MTi
Modern University
for technology & information

Modern University
Faculty of
Computers
& Artificial Intelligence

Database Systems 1

Project Name:

School Database system

Submitted By

Esraa Mahmoud

Table of Contents:

1. Introduction
2. Functional View
3. Entity Relationship Diagram (ERD) and Mapping to tables
4. Database Relationships and Tables with Sample Data
5. Input / Output Interface Design
   a. Main Menu
   b. Forms
   c. Transactions
   d. Queries
   e. Reports

INTRODUCTION

Consider the following set of requirements for a School database System:

The MTI School Database is designed to efficiently manage and organize information related to students, courses, teachers, enrollments, assignments, exams, attendance, and payments within the school.

Students Table:

- **Attributes:**
  - `StudentID`: Unique identifier for each student.
  - `FullName`: Full name of the student.
  - `Level`: Academic level or grade of the student.
  - `Address`: Address of the student.
  - `Birthdate`: Date of birth of the student.

Courses Table:

- **Attributes:**
  - `CourseID`: Unique identifier for each course.
  - `CourseName`: Name of the course.
  - `Credits`: Number of credits associated with the course.

Teachers Table:

- **Attributes:**
  - `TeacherID`: Unique identifier for each teacher.
  - `CourseID`: Foreign key referencing the Courses table.
  - `FullName`: Full name of the teacher.
  - `Address`: Address of the teacher.
  - `Phone`: Phone number of the teacher.
  - `TeachingClass`: Class or section taught by the teacher.

Enrollments Table:

- **Attributes:**
  - `StudentID`: Foreign key referencing the Students table.

        - `CourseID`: Foreign key referencing the Courses table.

Assignments Table:

- **Attributes:**
    - `AssID`: Unique identifier for each assignment.
    - `CourseID`: Foreign key referencing the Courses table.
    - `StudentID`: Foreign key referencing the Students table.
    - `AssName`: Name or title of the assignment.
    - `Deadline`: Deadline date for the assignment.
    - `AssStatus`: Status of the assignment (e.g., Pending, Completed).

Exams Table:

- **Attributes:**
    - `ExamID`: Unique identifier for each exam.
    - `CourseID`: Foreign key referencing the Courses table.
    - `StudentID`: Foreign key referencing the Students table.
    - `ExamName`: Name or title of the exam.
    - `ExamDate`: Date of the exam.
    - `Grade`: Grade obtained by the student in the exam.

Attendance Table:

- **Attributes:**
    - `AttendanceID`: Unique identifier for each attendance record.
    - `StudentID`: Foreign key referencing the Students table.
    - `AttendanceDate`: Date of the attendance record.
    - `Status`: Attendance status (e.g., Present, Absent).

Payments Table:

- **Attributes:**
    - `PaymentID`: Unique identifier for each payment.
    - `StudentID`: Foreign key referencing the Students table.
    - `Amount`: Payment amount.
    - `PaymentDate`: Date of the payment.
    - `PayType`: Type of payment (e.g., Tuition).

Project's Functional View

Maintain Basic Data (project not work without it Entities)

- ✓ students **(Studentid(pk),** fullname, level,address,birthdate)
- ✓ teachers(**techid(pk),** courseid(Fk), fullname,address,phone,teaching class)
- ✓ courses (**courseid(pk),**coursename,credits)
- ✓ enrollments (**studentid(pk),** coursed (FK))
- ✓ assignments (**assid(pk),** courseid(FK), studentid(FK),assname,deadline,assStatus)
- ✓ exams (**examid(pk),** courseid(FK), studentid(FK),examname,examdate,grade)
- ✓ attendance(**attendid(pk),** studid(FK),Status,attendDate)
- ✓ payments (**Payid(pk),**studid(FK),amount,paydate,paytype)

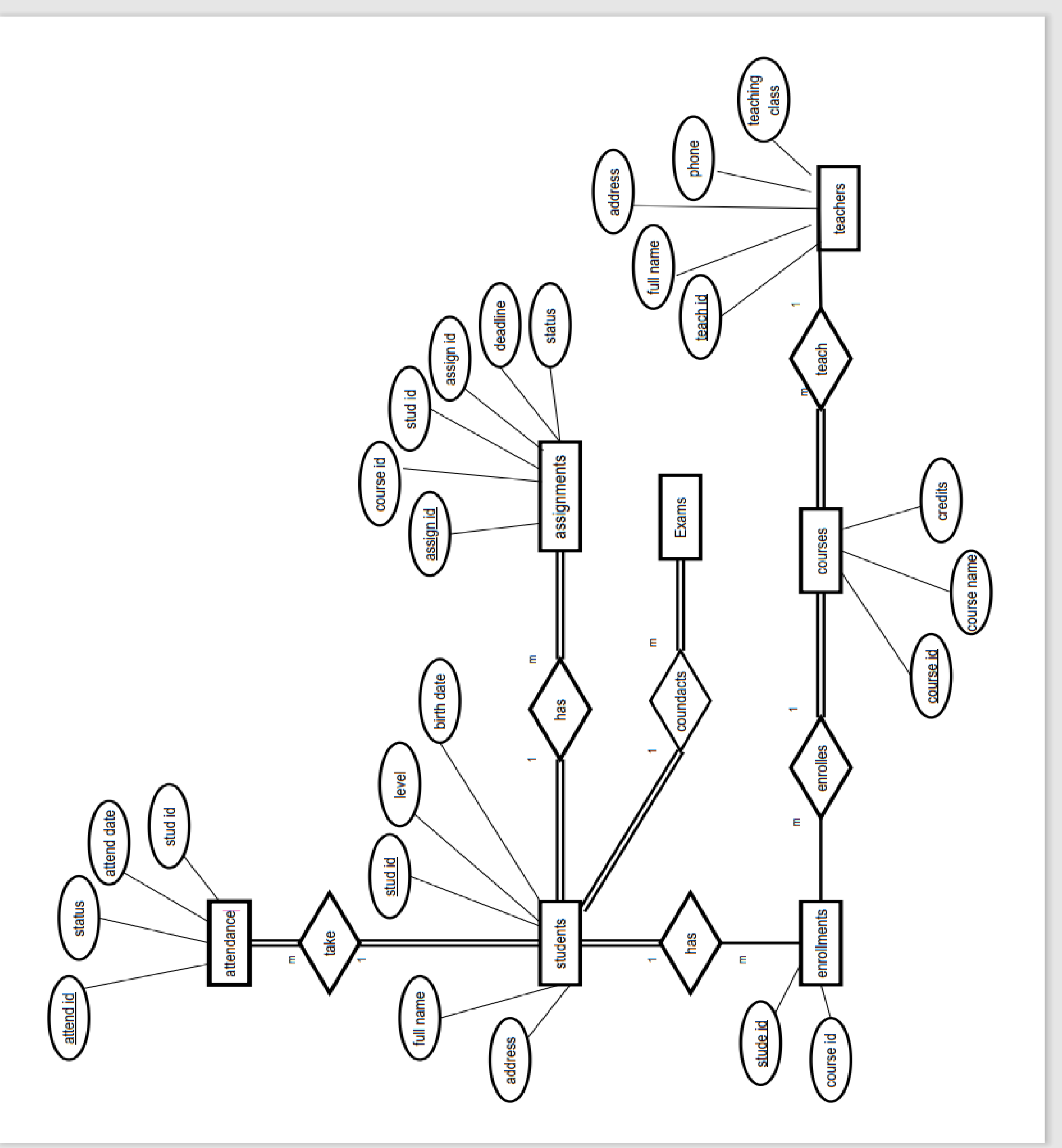
Issue Query Subsystems (it is Example)don't made as it , apply on your idea

- ✓ Search by student name.
- ✓ Search by teacher name.
- ✓ Search by course name.
- ✓ Search by exam name.
- ✓ Search by assignment name.

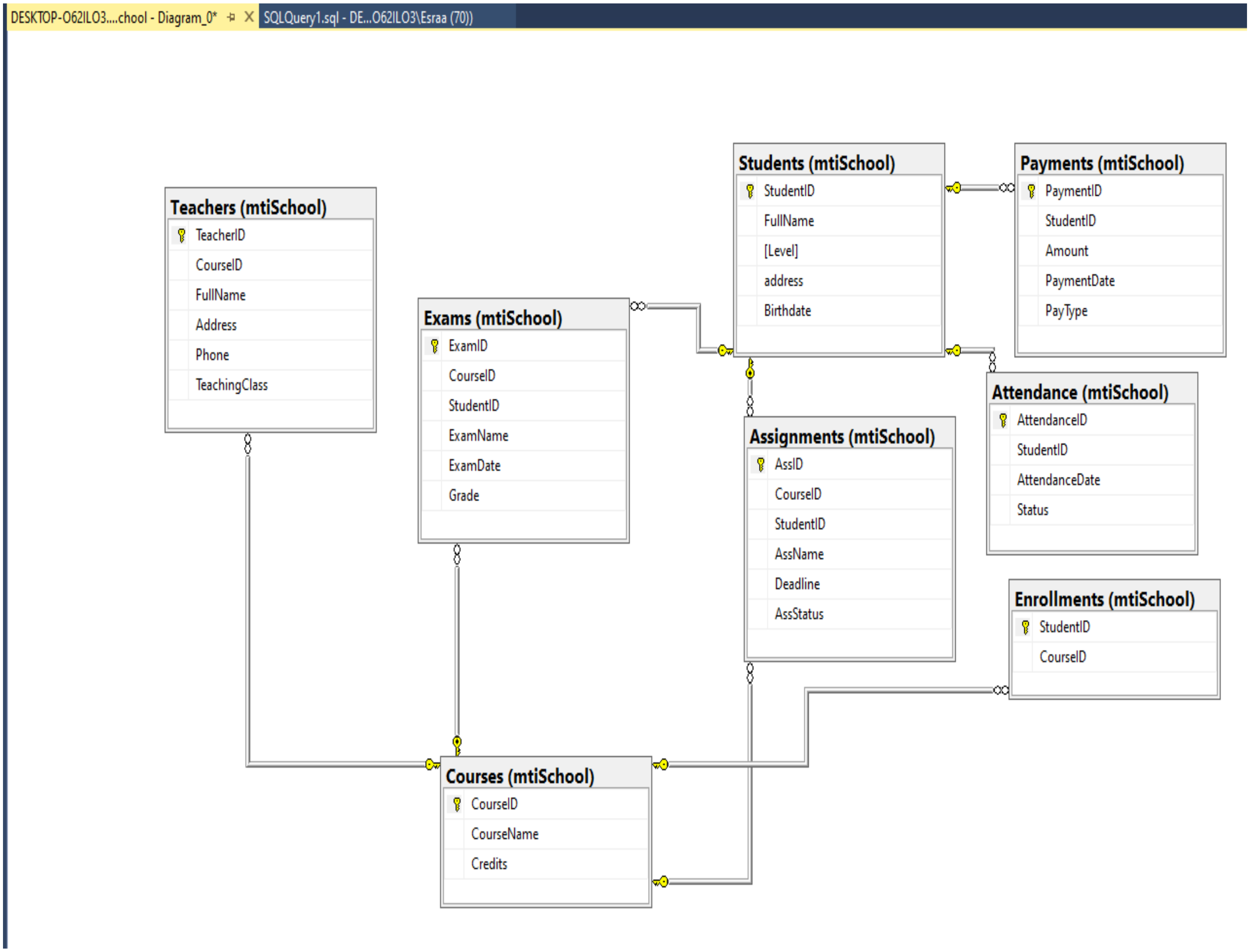
Perform transactions(write it to your project)

- ✓ What transactions need to Add (all entities).
- ✓ What transactions need to delete (all entities).
- ✓ What transactions need to update (all entitie

✓ Entity Relationship Diagram



Database Relationships and Tables with Sample Data


DESKTOP-O62ILO3....chool - Diagram_0*
SQLQuery1.sql - DE...O62ILO3\Esraa (70))

Teachers (mtiSchool)
TeacherID
CourseID
FullName
Address
Phone
TeachingClass

Exams (mtiSchool)
ExamID
CourseID
StudentID
ExamName
ExamDate
Grade

Students (mtiSchool)
StudentID
FullName
[Level]
address
Birthdate

Payments (mtiSchool)
PaymentID
StudentID
Amount
PaymentDate
PayType

Attendance (mtiSchool)
AttendanceID
StudentID
AttendanceDate
Status

Assignments (mtiSchool)
AssID
CourseID
StudentID
AssName
Deadline
AssStatus

Enrollments (mtiSchool)
StudentID
CourseID

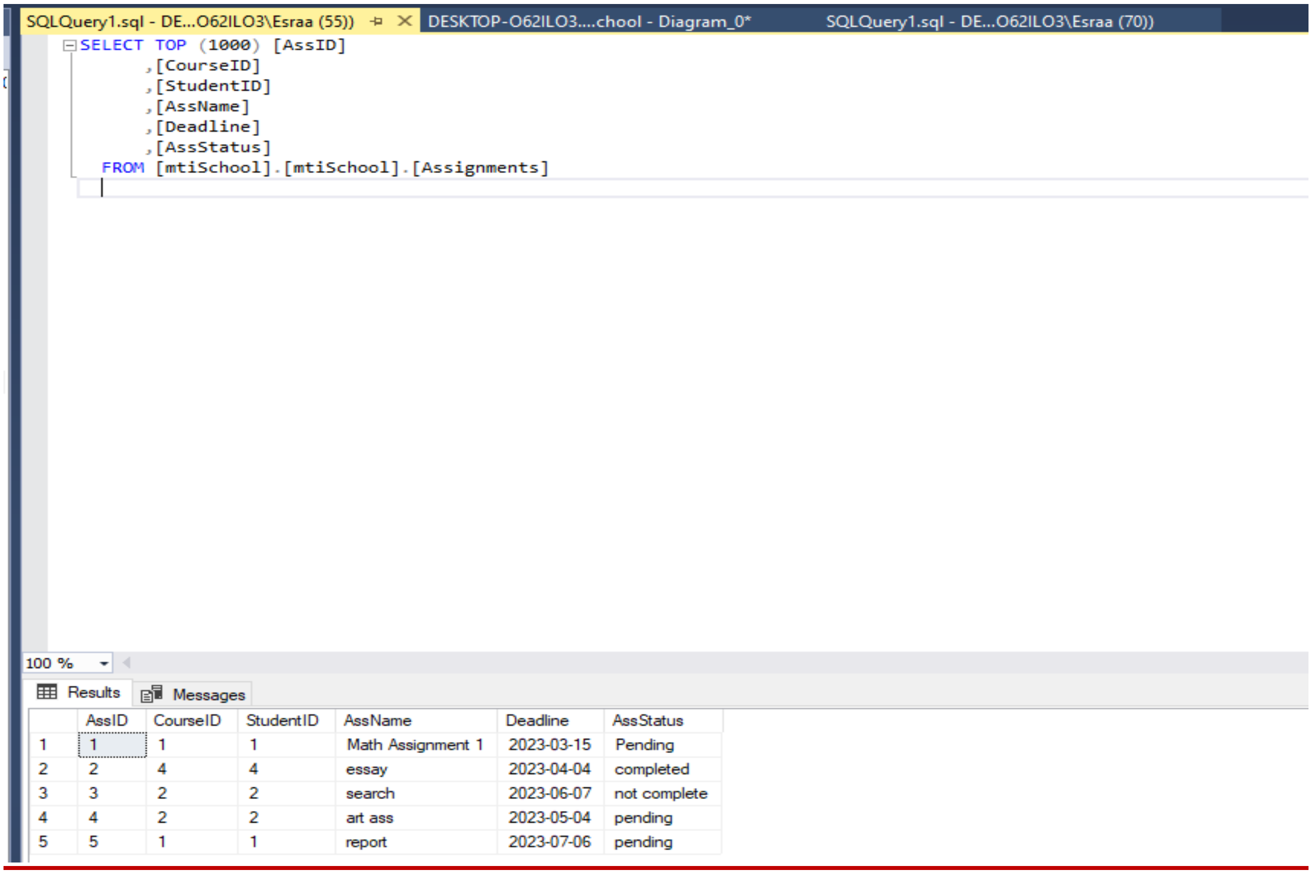
Courses (mtiSchool)
CourseID
CourseName
Credits

Sql (create mti school database ,create schema,create tables and insert sample data into tables)

```
CREATE DATABASE mtiSchool;
GO
USE mtiSchool;
GO
CREATE SCHEMA mtiSchool;
GO
USE mtiSchool;
GO
CREATE TABLE mtiSchool.Students (
    StudentID INT NOT NULL,
    FullName VARCHAR(50) NOT NULL,
        Level INT,
    address VARCHAR(90),
    Birthdate DATE,
       PRIMARY KEY (StudentID  ),
);
CREATE TABLE mtiSchool.Courses (
    CourseID INT NOT NULL,
    CourseName VARCHAR(30) NOT NULL,
    Credits INT NOT NULL,
    PRIMARY KEY (CourseID)
);
CREATE TABLE mtiSchool.Teachers (
    TeacherID INT NOT NULL ,
           CourseID INT NOT NULL,
    FullName VARCHAR(50) NOT NULL,
       Address VARCHAR(90),
    Phone VARCHAR(15),
    TeachingClass VARCHAR(5),
    PRIMARY KEY ( TeacherID ),
        FOREIGN KEY (CourseID) REFERENCES mtiSchool.Courses(CourseID) ON UPDATE CASCADE
ON DELETE CASCADE

);

CREATE TABLE mtiSchool.Enrollments (
    StudentID INT NOT NULL,
    CourseID INT NOT NULL,
    PRIMARY KEY (StudentID),
    FOREIGN KEY (CourseID) REFERENCES mtiSchool.Courses(CourseID) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE mtiSchool.Assignments (
    AssID INT NOT NULL,
    CourseID INT NOT NULL,
    StudentID INT NOT NULL,
    AssName VARCHAR(20) NOT NULL,
    Deadline DATE NOT NULL,
          AssStatus VARCHAR(20),
    PRIMARY KEY (AssID),
    FOREIGN KEY (CourseID) REFERENCES mtiSchool.Courses(CourseID) ON UPDATE CASCADE ON
DELETE CASCADE,
    FOREIGN KEY (StudentID) REFERENCES mtiSchool.Students(StudentID) ON UPDATE CASCADE ON
DELETE CASCADE
);
CREATE TABLE mtiSchool.Exams (
    ExamID INT NOT NULL,
    CourseID INT NOT NULL,
         StudentID INT NOT NULL,
    ExamName VARCHAR(30) NOT NULL,
    ExamDate DATE NOT NULL,
           Grade INT NOT NULL,
    PRIMARY KEY (ExamID),
    FOREIGN KEY (CourseID) REFERENCES mtiSchool.Courses(CourseID) ON UPDATE CASCADE ON
DELETE CASCADE,
       FOREIGN KEY ( StudentID ) REFERENCES mtiSchool.Students( StudentID ) ON UPDATE
CASCADE ON DELETE CASCADE
);
CREATE TABLE mtiSchool.Attendance (
    AttendanceID INT NOT NULL,
    StudentID INT NOT NULL,
    AttendanceDate DATE,
   Status VARCHAR(10) NOT NULL ,
    PRIMARY KEY (AttendanceID),
    FOREIGN KEY (StudentID) REFERENCES mtiSchool.Students(StudentID) ON UPDATE CASCADE ON
DELETE CASCADE,
);
CREATE TABLE mtiSchool.Payments (
    PaymentID INT NOT NULL,
    StudentID INT,
    Amount DECIMAL(10, 2),
    PaymentDate DATE,
        PayType VARCHAR(50),
        PRIMARY KEY ( PaymentID),
    FOREIGN KEY (StudentID) REFERENCES mtiSchool.Students(StudentID)
);



-- Insert data into Students table
INSERT INTO mtiSchool.Students (StudentID, FullName, Level, address, Birthdate)
VALUES
(1, 'John Doe', 10, '123 Main St', '2005-05-15'),
(2, 'Jane Smith', 11, '456 Oak St', '2004-09-22'),
(3, 'Bob Johnson', 9, '789 Pine St', '2006-02-10');

-- Insert data into Courses table
INSERT INTO mtiSchool.Courses (CourseID, CourseName, Credits)
VALUES
(1, 'Mathematics', 3),
(2, 'History', 4),
(3, 'Science', 3);

-- Insert data into Teachers table
INSERT INTO mtiSchool.Teachers (TeacherID, CourseID, FullName, Address, Phone,
TeachingClass)
VALUES
(1, 1, 'Mr. Adams', '111 Elm St', '123-456-7890', 'A101'),
(2, 2, 'Mrs. Davis', '222 Oak St', '987-654-3210', 'B202'),
(3, 3, 'Mr. Turner', '333 Pine St', '555-123-4567', 'C303');

-- Insert data into Enrollments table
INSERT INTO mtiSchool.Enrollments (StudentID, CourseID)
VALUES
(1, 1),
(1, 2),
(2, 2),
(3, 3);

-- Insert data into Assignments table
INSERT INTO mtiSchool.Assignments (AssID, CourseID, StudentID, AssName, Deadline,
AssStatus)
VALUES
(1, 1, 1, 'Math Assignment 1', '2023-03-15', 'Pending'),
(2, 2, 2, 'History Assignment 1', '2023-03-20', 'Completed'),
(3, 3, 3, 'Science Assignment 1', '2023-03-25', 'Pending');

-- Insert data into Exams table
INSERT INTO mtiSchool.Exams (ExamID, CourseID, StudentID, ExamName, ExamDate, Grade)
VALUES
(1, 1, 1, 'Math Exam 1', '2023-04-10', 85),
(2, 2, 2, 'History Exam 1', '2023-04-12', 92),
(3, 3, 3, 'Science Exam 1', '2023-04-15', 78);

-- Insert data into Attendance table
INSERT INTO mtiSchool.Attendance (AttendanceID, StudentID, AttendanceDate, Status)
VALUES
(1, 1, '2023-05-01', 'Present'),
(2, 2, '2023-05-01', 'Present'),
(3, 3, '2023-05-01', 'Absent');

-- Insert data into Payments table
INSERT INTO mtiSchool.Payments (PaymentID, StudentID, Amount, PaymentDate, PayType)
VALUES
(1, 1, 500.00, '2023-05-05', 'Tuition'),
(2, 2, 600.00, '2023-05-05', 'Tuition'),
(3, 3, 450.00, '2023-05-05', 'Tuition');
```

Input / Output Interface Design

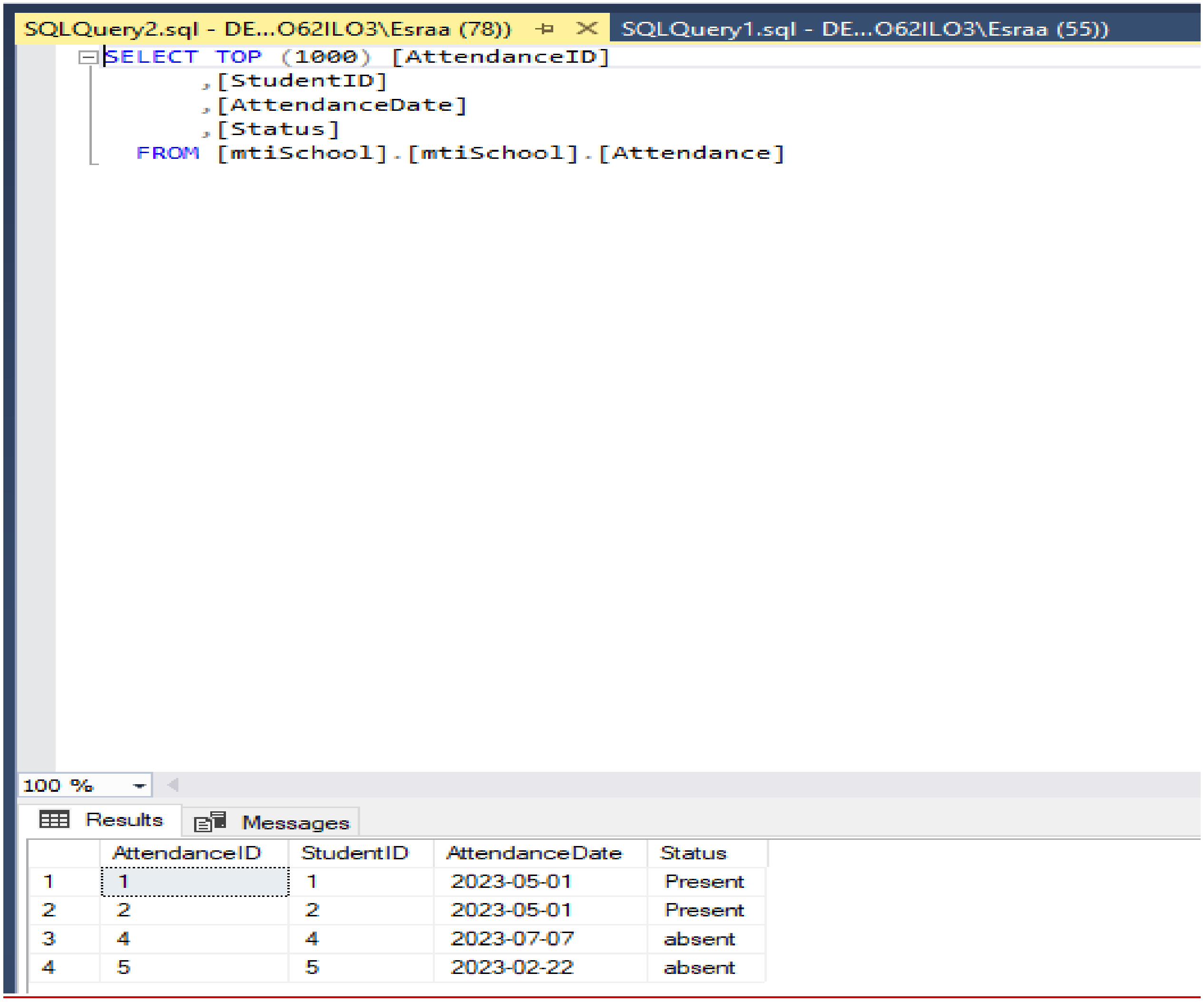
Queries



SQLQuery1.sql - DE...O62ILO3\Esraa (55))
DESKTOP-O62ILO3....chool - Diagram_0*
SQLQuery1.sql - DE...O62ILO3\Esraa (70))
100 %
Results
Messages

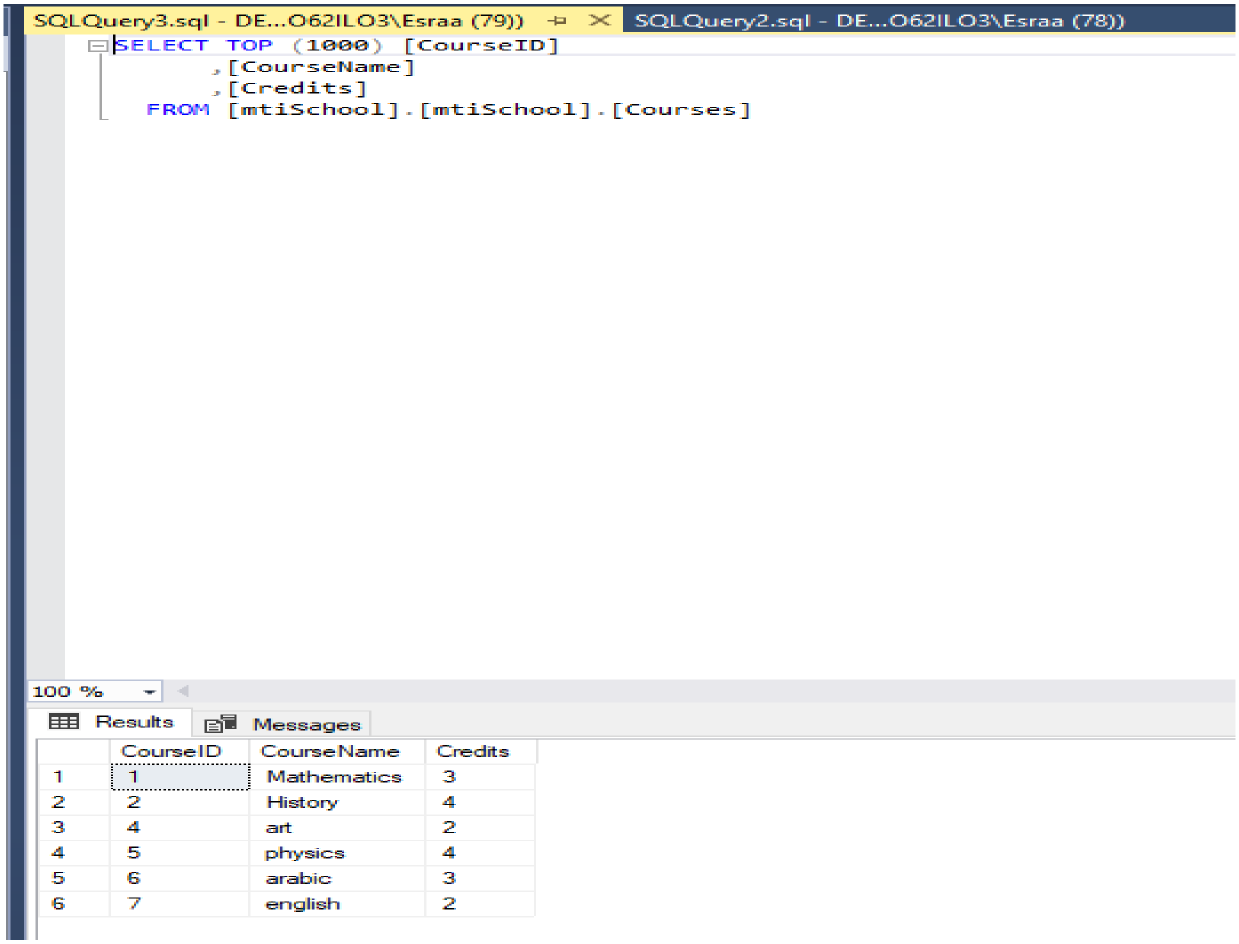
```
SELECT TOP (1000) [AssID]
      ,[CourseID]
      ,[StudentID]
      ,[AssName]
      ,[Deadline]
      ,[AssStatus]
  FROM [mtiSchool].[mtiSchool].[Assignments]
```

	AssID	CourseID	StudentID	AssName	Deadline	AssStatus
1	1	1	1	Math Assignment 1	2023-03-15	Pending
2	2	4	4	essay	2023-04-04	completed
3	3	2	2	search	2023-06-07	not complete
4	4	2	2	art ass	2023-05-04	pending
5	5	1	1	report	2023-07-06	pending


SQLQuery2.sql - DE...O62ILO3\Esraa (78))
SQLQuery1.sql - DE...O62ILO3\Esraa (55))
100 %
Results
Messages

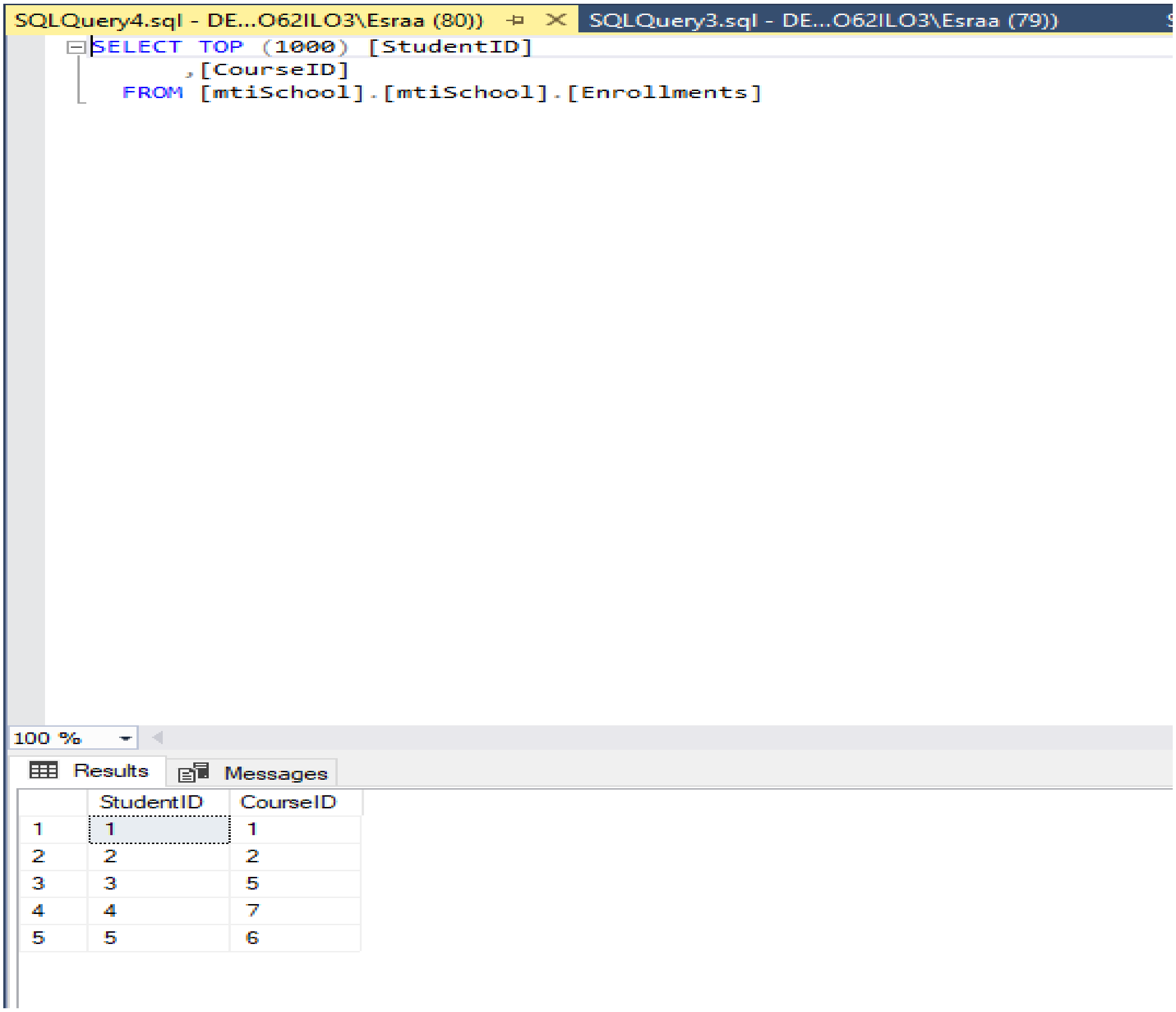
```
SELECT TOP (1000) [AttendanceID]
      ,[StudentID]
      ,[AttendanceDate]
      ,[Status]
  FROM [mtiSchool].[mtiSchool].[Attendance]
```

	AttendanceID	StudentID	AttendanceDate	Status
1	1	1	2023-05-01	Present
2	2	2	2023-05-01	Present
3	4	4	2023-07-07	absent
4	5	5	2023-02-22	absent


SQLQuery3.sql - DE...O62ILO3\Esraa (79))
SQLQuery2.sql - DE...O62ILO3\Esraa (78))
100 %
Results
Messages

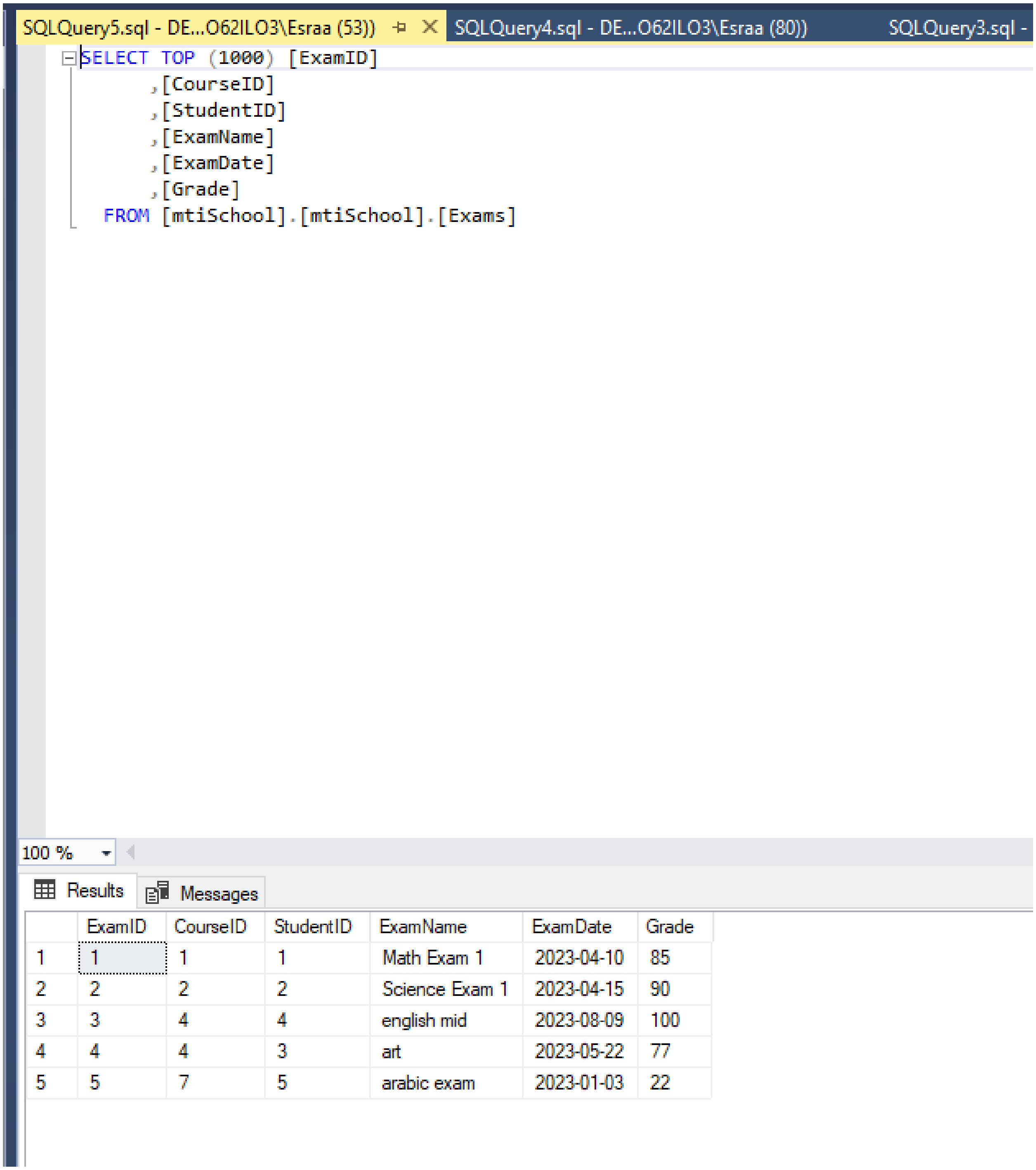
```sql
SELECT TOP (1000) [CourseID]
      ,[CourseName]
      ,[Credits]
  FROM [mtiSchool].[mtiSchool].[Courses]
```

	CourseID	CourseName	Credits
1	1	Mathematics	3
2	2	History	4
3	4	art	2
4	5	physics	4
5	6	arabic	3
6	7	english	2


SQLQuery4.sql - DE...O62ILO3\Esraa (80))
SQLQuery3.sql - DE...O62ILO3\Esraa (79))
100 %
Results
Messages

```
SELECT TOP (1000) [StudentID]
      ,[CourseID]
  FROM [mtiSchool].[mtiSchool].[Enrollments]
```

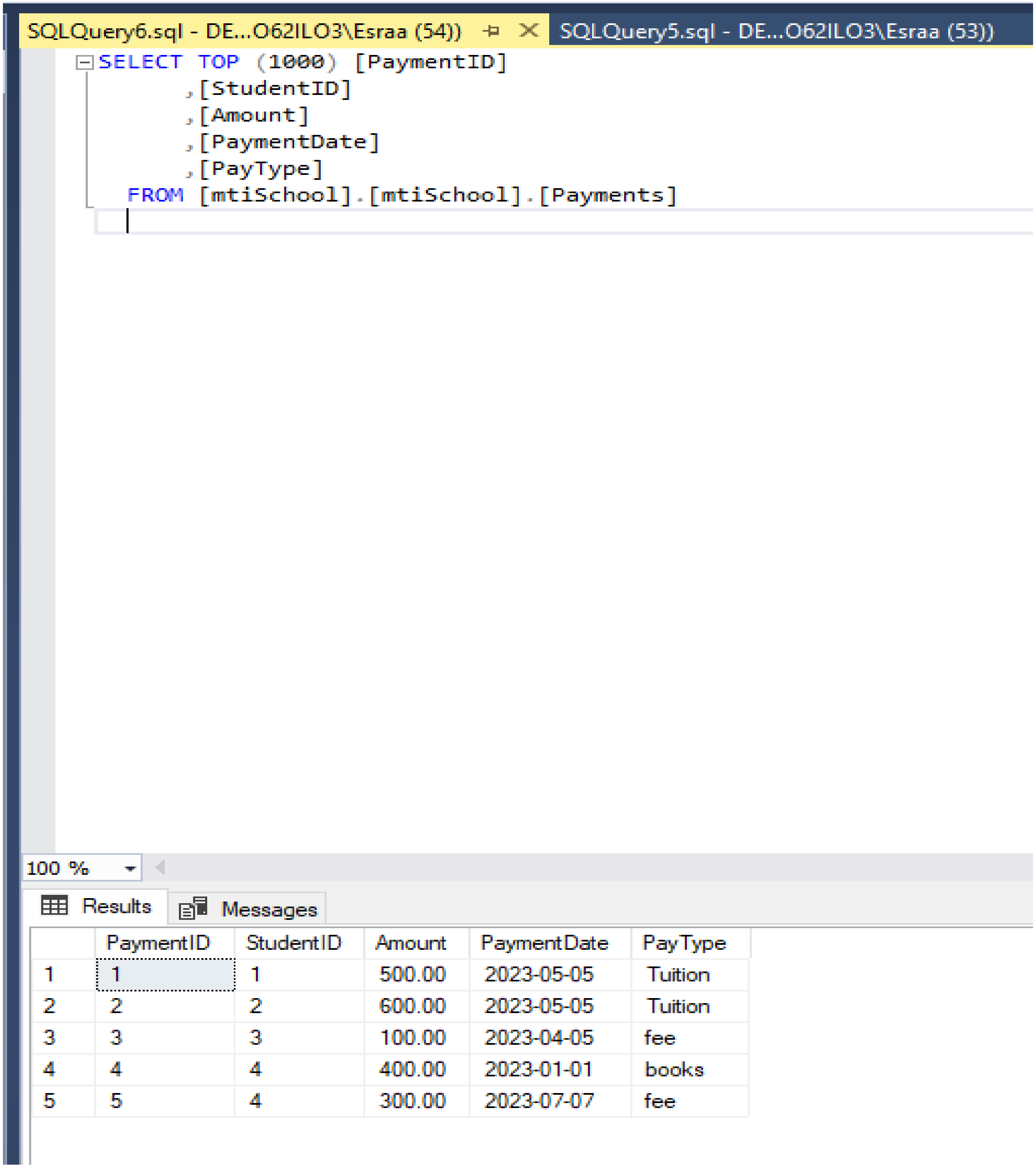
```
   StudentID  CourseID
1  1          1
2  2          2
3  3          5
4  4          7
5  5          6
```



SQLQuery5.sql - DE...O62ILO3\Esraa (53))
SQLQuery4.sql - DE...O62ILO3\Esraa (80))
SQLQuery3.sql -
100 %
Results
Messages

```sql
SELECT TOP (1000) [ExamID]
      ,[CourseID]
      ,[StudentID]
      ,[ExamName]
      ,[ExamDate]
      ,[Grade]
  FROM [mtiSchool].[mtiSchool].[Exams]
```

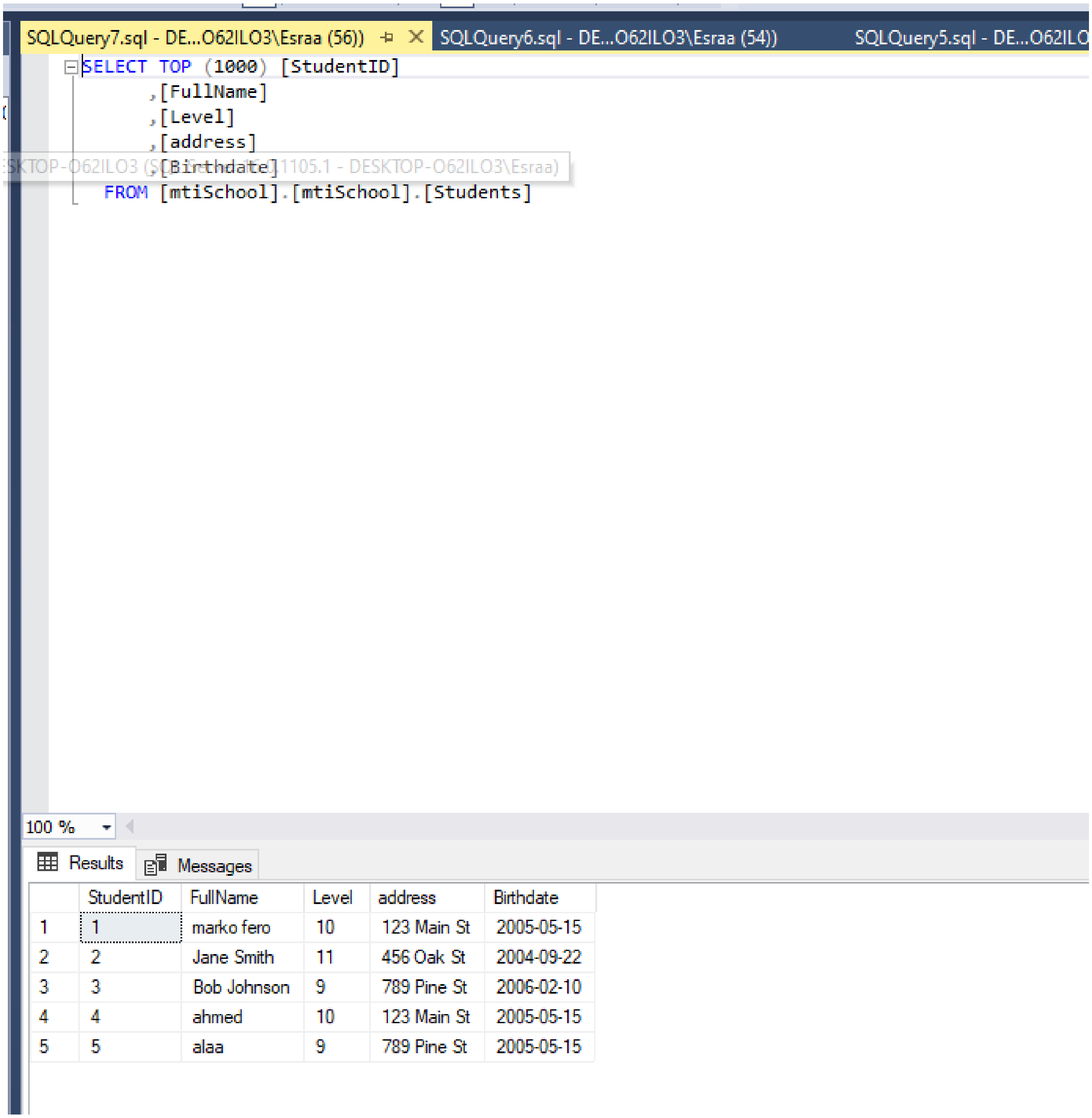
	ExamID	CourseID	StudentID	ExamName	ExamDate	Grade
1	1	1	1	Math Exam 1	2023-04-10	85
2	2	2	2	Science Exam 1	2023-04-15	90
3	3	4	4	english mid	2023-08-09	100
4	4	4	3	art	2023-05-22	77
5	5	7	5	arabic exam	2023-01-03	22



SQLQuery6.sql - DE...O62ILO3\Esraa (54))
SQLQuery5.sql - DE...O62ILO3\Esraa (53))
100 %
Results
Messages

```sql
SELECT TOP (1000) [PaymentID]
      ,[StudentID]
      ,[Amount]
      ,[PaymentDate]
      ,[PayType]
  FROM [mtiSchool].[mtiSchool].[Payments]
```

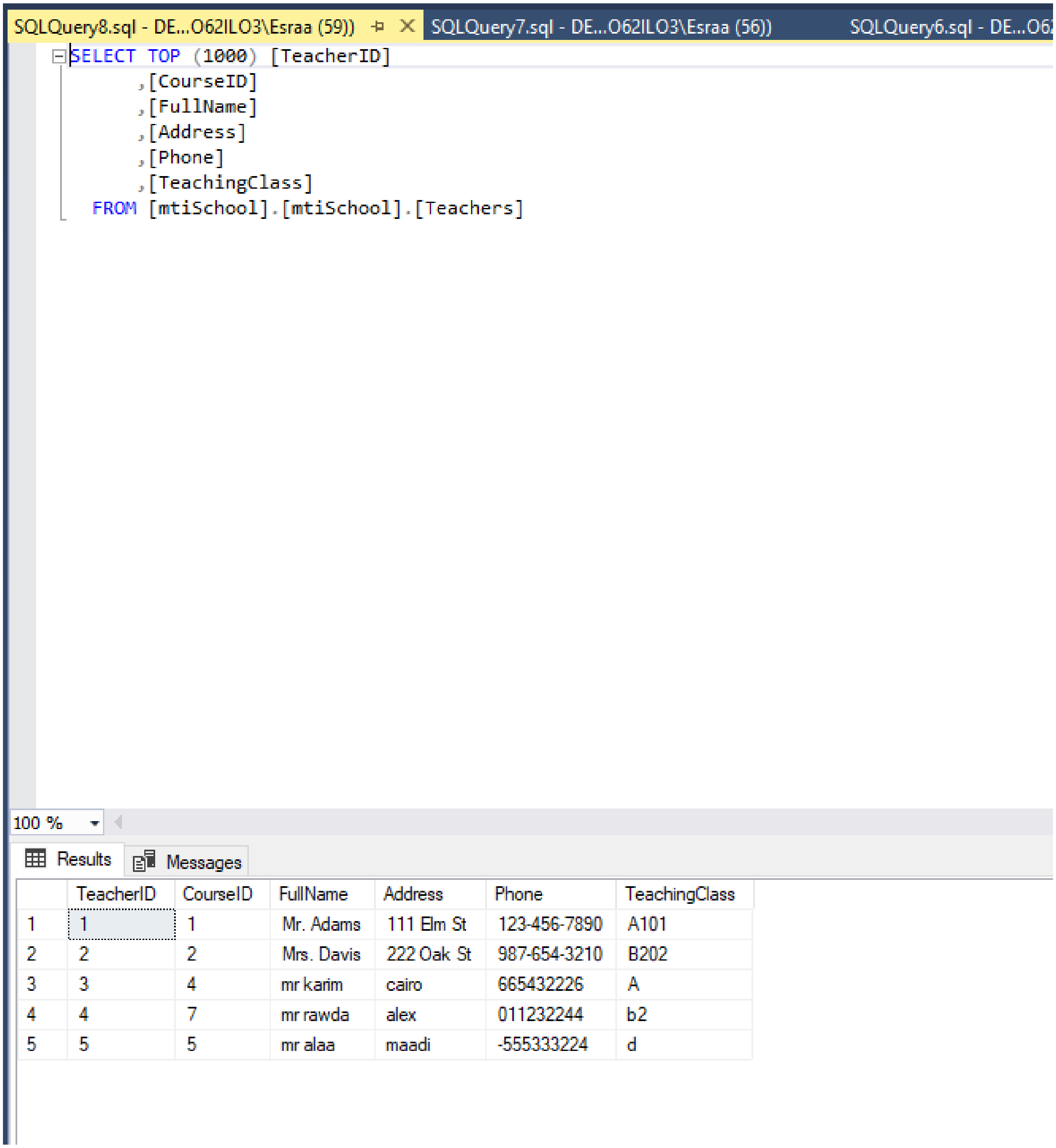
	PaymentID	StudentID	Amount	PaymentDate	PayType
1	1	1	500.00	2023-05-05	Tuition
2	2	2	600.00	2023-05-05	Tuition
3	3	3	100.00	2023-04-05	fee
4	4	4	400.00	2023-01-01	books
5	5	4	300.00	2023-07-07	fee



SQLQuery7.sql - DE...O62ILO3\Esraa (56))
SQLQuery6.sql - DE...O62ILO3\Esraa (54))
SQLQuery5.sql - DE...O62ILO
SKTOP-O62ILO3 (S... 1105.1 - DESKTOP-O62ILO3\Esraa)
100 %
Results
Messages

```
SELECT TOP (1000) [StudentID]
      ,[FullName]
      ,[Level]
      ,[address]
      ,[Birthdate]
  FROM [mtiSchool].[mtiSchool].[Students]
```

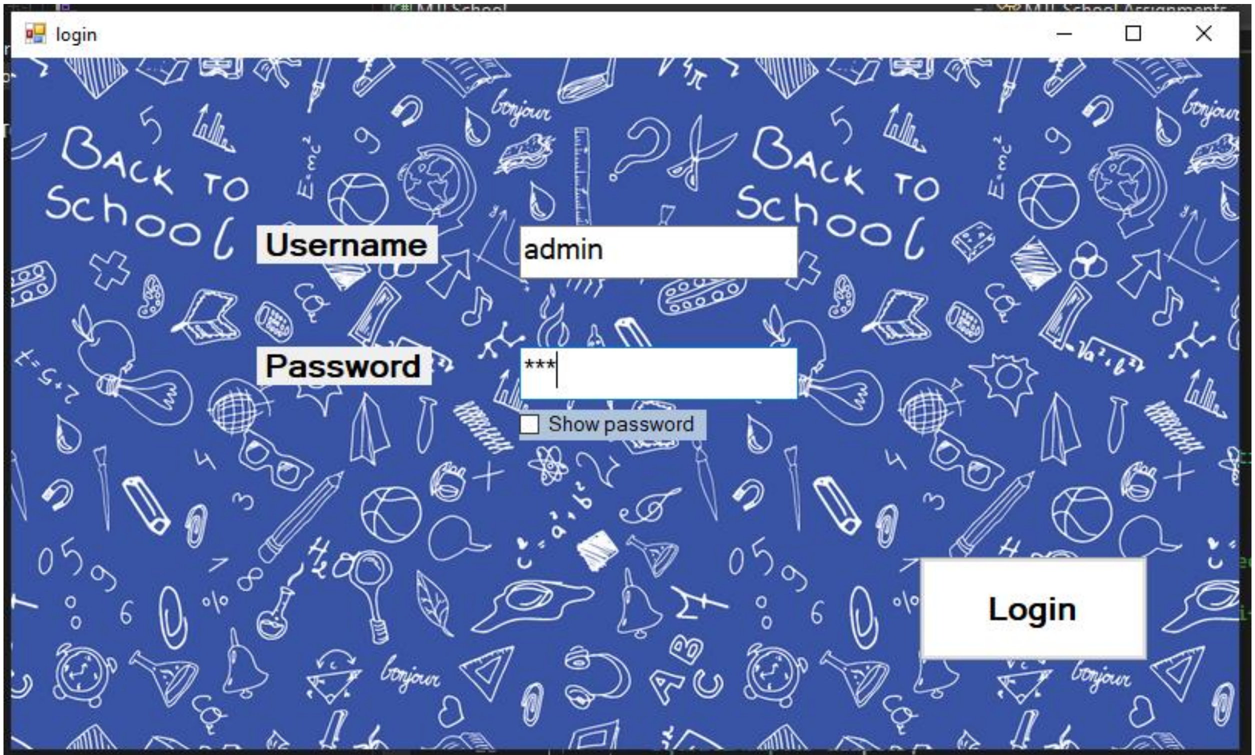
	StudentID	FullName	Level	address	Birthdate
1	1	marko fero	10	123 Main St	2005-05-15
2	2	Jane Smith	11	456 Oak St	2004-09-22
3	3	Bob Johnson	9	789 Pine St	2006-02-10
4	4	ahmed	10	123 Main St	2005-05-15
5	5	alaa	9	789 Pine St	2005-05-15


SQLQuery8.sql - DE...O62ILO3\Esraa (59))
SQLQuery7.sql - DE...O62ILO3\Esraa (56))
SQLQuery6.sql - DE...O6
100 %
Results
Messages

```sql
SELECT TOP (1000) [TeacherID]
      ,[CourseID]
      ,[FullName]
      ,[Address]
      ,[Phone]
      ,[TeachingClass]
  FROM [mtiSchool].[mtiSchool].[Teachers]
```

	TeacherID	CourseID	FullName	Address	Phone	TeachingClass
1	1	1	Mr. Adams	111 Elm St	123-456-7890	A101
2	2	2	Mrs. Davis	222 Oak St	987-654-3210	B202
3	3	4	mr karim	cairo	665432226	A
4	4	7	mr rawda	alex	011232244	b2
5	5	5	mr alaa	maadi	-555333224	d

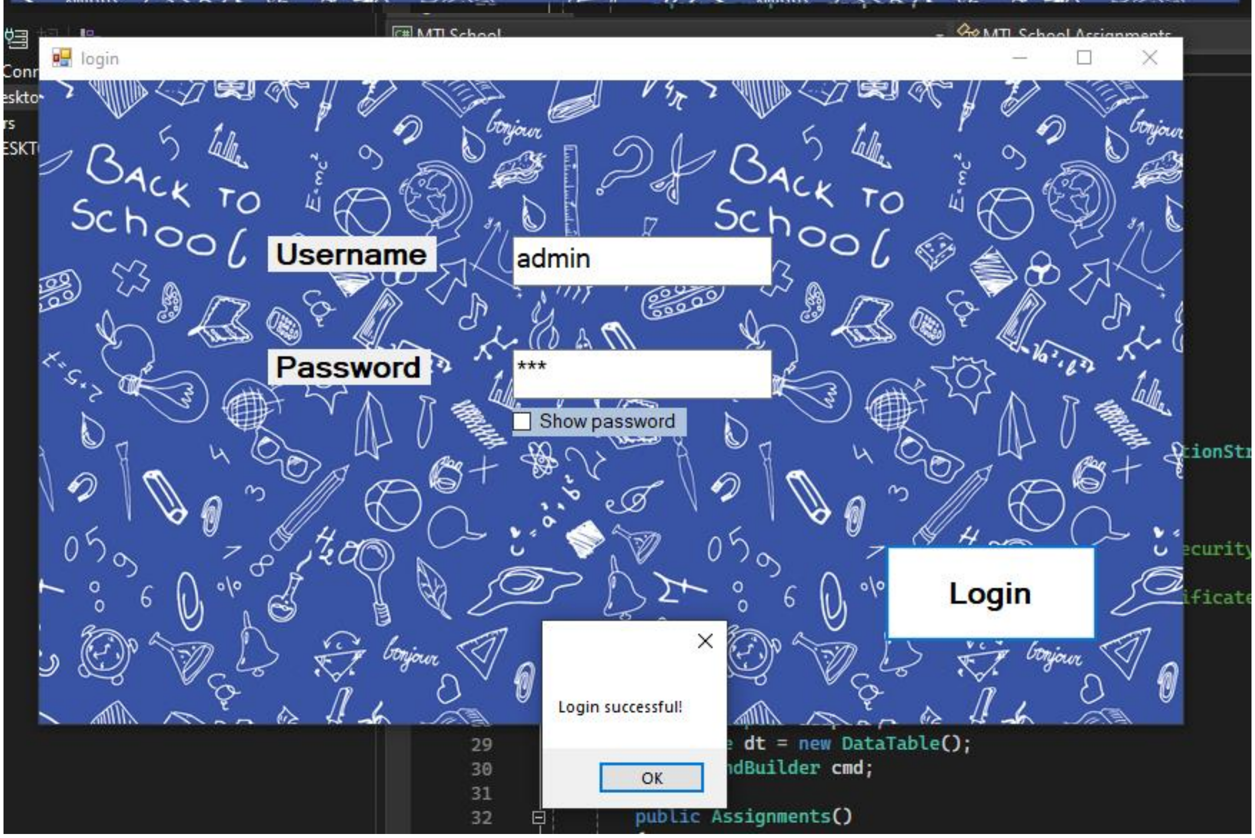
Forms(login form)



Username admin

Password ***

□ Show password



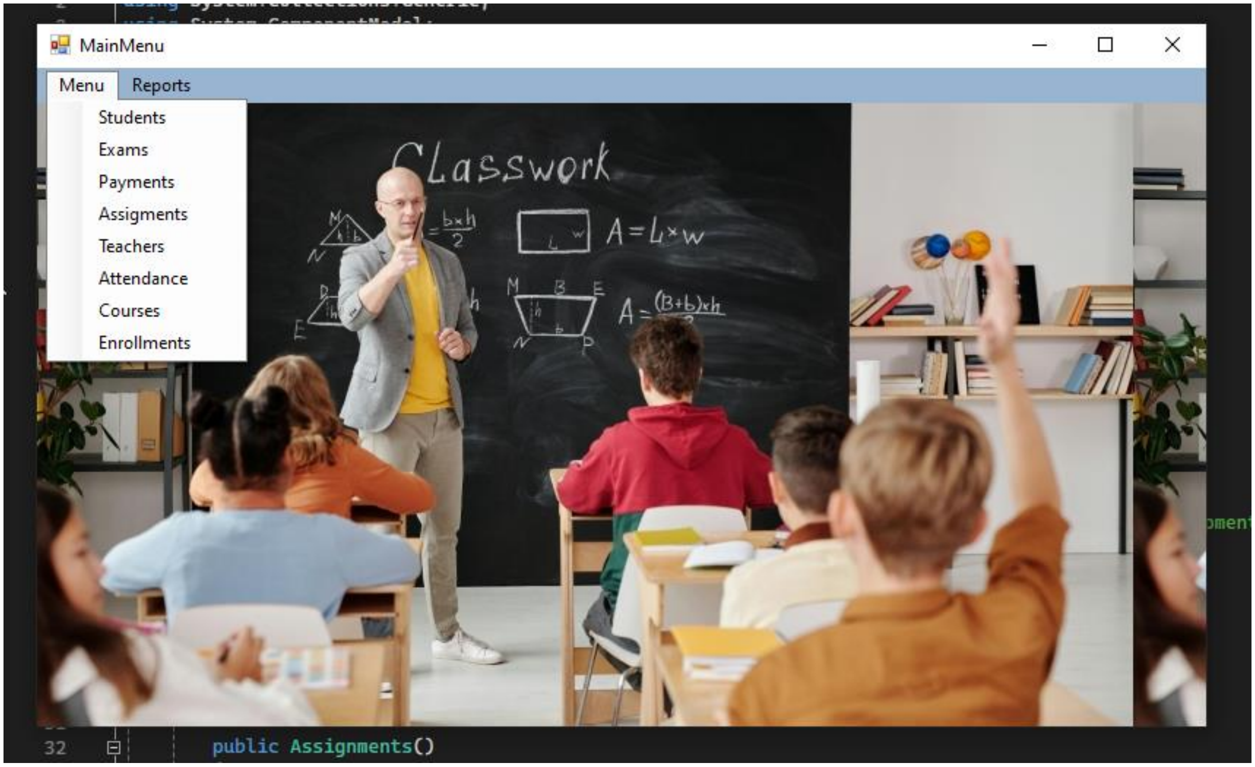
Username admin

Password ***

□ Show password

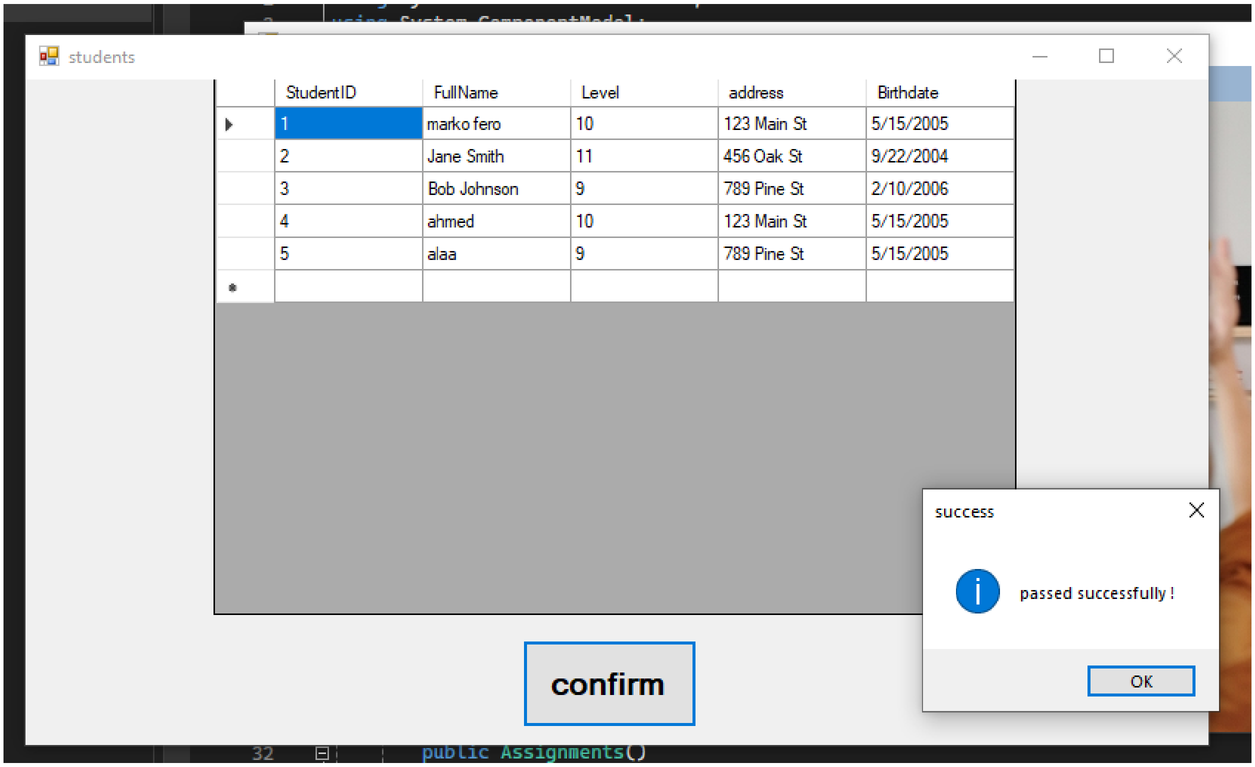
Login successful!

Main Form



```
public Assignments()
```

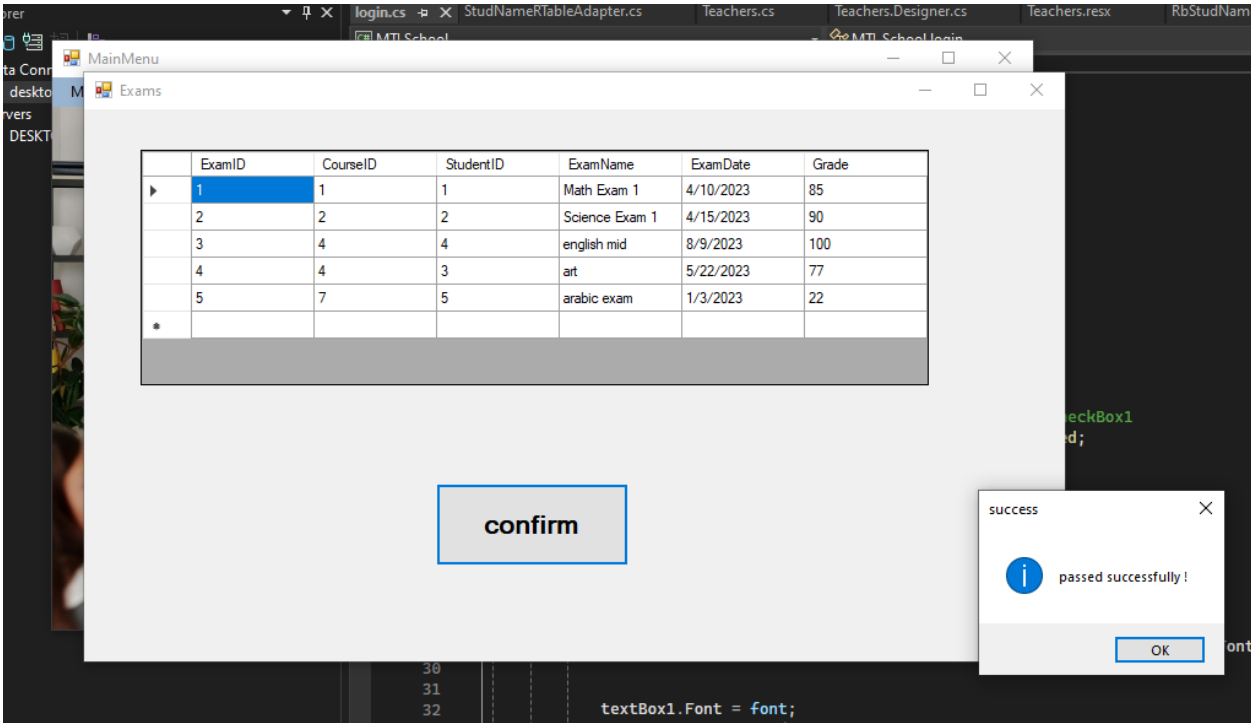
Transactions
(Add , Update ,Delete)



	StudentID	FullName	Level	address	Birthdate
▶	1	marko fero	10	123 Main St	5/15/2005
	2	Jane Smith	11	456 Oak St	9/22/2004
	3	Bob Johnson	9	789 Pine St	2/10/2006
	4	ahmed	10	123 Main St	5/15/2005
	5	alaa	9	789 Pine St	5/15/2005
*					

passed successfully !

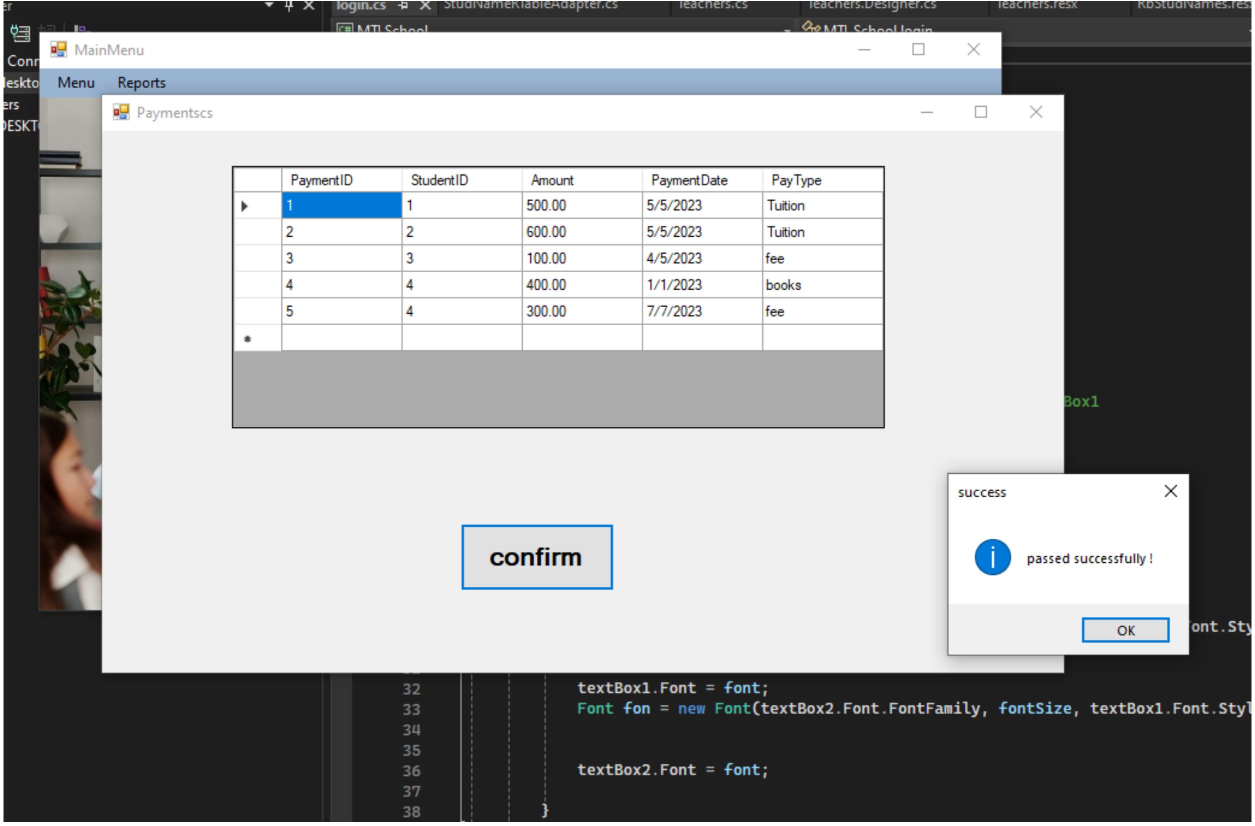
```
using System.ComponentModel;
public Assignments()
```



ExamID	CourseID	StudentID	ExamName	ExamDate	Grade
1	1	1	Math Exam 1	4/10/2023	85
2	2	2	Science Exam 1	4/15/2023	90
3	4	4	english mid	8/9/2023	100
4	4	3	art	5/22/2023	77
5	7	5	arabic exam	1/3/2023	22

passed successfully !

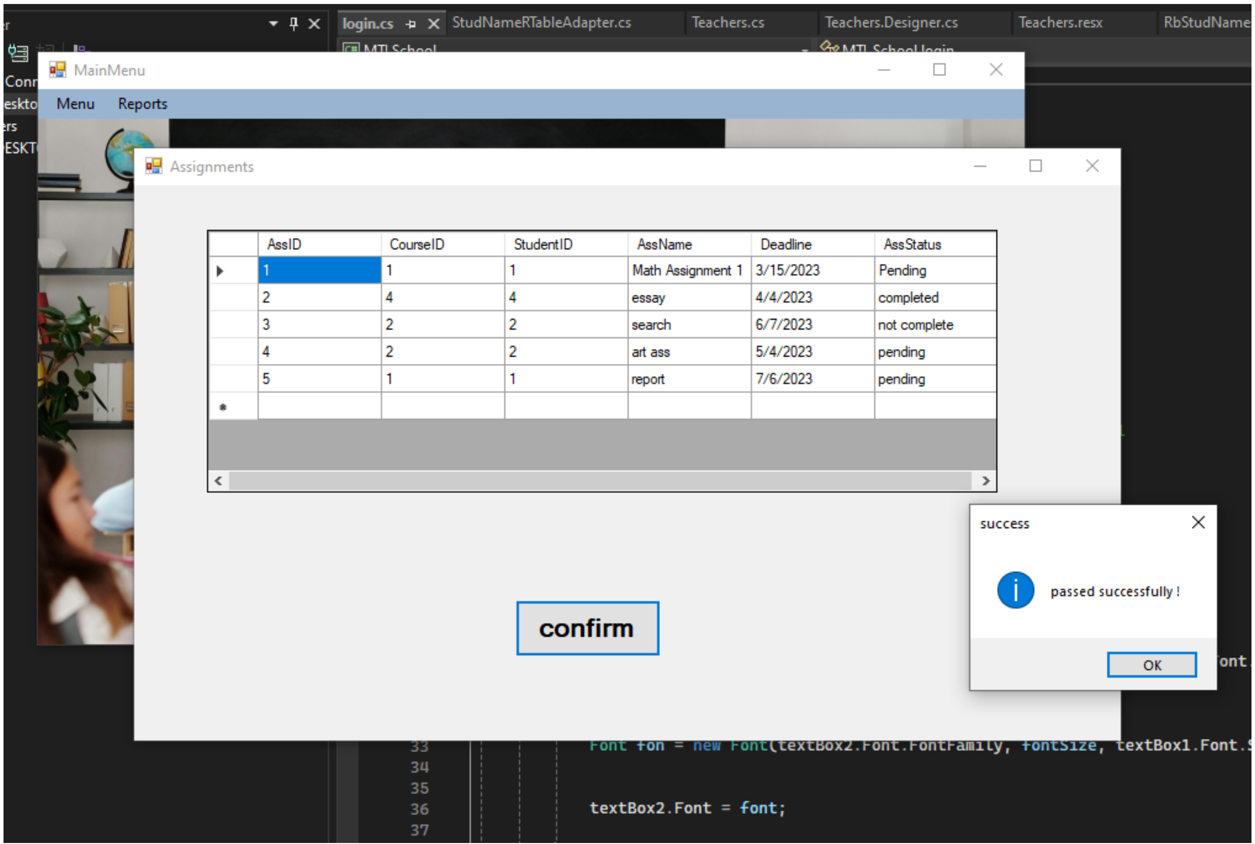
```
textBox1.Font = font;
```



PaymentID	StudentID	Amount	PaymentDate	PayType
1	1	500.00	5/5/2023	Tuition
2	2	600.00	5/5/2023	Tuition
3	3	100.00	4/5/2023	fee
4	4	400.00	1/1/2023	books
5	4	300.00	7/7/2023	fee

passed successfully !

```
            textBox1.Font = font;
            Font fon = new Font(textBox2.Font.FontFamily, fontSize, textBox1.Font.Styl


            textBox2.Font = font;
        }
```

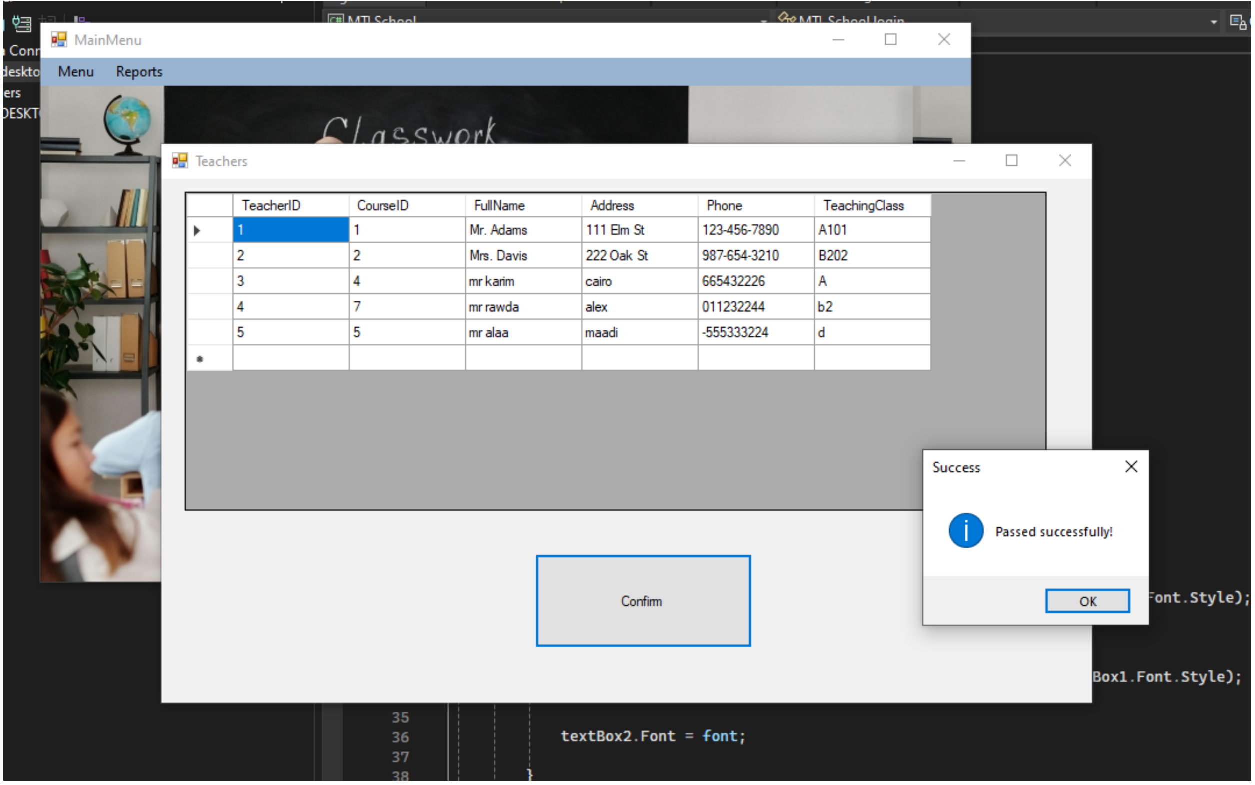


	AssID	CourseID	StudentID	AssName	Deadline	AssStatus
▶	1	1	1	Math Assignment 1	3/15/2023	Pending
	2	4	4	essay	4/4/2023	completed
	3	2	2	search	6/7/2023	not complete
	4	2	2	art ass	5/4/2023	pending
	5	1	1	report	7/6/2023	pending
*						

success

passed successfully !

```
Font fon = new Font(textBox2.Font.FontFamily, fontSize, textBox1.Font.S

textBox2.Font = font;
```

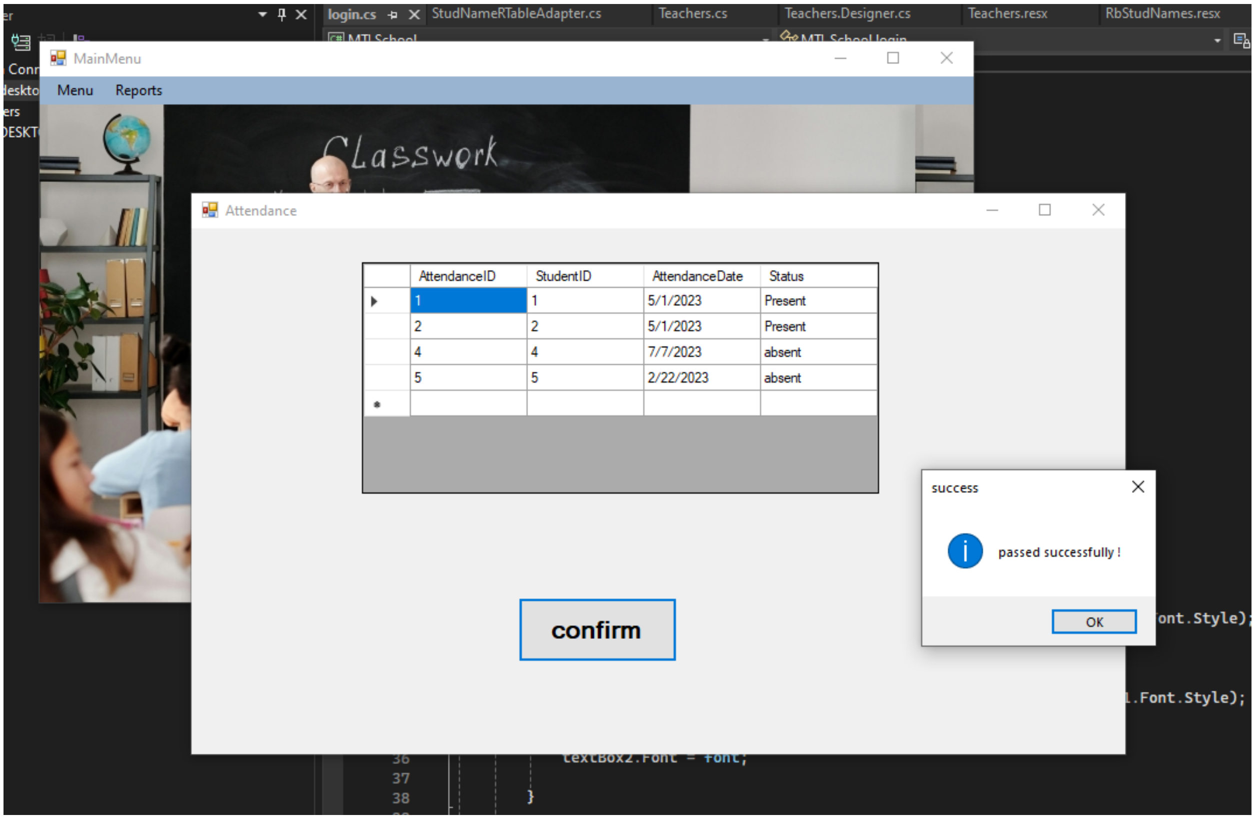


	TeacherID	CourseID	FullName	Address	Phone	TeachingClass
▶	1	1	Mr. Adams	111 Elm St	123-456-7890	A101
	2	2	Mrs. Davis	222 Oak St	987-654-3210	B202
	3	4	mr karim	cairo	665432226	A
	4	7	mr rawda	alex	011232244	b2
	5	5	mr alaa	maadi	-555333224	d
*						

Success

Passed successfully!

```
Font.Style);
Box1.Font.Style);

textBox2.Font = font;
}
```

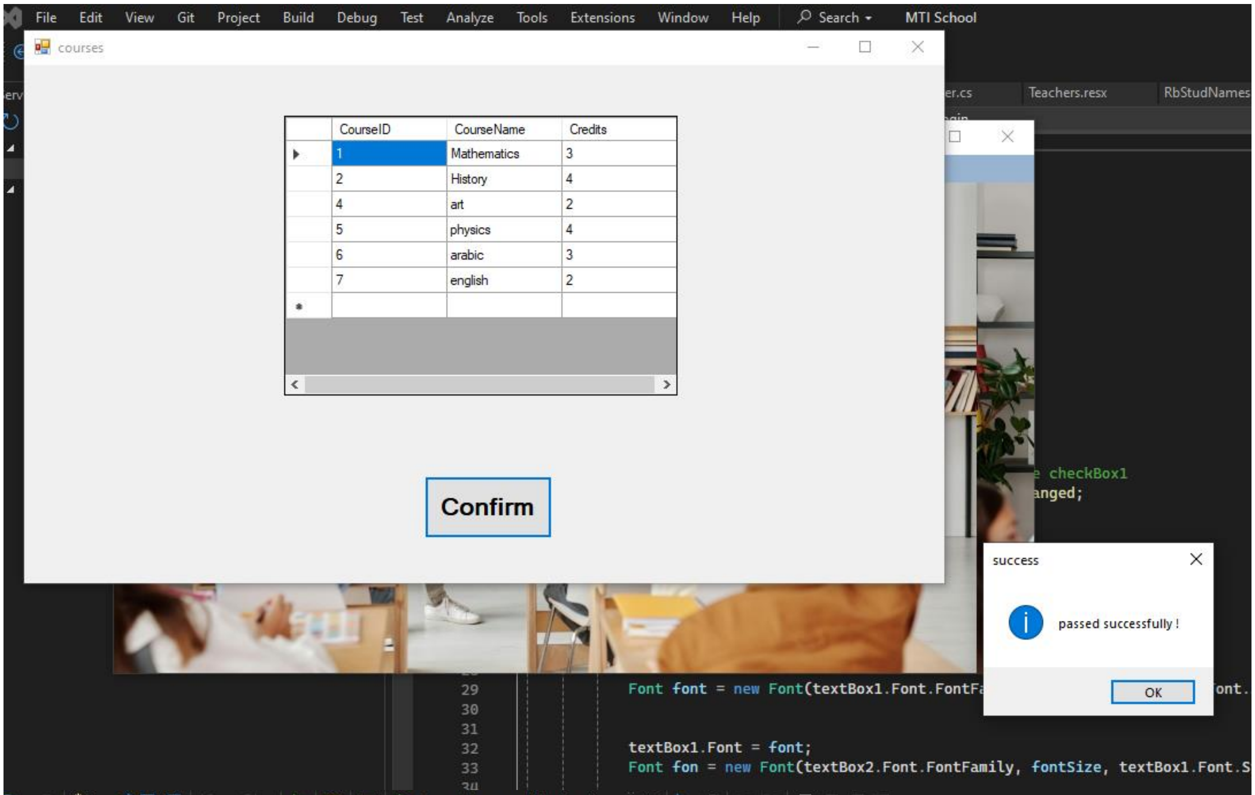


	AttendanceID	StudentID	AttendanceDate	Status
▸	1	1	5/1/2023	Present
	2	2	5/1/2023	Present
	4	4	7/7/2023	absent
	5	5	2/22/2023	absent
*				

success

passed successfully !

```
ont.Style);
l.Font.Style);
textBox2.Font = font;

}
```



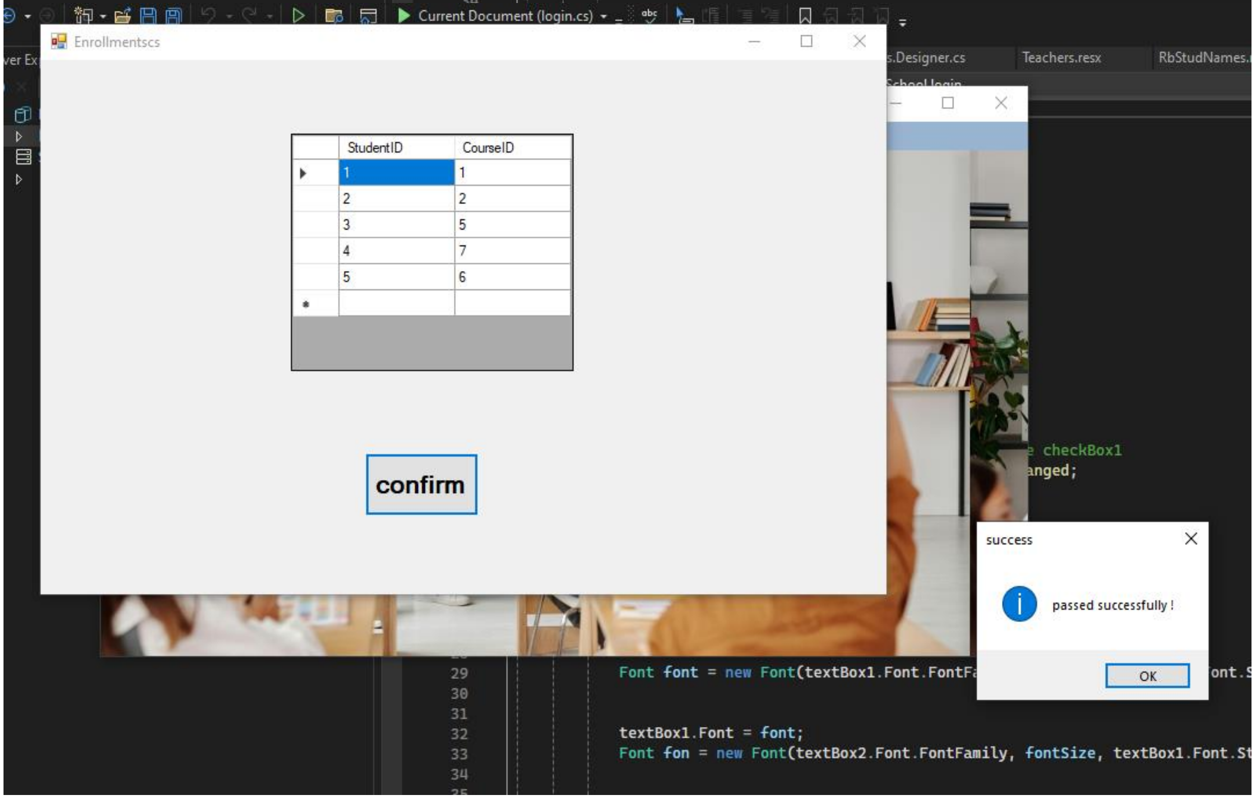
	CourseID	CourseName	Credits
▸	1	Mathematics	3
	2	History	4
	4	art	2
	5	physics	4
	6	arabic	3
	7	english	2
*			

success

passed successfully !

OK

```
e checkBox1
anged;

            Font font = new Font(textBox1.Font.FontFa                    ont.


            textBox1.Font = font;
            Font fon = new Font(textBox2.Font.FontFamily, fontSize, textBox1.Font.S
```



	StudentID	CourseID
▸	1	1
	2	2
	3	5
	4	7
	5	6
*		

success

passed successfully !

OK

```
e checkBox1
anged;

            Font font = new Font(textBox1.Font.FontFa                    ont.S


            textBox1.Font = font;
            Font fon = new Font(textBox2.Font.FontFamily, fontSize, textBox1.Font.St
```