

CSE 581: INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

PROJECT: 1

NAME: PRAPTI OZA

SUID: 951770399

DATE: 11/11/2022

ABSTRACT

The main goal of this project is to demonstrate how to load and manipulate databases using the SQL Server Management Studio. All of the fundamental SQL query concepts are explored. Views, scripts, scalar and table-valued functions, database implementation, and more concepts are covered. designing of a database using E-R diagram is also a significant part. This project is centered on most of the important SQL and database related concepts.

TABLE OF CONTENTS

Abstract.....	1
A. Database Setup.....	3
B. Essential SQL Skills.....	11
C. Advanced SQL Skills (views/stores procedures/ functions / scripts).....	40
D. Database Design.....	51
Remarks.....	53

A. DATABASE SETUP

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

Ans:

```

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
Completion time: 2022-10-31T17:55:56.2316004-04:00
112 % 112 %
Query executed successfully.

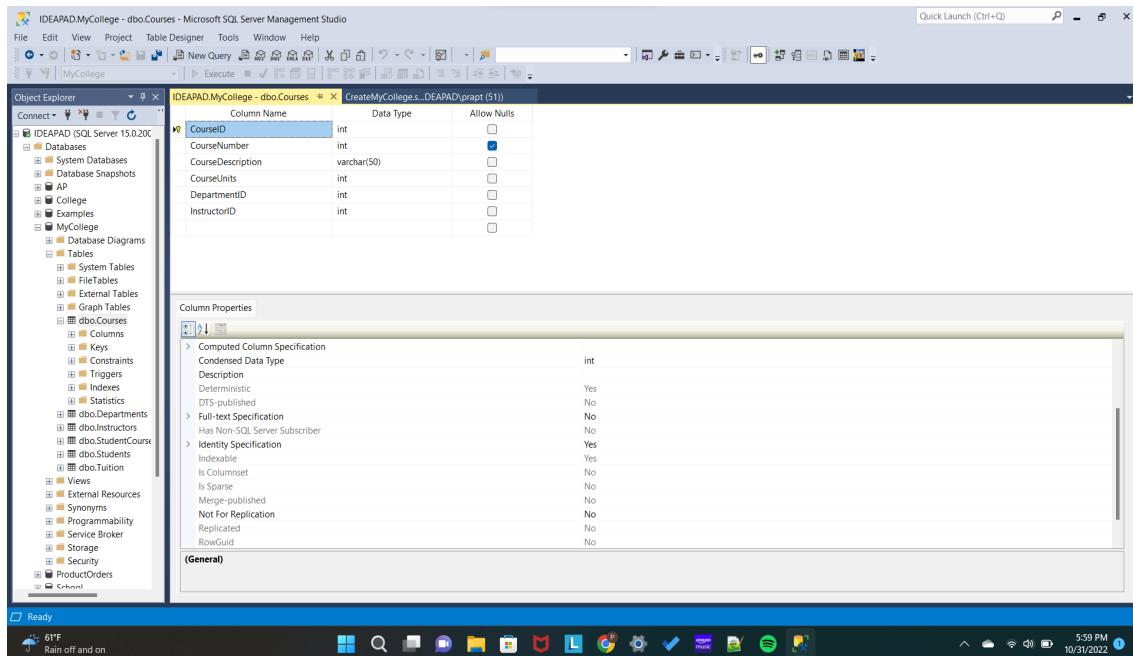
```

Comment: Running the script in SQL Server Management Studio gives the message as above in the Message tab after getting executed successfully.

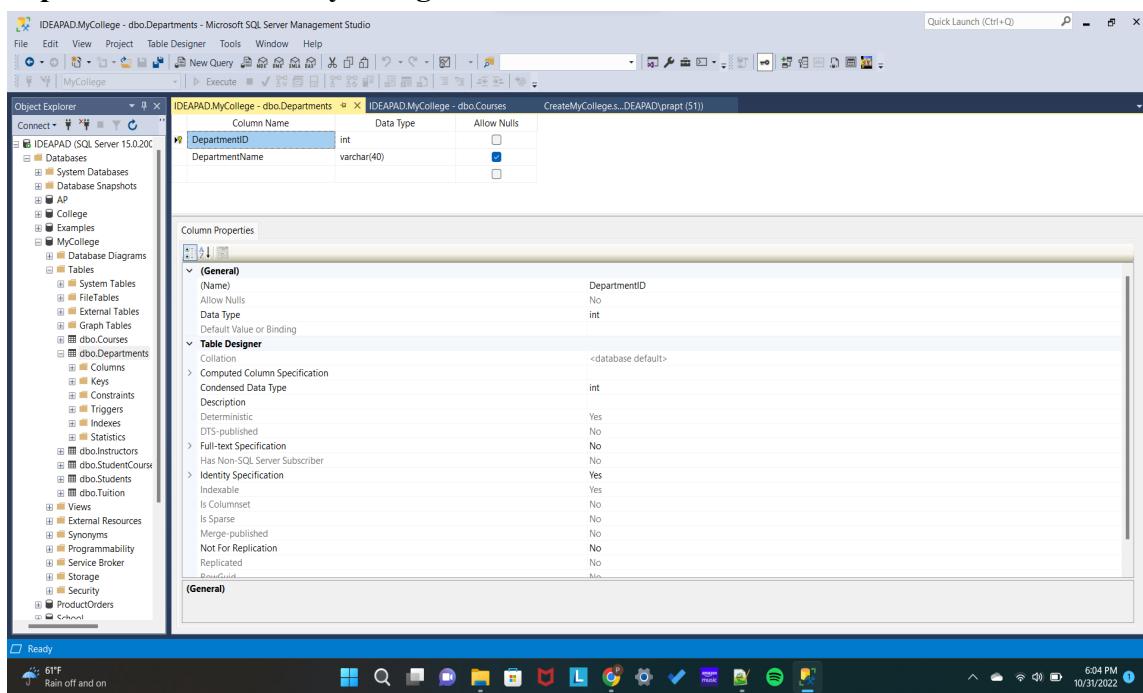
2. Navigate through the database objects and view the column definitions for each table. Open a new Query Editor window. Show details in Courses table and Instructors table using SELECT statement.

Ans:

Courses table in MyCollege database:



Departments table in MyCollege database:



Instructors table in MyCollege database:

Instructors Table Columns:

Column Name	Data Type	Allow Nulls
InstructorID	int	<input checked="" type="checkbox"/>
LastName	varchar(25)	<input type="checkbox"/>
FirstName	varchar(25)	<input checked="" type="checkbox"/>
Status	char(1)	<input type="checkbox"/>
DepartmentChairman	bit	<input type="checkbox"/>
HireDate	date	<input checked="" type="checkbox"/>
AnnualSalary	money	<input type="checkbox"/>
DepartmentID	int	<input type="checkbox"/>

InstructorID Column Properties:

- (General)**
 - Name: InstructorID
 - Allow Nulls: No
 - Data Type: int
 - Default Value or Binding: <database default>
- Table Designer**
 - Collation: <database default>
 - Computed Column Specification: Condensed Data Type
 - Description: int
 - Deterministic: Yes
 - DTS-published: No
 - Full-text Specification: Has Non-SQL Server Subscriber
 - Identity Specification: Yes
- (General)**
 - Is Columnset: No
 - Is Sparse: No
 - Merge-published: No
 - Not For Replication: No
 - Published: No

StudentCourses table in MyCollege database:

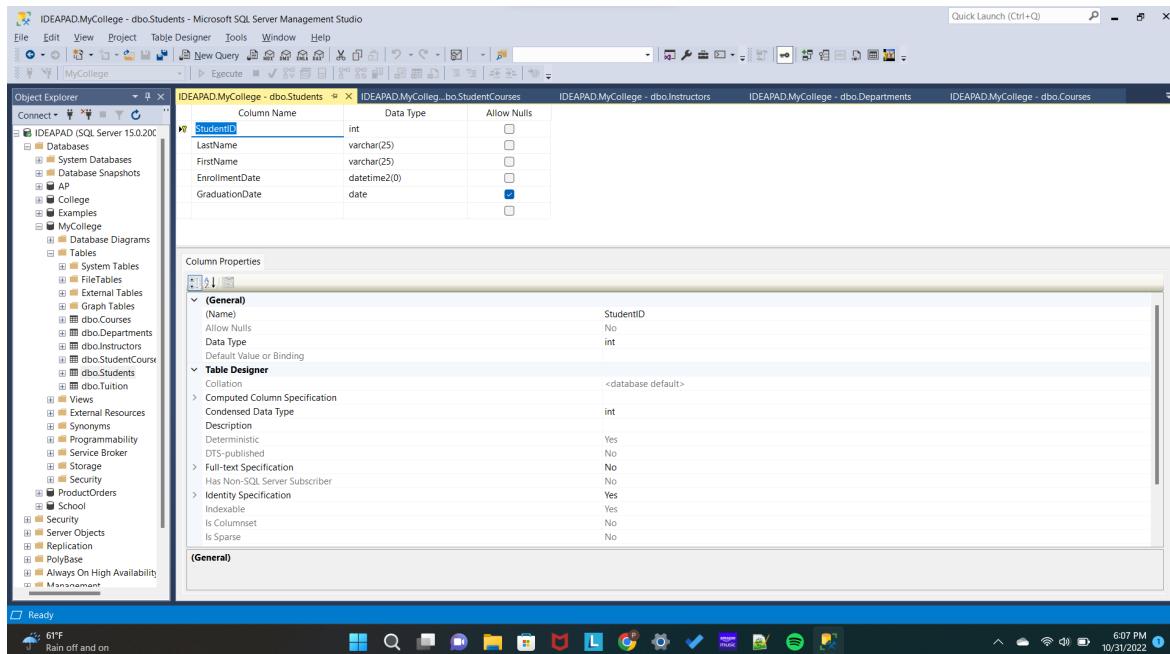
StudentCourses Table Columns:

Column Name	Data Type	Allow Nulls
StudentID	int	<input checked="" type="checkbox"/>
CourseID	int	<input type="checkbox"/>

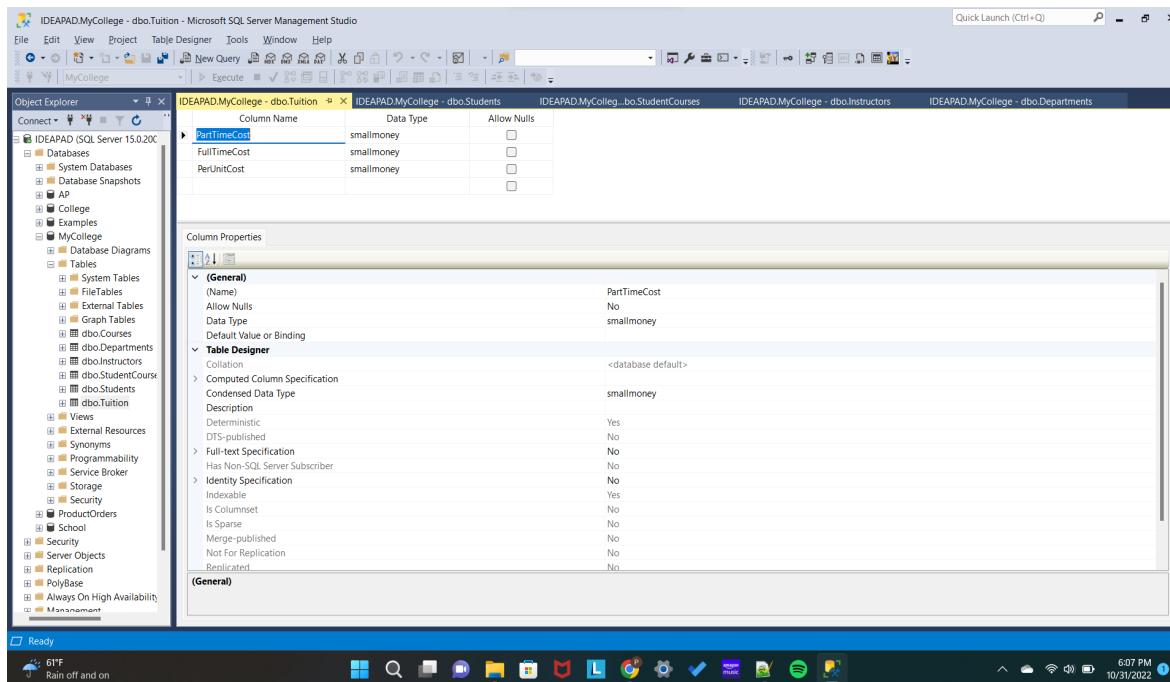
StudentID Column Properties:

- (General)**
 - Name: StudentID
 - Allow Nulls: No
 - Data Type: int
 - Default Value or Binding: <database default>
- Table Designer**
 - Collation: <database default>
 - Computed Column Specification: Condensed Data Type
 - Description: int
 - Deterministic: Yes
 - DTS-published: No
 - Full-text Specification: Has Non-SQL Server Subscriber
 - Identity Specification: No
- (General)**
 - Is Columnset: No
 - Is Sparse: No
 - Merge-published: No
 - Not For Replication: No
 - Published: No

Students table in MyCollege database:



Tuition table in MyCollege database:



SELECT * FROM Courses;

```
SQLQuery1.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (66)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Connect ...
SQLQuery1.sql - IDEAPAD\prapt (66) * 
SELECT * FROM Courses;
Results Messages
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 4 54321 Anatomy 3 6 16
5 5 82754 Social Psychology 3 7 9
6 6 13524 Social Analysis 3 1 11
7 24653 Intro to Marketing 3 4
8 8 22679 Intro to Calculus 3 4 5
9 9 80765 Intermediate Accounting 3 1 1
10 10 96032 Social Media 3 1 4
11 11 58230 Physiology 3 6 16
12 12 81256 Intro to Management 3 1 7
13 13 64321 Secondary Education 3 2 8
14 14 32751 Business Writing 2 1 10
15 15 46072 English 4 6 3
16 16 15407 Music Theory 3 5 12
17 17 28177 Classic Literature 3 3 15
18 18 90908 Educational Theory 3 2 13
19 19 55783 Shakespeare 3 3 15
20 20 63284 Population and Demo... 3 7 9
21 21 74832 Creative Writing 3 3 2
Query executed successfully.
Ln 1 Col 23 Ch 23 INS
8:21 PM 11/4/2022
```

SELECT * FROM Instructors;

```
SQLQuery1.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (66)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Connect ...
SQLQuery1.sql - IDEAPAD\prapt (66) * 
SELECT * FROM Instructors;
Results Messages
InstructorID LastName FirstName Status DepartmentChairman HireDate AnnualSalary DepartmentID
1 Brown Billy F 1 2016-01-10 77500.00 1
2 Thomas William P 0 2015-01-01 65000.00 3
3 Anderson Ahmed F 1 2016-06-05 75000.00 6
4 Green Denne F 0 2016-08-02 75000.00 1
5 McNegeor NULL F 1 2017-01-03 74000.00 4
6 Paxton Arnold P 0 2017-07-15 36000.00 5
7 Rogan 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 Derrick P 0 2018-01-17 35000.00 5
13 Black Bill P 0 2019-03-01 35000.00 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
Query executed successfully.
Ln 1 Col 26 Ch 26 INS
8:21 PM 11/4/2022
```

Comment: Column definitions of each table in the MyCollege database is shown. Using **SELECT *** statement all the details of the Courses and Instructors table is shown.

- Open another Query Editor window and then enter and run this statement:

SELECT COUNT(*) AS NumberOfInstructors FROM Instructors

Ans:

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying 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'. Below the results, a message says 'Query executed successfully.' The status bar at the bottom right indicates the date and time as '10/31/2022 6:10 PM'.

Comment: The COUNT(*) is used to return the count of all the rows of the table including the null values.

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

Ans:

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'InstructorDetails.sql' is open, displaying the following SQL statement:

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

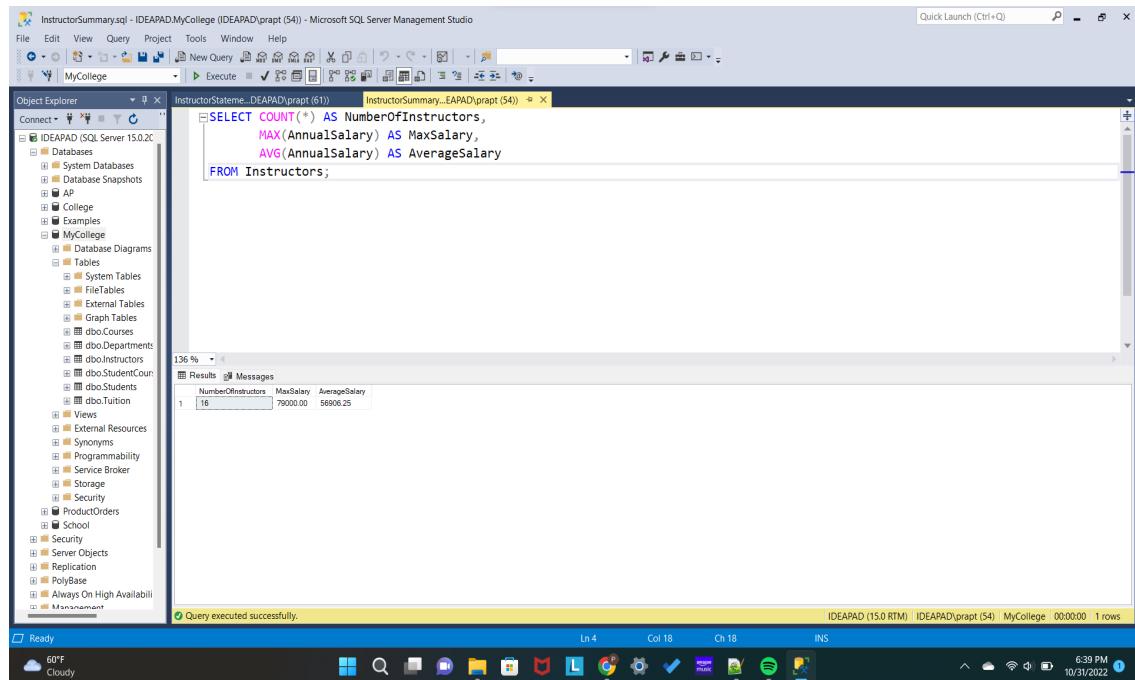
The results pane displays a table with 16 rows, showing columns for LastName, FirstName, HireDate, and AnnualSalary. The data includes names like Warren, Angela, Jonathan, and others, along with their hire dates and annual salaries. The status bar at the bottom right indicates the date and time as '10/31/2022 6:13 PM'.

Comment: Running the SQL statement in the InstructorDetails.sql script and getting 16 rows as output. The SELECT statement is used to select LastName, FirstName, HireDate,

and AnnualSalary from the Instructors table using FROM keyword and sorting the result set in ascending order on AnnualSalary column using the ORDER BY keyword.

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

Ans:



The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'MyCollege'. In the center, the 'Query Editor' window contains the following SQL script:

```

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

The 'Results' tab shows the output of the query:

NumberOfInstructors	MaxSalary	AverageSalary
18	79000.00	58906.25

At the bottom of the screen, the taskbar shows the system status: 'Ready', 'Cloudy', '60°F', '6:39 PM', '10/31/2022', and other icons.

Comment: Running the SQL statement in the InstructorSummary.sql script and getting 1 row as output. The SELECT statement is used to return the count of the number of instructors using COUNT(*) and naming the column as NumberOfInstructors using the AS keyword. Maximum of the AnuualSalary column is taken using the MAX() aggregate function and named as MaxSalary and average of the AnnualSalary is taken using AVG() aggregate function and named as AverageSalary.

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

Ans:

```

InstructorStatements.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (61)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
InstructorSummary_EAPAD(prapt (54)) InstructorStatement_EAPAD(prapt (61))
SELECT LastName, FirstName, HireDate, AnnualSalary
FROM Instructors
ORDER BY AnnualSalary;

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

```

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.

Ans:

LastName	FirstName	HireDate	AnnualSalary
Warren	Angela	2018-07-14	38000.00
Reed	Gill	2018-04-20	34000.00
Vlma	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	2018-03-30	38500.00
Gallegos	Tomas	2020-03-23	64000.00
Connors	Daniel	2018-03-04	71500.00
Drew	Daniel	2018-03-04	71500.00
Wright	John	2018-02-25	72000.00
Jones	Sally	2018-08-31	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	2018-06-05	79000.00

NumberofInstructors	MaxSalary	AverageSalary
16	79000.00	56906.25

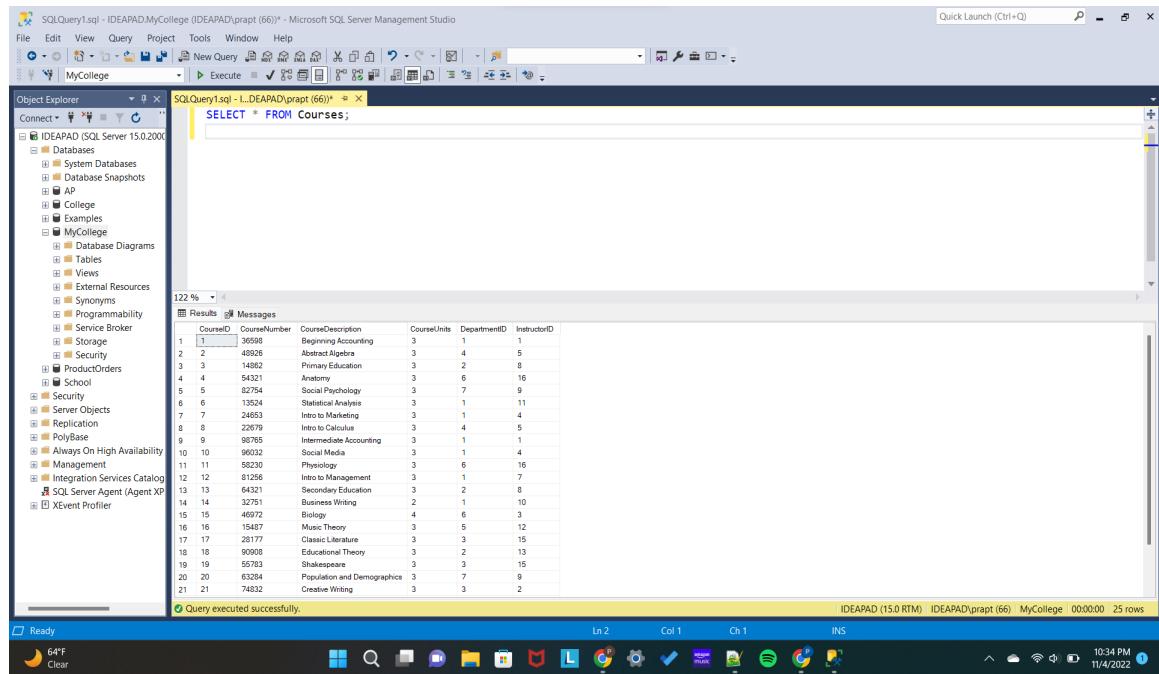
Comment: Running the `InstructorsStatements.sql` and getting total 17 rows as output.

8. Exit from SQL Server Management Studio.

B. ESSENTIAL SQL SKILLS

1. Write a SELECT statement that returns all of the columns from the Courses table. Then, run this statement to make sure it works correctly.

Ans: `SELECT * FROM Courses;`



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named 'MyCollege' containing various objects like Databases, Tables, and Views. The main window displays a query results grid for the 'Courses' table. The query is:

```
SELECT * FROM Courses;
```

The results grid shows 25 rows of course data with columns: CourseID, CourseNumber, CourseDescription, CourseUnits, DepartmentID, and InstructorID. The data includes courses like Beginning Accounting, Abstract Algebra, Primary Education, Anatomy, Secondary Psychology, Statistics Analysis, Intro to Marketing, Intro to Calculus, Intermediate Accounting, Social Media, Physiology, Intro to Management, Secondary Education, Business Writing, Biology, Music Theory, Classic Literature, Educational Theory, Shakespeare, Population and Demographics, and Creative Writing.

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	52754	Secondary Psychology	3	7	9
6	15524	Statistics Analysis	3	1	11
7	24653	Intro to Marketing	3	1	4
8	22879	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	55793	Shakespeare	3	3	15
20	63284	Population and Demographics	3	7	9
21	74832	Creative Writing	3	3	2

Query executed successfully.

Comment: Here, `SELECT *` statement is used to display all the records from the Courses table.

2. Write a SELECT statement that returns one column from the Students table named FullName that joins the LastName and FirstName columns. Format this column with the last name, a comma, a space, and the first name. 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.

Ans:

```
SELECT (LastName + ', ' + FirstName) AS FullName
FROM Students
WHERE LastName LIKE '[A-M]%'
ORDER BY LastName
```

```

SQLQuery1.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (66)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
SELECT (LastName + ', ' + FirstName) AS FullName
FROM Students WHERE LastName LIKE '[A-M]%'
ORDER BY LastName

```

Results Messages

FullName
Flores, Jesus
Franks, Karen
Gandy, Paul
Geara, Annette
George, Mona
Goodell, Conner
Griffin, Gerald
Hallowell, Jimmy
Hoffman, Wilma
Howard, Amber
Jackson, Floyd
Jones, Jeffrey
Jones, Andrew
Kent, Thomas
Kramer, Maggie
Landy, William
Listle, Lisa
MacNamara, Tony
Manning, Vincent
Morrisey, Monica

Query executed successfully.

Comment: Here, SELECT statement is used to display a column named FullName which is created by concatenating the LastName and FirstName columns using ‘+’ operator. Only such records are to be displayed where the LastName starts with A to M which is done using LIKE operator and the condition from A to M is written as ‘[A-M]%' using WHERE clause. The result set is then sorted in ascending order using the ORDER BY keyword in ascending order.

3. Write a SELECT statement that returns these column names and data from the Instructors table:

- LastName The LastName column
- FirstName The FirstName column
- HireDate The HireDate column

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

Ans:

```

SELECT LastName, FirstName, HireDate
FROM Instructors
WHERE HireDate LIKE '2019%'
ORDER BY HireDate;

```

```

SELECT LastName, FirstName, HireDate
FROM Instructors
WHERE HireDate LIKE '2019%'
ORDER BY HireDate;

```

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

Query executed successfully.

Comment: Here, SELECT statement is used to display LastName, FirstName and HireDate columns from Instructors table and only that records. Only those records are selected whose HireDate is in 2019.

Q4) Write a SELECT statement that returns these column names and data from the Students table:

- 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

Sort the result set in ascending sequence by the MonthsAttended column.

Ans:

```

SELECT FirstName, LastName, EnrollmentDate,
GETDATE() AS CurrentDate,
DATEDIFF(Month,EnrollmentDate,GETDATE()) AS MonthsAttended
FROM Students
ORDER BY MonthsAttended

```

```

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

	FirstName	LastName	EnrollmentDate	CurrentDate	MonthsAttended
1	Stanley	Silver	2020-01-04 11:16:19	2022-11-10 19:07:21.880	34
2	Jones	Biden	2020-01-05 13:47:21	2022-11-10 19:07:21.880	34
3	Walter	Cramden	2019-12-15 10:18:37	2022-11-10 19:07:21.880	35
4	Lisa	Lisle	2019-12-17 11:42:28	2022-11-10 19:07:21.880	35
5	Mona	George	2019-12-22 15:29:44	2022-11-10 19:07:21.880	35
6	Timothy	Johnson	2019-08-22 09:01:04	2022-11-10 19:07:21.880	39
7	Andrew	Walker	2019-08-22 13:48:26	2022-11-10 19:07:21.880	39
8	Fay	Gardner	2019-07-23 10:45:57	2022-11-10 19:07:21.880	40
9	Karen	Frasier	2019-07-23 11:42:23	2022-11-10 19:07:21.880	40
10	Osmer	Goodell	2019-01-02 14:21:58	2022-11-10 19:07:21.880	46
11	Gerald	Griffin	2019-01-02 16:04:04	2022-11-10 19:07:21.880	46
12	Letitia	Osborne	2018-12-12 17:14:22	2022-11-10 19:07:21.880	47
13	Vincent	Manning	2019-12-14 15:37:43	2022-11-10 19:07:21.880	47
14	Tanya	Sommers	2018-07-22 15:41:12	2022-11-10 19:07:21.880	52
15	Andrew	Jones	2018-07-24 10:53:26	2022-11-10 19:07:21.880	52
16	Floyd	Jackson	2018-07-25 09:27:53	2022-11-10 19:07:21.880	52
17	Annette	Geary	2018-07-26 09:33:47	2022-11-10 19:07:21.880	52
18	Jesus	Flores	2018-01-03 16:23:47	2022-11-10 19:07:21.880	58
19	James	Camden	2018-01-04 11:12:31	2022-11-10 19:07:21.880	58
20	Barney	Easton	2018-01-04 14:14:02	2022-11-10 19:07:21.880	58
21	Anthony	Rincon	2017-12-08 09:59:15	2022-11-10 19:07:21.880	59
22	Jimmy	Hollowell	2017-12-08 19:10:25	2022-11-10 19:07:21.880	59
23	George	Burns	2017-07-12 13:05:41	2022-11-10 19:07:21.880	64
24	Anderson	Yount	2017-07-18 14:21:07	2022-11-10 19:07:21.880	64

Query executed successfully.

Comment: Here, the SELECT statement is used to display the FirstName, LastName, EnrollmentDate and CurrentDate columns. CurrentDate column is obtained by the GETDATE() function that takes the current date. DATEDIFF() function is used to calculate the MonthsAttended column where Months are calculated subtracting EnrollmentDate from current date. The result set is sorted in ascending order on the MonthsAttended column.

Q5) Write a SELECT statement that returns these column names and data from the Instructors table:

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

Return only the top 20 percent of instructors based on annual salary.

Ans:

```

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

```
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q) P X
Object Explorer
File Explorer
MyCollege
Object Explorer
Connect
PAD (SQL Server 15.0.2002)
databases
System Databases
Database Snapshots
AP
College
Examples
MyCollege
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Courses
dbo.Departments
dbo.Instructors
dbo.StudentCourses
dbo.Students
dbo.Tuition
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
ProductOrders
School
Security
Server Objects
Replication
FileBase
ways On High Availability
Management
Results Messages
1. FirstName LastName AnnualSalary
   1. Rachel Amundsen 79000.00
   2. Billy Brown 77500.00
   3. Gene Green 75000.00
   4. NULL McGregor 74000.00
Query executed successfully.
IDEAPAD (15.0 RTM) IDEAPAD\prapt (62) MyCollege 00:00:00 4 rows
Ready
50°F Clear
Ln 5 Col 1 Ch 1 INS
7:08 PM 11/10/2022
```

Comment: Here, SELECT TOP statement is used to display specific number of statements that is 20 percent. FirstName, LastName and AnnualSalary columns are displayed from the Instructors table and is sorted in descending order on the AnnualSalary column.

Q6) Write a SELECT statement that returns these columns and data from the Tuition table, along with a constant value and two calculated values:

- 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

Ans:

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

```

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

```

	FullTimeCost	PerUnitCost	Units	TotalPerUnitCost	TotalTuition
1	1250.00	62.50	12	750.00	2000.00

Query executed successfully.

Comment: Here, SELECT statement is used to display FullTimeCost column, PerUnitCost, a constant value 12 named as Units column, calculated value of PerUnitCost * 12 named as TotalPerUnitCost and calculated value FullTimeCost + (PerUnitCost * 12) named as TotalTuition from Tuition table.

Q7) Write a SELECT statement that joins the Courses table to the Departments table and returns these columns: CourseNumber, CourseDescription, DepartmentName. Sort the result set by DepartmentName and then by CourseNumber in ascending order.

Ans:

```

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

```

```

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

```

CourseNumber	CourseDescription	DepartmentName
13524	Statistical Analysis	Business
24653	Intro to Marketing	Business
32751	Business Writing	Business
36598	Beginning Accounting	Business
81256	Intro to Management	Business
96032	Social Media	Business
98765	Intermediate Accounting	Business
14862	Primary Education	Education
64321	Secondary Education	Education
90908	Educational Theory	Education
28177	Classic Literature	English
37946	Composition	English
74793	Short Story	English
74632	Creative Writing	English
23079	Intro to Calculus	Mathematics
44386	Trigonometry	Mathematics
49056	Astronomical Calculus	Mathematics

Query executed successfully.

Comment: Here, SELECT statement is used to display CourseNumber and CourseDescription columns from the Courses table and DepartmentName from the Departments table using JOIN keyword joining the two tables on DepartmentID using the ON keyword. The result set is then ordered by DepartmentName and CourseNumber is ascending order.

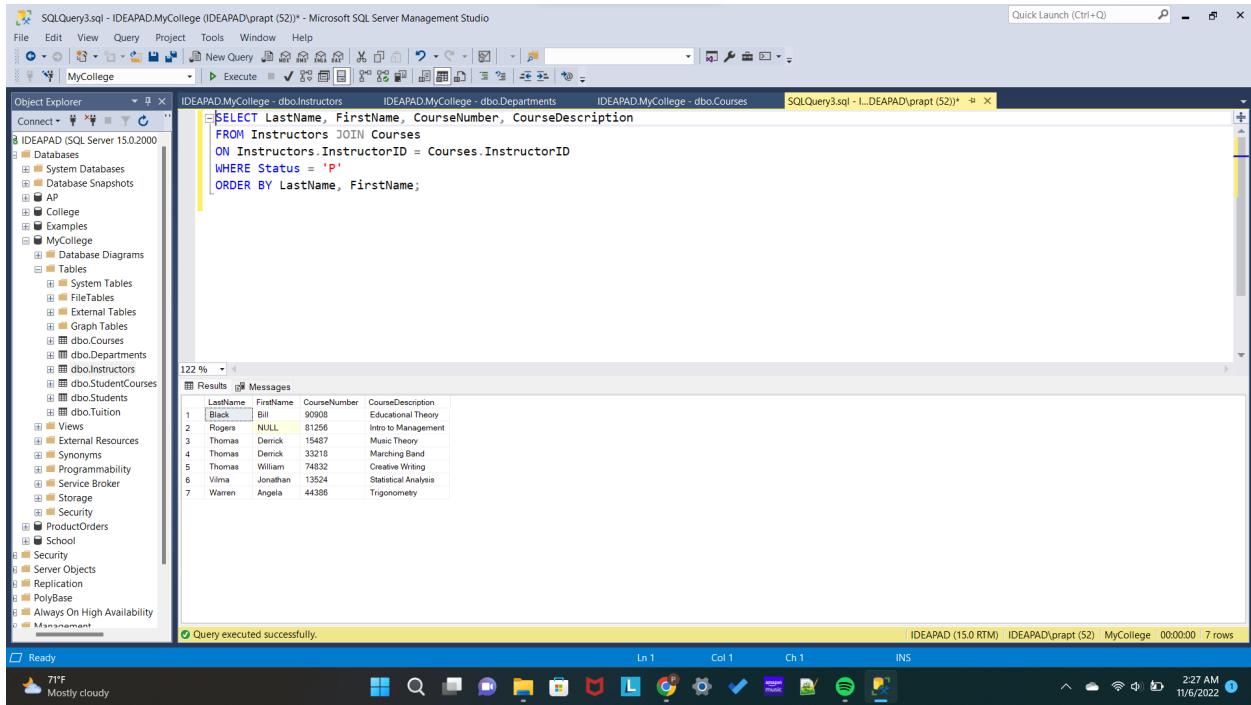
Q8) Write a SELECT statement that joins the Instructors table to the Courses table and returns these columns: LastName, FirstName, CourseNumber, CourseDescription. Return all courses for each instructor with a status of "P" (part time). Sort the result set by LastName and then by FirstName in ascending order.

Ans:

```

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

```



Comment: Here, SELECT statement is used to display the LastName and FirstName columns from the Instructors table and CourseNumber, CourseDescription from the Courses table by joining both the tables using JOIN keyword on InstructorID using the ON keyword. Only such records are displayed whose Status is ‘P’ which is done using the WHERE clause. The result set is then ordered by the LastName and then FirstName in ascending order.

Q9) Use the UNION operator to generate a result set consisting of five columns from the Students table:

- 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.

Ans:

```
SELECT 'GRADUATED' AS Status, FirstName, LastName,  
EnrollmentDate, GraduationDate FROM Students
```

```

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

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database structure. The right pane contains a query window with the following SQL code:

```

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

```

The results grid shows the output of the query. It has two columns: Status and FirstName. The Status column alternates between 'GRADUATED' and 'UNDERGRAD'. The FirstName column lists student names. The results are ordered by EnrollmentDate.

Status	FirstName
GRADUATED	Amber
GRADUATED	George
GRADUATED	Tony
GRADUATED	Jonathan
UNDERGRAD	Donna
GRADUATED	Rose
GRADUATED	Jesse
UNDERGRAD	Bonnie
UNDERGRAD	Thomas
GRADUATED	Maggie
UNDERGRAD	John
UNDERGRAD	Frank
GRADUATED	Roberts
UNDERGRAD	Wilma
UNDERGRAD	Brian
UNDERGRAD	Cel
UNDERGRAD	Charles
UNDERGRAD	William
UNDERGRAD	Monica
UNDERGRAD	George

At the bottom of the results grid, it says "Query executed successfully." The status bar at the bottom right shows the date and time: 11/10/2022 7:14 PM.

Comment: Here, For the first result set, SELECT statement is used to display the columns Status which is set to 'GRADUATED' as its value, FirstName, LastName, EnrollmentDate, GraduationDate from the table Students. Furthermore, records are selected such that the GraduationDate values are not null using the WHERE clause and IS NOT NULL operator.

In the second result set, SELECT statement is used to display the columns Status which is set to 'GRADUATED' as its value, FirstName, LastName, EnrollmentDate, GraduationDate from the table Students. Also, records are selected such that the GraduationDate values are NULL using the WHERE clause and IS NULL operator.

Finally both these result sets are combined using the UNION operator and the final result set is sorted in ascending order on EnrollmentDate using ORDER BY keyword.

Q10) Write a SELECT statement that returns these 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.

(Hint: You will need to join the Departments table to both the Instructors table and the Courses table, which will require you to use table aliases to distinguish the two tables.)

Ans:

```
SELECT Insdept.DepartmentName AS 'InstructorDept',
I.LastName, I.FirstName,C.CourseDescription,
Coudept.DepartmentName AS 'CourseDept'
FROM Courses AS C
JOIN Departments AS Coudept
ON C.DepartmentID = Coudept.DepartmentID
JOIN Instructors AS I
ON C.InstructorID = I.InstructorID
JOIN Departments AS Insdept
ON I.DepartmentID = Insdept.DepartmentID
WHERE Coudept.DepartmentID != Insdept.DepartmentID;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'MyCollege' selected. The right pane contains a query window with the following SQL code:

```
SELECT Insdept.DepartmentName AS 'InstructorDept',
I.LastName, I.FirstName,C.CourseDescription,
Coudept.DepartmentName AS 'CourseDept'
FROM Courses AS C
JOIN Departments AS Coudept
ON C.DepartmentID = Coudept.DepartmentID
JOIN Instructors AS I
ON C.InstructorID = I.InstructorID
JOIN Departments AS Insdept
ON I.DepartmentID = Insdept.DepartmentID
WHERE Coudept.DepartmentID != Insdept.DepartmentID;
```

The results pane shows the output of the query:

InstructorDept	LastName	FirstName	CourseDescription	CourseDept
English	Jones	Sally	Business Writing	Business

A message at the bottom of the results pane says "Query executed successfully."

Comment: Here SELECT statement displays the InstructorDept column that is the column that has the DepartmentName for a related instructor from Departments table, LastName and

FirstName from Instructors table, CourseDescription from Courses and CourseDept column that is the column with DepartmentName related to instructor. These tables are joined using the JOIN keyword and table aliasing to distinguish the tables.

Q11) Write a SELECT statement that returns 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

(Hint: You will need to concatenate the instructor first and last names again in the GROUP BY clause.)

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

Ans:

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

```
SQLQuery2.sql - IDEAPAD\prapt (62) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
MyCollege | Execute ✓ SQL Server Object Explorer
Object Explorer
PAD (SQL Server 15.0.2000)
databases System Databases Database Snapshots AP
College Examples MyCollege
Database Diagrams Tables
System Tables FileTables External Tables Graph Tables
dbo.Courses dbo.Instructors dbo.StudentCourse
dbo.Students dbo.Tuition Views
External Resources Synonyms Programmability Service Broker Storage Security ProductOrders School
Security Server Objects Application myBase ways On High Availability
ln 6 Col 1 Ch 1 INS
60°F Clear
7:17 PM 11/10/2022
```

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

NoOfCourses	InstructorName	TotalCourseUnits
3	NULL	9
2	Rachel Amundsen	8
2	Tomas Gallegos	6
2	Gene Green	6
2	John Smith	6
2	Billy Wilson	6
2	Daniel Connor	6
2	Daniel Drew	6
2	Derrick Thomas	5
2	Sally Jones	5
1	William Thomas	3
1	Jonathan Vilma	3
1	Angela Warren	3
1	Bill Black	3

Query executed successfully.

Comment: Here, SELECT statement is used to display the count of the number of courses offered by a particular instructor which is done by COUNT(*) aggregate function on the Courses table, the name of the instructor which is the concatenation of FirstName and LastName columns of the Instructors table done using ‘+’ operator, sum of the CourseUnits given by each instructor from the Courses table that is done using the SUM() aggregate function. The Instructors and Courses tables are joined using JOIN keyword on the InstructorID column using ON keyword. The result set is grouped by the concatenation of FirstName and LastName using GROUP BY keyword and finally it is sorted in descending order on TotalCourseUnits column using ORDER BY keyword.

Q12) Write a SELECT statement that answers this question: What is the total number of courses taught by parttime instructors? Return these columns:

- The 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.

Ans:

```
SELECT LastName + ', ' + FirstName AS InstructorName, COUNT(*) AS TotalCourses
FROM Instructors JOIN Courses
ON Instructors.InstructorID = Courses.InstructorID
WHERE Status = 'P'
GROUP BY ROLLUP (LastName + ', ' + FirstName)
```

```

SQLQuery2.sql - IDEAPAD\prapt (62) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query New Query Quick Launch (Ctrl+Q)
MyCollege
Object Explorer
File Server MyCollege
PAD (SQL Server 15.0.2000.16000)
databases System Databases Database Snapshots AP
College Examples MyCollege
Database Diagrams Tables System Tables FileTables External Tables Graph Tables
dbo.Courses dbo.Departments dbo.Instructors dbo.StudentCourse dbo.Students dbo.Tuition Views External Resources Synonyms Programmability Service Broker Storage Security ProductOrders School security server Objects replication Primary Key On High Availability
Results Messages
InstructorName TotalCourses
1 NULL 1
2 Black, Bill 1
3 Thomas, Derrick 2
4 Thomas, William 1
5 Vilma, Jonathan 1
6 Warren, Angela 1
7 NULL 7
Query executed successfully.

```

Comment: Here, SELECT statement is used to display the InstructorName that is the concatenation of the LastName and the FirstName columns from the Instructors table and the COUNT(*) aggregate function is used to count the total number of courses which is named as TotalCourses using AS keyword. The Instructors table and Courses table are joined using JOIN keyword on InstructorID column using ON keyword. Only those records are displayed whose Status is 'P' in the Instructors table which is done using the WHERE clause. The result set is grouped by on the concatenation of LastName and FirstName and ROLLBACK operator is added to the GROUP BY clause to display the grand total of the number of courses.

Q13) Write a SELECT statement that returns the same result set as this SELECT statement, but don't use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

```

SELECT DISTINCT LastName, FirstName
FROM Instructors i JOIN Courses c
    ON i.InstructorID = c.InstructorID
ORDER BY LastName, FirstName

```

Ans:

```

SELECT LastName, FirstName
FROM Instructors
WHERE InstructorID IN (SELECT DISTINCT InstructorID FROM Courses)
ORDER BY LastName, FirstName

```

```

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

```

LastName	FirstName
Arendsen	Michelle
Bloom	Bill
Brown	Billy
Conner	Daniel
Drew	Daniel
Gallegos	Tomas
Green	Gene
Jones	Sally
McGregor	NULL
Rogers	NULL
Smith	John
Thomas	Derrick
Thomas	William
Vilma	Jonathan
Warren	Angela

Query executed successfully.

Comment: Here, SELECT statement is used to display the only those LastName and FirstName from the Instructors table which are in the Courses table which is done using the WHERE clause and a subquery. SELECT statement is used to retrieve the distinct InstructorID from the Courses table and IN operator is used to retrieve the records from the Instructors table whose InstructorID are in the result set of the subquery. The result set is sorted in ascending order using ORDER BY keyword on LastName, FirstName.

Q14) Write a SELECT statement that returns one row for each course with these columns:

- The CourseID column from the Courses table
- The most recent enrollment date for that course from the Students table

Change the SELECT statement to a CTE. Then, write a SELECT statement that returns 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.

Ans:

```

WITH CourseDetails AS
(SELECT Courses.CourseID, MAX(EnrollmentDate) AS MaxOfEnrollmentDate
FROM Courses JOIN StudentCourses
ON Courses.CourseID = StudentCourses.CourseID
JOIN Students
ON StudentCourses.StudentID = Students.StudentID
GROUP BY Courses.CourseID)

```

```

SELECT CourseDescription, LastName, FirstName, EnrollmentDate
FROM Courses JOIN StudentCourses
ON Courses.CourseID = StudentCourses.CourseID
JOIN Students
ON StudentCourses.StudentID = Students.StudentID
JOIN CourseDetails
ON Courses.CourseID = CourseDetails.CourseID
AND Students.EnrollmentDate = CourseDetails.MaxOfEnrollmentDate

```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'MyCollege'. The central pane contains a multi-line T-SQL script. The results pane below shows a table with 24 rows of data, indicating successful execution. The status bar at the bottom right shows the date and time as 11/6/2022 5:07 PM.

```

WITH CourseDetails AS
(
    SELECT Courses.CourseID, MAX(EnrollmentDate) AS MaxOfEnrollmentDate
    FROM Courses JOIN StudentCourses |
    ON Courses.CourseID = StudentCourses.CourseID
    JOIN Students
    ON StudentCourses.StudentID = Students.StudentID
    GROUP BY Courses.CourseID
)

SELECT CourseDescription, LastName, FirstName, EnrollmentDate
FROM Courses JOIN StudentCourses
ON Courses.CourseID = StudentCourses.CourseID
JOIN Students
ON StudentCourses.StudentID = Students.StudentID
JOIN CourseDetails
ON Courses.CourseID = CourseDetails.CourseID
AND Students.EnrollmentDate = CourseDetails.MaxOfEnrollmentDate

```

CourseDescription	LastName	FirstName	EnrollmentDate
Biology	Bonwell	Brian	2016-12-12 14:22:53
Microbiology	Camden	James	2018-01-04 11:12:31
Social Media	Geary	Annette	2018-07-12 09:33:47
Intro to Calculus	Manning	Vincent	2018-12-18 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
Computer Science	Johnson	Timothy	2019-08-04 09:01:04
Finer Education	Walker	Andrew	2019-08-05 15:46:29
Social Psychology	Walker	Andrew	2019-08-05 15:46:26
Statistical Analysis	Cramden	Walter	2019-12-15 10:18:37
Intermediate Accounting	Cramden	Walter	2019-12-15 10:18:37
Business Writing	Cramden	Walter	2019-12-15 10:18:37
Music Theory	Lisle	Lisa	2019-12-17 11:42:28
Marching Band	Lisle	Lisa	2019-12-17 11:42:28
Abstract Algebra	George	Mona	2019-12-22 15:29:44

Comment: Here, the name of the CTE is CourseDetails and the CTE holds the result set of a select query. This SELECT statement is used to select CourseID column from Courses table, maximum of the EnrollmentDate column using MAX() aggregate function which is named as MaxOfEnrollmentDate column from Students table. The tables Courses and StudentCourses are joined using JOIN keyword on the CourseID column which is further joined with Students table on StudentID. This result set is then grouped by the GROUP BY clause.

SELECT statement is used to display the CourseDescription, LastName, FirstName, EnrollmentDate columns for the students with most recent enrollment date. The Courses table is joined with StudentCourses table on CourseID which is joined with Student on StudentID which is further joined with the CTE CourseDetails on CourseID and MaxOfEnrollmentDate.

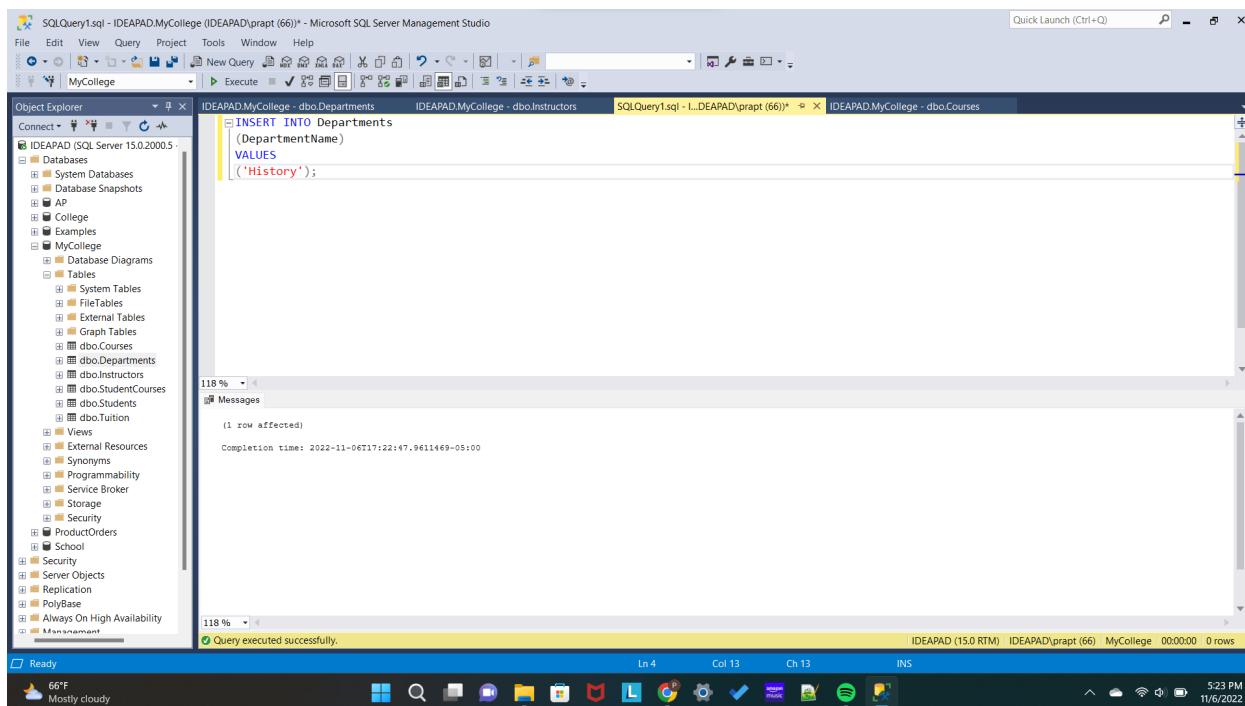
Q15) Write an INSERT statement that adds this row to the Departments table:

- DepartmentName: History

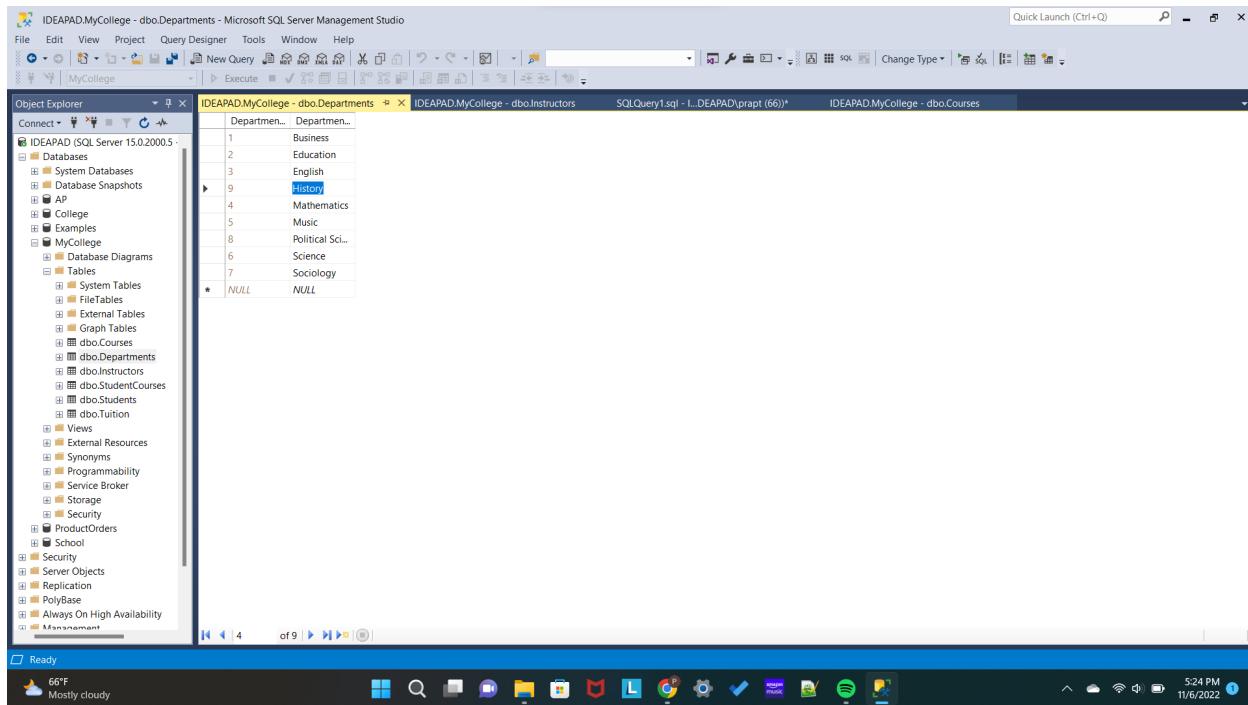
Code the INSERT statement so SQL Server automatically generates the value for the DepartmentID column.

Ans:

```
INSERT INTO Departments
(DepartmentName)
VALUES
('History');
```



Comment: Here, INSERT INTO statement is used to insert new values in the existing Departments table. Column list is specified followed by the correspond values to be added to that column. Here, DepartmentName column is taken and the value 'History' is inserted in it. This will generate a new record and SQL Server will automatically generate the new DepartmentID for this record.



Q16) Write a single INSERT statement that adds these rows to the Instructors table:

- InstructorID: The next automatically generated ID
LastName: Benedict
FirstName: Susan
Status: P
DepartmentChairman: 0
HireDate: Today's date
AnnualSalary: 34000.00
DepartmentID: 9
- InstructorID: The next automatically generated ID
LastName: Adams
FirstName: null
Status: F
DepartmentChairman: 1
HireDate: Today's date
AnnualSalary: 66000.00
DepartmentID: 9

Write this statement without using a column list.

Ans:

```
INSERT INTO Instructors
VALUES
('Benedict', 'Susan', 'P', 0, GETDATE(), 34000.00, 9),
('Adams', 'null', 'F', 1, GETDATE(), 66000.00, 9)
```

('Adams', NULL, 'F', 1, GETDATE(), 66000.00,9);

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 'SQLQuery1.sql - IDEAPAD\prapt (66)*' contains the following SQL code:

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

The message pane below the query window shows '(2 rows affected)' and 'Completion time: 2022-11-06 17:35:15.4089943-05:00'. At the bottom of the screen, a status bar indicates 'Query executed successfully.' and the system clock shows '5:35 PM 11/6/2022'.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'MyCollege' is selected. In the center pane, a results grid titled 'IDEAPAD.MyCollege - dbo.Instructors' displays the following data:

InstructorID	LastName	FirstName	Status	Department	HireDate	AnnualSalary	DepartmentID
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Paxton	Arnold	P	False	2017-07-15	36000.0000	5
7	Rogers	NULL	P	False	2017-10-22	38000.0000	1
8	Smith	John	F	True	2018-02-05	73000.0000	2
9	Connors	Daniel	F	True	2018-03-04	71500.0000	7
10	Jones	Sally	F	True	2018-09-21	74000.0000	3
11	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
12	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
13	Black	Bill	P	False	2019-04-20	34000.0000	2
14	Warren	Angela	P	False	2019-07-14	33000.0000	4
15	Drew	Daniel	F	False	2019-08-25	72000.0000	3
16	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
17	Benedict	Susan	P	False	2022-11-06	34000.0000	9
18	Adams	NULL	F	True	2022-11-06	66000.0000	9
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom of the screen, a status bar indicates '5:36 PM 11/6/2022'.

Comment: Here, `INSERT INTO` statement is used to insert values into the existing `Instructors` table. There is no need to specify the column list as values to be added can be specified in the specific column order. For the first record and second record to be added, all the values are mentioned in the same order as the column in the table followed by `VALUES`.

Q17) Write an UPDATE statement that modifies the first instructor you added in the above question. This statement should change the AnnualSalary column from 34,000 to 35,000, and it should use the InstructorID column to identify the row.

Ans:

UPDATE Instructors

SET

AnnualSalary = '35,000.00'

WHERE

InstructorID = 17;

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'IDEAPAD\MyCollege'. The central pane displays the following T-SQL code:

```

UPDATE Instructors
SET
    AnnualSalary = '35,000.00'
WHERE
    InstructorID = 17;

```

Below the code, the status bar indicates '(1 row affected)' and 'Completion time: 2022-11-06 17:43:21.7871080-05:00'. At the bottom, a green message bar says 'Query executed successfully.'

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists the database structure for 'IDEAPAD.MyCollege'. The main area displays the results of a query against the 'Instructors' table. The table has columns: InstructorID, LastName, FirstName, Status, Department, HireDate, AnnualSalary, and Department. The data shows 18 rows, with the last row being a placeholder with all NULL values.

InstructorID	LastName	FirstName	Status	Department	HireDate	AnnualSalary	Department
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Paxton	Arnold	P	False	2017-07-15	36000.0000	5
7	Rogers	NULL	P	False	2017-10-22	38000.0000	1
8	Smith	John	F	True	2018-02-05	73000.0000	2
9	Connors	Daniel	F	True	2018-03-04	71500.0000	7
10	Jones	Sally	F	True	2018-09-21	74000.0000	3
11	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
12	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
13	Black	Bill	P	False	2019-04-20	34000.0000	2
14	Warren	Angela	P	False	2019-07-14	33000.0000	4
15	Drew	Daniel	F	False	2019-08-25	72000.0000	3
16	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
17	Benedict	Susan	P	False	2022-11-06	35000.0000	9
18	Adams	NULL	F	True	2022-11-06	66000.0000	9
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Comment: Here, UPDATE statement is used to update the existing Instructors table. AnnualSalary is set to 35000 for the record whose InstructorID is 17 which is done using WHERE clause.

Q18) Write a DELETE statement that deletes the second instructor you added in question 21. This statement should use the InstructorID column to identify the row.

Ans:

DELETE FROM Instructors

WHERE

InstructorID = 18

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 'SQLQuery1.sql' contains the following SQL code:

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

The results pane shows the output of the query:

```
(1 row affected)

Completion time: 2022-11-06t18:02:10.8615253-05:00
```

At the bottom of the results pane, it says 'Query executed successfully.'

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 'SQLQuery1.sql' displays the contents of the 'Instructors' table:

InstructorID	LastName	FirstName	Status	Department	HireDate	AnnualSalary	Department
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Paxton	Arnold	P	False	2017-07-15	36000.0000	5
7	Rogers	NULL	P	False	2017-10-22	38000.0000	1
8	Smith	John	F	True	2018-02-05	73000.0000	2
9	Connors	Daniel	F	True	2018-03-04	71500.0000	7
10	Jones	Sally	F	True	2018-09-21	74000.0000	3
11	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
12	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
13	Black	Bill	P	False	2019-04-20	34000.0000	2
14	Warren	Angela	P	False	2019-07-14	33000.0000	4
15	Drew	Daniel	F	False	2019-08-25	72000.0000	3
16	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
17	Benedict	Susan	P	False	2022-11-06	35000.0000	9
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Comment: Here, DELETE FROM is used to delete a record from the Instructors table whose InstructorID is 18 using WHERE clause.

Q19) Write a DELETE statement that deletes the row in the Departments table that has an ID of 9. When you execute this statement, it will produce an error since the department has related rows in the Instructors table. To fix that, precede the DELETE statement with another DELETE statement that deletes all instructors in this department.

Ans:

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

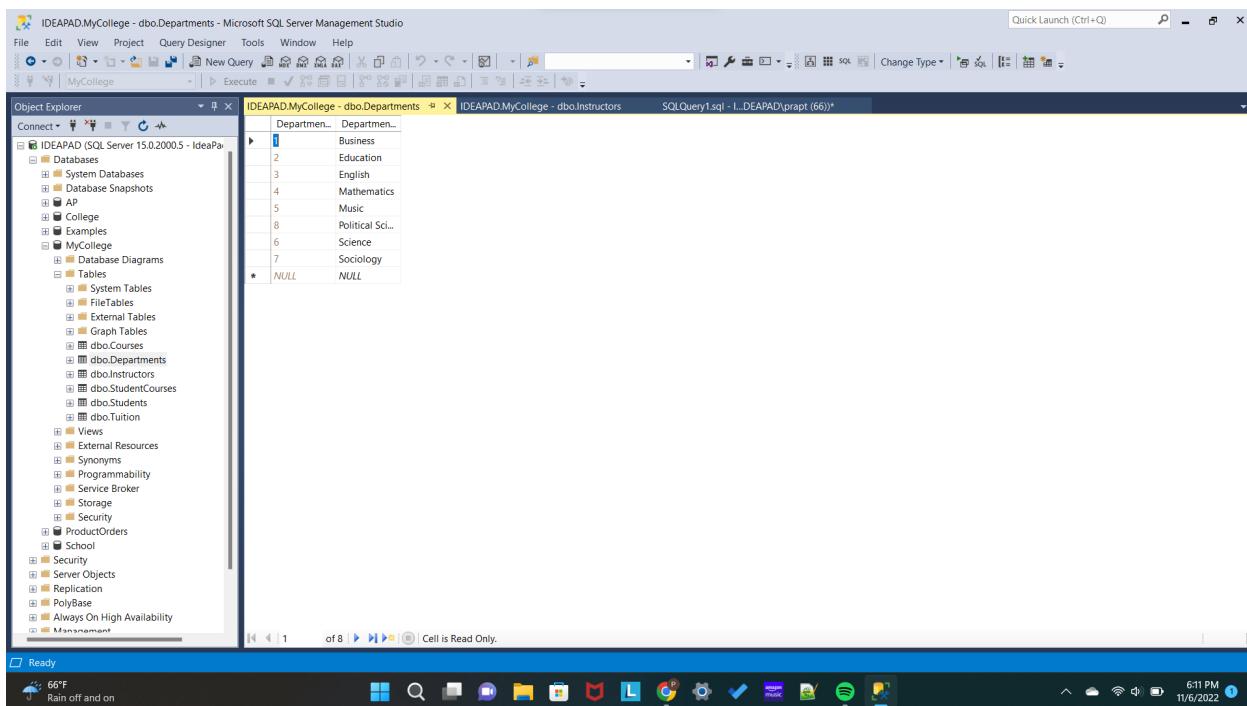
```
SQLQuery1.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (66)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
MyCollege | Execute New Query Find Replace Properties Task List Object Explorer
Object Explorer
Connect Connect to Database IDEAPAD (SQL Server 15.0.2000.5 - Ideapad)
Database System Databases Database Snapshots AP College Examples MyCollege Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.Courses dbo.Departments dbo.Instructors dbo.StudentCourses dbo.Students dbo.Tuition Views External Resources Synonyms Programmability Service Broker Storage Security ProductOrders School Security Server Objects Replication PolyBase Always On High Availability Management
SQLQuery1.sql - IDEAPAD\prapt (66)*
DELETE FROM Instructors
WHERE DepartmentID = 9
DELETE FROM Departments
WHERE DepartmentID = 9;

(1 row affected)
(1 row affected)
Completion time: 2022-11-06T18:10:52.5167028-05:00

Messages
Query executed successfully.

Ready 65°F Rain off and on 6:10 PM 11/6/2022
```

InstructorID	LastName	FirstName	Status	DepartmentID	HireDate	AnnualSalary	DepartmentName
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Paxton	Arnold	P	False	2017-07-15	36000.0000	5
7	Rogers	NULL	P	False	2017-10-22	38000.0000	1
8	Smith	John	F	True	2018-02-05	73000.0000	2
9	Connors	Daniel	F	True	2018-03-04	71500.0000	7
10	Jones	Sally	F	True	2018-09-21	74000.0000	3
11	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
12	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
13	Black	Bill	P	False	2019-04-20	34000.0000	2
14	Warren	Angela	P	False	2019-07-14	33000.0000	4
15	Drew	Daniel	F	False	2019-08-25	72000.0000	3
16	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



Comment: Here, to delete the record from the Departments table, first that reference is to be deleted from the Instructors table of that particular DepartmentID. So, first DELETE FROM statement is used to delete the record from Instructors table where the DepartmentID is 9 and the DELETE FROM is used to delete the record from the Departments table with same id.

Q20) Write an UPDATE statement that increases the annual salary for all instructors in the Education department by 5%. To do that, join the Departments and Instructors tables and then filter the rows by the department name.

Ans:

```
UPDATE Instructors
SET
AnnualSalary = AnnualSalary + (AnnualSalary * 0.5)
FROM Instructors JOIN Departments
ON Instructors.DepartmentID = Departments.DepartmentID
WHERE DepartmentName = 'Education';
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'IDEAPAD'. The central pane displays the following T-SQL code:

```
UPDATE Instructors
SET
    AnnualSalary = AnnualSalary + (AnnualSalary * 0.5)
FROM Instructors JOIN Departments
ON Instructors.DepartmentID = Departments.DepartmentID
WHERE DepartmentName = 'Education';
```

The status bar at the bottom indicates 'Query executed successfully.' and shows the completion time as 2022-11-07T21:21:53.4051048-05:00.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'IDEAPAD'. The central pane displays the results of the executed UPDATE query, showing a list of instructors with their updated annual salaries. The table has columns: InstructorID, LastName, FirstName, Status, DepartmentID, HireDate, AnnualSalary, and DepartmentName.

InstructorID	LastName	FirstName	Status	DepartmentID	HireDate	AnnualSal...	Departmen...
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Paxton	Arnold	P	False	2017-07-15	36000.0000	5
7	Rogers	NULL	P	False	2017-10-22	38000.0000	1
8	Smith	John	F	True	2018-02-05	109500.0000	2
9	Connors	Daniel	F	True	2018-03-04	71500.0000	7
10	Jones	Sally	F	True	2018-09-21	74000.0000	3
11	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
12	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
13	Black	Bill	P	False	2019-04-20	51000.0000	2
14	Warren	Angela	P	False	2019-07-14	33000.0000	4
15	Drew	Daniel	F	False	2019-08-25	72000.0000	3
16	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Comment: Here, UPDATE statement is used to update the Instructors table. The AnnualSalary column is updated by increasing the value to 5% for those records whose DepartmentName is 'Education'. This is done by joining the two tables Instructors and Departments table using JOIN keyword on DepartmentID column using WHERE clause and giving the condition of DepartmentName = 'Education'.

Q21) Write a DELETE statement that deletes instructors that aren't teaching any courses. To do that, use a subquery in the WHERE clause.

Ans:

DELETE FROM Instructors

WHERE InstructorID NOT IN (SELECT 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 main query window contains the following SQL code:

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

The status bar at the bottom indicates 'Query executed successfully.' and shows the completion time as 2022-11-07 21:56:46.3091918-05:00. The taskbar at the bottom right shows the date as 11/7/2022 and the time as 9:57 PM.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'MyCollege'. The main results window displays the 'Instructors' table with 16 rows of data. The 'Courses' table is also visible in the background.

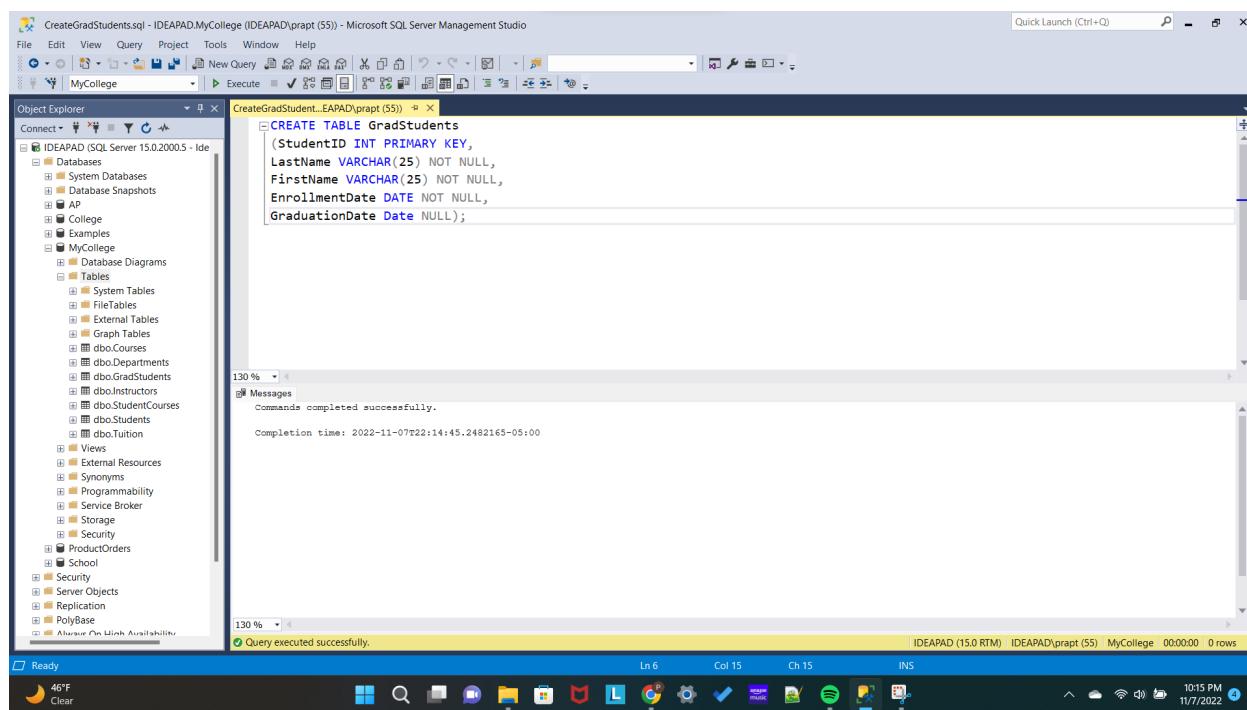
InstructorID	LastName	FirstName	Status	Department	HireDate	AnnualSalary	Department
1	Brown	Billy	F	True	2016-01-10	77500.0000	1
2	Thomas	William	P	False	2016-03-30	38500.0000	3
3	Amundsen	Rachel	F	True	2016-06-05	79000.0000	6
4	Green	Gene	F	False	2016-08-02	75000.0000	1
5	McGregor	NULL	F	True	2017-01-03	74000.0000	4
6	Rogers	NULL	P	False	2017-10-22	38000.0000	1
7	Smith	John	F	True	2018-02-05	109500.0000	2
8	Connors	Daniel	F	True	2018-03-04	71500.0000	7
9	Jones	Sally	F	True	2018-09-21	74000.0000	3
10	Vilma	Jonathan	P	False	2018-11-18	35500.0000	1
11	Thomas	Derrick	P	False	2019-01-17	35500.0000	5
12	Black	Bill	P	False	2019-04-20	51000.0000	2
13	Warren	Angela	P	False	2019-07-14	33000.0000	4
14	Drew	Daniel	F	False	2019-08-25	72000.0000	3
15	Gallegos	Tomas	F	False	2020-03-23	64000.0000	6
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The status bar at the bottom indicates 'Cell is Read Only.' and shows the date as 11/7/2022 and the time as 9:57 PM.

Comment: Here, DELETE FROM statement is used to delete the record from the Instructors table. WHERE clause is used to condition for deleting only those records from Instructors table whose InstructorID is not referenced in the Courses table. This is done by using the NOT IN operator and a subquery. The subquery uses a SELECT statement that selects all the InstructorID from the Course table.

Q22) Open the script named CreateGradStudents.sql that's attached above. Run this file to create a table named GradStudents. This table has the same columns as the Students table, but the StudentID column isn't defined as an identity column.

Ans:



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "CreateGradStudents.sql - IDEAPAD\MyCollege (IDEAPAD\prapt (55)) - Microsoft SQL Server Management Studio". The left pane is the Object Explorer, showing the database structure under "MyCollege". The right pane contains a query window titled "CreateGradStudent... (IDEAPAD\prapt (55))". The query is:

```
CREATE TABLE GradStudents
(
    StudentID INT PRIMARY KEY,
    LastName VARCHAR(25) NOT NULL,
    FirstName VARCHAR(25) NOT NULL,
    EnrollmentDate DATE NOT NULL,
    GraduationDate Date NULL
);
```

Below the query window, the "Messages" pane shows the output:

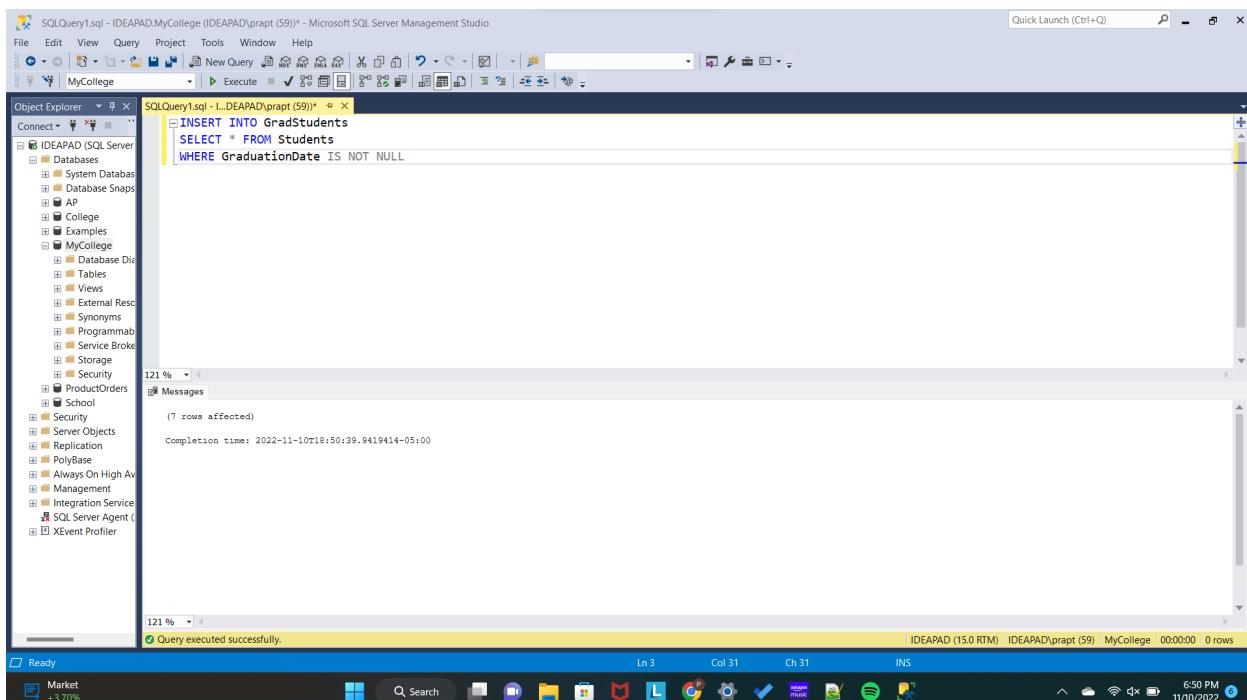
```
Commands completed successfully.
Completion time: 2022-11-07 22:14:45.2482165-05:00
```

At the bottom, the status bar indicates "Query executed successfully." and "0 rows". The taskbar at the bottom shows various application icons, and the system tray shows the date and time as "11/7/2022 10:15 PM".

Q23) Write an INSERT statement that inserts rows from the Students table into the GradStudents table. Include only the rows for students that have graduated, and don't use a column list.

Ans:

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



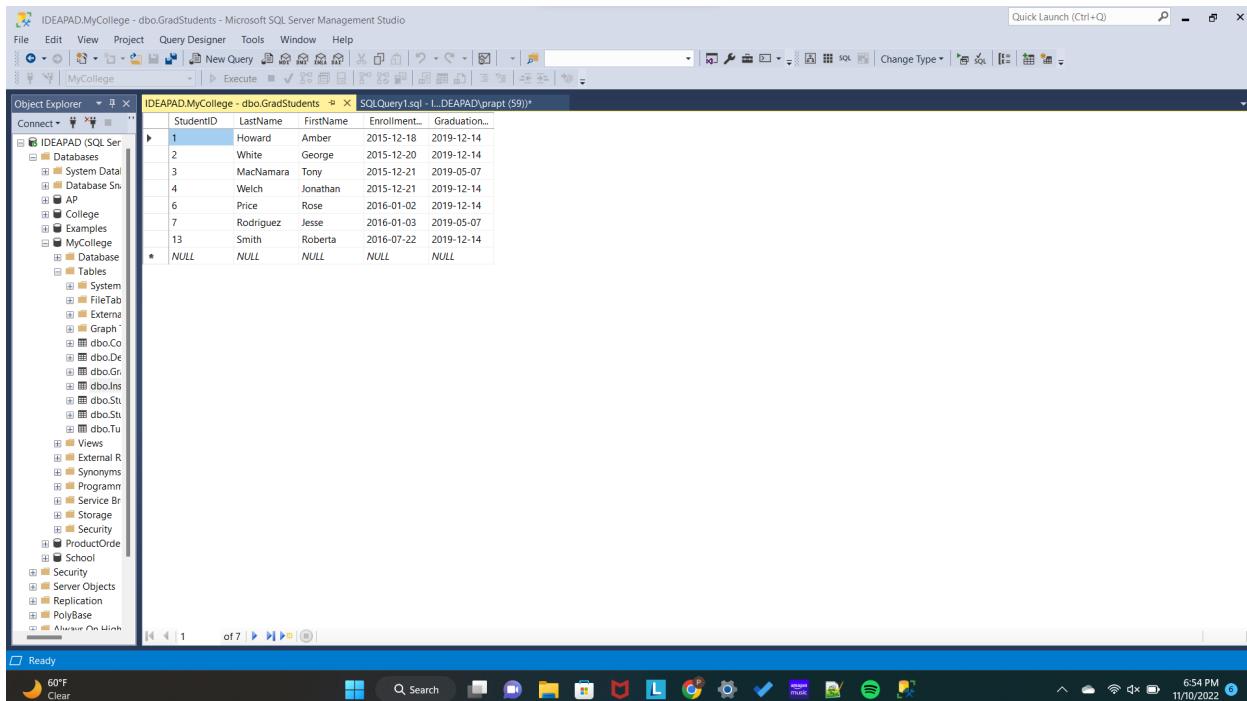
The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following SQL code:

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

The results pane shows the output of the query:

```
(7 rows affected)
Completion time: 2022-11-10T18:50:39.9419414-05:00
```

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



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'IDEAPAD.MyCollege - dbo.GradStudents' is open, displaying the following SQL code:

```
SELECT * FROM GradStudents
```

The results pane shows the data inserted into the 'GradStudents' table:

StudentID	LastName	FirstName	Enrollment...	Graduation...
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
6	Price	Rose	2016-01-02	2019-12-14
7	Rodriguez	Jesse	2016-01-03	2019-05-07
13	Smith	Roberta	2016-07-22	2019-12-14
*	NULL	NULL	NULL	NULL

Comment: Here, `INSERT INTO` statement is used to insert rows into the `GradStudents` table from the `Students` table which is done by using a `SELECT` statement followed by `INSERT INTO`. `SELECT` statement is used to select those records from the `Students` table whose `GraduationDate IS NOT NULL`.

Q24) Open the script named CreateMyCollege.sql. Then, run this script. That should restore the data that's in the database. If an error message is displayed indicating that the database is in use, you'll need to close and restart the Management Studio and then run the script again.

Ans:



```
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
CREATE TABLE [dbo].[Courses](
    [CourseID] [int] IDENTITY(1,1) NOT NULL,
    [CourseNumber] [int] NULL,
    CONSTRAINT [PK_Courses] PRIMARY KEY CLUSTERED ([CourseID])
) ON [PRIMARY]

Completion time: 2022-11-10T19:58:02.1117984-08:00
```

91 %

Messages

(1 row affected)

Completion time: 2022-11-10T19:58:02.1117984-08:00

91 %

Query executed successfully.

Comment: Close and restart the SQL Server Management Studio and open the script CreateMyCollege.sql script. By running this script the original version of the database will be restored.

Q25) Write a SELECT statement that returns these columns from the Students table:

- A column that uses the CONVERT function to return the EnrollmentDate column in this format: MM/DD/YYYY. In other words, use 2-digit months and days and a 4-digit year, and separate each date component with slashes.
 - A column that uses the CONVERT function to return the EnrollmentDate column with the date, and the hours and minutes on a 12-hour clock with an am/pm indicator.
 - A column that uses the CONVERT function to return the EnrollmentDate column with just the time in a 24-hour format, including the milliseconds.
 - A column that uses the CONVERT function to return the EnrollmentDate column with just the month and day.

Ans:

```
SELECT CONVERT(VARCHAR, EnrollmentDate,101) AS 'MM/DD/YYYY',
```

```
CONVERT(VARCHAR,EnrollmentDate,9) AS '12HourFormat',
CONVERT(VARCHAR,EnrollmentDate,114) AS '24HourFormat',
CONVERT(VARCHAR(5), EnrollmentDate,101) AS 'MonthDay'
FROM Students
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window is open with the following T-SQL code:

```
SELECT CONVERT(VARCHAR, EnrollmentDate,101) AS 'MM/DD/YYYY',
CONVERT(VARCHAR,EnrollmentDate,9) AS '12hourFormat',
CONVERT(VARCHAR,EnrollmentDate,114) AS '24HourFormat',
CONVERT(VARCHAR(5), EnrollmentDate,101) AS MonthDay
FROM Students
```

The results pane displays 20 rows of data from the Students table. The columns are labeled as specified in the query: MM/DD/YYYY, 12HourFormat, 24HourFormat, and MonthDay. The data includes various dates ranging from December 2015 to July 2017.

	MM/DD/YYYY	12HourFormat	24HourFormat	MonthDay
1	12/18/2015	Dec 18 2015 4:44:26PM	16:44:26	12/18
2	12/20/2015	Dec 20 2015 11:12:26AM	11:12:26	12/20
3	12/21/2015	Dec 21 2015 9:21:55AM	09:21:55	12/21
4	12/21/2015	Dec 21 2015 1:23:10PM	13:23:10	12/21
5	12/28/2015	Dec 28 2015 10:32:16AM	10:32:16	12/28
6	01/02/2016	Jan 2 2016 12:37:31PM	12:37:31	01/02
7	01/03/2016	Jan 3 2016 1:08:37PM	13:08:37	01/03
8	01/03/2016	Jan 3 2016 3:44:56PM	15:44:56	01/03
9	07/15/2016	Jul 15 2016 1:42:21AM	01:42:21	07/15
10	07/15/2016	Jul 15 2016 5:00:28AM	07:00:28	07/15
11	07/15/2016	Jul 15 2016 12:45:43PM	12:45:43	07/15
12	07/20/2016	Jul 20 2016 9:37:53AM	09:37:53	07/20
13	07/20/2016	Jul 22 2016 11:18:25AM	11:18:25	07/22
14	12/1/2016	Dec 10 2016 3:31:28PM	15:31:28	12/10
15	12/1/2016	Dec 12 2016 2:22:53PM	14:22:53	12/12
16	12/14/2016	Dec 14 2016 4:42:11PM	16:42:11	12/14
17	12/22/2016	Dec 22 2016 8:43:48AM	08:43:48	12/22
18	01/02/2017	Jan 2 2017 11:28:49AM	11:28:49	01/02
19	01/04/2017	Jan 4 2017 10:42:06AM	10:42:06	01/04
20	07/1/2017	Jul 12 2017 1:05:41PM	13:05:41	07/12

At the bottom of the screen, a status bar shows "Query executed successfully." and the system clock "7:29 PM 11/10/2022".

Comment: Here, SELECT statement is used to select the column EnrollmentDate and perform CONVERT() function. For the first column, CONVERT() function is used to convert the EnrollmentDate datatype to VARCHAR and format style 101 to show it in MM/DD/YYYY format. For the second column, CONVERT() function is used to convert the EnrollmentDate to VARCHAR datatype and format style 9 to show it in the 12 hour date format. For the third column, the CONVERT() function is used to convert the datatype of the EnrollmentDate column to VARCHAR and set the format style as 114 to show it in the 24 hour format. For the fourth column, CONVERT() function is used to convert the datatype of the EnrollmentDate column to VARCHAR(5) and format style 101 to show it in MM/DD/YYYY format and the VARCHAR(5) displays only MM/DD.

C. ADVANCED SQL SKILLS

1. Create a view named DepartmentInstructors that returns these columns: the DepartmentName column from the Departments table and the LastName, FirstName, Status, and AnnualSalary columns from the Instructors table.

Ans:

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

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a database structure for 'MyCollege' including 'Databases', 'Tables', 'Security', and other system objects. The central pane contains a query window titled 'SQLQuery2.sql - IDEAPAD\prapt (51)* - Microsoft SQL Server Management Studio'. The query is:

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

Below the query, the 'Messages' section shows the execution results:

```
Commands completed successfully.

Completion time: 2022-11-11T14:17:46.3229315-05:00
```

The status bar at the bottom indicates 'Query executed successfully.' and shows the completion time as 2022-11-11T14:17:46.3229315-05:00. The taskbar at the bottom of the screen shows various application icons and the date/time as 11/11/2022 2:17 PM.

Comment: Here, CREATE VIEW statement is used to create a view named DepartmentInstructors. This view returns the DepartmentName, LastName, FirstName, Status, AnnualSalary columns. This is done using a SELECT statement which retrieves these columns from the Departments and the Instructors tables using the JOIN keyword on the DepartmentID column.

2. Write a SELECT statement that returns all the columns from the DepartmentInstructors view that you created in exercise 1.

Ans:

```
SELECT * FROM DepartmentInstructors
```

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 displays the following SQL code:

```
SQLQuery2.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (51)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Connect New Query Execute
SELECT * FROM DepartmentInstructors
```

The results pane shows a table with 16 rows of data from the 'DepartmentInstructors' view. The columns are: DepartmentName, LastName, FirstName, Status, and AnnualSalary. The data includes various departments like Business, Science, English, etc., with instructors like Jones, Smith, Rogers, etc.

DepartmentName	LastName	FirstName	Status	AnnualSalary
Business	Adam	Billy	P	77500.00
Business	Thomas	William	P	38500.00
Science	Amundsen	Rachel	P	79000.00
Business	Green	Gene	P	75000.00
Mathematics	McGregor	NULL	P	74000.00
Music	Paxton	Arnold	P	36500.00
Business	Rogers	NULL	P	38000.00
Education	Smith	John	F	73000.00
Sociology	Connors	Daniel	F	71500.00
English	Jones	Sally	F	74000.00
Business	Vilma	Jonathan	P	35500.00
Music	Thomas	Derrick	P	35500.00
Education	Black	Bill	P	34000.00
Mathematics	Warren	Angela	P	33000.00
English	Drew	Daniel	F	72000.00
Science	Gallegos	Tomas	P	64000.00

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 2:21 PM.

Comment: Here, `SELECT *` statement is used to display all the details of the `DepartmentInstructors` view.

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

Ans:

```
SELECT * FROM DepartmentInstructors
WHERE DepartmentName = 'English' AND Status = 'F'
```

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 displays the following SQL code:

```
SQLQuery2.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (51)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer Connect New Query Execute
SELECT * FROM DepartmentInstructors
WHERE DepartmentName = 'English' AND Status = 'F'
```

The results pane shows a table with 2 rows of data from the 'DepartmentInstructors' view, filtered by DepartmentName = 'English' and Status = 'F'. The columns are: DepartmentName, LastName, FirstName, Status, and AnnualSalary.

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

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 2:35 PM.

Comment: Here, SELECT * statement is used to select all the details from the DepartmentInstructors view such that the DepartmentName is 'English' and the Status is 'F' using the WHERE clause.

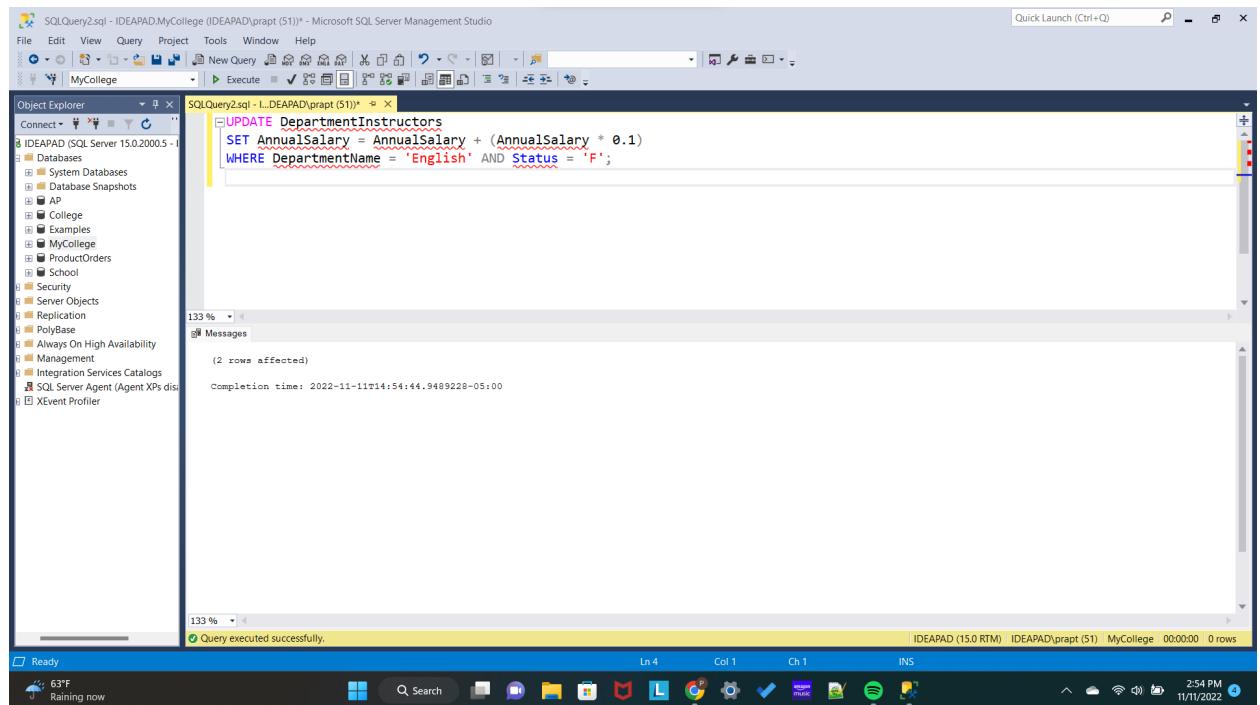
4. Write an UPDATE statement that updates the DepartmentInstructors view you created in exercise 1 so it increases the annual salary for each fulltime instructor in the English department by 10%. Then, run the SELECT statement you wrote in exercise 2 to be sure this worked correctly.

Ans:

UPDATE DepartmentInstructors

SET AnnualSalary = AnnualSalary + (AnnualSalary * 0.1)

WHERE DepartmentName = 'English' AND Status = 'F';



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery2.sql - IDEAPAD\prapt (51)* - Microsoft SQL Server Management Studio' contains the following SQL code:

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

The 'Messages' pane below the query window displays the results of the execution:

- (2 rows affected)
- Completion time: 2022-11-11T14:54:44.9489280-05:00

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

SELECT * FROM DepartmentInstructors

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'MyCollege' is selected. A query window titled 'SQLQuery2.sql - IDEAPAD.MyCollege (IDEAPAD\prapt (51)) - Microsoft SQL Server Management Studio' contains the following SQL code:

```
SELECT * FROM DepartmentInstructors
```

The results grid displays 16 rows of data from the 'DepartmentInstructors' view. The columns are: DepartmentName, LastName, FirstName, Status, and AnnualSalary. The data includes various departments like Business, Science, and English, with instructors like William, Amundsen, and Jones. The 'Status' column shows values like P and F, and the 'AnnualSalary' column shows values ranging from 35000.00 to 79200.00.

DepartmentName	LastName	FirstName	Status	AnnualSalary
Business	Thomson	Billy	P	77500.00
Business	Thomas	William	P	35000.00
Science	Amundsen	Rachel	F	79000.00
Business	Green	Gene	P	75000.00
Mathematics	McGregor	NULL	F	74000.00
Music	Paxton	Arnold	P	36000.00
Business	Rogers	NULL	P	38000.00
Education	Smith	John	F	73000.00
Sociology	Connors	Daniel	F	71500.00
English	Jones	Sally	F	81400.00
Business	Vilma	Jonathan	P	35500.00
Music	Thomas	Derrick	P	35500.00
Education	Black	Bill	P	34000.00
Mathematics	Warren	Angela	P	33000.00
English	Drew	Daniel	F	79200.00
Science	Gallegos	Tomas	F	64000.00

At the bottom of the results grid, a message says 'Query executed successfully.' The status bar at the bottom right shows 'IDEAPAD (15.0 RTM) | IDEAPAD\prapt (51) | MyCollege | 00:00:00 | 16 rows'. The taskbar at the bottom of the screen shows various application icons and the system clock.

Comment: Here, UPDATE statement is used to update the DepartmentInstructors view and increase the AnnualSalary by 10% for those records whose DepartmentName is 'English' and the Status is 'F' using the WHERE clause.

5. Create a view named StudentCoursesMin that returns these columns: the FirstName and LastName from the Students table and the CourseNumber, CourseDescription, and CourseUnits from the Courses table.

Ans:

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

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 'SQLQuery2.sql' contains the following T-SQL code:

```

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

The 'Messages' pane at the bottom of the query window displays the message: "Commands completed successfully." Below the window, the taskbar shows the system status: "Ready", "62°F Raining now", and the date/time: "3:09 PM 11/11/2022".

`SELECT * FROM StudentCoursesMin`

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 'SQLQuery2.sql' contains the following T-SQL code:

```

SELECT * FROM StudentCoursesMin
  
```

The 'Results' pane at the bottom of the query window displays the results of the query, which is a list of 21 rows from the 'StudentCoursesMin' view. The columns are: FirstName, LastName, CourseNumber, CourseDescription, and CourseUnits. The data includes various student names and course details such as 'Social Media', 'Intro to Management', 'Biology', etc.

The 'Messages' pane at the bottom of the query window displays the message: "Commands completed successfully." Below the window, the taskbar shows the system status: "Ready", "62°F Raining now", and the date/time: "3:10 PM 11/11/2022".

Comment: Here, `CREATE VIEW` statement is used to create a view named `StudentCoursesMin`. `SELECT` statement selects the `FirstName`, `LastName`, `CourseNumber`, `CourseDescription`, `CourseUnits` columns from the `Students`, `StudentCourses` and `Courses` tables by joining them using `JOIN` keyword on `StudentID` and `CourseID` columns.

6. Write a script that declares a variable and sets it to the count of all students in the Students table that haven't graduated. If the count is greater than or equal to 100, the script should display a message that says, "The number of undergrad students is greater than or equal to 100". Otherwise, it should say, "The number of undergrad students is less than 100".

Ans:

```
USE MyCollege;
DECLARE @NotGraduated int
SET @NotGraduated = (SELECT COUNT(StudentID)
                      FROM Students
                     WHERE GraduationDate IS NULL AND GraduationDate < GETDATE())
IF @NotGraduated >= 100
    PRINT 'The number of undergrad students is greater than or equal to 100';
ELSE
    PRINT 'The number of undergrad students is less than 100';
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database 'MyCollege'. In the center is the 'SQLQuery1.sql' query editor window. The code is identical to the one provided above. To the right of the editor is the 'Messages' pane, which displays the output of the query. The output shows the message 'The number of undergrad students is less than 100'. At the bottom of the screen, the taskbar shows the system status (62°F Rain coming), the date and time (7:48 PM 11/11/2022), and the SQL Server session information (IDEAPAD (15.0 RTM) IDEAPAD\prapt (61) MyCollege 00:00:00 0 rows).

Comment: Here, USE MyCollege is used to use the database MyCollege i.e, perform the query in MyCollege database. DECLARE statement is used to declare a variable after (@) sign. Datatype of that variable is also declared. SELECT statement is used to assign a value to the variable declared using the DECLARE statement. Here, a variable named NotGraduated is declared using the DECLARE statement with int datatype. The count of the Students from the Students table where the GraduationDate IS NULL is obtained by the COUNT() aggregate function and the WHERE clause, is assigned to the variable.

using SELECT statement. IF...ELSE is used to print the messages. If the NotGraduated is greater than or equal to 100 then print 'The number of undergrad students is greater than or equal to 100' this message. Else print 'The number of undergrad students is less than 100' this message using PRINT.

7. Write a script that uses two variables to store (1) the count of all of the instructors in the Instructors table and (2) the average annual salary for those instructors. If the instructor count is greater than or equal to 10, the script should print a message that displays the values of both variables. Otherwise, the script should print a message that says, "The number of fulltime instructors is less than 10".

Ans:

```
USE MyCollege;
DECLARE @NoOfInstructors int;
DECLARE @AvgAnnualSalary Money;
SET @NoOfInstructors = (SELECT COUNT(InstructionID)
                        FROM Instructors);
SET @AvgAnnualSalary = (SELECT AVG(AnnualSalary)
                        FROM Instructors);

IF @NoOfInstructors >= 10
PRINT      'The      number      of      Instructors      =      '      +
CONVERT(VARCHAR,@NoOfInstructors, 3) + ' and their Average Annual Salary = ' +
CONVERT(VARCHAR, @NoOfInstructors, 4);
ELSE
PRINT 'The number of fulltime instructors is less than 10';
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'MyCollege' is selected. A query window titled 'SQLQuery2.sql' is open, displaying the following T-SQL script:

```

USE MyCollege;
DECLARE @NoofInstructors int;
DECLARE @AvgAnnualSalary Money;
SET @NoofInstructors = (SELECT COUNT(InstructionID)
                        FROM Instructors);
SET @AvgAnnualSalary = (SELECT AVG(AnnualSalary)
                        FROM Instructors);
IF @NoofInstructors >= 10
    PRINT 'The number of Instructors = ' + CONVERT(VARCHAR, @NoofInstructors, 3) + ' and their Average Annual Salary = ' + CONVERT(VARCHAR, @AvgAnnualSalary, 4);
ELSE
    PRINT 'The number of fulltime instructors is less than 10';

```

The 'Messages' pane at the bottom shows the output of the query:

```

The number of Instructors = 16 and their Average Annual Salary = 16
Completion time: 2022-11-11T18:08:22.18932055-05:00

```

The status bar at the bottom right indicates 'Query executed successfully.' and shows the system time as 6:08 PM on 11/11/2022.

Comment: Here, USE MyCollege is used to use the database MyCollege I.e, perform the query in MyCollege database. DECLARE statement is used to declare a variable after (@) sign. Datatype of that variable is also declared. SELECT statement is used to assign a value to the variable declared using the DECLARE statement. Here, a variable named NoOfInstructors is declared using the DECLARE statement with int datatype and another variable AvgAnnualSalary with money datatype. The count of the InstructorID column from the Instructors table is calculated using COUNT() aggregate function in a SELECT statement and is assigned to NoOfInstructors. The average of the AnnualSalary is calculated from the Instructors table using the AVG() aggregate function in a SELECT statement and is assigned to the AvgAnnualSalary. IF..ELSE is used to print the messages. If the NoOfInstructors variable is greater than or equal to 10 then print a message that displays the value of both the variables using the CONVERT() function. Else print 'The number of fulltime instructors is less than 10' this message.

8. Write a script that attempts to delete the department with the name 'Sociology' from the Departments table. If the delete is successful, the script should display this message:

SUCCESS: Record was deleted.

If the delete is unsuccessful, the script should display a message something like this:

FAILURE: Record was not deleted.

Error 547: The DELETE statement conflicted with the REFERENCE constraint "FK_Instructor_Depart_267ABA7A". The conflict occurred in database "MurachCollege", table "dbo.Instructors", column 'DepartmentID'.

Ans:

```
BEGIN TRY
BEGIN TRANSACTION
DELETE FROM Departments
WHERE DepartmentName ='Sociology'
PRINT 'SUCCESS: Record was deleted'
COMMIT TRANSACTION
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION
PRINT 'FAILURE: Record was not deleted.'
PRINT 'Error 547: '+ERROR_MESSAGE()
END CATCH
```

Comment: Here, BEGIN TRY...END TRY block is used for the try block. BEGIN TRANSACTION...COMMIT TRANSACTION block is used to start a transaction and commit the transaction if it finishes without any errors. Thus, DELETE FROM statement is written inside this block so that when the DepartmentName is 'Sociology' records are deleted successfully then a message is printed 'SUCCESS: Record was deleted' and that transaction is committed. If error occurs while deleting then it goes in the BEGIN CATCH...END CATCH block where the transaction can be rolled back to its previous state and a message 'FAILURE: Record was not deleted.' is printed followed by the ERROR_MESSAGE() which is used to return the error message relevant to the error generated.

9. Write a script that creates and calls a function named fnStudentUnits that calculates the total course units of a student in the StudentCourses table. To do that, this function should accept one parameter for the student ID, and it should return the sum of the course units for the student.

Ans:

```
CREATE FUNCTION fnStudentUnits
(@StudentID int)
RETURNS int
AS
BEGIN
DECLARE @NoOfUnits int
SET @NoOfUnits = 0
SELECT @NoOfUnits = SUM(CourseUnits)
FROM StudentCourses JOIN Students
ON Students.StudentID = StudentCourses.StudentID
JOIN Courses
ON Courses.CourseID = StudentCourses.CourseID
RETURN @NoOfUnits
END
GO
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with a tree view of the database structure, including databases like IDEAPAD and MyCollege, and various system objects. The right pane contains a query window titled 'SQLQuery1.sql - IDEAPAD\prapt (61)* - Microsoft SQL Server Management Studio'. The query itself is the T-SQL code provided above, which creates a function named 'fnStudentUnits' that takes a parameter '@StudentID' and returns an integer. It uses a SELECT statement with a SUM aggregate function to calculate the total 'CourseUnits' for a given student. Below the query window, the 'Messages' pane shows the output: 'Commands completed successfully.' and 'Completion time: 2022-11-11T19:58:20.1329568-05:00'. At the bottom of the screen, the Windows taskbar is visible with various icons and the system clock showing '7:58 PM 11/11/2022'.

```
CREATE FUNCTION fnStudentUnits
(@StudentID int)
RETURNS int
AS
BEGIN
DECLARE @NoOfUnits int
SET @NoOfUnits = 0
SELECT @NoOfUnits = SUM(CourseUnits)
FROM StudentCourses JOIN Students
ON Students.StudentID = StudentCourses.StudentID
JOIN Courses
ON Courses.CourseID = StudentCourses.CourseID
RETURN @NoOfUnits
END
GO
```

Comment: Here, CREATE FUNCTION is used to create a function named fnStudentUnits with a parameter @StudentID of int type and this function returns an int value. A variable @NoOfUnits is declared with datatype int and is set to the value 0. SELECT statement is used to select the records whose NoOfUnits is sum of CourseUnits that is done using the SUM() aggregate function from StudentCourses and Students table on StudetID and Courses table on CourseID using the JOIN keyword to join three tables.

10. The SELECT statement that calls the function should 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.

Ans:

```
SELECT StudentID, c.CourseID, CourseUnits,
MyCollege.dbo.fnStudentUnits (StudentID) AS 'NoOfUnits'
FROM StudentCourses sc JOIN Courses c
ON c.CourseID = sc.CourseID;
```

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 'SQLQuery1.sql' contains the following T-SQL code:

```
SELECT StudentID, c.CourseID, CourseUnits,
MyCollege.dbo.fnStudentUnits (StudentID) AS 'NoOfUnits'
FROM StudentCourses sc JOIN Courses c
ON c.CourseID = sc.CourseID;
```

Below the query window is a results grid titled 'Results'. The grid displays the following data:

	StudentID	CourseID	CourseUnits	NoOfUnits
1	5	10	3	298
2	5	12	3	298
3	5	15	4	298
4	5	21	3	298
5	8	1	3	298
6	8	7	3	298
7	8	19	3	298
8	9	6	3	298
9	9	8	3	298
10	9	10	3	298
11	10	2	3	298
12	10	24	4	298
13	11	7	3	298
14	11	13	3	298
15	11	14	2	298
16	13	1	3	298
17	13	7	3	298
18	13	12	3	298
19	13	6	3	298

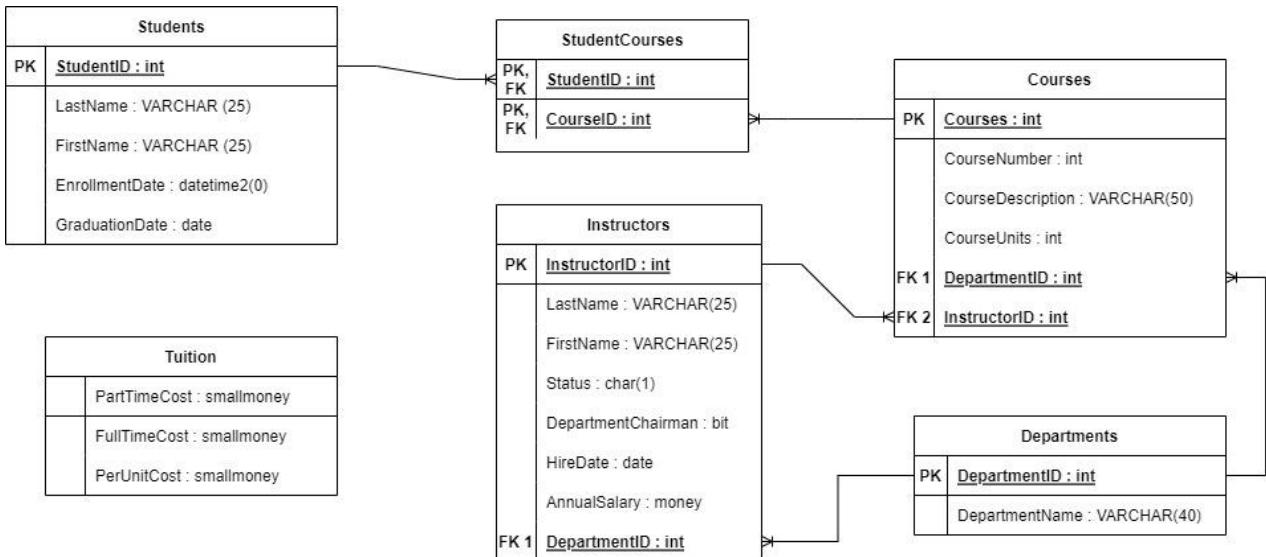
At the bottom of the results grid, it says 'Query executed successfully.'

Comment: Here, SELECT statement is used to select the StudentID, CourseID, CourseUnits and the StudentID returned by the function fnStudentUnits from the StudentCourses and Courses table using JOIN keyword on the CourseID column.

D. DATABASE DESIGN

1. Create a database diagram that shows the relationships between the six tables in the MyCollege database. You can use any digital tool to draw your diagrams. "Resources and Links" tab on this Blackboard page lists several tools for drawing E/R diagrams.

Ans:

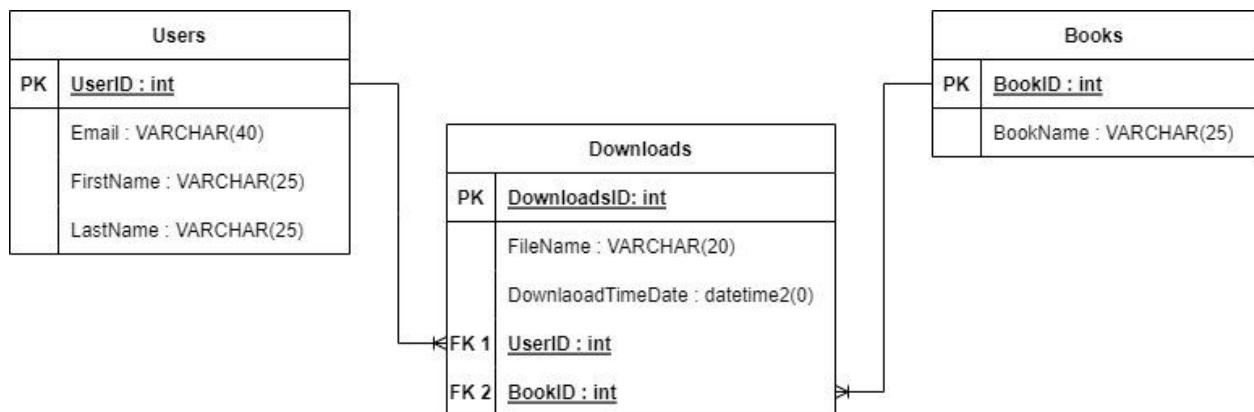


Comment:

- The Students table has five columns. StudentID column is of type int and is a primary key. LastName and FirstName columns are of type varchar(25). EnrollmentDate column is of type datetime2(0). GraduationDate column is of type date.
- The Courses table has six columns. CourseID column is of type int and is a primary key. CourseNumber column is of type int. CourseDescription column is of type varchar(50). CourseUnits column is of type int. DepartmentID and InstructorID columns are foreign keys taken as references from Departments and Instructors tables respectively and are of the type int.
- The StudentCourses table has two columns. StudentID and CourseID columns are foreign keys taken as references from Students and Courses tables respectively and are of the type int.
- The Instructors table has eight columns. InstructorID is a primary key of the type int. LastName and FirstName columns are of the type varchar(25). Status column is of the type char(1). DepartmentChairman column is of the type bit. HireDate column is of the type date. AnnualSalary column is of type money. DepartmentID column is of the type int and is a foreign key taken as reference from the Departments table.
- The Departments table has two columns. DepartmentID is of type int and is a primary key. DepartmentName column is of type varchar(40).

- The Tuition table has three columns. PartTimeCost, FullTimeCost and PerUnitCost columns are of the type smallmoney.
 - There is one to many relationship between : Students and StudentCourses tables, Courses and StudentCourses tables, Departments and Courses tables, Instructors and Courses tables, Departments and Instructors tables.
2. Design a database diagram for a database that stores information about the downloads that users make from a book website.
- Each user must have an email address, first name, and last name.
 - Each user can have one or more downloads.
 - Each download must have a filename and download date/time.
 - Each book can be related to one or more downloads.
 - Each book must have a name

Ans:



Comment:

- The Users table has four columns. UserID column is a primary key and is of the type int. Email column is of the type varchar(40). FirstName and LastName columns are of the type varchar(25)
- The Downloads table has five columns. DownloadsID column is a primary key and is of the type int. FileName column is of the type varchar(20). DownloadTimeDate is of the type datetime2(0). UserID and BookID columns are of the type int and are foreign keys taken as reference from the Users and Books table respectively.
- The Books table has two columns. BookID column is a primary key and is of the type int. BookName column is of the type varchar(25).
- There is a one to many relationship between: Users and Downloads tables, Books and Downloads tables.

REMARKS

- This lab contains all the topics taught till now in class. It covers most of the concepts and gives a lot of different problems to solve.
- Topics covered are: basic queries, keywords, statements, operators, joins, subqueries, summary queries, aggregate functions, datatypes, tables, aliasing, insertion, deletion, update, views, Scripts, scalar and table valued functions, errors, try...catch, database designing.
- Topics related to views, scripts, errors take time and efforts to solve as compared to other topics.