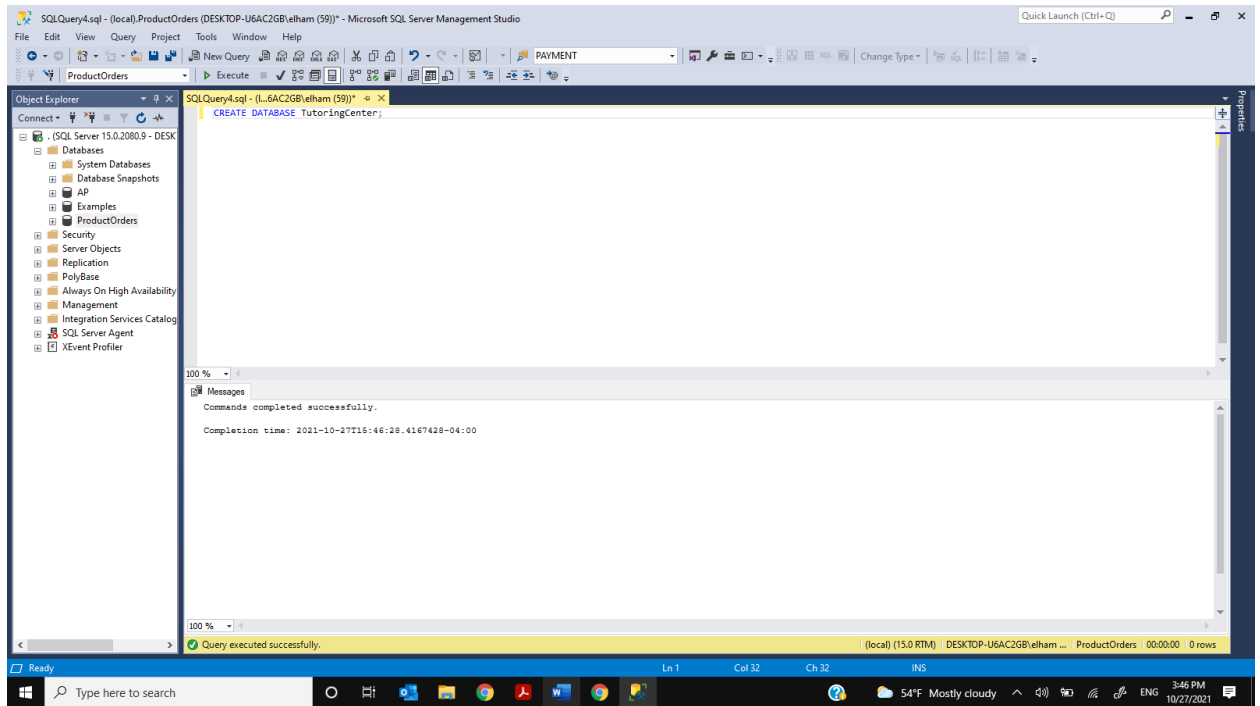


1. You are asked to create a new database named TutoringCenter.

```
CREATE DATABASE TutoringCenter;
```



2. (1) Describe the relationship type shown in figure (one-to-one, one-to-many or many-to-many). (2) Write the CREATE TABLE statements needed to implement the following design in the TutoringCenter database. Include foreign key constraints. Define StudentID and CourseID as identity columns. Decide which columns should allow null values, if any, and explain your decision. Define the CoursePrice column with a default of zero and a check constraint to allow only positive values.



The relationship is many to many because each student could take many courses and each course can be assigned to many students as well.

StudentID, Name, Phone cannot be null but we may not need their email and address

CourseID, Title, Time and room cannot be null but we may not need the price

In StudyGroups table StudentId and CourseId cannot be null because each student should be assigned to a course

```

USE TutoringCenter;
CREATE TABLE Students
(
    StudentId INT PRIMARY KEY IDENTITY,
    NAME NVARCHAR(100) NOT NULL,
    Phone CHAR(10) NOT NULL,
    Email NVARCHAR(100) NULL,
    Address NVARCHAR(200) NULL
)
GO
CREATE TABLE Courses
(
    CourseId INT PRIMARY KEY IDENTITY,
    Title NVARCHAR(50) NOT NULL,
    Price MONEY DEFAULT 0 NULL,
    CHECK (Price >= 0),
    [Time] TIME NOT NULL,
    Room CHAR(5) NOT NULL
)
GO
CREATE TABLE StudyGroups
(
    StudentId INT FOREIGN KEY REFERENCES Students(StudentId),
    CourseId INT FOREIGN KEY REFERENCES Courses(CourseId),
    PRIMARY KEY (StudentId , CourseId)
)
  
```

SQLQuery4.sql - (local).TutoringCenter (DESKTOP-U6AC2GB\elham (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

SQLQuery4.sql - (local).TutoringCenter

Object Explorer

- SQL Server 15.0.2080.9 - DESKTOP-U6AC2GB\elham (59)
- Databases
 - System Databases
 - Database Snapshots
 - AP
 - Examples
 - ProductOrders
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Always On High Availability
 - Management
 - Integration Services Catalog
 - SQL Server Agent
 - XEvent Profiler

SQLQuery4.sql - (local).TutoringCenter

```
USE TutoringCenter;
CREATE TABLE Students
(
    StudentId INT PRIMARY KEY IDENTITY,
    NAME NVARCHAR(100) NOT NULL,
    Phone CHAR(10) NOT NULL,
    Email NVARCHAR(100) NULL,
    Address NVARCHAR(200) NULL
)
GO
CREATE TABLE Courses
(
    CourseId INT PRIMARY KEY IDENTITY,
    Title NVARCHAR(50) NOT NULL,
    Price MONEY DEFAULT 0 NULL,
    CHECK (Price >= 0),
    [Time] TIME NOT NULL,
    Room CHAR(5) NOT NULL
)
GO
CREATE TABLE StudyGroups
(
    StudentId INT FOREIGN KEY REFERENCES Students(StudentId),
    CourseId INT FOREIGN KEY REFERENCES Courses(CourseId),
    PRIMARY KEY (StudentId, CourseId)
)
```

Messages

Commands completed successfully.

Completion time: 2021-10-27T15:51:15.2169897-04:00

Query executed successfully.

(local) (15.0 RTM) | DESKTOP-U6AC2GB\elham (59) | TutoringCenter 00:00:00 0 rows

Ready

Type here to search

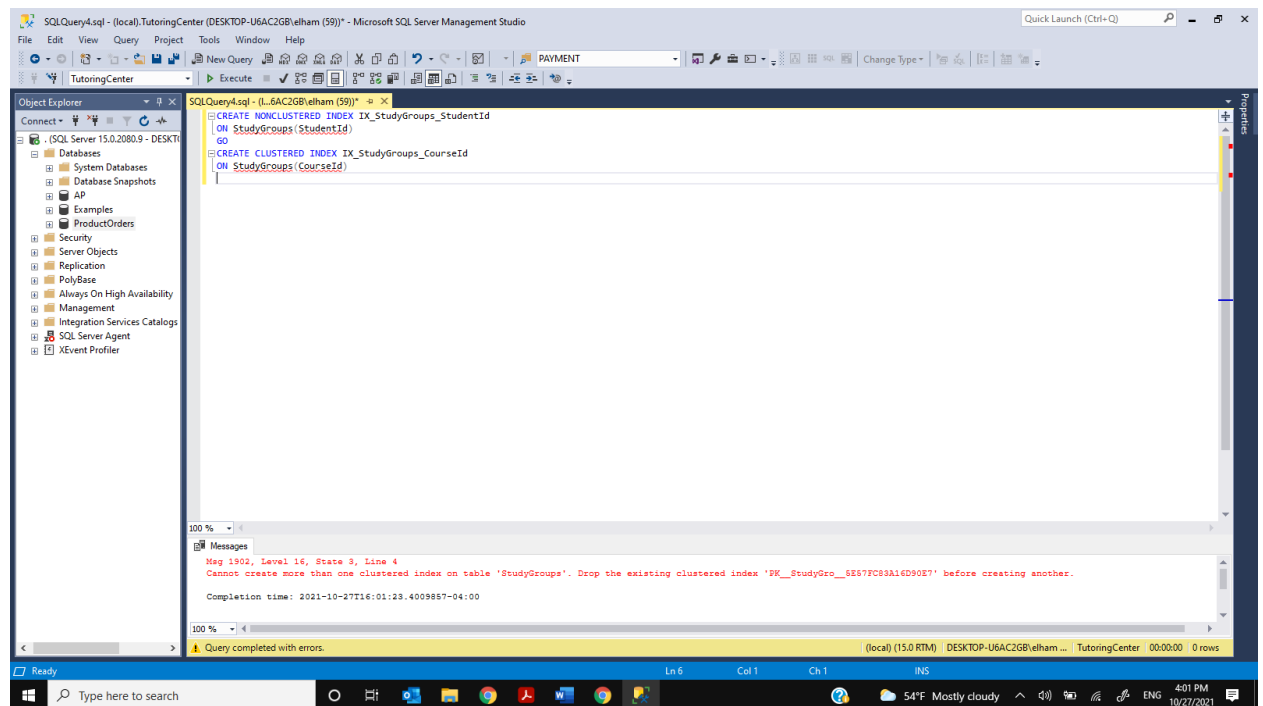
Ln 25 Col 39 Ch 36 INS

S4°F Mostly cloudy 3:51 PM 10/27/2021

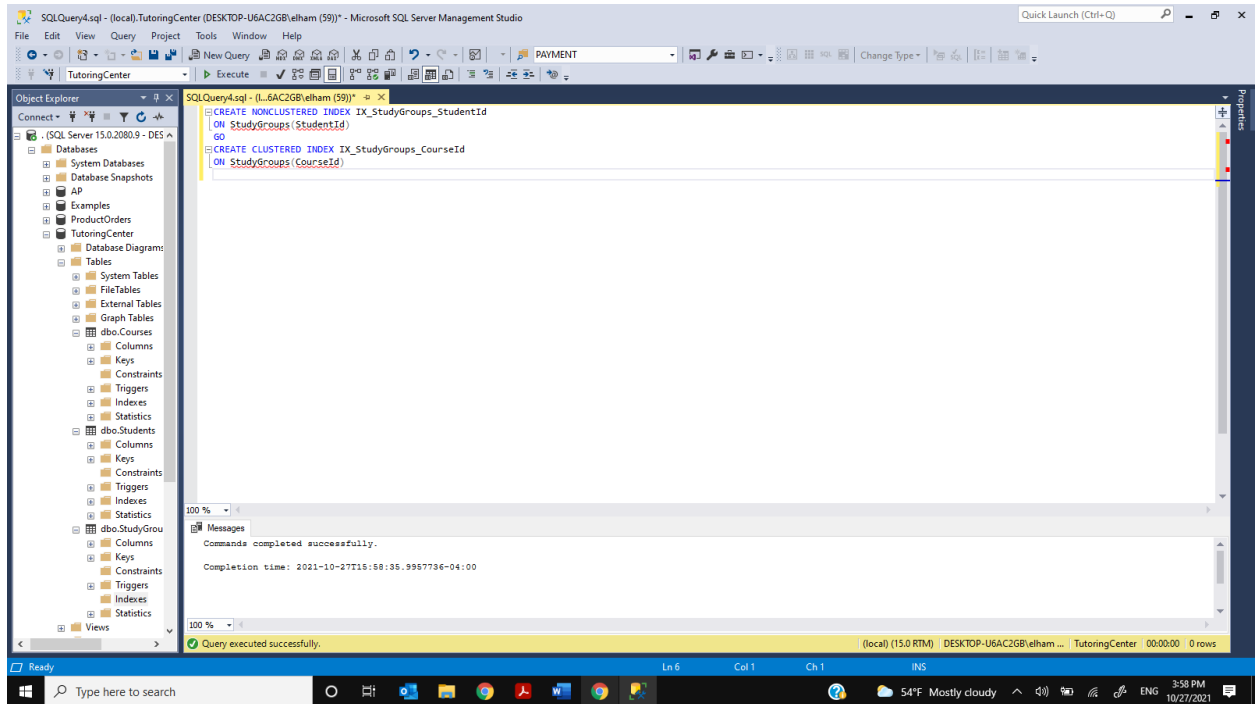
3. Write the CREATE INDEX statements to create a nonclustered index on the StudentID column and a clustered index on the CourseID column of the StudyGroups table.

```
CREATE NONCLUSTERED INDEX IX_StudyGroups_StudentId
ON StudyGroups(StudentId)
GO
CREATE CLUSTERED INDEX IX_StudyGroups_CourseId
ON StudyGroups(CourseId)
```

The second statement will result in an error because we can only have one clustered index on table and CourseId is already a part of primary key which is by default clustered index

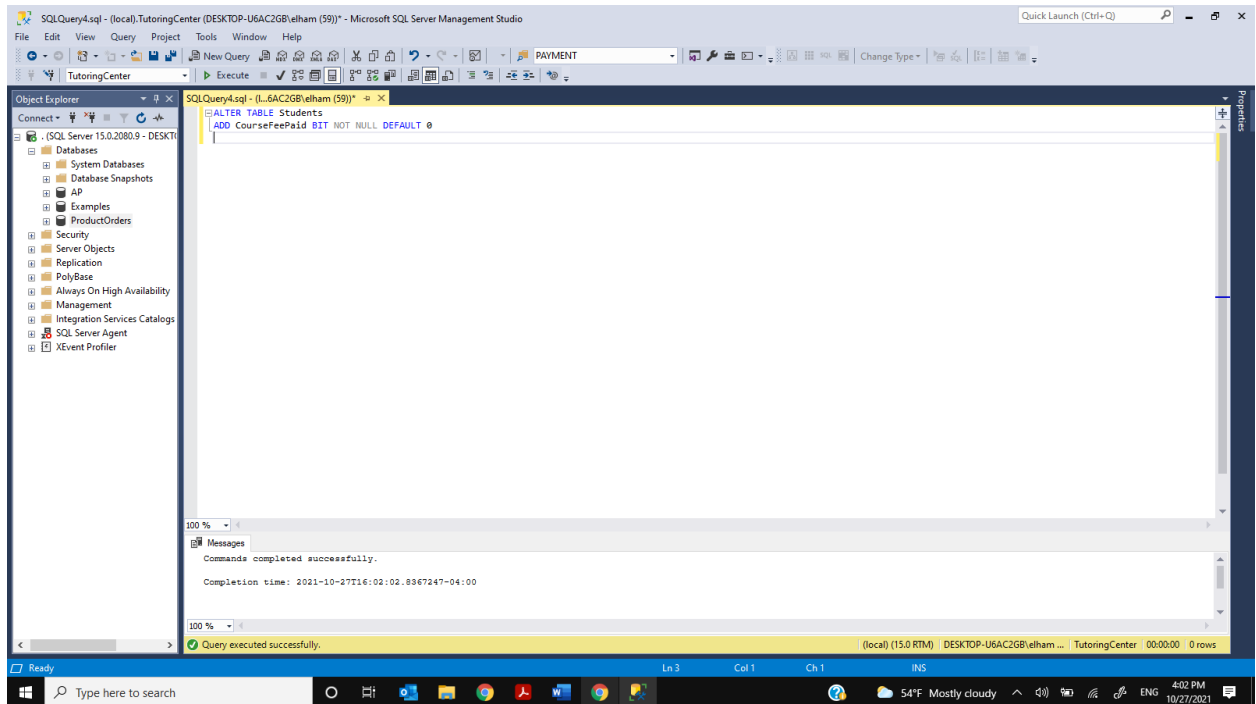


But when we delete the current Index:



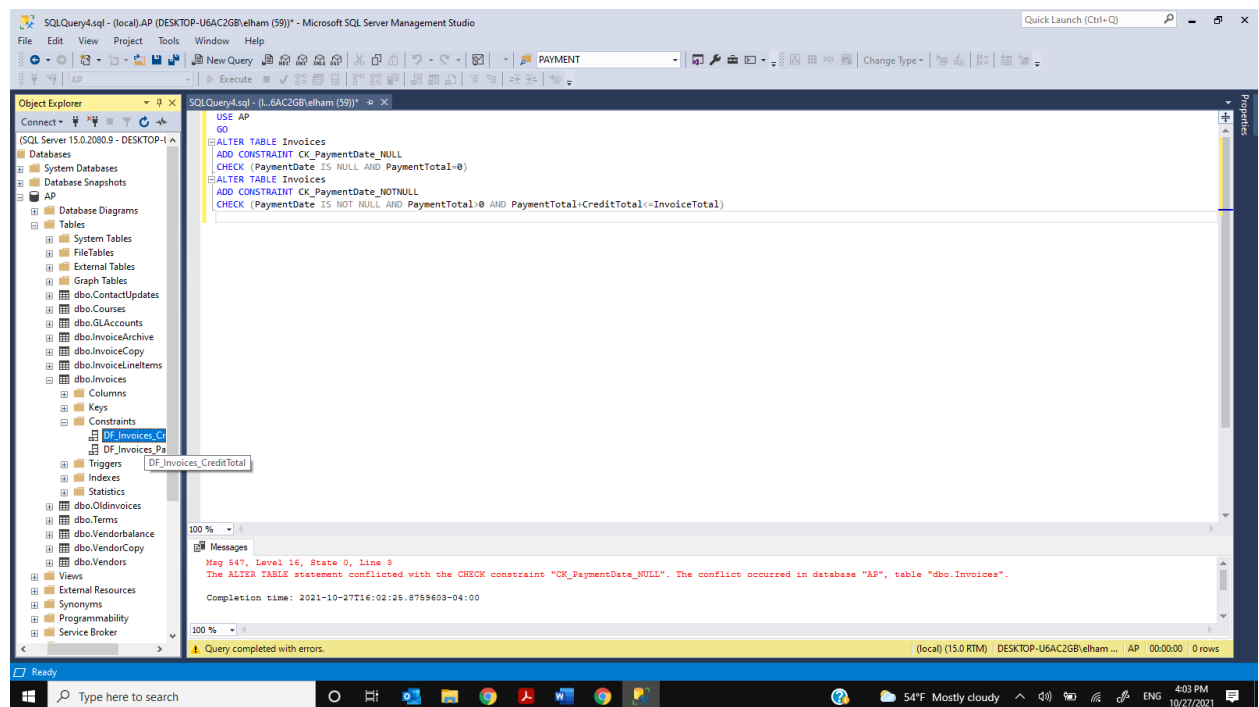
4. Write an ALTER TABLE statement that adds a new column, CourseFeePaid, to the Students table. Use the bit data type, disallow null values, and assign a default Boolean value of False.

```
ALTER TABLE Students  
ADD CourseFeePaid BIT NOT NULL DEFAULT 0
```



5. Write an ALTER TABLE statement that adds two new check constraints to the Invoices table (in AP database) of the AP database. The first should allow (1) PaymentDate to be null only if PaymentTotal is zero and (2) PaymentDate to be not null only if PaymentTotal is greater than zero. The second constraint should prevent the sum of PaymentTotal and CreditTotal from being greater than InvoiceTotal.

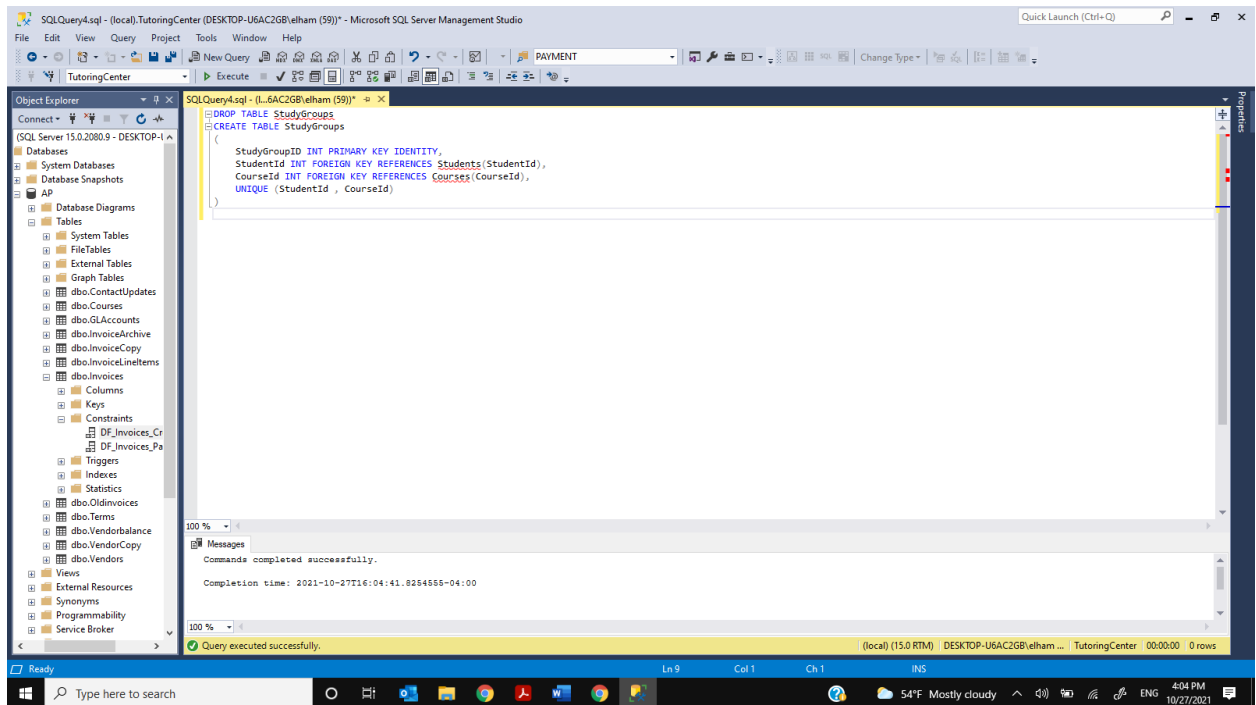
```
USE AP
GO
ALTER TABLE Invoices
ADD CONSTRAINT CK_PaymentDate_NULL
CHECK (PaymentDate IS NULL AND PaymentTotal=0)
ALTER TABLE Invoices
ADD CONSTRAINT CK_PaymentDate_NOTNULL
CHECK (PaymentDate IS NOT NULL AND PaymentTotal>0 AND PaymentTotal+CreditTotal<=InvoiceTotal)
```



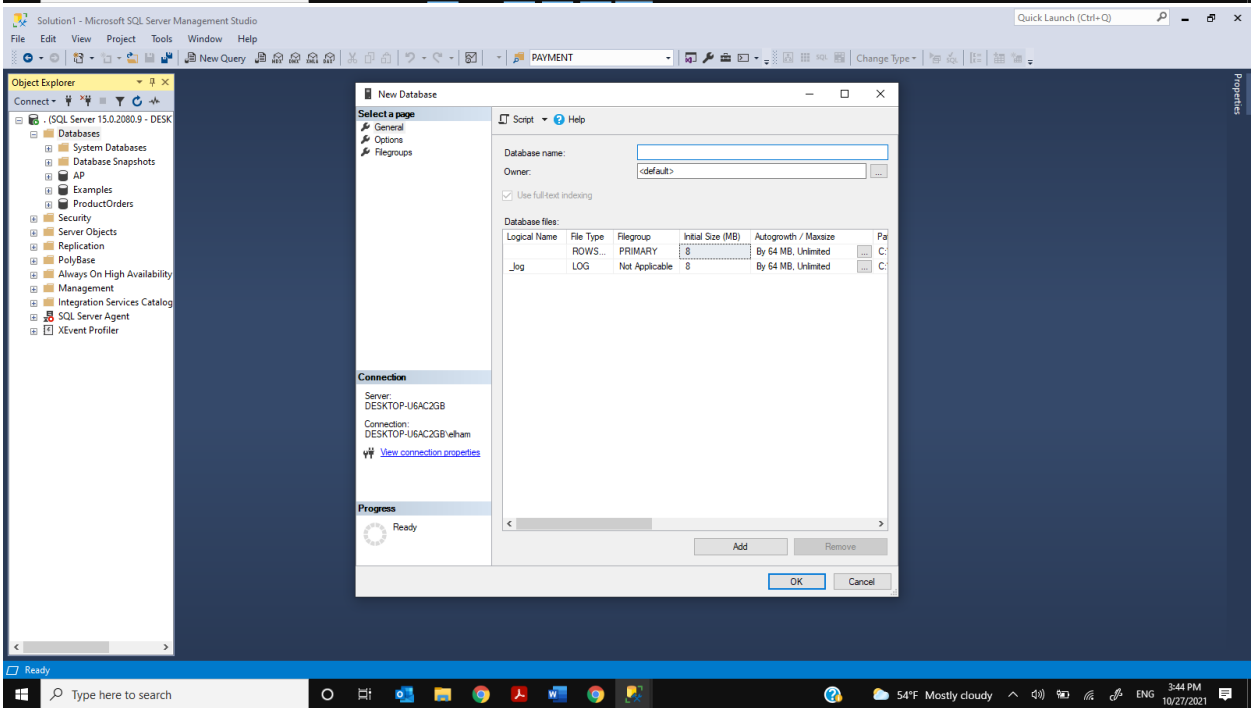
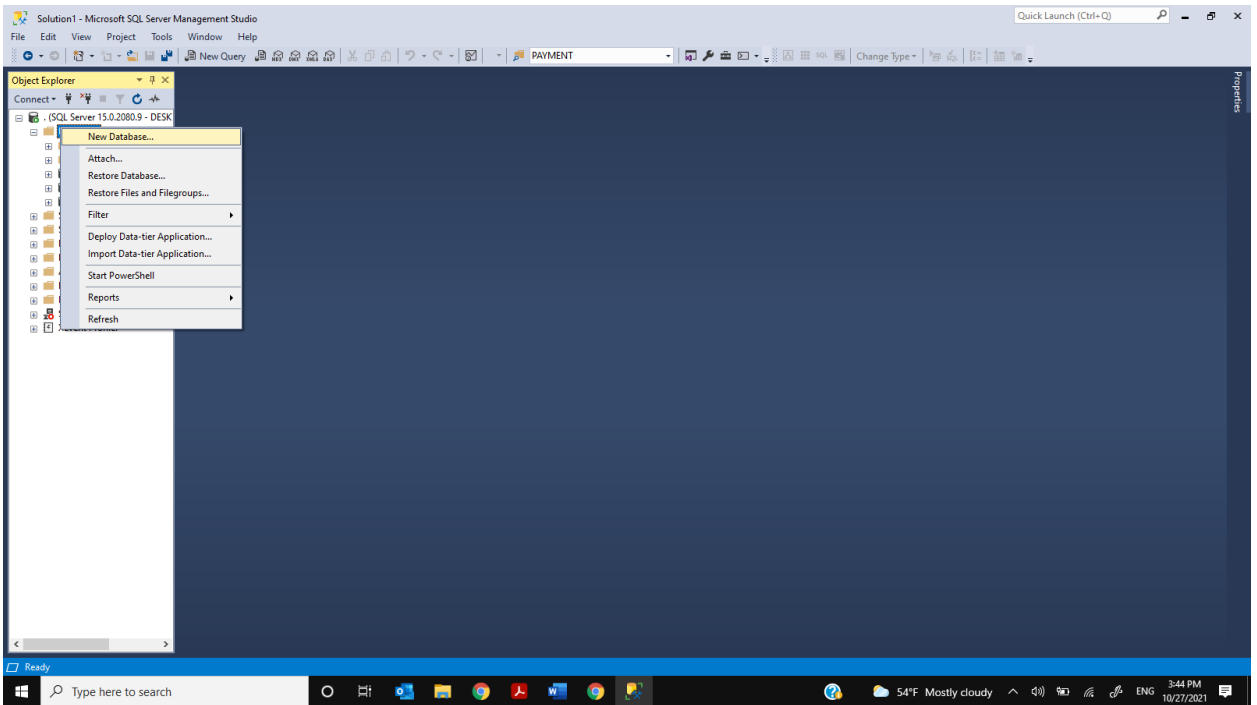
6. Delete the StudyGroups table from the TutoringCenter database. Then, write a CREATE TABLE statement that recreates the table, this time with a unique constraint that prevents a student from being a study-group member in the same course twice.

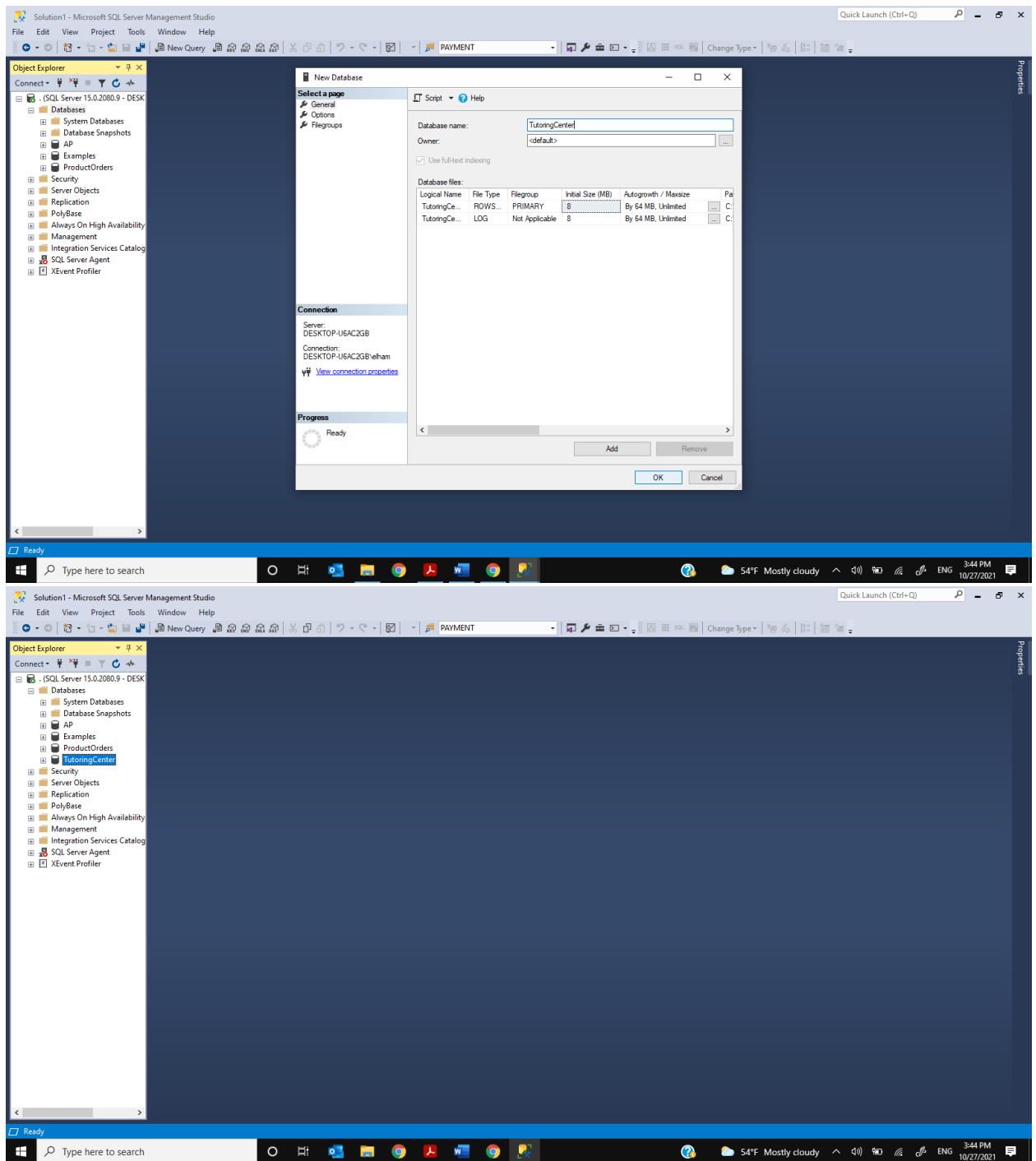
DROP TABLE StudyGroups
CREATE TABLE StudyGroups

```
(  
    StudyGroupID INT PRIMARY KEY IDENTITY,  
    StudentId INT FOREIGN KEY REFERENCES Students(StudentId),  
    CourseId INT FOREIGN KEY REFERENCES Courses(CourseId),  
    UNIQUE (StudentId , CourseId)  
)
```



7. Use the Management Studio to create a new database called TutoringCenter using the default settings. (Do not use SQL query to do this).





Thank you Devendar

Sincerely,

Sayed