar account, housing services are handled via another distinct account, and dining plans are overseen by UTD Services, creating multiple systems that students need to navigate. This segmentation results in inefficient and unstreamlined access to critical student services.

Our platform aims to integrate these functionalities, combining academic, financial, housing, and dining services into a unified system. While the individual components already exist in various forms across different institutions, no system has yet to fully integrate them into a single platform for students. The strength of our implementation lies in its ability to consolidate these aspects, making it easier for students to manage all facets of their college experience from one centralized system with a singular database rather than different accounts and websites that each deal with different databases, a concept that has yet to be fully realized despite the availability of these individual services.

Related Work:

Our project builds on several important studies and systems in the areas of student record management. Below is a chronological list of these works, with a brief explanation of their contributions and how our work extends beyond them.

1. Eludire, A. (2011) [1] – The Design and Implementation of Student Academic Record Management System: This paper focuses on tracking student performance through grades and

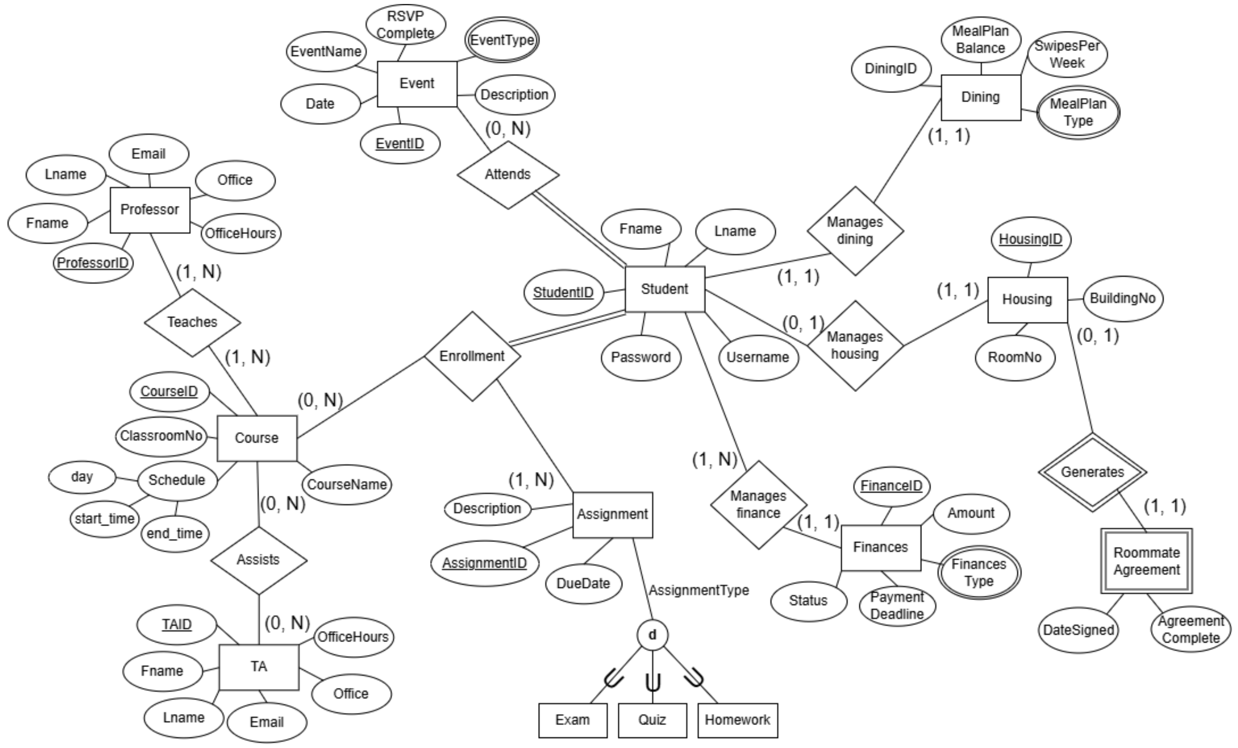
class schedules, with an emphasis on data accuracy and usability. It provides a system to manage academic records efficiently but does not extend to other aspects of student life. Our project builds on this idea by integrating non-academic services like financial payments, housing, and dining to create a more comprehensive platform. This added integration ensures students can manage all aspects of their college experience in one place. As a result, our solution not only tracks academic progress but also simplifies life beyond the classroom.

2. Bidyarthy, A. S., & Kumar, A. (2012) [2] – Student Database Management System: This paper automates the management of academic data, including grades, attendance, and course schedules. While it efficiently handles academic processes, its scope is restricted to academic records. Our project expands on this by integrating other areas of student life, such as housing assignments, dining plans, and financial payments. This holistic approach ensures students have a unified platform to manage both academic and non-academic tasks. With a broader scope, our system meets a wider range of student needs.

3. Tamboli, A. (2017) [3] – Institute Administration Automation and Student Database Management System: This paper emphasizes the automation of institutional processes, focusing on course registration, admissions, and student records through a software platform. Although it improves institutional efficiency, the system is limited to academic and administrative tasks. Our project extends the automation to areas like housing and dining, along with financial accounts, providing students with a seamless experience. Additionally, our platform includes personalized features, such as roommate agreements and housing costs, designed to address the specific needs of college students at institutions like UTD. This personalization makes our solution more relevant to the everyday lives of students.

3. Design & Implementation (Phase I):

3.1. EER Conceptual Data Model Design:



Entities, Attributes, Data Types:

- Student
  - StudentID INT
  - Fname VARCHAR(20)
  - Lname VARCHAR(20)
  - Username VARCHAR(20)
  - Password VARCHAR(20)
  - Note: The student is the one using the website to track their college life. So, we need to collect their login information so students can access the website based on their personal information. We don’t need to collect their email or phone numbers because this platform will be meant for students.
- Event
  - EventID INT
  - Date DATETIME
  - EventName VARCHAR(45)
  - RSVPComplete BIT
  - EventType VARCHAR(20)
    - Note: multivalued attribute (club, volunteering, etc)
  - Description VARCHAR(45)
- Course
  - CourseID INT
  - CourseName VARCHAR(45)
  - ClassroomNo VARCHAR(10)
  - Schedule: {day, start_time, end_time} DATETIME

        - Note: composite attribute
- Professor
    - ProfessorID INT
    - Fname VARCHAR(20)
    - Lname VARCHAR(20)
    - Email VARCHAR(45)
    - Office VARCHAR(45)
    - OfficeHours DATETIME
- TA
    - TAID INT
    - Fname VARCHAR(20)
    - Lname VARCHAR(20)
    - Email VARCHAR(45)
    - Office VARCHAR(45)
    - OfficeHours DATETIME
- Assignment (Supertype)
    - AssignmentID INT
    - DueDate DATETIME
    - Description VARCHAR(45)
    - AssignmentType (disjoint subclasses)
        - Exam
        - Homework
        - Quiz
- Finances
    - FinancesID
    - Amount DECIMAL
    - FinancesType VARCHAR(45)
        - (tuition, books, room and board, dining, etc)
        - Multivalued
    - PaymentDeadline DATETIME
    - Status BIT
        - (1- paid, 0 - unpaid)
- Dining
    - DiningID INT
    - MealPlanBalance DECIMAL
    - SwipesPerWeek INT
    - MealPlanType VARCHAR(45)
        - multivalued
- Housing
    - HousingID INT
    - BuildingNo VARCHAR(45)
    - RoomNo VARCHAR(45)
- RoommateAgreement (Weak entity of Housing)
    - DateSigned DATE

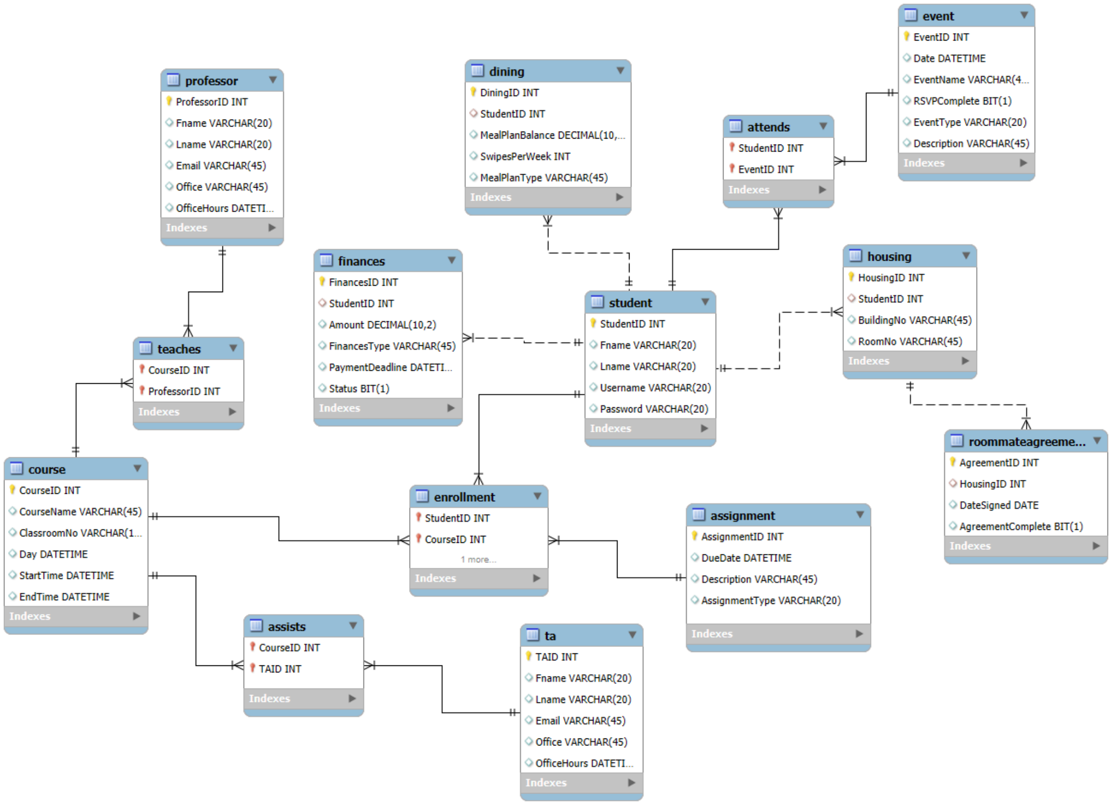
- AgreementComplete BIT
    - (0 - No, 1 - Yes)

Relationships:

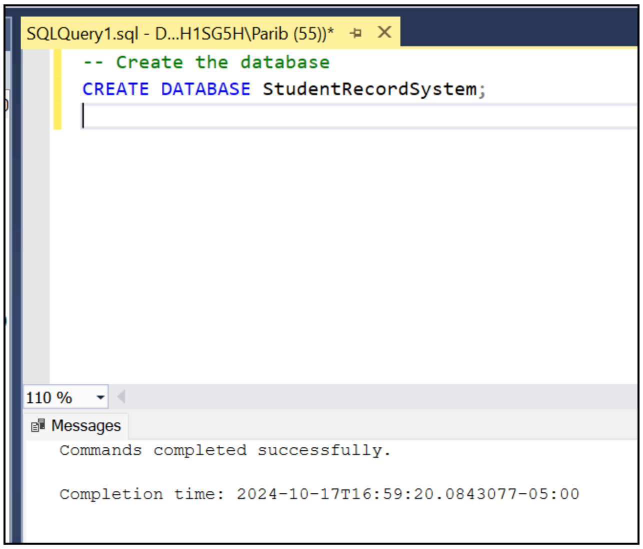
1. Attends: Student (Many) - Event (Many)
2. Teaches: Course (Many) - Professor (One)
3. Assists: Course (One) - TA (Many)
4. Enrollment: Student (Many), Course (Many), Assignment (Many)
5. Manages finance: Student (One) - Finances (Many)
6. Manages dining: Student (One) - Dining (One)
7. Manages housing: Student (One) - Housing (One)
8. Generates: Housing (One) - RoommateAgreement (One)
9. Super/Subclass relationship (disjoint) - Assignment superclass with exam, quiz, homework subclasses

3.2. Relational Data Model Design:

Note: To map an EER Model to a Relational Model, we chose Option 8C: Single relation with one type attribute which is defined by R. Elmasri and S. Navathe [4]. The superclass "Assignment" has an "AssignmentType" attribute to specify its subclasses: Exam, Homework, and Quiz.

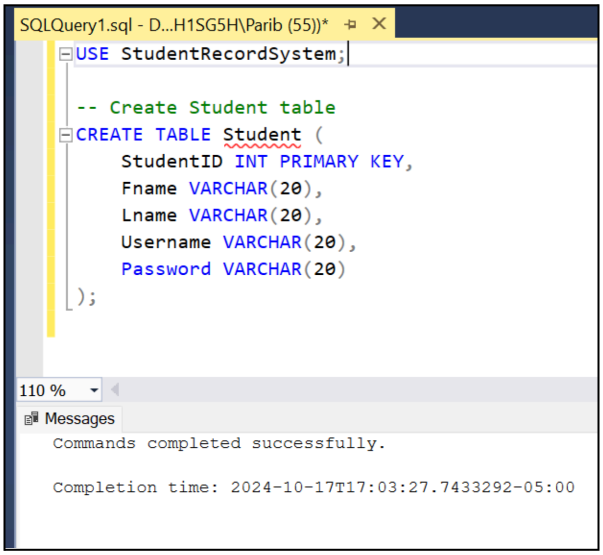


3.3. Create your Database and Populate:



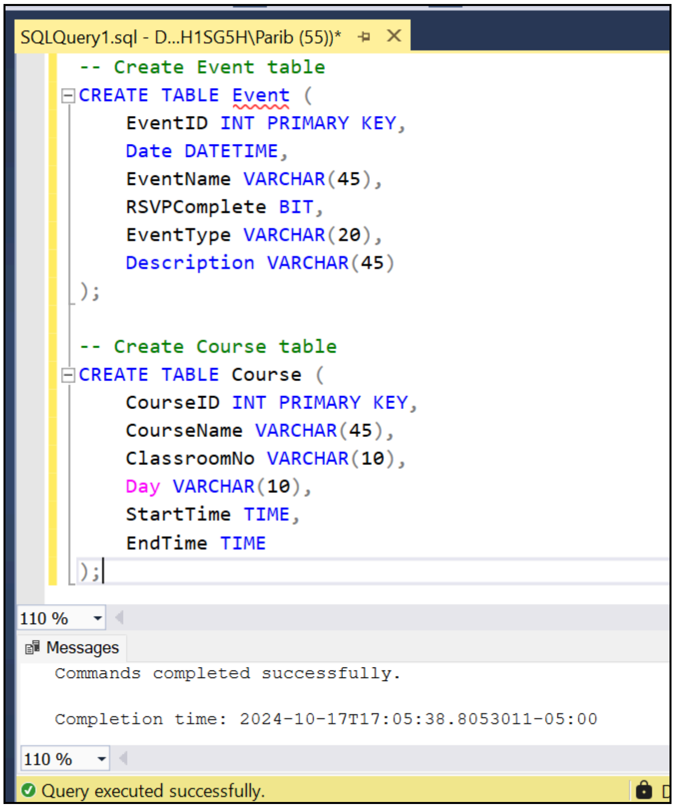
```sql
-- Create the database
CREATE DATABASE StudentRecordSystem;
```

```
Commands completed successfully.

Completion time: 2024-10-17T16:59:20.0843077-05:00
```



```sql
USE StudentRecordSystem;

-- Create Student table
CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    Fname VARCHAR(20),
    Lname VARCHAR(20),
    Username VARCHAR(20),
    Password VARCHAR(20)
);
```

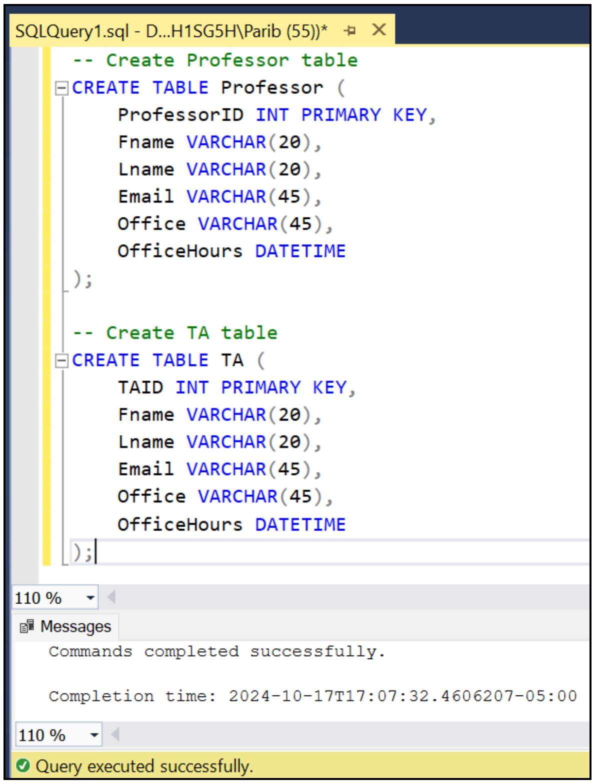
```
Commands completed successfully.

Completion time: 2024-10-17T17:03:27.7433292-05:00
```



```sql
-- Create Event table
CREATE TABLE Event (
    EventID INT PRIMARY KEY,
    Date DATETIME,
    EventName VARCHAR(45),
    RSVPComplete BIT,
    EventType VARCHAR(20),
    Description VARCHAR(45)
);

-- Create Course table
CREATE TABLE Course (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(45),
    ClassroomNo VARCHAR(10),
    Day VARCHAR(10),
    StartTime TIME,
    EndTime TIME
);
```

```
Commands completed successfully.

Completion time: 2024-10-17T17:05:38.8053011-05:00
```

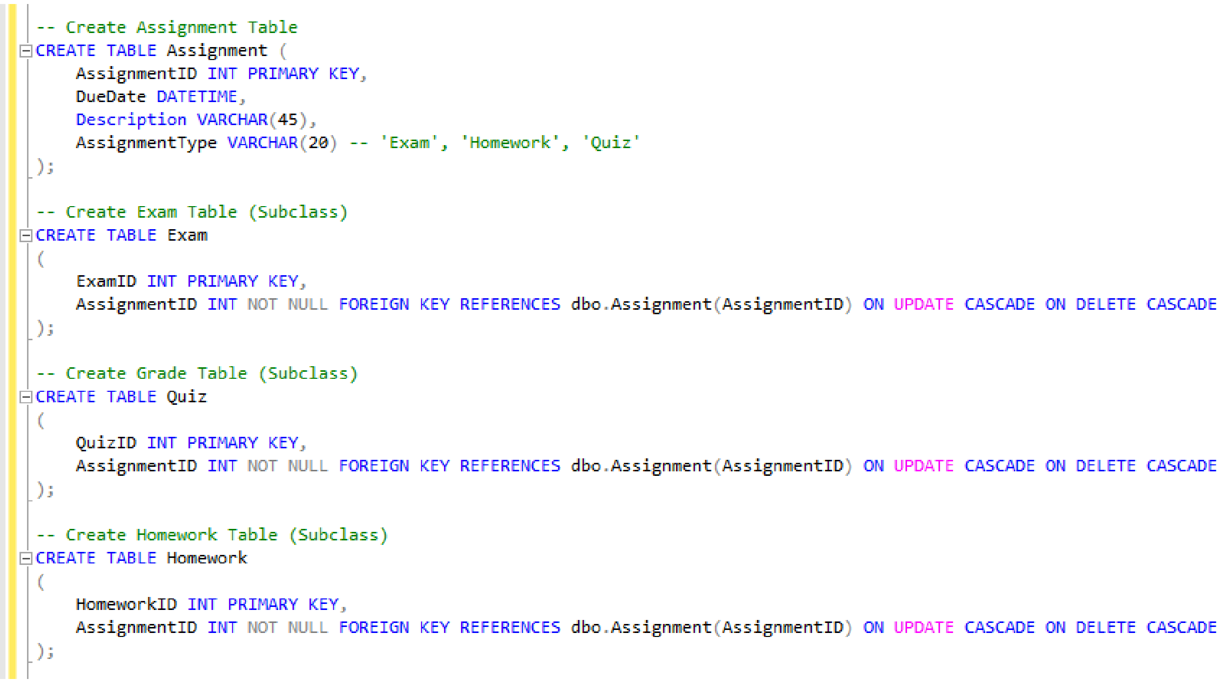
Query executed successfully.



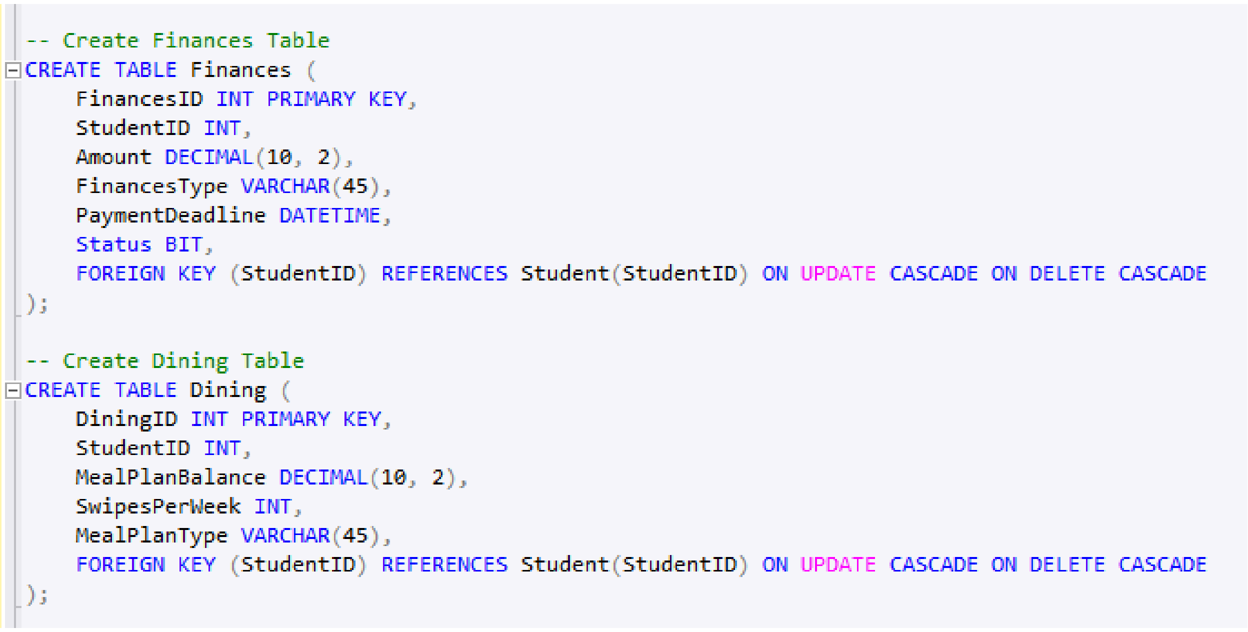
```sql
-- Create Professor table
CREATE TABLE Professor (
    ProfessorID INT PRIMARY KEY,
    Fname VARCHAR(20),
    Lname VARCHAR(20),
    Email VARCHAR(45),
    Office VARCHAR(45),
    OfficeHours DATETIME
);

-- Create TA table
CREATE TABLE TA (
    TAID INT PRIMARY KEY,
    Fname VARCHAR(20),
    Lname VARCHAR(20),
    Email VARCHAR(45),
    Office VARCHAR(45),
    OfficeHours DATETIME
);
```

```
Commands completed successfully.

Completion time: 2024-10-17T17:07:32.4606207-05:00
```

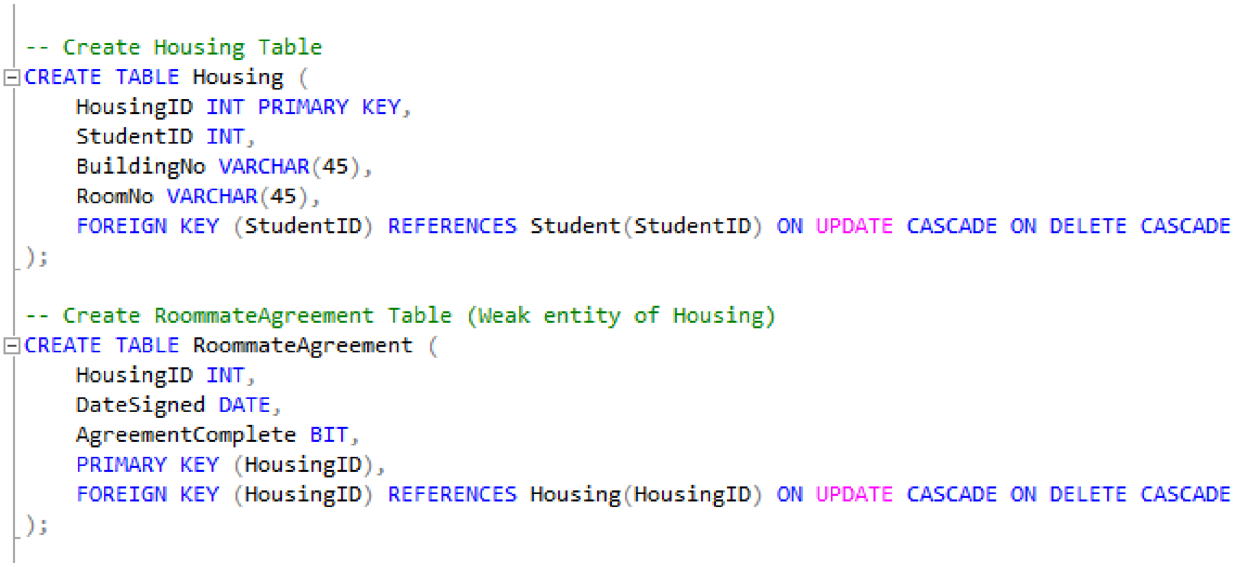
Query executed successfully.



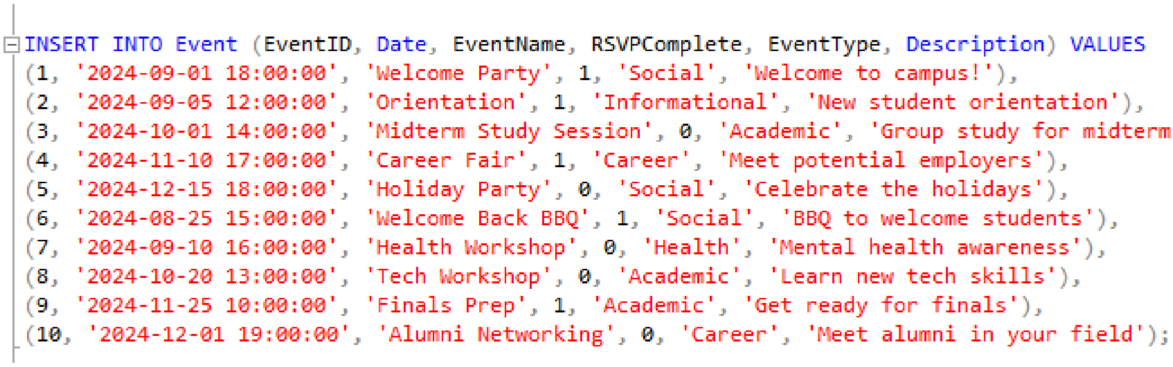
```sql
-- Create Assignment Table
CREATE TABLE Assignment (
    AssignmentID INT PRIMARY KEY,
    DueDate DATETIME,
    Description VARCHAR(45),
    AssignmentType VARCHAR(20) -- 'Exam', 'Homework', 'Quiz'
);

-- Create Exam Table (Subclass)
CREATE TABLE Exam
(
    ExamID INT PRIMARY KEY,
    AssignmentID INT NOT NULL FOREIGN KEY REFERENCES dbo.Assignment(AssignmentID) ON UPDATE CASCADE ON DELETE CASCADE
);

-- Create Grade Table (Subclass)
CREATE TABLE Quiz
(
    QuizID INT PRIMARY KEY,
    AssignmentID INT NOT NULL FOREIGN KEY REFERENCES dbo.Assignment(AssignmentID) ON UPDATE CASCADE ON DELETE CASCADE
);

-- Create Homework Table (Subclass)
CREATE TABLE Homework
(
    HomeworkID INT PRIMARY KEY,
    AssignmentID INT NOT NULL FOREIGN KEY REFERENCES dbo.Assignment(AssignmentID) ON UPDATE CASCADE ON DELETE CASCADE
);
```



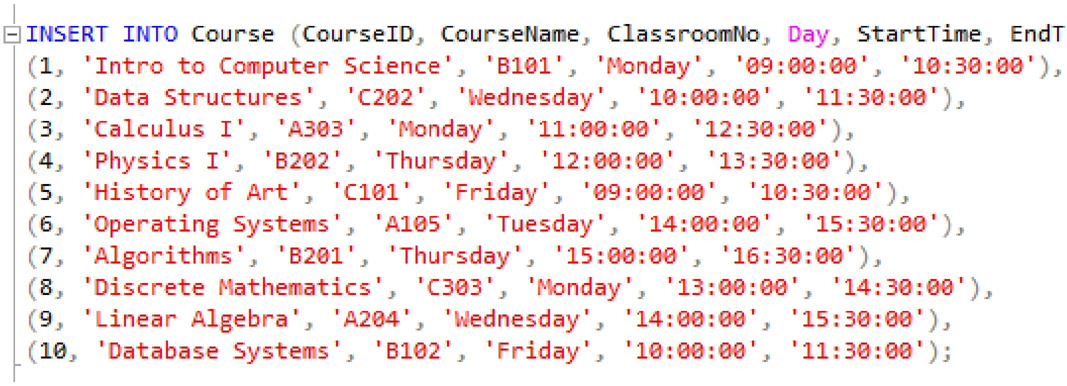
```sql
-- Create Finances Table
CREATE TABLE Finances (
    FinancesID INT PRIMARY KEY,
    StudentID INT,
    Amount DECIMAL(10, 2),
    FinancesType VARCHAR(45),
    PaymentDeadline DATETIME,
    Status BIT,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON UPDATE CASCADE ON DELETE CASCADE
);

-- Create Dining Table
CREATE TABLE Dining (
    DiningID INT PRIMARY KEY,
    StudentID INT,
    MealPlanBalance DECIMAL(10, 2),
    SwipesPerWeek INT,
    MealPlanType VARCHAR(45),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON UPDATE CASCADE ON DELETE CASCADE
);
```



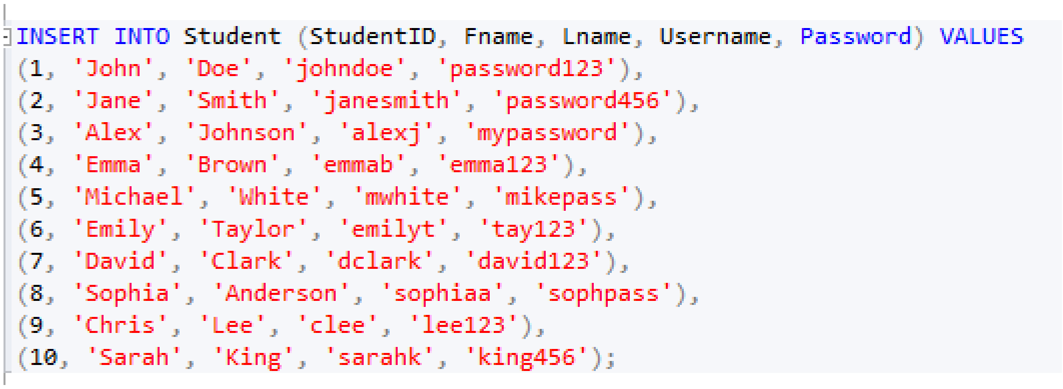
```sql
-- Create Housing Table
CREATE TABLE Housing (
    HousingID INT PRIMARY KEY,
    StudentID INT,
    BuildingNo VARCHAR(45),
    RoomNo VARCHAR(45),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON UPDATE CASCADE ON DELETE CASCADE
);

-- Create RoommateAgreement Table (Weak entity of Housing)
CREATE TABLE RoommateAgreement (
    HousingID INT,
    DateSigned DATE,
    AgreementComplete BIT,
    PRIMARY KEY (HousingID),
    FOREIGN KEY (HousingID) REFERENCES Housing(HousingID) ON UPDATE CASCADE ON DELETE CASCADE
);
```



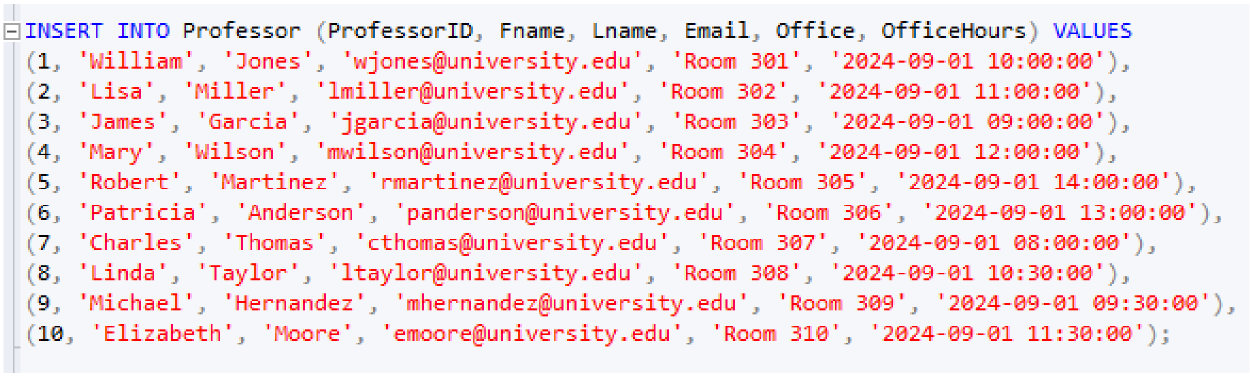
```sql
INSERT INTO Event (EventID, Date, EventName, RSVPComplete, EventType, Description) VALUES
(1, '2024-09-01 18:00:00', 'Welcome Party', 1, 'Social', 'Welcome to campus!'),
(2, '2024-09-05 12:00:00', 'Orientation', 1, 'Informational', 'New student orientation'),
(3, '2024-10-01 14:00:00', 'Midterm Study Session', 0, 'Academic', 'Group study for midterms'),
(4, '2024-11-10 17:00:00', 'Career Fair', 1, 'Career', 'Meet potential employers'),
(5, '2024-12-15 18:00:00', 'Holiday Party', 0, 'Social', 'Celebrate the holidays'),
(6, '2024-08-25 15:00:00', 'Welcome Back BBQ', 1, 'Social', 'BBQ to welcome students'),
(7, '2024-09-10 16:00:00', 'Health Workshop', 0, 'Health', 'Mental health awareness'),
(8, '2024-10-20 13:00:00', 'Tech Workshop', 0, 'Academic', 'Learn new tech skills'),
(9, '2024-11-25 10:00:00', 'Finals Prep', 1, 'Academic', 'Get ready for finals'),
(10, '2024-12-01 19:00:00', 'Alumni Networking', 0, 'Career', 'Meet alumni in your field');
```



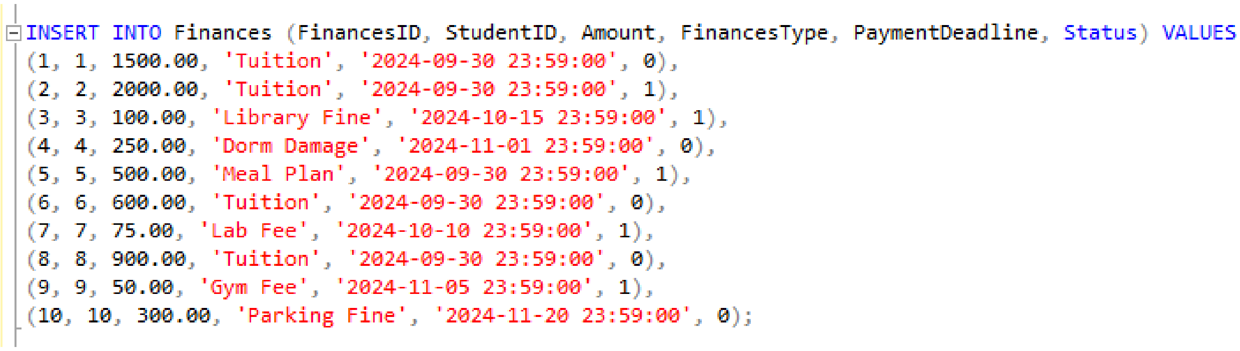
```
INSERT INTO Course (CourseID, CourseName, ClassroomNo, Day, StartTime, EndTime) VALUES
(1, 'Intro to Computer Science', 'B101', 'Monday', '09:00:00', '10:30:00'),
(2, 'Data Structures', 'C202', 'Wednesday', '10:00:00', '11:30:00'),
(3, 'Calculus I', 'A303', 'Monday', '11:00:00', '12:30:00'),
(4, 'Physics I', 'B202', 'Thursday', '12:00:00', '13:30:00'),
(5, 'History of Art', 'C101', 'Friday', '09:00:00', '10:30:00'),
(6, 'Operating Systems', 'A105', 'Tuesday', '14:00:00', '15:30:00'),
(7, 'Algorithms', 'B201', 'Thursday', '15:00:00', '16:30:00'),
(8, 'Discrete Mathematics', 'C303', 'Monday', '13:00:00', '14:30:00'),
(9, 'Linear Algebra', 'A204', 'Wednesday', '14:00:00', '15:30:00'),
(10, 'Database Systems', 'B102', 'Friday', '10:00:00', '11:30:00');
```



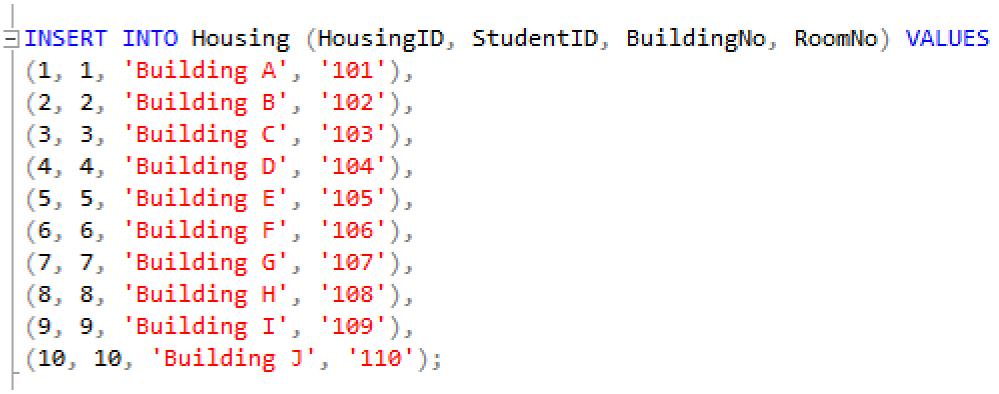
```
INSERT INTO Student (StudentID, Fname, Lname, Username, Password) VALUES
(1, 'John', 'Doe', 'johndoe', 'password123'),
(2, 'Jane', 'Smith', 'janesmith', 'password456'),
(3, 'Alex', 'Johnson', 'alexj', 'mypassword'),
(4, 'Emma', 'Brown', 'emmab', 'emma123'),
(5, 'Michael', 'White', 'mwhite', 'mikepass'),
(6, 'Emily', 'Taylor', 'emilyt', 'tay123'),
(7, 'David', 'Clark', 'dclark', 'david123'),
(8, 'Sophia', 'Anderson', 'sophiaa', 'sophpass'),
(9, 'Chris', 'Lee', 'clee', 'lee123'),
(10, 'Sarah', 'King', 'sarahk', 'king456');
```



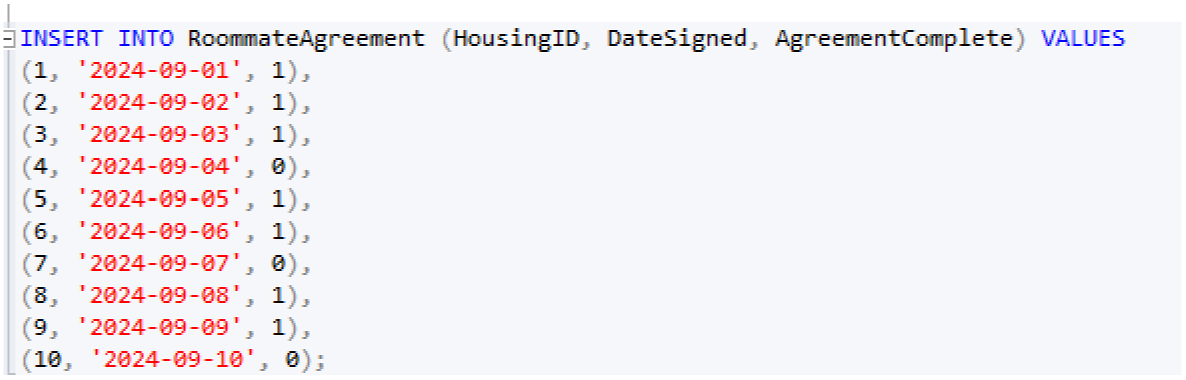
```
INSERT INTO Professor (ProfessorID, Fname, Lname, Email, Office, OfficeHours) VALUES
(1, 'William', 'Jones', 'wjones@university.edu', 'Room 301', '2024-09-01 10:00:00'),
(2, 'Lisa', 'Miller', 'lmiller@university.edu', 'Room 302', '2024-09-01 11:00:00'),
(3, 'James', 'Garcia', 'jgarcia@university.edu', 'Room 303', '2024-09-01 09:00:00'),
(4, 'Mary', 'Wilson', 'mwilson@university.edu', 'Room 304', '2024-09-01 12:00:00'),
(5, 'Robert', 'Martinez', 'rmartinez@university.edu', 'Room 305', '2024-09-01 14:00:00'),
(6, 'Patricia', 'Anderson', 'panderson@university.edu', 'Room 306', '2024-09-01 13:00:00'),
(7, 'Charles', 'Thomas', 'cthomas@university.edu', 'Room 307', '2024-09-01 08:00:00'),
(8, 'Linda', 'Taylor', 'ltaylor@university.edu', 'Room 308', '2024-09-01 10:30:00'),
(9, 'Michael', 'Hernandez', 'mhernandez@university.edu', 'Room 309', '2024-09-01 09:30:00'),
(10, 'Elizabeth', 'Moore', 'emoore@university.edu', 'Room 310', '2024-09-01 11:30:00');
```



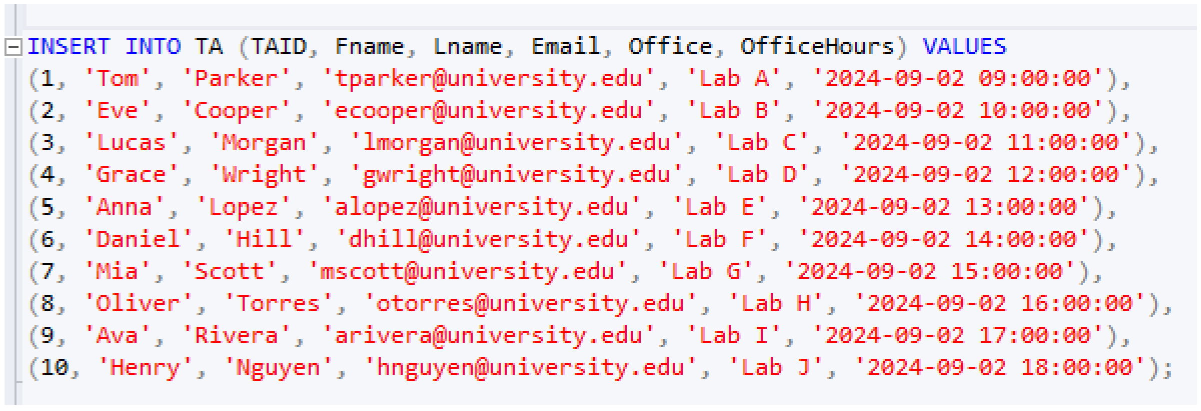
```
INSERT INTO Finances (FinancesID, StudentID, Amount, FinancesType, PaymentDeadline, Status) VALUES
(1, 1, 1500.00, 'Tuition', '2024-09-30 23:59:00', 0),
(2, 2, 2000.00, 'Tuition', '2024-09-30 23:59:00', 1),
(3, 3, 100.00, 'Library Fine', '2024-10-15 23:59:00', 1),
(4, 4, 250.00, 'Dorm Damage', '2024-11-01 23:59:00', 0),
(5, 5, 500.00, 'Meal Plan', '2024-09-30 23:59:00', 1),
(6, 6, 600.00, 'Tuition', '2024-09-30 23:59:00', 0),
(7, 7, 75.00, 'Lab Fee', '2024-10-10 23:59:00', 1),
(8, 8, 900.00, 'Tuition', '2024-09-30 23:59:00', 0),
(9, 9, 50.00, 'Gym Fee', '2024-11-05 23:59:00', 1),
(10, 10, 300.00, 'Parking Fine', '2024-11-20 23:59:00', 0);
```



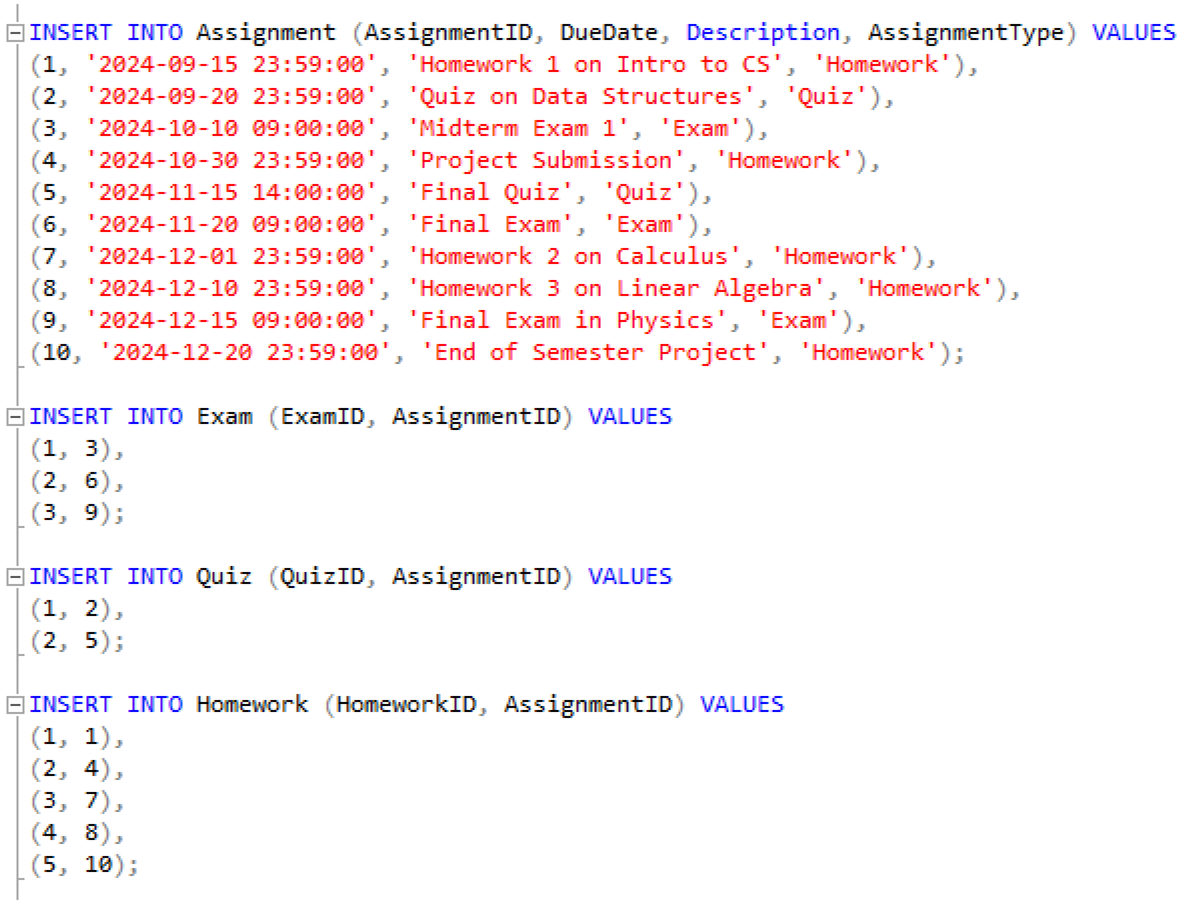
```sql
INSERT INTO Housing (HousingID, StudentID, BuildingNo, RoomNo) VALUES
(1, 1, 'Building A', '101'),
(2, 2, 'Building B', '102'),
(3, 3, 'Building C', '103'),
(4, 4, 'Building D', '104'),
(5, 5, 'Building E', '105'),
(6, 6, 'Building F', '106'),
(7, 7, 'Building G', '107'),
(8, 8, 'Building H', '108'),
(9, 9, 'Building I', '109'),
(10, 10, 'Building J', '110');
```



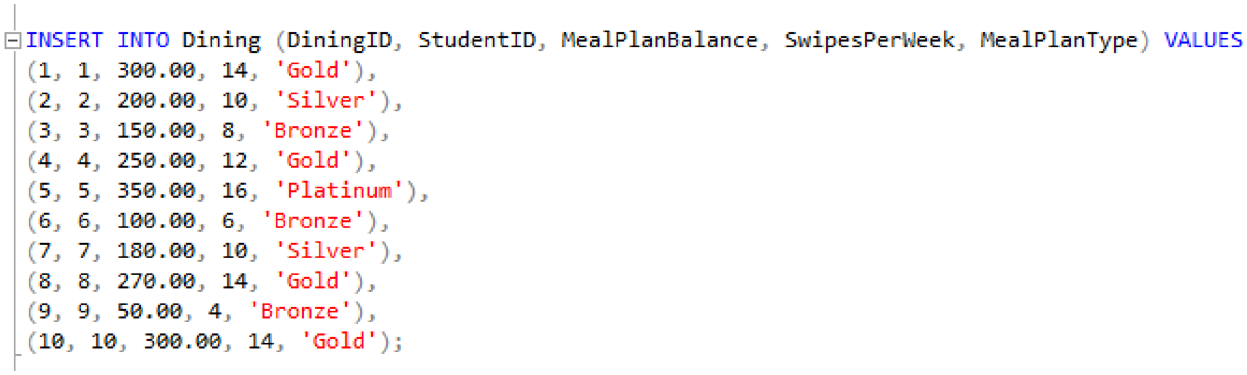
```sql
INSERT INTO RoommateAgreement (HousingID, DateSigned, AgreementComplete) VALUES
(1, '2024-09-01', 1),
(2, '2024-09-02', 1),
(3, '2024-09-03', 1),
(4, '2024-09-04', 0),
(5, '2024-09-05', 1),
(6, '2024-09-06', 1),
(7, '2024-09-07', 0),
(8, '2024-09-08', 1),
(9, '2024-09-09', 1),
(10, '2024-09-10', 0);
```



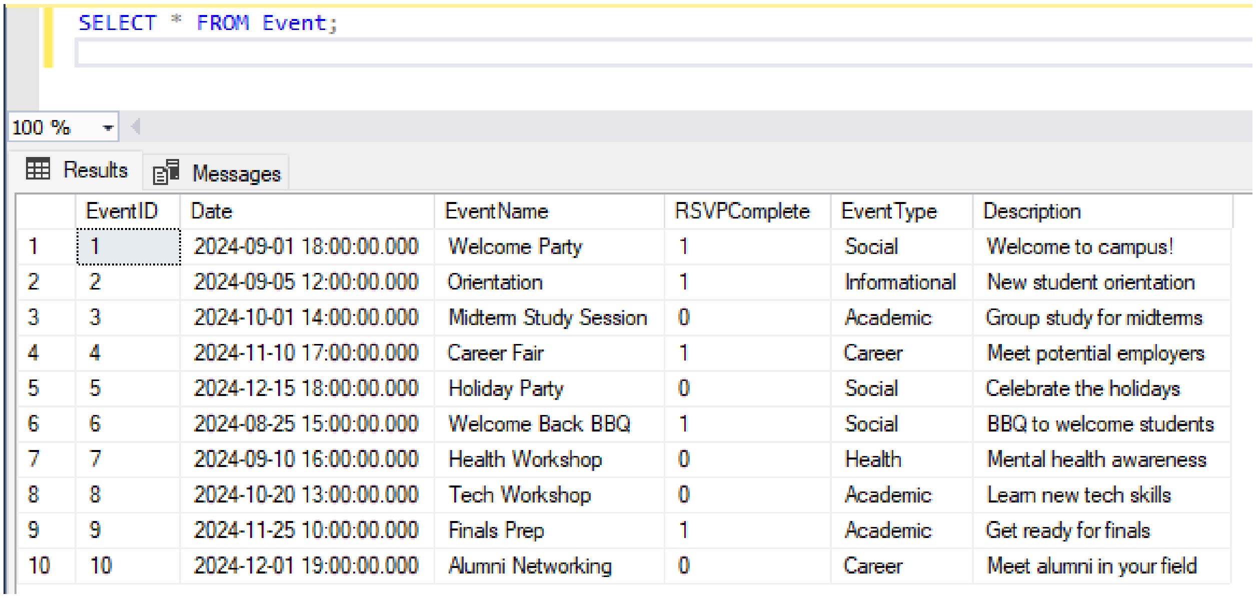
```sql
INSERT INTO TA (TAID, Fname, Lname, Email, Office, OfficeHours) VALUES
(1, 'Tom', 'Parker', 'tparker@university.edu', 'Lab A', '2024-09-02 09:00:00'),
(2, 'Eve', 'Cooper', 'ecooper@university.edu', 'Lab B', '2024-09-02 10:00:00'),
(3, 'Lucas', 'Morgan', 'lmorgan@university.edu', 'Lab C', '2024-09-02 11:00:00'),
(4, 'Grace', 'Wright', 'gwright@university.edu', 'Lab D', '2024-09-02 12:00:00'),
(5, 'Anna', 'Lopez', 'alopez@university.edu', 'Lab E', '2024-09-02 13:00:00'),
(6, 'Daniel', 'Hill', 'dhill@university.edu', 'Lab F', '2024-09-02 14:00:00'),
(7, 'Mia', 'Scott', 'mscott@university.edu', 'Lab G', '2024-09-02 15:00:00'),
(8, 'Oliver', 'Torres', 'otorres@university.edu', 'Lab H', '2024-09-02 16:00:00'),
(9, 'Ava', 'Rivera', 'arivera@university.edu', 'Lab I', '2024-09-02 17:00:00'),
(10, 'Henry', 'Nguyen', 'hnguyen@university.edu', 'Lab J', '2024-09-02 18:00:00');
```



```sql
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType) VALUES
(1, '2024-09-15 23:59:00', 'Homework 1 on Intro to CS', 'Homework'),
(2, '2024-09-20 23:59:00', 'Quiz on Data Structures', 'Quiz'),
(3, '2024-10-10 09:00:00', 'Midterm Exam 1', 'Exam'),
(4, '2024-10-30 23:59:00', 'Project Submission', 'Homework'),
(5, '2024-11-15 14:00:00', 'Final Quiz', 'Quiz'),
(6, '2024-11-20 09:00:00', 'Final Exam', 'Exam'),
(7, '2024-12-01 23:59:00', 'Homework 2 on Calculus', 'Homework'),
(8, '2024-12-10 23:59:00', 'Homework 3 on Linear Algebra', 'Homework'),
(9, '2024-12-15 09:00:00', 'Final Exam in Physics', 'Exam'),
(10, '2024-12-20 23:59:00', 'End of Semester Project', 'Homework');

INSERT INTO Exam (ExamID, AssignmentID) VALUES
(1, 3),
(2, 6),
(3, 9);

INSERT INTO Quiz (QuizID, AssignmentID) VALUES
(1, 2),
(2, 5);

INSERT INTO Homework (HomeworkID, AssignmentID) VALUES
(1, 1),
(2, 4),
(3, 7),
(4, 8),
(5, 10);
```

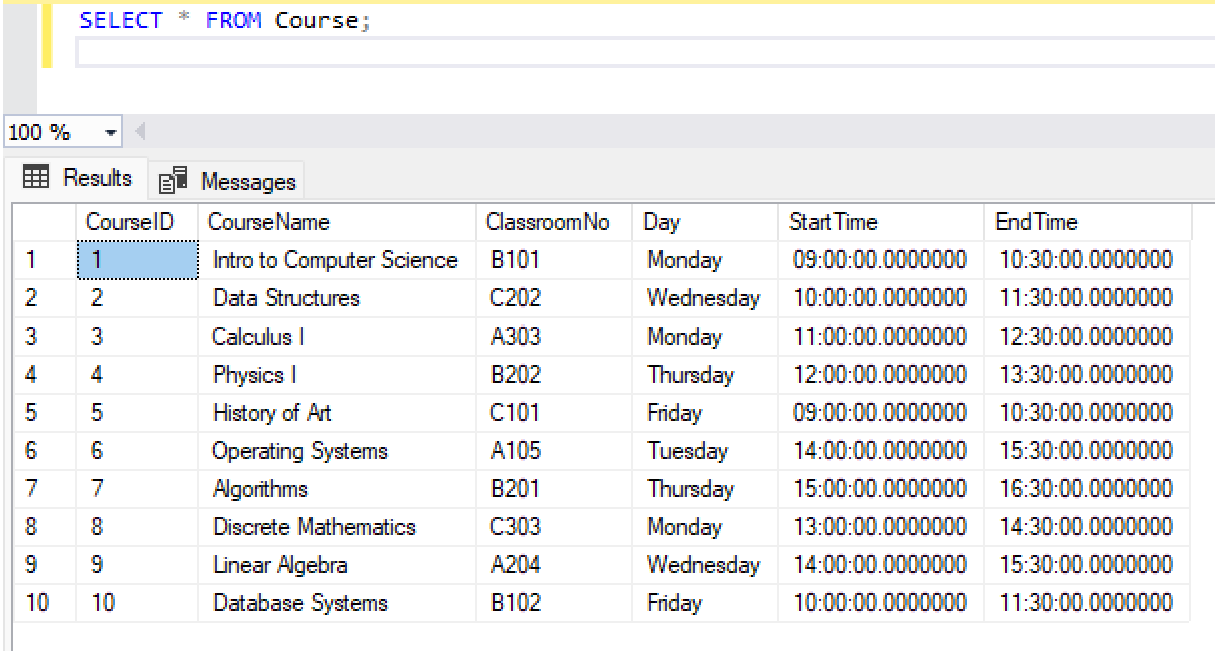


```sql
INSERT INTO Dining (DiningID, StudentID, MealPlanBalance, SwipesPerWeek, MealPlanType) VALUES
(1, 1, 300.00, 14, 'Gold'),
(2, 2, 200.00, 10, 'Silver'),
(3, 3, 150.00, 8, 'Bronze'),
(4, 4, 250.00, 12, 'Gold'),
(5, 5, 350.00, 16, 'Platinum'),
(6, 6, 100.00, 6, 'Bronze'),
(7, 7, 180.00, 10, 'Silver'),
(8, 8, 270.00, 14, 'Gold'),
(9, 9, 50.00, 4, 'Bronze'),
(10, 10, 300.00, 14, 'Gold');
```



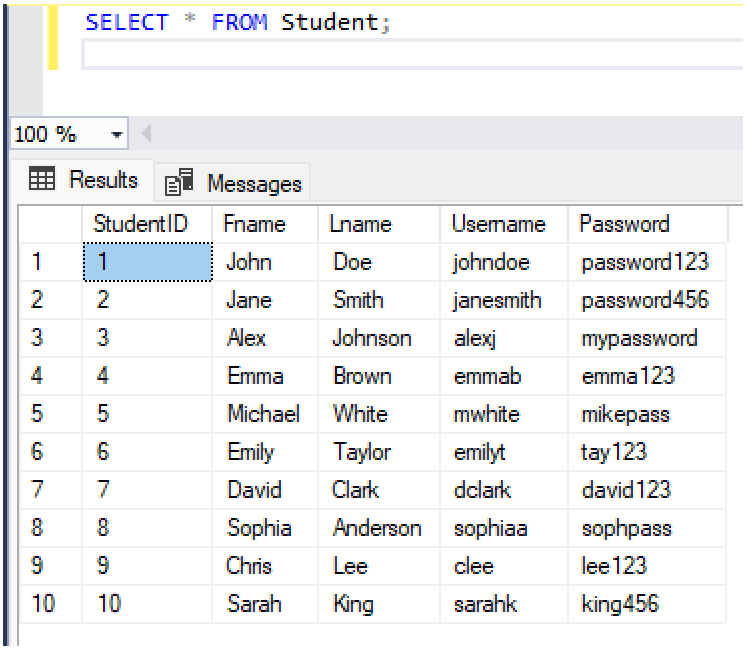
```
SELECT * FROM Event;
```

	EventID	Date	EventName	RSVPComplete	EventType	Description
1	1	2024-09-01 18:00:00.000	Welcome Party	1	Social	Welcome to campus!
2	2	2024-09-05 12:00:00.000	Orientation	1	Informational	New student orientation
3	3	2024-10-01 14:00:00.000	Midterm Study Session	0	Academic	Group study for midterms
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome students
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills
9	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals
10	10	2024-12-01 19:00:00.000	Alumni Networking	0	Career	Meet alumni in your field



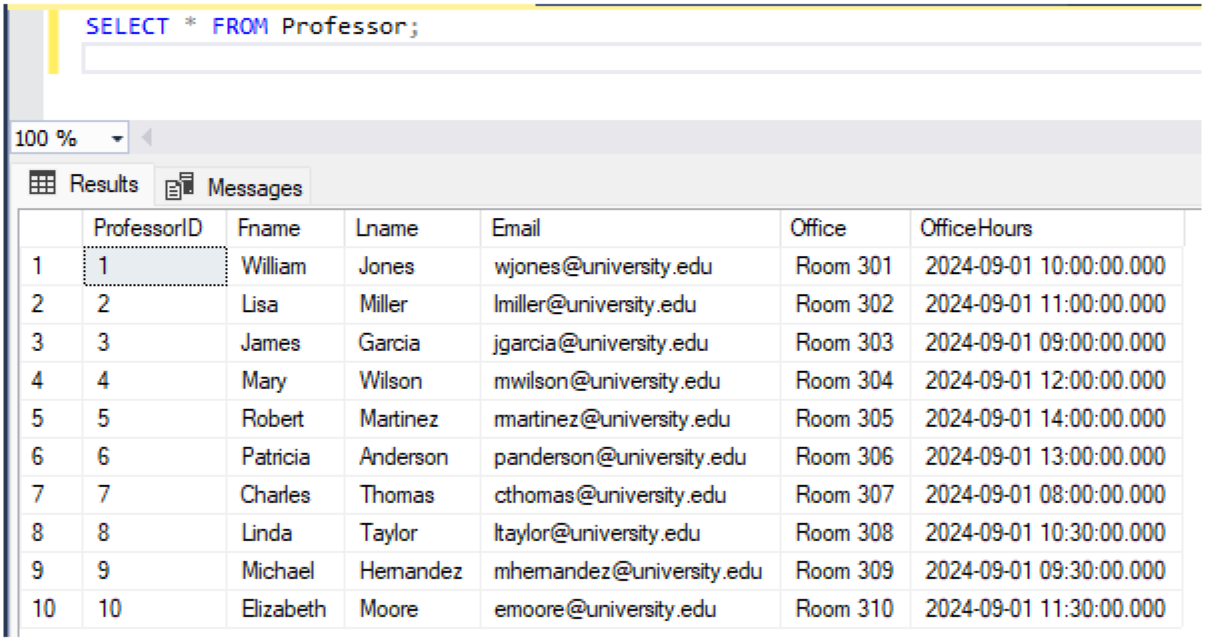
```
SELECT * FROM Course;
```

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
1	1	Intro to Computer Science	B101	Monday	09:00:00.0000000	10:30:00.0000000
2	2	Data Structures	C202	Wednesday	10:00:00.0000000	11:30:00.0000000
3	3	Calculus I	A303	Monday	11:00:00.0000000	12:30:00.0000000
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	6	Operating Systems	A105	Tuesday	14:00:00.0000000	15:30:00.0000000
7	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
8	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
9	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000
10	10	Database Systems	B102	Friday	10:00:00.0000000	11:30:00.0000000



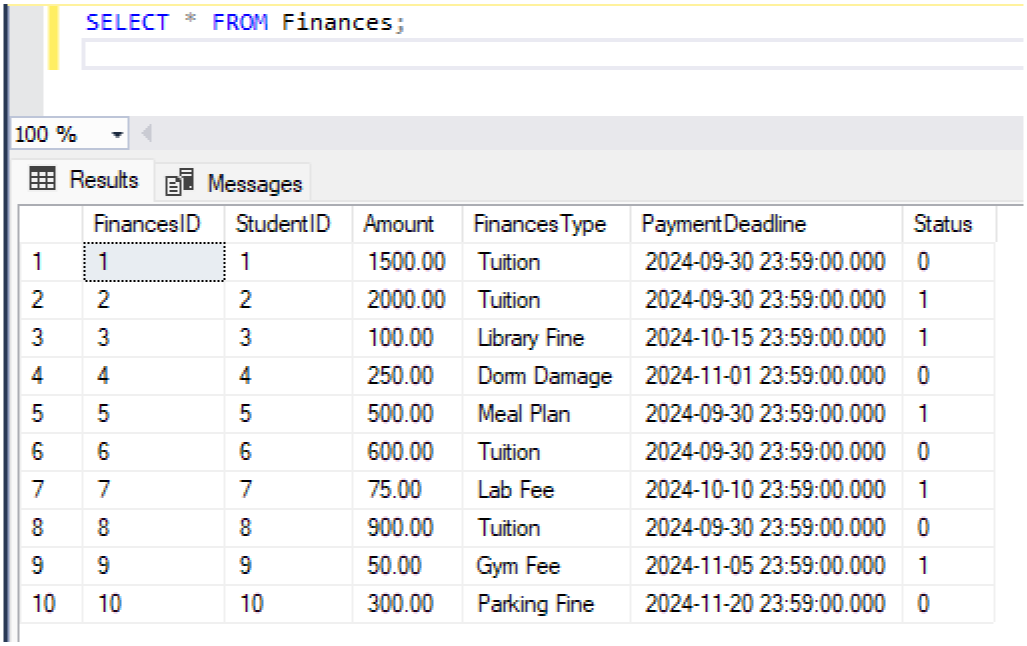
```
SELECT * FROM Student;
```

	StudentID	Fname	Lname	Username	Password
1	1	John	Doe	johndoe	password123
2	2	Jane	Smith	janesmith	password456
3	3	Alex	Johnson	alexj	mypassword
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anderson	sophiaa	sophpass
9	9	Chris	Lee	clee	lee123
10	10	Sarah	King	sarahk	king456



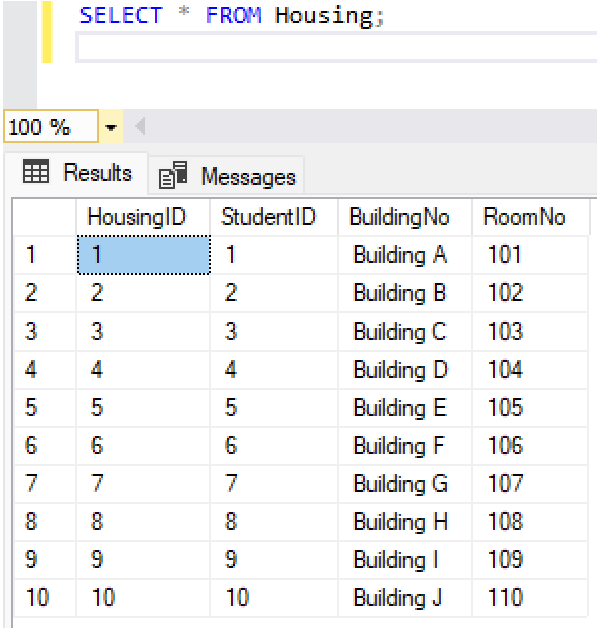
```
SELECT * FROM Professor;
```

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
1	1	William	Jones	wjones@university.edu	Room 301	2024-09-01 10:00:00.000
2	2	Lisa	Miller	lmiller@university.edu	Room 302	2024-09-01 11:00:00.000
3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	6	Patricia	Anderson	panderson@university.edu	Room 306	2024-09-01 13:00:00.000
7	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
8	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
9	9	Michael	Hernandez	mhernandez@university.edu	Room 309	2024-09-01 09:30:00.000
10	10	Elizabeth	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000



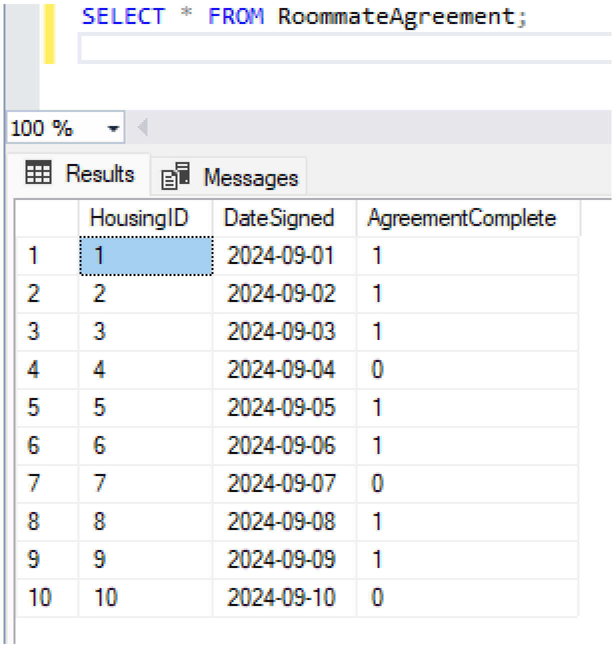
```sql
SELECT * FROM Finances;
```

	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0



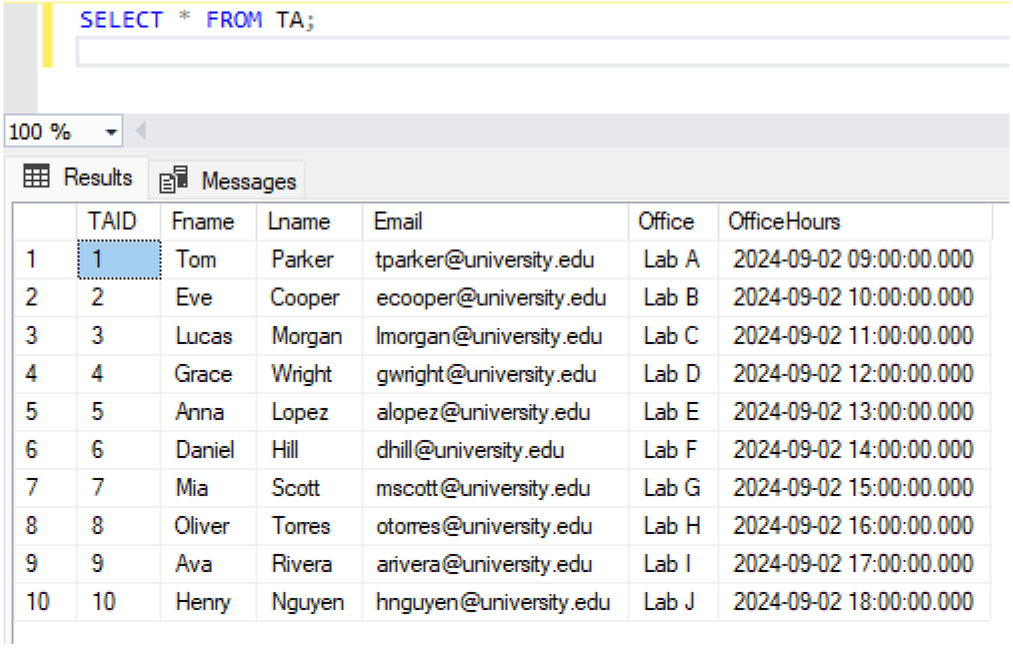
```sql
SELECT * FROM Housing;
```

	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	10	10	Building J	110



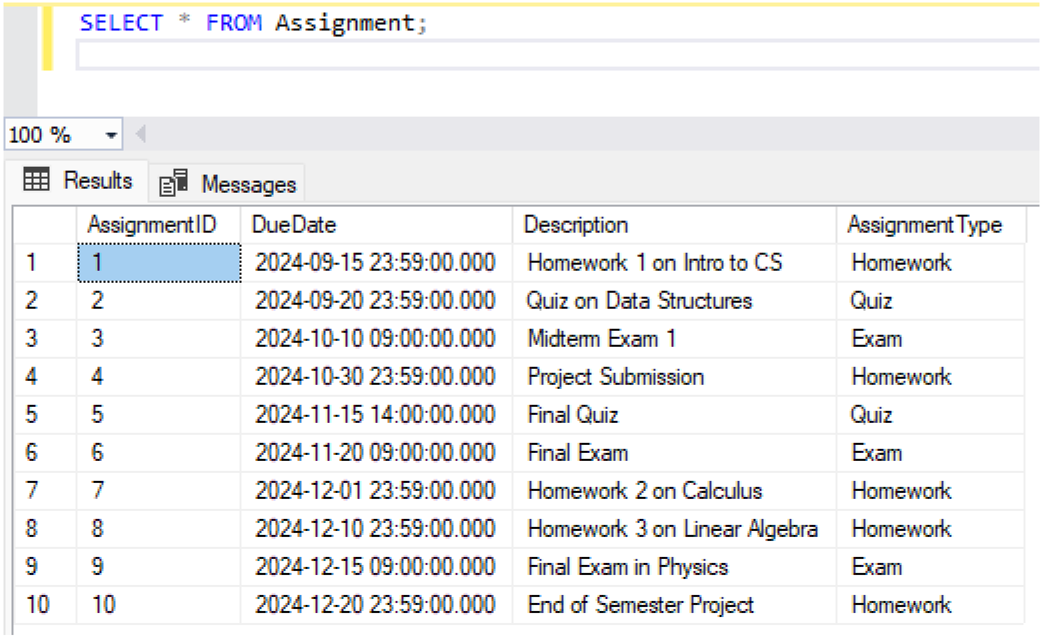
```
SELECT * FROM RoommateAgreement;
```

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1
10	10	2024-09-10	0



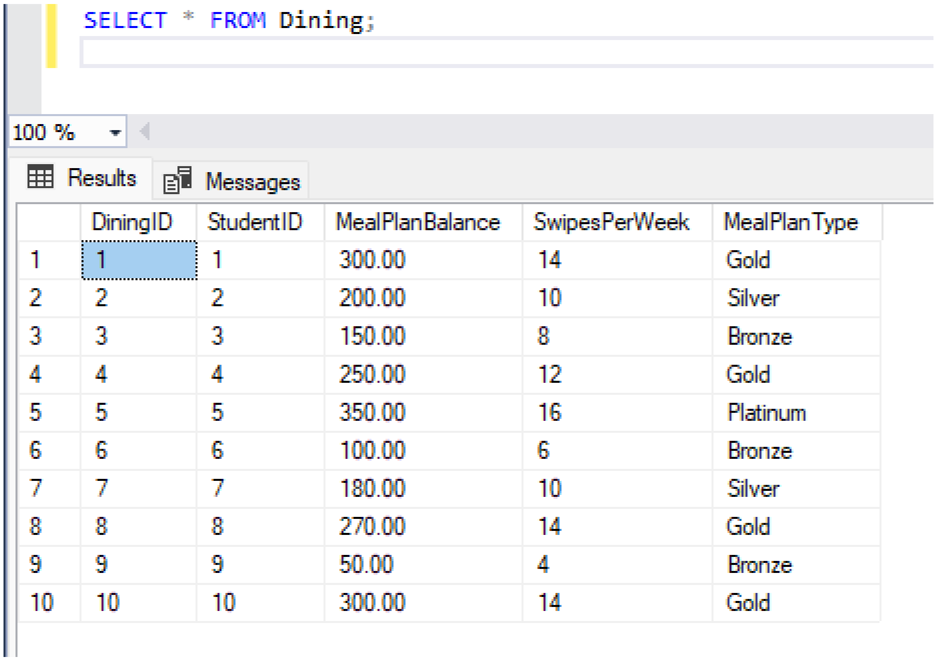
```
SELECT * FROM TA;
```

	TAID	Fname	Lname	Email	Office	OfficeHours
1	1	Tom	Parker	tparker@university.edu	Lab A	2024-09-02 09:00:00.000
2	2	Eve	Cooper	ecooper@university.edu	Lab B	2024-09-02 10:00:00.000
3	3	Lucas	Morgan	lmorgan@university.edu	Lab C	2024-09-02 11:00:00.000
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	6	Daniel	Hill	dhill@university.edu	Lab F	2024-09-02 14:00:00.000
7	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
8	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
9	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000
10	10	Henry	Nguyen	hnguyen@university.edu	Lab J	2024-09-02 18:00:00.000



```
SELECT * FROM Assignment;
```

	AssignmentID	DueDate	Description	AssignmentType
1	1	2024-09-15 23:59:00.000	Homework 1 on Intro to CS	Homework
2	2	2024-09-20 23:59:00.000	Quiz on Data Structures	Quiz
3	3	2024-10-10 09:00:00.000	Midterm Exam 1	Exam
4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework



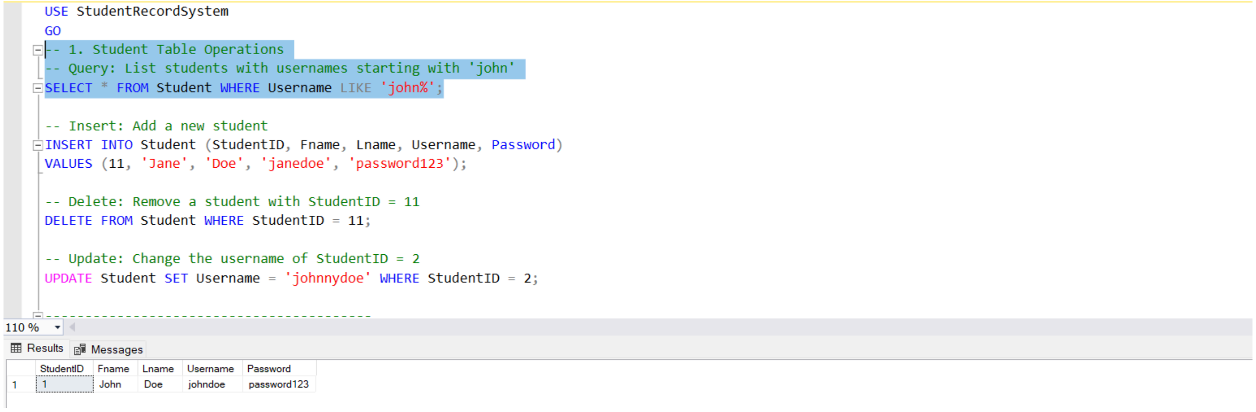
```
SELECT * FROM Dining;
```

	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold

3.4. Database Query Execution (from inside your SQL client):

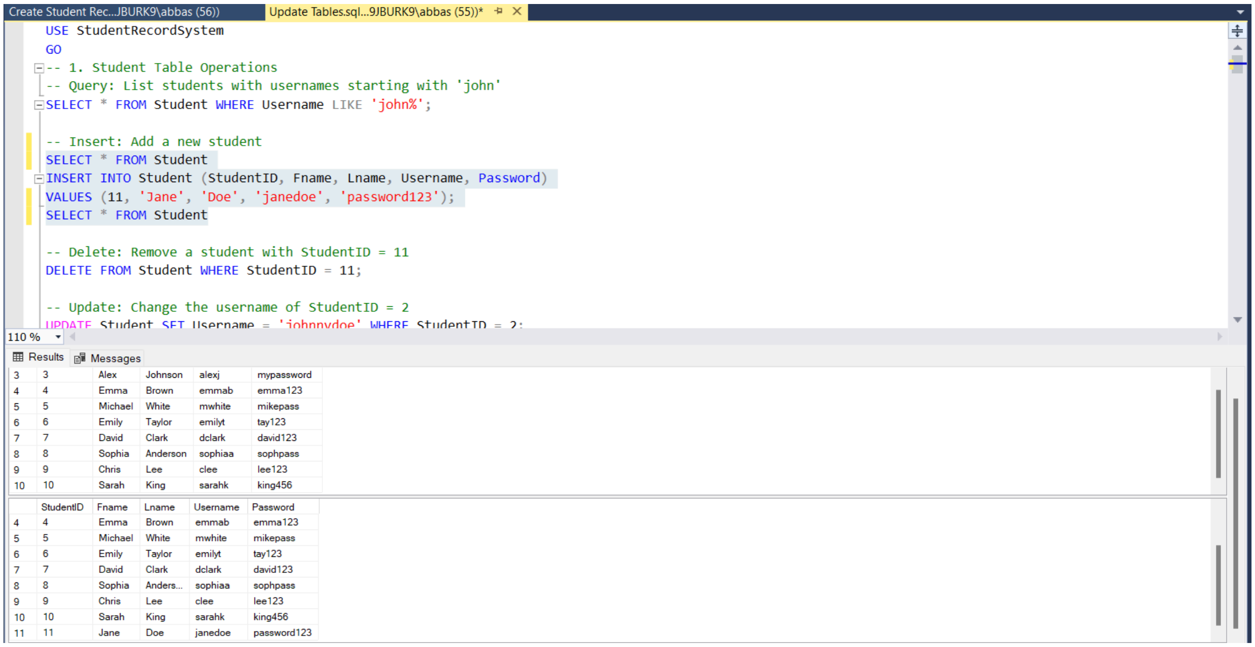
Use an SQL platform to provide sample executions for each of the following operations on each table of your database: o Query – to perform operations such as list employees earning more than 150K per year o Insert – to add new tuple(s), and/or field(s) to your table(s) o Delete - to remove tuple(s), and/or field(s) from your table(s) o Update - to modify tuple(s), and/or field(s) from your table(s) Provide a screenshot of each operation and each resulting table in your report exactly in this section.

1. Student Table



```
USE StudentRecordSystem
GO
-- 1. Student Table Operations
-- Query: List students with usernames starting with 'john'
SELECT * FROM Student WHERE Username LIKE 'john%';

-- Insert: Add a new student
INSERT INTO Student (StudentID, Fname, Lname, Username, Password)
VALUES (11, 'Jane', 'Doe', 'janedoe', 'password123');

-- Delete: Remove a student with StudentID = 11
DELETE FROM Student WHERE StudentID = 11;

-- Update: Change the username of StudentID = 2
UPDATE Student SET Username = 'johnnydoe' WHERE StudentID = 2;

----------------------------------------------
```

	StudentID	Fname	Lname	Username	Password
1	1	John	Doe	johndoe	password123

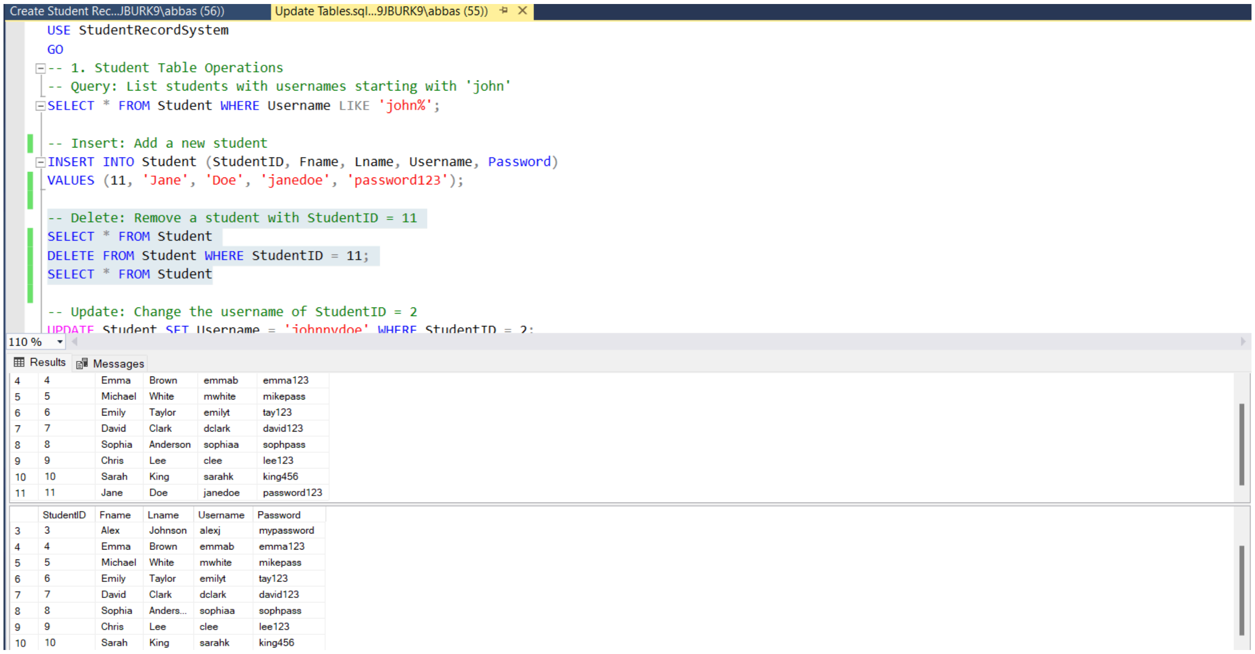


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*

```
USE StudentRecordSystem
GO
-- 1. Student Table Operations
-- Query: List students with usernames starting with 'john'
SELECT * FROM Student WHERE Username LIKE 'john%';

-- Insert: Add a new student
SELECT * FROM Student
INSERT INTO Student (StudentID, Fname, Lname, Username, Password)
VALUES (11, 'Jane', 'Doe', 'janedoe', 'password123');
SELECT * FROM Student

-- Delete: Remove a student with StudentID = 11
DELETE FROM Student WHERE StudentID = 11;

-- Update: Change the username of StudentID = 2
UPDATE Student SET Username = 'johnnydoe' WHERE StudentID = 2;
```

3	3	Alex	Johnson	alexj	mypassword
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anderson	sophiaa	sophpass
9	9	Chris	Lee	clee	lee123
10	10	Sarah	King	sarahk	king456

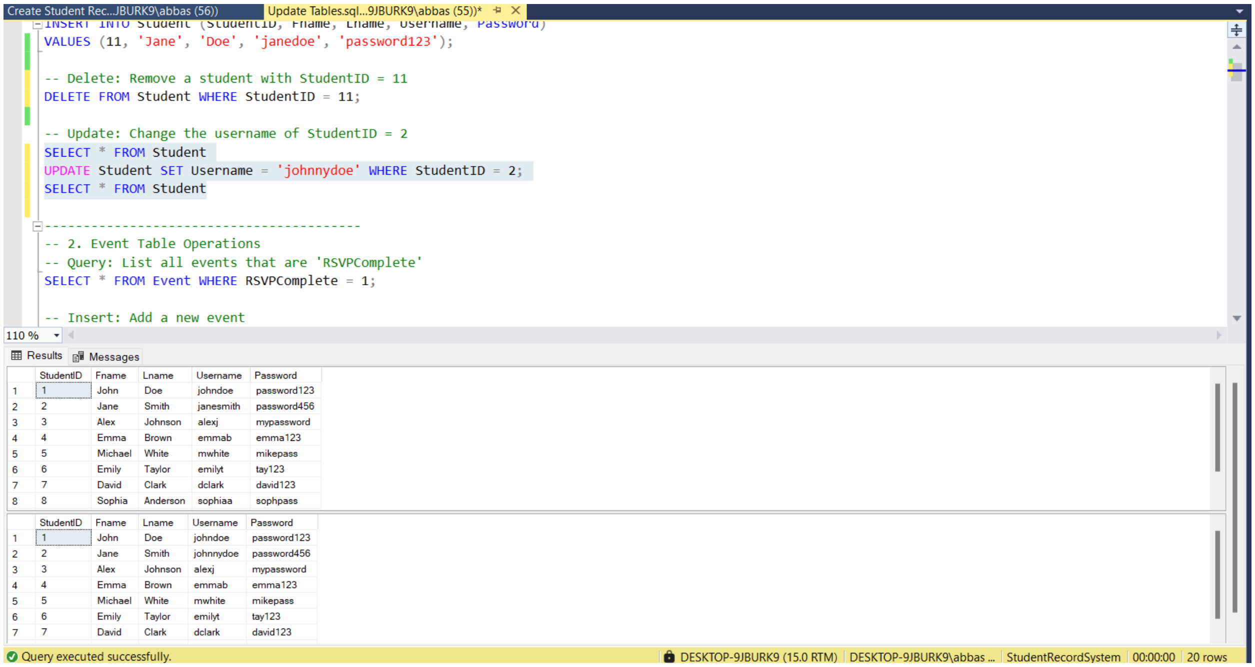
	StudentID	Fname	Lname	Username	Password
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anders...	sophiaa	sophpass
9	9	Chris	Lee	clee	lee123
10	10	Sarah	King	sarahk	king456
11	11	Jane	Doe	janedoe	password123


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))
110 %
Results
Messages

```sql
USE StudentRecordSystem
GO
-- 1. Student Table Operations
-- Query: List students with usernames starting with 'john'
SELECT * FROM Student WHERE Username LIKE 'john%';

-- Insert: Add a new student
INSERT INTO Student (StudentID, Fname, Lname, Username, Password)
VALUES (11, 'Jane', 'Doe', 'janedoe', 'password123');

-- Delete: Remove a student with StudentID = 11
SELECT * FROM Student
DELETE FROM Student WHERE StudentID = 11;
SELECT * FROM Student

-- Update: Change the username of StudentID = 2
UPDATE Student SET Username = 'johnnydoe' WHERE StudentID = 2;
```

4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anderson	sophiaa	sophpass
9	9	Chris	Lee	clee	lee123
10	10	Sarah	King	sarahk	king456
11	11	Jane	Doe	janedoe	password123

	StudentID	Fname	Lname	Username	Password
3	3	Alex	Johnson	alexj	mypassword
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anders...	sophiaa	sophpass
9	9	Chris	Lee	clee	lee123
10	10	Sarah	King	sarahk	king456


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

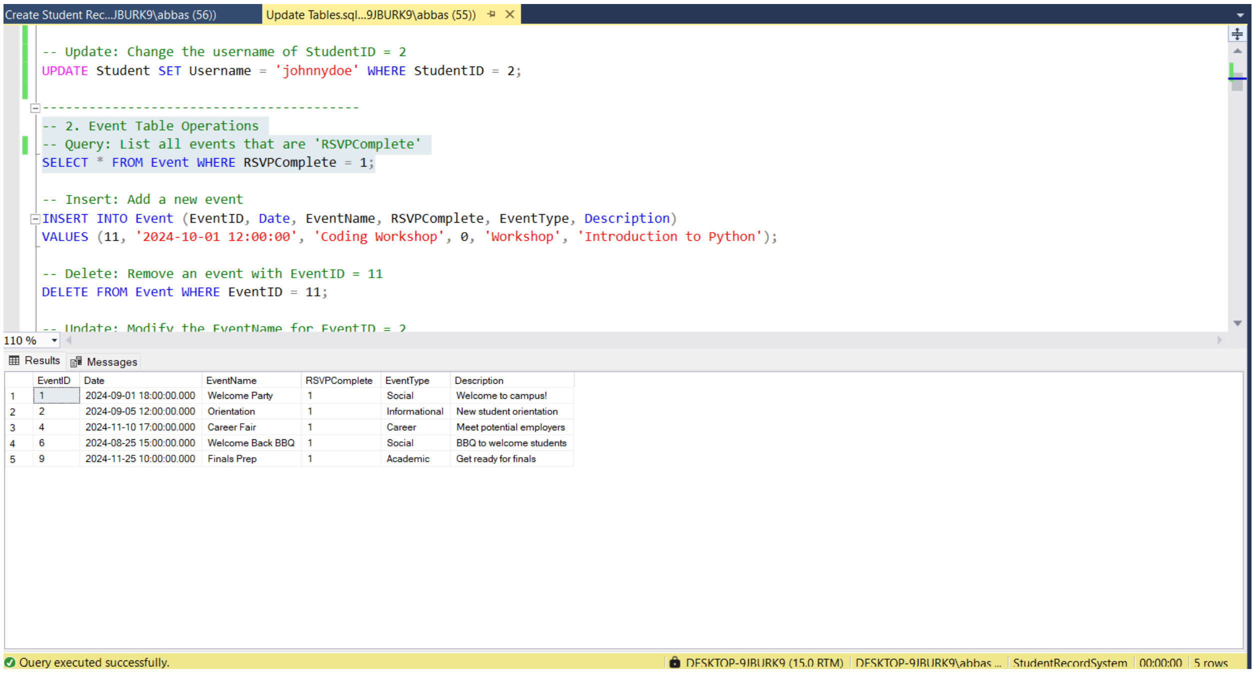
```sql
INSERT INTO Student (StudentID, Fname, Lname, Username, Password)
VALUES (11, 'Jane', 'Doe', 'janedoe', 'password123');

-- Delete: Remove a student with StudentID = 11
DELETE FROM Student WHERE StudentID = 11;

-- Update: Change the username of StudentID = 2
SELECT * FROM Student
UPDATE Student SET Username = 'johnnydoe' WHERE StudentID = 2;
SELECT * FROM Student

-----------------------------------------
-- 2. Event Table Operations
-- Query: List all events that are 'RSVPComplete'
SELECT * FROM Event WHERE RSVPComplete = 1;

-- Insert: Add a new event
```

	StudentID	Fname	Lname	Username	Password
1	1	John	Doe	johndoe	password123
2	2	Jane	Smith	janesmith	password456
3	3	Alex	Johnson	alexj	mypassword
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123
8	8	Sophia	Anderson	sophiaa	sophpass

	StudentID	Fname	Lname	Username	Password
1	1	John	Doe	johndoe	password123
2	2	Jane	Smith	johnnydoe	password456
3	3	Alex	Johnson	alexj	mypassword
4	4	Emma	Brown	emmab	emma123
5	5	Michael	White	mwhite	mikepass
6	6	Emily	Taylor	emilyt	tay123
7	7	David	Clark	dclark	david123

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 20 rows

2. Event Table

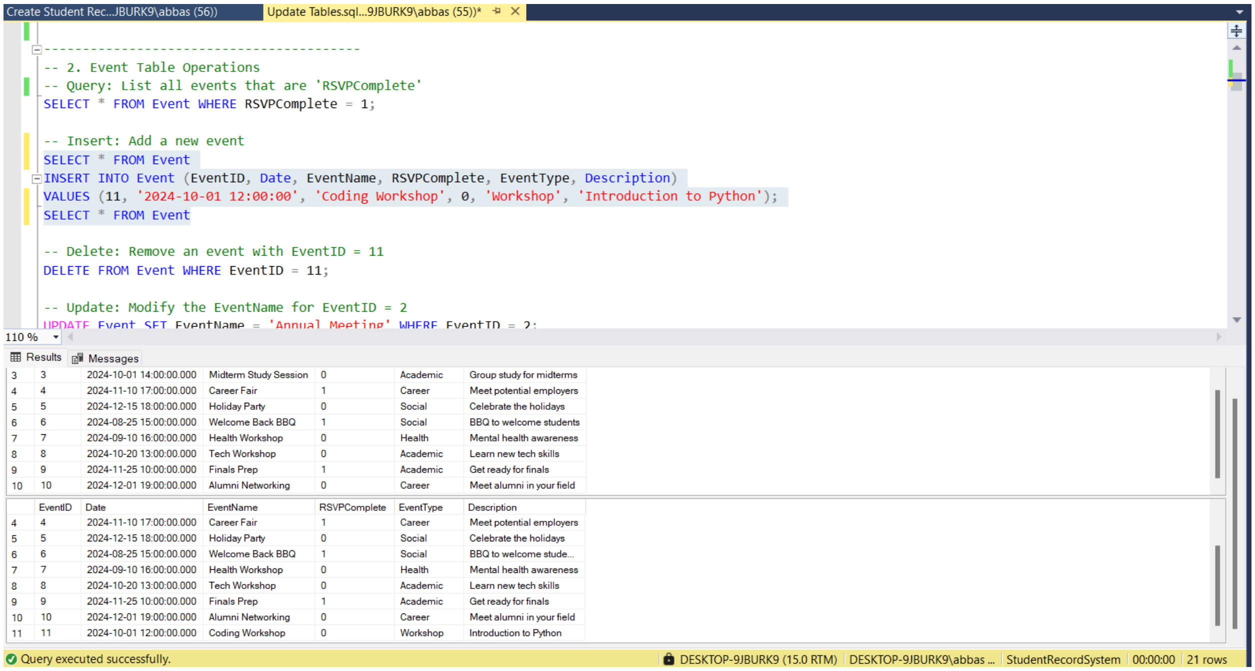


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))
110 %
Results
Messages

```sql
-- Update: Change the username of StudentID = 2
UPDATE Student SET Username = 'johnnydoe' WHERE StudentID = 2;

----------------------------------------
-- 2. Event Table Operations
-- Query: List all events that are 'RSVPComplete'
SELECT * FROM Event WHERE RSVPComplete = 1;

-- Insert: Add a new event
INSERT INTO Event (EventID, Date, EventName, RSVPComplete, EventType, Description)
VALUES (11, '2024-10-01 12:00:00', 'Coding Workshop', 0, 'Workshop', 'Introduction to Python');

-- Delete: Remove an event with EventID = 11
DELETE FROM Event WHERE EventID = 11;

-- Update: Modify the EventName for EventID = 2
```

	EventID	Date	EventName	RSVPComplete	EventType	Description
1	1	2024-09-01 18:00:00.000	Welcome Party	1	Social	Welcome to campus!
2	2	2024-09-05 12:00:00.000	Orientation	1	Informational	New student orientation
3	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
4	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome students
5	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 5 rows



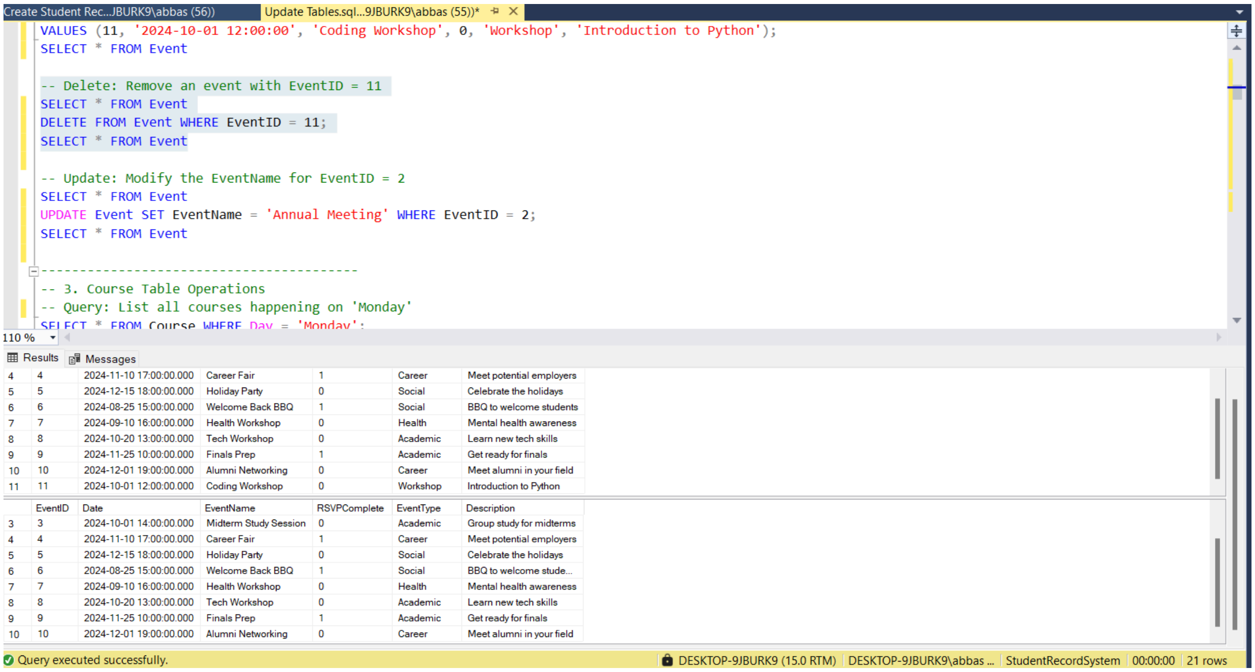
Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
----------------------------------------
-- 2. Event Table Operations
-- Query: List all events that are 'RSVPComplete'
SELECT * FROM Event WHERE RSVPComplete = 1;

-- Insert: Add a new event
SELECT * FROM Event
INSERT INTO Event (EventID, Date, EventName, RSVPComplete, EventType, Description)
VALUES (11, '2024-10-01 12:00:00', 'Coding Workshop', 0, 'Workshop', 'Introduction to Python');
SELECT * FROM Event

-- Delete: Remove an event with EventID = 11
DELETE FROM Event WHERE EventID = 11;

-- Update: Modify the EventName for EventID = 2
UPDATE Event SET EventName = 'Annual Meeting' WHERE EventID = 2;
```

3	3	2024-10-01 14:00:00.000	Midterm Study Session	0	Academic	Group study for midterms
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome students
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills
9	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals
10	10	2024-12-01 19:00:00.000	Alumni Networking	0	Career	Meet alumni in your field

	EventID	Date	EventName	RSVPComplete	EventType	Description
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome stude...
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills
9	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals
10	10	2024-12-01 19:00:00.000	Alumni Networking	0	Career	Meet alumni in your field
11	11	2024-10-01 12:00:00.000	Coding Workshop	0	Workshop	Introduction to Python

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows

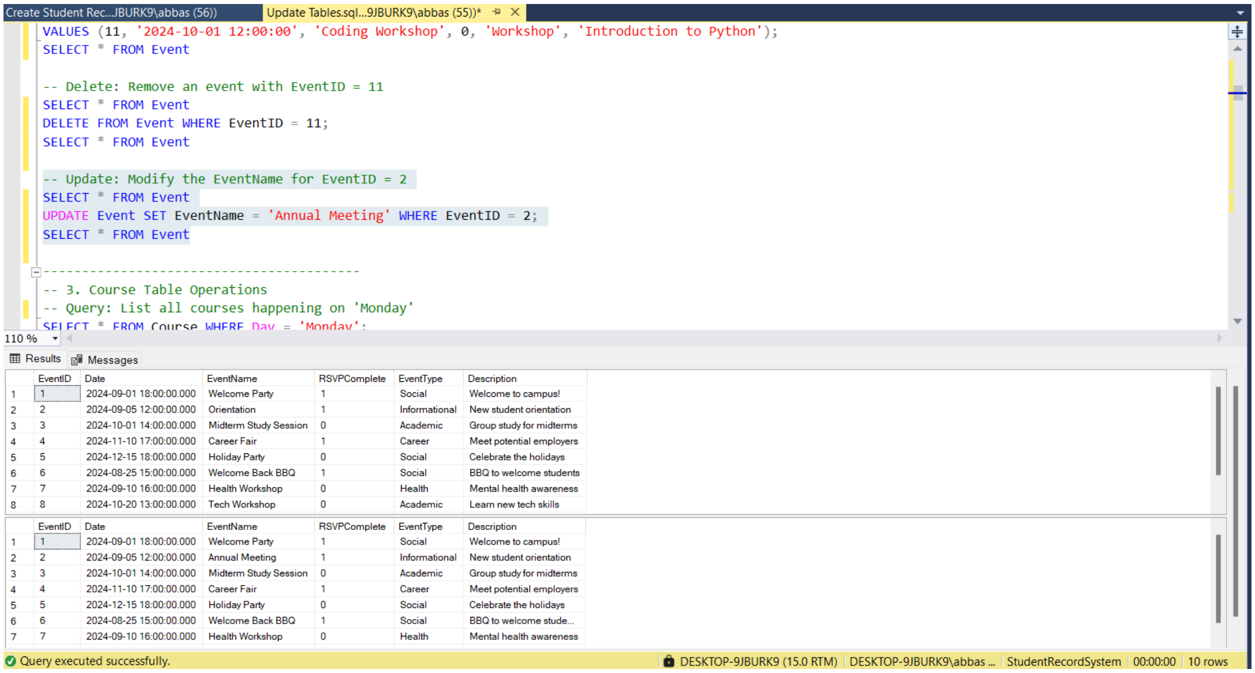

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```
VALUES (11, '2024-10-01 12:00:00', 'Coding Workshop', 0, 'Workshop', 'Introduction to Python');
SELECT * FROM Event

-- Delete: Remove an event with EventID = 11
SELECT * FROM Event
DELETE FROM Event WHERE EventID = 11;
SELECT * FROM Event

-- Update: Modify the EventName for EventID = 2
SELECT * FROM Event
UPDATE Event SET EventName = 'Annual Meeting' WHERE EventID = 2;
SELECT * FROM Event

-----------------------------------------
-- 3. Course Table Operations
-- Query: List all courses happening on 'Monday'
SELECT * FROM Course WHERE Day = 'Monday';
```

4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome students
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills
9	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals
10	10	2024-12-01 19:00:00.000	Alumni Networking	0	Career	Meet alumni in your field
11	11	2024-10-01 12:00:00.000	Coding Workshop	0	Workshop	Introduction to Python

	EventID	Date	EventName	RSVPComplete	EventType	Description
3	3	2024-10-01 14:00:00.000	Midterm Study Session	0	Academic	Group study for midterms
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome stude...
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills
9	9	2024-11-25 10:00:00.000	Finals Prep	1	Academic	Get ready for finals
10	10	2024-12-01 19:00:00.000	Alumni Networking	0	Career	Meet alumni in your field

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

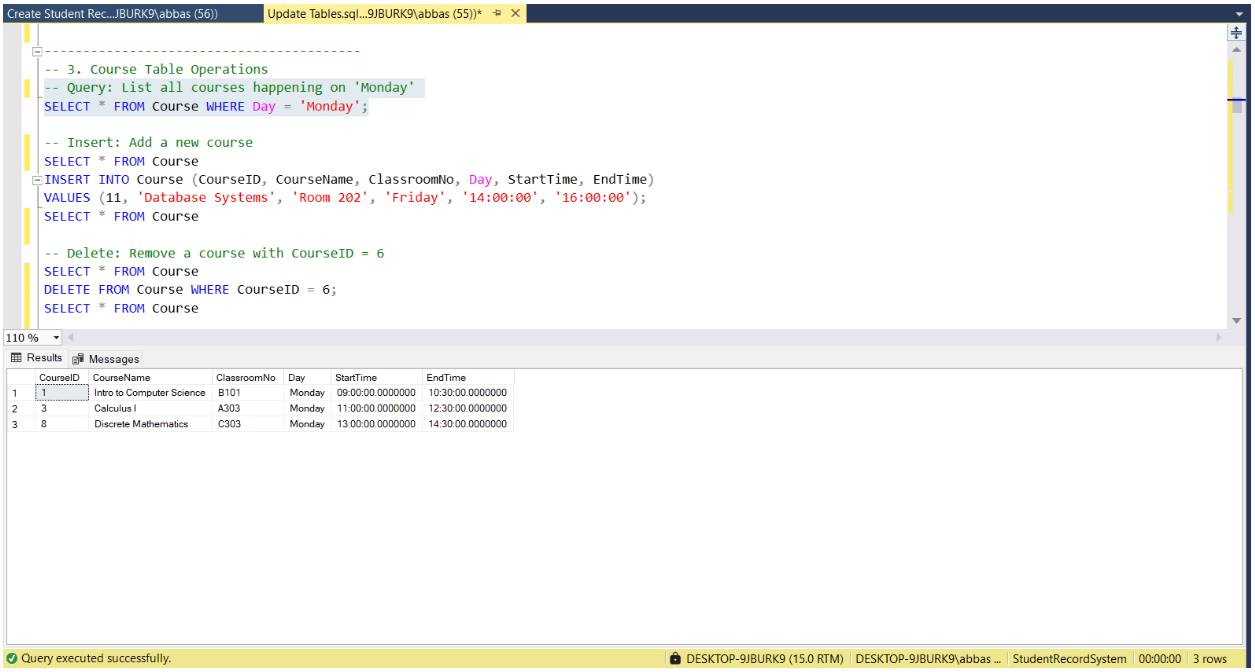
```
VALUES (11, '2024-10-01 12:00:00', 'Coding Workshop', 0, 'Workshop', 'Introduction to Python');
SELECT * FROM Event

-- Delete: Remove an event with EventID = 11
SELECT * FROM Event
DELETE FROM Event WHERE EventID = 11;
SELECT * FROM Event

-- Update: Modify the EventName for EventID = 2
SELECT * FROM Event
UPDATE Event SET EventName = 'Annual Meeting' WHERE EventID = 2;
SELECT * FROM Event

-----------------------------------------
-- 3. Course Table Operations
-- Query: List all courses happening on 'Monday'
SELECT * FROM Course WHERE Day = 'Monday';
```

	EventID	Date	EventName	RSVPComplete	EventType	Description
1	1	2024-09-01 18:00:00.000	Welcome Party	1	Social	Welcome to campus!
2	2	2024-09-05 12:00:00.000	Orientation	1	Informational	New student orientation
3	3	2024-10-01 14:00:00.000	Midterm Study Session	0	Academic	Group study for midterms
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome students
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness
8	8	2024-10-20 13:00:00.000	Tech Workshop	0	Academic	Learn new tech skills

	EventID	Date	EventName	RSVPComplete	EventType	Description
1	1	2024-09-01 18:00:00.000	Welcome Party	1	Social	Welcome to campus!
2	2	2024-09-05 12:00:00.000	Annual Meeting	1	Informational	New student orientation
3	3	2024-10-01 14:00:00.000	Midterm Study Session	0	Academic	Group study for midterms
4	4	2024-11-10 17:00:00.000	Career Fair	1	Career	Meet potential employers
5	5	2024-12-15 18:00:00.000	Holiday Party	0	Social	Celebrate the holidays
6	6	2024-08-25 15:00:00.000	Welcome Back BBQ	1	Social	BBQ to welcome stude...
7	7	2024-09-10 16:00:00.000	Health Workshop	0	Health	Mental health awareness

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 10 rows

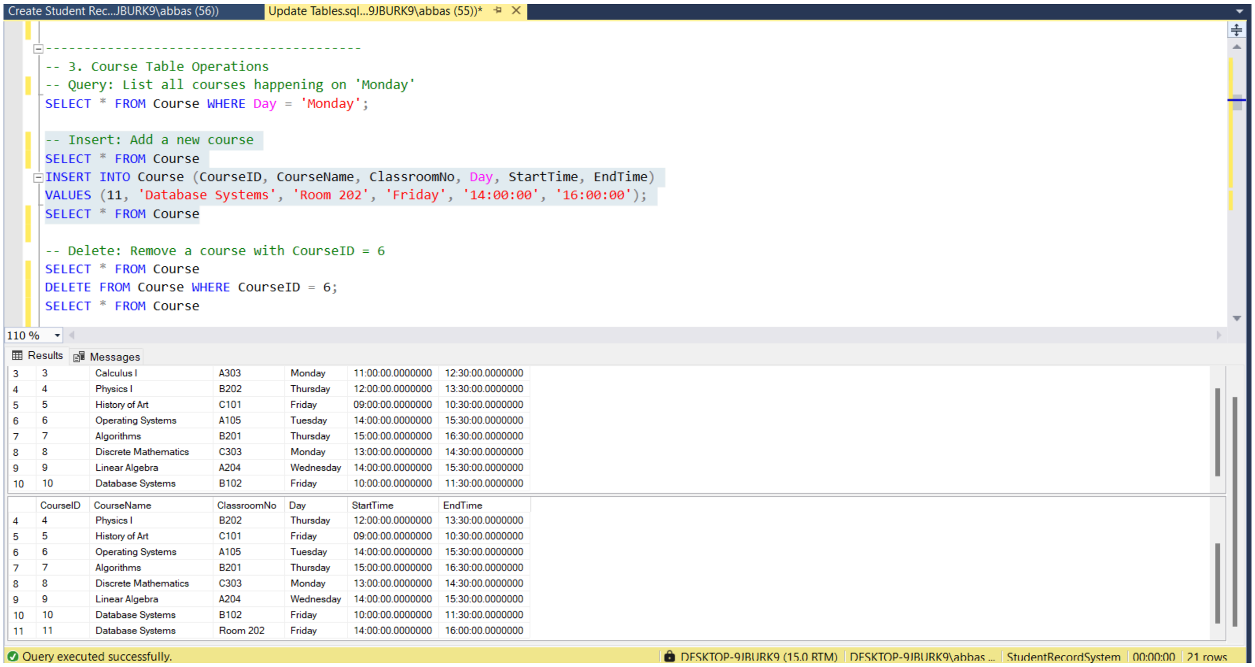
3. Course Table


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-----------------------------------------
-- 3. Course Table Operations
-- Query: List all courses happening on 'Monday'
SELECT * FROM Course WHERE Day = 'Monday';

-- Insert: Add a new course
SELECT * FROM Course
INSERT INTO Course (CourseID, CourseName, ClassroomNo, Day, StartTime, EndTime)
VALUES (11, 'Database Systems', 'Room 202', 'Friday', '14:00:00', '16:00:00');
SELECT * FROM Course

-- Delete: Remove a course with CourseID = 6
SELECT * FROM Course
DELETE FROM Course WHERE CourseID = 6;
SELECT * FROM Course
```

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
1	1	Intro to Computer Science	B101	Monday	09:00:00.0000000	10:30:00.0000000
2	3	Calculus I	A303	Monday	11:00:00.0000000	12:30:00.0000000
3	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 3 rows

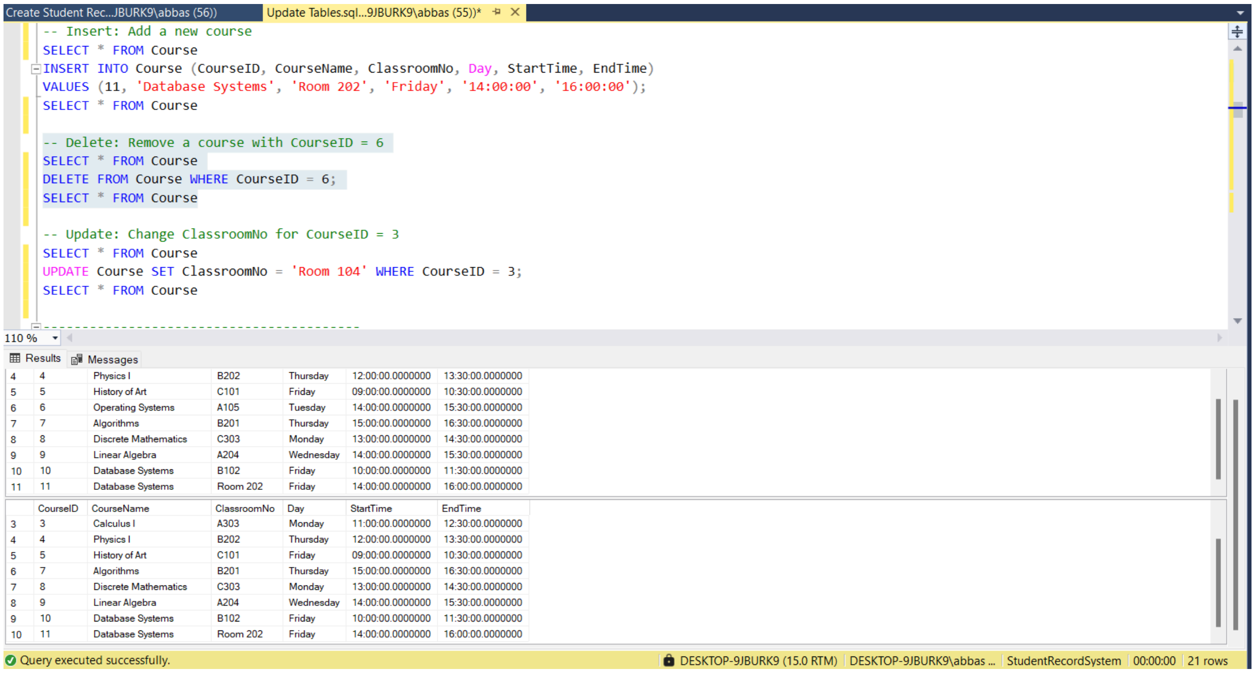

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-----------------------------------------
-- 3. Course Table Operations
-- Query: List all courses happening on 'Monday'
SELECT * FROM Course WHERE Day = 'Monday';

-- Insert: Add a new course
SELECT * FROM Course
INSERT INTO Course (CourseID, CourseName, ClassroomNo, Day, StartTime, EndTime)
VALUES (11, 'Database Systems', 'Room 202', 'Friday', '14:00:00', '16:00:00');
SELECT * FROM Course

-- Delete: Remove a course with CourseID = 6
SELECT * FROM Course
DELETE FROM Course WHERE CourseID = 6;
SELECT * FROM Course
```

3	3	Calculus I	A303	Monday	11:00:00.0000000	12:30:00.0000000
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	6	Operating Systems	A105	Tuesday	14:00:00.0000000	15:30:00.0000000
7	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
8	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
9	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000
10	10	Database Systems	B102	Friday	10:00:00.0000000	11:30:00.0000000

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	6	Operating Systems	A105	Tuesday	14:00:00.0000000	15:30:00.0000000
7	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
8	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
9	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000
10	10	Database Systems	B102	Friday	10:00:00.0000000	11:30:00.0000000
11	11	Database Systems	Room 202	Friday	14:00:00.0000000	16:00:00.0000000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows

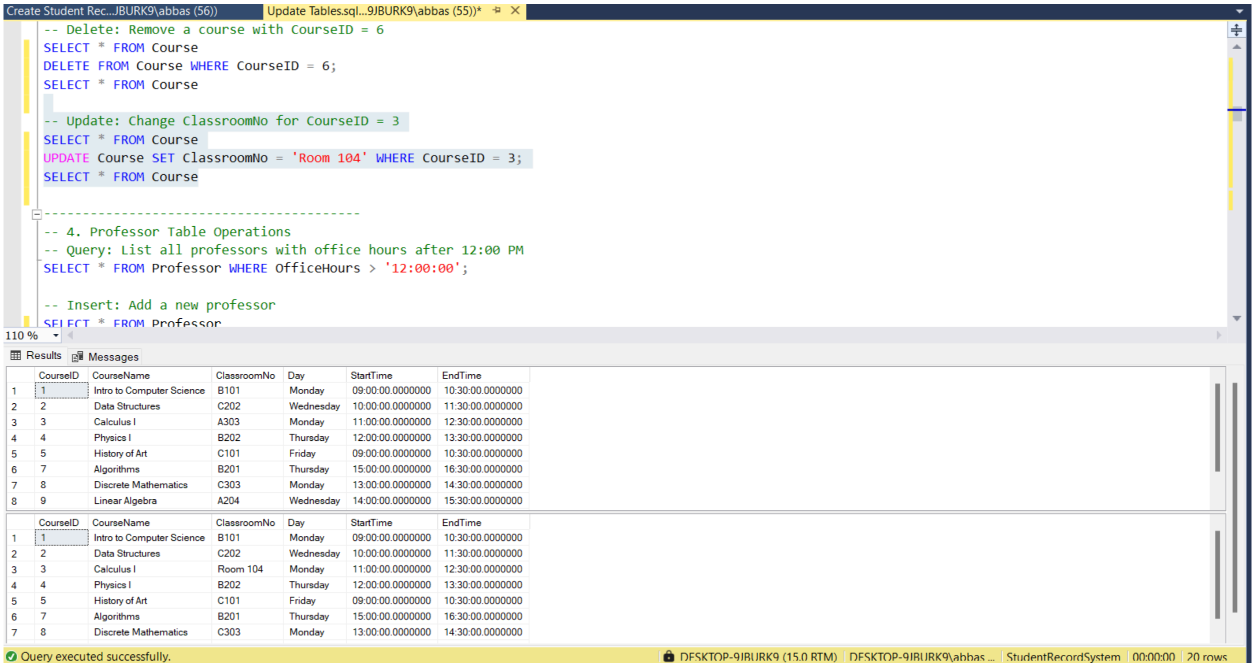

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-- Insert: Add a new course
SELECT * FROM Course
INSERT INTO Course (CourseID, CourseName, ClassroomNo, Day, StartTime, EndTime)
VALUES (11, 'Database Systems', 'Room 202', 'Friday', '14:00:00', '16:00:00');
SELECT * FROM Course

-- Delete: Remove a course with CourseID = 6
SELECT * FROM Course
DELETE FROM Course WHERE CourseID = 6;
SELECT * FROM Course

-- Update: Change ClassroomNo for CourseID = 3
SELECT * FROM Course
UPDATE Course SET ClassroomNo = 'Room 104' WHERE CourseID = 3;
SELECT * FROM Course

----------------------------------------
```

4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	6	Operating Systems	A105	Tuesday	14:00:00.0000000	15:30:00.0000000
7	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
8	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
9	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000
10	10	Database Systems	B102	Friday	10:00:00.0000000	11:30:00.0000000
11	11	Database Systems	Room 202	Friday	14:00:00.0000000	16:00:00.0000000

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
3	3	Calculus I	A303	Monday	11:00:00.0000000	12:30:00.0000000
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
7	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
8	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000
9	10	Database Systems	B102	Friday	10:00:00.0000000	11:30:00.0000000
10	11	Database Systems	Room 202	Friday	14:00:00.0000000	16:00:00.0000000

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

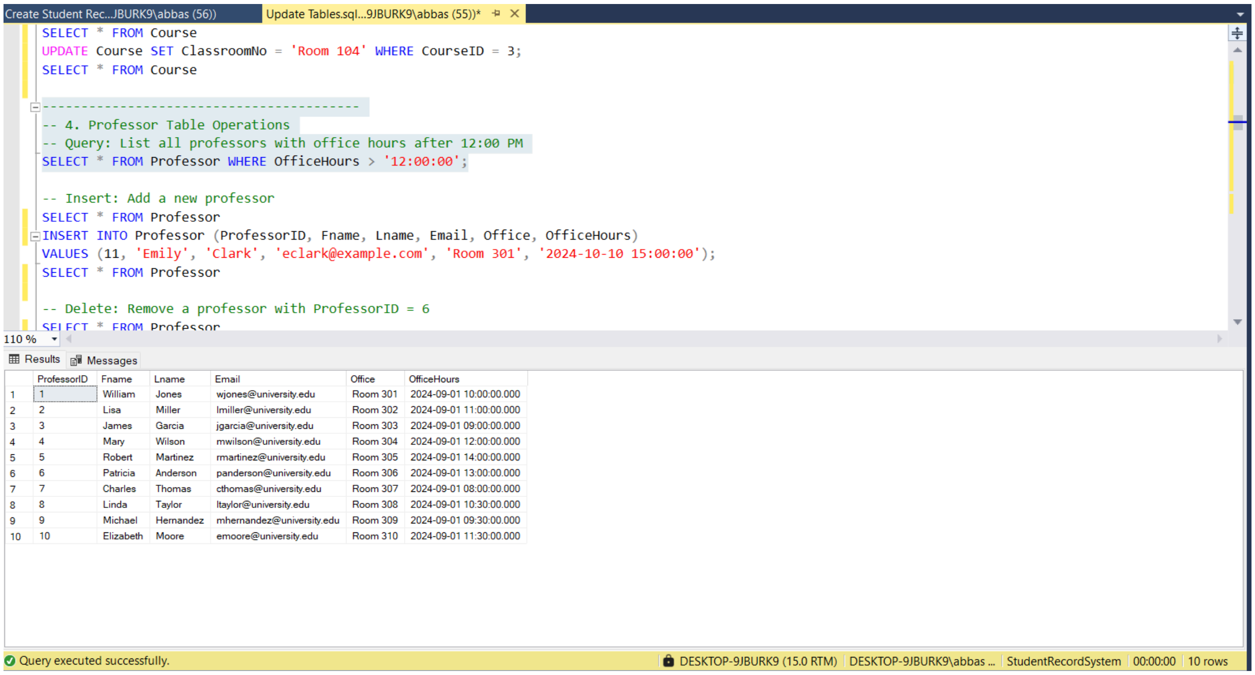
```sql
-- Delete: Remove a course with CourseID = 6
SELECT * FROM Course
DELETE FROM Course WHERE CourseID = 6;
SELECT * FROM Course

-- Update: Change ClassroomNo for CourseID = 3
SELECT * FROM Course
UPDATE Course SET ClassroomNo = 'Room 104' WHERE CourseID = 3;
SELECT * FROM Course

----------------------------------------
-- 4. Professor Table Operations
-- Query: List all professors with office hours after 12:00 PM
SELECT * FROM Professor WHERE OfficeHours > '12:00:00';

-- Insert: Add a new professor
SELECT * FROM Professor
```

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
1	1	Intro to Computer Science	B101	Monday	09:00:00.0000000	10:30:00.0000000
2	2	Data Structures	C202	Wednesday	10:00:00.0000000	11:30:00.0000000
3	3	Calculus I	A303	Monday	11:00:00.0000000	12:30:00.0000000
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
7	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000
8	9	Linear Algebra	A204	Wednesday	14:00:00.0000000	15:30:00.0000000

	CourseID	CourseName	ClassroomNo	Day	StartTime	EndTime
1	1	Intro to Computer Science	B101	Monday	09:00:00.0000000	10:30:00.0000000
2	2	Data Structures	C202	Wednesday	10:00:00.0000000	11:30:00.0000000
3	3	Calculus I	Room 104	Monday	11:00:00.0000000	12:30:00.0000000
4	4	Physics I	B202	Thursday	12:00:00.0000000	13:30:00.0000000
5	5	History of Art	C101	Friday	09:00:00.0000000	10:30:00.0000000
6	7	Algorithms	B201	Thursday	15:00:00.0000000	16:30:00.0000000
7	8	Discrete Mathematics	C303	Monday	13:00:00.0000000	14:30:00.0000000

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 20 rows

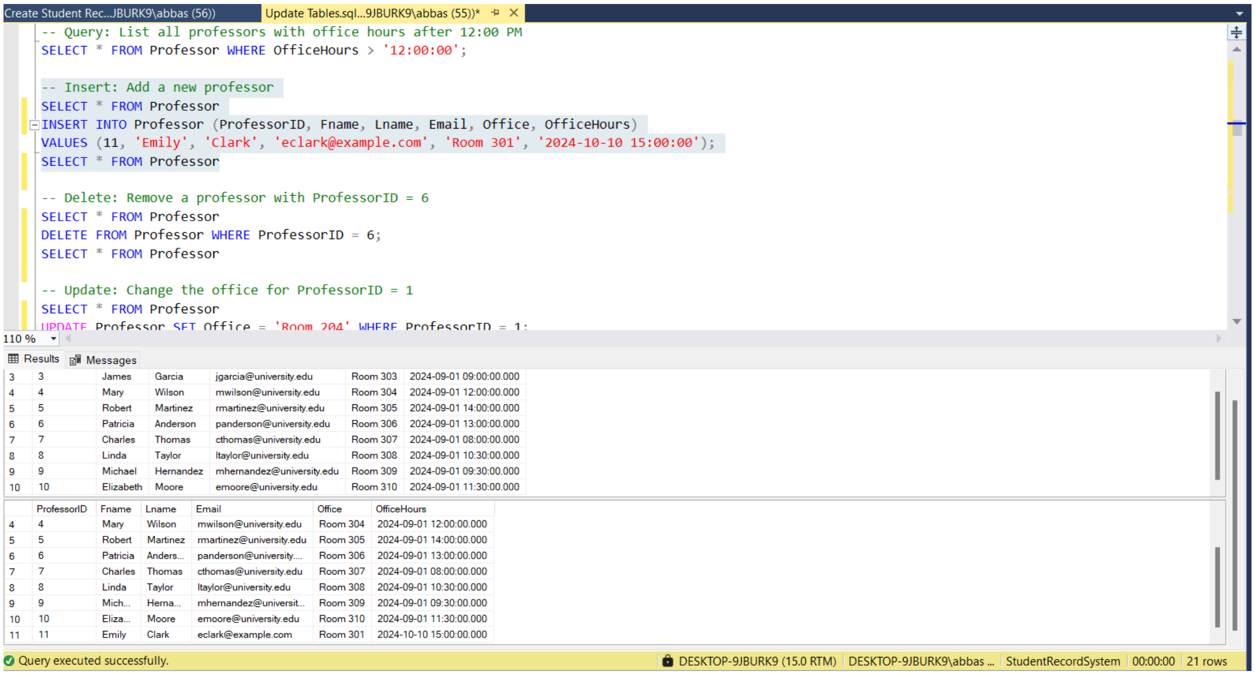
4. Professor Table


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*

```sql
SELECT * FROM Course
UPDATE Course SET ClassroomNo = 'Room 104' WHERE CourseID = 3;
SELECT * FROM Course

-----------------------------------------
-- 4. Professor Table Operations
-- Query: List all professors with office hours after 12:00 PM
SELECT * FROM Professor WHERE OfficeHours > '12:00:00';

-- Insert: Add a new professor
SELECT * FROM Professor
INSERT INTO Professor (ProfessorID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Emily', 'Clark', 'eclark@example.com', 'Room 301', '2024-10-10 15:00:00');
SELECT * FROM Professor

-- Delete: Remove a professor with ProfessorID = 6
SELECT * FROM Professor
```

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
1	1	William	Jones	wjones@university.edu	Room 301	2024-09-01 10:00:00.000
2	2	Lisa	Miller	lmiller@university.edu	Room 302	2024-09-01 11:00:00.000
3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	6	Patricia	Anderson	panderson@university.edu	Room 306	2024-09-01 13:00:00.000
7	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
8	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
9	9	Michael	Hernandez	mhernandez@university.edu	Room 309	2024-09-01 09:30:00.000
10	10	Elizabeth	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 10 rows

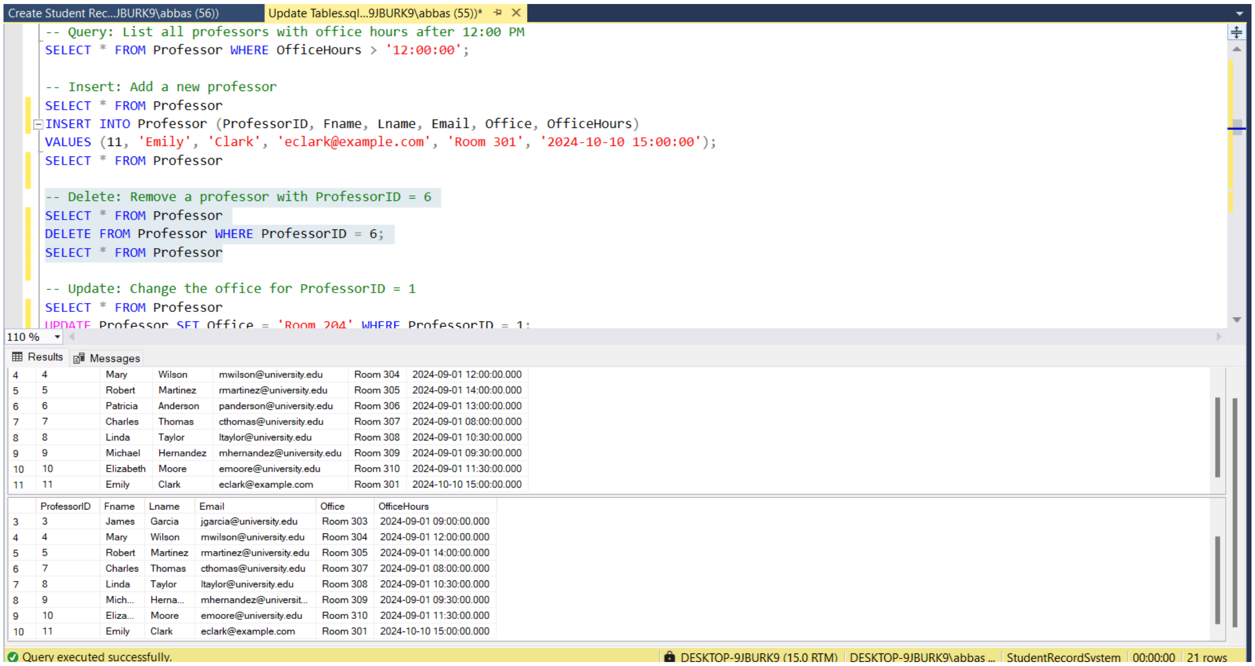

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*

```sql
-- Query: List all professors with office hours after 12:00 PM
SELECT * FROM Professor WHERE OfficeHours > '12:00:00';

-- Insert: Add a new professor
SELECT * FROM Professor
INSERT INTO Professor (ProfessorID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Emily', 'Clark', 'eclark@example.com', 'Room 301', '2024-10-10 15:00:00');
SELECT * FROM Professor

-- Delete: Remove a professor with ProfessorID = 6
SELECT * FROM Professor
DELETE FROM Professor WHERE ProfessorID = 6;
SELECT * FROM Professor

-- Update: Change the office for ProfessorID = 1
SELECT * FROM Professor
UPDATE Professor SET Office = 'Room 204' WHERE ProfessorID = 1;
```

3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	6	Patricia	Anderson	panderson@university.edu	Room 306	2024-09-01 13:00:00.000
7	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
8	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
9	9	Michael	Hernandez	mhernandez@university.edu	Room 309	2024-09-01 09:30:00.000
10	10	Elizabeth	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	6	Patricia	Anders...	panderson@university....	Room 306	2024-09-01 13:00:00.000
7	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
8	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
9	9	Mich...	Herna...	mhernandez@universit...	Room 309	2024-09-01 09:30:00.000
10	10	Eliza...	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000
11	11	Emily	Clark	eclark@example.com	Room 301	2024-10-10 15:00:00.000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows

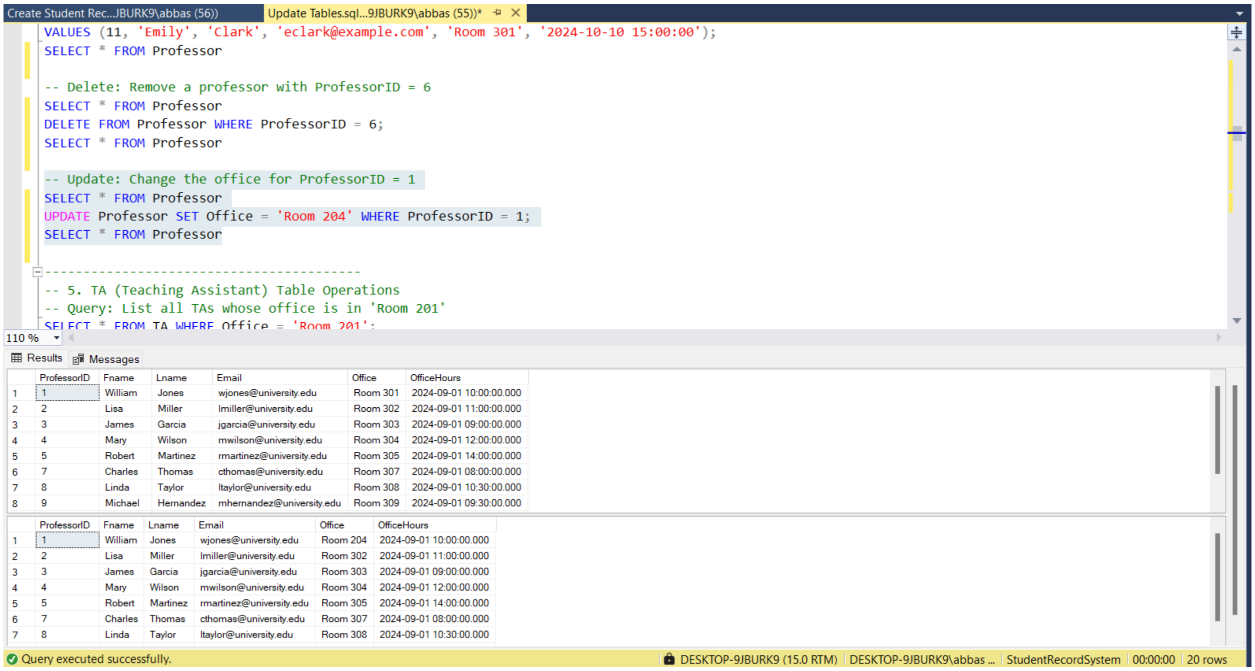

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-- Query: List all professors with office hours after 12:00 PM
SELECT * FROM Professor WHERE OfficeHours > '12:00:00';

-- Insert: Add a new professor
SELECT * FROM Professor
INSERT INTO Professor (ProfessorID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Emily', 'Clark', 'eclark@example.com', 'Room 301', '2024-10-10 15:00:00');
SELECT * FROM Professor

-- Delete: Remove a professor with ProfessorID = 6
SELECT * FROM Professor
DELETE FROM Professor WHERE ProfessorID = 6;
SELECT * FROM Professor

-- Update: Change the office for ProfessorID = 1
SELECT * FROM Professor
UPDATE Professor SET Office = 'Room 204' WHERE ProfessorID = 1;
```

4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	6	Patricia	Anderson	panderson@university.edu	Room 306	2024-09-01 13:00:00.000
7	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
8	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
9	9	Michael	Hernandez	mhernandez@university.edu	Room 309	2024-09-01 09:30:00.000
10	10	Elizabeth	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000
11	11	Emily	Clark	eclark@example.com	Room 301	2024-10-10 15:00:00.000

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
7	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
8	9	Mich...	Herna...	mhernandez@universit...	Room 309	2024-09-01 09:30:00.000
9	10	Eliza...	Moore	emoore@university.edu	Room 310	2024-09-01 11:30:00.000
10	11	Emily	Clark	eclark@example.com	Room 301	2024-10-10 15:00:00.000

```
Query executed successfully.    DESKTOP-9JBURK9 (15.0 RTM)  DESKTOP-9JBURK9\abbas ...  StudentRecordSystem  00:00:00  21 rows
```


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

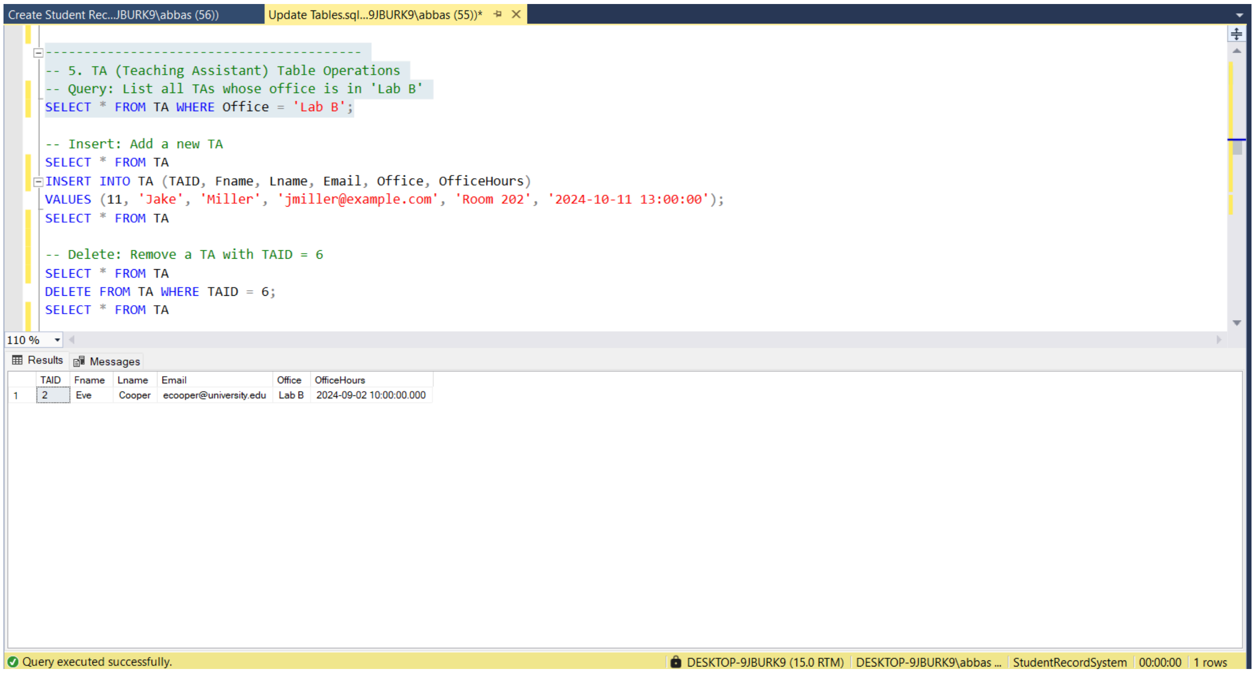
```sql
VALUES (11, 'Emily', 'Clark', 'eclark@example.com', 'Room 301', '2024-10-10 15:00:00');
SELECT * FROM Professor

-- Delete: Remove a professor with ProfessorID = 6
SELECT * FROM Professor
DELETE FROM Professor WHERE ProfessorID = 6;
SELECT * FROM Professor

-- Update: Change the office for ProfessorID = 1
SELECT * FROM Professor
UPDATE Professor SET Office = 'Room 204' WHERE ProfessorID = 1;
SELECT * FROM Professor

----------------------------------------
-- 5. TA (Teaching Assistant) Table Operations
-- Query: List all TAs whose office is in 'Room 201'
SELECT * FROM TA WHERE Office = 'Room 201';
```

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
1	1	William	Jones	wjones@university.edu	Room 301	2024-09-01 10:00:00.000
2	2	Lisa	Miller	lmiller@university.edu	Room 302	2024-09-01 11:00:00.000
3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
7	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000
8	9	Michael	Hernandez	mhernandez@university.edu	Room 309	2024-09-01 09:30:00.000

	ProfessorID	Fname	Lname	Email	Office	OfficeHours
1	1	William	Jones	wjones@university.edu	Room 204	2024-09-01 10:00:00.000
2	2	Lisa	Miller	lmiller@university.edu	Room 302	2024-09-01 11:00:00.000
3	3	James	Garcia	jgarcia@university.edu	Room 303	2024-09-01 09:00:00.000
4	4	Mary	Wilson	mwilson@university.edu	Room 304	2024-09-01 12:00:00.000
5	5	Robert	Martinez	rmartinez@university.edu	Room 305	2024-09-01 14:00:00.000
6	7	Charles	Thomas	cthomas@university.edu	Room 307	2024-09-01 08:00:00.000
7	8	Linda	Taylor	ltaylor@university.edu	Room 308	2024-09-01 10:30:00.000

```
Query executed successfully.    DESKTOP-9JBURK9 (15.0 RTM)  DESKTOP-9JBURK9\abbas ...  StudentRecordSystem  00:00:00  20 rows
```

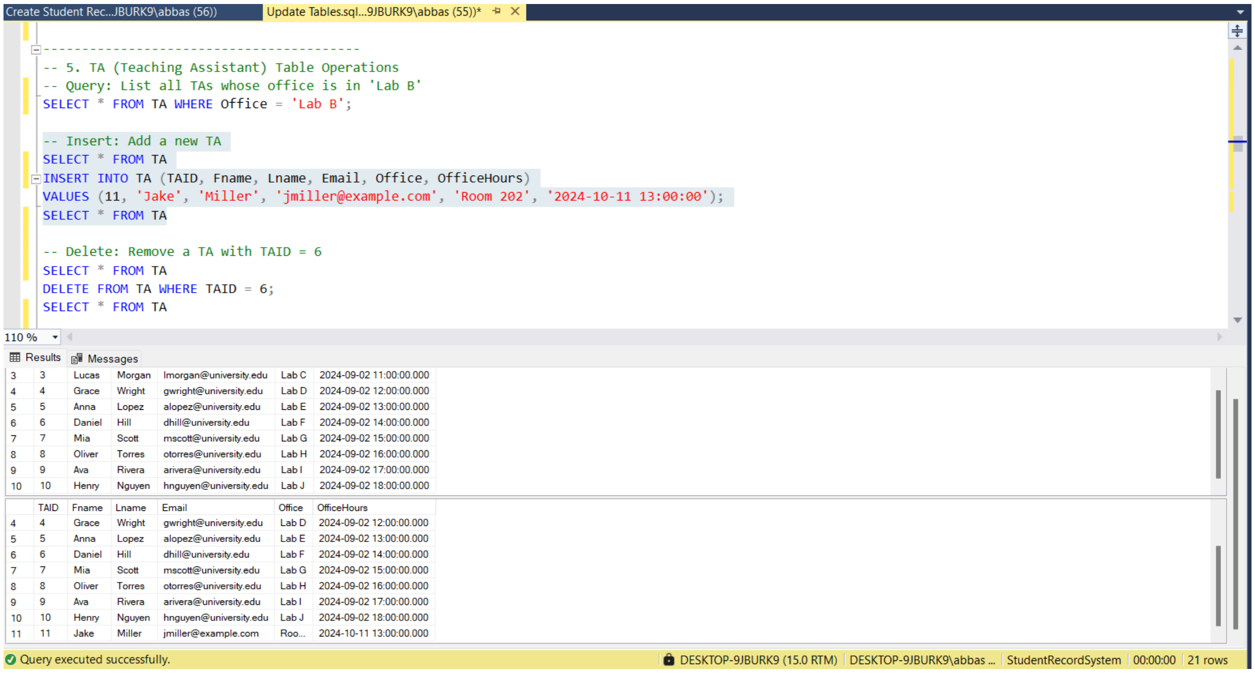
5. TA Table


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
----------------------------------------
-- 5. TA (Teaching Assistant) Table Operations
-- Query: List all TAs whose office is in 'Lab B'
SELECT * FROM TA WHERE Office = 'Lab B';

-- Insert: Add a new TA
SELECT * FROM TA
INSERT INTO TA (TAID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Jake', 'Miller', 'jmiller@example.com', 'Room 202', '2024-10-11 13:00:00');
SELECT * FROM TA

-- Delete: Remove a TA with TAID = 6
SELECT * FROM TA
DELETE FROM TA WHERE TAID = 6;
SELECT * FROM TA
```

	TAID	Fname	Lname	Email	Office	OfficeHours
1	2	Eve	Cooper	ecooper@university.edu	Lab B	2024-09-02 10:00:00.000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 1 rows

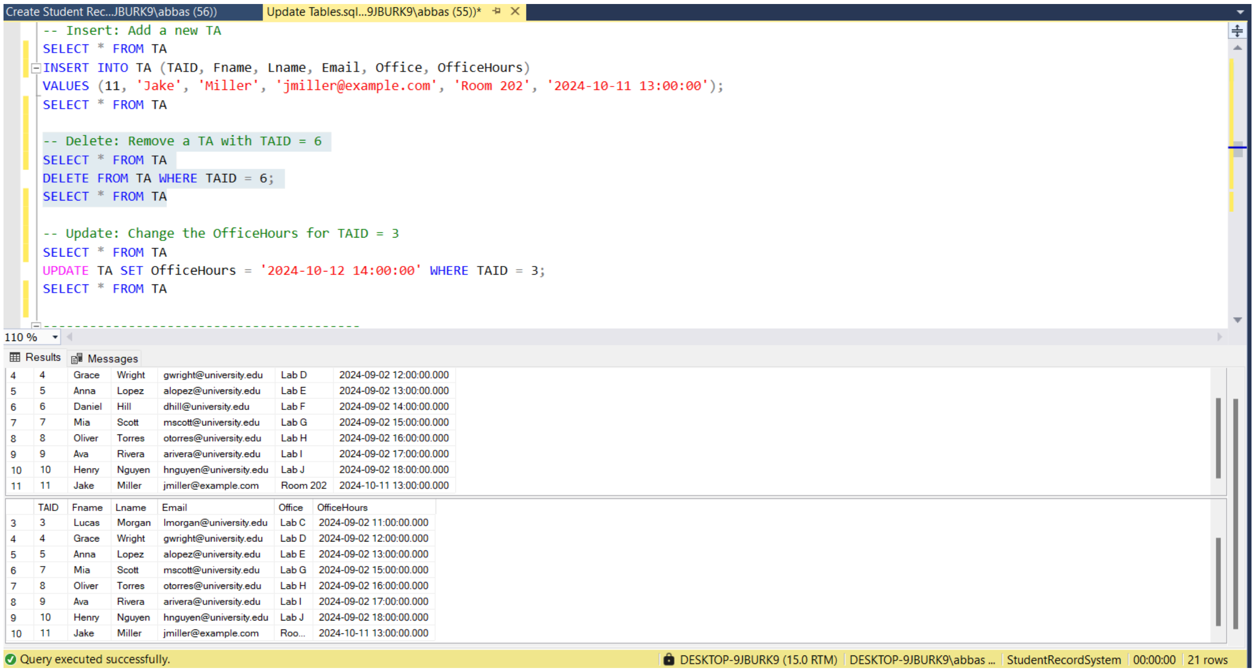

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
----------------------------------------
-- 5. TA (Teaching Assistant) Table Operations
-- Query: List all TAs whose office is in 'Lab B'
SELECT * FROM TA WHERE Office = 'Lab B';

-- Insert: Add a new TA
SELECT * FROM TA
INSERT INTO TA (TAID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Jake', 'Miller', 'jmiller@example.com', 'Room 202', '2024-10-11 13:00:00');
SELECT * FROM TA

-- Delete: Remove a TA with TAID = 6
SELECT * FROM TA
DELETE FROM TA WHERE TAID = 6;
SELECT * FROM TA
```

3	3	Lucas	Morgan	lmorgan@university.edu	Lab C	2024-09-02 11:00:00.000
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	6	Daniel	Hill	dhill@university.edu	Lab F	2024-09-02 14:00:00.000
7	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
8	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
9	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000
10	10	Henry	Nguyen	hnguyen@university.edu	Lab J	2024-09-02 18:00:00.000

	TAID	Fname	Lname	Email	Office	OfficeHours
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	6	Daniel	Hill	dhill@university.edu	Lab F	2024-09-02 14:00:00.000
7	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
8	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
9	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000
10	10	Henry	Nguyen	hnguyen@university.edu	Lab J	2024-09-02 18:00:00.000
11	11	Jake	Miller	jmiller@example.com	Roo...	2024-10-11 13:00:00.000

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows



Create Student Rec...JBURK9\abbas (56))   Update Tables.sql...9JBURK9\abbas (55))*

```sql
-- Insert: Add a new TA
SELECT * FROM TA
INSERT INTO TA (TAID, Fname, Lname, Email, Office, OfficeHours)
VALUES (11, 'Jake', 'Miller', 'jmiller@example.com', 'Room 202', '2024-10-11 13:00:00');
SELECT * FROM TA

-- Delete: Remove a TA with TAID = 6
SELECT * FROM TA
DELETE FROM TA WHERE TAID = 6;
SELECT * FROM TA

-- Update: Change the OfficeHours for TAID = 3
SELECT * FROM TA
UPDATE TA SET OfficeHours = '2024-10-12 14:00:00' WHERE TAID = 3;
SELECT * FROM TA

----------------------------------------
```

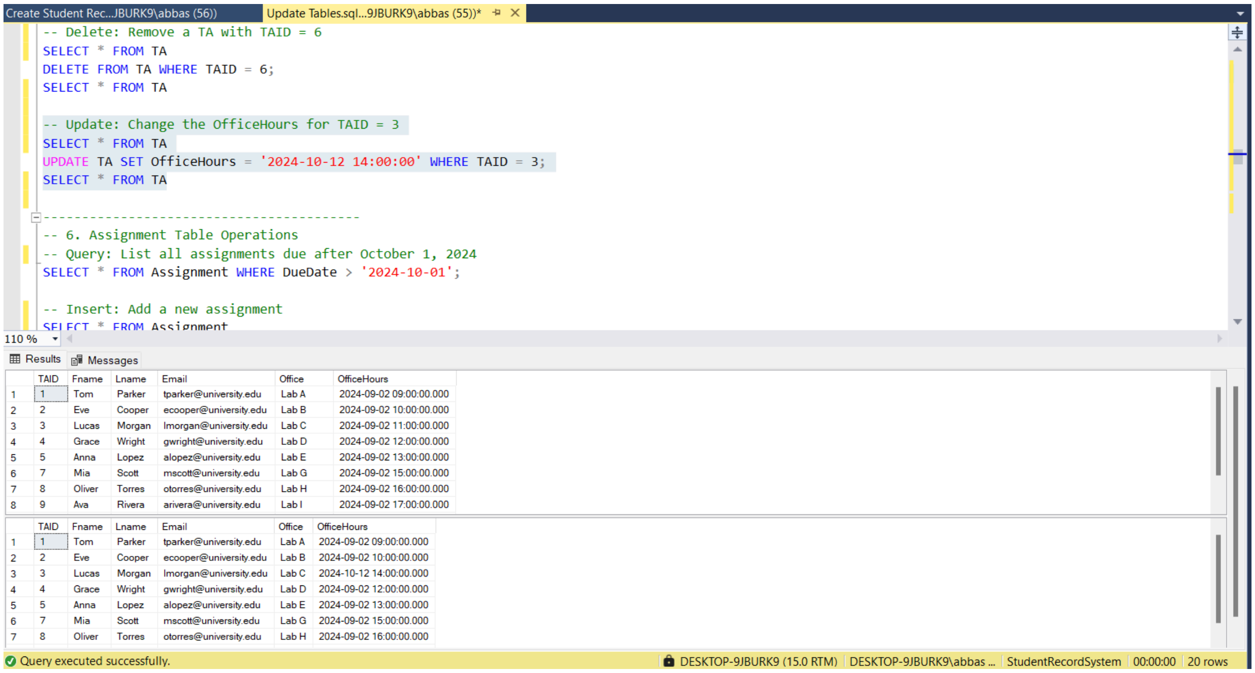
110 %

Results   Messages

4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	6	Daniel	Hill	dhill@university.edu	Lab F	2024-09-02 14:00:00.000
7	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
8	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
9	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000
10	10	Henry	Nguyen	hnguyen@university.edu	Lab J	2024-09-02 18:00:00.000
11	11	Jake	Miller	jmiller@example.com	Room 202	2024-10-11 13:00:00.000

	TAID	Fname	Lname	Email	Office	OfficeHours
3	3	Lucas	Morgan	lmorgan@university.edu	Lab C	2024-09-02 11:00:00.000
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
7	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
8	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000
9	10	Henry	Nguyen	hnguyen@university.edu	Lab J	2024-09-02 18:00:00.000
10	11	Jake	Miller	jmiller@example.com	Roo...	2024-10-11 13:00:00.000

Query executed successfully.   DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 21 rows



Create Student Rec...JBURK9\abbas (56))   Update Tables.sql...9JBURK9\abbas (55))*

```sql
-- Delete: Remove a TA with TAID = 6
SELECT * FROM TA
DELETE FROM TA WHERE TAID = 6;
SELECT * FROM TA

-- Update: Change the OfficeHours for TAID = 3
SELECT * FROM TA
UPDATE TA SET OfficeHours = '2024-10-12 14:00:00' WHERE TAID = 3;
SELECT * FROM TA

----------------------------------------
-- 6. Assignment Table Operations
-- Query: List all assignments due after October 1, 2024
SELECT * FROM Assignment WHERE DueDate > '2024-10-01';

-- Insert: Add a new assignment
SELECT * FROM Assignment
```

110 %

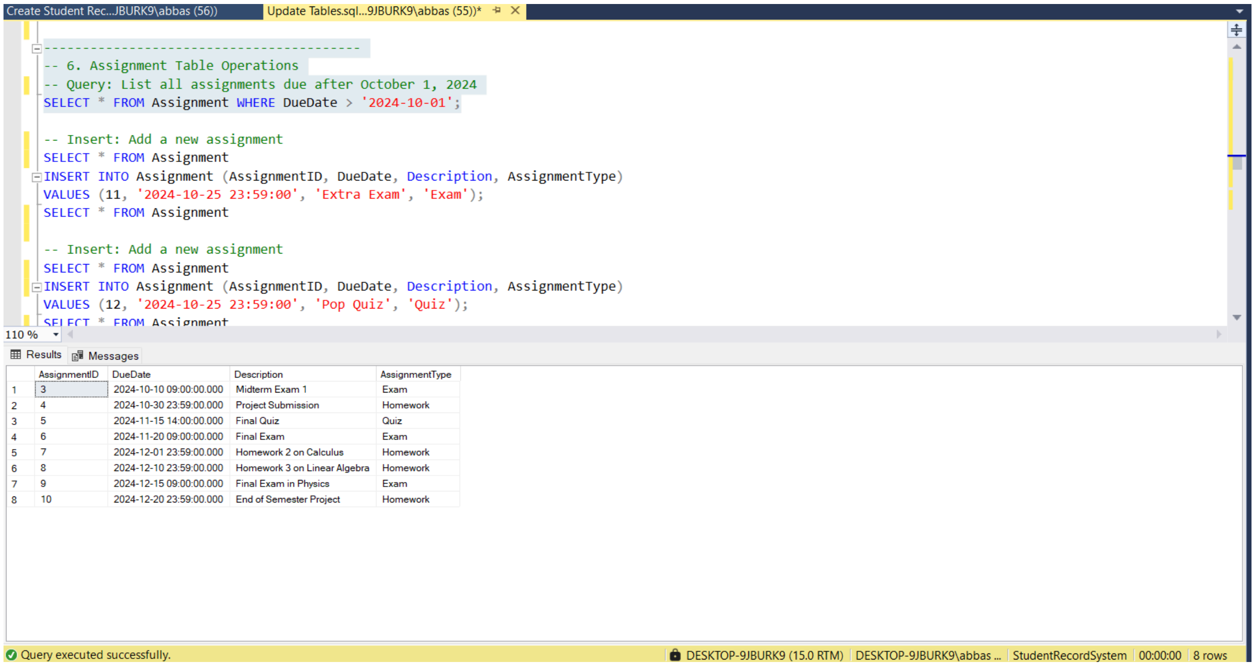
Results   Messages

	TAID	Fname	Lname	Email	Office	OfficeHours
1	1	Tom	Parker	tparker@university.edu	Lab A	2024-09-02 09:00:00.000
2	2	Eve	Cooper	ecooper@university.edu	Lab B	2024-09-02 10:00:00.000
3	3	Lucas	Morgan	lmorgan@university.edu	Lab C	2024-09-02 11:00:00.000
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
7	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000
8	9	Ava	Rivera	arivera@university.edu	Lab I	2024-09-02 17:00:00.000

	TAID	Fname	Lname	Email	Office	OfficeHours
1	1	Tom	Parker	tparker@university.edu	Lab A	2024-09-02 09:00:00.000
2	2	Eve	Cooper	ecooper@university.edu	Lab B	2024-09-02 10:00:00.000
3	3	Lucas	Morgan	lmorgan@university.edu	Lab C	2024-10-12 14:00:00.000
4	4	Grace	Wright	gwright@university.edu	Lab D	2024-09-02 12:00:00.000
5	5	Anna	Lopez	alopez@university.edu	Lab E	2024-09-02 13:00:00.000
6	7	Mia	Scott	mscott@university.edu	Lab G	2024-09-02 15:00:00.000
7	8	Oliver	Torres	otorres@university.edu	Lab H	2024-09-02 16:00:00.000

Query executed successfully.   DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 20 rows

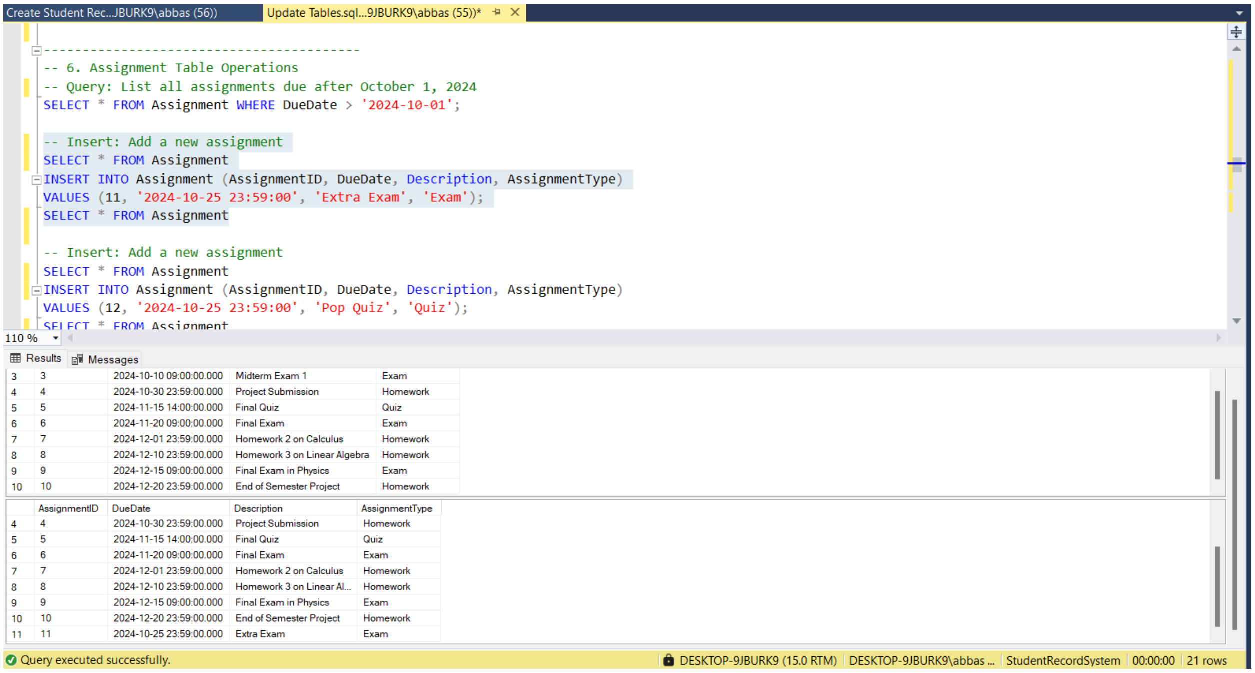
6. Assignment Table


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages
DESKTOP-9JBURK9 (15.0 RTM)
DESKTOP-9JBURK9\abbas ...
StudentRecordSystem
00:00:00
8 rows

```sql
-----------------------------------------
-- 6. Assignment Table Operations
-- Query: List all assignments due after October 1, 2024
SELECT * FROM Assignment WHERE DueDate > '2024-10-01';

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (11, '2024-10-25 23:59:00', 'Extra Exam', 'Exam');
SELECT * FROM Assignment

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (12, '2024-10-25 23:59:00', 'Pop Quiz', 'Quiz');
SELECT * FROM Assignment
```

	AssignmentID	DueDate	Description	AssignmentType
1	3	2024-10-10 09:00:00.000	Midterm Exam 1	Exam
2	4	2024-10-30 23:59:00.000	Project Submission	Homework
3	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
4	6	2024-11-20 09:00:00.000	Final Exam	Exam
5	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
6	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
7	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
8	10	2024-12-20 23:59:00.000	End of Semester Project	Homework

✓ Query executed successfully.

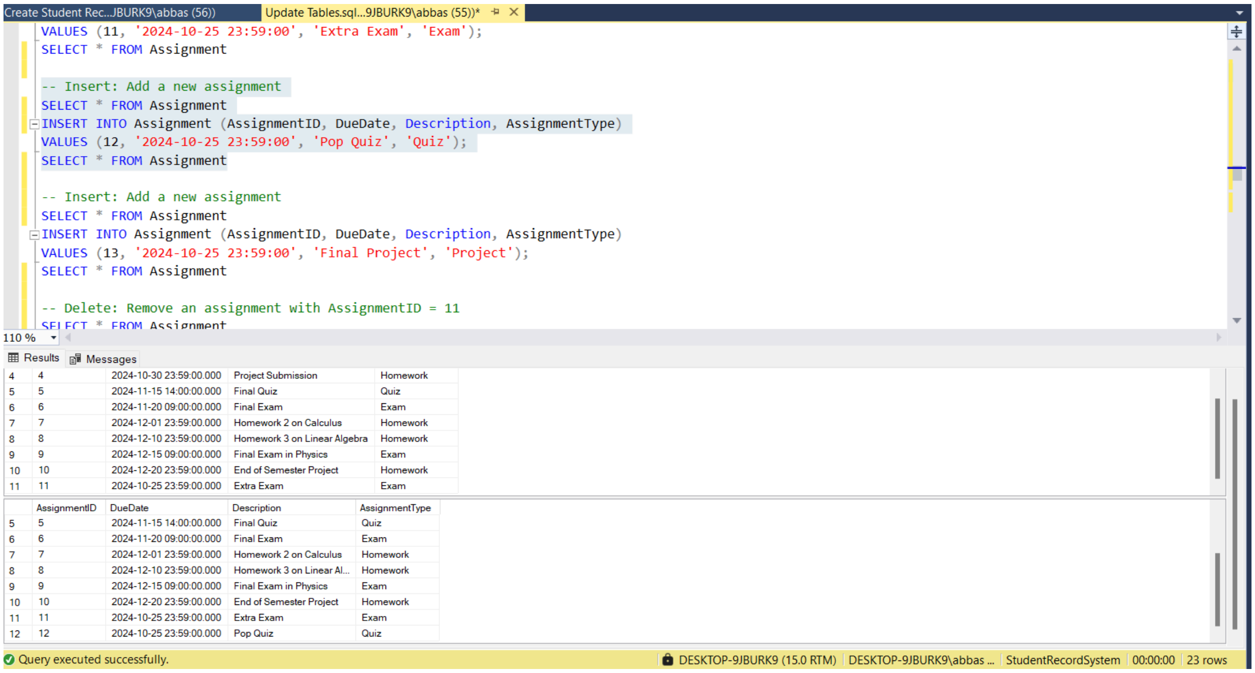

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages
DESKTOP-9JBURK9 (15.0 RTM)
DESKTOP-9JBURK9\abbas ...
StudentRecordSystem
00:00:00
21 rows

```sql
-----------------------------------------
-- 6. Assignment Table Operations
-- Query: List all assignments due after October 1, 2024
SELECT * FROM Assignment WHERE DueDate > '2024-10-01';

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (11, '2024-10-25 23:59:00', 'Extra Exam', 'Exam');
SELECT * FROM Assignment

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (12, '2024-10-25 23:59:00', 'Pop Quiz', 'Quiz');
SELECT * FROM Assignment
```

3	3	2024-10-10 09:00:00.000	Midterm Exam 1	Exam
4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework

	AssignmentID	DueDate	Description	AssignmentType
4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Al...	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam

✓ Query executed successfully.

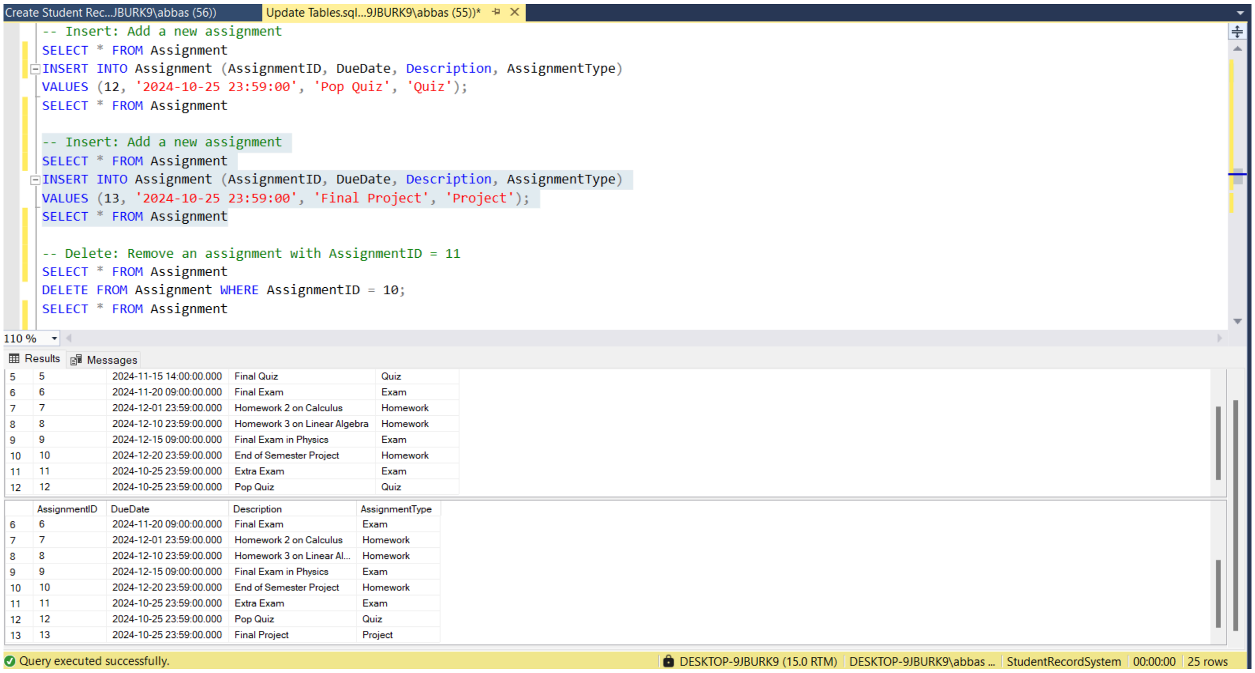

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
VALUES (11, '2024-10-25 23:59:00', 'Extra Exam', 'Exam');
SELECT * FROM Assignment

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (12, '2024-10-25 23:59:00', 'Pop Quiz', 'Quiz');
SELECT * FROM Assignment

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (13, '2024-10-25 23:59:00', 'Final Project', 'Project');
SELECT * FROM Assignment

-- Delete: Remove an assignment with AssignmentID = 11
SELECT * FROM Assignment
```

4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam

	AssignmentID	DueDate	Description	AssignmentType
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Al...	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam
12	12	2024-10-25 23:59:00.000	Pop Quiz	Quiz

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) DESKTOP-9JBURK9\abbas ... StudentRecordSystem 00:00:00 23 rows

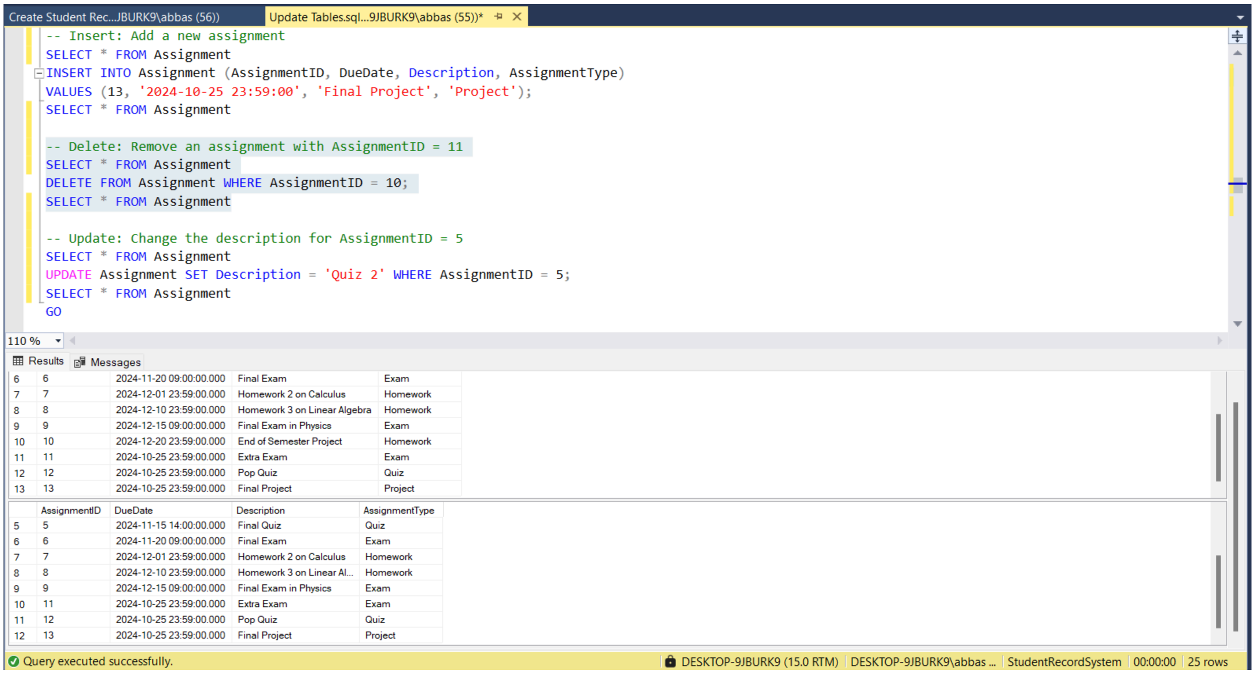

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (12, '2024-10-25 23:59:00', 'Pop Quiz', 'Quiz');
SELECT * FROM Assignment

-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (13, '2024-10-25 23:59:00', 'Final Project', 'Project');
SELECT * FROM Assignment

-- Delete: Remove an assignment with AssignmentID = 11
SELECT * FROM Assignment
DELETE FROM Assignment WHERE AssignmentID = 10;
SELECT * FROM Assignment
```

5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam
12	12	2024-10-25 23:59:00.000	Pop Quiz	Quiz

	AssignmentID	DueDate	Description	AssignmentType
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Al...	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam
12	12	2024-10-25 23:59:00.000	Pop Quiz	Quiz
13	13	2024-10-25 23:59:00.000	Final Project	Project

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) DESKTOP-9JBURK9\abbas ... StudentRecordSystem 00:00:00 25 rows



Create Student Rec...JBURK9\abbas (56))   Update Tables.sql...9JBURK9\abbas (55))*

```sql
-- Insert: Add a new assignment
SELECT * FROM Assignment
INSERT INTO Assignment (AssignmentID, DueDate, Description, AssignmentType)
VALUES (13, '2024-10-25 23:59:00', 'Final Project', 'Project');
SELECT * FROM Assignment

-- Delete: Remove an assignment with AssignmentID = 11
SELECT * FROM Assignment
DELETE FROM Assignment WHERE AssignmentID = 10;
SELECT * FROM Assignment

-- Update: Change the description for AssignmentID = 5
SELECT * FROM Assignment
UPDATE Assignment SET Description = 'Quiz 2' WHERE AssignmentID = 5;
SELECT * FROM Assignment
GO
```

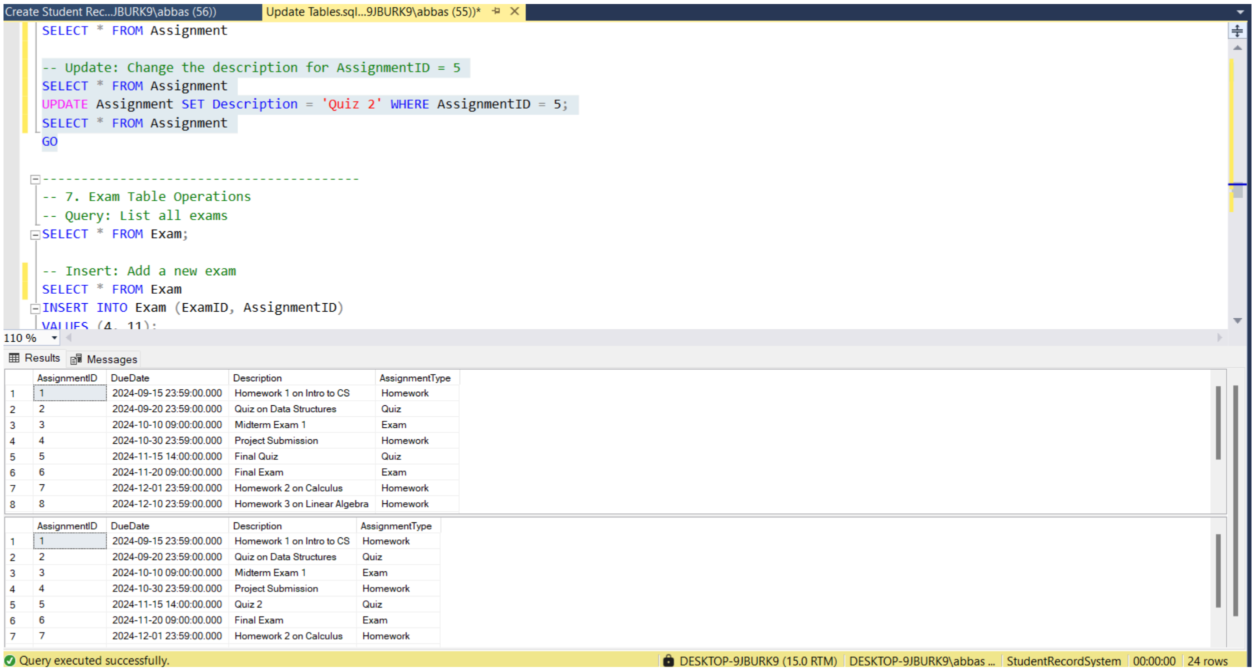
110 %

Results   Messages

6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	10	2024-12-20 23:59:00.000	End of Semester Project	Homework
11	11	2024-10-25 23:59:00.000	Extra Exam	Exam
12	12	2024-10-25 23:59:00.000	Pop Quiz	Quiz
13	13	2024-10-25 23:59:00.000	Final Project	Project

	AssignmentID	DueDate	Description	AssignmentType
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Al...	Homework
9	9	2024-12-15 09:00:00.000	Final Exam in Physics	Exam
10	11	2024-10-25 23:59:00.000	Extra Exam	Exam
11	12	2024-10-25 23:59:00.000	Pop Quiz	Quiz
12	13	2024-10-25 23:59:00.000	Final Project	Project

Query executed successfully.   DESKTOP-9JBURK9 (15.0 RTM) DESKTOP-9JBURK9\abbas ... StudentRecordSystem 00:00:00 25 rows



Create Student Rec...JBURK9\abbas (56))   Update Tables.sql...9JBURK9\abbas (55))*

```sql
SELECT * FROM Assignment

-- Update: Change the description for AssignmentID = 5
SELECT * FROM Assignment
UPDATE Assignment SET Description = 'Quiz 2' WHERE AssignmentID = 5;
SELECT * FROM Assignment
GO

-----------------------------------------
-- 7. Exam Table Operations
-- Query: List all exams
SELECT * FROM Exam;

-- Insert: Add a new exam
SELECT * FROM Exam
INSERT INTO Exam (ExamID, AssignmentID)
VALUES (4, 11);
```

110 %

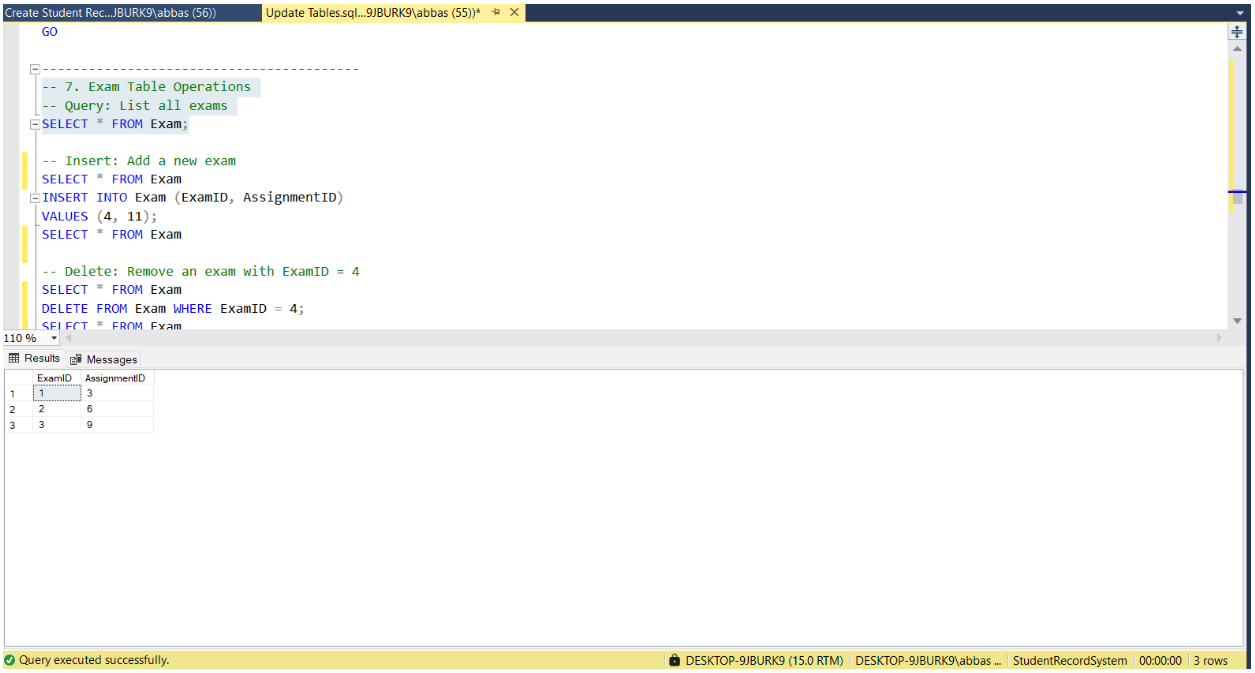
Results   Messages

	AssignmentID	DueDate	Description	AssignmentType
1	1	2024-09-15 23:59:00.000	Homework 1 on Intro to CS	Homework
2	2	2024-09-20 23:59:00.000	Quiz on Data Structures	Quiz
3	3	2024-10-10 09:00:00.000	Midterm Exam 1	Exam
4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Final Quiz	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework
8	8	2024-12-10 23:59:00.000	Homework 3 on Linear Algebra	Homework

	AssignmentID	DueDate	Description	AssignmentType
1	1	2024-09-15 23:59:00.000	Homework 1 on Intro to CS	Homework
2	2	2024-09-20 23:59:00.000	Quiz on Data Structures	Quiz
3	3	2024-10-10 09:00:00.000	Midterm Exam 1	Exam
4	4	2024-10-30 23:59:00.000	Project Submission	Homework
5	5	2024-11-15 14:00:00.000	Quiz 2	Quiz
6	6	2024-11-20 09:00:00.000	Final Exam	Exam
7	7	2024-12-01 23:59:00.000	Homework 2 on Calculus	Homework

Query executed successfully.   DESKTOP-9JBURK9 (15.0 RTM) DESKTOP-9JBURK9\abbas ... StudentRecordSystem 00:00:00 24 rows

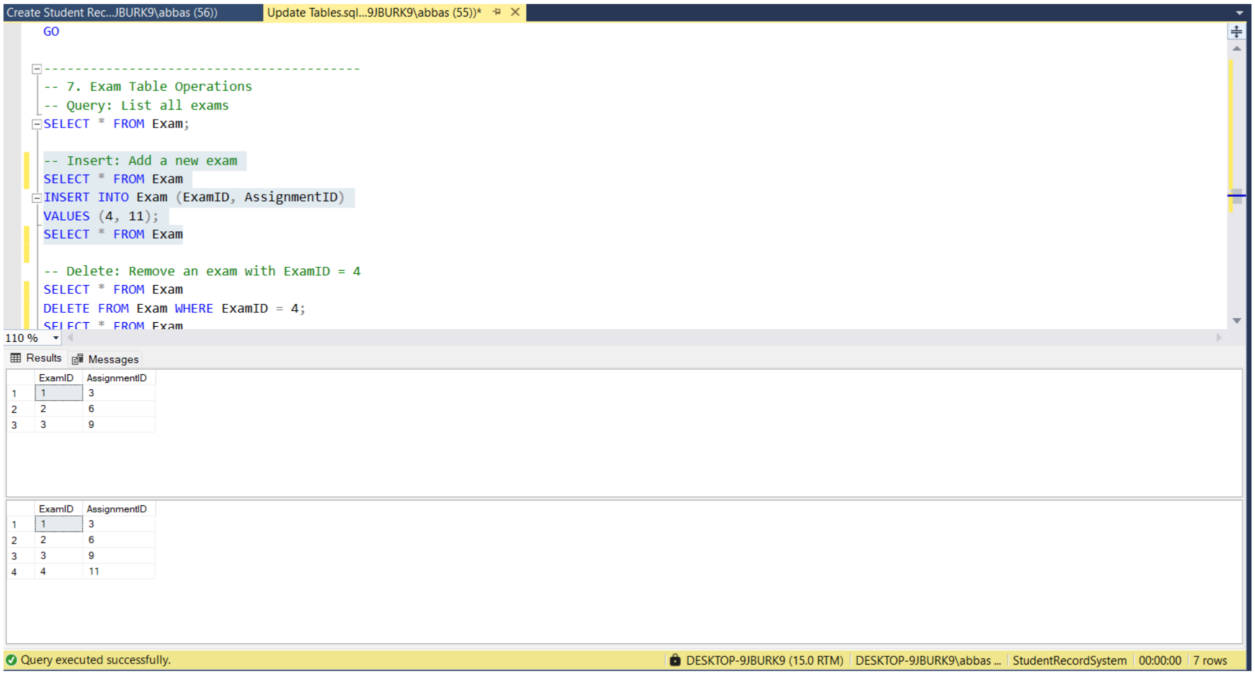
7. Exam Table


Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```
GO

------------------------------------------
-- 7. Exam Table Operations
-- Query: List all exams
SELECT * FROM Exam;

-- Insert: Add a new exam
SELECT * FROM Exam
INSERT INTO Exam (ExamID, AssignmentID)
VALUES (4, 11);
SELECT * FROM Exam

-- Delete: Remove an exam with ExamID = 4
SELECT * FROM Exam
DELETE FROM Exam WHERE ExamID = 4;
SELECT * FROM Exam
```

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 3 rows

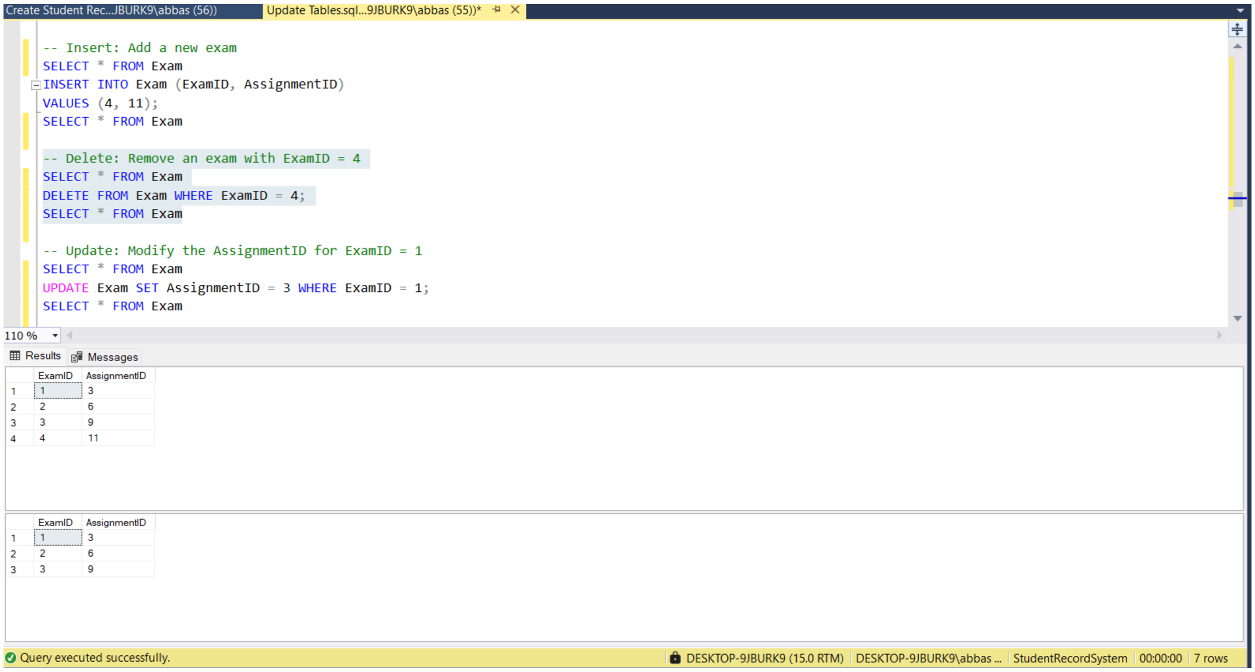

Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```
GO

------------------------------------------
-- 7. Exam Table Operations
-- Query: List all exams
SELECT * FROM Exam;

-- Insert: Add a new exam
SELECT * FROM Exam
INSERT INTO Exam (ExamID, AssignmentID)
VALUES (4, 11);
SELECT * FROM Exam

-- Delete: Remove an exam with ExamID = 4
SELECT * FROM Exam
DELETE FROM Exam WHERE ExamID = 4;
SELECT * FROM Exam
```

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9
4	4	11

Query executed successfully. | DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 7 rows



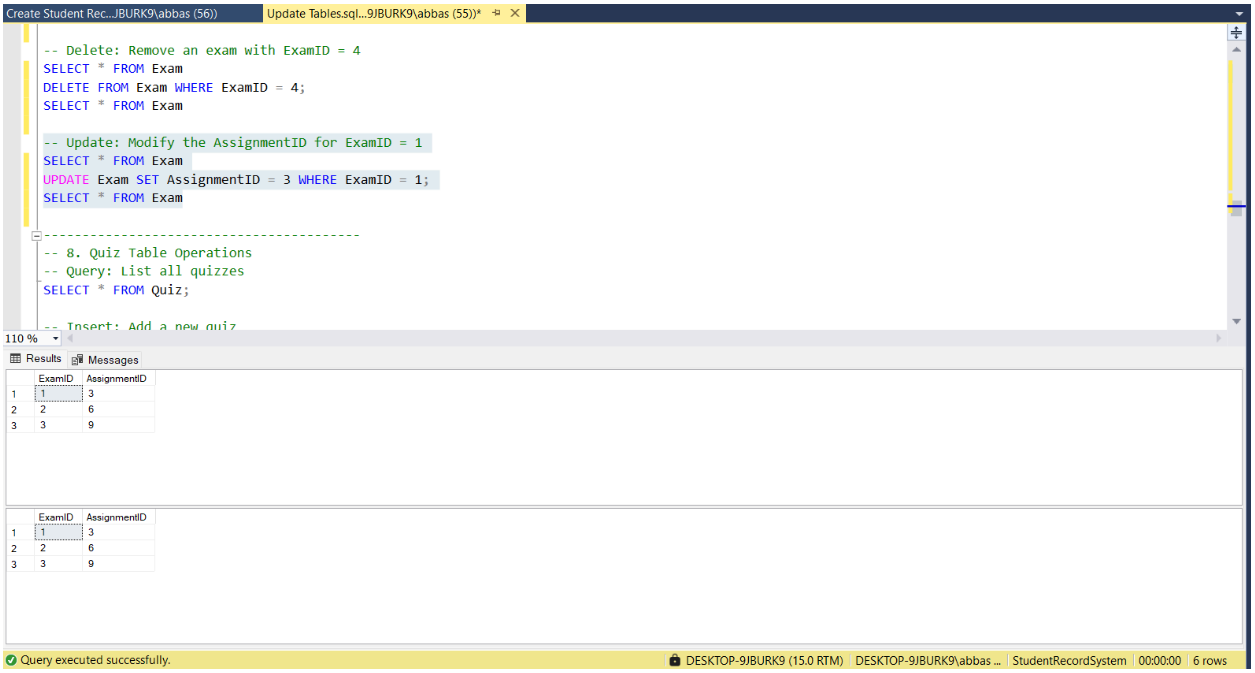
Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

```sql
-- Insert: Add a new exam
SELECT * FROM Exam
INSERT INTO Exam (ExamID, AssignmentID)
VALUES (4, 11);
SELECT * FROM Exam

-- Delete: Remove an exam with ExamID = 4
SELECT * FROM Exam
DELETE FROM Exam WHERE ExamID = 4;
SELECT * FROM Exam

-- Update: Modify the AssignmentID for ExamID = 1
SELECT * FROM Exam
UPDATE Exam SET AssignmentID = 3 WHERE ExamID = 1;
SELECT * FROM Exam
```

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9
4	4	11

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 7 rows



Create Student Rec...JBURK9\abbas (56))
Update Tables.sql...9JBURK9\abbas (55))*
110 %
Results
Messages

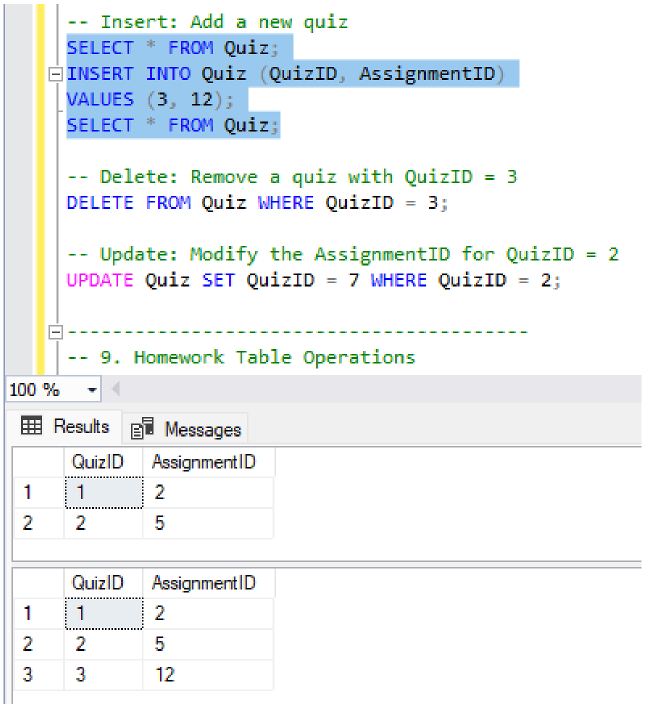
```sql
-- Delete: Remove an exam with ExamID = 4
SELECT * FROM Exam
DELETE FROM Exam WHERE ExamID = 4;
SELECT * FROM Exam

-- Update: Modify the AssignmentID for ExamID = 1
SELECT * FROM Exam
UPDATE Exam SET AssignmentID = 3 WHERE ExamID = 1;
SELECT * FROM Exam

-----------------------------------------
-- 8. Quiz Table Operations
-- Query: List all quizzes
SELECT * FROM Quiz;

-- Insert: Add a new quiz
```

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9

	ExamID	AssignmentID
1	1	3
2	2	6
3	3	9

Query executed successfully. DESKTOP-9JBURK9 (15.0 RTM) | DESKTOP-9JBURK9\abbas ... | StudentRecordSystem | 00:00:00 | 6 rows

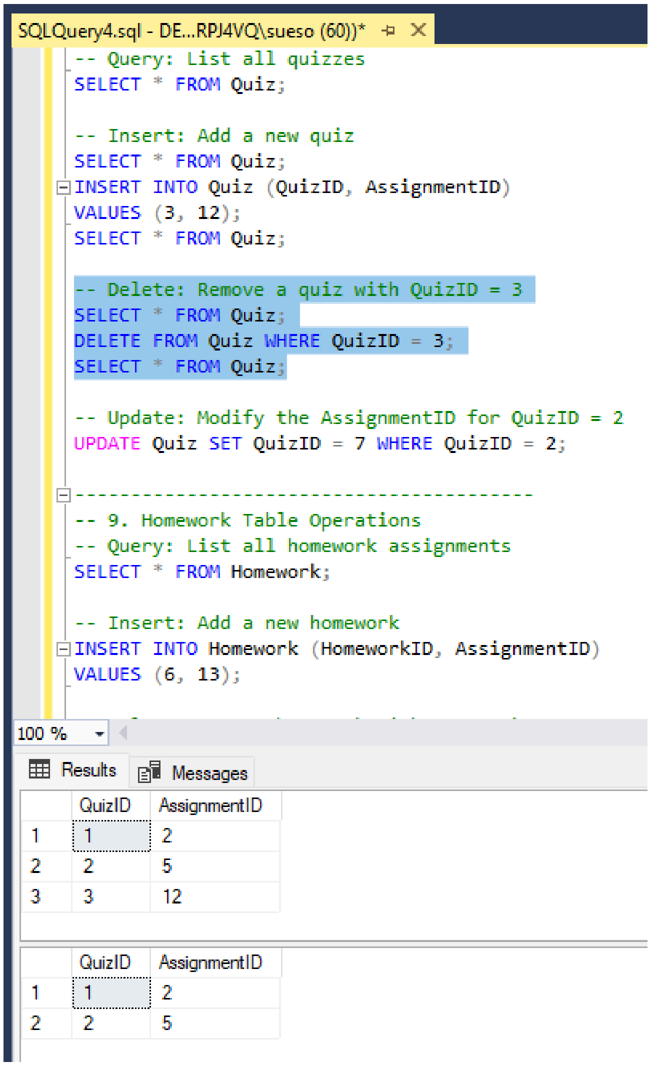
8. Quiz Table



```
-- Insert: Add a new quiz
SELECT * FROM Quiz;
INSERT INTO Quiz (QuizID, AssignmentID)
VALUES (3, 12);
SELECT * FROM Quiz;

-- Delete: Remove a quiz with QuizID = 3
DELETE FROM Quiz WHERE QuizID = 3;

-- Update: Modify the AssignmentID for QuizID = 2
UPDATE Quiz SET QuizID = 7 WHERE QuizID = 2;

-----------------------------------------
-- 9. Homework Table Operations
```

	QuizID	AssignmentID
1	1	2
2	2	5

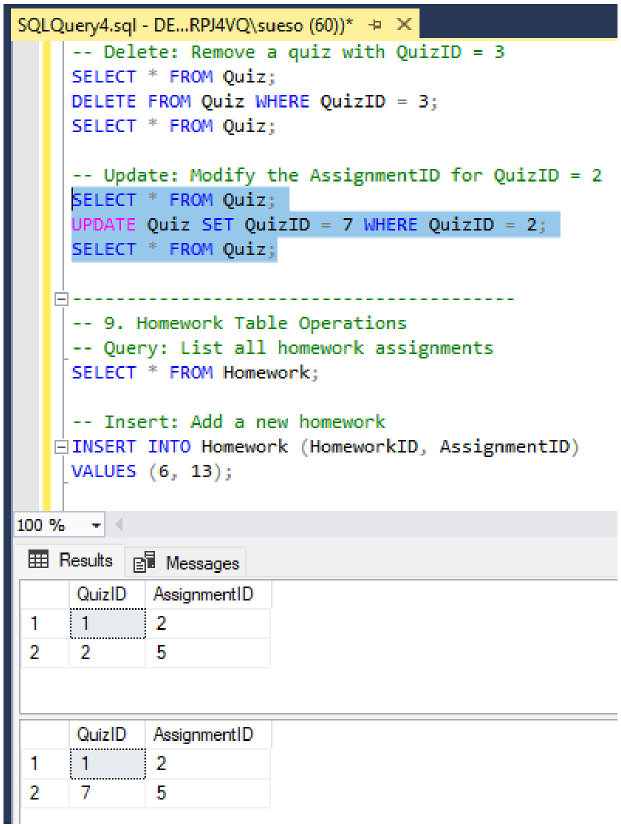
	QuizID	AssignmentID
1	1	2
2	2	5
3	3	12



```sql
-- Query: List all quizzes
SELECT * FROM Quiz;

-- Insert: Add a new quiz
SELECT * FROM Quiz;
INSERT INTO Quiz (QuizID, AssignmentID)
VALUES (3, 12);
SELECT * FROM Quiz;

-- Delete: Remove a quiz with QuizID = 3
SELECT * FROM Quiz;
DELETE FROM Quiz WHERE QuizID = 3;
SELECT * FROM Quiz;

-- Update: Modify the AssignmentID for QuizID = 2
UPDATE Quiz SET QuizID = 7 WHERE QuizID = 2;

-----------------------------------------
-- 9. Homework Table Operations
-- Query: List all homework assignments
SELECT * FROM Homework;

-- Insert: Add a new homework
INSERT INTO Homework (HomeworkID, AssignmentID)
VALUES (6, 13);
```

	QuizID	AssignmentID
1	1	2
2	2	5
3	3	12

	QuizID	AssignmentID
1	1	2
2	2	5

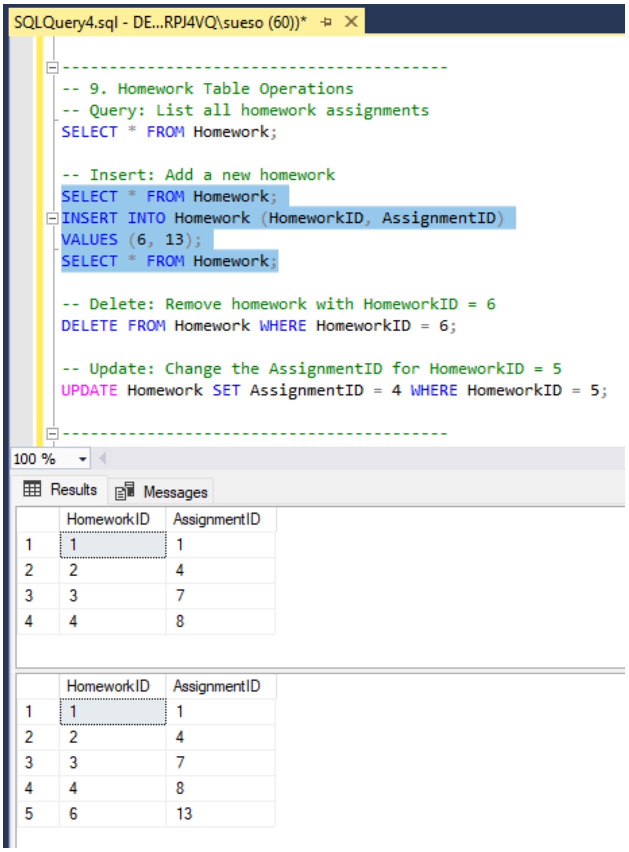


```
-- Delete: Remove a quiz with QuizID = 3
SELECT * FROM Quiz;
DELETE FROM Quiz WHERE QuizID = 3;
SELECT * FROM Quiz;

-- Update: Modify the AssignmentID for QuizID = 2
SELECT * FROM Quiz;
UPDATE Quiz SET QuizID = 7 WHERE QuizID = 2;
SELECT * FROM Quiz;

-----------------------------------------
-- 9. Homework Table Operations
-- Query: List all homework assignments
SELECT * FROM Homework;

-- Insert: Add a new homework
INSERT INTO Homework (HomeworkID, AssignmentID)
VALUES (6, 13);
```

	QuizID	AssignmentID
1	1	2
2	2	5

	QuizID	AssignmentID
1	1	2
2	7	5

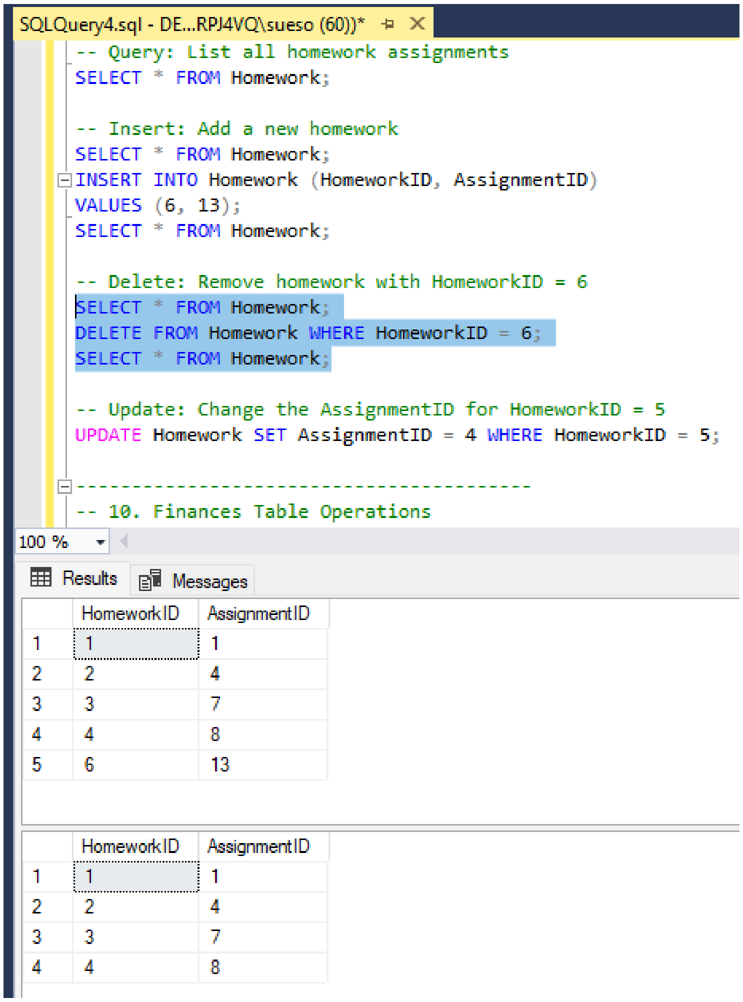
9. Homework Table



```sql
------------------------------------------
-- 9. Homework Table Operations
-- Query: List all homework assignments
SELECT * FROM Homework;

-- Insert: Add a new homework
SELECT * FROM Homework;
INSERT INTO Homework (HomeworkID, AssignmentID)
VALUES (6, 13);
SELECT * FROM Homework;

-- Delete: Remove homework with HomeworkID = 6
DELETE FROM Homework WHERE HomeworkID = 6;

-- Update: Change the AssignmentID for HomeworkID = 5
UPDATE Homework SET AssignmentID = 4 WHERE HomeworkID = 5;

------------------------------------------
```

	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	8

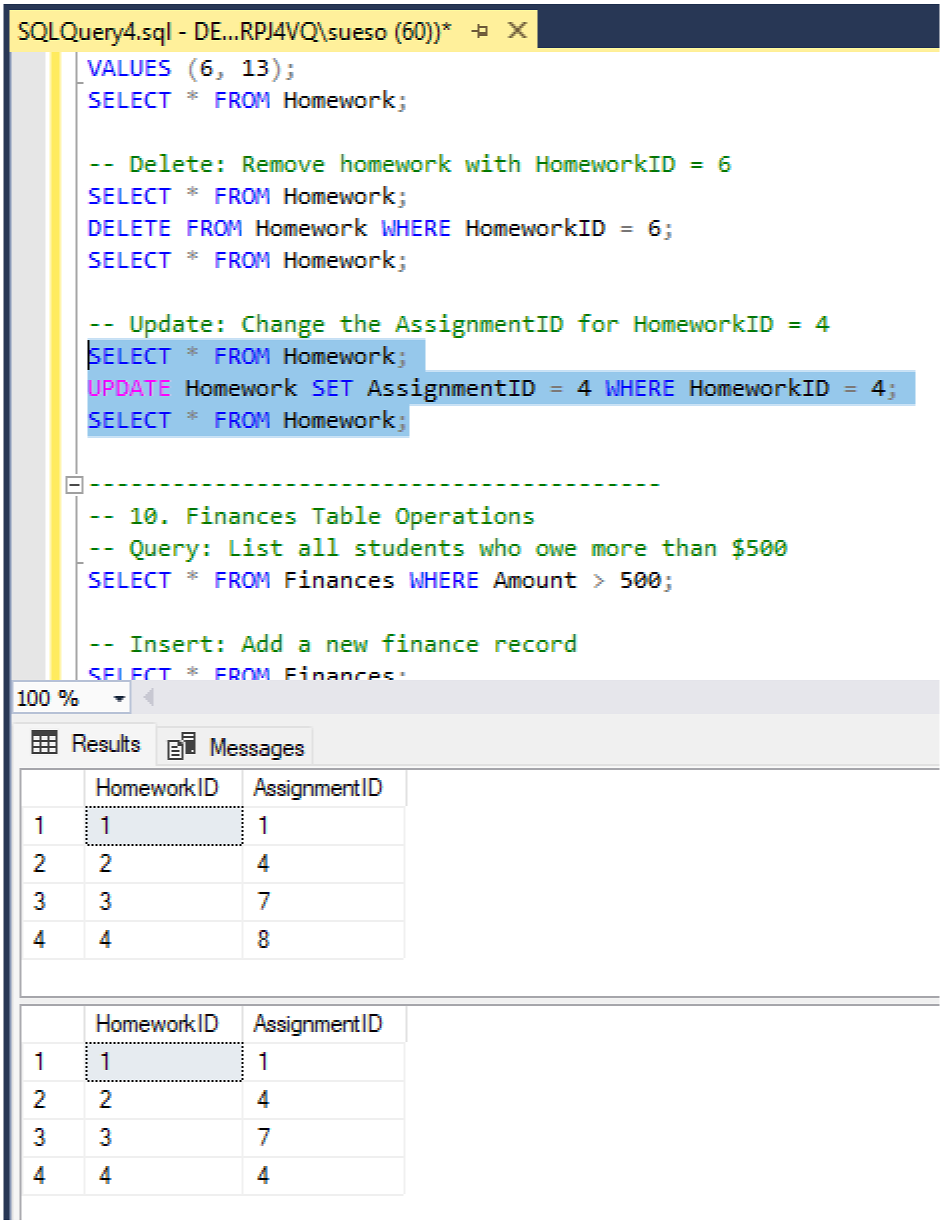
	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	8
5	6	13



```sql
-- Query: List all homework assignments
SELECT * FROM Homework;

-- Insert: Add a new homework
SELECT * FROM Homework;
INSERT INTO Homework (HomeworkID, AssignmentID)
VALUES (6, 13);
SELECT * FROM Homework;

-- Delete: Remove homework with HomeworkID = 6
SELECT * FROM Homework;
DELETE FROM Homework WHERE HomeworkID = 6;
SELECT * FROM Homework;

-- Update: Change the AssignmentID for HomeworkID = 5
UPDATE Homework SET AssignmentID = 4 WHERE HomeworkID = 5;

-------------------------------------------
-- 10. Finances Table Operations
```

	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	8
5	6	13

	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	8

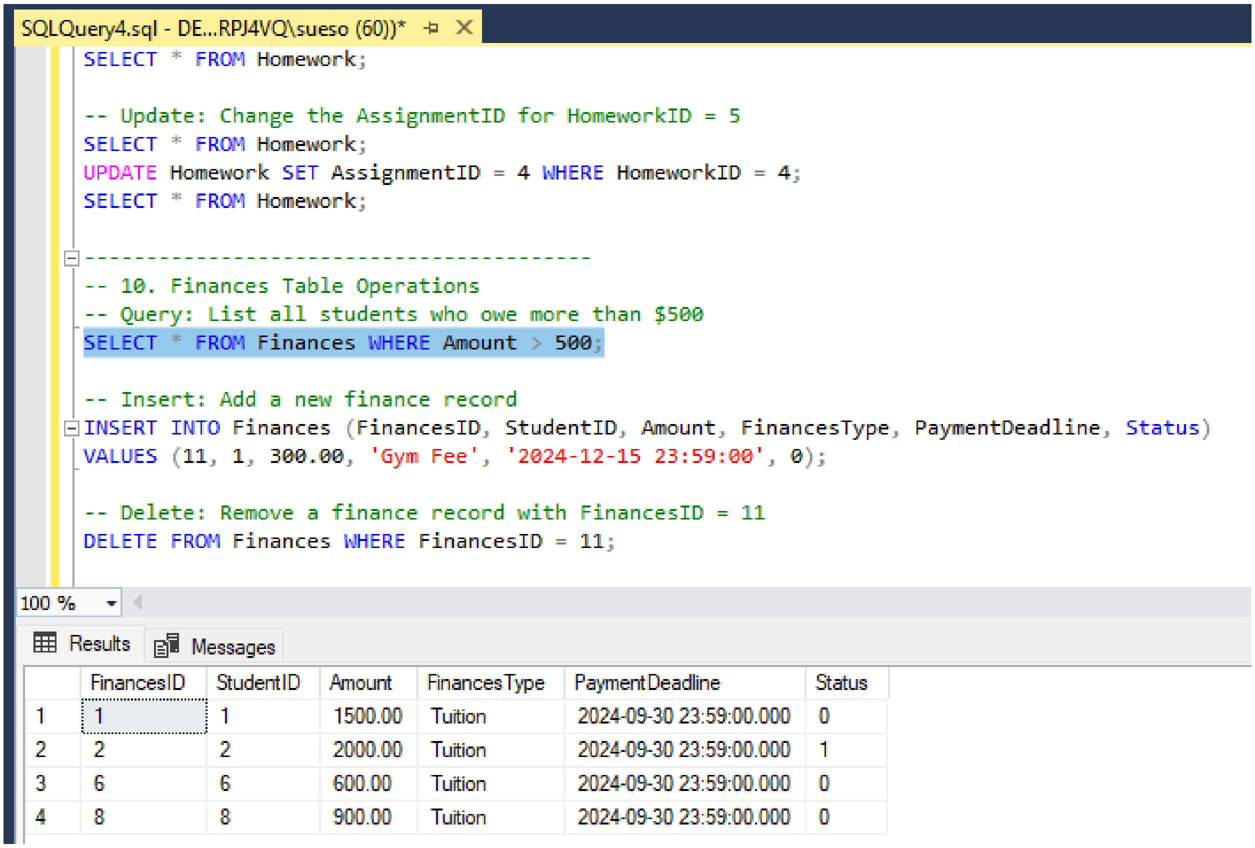


```sql
VALUES (6, 13);
SELECT * FROM Homework;

-- Delete: Remove homework with HomeworkID = 6
SELECT * FROM Homework;
DELETE FROM Homework WHERE HomeworkID = 6;
SELECT * FROM Homework;

-- Update: Change the AssignmentID for HomeworkID = 4
SELECT * FROM Homework;
UPDATE Homework SET AssignmentID = 4 WHERE HomeworkID = 4;
SELECT * FROM Homework;

------------------------------------------
-- 10. Finances Table Operations
-- Query: List all students who owe more than $500
SELECT * FROM Finances WHERE Amount > 500;

-- Insert: Add a new finance record
SELECT * FROM Finances;
```

	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	8

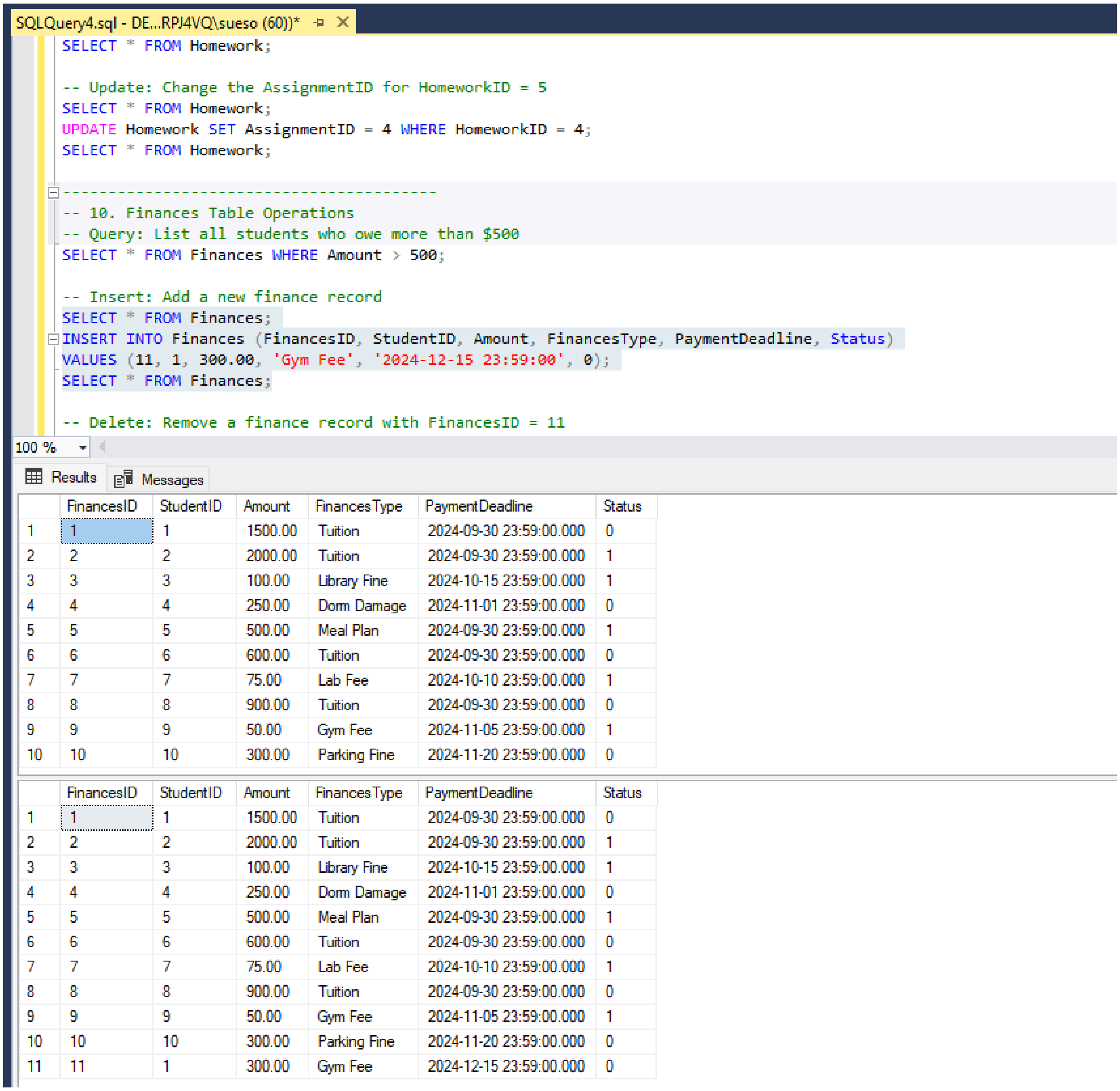
	HomeworkID	AssignmentID
1	1	1
2	2	4
3	3	7
4	4	4

10. Finances Table



```sql
SELECT * FROM Homework;

-- Update: Change the AssignmentID for HomeworkID = 5
SELECT * FROM Homework;
UPDATE Homework SET AssignmentID = 4 WHERE HomeworkID = 4;
SELECT * FROM Homework;

-----------------------------------------
-- 10. Finances Table Operations
-- Query: List all students who owe more than $500
SELECT * FROM Finances WHERE Amount > 500;

-- Insert: Add a new finance record
INSERT INTO Finances (FinancesID, StudentID, Amount, FinancesType, PaymentDeadline, Status)
VALUES (11, 1, 300.00, 'Gym Fee', '2024-12-15 23:59:00', 0);

-- Delete: Remove a finance record with FinancesID = 11
DELETE FROM Finances WHERE FinancesID = 11;
```

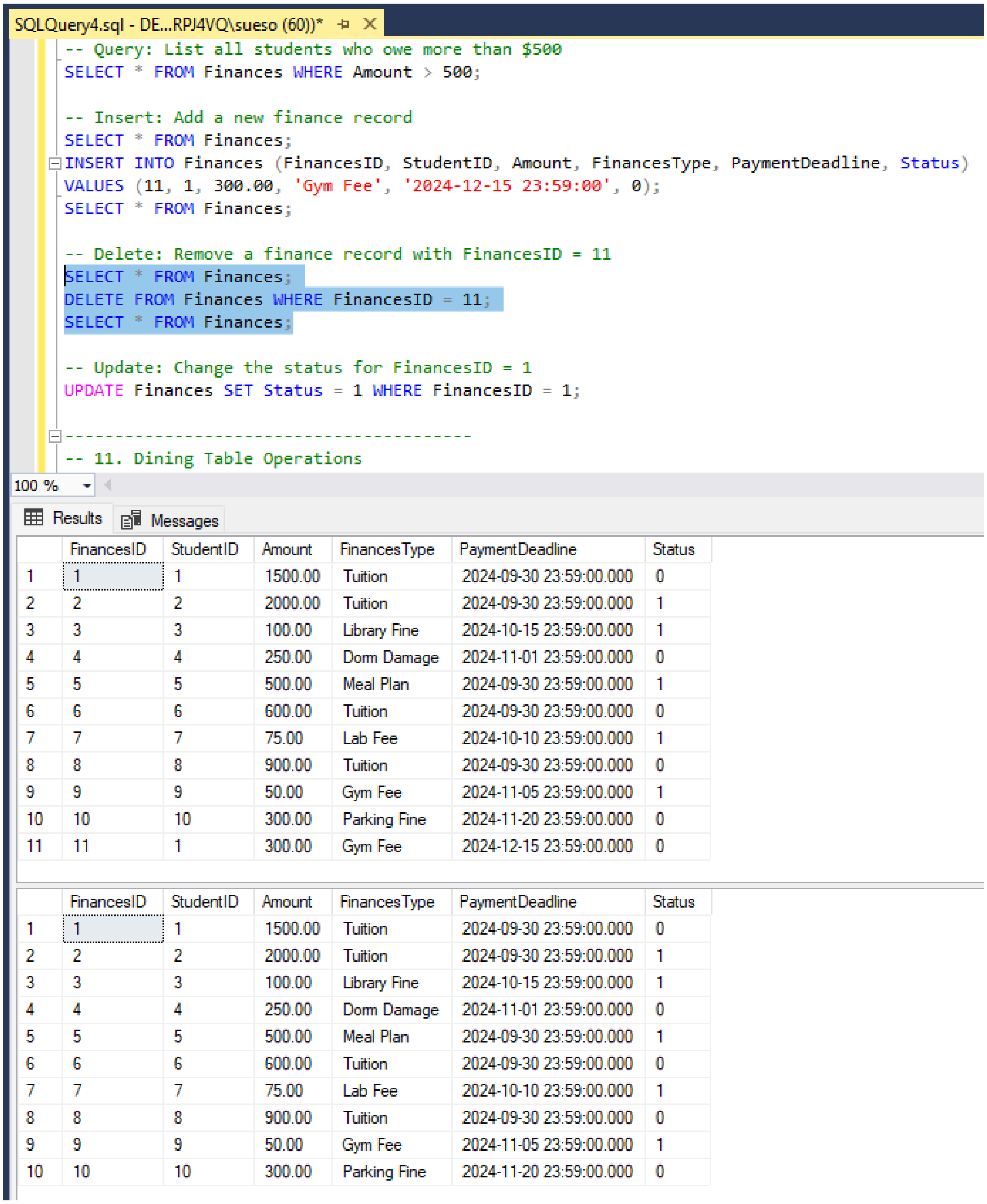
	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
4	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0



```sql
SELECT * FROM Homework;

-- Update: Change the AssignmentID for HomeworkID = 5
SELECT * FROM Homework;
UPDATE Homework SET AssignmentID = 4 WHERE HomeworkID = 4;
SELECT * FROM Homework;

-----------------------------------------
-- 10. Finances Table Operations
-- Query: List all students who owe more than $500
SELECT * FROM Finances WHERE Amount > 500;

-- Insert: Add a new finance record
SELECT * FROM Finances;
INSERT INTO Finances (FinancesID, StudentID, Amount, FinancesType, PaymentDeadline, Status)
VALUES (11, 1, 300.00, 'Gym Fee', '2024-12-15 23:59:00', 0);
SELECT * FROM Finances;

-- Delete: Remove a finance record with FinancesID = 11
```

	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0

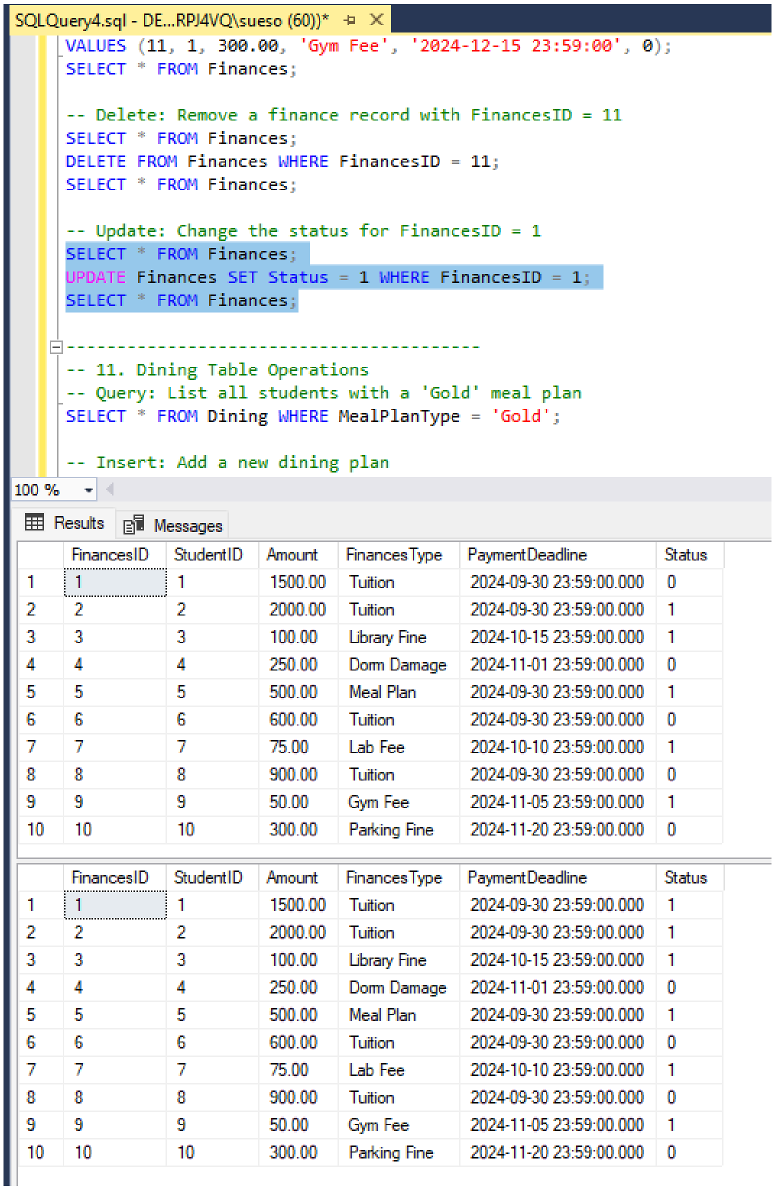
	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0
11	11	1	300.00	Gym Fee	2024-12-15 23:59:00.000	0



```sql
-- Query: List all students who owe more than $500
SELECT * FROM Finances WHERE Amount > 500;

-- Insert: Add a new finance record
SELECT * FROM Finances;
INSERT INTO Finances (FinancesID, StudentID, Amount, FinancesType, PaymentDeadline, Status)
VALUES (11, 1, 300.00, 'Gym Fee', '2024-12-15 23:59:00', 0);
SELECT * FROM Finances;

-- Delete: Remove a finance record with FinancesID = 11
SELECT * FROM Finances;
DELETE FROM Finances WHERE FinancesID = 11;
SELECT * FROM Finances;

-- Update: Change the status for FinancesID = 1
UPDATE Finances SET Status = 1 WHERE FinancesID = 1;

-----------------------------------------
-- 11. Dining Table Operations
```

	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0
11	11	1	300.00	Gym Fee	2024-12-15 23:59:00.000	0

	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0

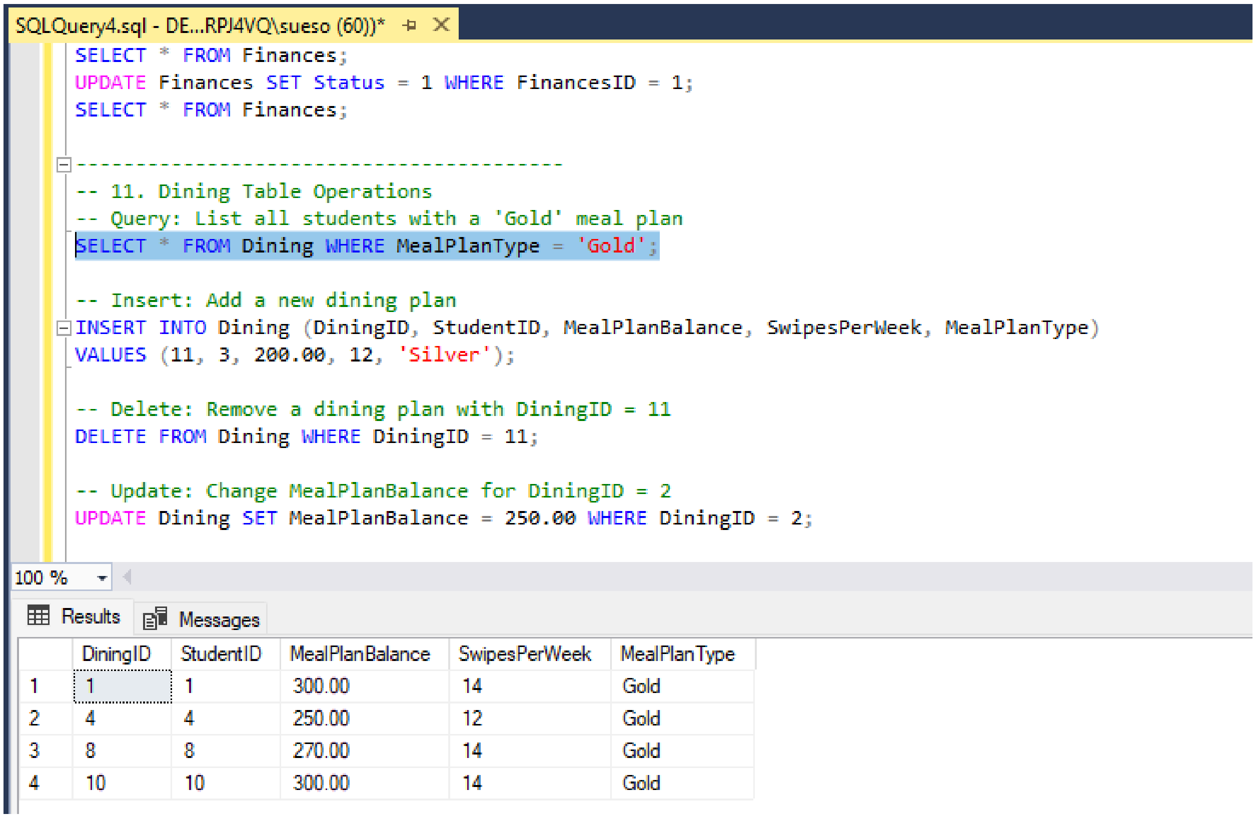


```sql
VALUES (11, 1, 300.00, 'Gym Fee', '2024-12-15 23:59:00', 0);
SELECT * FROM Finances;

-- Delete: Remove a finance record with FinancesID = 11
SELECT * FROM Finances;
DELETE FROM Finances WHERE FinancesID = 11;
SELECT * FROM Finances;

-- Update: Change the status for FinancesID = 1
SELECT * FROM Finances;
UPDATE Finances SET Status = 1 WHERE FinancesID = 1;
SELECT * FROM Finances;

------------------------------------------
-- 11. Dining Table Operations
-- Query: List all students with a 'Gold' meal plan
SELECT * FROM Dining WHERE MealPlanType = 'Gold';

-- Insert: Add a new dining plan
```

	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	0
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0

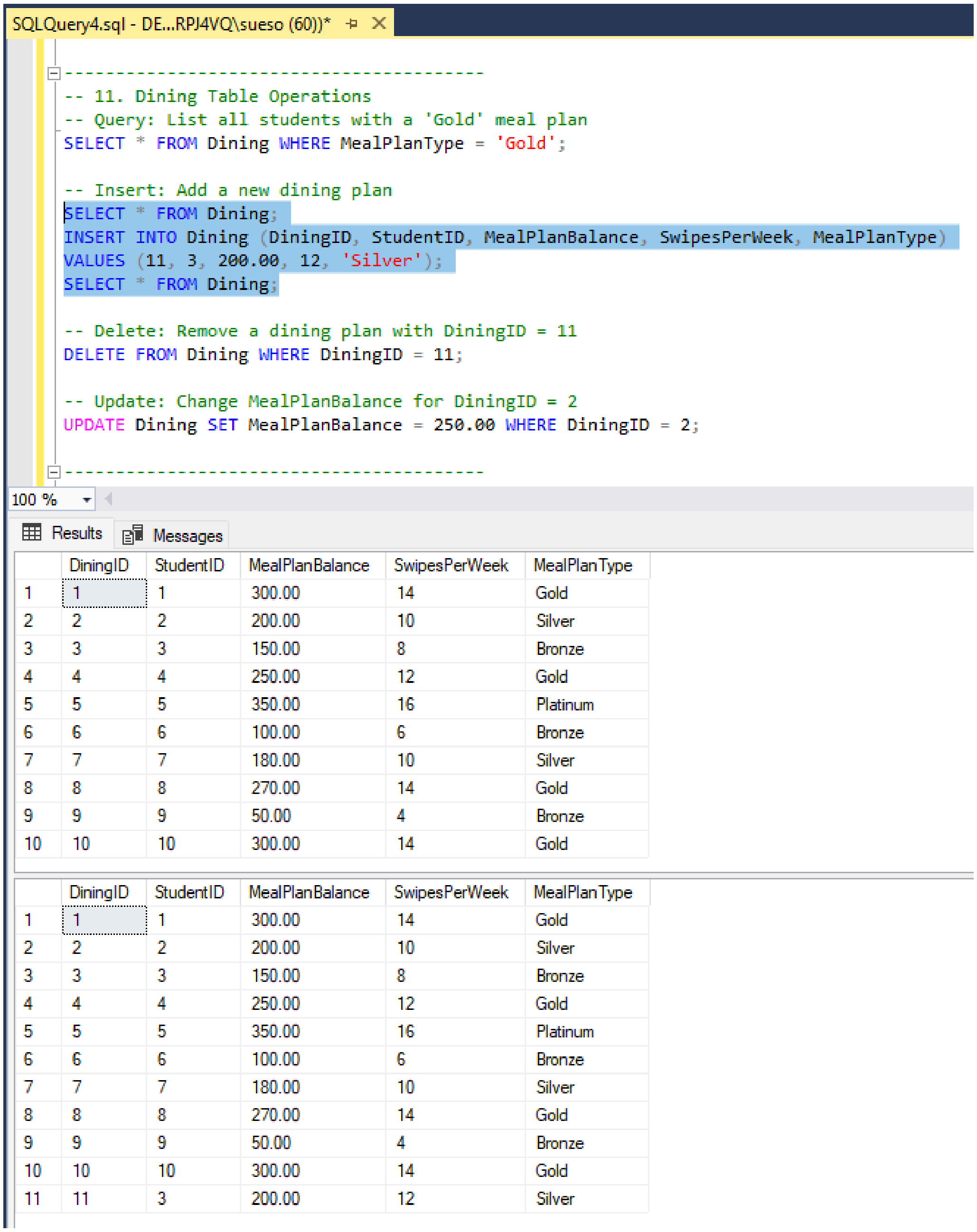
	FinancesID	StudentID	Amount	FinancesType	PaymentDeadline	Status
1	1	1	1500.00	Tuition	2024-09-30 23:59:00.000	1
2	2	2	2000.00	Tuition	2024-09-30 23:59:00.000	1
3	3	3	100.00	Library Fine	2024-10-15 23:59:00.000	1
4	4	4	250.00	Dorm Damage	2024-11-01 23:59:00.000	0
5	5	5	500.00	Meal Plan	2024-09-30 23:59:00.000	1
6	6	6	600.00	Tuition	2024-09-30 23:59:00.000	0
7	7	7	75.00	Lab Fee	2024-10-10 23:59:00.000	1
8	8	8	900.00	Tuition	2024-09-30 23:59:00.000	0
9	9	9	50.00	Gym Fee	2024-11-05 23:59:00.000	1
10	10	10	300.00	Parking Fine	2024-11-20 23:59:00.000	0

11. Dining Table



```sql
SELECT * FROM Finances;
UPDATE Finances SET Status = 1 WHERE FinancesID = 1;
SELECT * FROM Finances;

------------------------------------------
-- 11. Dining Table Operations
-- Query: List all students with a 'Gold' meal plan
SELECT * FROM Dining WHERE MealPlanType = 'Gold';

-- Insert: Add a new dining plan
INSERT INTO Dining (DiningID, StudentID, MealPlanBalance, SwipesPerWeek, MealPlanType)
VALUES (11, 3, 200.00, 12, 'Silver');

-- Delete: Remove a dining plan with DiningID = 11
DELETE FROM Dining WHERE DiningID = 11;

-- Update: Change MealPlanBalance for DiningID = 2
UPDATE Dining SET MealPlanBalance = 250.00 WHERE DiningID = 2;
```

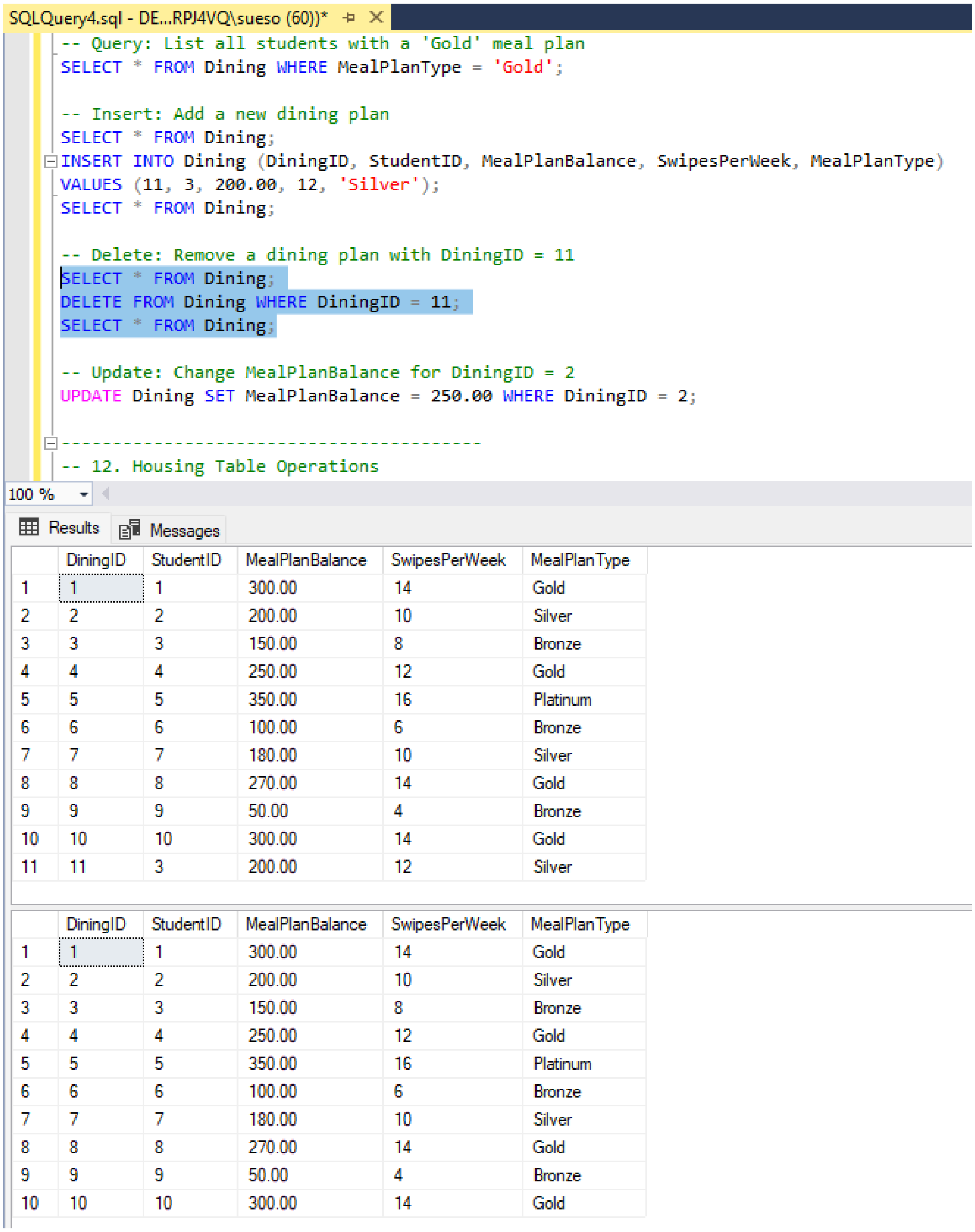
	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	4	4	250.00	12	Gold
3	8	8	270.00	14	Gold
4	10	10	300.00	14	Gold



```sql
------------------------------------------
-- 11. Dining Table Operations
-- Query: List all students with a 'Gold' meal plan
SELECT * FROM Dining WHERE MealPlanType = 'Gold';

-- Insert: Add a new dining plan
SELECT * FROM Dining;
INSERT INTO Dining (DiningID, StudentID, MealPlanBalance, SwipesPerWeek, MealPlanType)
VALUES (11, 3, 200.00, 12, 'Silver');
SELECT * FROM Dining;

-- Delete: Remove a dining plan with DiningID = 11
DELETE FROM Dining WHERE DiningID = 11;

-- Update: Change MealPlanBalance for DiningID = 2
UPDATE Dining SET MealPlanBalance = 250.00 WHERE DiningID = 2;

------------------------------------------
```

	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold

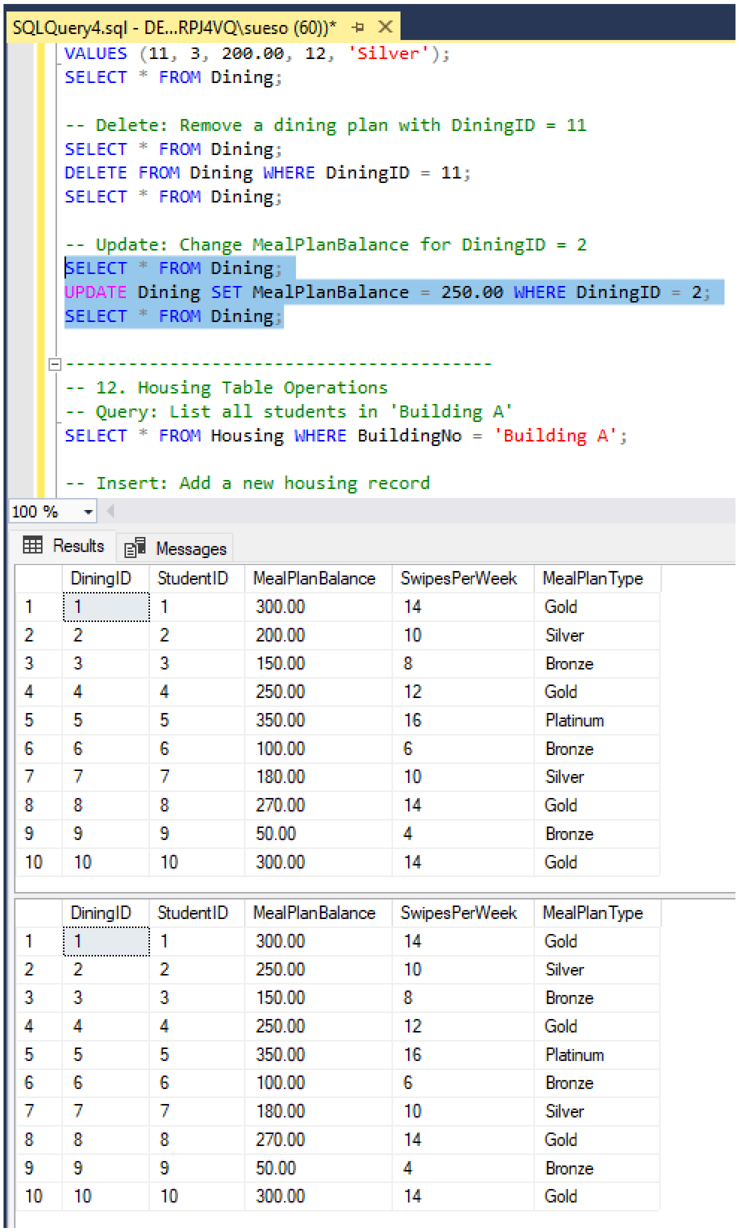
	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold
11	11	3	200.00	12	Silver



```sql
-- Query: List all students with a 'Gold' meal plan
SELECT * FROM Dining WHERE MealPlanType = 'Gold';

-- Insert: Add a new dining plan
SELECT * FROM Dining;
INSERT INTO Dining (DiningID, StudentID, MealPlanBalance, SwipesPerWeek, MealPlanType)
VALUES (11, 3, 200.00, 12, 'Silver');
SELECT * FROM Dining;

-- Delete: Remove a dining plan with DiningID = 11
SELECT * FROM Dining;
DELETE FROM Dining WHERE DiningID = 11;
SELECT * FROM Dining;

-- Update: Change MealPlanBalance for DiningID = 2
UPDATE Dining SET MealPlanBalance = 250.00 WHERE DiningID = 2;

-----------------------------------------
-- 12. Housing Table Operations
```

	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold
11	11	3	200.00	12	Silver

	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold

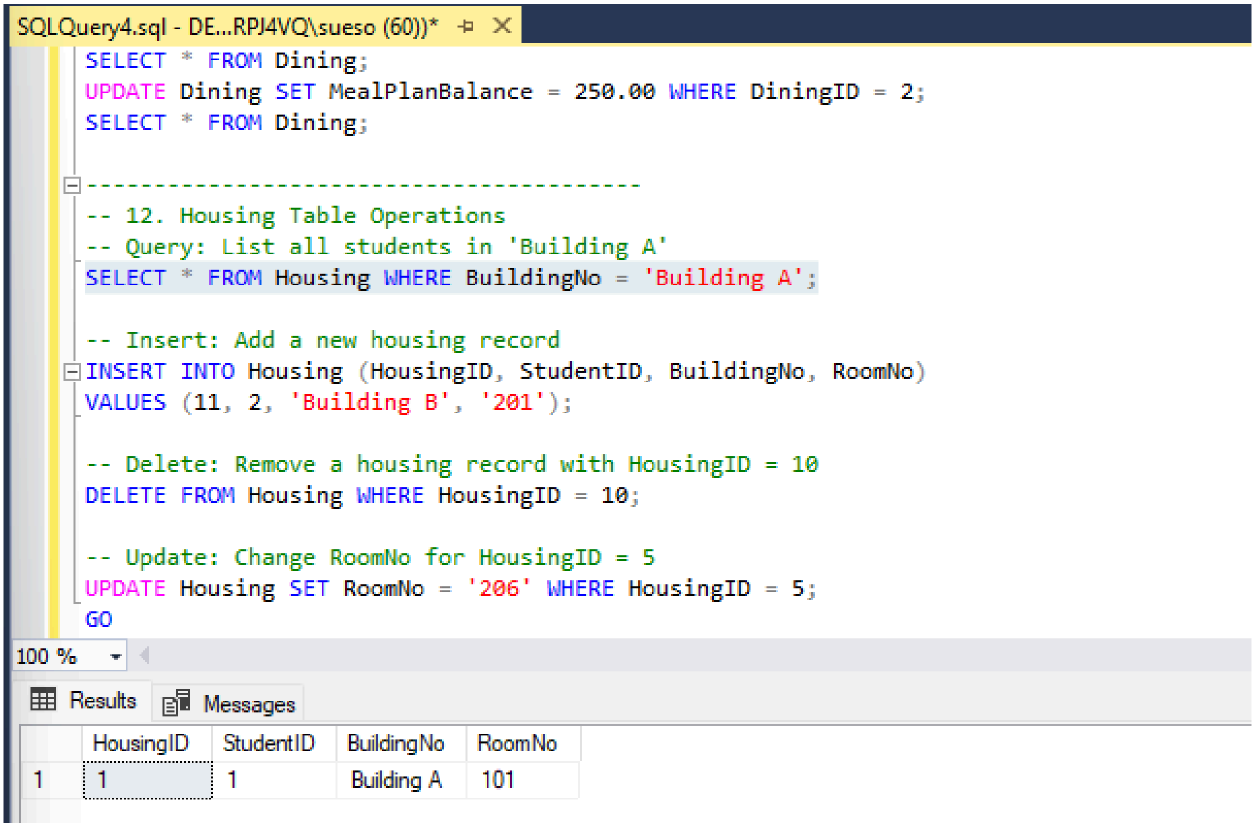


```sql
VALUES (11, 3, 200.00, 12, 'Silver');
SELECT * FROM Dining;

-- Delete: Remove a dining plan with DiningID = 11
SELECT * FROM Dining;
DELETE FROM Dining WHERE DiningID = 11;
SELECT * FROM Dining;

-- Update: Change MealPlanBalance for DiningID = 2
SELECT * FROM Dining;
UPDATE Dining SET MealPlanBalance = 250.00 WHERE DiningID = 2;
SELECT * FROM Dining;

------------------------------------------
-- 12. Housing Table Operations
-- Query: List all students in 'Building A'
SELECT * FROM Housing WHERE BuildingNo = 'Building A';

-- Insert: Add a new housing record
```

	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	200.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold

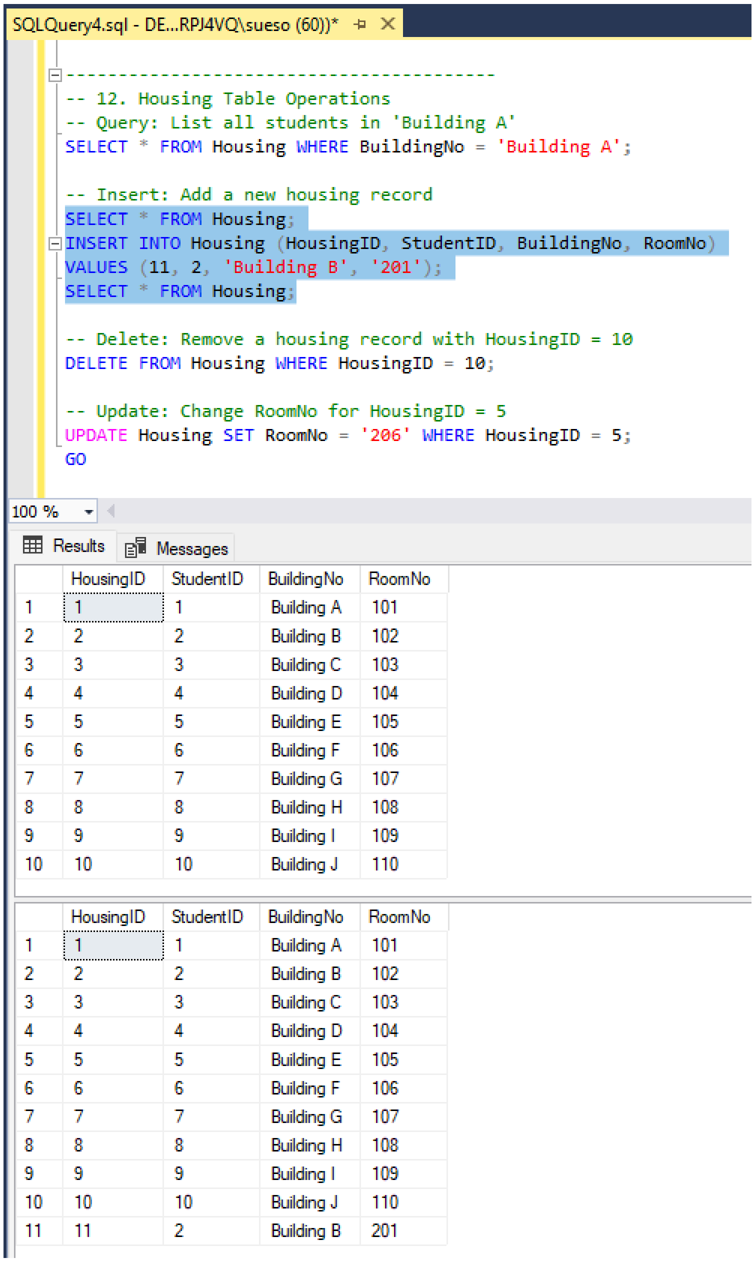
	DiningID	StudentID	MealPlanBalance	SwipesPerWeek	MealPlanType
1	1	1	300.00	14	Gold
2	2	2	250.00	10	Silver
3	3	3	150.00	8	Bronze
4	4	4	250.00	12	Gold
5	5	5	350.00	16	Platinum
6	6	6	100.00	6	Bronze
7	7	7	180.00	10	Silver
8	8	8	270.00	14	Gold
9	9	9	50.00	4	Bronze
10	10	10	300.00	14	Gold

12. Housing Table



```sql
SELECT * FROM Dining;
UPDATE Dining SET MealPlanBalance = 250.00 WHERE DiningID = 2;
SELECT * FROM Dining;

-----------------------------------------
-- 12. Housing Table Operations
-- Query: List all students in 'Building A'
SELECT * FROM Housing WHERE BuildingNo = 'Building A';

-- Insert: Add a new housing record
INSERT INTO Housing (HousingID, StudentID, BuildingNo, RoomNo)
VALUES (11, 2, 'Building B', '201');

-- Delete: Remove a housing record with HousingID = 10
DELETE FROM Housing WHERE HousingID = 10;

-- Update: Change RoomNo for HousingID = 5
UPDATE Housing SET RoomNo = '206' WHERE HousingID = 5;
GO
```

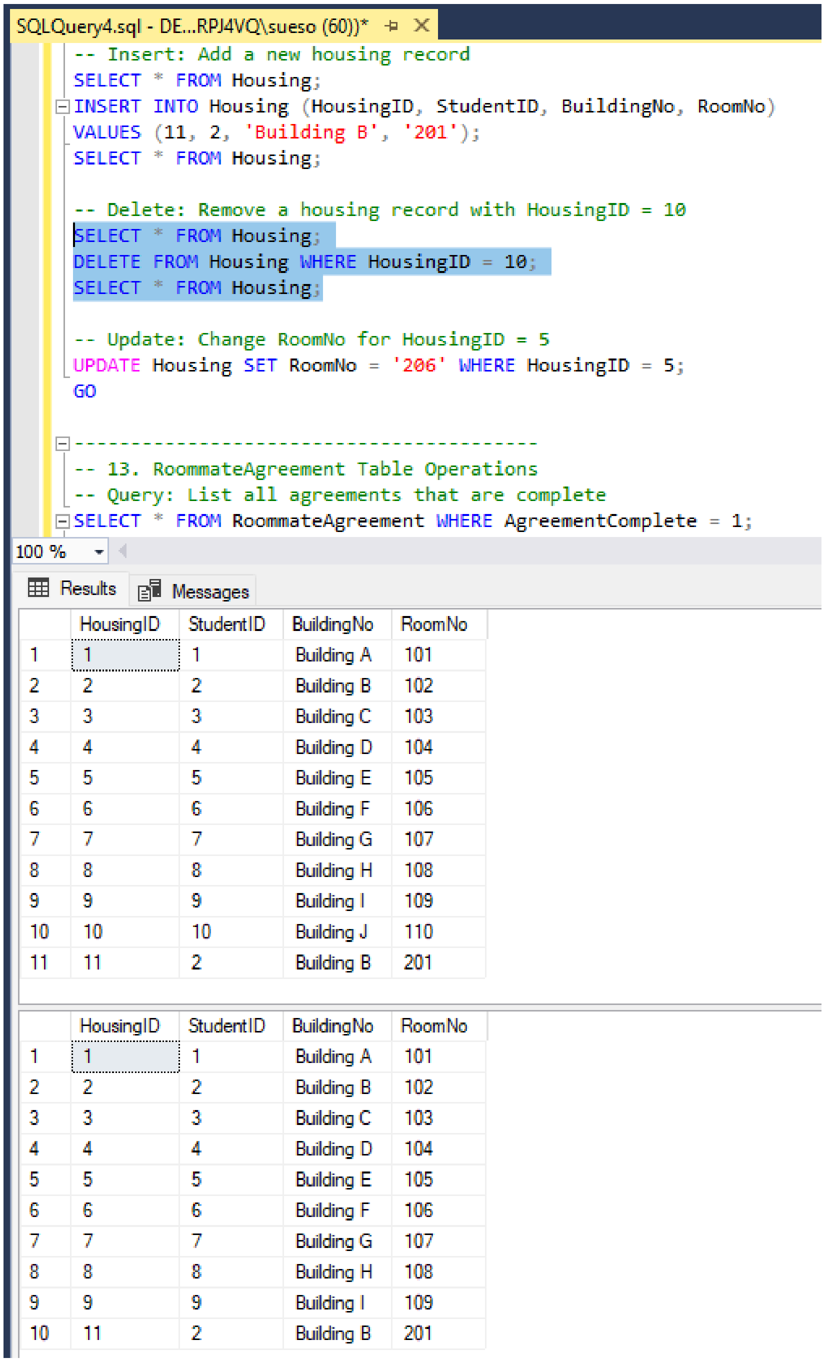
	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101



```sql
------------------------------------------
-- 12. Housing Table Operations
-- Query: List all students in 'Building A'
SELECT * FROM Housing WHERE BuildingNo = 'Building A';

-- Insert: Add a new housing record
SELECT * FROM Housing;
INSERT INTO Housing (HousingID, StudentID, BuildingNo, RoomNo)
VALUES (11, 2, 'Building B', '201');
SELECT * FROM Housing;

-- Delete: Remove a housing record with HousingID = 10
DELETE FROM Housing WHERE HousingID = 10;

-- Update: Change RoomNo for HousingID = 5
UPDATE Housing SET RoomNo = '206' WHERE HousingID = 5;
GO
```

	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	10	10	Building J	110

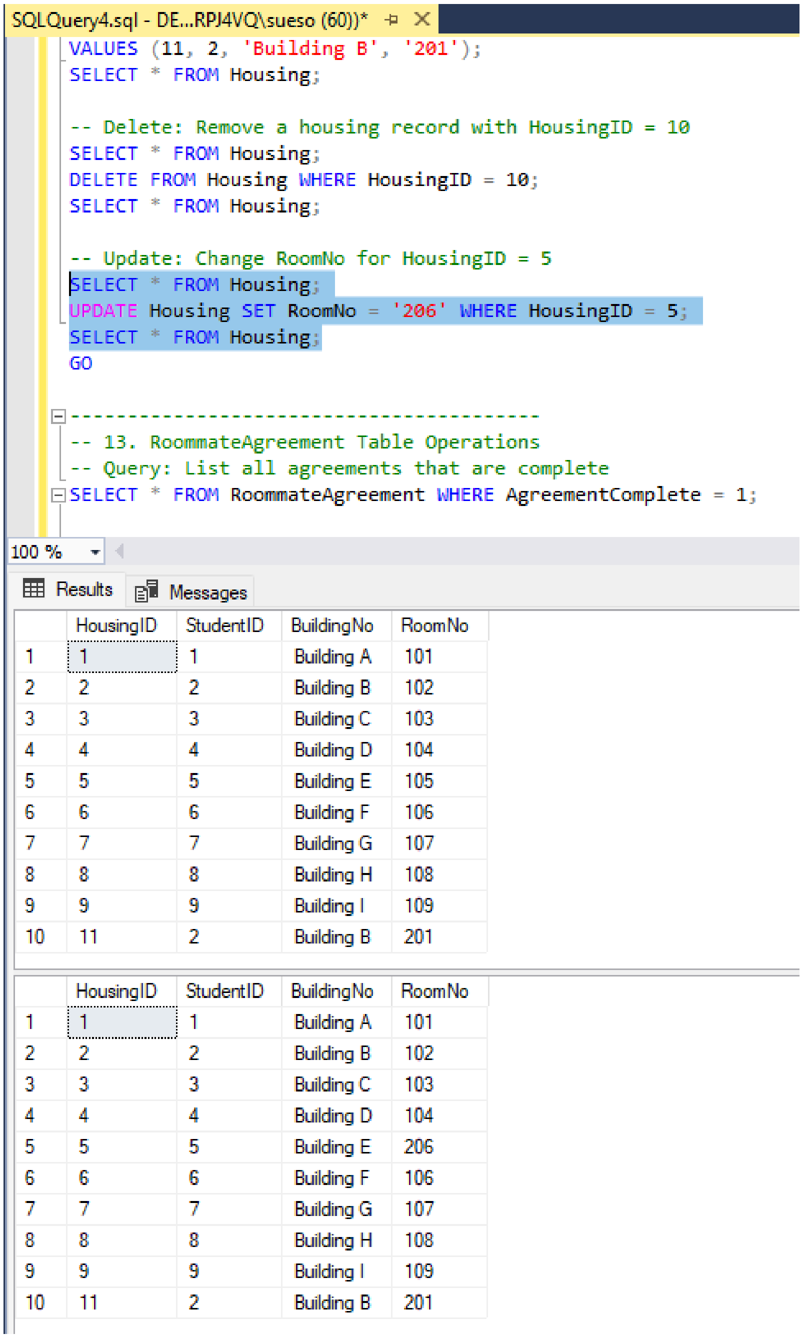
	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	10	10	Building J	110
11	11	2	Building B	201



```sql
-- Insert: Add a new housing record
SELECT * FROM Housing;
INSERT INTO Housing (HousingID, StudentID, BuildingNo, RoomNo)
VALUES (11, 2, 'Building B', '201');
SELECT * FROM Housing;

-- Delete: Remove a housing record with HousingID = 10
SELECT * FROM Housing;
DELETE FROM Housing WHERE HousingID = 10;
SELECT * FROM Housing;

-- Update: Change RoomNo for HousingID = 5
UPDATE Housing SET RoomNo = '206' WHERE HousingID = 5;
GO

-----------------------------------------
-- 13. RoommateAgreement Table Operations
-- Query: List all agreements that are complete
SELECT * FROM RoommateAgreement WHERE AgreementComplete = 1;
```

	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	10	10	Building J	110
11	11	2	Building B	201

	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	11	2	Building B	201

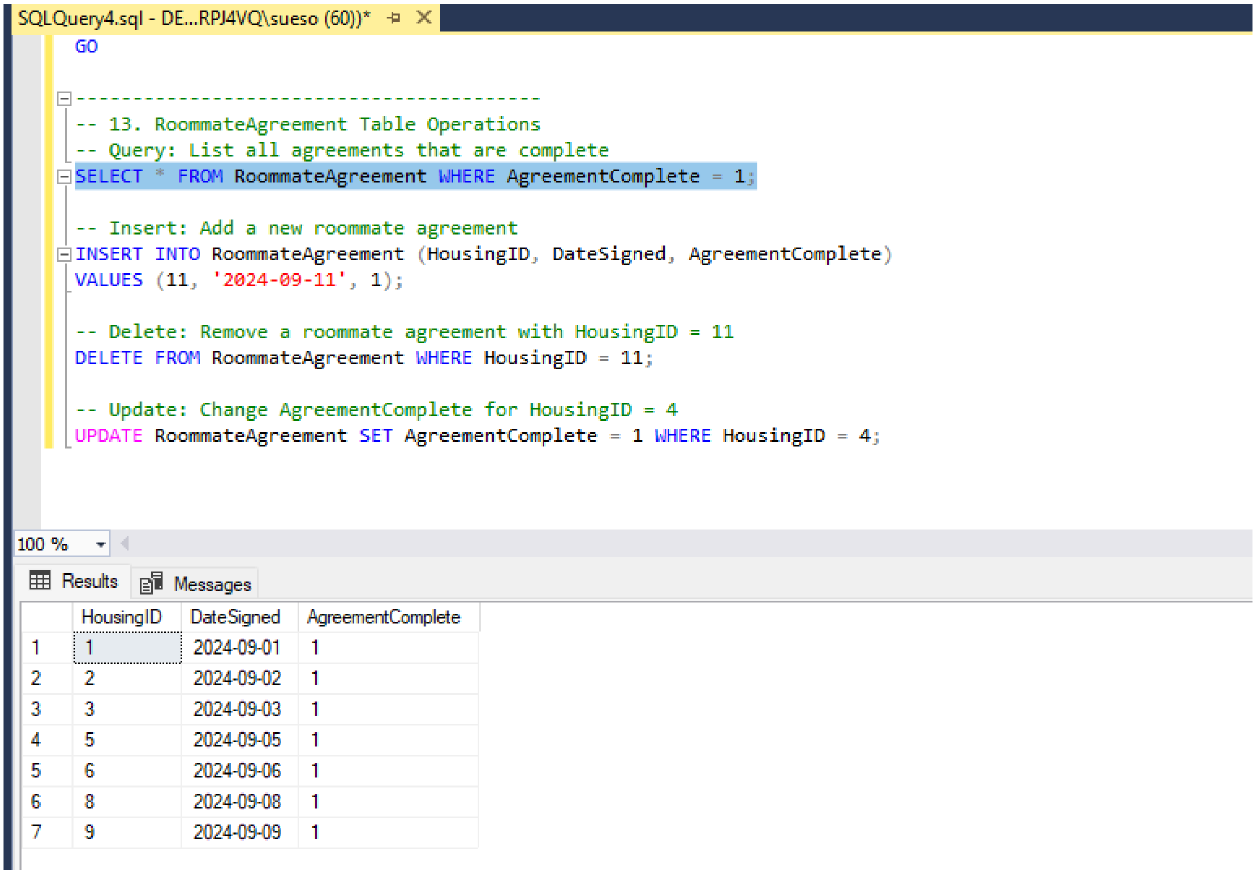


```sql
VALUES (11, 2, 'Building B', '201');
SELECT * FROM Housing;

-- Delete: Remove a housing record with HousingID = 10
SELECT * FROM Housing;
DELETE FROM Housing WHERE HousingID = 10;
SELECT * FROM Housing;

-- Update: Change RoomNo for HousingID = 5
SELECT * FROM Housing;
UPDATE Housing SET RoomNo = '206' WHERE HousingID = 5;
SELECT * FROM Housing;
GO

------------------------------------------
-- 13. RoommateAgreement Table Operations
-- Query: List all agreements that are complete
SELECT * FROM RoommateAgreement WHERE AgreementComplete = 1;
```

	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	105
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	11	2	Building B	201

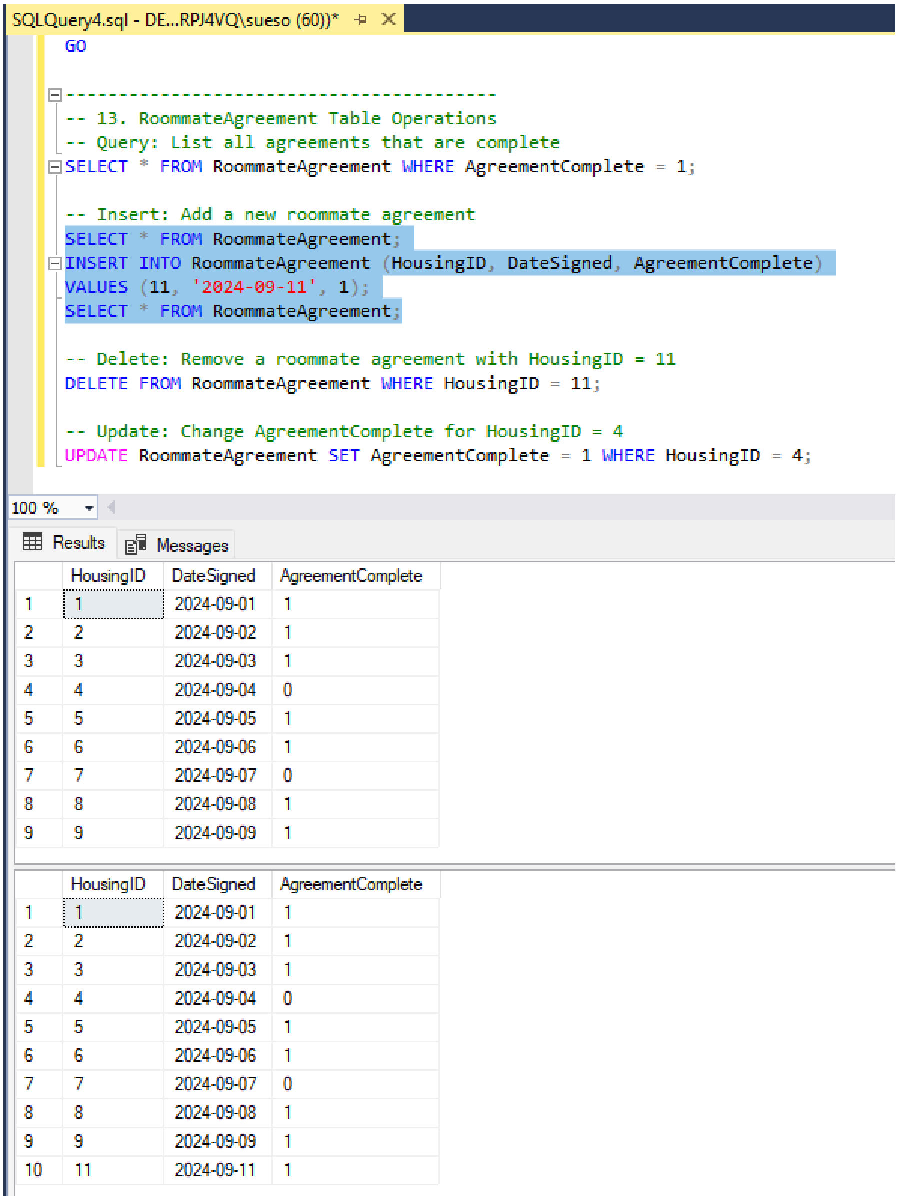
	HousingID	StudentID	BuildingNo	RoomNo
1	1	1	Building A	101
2	2	2	Building B	102
3	3	3	Building C	103
4	4	4	Building D	104
5	5	5	Building E	206
6	6	6	Building F	106
7	7	7	Building G	107
8	8	8	Building H	108
9	9	9	Building I	109
10	11	2	Building B	201

13. RoommateAgreement Table



```
GO

-----------------------------------------
-- 13. RoommateAgreement Table Operations
-- Query: List all agreements that are complete
SELECT * FROM RoommateAgreement WHERE AgreementComplete = 1;

-- Insert: Add a new roommate agreement
INSERT INTO RoommateAgreement (HousingID, DateSigned, AgreementComplete)
VALUES (11, '2024-09-11', 1);

-- Delete: Remove a roommate agreement with HousingID = 11
DELETE FROM RoommateAgreement WHERE HousingID = 11;

-- Update: Change AgreementComplete for HousingID = 4
UPDATE RoommateAgreement SET AgreementComplete = 1 WHERE HousingID = 4;
```

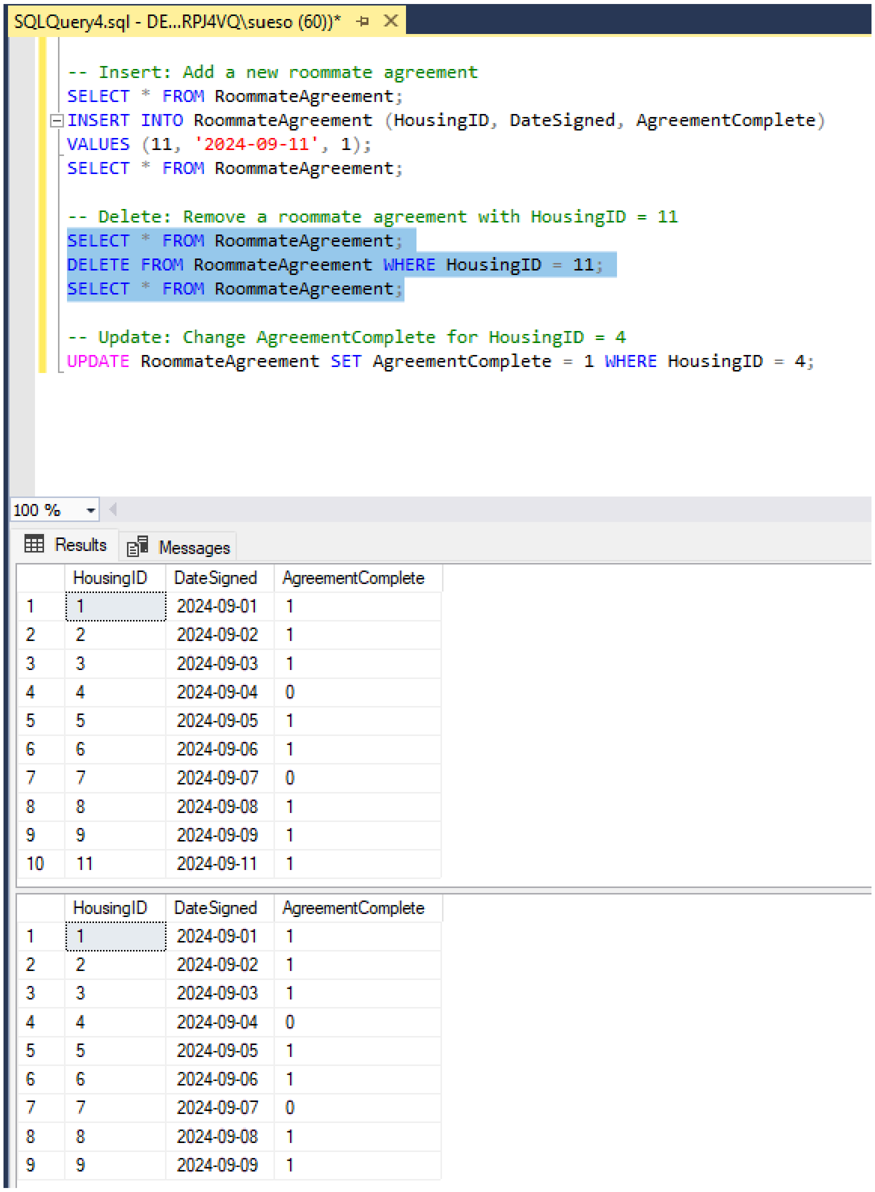
	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	5	2024-09-05	1
5	6	2024-09-06	1
6	8	2024-09-08	1
7	9	2024-09-09	1



```sql
GO

------------------------------------------
-- 13. RoommateAgreement Table Operations
-- Query: List all agreements that are complete
SELECT * FROM RoommateAgreement WHERE AgreementComplete = 1;

-- Insert: Add a new roommate agreement
SELECT * FROM RoommateAgreement;
INSERT INTO RoommateAgreement (HousingID, DateSigned, AgreementComplete)
VALUES (11, '2024-09-11', 1);
SELECT * FROM RoommateAgreement;

-- Delete: Remove a roommate agreement with HousingID = 11
DELETE FROM RoommateAgreement WHERE HousingID = 11;

-- Update: Change AgreementComplete for HousingID = 4
UPDATE RoommateAgreement SET AgreementComplete = 1 WHERE HousingID = 4;
```

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1

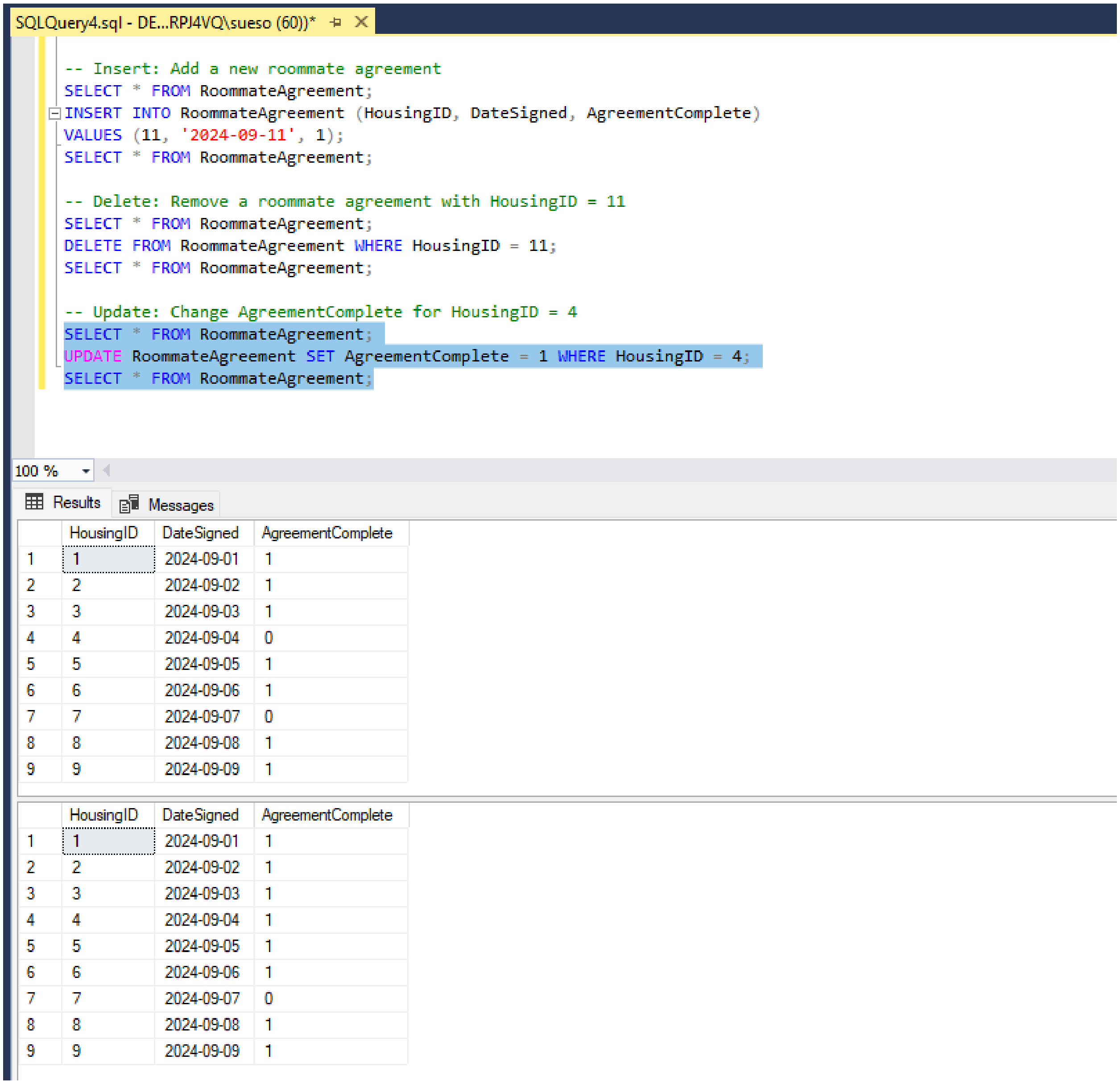
	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1
10	11	2024-09-11	1



```sql
-- Insert: Add a new roommate agreement
SELECT * FROM RoommateAgreement;
INSERT INTO RoommateAgreement (HousingID, DateSigned, AgreementComplete)
VALUES (11, '2024-09-11', 1);
SELECT * FROM RoommateAgreement;

-- Delete: Remove a roommate agreement with HousingID = 11
SELECT * FROM RoommateAgreement;
DELETE FROM RoommateAgreement WHERE HousingID = 11;
SELECT * FROM RoommateAgreement;

-- Update: Change AgreementComplete for HousingID = 4
UPDATE RoommateAgreement SET AgreementComplete = 1 WHERE HousingID = 4;
```

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1
10	11	2024-09-11	1

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1



```sql
-- Insert: Add a new roommate agreement
SELECT * FROM RoommateAgreement;
INSERT INTO RoommateAgreement (HousingID, DateSigned, AgreementComplete)
VALUES (11, '2024-09-11', 1);
SELECT * FROM RoommateAgreement;

-- Delete: Remove a roommate agreement with HousingID = 11
SELECT * FROM RoommateAgreement;
DELETE FROM RoommateAgreement WHERE HousingID = 11;
SELECT * FROM RoommateAgreement;

-- Update: Change AgreementComplete for HousingID = 4
SELECT * FROM RoommateAgreement;
UPDATE RoommateAgreement SET AgreementComplete = 1 WHERE HousingID = 4;
SELECT * FROM RoommateAgreement;
```

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	0
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1

	HousingID	DateSigned	AgreementComplete
1	1	2024-09-01	1
2	2	2024-09-02	1
3	3	2024-09-03	1
4	4	2024-09-04	1
5	5	2024-09-05	1
6	6	2024-09-06	1
7	7	2024-09-07	0
8	8	2024-09-08	1
9	9	2024-09-09	1

4. References:

1. A. Eludire, “The Design and Implementation of Student Academic Record Management System,” International Journal of Computing and ICT Research, vol. 5, no. 2, pp. 20-25, 2011.

2. A. S. Bidyarthy, A. Kumar, “Student Database Management System,” International Journal of Emerging Technologies, vol. 3, no. 2, pp. 28-35, 2012.

3. A. Tamboli, “Institute Administration Automation and Student Database Management System,” Journal of Automation and Control Engineering, vol. 5, no. 4, pp. 210-216, 2017.

4. R. Elmasri, S. Navathe, "Relational Database Design by ER- and EER-to-Relational Mapping" in Fundamentals of Database Systems, 7th ed., USA: Pearson, 2021, ch. 9, pp. 289 - 306.