

CSE 581 – Intro To Database Management Systems

Project 1

Rachana Mohan Fulsundar

SUID : 512373928

11/11/2022

Abstract :

In this project, we'll use SQL Server Management Studio to create MyCollege database and enter SQL statements and run them against this database. This database consists Courses, Departments, Instructors, StudentCourses, Students and Tuition tables.

We will first do database setup by executing the scripts provided, then perform DML operations on this database. We'll also write views, scripts, stored procedures, functions and design database diagrams.

Table of Contents

Section A : Database Setup	4
Section B : Essential SQL Skills.....	9
Section C : Advanced SQL Skills (views/stores procedures/ functions / scripts)	29
Section D : Database Design	38

Section A : Database Setup

1. Download sql files from this project tab and open CreatMyCollege.sql in SQL server management studio. Execute the entire script and show the message in the Message tab, indicating the script is executed successfully.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'DESKTOP-5MGKLNNU' is selected. In the center pane, a query window titled 'CreateMyCollege(3)...GKLNNU\admin (52)' contains the following T-SQL script:

```
USE master
GO

IF DB_ID('MyCollege') IS NOT NULL
    DROP DATABASE MyCollege
CREATE DATABASE MyCollege
GO

USE [MyCollege]
/
***** Object: Table [dbo].[Courses] Script Date: 10/12/2022 10:15:00 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

(1 row affected)
```

The status bar at the bottom indicates 'Query executed successfully.' The system tray shows the date and time as 08-11-2022 13:42.

2. Navigate through the database objects and view the column definitions for each table. Open a new Query Editor window. Show details in Courses table and Instructors table using SELECT statement.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The central pane displays the results of a query: 'SELECT * FROM Courses'. The results grid contains 25 rows of course information, including CourseID, CourseNumber, CourseDescription, CourseUnits, DepartmentID, and InstructorID. The status bar at the bottom indicates the query was executed successfully on 'DESKTOP-5MGKLN0' at '08-11-2022 00:00:00'.

CourseID	CourseNumber	CourseDescription	CourseUnits	DepartmentID	InstructorID
1	36598	Beginning Accounting	3	1	1
2	48926	Abstract Algebra	3	4	5
3	14862	Primary Education	3	2	8
4	54321	Anatomy	3	6	16
5	82754	Social Psychology	3	7	9
6	13524	Statistical Analysis	3	1	11
7	24653	Intro to Marketing	3	1	4
8	22679	Intro to Calculus	3	4	5
9	98765	Intermediate Accounting	3	1	1
10	96032	Social Media	3	1	4
11	58230	Physiology	3	6	16
12	81256	Intro to Management	3	1	7
13	64321	Secondary Education	3	2	8
14	32751	Business Writing	2	1	10
15	46972	Biology	4	6	3
16	15487	Music Theory	3	5	12
17	28177	Classic Literature	3	3	15
18	90908	Educational Theory	3	2	13
19	55783	Shakespeare	3	3	15
20	63284	Population and Demo...	3	7	9
21	74832	Creative Writing	3	3	2
22	33218	Marching Band	2	5	12

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The central pane displays the results of a query: 'SELECT * FROM Instructors'. The results grid contains 16 rows of instructor information, including InstructorID, LastName, FirstName, Status, DepartmentChairman, HireDate, AnnualSalary, and DepartmentID. The status bar at the bottom indicates the query was executed successfully on 'DESKTOP-5MGKLN0' at '08-11-2022 00:00:00'.

InstructorID	LastName	FirstName	Status	DepartmentChairman	HireDate	AnnualSalary	DepartmentID
1	Brown	Billy	F	1	2016-01-10	77500.00	1
2	Thomas	William	P	0	2016-03-30	38500.00	3
3	Amundsen	Rachel	F	1	2016-06-05	79000.00	6
4	Green	Gene	F	0	2016-08-02	75000.00	1
5	McGregor	NULL	F	1	2017-01-03	74000.00	4
6	Paxton	Arnold	P	0	2017-07-15	36000.00	5
7	Rogers	NULL	P	0	2017-10-22	38000.00	1
8	Smith	John	F	1	2018-02-05	73000.00	2
9	Connors	Daniel	F	1	2018-03-04	71500.00	7
10	Jones	Sally	F	1	2018-09-21	74000.00	3
11	Vilma	Jonathan	P	0	2018-11-18	35500.00	1
12	Thomas	Demick	P	0	2019-01-17	35500.00	5
13	Black	Bill	P	0	2019-04-20	34000.00	2
14	Warren	Angela	P	0	2019-07-14	33000.00	4
15	Drew	Daniel	F	0	2019-08-25	72000.00	3
16	Gallegos	Tomas	F	0	2020-03-23	64000.00	6

3. Open another Query Editor window and then enter and run this statement

```
SELECT COUNT(*) AS NumberOfInstructors FROM Instructors
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various tables like 'Instructors', 'Courses', and 'Students'. The main query window contains the following SQL statement:

```
SELECT COUNT(*) AS NumberOfInstructors FROM Instructors
```

The results pane shows a single row with the value 16 under the column 'NumberOfInstructors'.

NumberOfInstructors
16

At the bottom, a status bar indicates "Query executed successfully." and "1 rows".

4. Open the script named InstructorDetails.sql. Note that this script contains just one SQL statement. Then, run the statement.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the 'MyCollege' database. The main query window contains the following SQL statement:

```
SELECT LastName, FirstName, HireDate, AnnualSalary
FROM Instructors
ORDER BY AnnualSalary;
```

The results pane displays 16 rows of data from the 'Instructors' table, ordered by 'AnnualSalary'. The columns are 'LastName', 'FirstName', 'HireDate', and 'AnnualSalary'.

Lastname	Firstname	Hiredate	AnnualSalary
Warren	Angela	2019-07-14	33000.00
Black	Bill	2019-04-20	34000.00
Vilma	Jonathan	2018-11-18	35500.00
Thomas	Derrick	2019-01-17	35500.00
Paxton	Arnold	2017-07-15	36000.00
Rogers	NULL	2017-10-22	38000.00
Thomas	William	2016-03-30	38500.00
Gallegos	Tomas	2020-03-23	64000.00
Connors	Daniel	2018-03-04	71500.00
Drew	Daniel	2019-08-25	72000.00
Smith	John	2018-02-05	73000.00
Jones	Sally	2018-09-21	74000.00
McGregor	NULL	2017-01-03	74000.00
Green	Gene	2016-08-02	75000.00
Brown	Billy	2016-01-10	77500.00
Amundsen	Rachel	2016-06-05	79000.00

At the bottom, a status bar indicates "Query executed successfully." and "16 rows".

5. Open the script named InstructorSummary.sql. Note that this opens another Query Editor window.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The main pane displays the following SQL query:

```

SELECT COUNT(*) AS NumberOfInstructors,
       MAX(AnnualSalary) AS MaxSalary,
       AVG(AnnualSalary) AS AverageSalary
  FROM Instructors;
  
```

The results pane shows the output of the query:

	NumberOfInstructors	MaxSalary	AverageSalary
1	16	79000.00	56906.25

At the bottom, a message indicates: "Query executed successfully." The status bar at the bottom right shows the date and time: "08-11-2022 13:52".

6. Open the script named InstructorStatements.sql. Note that this script contains two SQL statements that end with semicolons.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The main pane displays the following SQL script:

```

SELECT LastName, FirstName, HireDate, AnnualSalary
  FROM Instructors
 ORDER BY AnnualSalary;

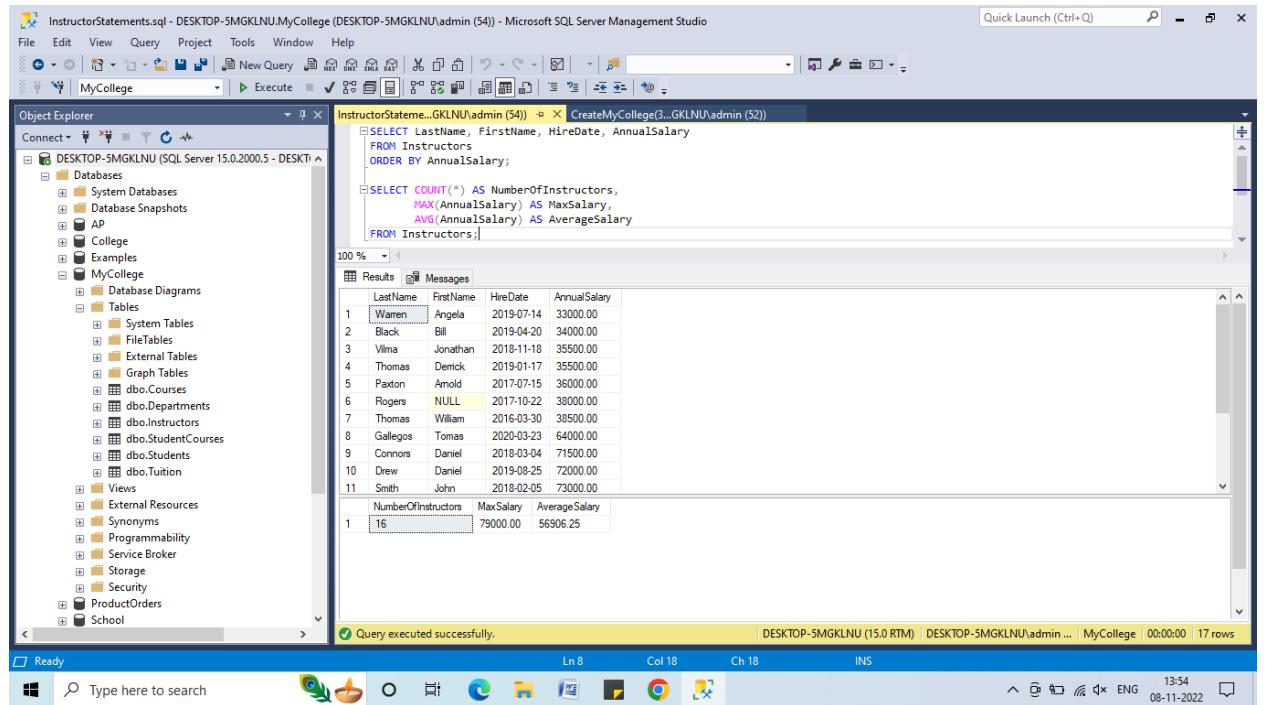
SELECT COUNT(*) AS NumberOfInstructors,
       MAX(AnnualSalary) AS MaxSalary,
       AVG(AnnualSalary) AS AverageSalary
  FROM Instructors;
  
```

The results pane shows the output of the second query from the script:

	NumberOfInstructors	MaxSalary	AverageSalary
1	16	79000.00	56906.25

At the bottom, a message indicates: "Query executed successfully." The status bar at the bottom right shows the date and time: "08-11-2022 13:57".

7. Press the F5 key or click the Execute button to run both statements in this script. Note that this displays the results in two Results tabs. Make sure to view the results of both SELECT statements.



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various objects like Tables, Views, and Procedures. A query window titled 'InstructorStatements...GKLNLU\admin (54)' contains two SELECT statements. The first statement retrieves data from the 'Instructors' table, and the second statement calculates summary statistics. Below the query window are two results tabs: 'Results' and 'Messages'. The 'Results' tab displays the data from the first query, and the 'Messages' tab shows the output of the second query, which includes the count of instructors, maximum salary, and average salary. At the bottom, a status bar indicates the query was executed successfully.

Last Name	First Name	Hire Date	Annual Salary
Warren	Angela	2019-07-14	33000.00
Black	Bill	2019-04-20	34000.00
Vilma	Jonathan	2018-11-18	35500.00
Thomas	Derrick	2019-01-17	35500.00
Paxton	Arnold	2017-07-15	36000.00
Rogers	NULL	2017-10-22	38000.00
Thomas	William	2016-03-30	38500.00
Gallegos	Tomas	2020-03-23	64000.00
Connors	Daniel	2018-03-04	71500.00
Drew	Daniel	2019-08-25	72000.00
Smith	John	2018-02-05	73000.00

Number of Instructors	Max Salary	Average Salary
16	79000.00	56906.25

8. Exit from SQL Server Management Studio.

Section B : Essential SQL Skills

1. Display all the columns from the Courses table

```
SELECT col.name FROM MyCollege.sys.tables tbl INNER JOIN MyCollege.sys.columns col  
ON tbl.object_id = col.object_id WHERE tbl.name= 'Courses'
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the MyCollege database which contains tables like Courses, Departments, and Students. The central pane displays the results of the executed query:

name
CourseID
CourseNumber
CourseDescription
CourseUnits
DepartmentID
InstructorID

Below the results, a message indicates "Query executed successfully." The status bar at the bottom right shows the date and time: 08-11-2022 15:01.

2. Display FullName as LastName + ',' + FirstName. Sort the result set by last name in ascending sequence.Return only the students whose last name begins with a letter from A to M.

```
SELECT LastName + ',' + FirstName as FullName FROM Students  
WHERE LastName LIKE '[A-M]%' ORDER BY LastName ASC;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various objects like Examples, Database Diagrams, Tables (including System Tables, FileTables, External Tables, Graph Tables, dbo.Courses, dbo.Departments, dbo.Instructors, dbo.StudentCourses, dbo.Students), Columns, Keys, Constraints, Triggers, Indexes, Statistics, Views, External Resources, Synonyms, and Programmability. The central pane displays a query results window for a SELECT statement:

```
SELECT LastName + ', ' + FirstName as FullName
FROM Students
WHERE LastName LIKE '[A-M]%' ORDER BY LastName ASC;
```

The results show 18 rows of data:

FullName
Biden, Jonas
Bonwell, Brian
Butler, George
Canden, James
Clement, Cal
Cramden, Walter
DeLorean, Cameron
Easton, Bamey
Flores, Jesus
Franks, Karen
Gardner, Faye
Geary, Annette
George, Mona
Goodell, Conner
Griffin, Gerald
Hallowell, Jimmy
Hoffman, Wilma
Howard, Amber

At the bottom of the results pane, it says "Query executed successfully." The status bar at the bottom right shows the date and time as 11-11-2022 12:35.

3. Display following column and Return only the rows with a hire date that's in 2019. Sort the result set in ascending sequence by the HireDate column

LastName The LastName column

FirstName The FirstName column

HireDate The HireDate column

```
SELECT LastName, FirstName, HireDate FROM Instructors WHERE YEAR(HireDate) = 2019
ORDER BY HireDate
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various objects like Examples, Database Diagrams, Tables (including System Tables, FileTables, External Tables, Graph Tables, dbo.Courses, dbo.Departments, dbo.Instructors, dbo.StudentCourses, dbo.Students), Columns, Keys, Constraints, Triggers, Indexes, Statistics, Views, External Resources, Synonyms, and Programmability. The central pane displays a query results window for a SELECT statement:

```
SELECT LastName, FirstName, HireDate FROM Instructors WHERE YEAR(HireDate) = 2019 ORDER BY HireDate;
```

The results show 4 rows of data:

LastName	FirstName	HireDate
Thomas	Derrick	2019-01-17
Black	Bill	2019-04-20
Warren	Angela	2019-07-14
Drew	Daniel	2019-08-25

At the bottom of the results pane, it says "Query executed successfully." The status bar at the bottom right shows the date and time as 08-11-2022 15:42.

4. Display below columns and Sort result set in ascending sequence by MonthsAttended column

FirstName	The FirstName column
LastName	The LastName column
EnrollmentDate	The EnrollmentDate column
CurrentDate	The current date
MonthsAttended	A column that's calculated by getting the difference between the enrollment date and the current date

```
SELECT FirstName, LastName, EnrollmentDate, SYSDATETIME() AS CurrentDate,
DATEDIFF(MONTH, EnrollmentDate, SYSDATETIME()) AS MonthsAttended FROM Students
ORDER BY MonthsAttended
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the 'MyCollege' database selected, showing tables like 'Students', 'Instructors', and 'Courses'. The right pane shows the results of the executed query:

```
SELECT FirstName, LastName, EnrollmentDate, SYSDATETIME() AS CurrentDate,
DATEDIFF(MONTH, EnrollmentDate, SYSDATETIME()) AS MonthsAttended FROM Students
ORDER BY MonthsAttended
```

The results grid contains 43 rows of data, with columns: FirstName, LastName, EnrollmentDate, CurrentDate, and MonthsAttended. The data includes names like Stanley, Silver, Biden, Walter, Cramden, Lisa, etc., along with their enrollment dates and calculated months attended.

	FirstName	LastName	EnrollmentDate	CurrentDate	MonthsAttended
1	Stanley	Silver	2020-01-04 11:16:19	2022-11-08 15:53:06 0247485	34
2	Jonas	Biden	2020-01-05 13:47:21	2022-11-08 15:53:06 0247485	34
3	Walter	Cramden	2019-12-15 10:18:37	2022-11-08 15:53:06 0247485	35
4	Lisa	Urie	2019-12-17 11:42:28	2022-11-08 15:53:06 0247485	35
5	Mona	George	2019-12-22 15:29:44	2022-11-08 15:53:06 0247485	35
6	Timothy	Johnson	2019-08-04 09:01:04	2022-11-08 15:53:06 0247485	39
7	Andrew	Walker	2019-08-05 13:48:26	2022-11-08 15:53:06 0247485	39
8	Faye	Gardner	2019-07-22 08:15:57	2022-11-08 15:53:06 0247485	40
9	Karen	Franks	2019-07-23 10:42:03	2022-11-08 15:53:06 0247485	40
10	Conner	Goodell	2019-01-02 14:21:58	2022-11-08 15:53:06 0247485	46
11	Gerald	Griffin	2019-01-02 16:04:04	2022-11-08 15:53:06 0247485	46
12	Lettia	Osborne	2018-12-12 17:14:22	2022-11-08 15:53:06 0247485	47
13	Vincent	Manning	2018-12-14 15:37:43	2022-11-08 15:53:06 0247485	47
14	Tanya	Sommers	2018-07-22 15:41:12	2022-11-08 15:53:06 0247485	52
15	Andrew	Jones	2018-07-24 10:53:26	2022-11-08 15:53:06 0247485	52
16	Floyd	Jackson	2018-07-25 09:27:53	2022-11-08 15:53:06 0247485	52
17	Annette	Geary	2018-07-12 09:33:47	2022-11-08 15:53:06 0247485	52
18	Jesus	Flores	2018-01-03 16:23:47	2022-11-08 15:53:06 0247485	58
19	James	Camden	2018-01-04 11:12:31	2022-11-08 15:53:06 0247485	58
20	Barney	Easton	2018-01-04 14:14:02	2022-11-08 15:53:06 0247485	58
21	Anthony	Rincon	2017-12-08 09:55:15	2022-11-08 15:53:06 0247485	59

Query executed successfully.

5. Display below columns and Return only the top 20 percent of instructors based on annual salary

FirstName	The FirstName column
LastName	The LastName column
AnnualSalary	The AnnualSalary column

```
SELECT TOP(20) PERCENT FirstName, LastName, AnnualSalary FROM Instructors ORDER BY
AnnualSalary DESC
```

```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
Object Explorer Connect
Proj1B.sql - DESKTOP-5MGKLN0\admin (51) CreateMyCollege3.sql - not connected
SELECT TOP(20) PERCENT FirstName, LastName, AnnualSalary FROM Instructors ORDER BY AnnualSalary DESC
Results Messages
100 %
1 Rachel Amundsen 79000.00
2 Billy Brown 77500.00
3 Gene Green 75000.00
4 NULL McGregor 74000.00
Query executed successfully.

```

6. Display below columns

FullTimeCost	The FullTimeCost column
PerUnitCost	The PerUnitCost column
Units	12
TotalPerUnitCost	A column that's calculated by multiplying the per unit cost by the units
TotalTuition	A column that's calculated by adding the full time cost to the total per unit cost

```

SELECT FullTimeCost, PerUnitCost, 12 AS Units, PerUnitCost*12 AS TotalPerUnitCost,
FullTimeCost+(PerUnitCost*12) AS TotalTuition FROM Tuition

```

```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
MyCollege | Execute | Quick Launch (Ctrl+Q) | X
Object Explorer
Connect ▾ MyCollege
    Examples
    Database Diagrams
    Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.Courses
        dbo.Departments
        dbo.Instructors
        dbo.StudentCourses
        dbo.Students
            Columns
                StudentID (PK int, not null)
                LastName (varchar(25), not null)
                FirstName (varchar(25), not null)
                EnrollmentDate (datetime2(0), not null)
                GraduationDate (date, null)
            Keys
            Constraints
            Triggers
            Indexes
            Statistics
        dbo.Tuition
        Views
        External Resources
        Synonyms
        Programmability
    Keys
    Constraints
    Triggers
    Indexes
    Statistics
    Views
    External Resources
    Synonyms
    Programmability
Results
SELECT FullTimeCost, PerUnitCost, 12 AS Units, PerUnitCost*12 AS TotalPerUnitCost,
       FullTimeCost*(PerUnitCost*12) AS TotalTuition FROM Tuition
100 %
1 1250.00 62.50 12 750.00 2000.00
Messages
Query executed successfully.
Ln 25 Col 1 Ch 1 INS
DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 | 1 rows
Item(s) Saved Type here to search 16:25 08-11-2022

```

7. Join Courses table to Departments table and return: CourseNumber, CourseDescription, DepartmentName. Sort result set by DepartmentName and CourseNumber in ascending order.

```

SELECT CourseNumber, CourseDescription, DepartmentName FROM Courses JOIN
Departments ON Courses.DepartmentID = Departments.DepartmentID ORDER BY
DepartmentName, CourseNumber

```

```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
MyCollege | Execute | Quick Launch (Ctrl+Q) | X
Object Explorer
Connect ▾ MyCollege
    Examples
    Database Diagrams
    Tables
        System Tables
        FileTables
        External Tables
        Graph Tables
        dbo.Courses
        dbo.Departments
        dbo.Instructors
        dbo.StudentCourses
        dbo.Students
            Columns
                StudentID (PK int, not null)
                LastName (varchar(25), not null)
                FirstName (varchar(25), not null)
                EnrollmentDate (datetime2(0), not null)
                GraduationDate (date, null)
            Keys
            Constraints
            Triggers
            Indexes
            Statistics
        dbo.Tuition
        Views
        External Resources
        Synonyms
        Programmability
    Keys
    Constraints
    Triggers
    Indexes
    Statistics
    Views
    External Resources
    Synonyms
    Programmability
Results
SELECT CourseNumber, CourseDescription, DepartmentName FROM Courses JOIN Departments
ON Courses.DepartmentID = Departments.DepartmentID ORDER BY DepartmentName, CourseNumber
100 %
1 13524 Statistical Analysis Business
2 24653 Intro to Marketing Business
3 32751 Business Writing Business
4 36598 Beginning Accounting Business
5 81256 Intro to Management Business
6 96032 Social Media Business
7 98765 Intermediate Accounting Business
8 14062 Primary Education Education
9 64321 Secondary Education Education
10 90908 Educational Theory Education
11 28177 Classic Literature English
12 37645 Composition English
13 55783 Shakespeare English
14 74832 Creative Writing English
15 22679 Intro to Calculus Mathematics
16 44386 Trigonometry Mathematics
17 48926 Abstract Algebra Mathematics
18 15487 Music Theory Music
19 33218 Marching Band Music
20 46972 Biology Science
21 54321 Anatomy Science
Messages
Query executed successfully.
Ln 32 Col 89 Ch 89 INS
DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 | 25 rows
Item(s) Saved Type here to search 16:41 08-11-2022

```

8. Join Instructors table to Courses table and returns: LastName, FirstName, CourseNumber, CourseDescription. Return all courses for each instructor with a status of "P" (part time). Sort the result set by LastName and FirstName in ascending order

```
SELECT LastName, FirstName, CourseNumber, CourseDescription FROM Instructors JOIN
Courses ON Instructors.InstructorID = Courses.InstructorID WHERE Status = 'P'
ORDER BY LastName, FirstName
```

Last Name	First Name	Course Number	Course Description
Black	Bill	90908	Educational Theory
Rogers	NULL	81256	Intro to Management
Thomas	Derick	15487	Music Theory
Thomas	Derick	33218	Marching Band
Thomas	William	74832	Creative Writing
Vilma	Jonathan	13524	Statistical Analysis
Warren	Angela	44386	Trigonometry

9. Use the UNION operator to generate a result set consisting below columns

Status A calculated column that contains a value of UNDERGRAD or GRADUATED

FirstName The FirstName column

LastName The LastName column

EnrollmentDate The EnrollmentDate column

GraduationDate The GraduationDate column

If the student doesn't have a value in the GraduationDate column, the Status column should contain a value of UNDERGRAD. Otherwise, it should contain a value of GRADUATED.

Sort the final result set by EnrollmentDate.

```
SELECT 'UNDERGRAD' AS Status, FirstName, LastName, EnrollmentDate, GraduationDate
FROM Students WHERE GraduationDate IS NULL
UNION
SELECT 'GRADUATED' AS Status, FirstName, LastName, EnrollmentDate, GraduationDate
FROM Students WHERE GraduationDate IS NOT NULL
ORDER BY EnrollmentDate
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure for 'MyCollege'. The central pane displays a query results grid titled 'Results' with the following data:

Status	FstName	LastName	EnrollmentDate	GraduationDate
GRADUATED	Amber	Howard	2015-12-18 16:44:26	2019-12-14
GRADUATED	George	White	2015-12-20 11:12:26	2019-12-14
GRADUATED	Tony	MacNamara	2015-12-21 09:21:55	2019-05-07
GRADUATED	Jonathan	Welch	2015-12-21 13:23:10	2019-12-14
UNDERGRAD	Donna	Taylor	2015-12-28 10:32:16	NULL
GRADUATED	Rose	Price	2016-01-02 12:37:31	2019-12-14
GRADUATED	Jesse	Rodriguez	2016-01-03 13:08:37	2019-05-07
UNDERGRAD	Bonnie	Williams	2016-01-03 15:44:56	NULL
UNDERGRAD	Thomas	Kent	2016-07-15 11:14:23	NULL
UNDERGRAD	Maggie	Kramer	2016-07-15 17:02:45	NULL
UNDERGRAD	Cameron	DeLocean	2016-07-18 12:48:43	NULL
UNDERGRAD	Frank	Sanchez	2016-07-20 09:37:53	NULL
GRADUATED	Roberta	Smith	2016-07-22 11:18:25	2019-12-14
UNDERGRAD	Wilma	Hoffman	2016-12-10 15:31:28	NULL
UNDERGRAD	Brian	Bonwell	2016-12-12 14:22:53	NULL
UNDERGRAD	Cal	Clement	2016-12-14 16:42:11	NULL
UNDERGRAD	Charles	Patrick	2016-12-22 08:43:48	NULL
UNDERGRAD	William	Landy	2017-01-02 11:28:49	NULL
UNDERGRAD	Monica	Momsey	2017-01-04 10:42:06	NULL
UNDERGRAD	George	Butler	2017-07-12 13:05:41	NULL

Below the grid, a message indicates: 'Query executed successfully.' The status bar at the bottom shows: DESKTOP-5MGKLN| (15.0 RTM) | DESKTOP-5MGKLN\administrator | MyCollege | 00:00:00 | 43 rows.

10. Display below columns:

InstructorDept The DepartmentName column from the Departments table for a related instructor

LastName The LastName column from the Instructors table

FirstName The FirstName column from the Instructors table

CourseDescription The CourseDescription column from the Courses table

CourseDept The DepartmentName column from the Departments table for a related instructor

Return one row for each course that's in a different department than the department of the instructor assigned to teach that course

```

SELECT Subquery_2.DepartmentName AS InstructorDept, Subquery_2.FirstName,
Subquery_2.LastName, Subquery_1.CourseDescription,
Subquery_1.DepartmentName AS CourseDept FROM (
SELECT c.CourseDescription,d.DepartmentName,c.DepartmentID, c.InstructorID
FROM Courses c JOIN Departments d
ON d.DepartmentID = c.DepartmentID) AS Subquery_1
JOIN (
SELECT i.DepartmentID, d.DepartmentName, i.InstructorID,i.FirstName, i.LastName
FROM Instructors i JOIN Departments d
ON i.DepartmentID = d.DepartmentID) AS Subquery_2
ON Subquery_1.DepartmentID!= Subquery_2.DepartmentID and Subquery_1.InstructorID =
Subquery_2.InstructorID
GO
    
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'MyCollege' is selected. In the center pane, a query window titled 'Proj1B.sql - DESKTOP-5MGKLN0\MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio' displays the following T-SQL code:

```

SELECT Subquery_2.DepartmentName AS InstructorDept, Subquery_2.FirstName, Subquery_2.LastName, Subquery_1.CourseDescription,
       Subquery_1.DepartmentName AS CourseDept
  FROM (
    SELECT c.CourseDescription, d.DepartmentName, c.DepartmentID, c.InstructorID
      FROM Courses c
      JOIN Departments d
        ON d.DepartmentID = c.DepartmentID) AS Subquery_1
 JOIN (
    SELECT i.DepartmentID, d.DepartmentName, i.InstructorID, i.FirstName, i.LastName
      FROM Instructors i
      JOIN Departments d
        ON i.DepartmentID = d.DepartmentID) AS Subquery_2
   ON Subquery_1.DepartmentID = Subquery_2.DepartmentID AND Subquery_1.InstructorID = Subquery_2.InstructorID
  GO

```

The results grid shows one row:

InstructorDept	FirstName	LastName	CourseDescription	CourseDept
English	Sally	Jones	Business Writing	Business

Below the results, a message states "Query executed successfully."

11. Display one row for each instructor that has courses with these columns:

The instructor first and last names from the Instructors table in this format: John Doe (Note: If the instructor first name has a null value, the concatenation of the first and last name will result in a null value.)

A count of the number of courses in the Courses table

The sum of the course units in the Courses table

Sort the result set in descending sequence by the total course units for each instructor

```

SELECT FirstName + ' ' + LastName AS InstructorName, COUNT(*) AS NumOfCourses,
       SUM(CourseUnits) AS CourseUnitsSum FROM Instructors
       LEFT JOIN Courses ON Instructors.InstructorID = Courses.InstructorID
       GROUP BY FirstName + ' ' + LastName ORDER BY SUM(CourseUnits) DESC

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'MyCollege' is selected. In the center pane, a query window titled 'CreateMyCollege(3).sql - not connected' contains the following SQL code:

```
SELECT FirstName+' '+LastName AS InstructorName, COUNT(*) AS NumOfCourses, SUM(CourseUnits) AS CourseUnitsSum
FROM Instructors
LEFT JOIN Courses ON Instructors.InstructorID = Courses.InstructorID
GROUP BY FirstName+' '+LastName ORDER BY SUM(CourseUnits) DESC
```

The results grid displays the following data:

InstructorName	NumOfCourses	CourseUnitsSum
NULL	3	9
Rachel Amundsen	2	8
Tomas Gallegos	2	6
Gene Green	2	6
John Smith	2	6
Billy Brown	2	6
Daniel Connor	2	6
Daniel Drew	2	6
Demick Thomas	2	5
Sally Jones	2	5
William Thomas	1	3

At the bottom of the results grid, a message says 'Query executed successfully.'

12. Display the total number of courses taught by part time instructors, instructor last name and first name from the Instructors table in this format: Doe, John (Note: If the instructor first name has a null value, the concatenation of the first and last name will result in a null value.)
The total number of courses taught for each instructor in the Courses table
Use the ROLLUP operator to include a row that gives the grand total.

```
SELECT LastName+', '+FirstName AS InstructorName, COUNT(*) AS NumOfCourses FROM
Instructors
LEFT JOIN Courses ON Instructors.InstructorID = Courses.InstructorID
GROUP BY LastName+', '+FirstName WITH ROLLUP
```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

MyCollege

Tables

Results

```
SELECT LastName+', '+FirstName AS InstructorName, COUNT(*) AS NumOfCourses FROM Instructors
LEFT JOIN Courses ON Instructors.InstructorID = Courses.InstructorID
GROUP BY LastName+', '+FirstName WITH ROLLUP
```

InstructorName	NumOfCourses
Gallegos, Tomas	2
Green, Gene	2
Jones, Sally	2
Paxton, Arnold	1
Smith, John	2
Thomas, Demick	2
Thomas, William	1
Vilma, Jonathan	1
Warren, Angela	1
NULL	26

Query executed successfully.

13. use a subquery in a WHERE clause that uses the IN keyword.

```
SELECT DISTINCT LastName, FirstName
FROM Instructors i
WHERE i.InstructorID IN (SELECT DISTINCT InstructorID FROM Courses)
ORDER BY LastName, FirstName;
```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

MyCollege

Tables

Results

```
SELECT DISTINCT LastName, FirstName
FROM Instructors i
WHERE i.InstructorID IN (SELECT DISTINCT InstructorID FROM Courses)
ORDER BY LastName, FirstName;
```

LastName	FirstName
Amundsen	Rachel
Black	Bill
Brown	Billy
Connors	Daniel
Drew	Daniel
Gallegos	Tomas
Green	Gene
Jones	Sally
McGregor	NULL
Rogers	NULL
Smith	John
Thomas	Demick
Thomas	William
Vilma	Jonathan
Warren	Angela

Query executed successfully.

14. Display one row for each course with these columns :CourseID and enrollment date

```
SELECT c.CourseID, MAX(EnrollmentDate) AS MaxEnrollmentDate FROM Courses AS c
LEFT JOIN StudentCourses AS sc ON c.CourseID = sc.CourseID
LEFT JOIN Students AS s ON sc.StudentID = s.StudentID
GROUP BY c.CourseID
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database schema, including tables like Courses, StudentCourses, and Students. The central pane contains the query:

```
SELECT c.CourseID, MAX(EnrollmentDate) AS MaxEnrollmentDate FROM Courses AS c
LEFT JOIN StudentCourses AS sc ON c.CourseID = sc.CourseID
LEFT JOIN Students AS s ON sc.StudentID = s.StudentID
GROUP BY c.CourseID
```

The Results pane shows the output of the query, which is a table with two columns: CourseID and MaxEnrollmentDate. The data is as follows:

CourseID	MaxEnrollmentDate
3	2019-08-05 13:48:26
4	2020-01-05 13:47:21
5	2019-08-05 13:48:26
6	2019-12-15 10:18:37
7	2019-08-04 09:01:04
8	2018-12-14 15:37:43
9	2019-12-15 10:18:37
10	2018-07-12 09:33:47
11	2020-01-04 11:16:19
12	2020-01-05 13:47:21
13	2019-12-22 15:29:44
14	2019-12-15 10:18:37
15	2016-12-12 14:22:53
16	2019-12-17 11:42:28
17	2020-01-05 13:47:21
18	NULL
19	2019-01-02 14:21:58
20	2020-01-04 11:16:19

At the bottom of the Results pane, a message indicates "Query executed successfully". The status bar at the bottom right shows the system information: DESKTOP-5MGKLN (15.0 RTM), DESKTOP-5MGKLN\administrator, MyCollege, 00:00:00, 25 rows, 12:55, 11-11-2022.

Use CTE and display one row per course that shows the CourseDescription for the course and the LastName, FirstName, and EnrollmentDate for the student with the most recent enrollment data.

```
WITH CourseSummary AS
(
SELECT c.CourseID, MAX(EnrollmentDate) AS MaxEnrollmentDate
FROM Courses AS c
LEFT JOIN StudentCourses AS sc ON c.CourseID = sc.CourseID
LEFT JOIN Students AS s ON sc.StudentID = s.StudentID
GROUP BY c.CourseID
)
SELECT CourseDescription, LastName, FirstName, EnrollmentDate
FROM Courses AS c
JOIN StudentCourses AS sc ON c.CourseID = sc.CourseID
JOIN Students AS s ON sc.StudentID = s.StudentID
JOIN CourseSummary AS cs ON c.CourseID = cs.CourseID
AND s.EnrollmentDate = cs.MaxEnrollmentDate
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'MyCollege' database is selected. In the center pane, a query window titled 'Proj1B.sql - DESKTOP-5MGKLN0\MyCollege (DESKTOP-5MGKLN0\admin (51))' displays a complex query involving multiple joins and a WITH clause. Below the query, the results grid shows 12 rows of course information. The status bar at the bottom indicates 'Query executed successfully.'

CourseDescription	LastName	FirstName	EnrollmentDate
Biology	Bonwell	Brian	2016-12-12 14:22:53
Microbiology	Camden	James	2018-04-11 11:23:31
Social Media	Geary	Annette	2018-07-12 09:33:47
Intro to Calculus	Manning	Vincent	2018-12-14 15:37:43
Shakespeare	Goodell	Conner	2019-01-02 14:21:58
Beginning Accounting	Johnson	Timothy	2019-08-04 09:01:04
Intro to Marketing	Johnson	Timothy	2019-08-04 09:01:04
Composition	Johnson	Timothy	2019-08-04 09:01:04
Primary Education	Walker	Andrew	2019-08-05 13:48:26
Social Psychology	Walker	Andrew	2019-08-05 13:48:26
Statistical Analysis	Cramden	Walter	2019-12-15 10:18:37
Intermediate Accou...	Cramden	Walter	2019-12-15 10:18:37

15. Add below row to the Departments table:

DepartmentName: History

```
INSERT INTO Departments values ('History')
SELECT * FROM Departments
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'MyCollege' database is selected. In the center pane, a query window titled 'Proj1B.sql - DESKTOP-5MGKLN0\MyCollege (DESKTOP-5MGKLN0\admin (59))' contains the provided SQL code. The results grid shows the message '(1 row affected)' and the completion time. The status bar at the bottom indicates 'Query executed successfully.'

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

MyCollege

Tables

DepartmentID DepartmentName

1	Business
2	Education
3	English
4	History
5	Mathematics
6	Music
7	Political Science
8	Science
9	Sociology

Results Messages

Query executed successfully.

DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 9 rows

Item(s) Saved Type here to search Ln 95 Col 1 Ch 1 INS 19:51 08-11-2022

16. Write a single INSERT statement that adds these rows to the Instructors table:

```
INSERT INTO Instructors VALUES
('Benedict', 'Susan', 'P', 0, GETDATE(), 34000.00,9),
('Adams', NULL, 'F', 1, GETDATE(), 66000.00,9);
SELECT * FROM Instructors
```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (51)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

MyCollege

Tables

InstructionID LastName FirstName Status DepartmentChairman HireDate AnnualSalary DepartmentID

1	Brown	Billy	F	1	2016-01-10	77500.00	1
2	Thomas	William	P	0	2016-03-30	38500.00	3
3	Amundsen	Rachel	F	1	2016-06-05	79000.00	6
4	Green	Gene	F	0	2016-08-02	75000.00	1
5	McGregor	NULL	F	1	2017-01-03	74000.00	4
6	Paxton	Arnold	P	0	2017-07-15	36000.00	5
7	Rogers	NULL	P	0	2017-10-22	38000.00	1
8	Smith	John	F	1	2018-02-05	73000.00	2
9	Connors	Daniel	F	1	2018-03-04	71500.00	7
10	Jones	Sally	F	1	2018-09-21	74000.00	3
11	Vilma	Jonathan	P	0	2018-11-18	35500.00	1
12	Thomas	Demick	P	0	2019-01-17	35500.00	5
13	Black	Bill	P	0	2019-04-20	34000.00	2
14	Warren	Angela	P	0	2019-07-14	33000.00	4
15	Drew	Daniel	F	0	2019-08-25	72000.00	3
16	Gallegos	Tomas	F	0	2020-03-23	64000.00	6
17	Benedict	Susan	P	0	2022-11-11	34000.00	9
18	Adams	NULL	F	1	2022-11-11	66000.00	9

Results Messages

Query executed successfully.

DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 18 rows

Item(s) Saved Type here to search Ln 114 Col 1 Ch 1 INS 13:15 11-11-2022

17. UPDATE AnnualSalary column from 34,000 to 35,0000 using InstructorID

```
UPDATE Instructors SET AnnualSalary = 35000.00 WHERE InstructorID = 17
SELECT * FROM Instructors WHERE InstructorID = 17
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The 'Tables' node is expanded, showing 'Instructors' and other tables. The 'Instructors' table node is also expanded, showing columns like StudentID, LastName, FirstName, Status, DepartmentChairman, HireDate, AnnualSalary, and DepartmentID. The 'Script' button next to the 'Instructors' table is highlighted. The 'Script' pane at the bottom contains the SQL code for updating the AnnualSalary column. The 'Results' pane shows the output of the SELECT query, which returns one row for InstructorID 17, Benedict, Susan, P, 0, 2022-11-08, 35000.00, and 9. A status bar at the bottom indicates 'Query executed successfully.'

18. Write a DELETE row using InstructorID

```
DELETE FROM Instructors WHERE InstructorID = 18
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The 'Tables' node is expanded, showing 'Instructors' and other tables. The 'Instructors' table node is also expanded, showing columns like StudentID, LastName, FirstName, Status, DepartmentChairman, HireDate, AnnualSalary, and DepartmentID. The 'Script' button next to the 'Instructors' table is highlighted. The 'Script' pane at the bottom contains the SQL code for deleting the row where InstructorID is 18. The 'Messages' pane shows the output '(1 row affected)'. A status bar at the bottom indicates 'Completion time: 2022-11-08T20:17:03.5697988-05:00'.

```
SELECT * FROM Instructors WHERE InstructorID = 18
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various tables like 'Instructors', 'Students', and 'Courses'. The central pane displays a query window with the following SQL code:

```
DELETE FROM Instructors WHERE InstructorID = 18
SELECT * FROM Instructors WHERE InstructorID = 18
```

The results pane below shows a table with columns: InstructorID, LastName, FirstName, Status, DepartmentChairman, HireDate, AnnualSalary, and DepartmentID. A message at the bottom of the results pane says "Query executed successfully." The status bar at the bottom right indicates the query was run at 08-11-2022 20:17.

19. DELETE the row in the Departments table that has an ID of 9.

```
DELETE FROM Instructors WHERE DepartmentID = 9
DELETE FROM Departments WHERE DepartmentID = 9
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' with various tables like 'Instructors', 'Students', and 'Courses'. The central pane displays a query window with the following SQL code:

```
DELETE FROM Departments WHERE DepartmentID = 9
```

The results pane below shows a message in red: "Msg 547, Level 16, State 0, Line 118 The DELETE statement conflicted with the REFERENCE constraint "FK_Instructors_Depart_30F848ED". The conflict occurred in database "MyCollege". The statement has been terminated." The status bar at the bottom right indicates the query was run at 08-11-2022 20:33.

```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (59)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q) 
MyCollege
Object Explorer
Connect 
Proj1B.sql - DESKTOP-5MGKLN0.admin (59) CreateMyCollege(3).sql - not connected
DELETE FROM Instructors WHERE DepartmentID = 9
DELETE FROM Departments WHERE DepartmentID = 9

Messages
(1 row affected)
(1 row affected)
Completion time: 2022-11-08T20:39:47.5498986-05:00

100 % 
100 % 
Item(s) Saved 
Ln 118 Col 47 Ch 47 INS 
DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 0 rows 
20:33 08-11-2022 
Type here to search 

```

20. UPDATE annual salary for all instructors in the Education department by 5%.

```

UPDATE Instructors SET AnnualSalary =( AnnualSalary + (0.05* AnnualSalary)) FROM
Instructors
JOIN Departments ON Instructors.DepartmentID = Departments.DepartmentID WHERE
Departments.DepartmentName='Education'

```

```

Proj1B.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (59)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q) 
MyCollege
Object Explorer
Connect 
Proj1B.sql - DESKTOP-5MGKLN0.admin (59) CreateMyCollege(3).sql - not connected
UPDATE Instructors SET AnnualSalary =( AnnualSalary + (0.05* AnnualSalary)) FROM Instructors
JOIN Departments ON Instructors.DepartmentID = Departments.DepartmentID WHERE Departments.DepartmentName='Education'

Messages
(2 rows affected)

Completion time: 2022-11-08T20:45:21.2134906-05:00

100 % 
100 % 
Item(s) Saved 
Ln 127 Col 1 Ch 1 INS 
DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 0 rows 
20:45 08-11-2022 
Type here to search 

```

21. Delete instructors that aren't teaching any courses.

```
DELETE FROM Instructors WHERE InstructorID NOT IN(SELECT DISTINCT InstructorID  
FROM Courses);
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The central pane displays the query: `DELETE FROM Instructors WHERE InstructorID NOT IN(SELECT DISTINCT InstructorID FROM Courses);`. The status bar at the bottom indicates 'Query executed successfully.'

22. Open the script named CreateGradStudents.sql that's attached above. Run this file to create a table named GradStudents.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The central pane displays the query: `CREATE TABLE GradStudents
(StudentID INT PRIMARY KEY,
LastName VARCHAR(25) NOT NULL,
FirstName VARCHAR(25) NOT NULL,
EnrollmentDate DATE NOT NULL,
GraduationDate Date NULL);`. The status bar at the bottom indicates 'Query executed successfully.'

23. Insert rows from Students table GradStudents table Include only rows of graduated students

```
INSERT INTO GradStudents SELECT * FROM Students WHERE GraduationDate IS NOT NULL  
SELECT * FROM GradStudents
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-SMGKLN0 (SQL Server 15.0.2000.5 - DESKTOP-SMGKLN0)' under the 'MyCollege' database. The main window displays a query editor with the following SQL code:

```
INSERT INTO GradStudents SELECT * FROM Students WHERE GraduationDate IS NOT NULL  
SELECT * FROM GradStudents
```

Below the code, the 'Messages' pane shows the output:

(7 rows affected)
Completion time: 2022-11-08T21:06:56.6016057-05:00

The status bar at the bottom indicates 'Query executed successfully.' and provides system information: DESKTOP-SMGKLN0 (15.0 RTM) | DESKTOP-SMGKLN0\admin ... | MyCollege | 00:00:00 | 0 rows.

The screenshot shows the Microsoft SQL Server Management Studio interface, similar to the previous one. The Object Explorer on the left shows the same connection details. The main window now displays the results of the query in the 'Results' pane. The table structure is as follows:

StudentID	LastName	FirstName	EnrollmentDate	GraduationDate
1	Howard	Amber	2015-12-18	2019-12-14
2	White	George	2015-12-20	2019-12-14
3	MacNamara	Tony	2015-12-21	2019-05-07
4	Welch	Jonathan	2015-12-21	2019-12-14
5	Price	Rose	2016-01-02	2019-12-14
6	Rodriguez	Jesse	2016-01-03	2019-05-07
7	Smith	Roberta	2016-07-22	2019-12-14

The status bar at the bottom indicates 'Query executed successfully.' and provides system information: DESKTOP-SMGKLN0 (15.0 RTM) | DESKTOP-SMGKLN0\admin ... | MyCollege | 00:00:00 | 7 rows.

24. Restore database

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'MyCollege' is selected under 'DESKTOP-5MGKLN0\GKLNU'. In the center pane, a query window titled 'CreateMyCollege(3).sql - DESKTOP-5MGKLN0\GKLNU.MyCollege (DESKTOP-5MGKLN0\admin (62)) - Microsoft SQL Server Management Studio' contains the following T-SQL script:

```
USE master
GO

IF DB_ID('MyCollege') IS NOT NULL
    DROP DATABASE MyCollege
CREATE DATABASE MyCollege
GO

USE [MyCollege]
GO
/****** Object: Table [dbo].[Courses] Script Date: 10/12/2022 10:15:00 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

The 'Messages' pane at the bottom displays several error messages in red:

```
Msg 3702, Level 16, State 4, Line 5
Cannot drop database "MyCollege" because it is currently in use.

Msg 1801, Level 16, State 3, Line 7
Database 'MyCollege' already exists. Choose a different database name.

Msg 2714, Level 16, State 6, Line 18
There is already an object named 'Courses' in the database.

Msg 2714, Level 16, State 6, Line 36
There is already an object named 'Departments' in the database.

Msg 2714, Level 16, State 6, Line 50
There is already an object named 'Instructors' in the database.

Msg 2714, Level 16, State 6, Line 70
There is already an object named 'StudentCourses' in the database.

Msg 2714, Level 16, State 6, Line 80
There is already an object named 'Students' in the database.
```

A status bar at the bottom right indicates 'Query completed with errors.'

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'MyCollege' is selected under 'DESKTOP-5MGKLN0\GKLNU'. In the center pane, a query window titled 'CreateMyCollege(3).sql - DESKTOP-5MGKLN0\GKLNU.MyCollege (DESKTOP-5MGKLN0\admin (52)) - Microsoft SQL Server Management Studio' contains the same T-SQL script as the previous screenshot.

```
USE master
GO

IF DB_ID('MyCollege') IS NOT NULL
    DROP DATABASE MyCollege
CREATE DATABASE MyCollege
GO

USE [MyCollege]
GO
/****** Object: Table [dbo].[Courses] Script Date: 10/12/2022 10:15:00 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

The 'Messages' pane at the bottom displays success messages in green:

```
(1 row affected)
```

A status bar at the bottom right indicates 'Query executed successfully.'

25. Display columns using CONVERT functions

```
SELECT CONVERT(varchar, EnrollmentDate,101) AS Date1, CONVERT(varchar,  
EnrollmentDate,22) AS Date2,  
CONVERT(varchar, EnrollmentDate,114) AS Date3, CONVERT(varchar(5),  
EnrollmentDate,110) AS Date4 FROM Students
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-5MGKLN0 (SQL Server 15.0.2000.5 - DESKTOP-5MGKLN0)' with various database objects like 'Security', 'Replication', and 'PolyBase'. The main window displays a query results grid for the following SQL statement:

```
SELECT CONVERT(varchar, EnrollmentDate,101) AS Date1, CONVERT(varchar, EnrollmentDate,22) AS Date2,  
CONVERT(varchar, EnrollmentDate,114) AS Date3, CONVERT(varchar(5), EnrollmentDate,110) AS Date4 FROM Students
```

The results grid has four columns: Date1, Date2, Date3, and Date4. The Date1 column shows dates in 'MM/DD/YY' format, Date2 in 'MM/DD/YY HH:MM:SS AM/PM' format, Date3 in 'MM/DD/YY HH:MM:SS' format, and Date4 in 'MM/DD/YY HH:MM:SS AM' format. There are 43 rows of data, each representing a student's enrollment date converted into different formats.

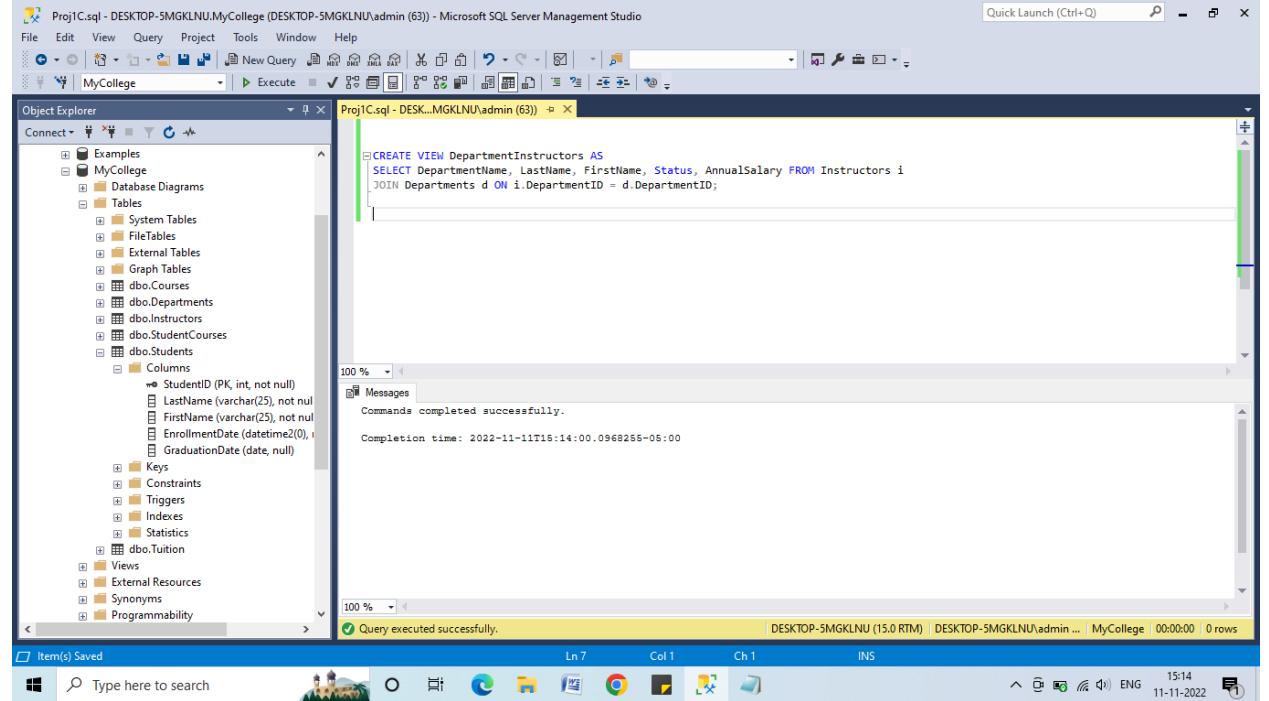
	Date1	Date2	Date3	Date4
1	12/18/2015	12/18/15 4:44:26 PM	16:44:26	12-18
2	12/20/2015	12/20/15 11:12:26 AM	11:12:26	12-20
3	12/21/2015	12/21/15 9:21:55 AM	09:21:55	12-21
4	12/21/2015	12/21/15 1:23:10 PM	13:23:10	12-21
5	12/28/2015	12/28/15 10:32:16 AM	10:32:16	12-28
6	01/02/2016	01/02/16 12:37:31 PM	12:37:31	01-02
7	01/03/2016	01/03/16 1:08:37 PM	13:08:37	01-03
8	01/03/2016	01/03/16 3:44:56 PM	15:44:56	01-03
9	07/15/2016	07/15/16 11:14:23 AM	11:14:23	07-15
10	07/15/2016	07/15/16 5:02:45 PM	17:02:45	07-15
11	07/18/2016	07/18/16 12:48:43 PM	12:48:43	07-18
12	07/20/2016	07/20/16 9:37:53 AM	09:37:53	07-20
13	07/22/2016	07/22/16 11:18:25 AM	11:18:25	07-22
14	12/10/2016	12/10/16 3:31:28 PM	15:31:28	12-10
15	12/12/2016	12/12/16 2:22:53 PM	14:22:53	12-12
16	12/14/2016	12/14/16 4:42:11 PM	16:42:11	12-14
17	12/22/2016	12/22/16 8:43:48 AM	08:43:48	12-22
18	01/02/2017	01/02/17 11:28:49 AM	11:28:49	01-02
19	01/04/2017	01/04/17 10:42:06 AM	10:42:06	01-04
20	07/12/2017	07/12/17 1:05:41 PM	13:05:41	07-12

At the bottom of the results grid, a message says 'Query executed successfully.' The status bar at the bottom right shows 'DESKTOP-5MGKLN0 (15.0 RTM) | DESKTOP-5MGKLN0\admin... | MyCollege | 00:00:00 | 43 rows'. The taskbar at the very bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and system status.

Section C : Advanced SQL Skills (views/stores procedures/ functions / scripts)

1. Create a view named DepartmentInstructors that returns the DepartmentName ,LastName, FirstName, Status, and AnnualSalary columns

```
CREATE VIEW DepartmentInstructors AS
SELECT DepartmentName, LastName, FirstName, Status, AnnualSalary FROM
Instructors i JOIN Departments d ON i.DepartmentID = d.DepartmentID;
```



2. Display DepartmentInstructors view

```
SELECT * FROM DepartmentInstructors;
```

Proj1C.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (63)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

MyCollege Execute

Object Explorer

Connect ▾

DESKTOP-5MGKLN0 (SQL Server 15.0.2000.5 - DESKTOP-5MGKLN0) Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalogs SQL Server Agent (Agent XPs disabled) XEvent Profiler

Proj1C.sql - DESKTOP-5MGKLN0\admin (63)

```
/*
-----C1-----
CREATE VIEW DepartmentInstructors AS
SELECT DepartmentName, LastName, FirstName, Status, AnnualSalary FROM Instructors i
JOIN Departments d ON i.DepartmentID = d.DepartmentID;

/*
-----C2-----
SELECT * FROM DepartmentInstructors;
```

Results Messages

	DepartmentName	LastName	FirstName	Status	AnnualSalary
1	Business	Brown	Billy	F	77500.00
2	English	Thomas	William	P	38500.00
3	Science	Amundsen	Rachel	F	79000.00
4	Business	Green	Gene	F	75000.00
5	Mathematics	McGregor	NULL	F	74000.00
6	Business	Rogers	NULL	P	38000.00
7	Education	Smith	John	F	73000.00
8	Sociology	Connors	Daniel	F	71500.00
9	English	Jones	Sally	F	74000.00
10	Business	Vilma	Jonathan	P	35500.00
11	Music	Thomas	Derick	P	35500.00
12	Education	Black	Bill	P	34000.00
13	Mathematics	Warren	Angela	P	33000.00
14	English	Drew	Daniel	F	72000.00
15	Science	Gallegos	Tomas	F	64000.00

Query executed successfully.

DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 | 15 rows

Ready Type here to search

Ln 9 Col 38 Ch 38 INS

15:17 11-11-2022 ENG

3. Return one row for each fulltime instructor in the English department

```
SELECT DISTINCT * FROM DepartmentInstructors
WHERE Status = 'F' and DepartmentName = 'English';
```

Proj1C.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (63)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

MyCollege Execute

Object Explorer

Connect ▾

DESKTOP-5MGKLN0 (SQL Server 15.0.2000.5 - DESKTOP-5MGKLN0) Databases Security Server Objects Replication PolyBase Always On High Availability Management Integration Services Catalogs SQL Server Agent (Agent XPs disabled) XEvent Profiler

Proj1C.sql - DESKTOP-5MGKLN0\admin (63)

```
SELECT DISTINCT * FROM DepartmentInstructors
WHERE Status = 'F' and DepartmentName = 'English';
```

Results Messages

	DepartmentName	LastName	FirstName	Status	AnnualSalary
1	English	Drew	Daniel	F	72000.00
2	English	Jones	Sally	F	74000.00

Query executed successfully.

DESKTOP-5MGKLN0 (15.0 RTM) DESKTOP-5MGKLN0\admin ... MyCollege 00:00:00 | 2 rows

Item(s) Saved Type here to search

Ln 15 Col 1 Ch 1 INS

15:19 11-11-2022 ENG

4. Update the annual salary for each fulltime instructor in the English department by 10%

```
UPDATE DepartmentInstructors
SET AnnualSalary = AnnualSalary + (AnnualSalary * 0.1)
WHERE Status = 'F' and DepartmentName = 'English';
SELECT * FROM DepartmentInstructors;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure for 'DESKTOP-SMGKLN1 (SQL Server 15.0.2000.5 - DESKTOP-SMGKLN1\MyCollege)'. In the center, the 'Results' tab of a query window titled 'Proj1C.sql - DESKTOP-SMGKLN1\MyCollege (admin (63))' shows the output of the provided SQL code. The results table has columns: DepartmentName, LastName, FirstName, Status, and AnnualSalary. The data shows 15 rows, all of which have been updated with a 10% salary increase.

	DepartmentName	LastName	FirstName	Status	AnnualSalary
1	Business	Brown	Billy	F	77500.00
2	English	Thomas	William	P	38500.00
3	Science	Amundsen	Rachel	F	79000.00
4	Business	Green	Gene	F	75000.00
5	Mathematics	McGregor	NULL	F	74000.00
6	Business	Rogers	NULL	P	38000.00
7	Education	Smith	John	F	73000.00
8	Sociology	Connors	Daniel	F	71500.00
9	English	Jones	Sally	F	81400.00
10	Business	Vilma	Jonathan	P	35500.00
11	Music	Thomas	Demick	P	35500.00
12	Education	Black	Bill	P	34000.00
13	Mathematics	Warren	Angela	P	33000.00
14	English	Drew	Daniel	F	79200.00
15	Science	Gallegos	Tomas	F	64000.00

At the bottom of the results window, a message states 'Query executed successfully.' The status bar at the bottom right shows the date and time: '11-11-2022 15:22'.

5. Display FirstName and LastName ,CourseNumber, CourseDescription, and CourseUnits

```
CREATE VIEW StudentCoursesMin AS
SELECT s.FirstName, s.LastName, c.CourseNumber, c.CourseDescription,
c.CourseUnits
FROM Students s JOIN StudentCourses sc ON s.StudentID = sc.StudentID
```

```

JOIN Courses c ON sc.CourseID = c.CourseID;

```

```

Proj1C.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (63)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
MyCollege Execute
Object Explorer
Proj1C.sql - DESKTOP-5MGKLN0.MyCollege (DESKTOP-5MGKLN0\admin (63))
CREATE VIEW StudentCoursesMin AS
SELECT s.FirstName, s.LastName, c.CourseNumber, c.CourseDescription, c.CourseUnits
FROM Students s JOIN StudentCourses sc ON s.StudentID = sc.StudentID
JOIN Courses c ON sc.CourseID = c.CourseID;

Messages
Commands completed successfully.

Completion time: 2022-11-11T15:29:07.6318960-05:00

Item(s) Saved
Ln 31 Col 1 Ch 1 INS
Type here to search
DESKTOP-5MGKLN0 (15.0 RTM) | DESKTOP-5MGKLN0\admin ... | MyCollege | 00:00:00 | 0 rows
15:29 11-11-2022

```

6. Display number of grad students using scripts

```

DECLARE @countUGrad int;
SELECT @countUGrad = COUNT(StudentID) FROM Students WHERE GraduationDate IS
NULL;
IF @countUGrad >= 100
    PRINT 'The number of undergrad students is greater than or equal to 100';
ELSE

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query editor window titled "Proj1C.sql - DESKTOP-5MGKLNUI.MyCollege (DESKTOP-5MGKLNUI\MyCollege (63))". The code in the editor is:

```

PRINT 'The number of undergrad students is less than 100';

```

Below the code, the "Messages" pane displays the output of the execution:

```

The number of undergrad students is less than 100
Completion time: 2022-11-11T15:41:10.5458883-05:00

```

The status bar at the bottom of the screen shows "Query executed successfully." and other system information.

7. Count of instructors and their average salary

```

DECLARE @countOfInstructors int;
DECLARE @avgAnnualSalary money;
SELECT @countOfInstructors = COUNT(InstructionID), @avgAnnualSalary =
AVG(AnnualSalary) FROM Instructors;
IF @countOfInstructors >=10
    PRINT 'Count of Instructors is : '+ CAST(@countOfInstructors as
NVARCHAR(9))+'
        ' Average Annual Income of Instructors is : '+ CAST(@avgAnnualSalary as
NVARCHAR(9));
ELSE

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query window with the following T-SQL script:

```

PRINT 'The number of fulltime instructors is less than 10';

```

The script also includes logic to calculate the average annual salary of instructors and print it if the count of instructors is greater than or equal to 10. The execution results show the output of the PRINT statements and the completion time.

8. Delete the department with the name 'Sociology' using scripts

```

BEGIN TRY
DELETE FROM Departments WHERE DepartmentName = 'Sociology'
PRINT 'SUCCESS: Record was deleted';
END TRY
BEGIN CATCH
PRINT 'FAILURE: Record was not deleted.';
PRINT 'Error ' + CONVERT(varchar, ERROR_NUMBER(), 1)+ ':' +
ERROR_MESSAGE();

```

```
END CATCH;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "Proj1C.sql - DESKTOP-5MGKLN\MyCollege (DESKTOP-5MGKLN\admin (63)) - Microsoft SQL Server Management Studio". The Object Explorer pane on the left shows the database structure for "DESKTOP-5MGKLN\SQL Server 15.0.2000.5 - DESKTOP-5MGKLN". The main pane contains a T-SQL script:

```
BEGIN TRY
    DELETE FROM Departments WHERE DepartmentName = 'Sociology'
    PRINT 'SUCCESS: Record was deleted.'
END TRY
BEGIN CATCH
    PRINT 'FAILURE: Record was not deleted.'
    PRINT 'Error ' + CONVERT(varchar, ERROR_NUMBER(), 1) + ':' + ERROR_MESSAGE();
END CATCH;
```

The status bar at the bottom indicates "Query executed successfully." and shows the completion time: "Completion time: 2022-11-11T16:15:26.8452344-05:00".

9. Total course units using scripts

```
CREATE FUNCTION fnStudentUnits(@studentID int)
RETURNS INT
BEGIN
RETURN
(SELECT sc.StudentID, SUM(CourseUnits) as TotalCourseUnits FROM Students s JOIN
StudentCourses sc ON s.StudentID = sc.StudentID
JOIN Courses c ON sc.CourseID = c.CourseID WHERE s.StudentID = @studentID GROUP
BY sc.StudentID);
```

END

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the connection is to DESKTOP-5MGKLNUI.MyCollege (DESKTOP-5MGKLNUI\admin (63)). A query window titled 'Proj1C.sql - DESKTOP-5MGKLNUI.admin (63)' contains the following T-SQL code:

```
CREATE FUNCTION fnStudentUnits(@studentID int)
RETURNS INT
BEGIN
RETURN
(SELECT SUM(CourseUnits) AS TotalCourseUnits FROM Students s JOIN StudentCourses sc ON s.StudentID = sc.StudentID
JOIN Courses c ON sc.CourseID = c.CourseID WHERE s.StudentID = @studentID);
END
```

The status bar at the bottom indicates 'Completion time: 2022-11-11T16:21:11.8544385-06:00'.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the connection is to DESKTOP-5MGKLNUI.MyCollege (DESKTOP-5MGKLNUI\admin (63)). A query window titled 'Proj1C.sql - DESKTOP-5MGKLNUI.admin (63)' contains the following T-SQL code:

```
SELECT dbo.fnStudentUnits(10) AS StudentUnits10;
```

The results pane shows the output:

StudentUnits10
1
7

The status bar at the bottom indicates 'Completion time: 2022-11-11T16:22:00.0000000-06:00'.

10. return the StudentID from the StudentCourses table, the CourseNumber and CourseUnits from the Courses table, and the value return by the fnStudentUnits function for that student

```
CREATE FUNCTION fnStudentUnits(@studentID int)
RETURNS TABLE
RETURN
```

```

(SELECT s.StudentID, SUM(CourseUnits) as TotalCourseUnits
FROM Students s JOIN StudentCourses sc ON s.StudentID = sc.StudentID
JOIN Courses c ON sc.CourseID = c.CourseID
WHERE s.StudentID = @studentID
GROUP BY s.StudentID);
SELECT * FROM fnStudentUnits(20);
SELECT sc.StudentID, c.CourseNumber, c.CourseUnits, f.TotalCourseUnits
FROM Students s JOIN StudentCourses sc ON s.StudentID = sc.StudentID
JOIN Courses c ON sc.CourseID = c.CourseID
JOIN fnStudentUnits(20) f ON s.StudentID = f.StudentID;

```

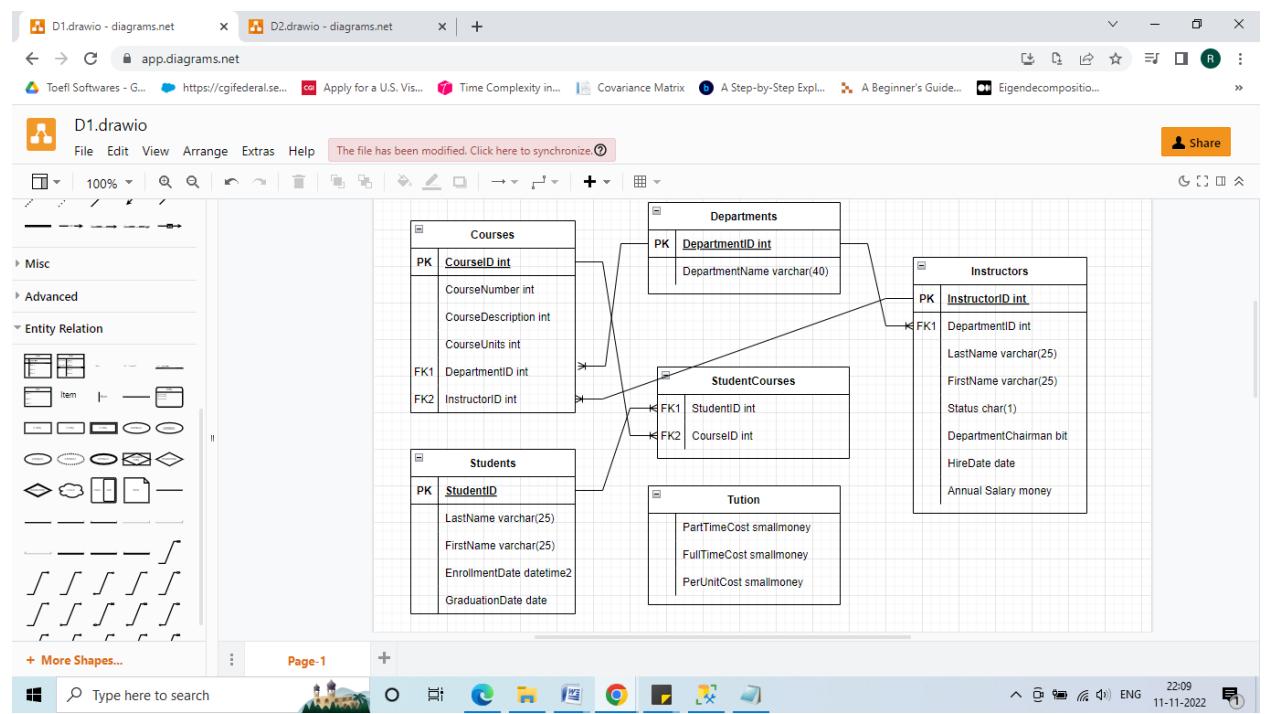
The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure for 'MyCollege'. In the center, the main query window contains the T-SQL code provided above. The results pane at the bottom shows a table with two rows of data:

StudentID	CourseNumber	CourseUnits	TotalCourseUnits
25	34301	3	6
29	15407	3	6

A status bar at the bottom indicates 'Query executed successfully.'

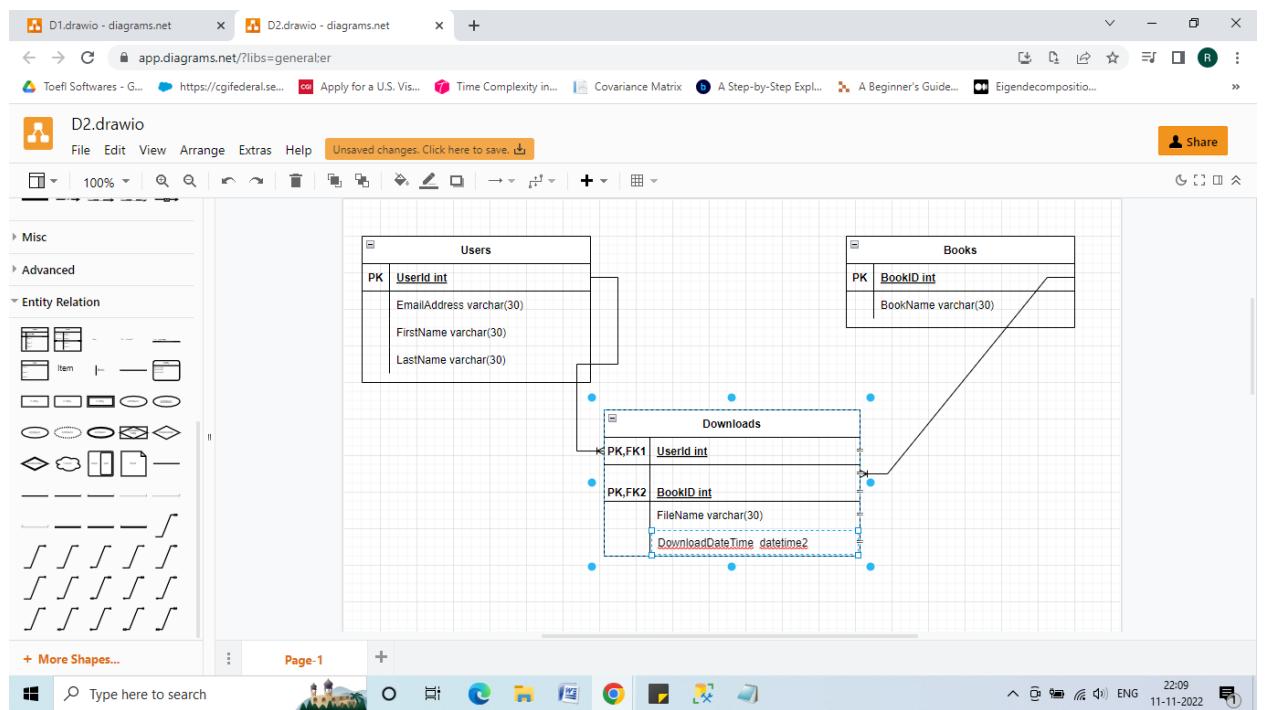
Database Design :

1.



In this database, there is a many to many relationship between Courses and Students table. To eliminate this, we defined a linking table named StudnetsCourses table. Course ID and Student ID are the primary keys of Courses and Students table respectively. One to many relationship between Departments and Courses table, one department can have many courses and many instructors. There is a one to many relationship between Student and StudentCourses table as a student can take many courses. One to many relationship between Instructor and Courses table as an instructor can take many courses.

2.



There will be three tables, Users, Books and Downloads. Users table have UserID as its primary key and BookId as primary key of Books table. There is a many to many relationship Users and Books table. One user can download many books and one book can be downloaded by many users. To eliminate this, we defined a linking table Downloads which will have composite key of UserID and BookId.

Remarks :

In this project, we did database setup, used DML statements .

Used views, scripts, stored procedures and functions on the tables in the database.

Also, we designed database diagrams based on the specifications given.