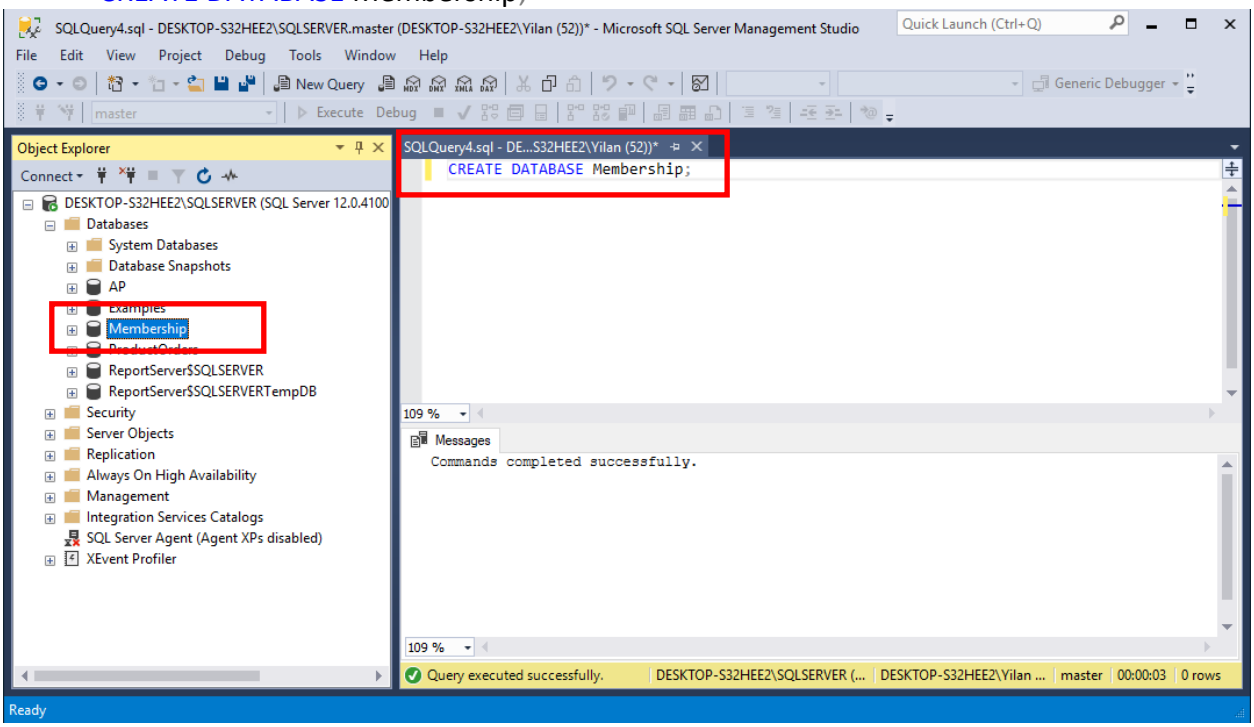
Lab 7: Database Implementation Solution

1. Create a new database named Membership.

```
CREATE DATABASE Membership;
```

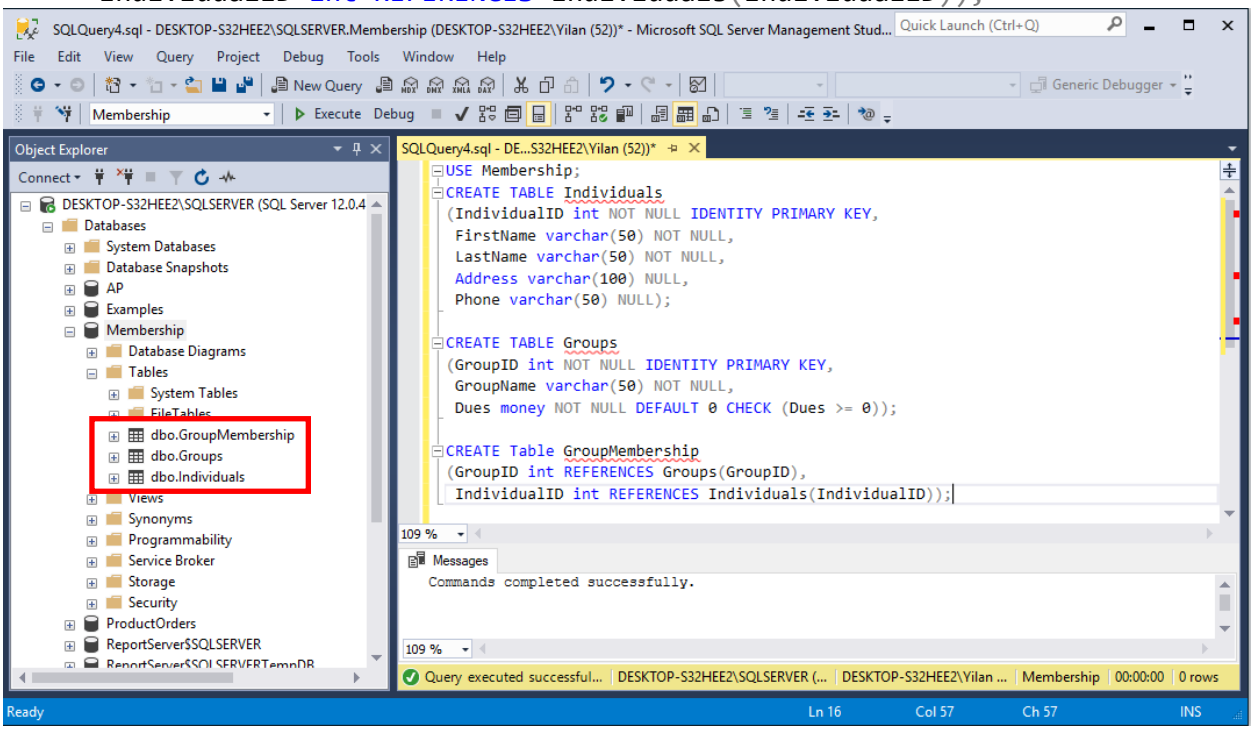


2. (1) Describe the relationship type shown in figure.

It is a many-to-many relationship between Individuals and Groups tables. For each individual, each individual can join multiple groups and for each group, and each group includes many different individuals. GroupMembership is the linking table.

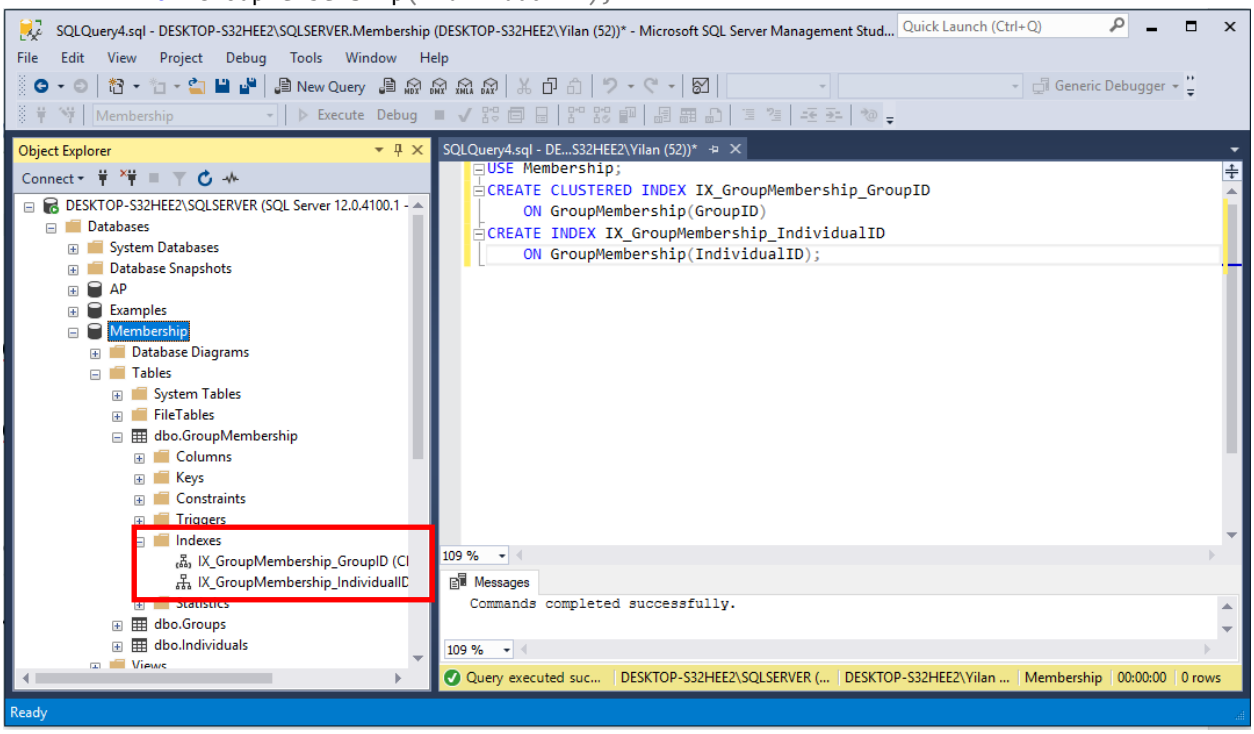
(2) Write the CREATE TABLE statements needed to implement the following design in the Membership database. Include foreign key constraints. Define IndividualID and GroupID as identity columns. Decide which columns should allow null values, if any, and explain your decision. Define the Dues column with a default of zero and a check constraint to allow only positive values.

```
USE Membership;
CREATE TABLE Individuals
(IndividualID int NOT NULL IDENTITY PRIMARY KEY,
 FirstName varchar(50) NOT NULL,
 LastName varchar(50) NOT NULL,
 Address varchar(100) NULL,
 Phone varchar(50) NULL);

CREATE TABLE Groups
(GroupID int NOT NULL IDENTITY PRIMARY KEY,
 GroupName varchar(50) NOT NULL,
 Dues money NOT NULL DEFAULT 0 CHECK (Dues >= 0));

CREATE Table GroupMembership
(GroupID int REFERENCES Groups(GroupID),
IndividualID int REFERENCES Individuals(IndividualID));
```



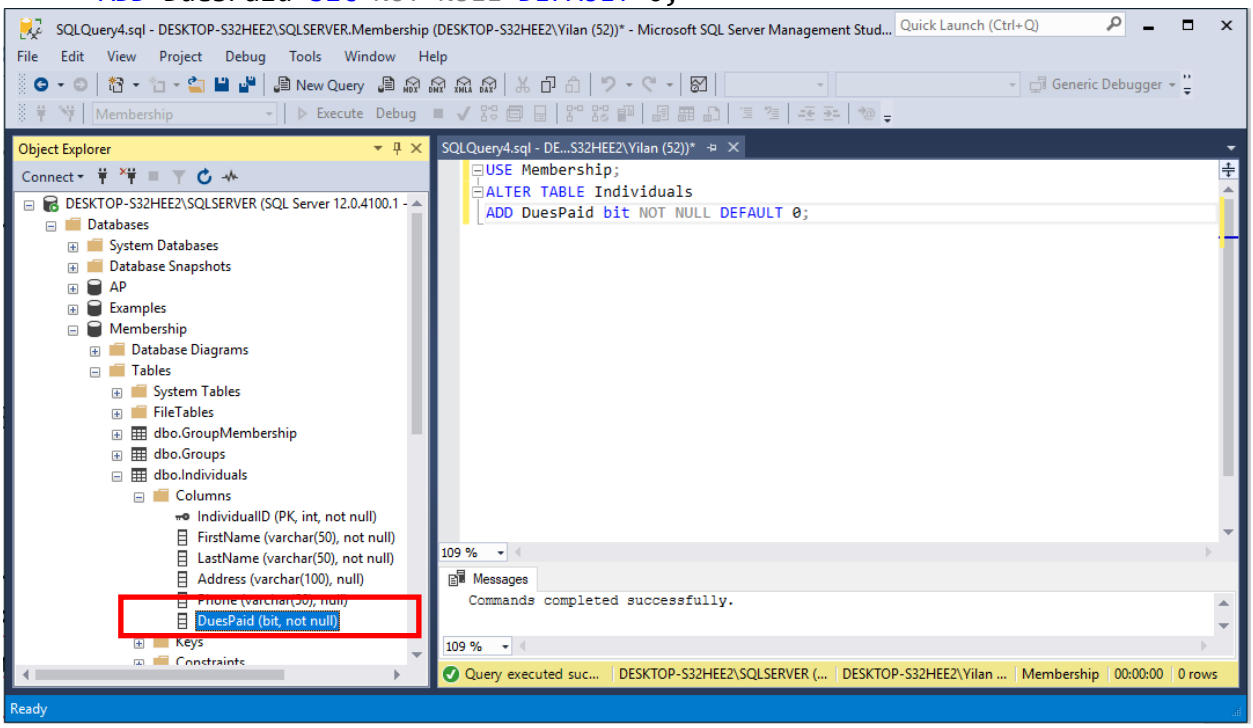
3. Write the CREATE INDEX statements to create a clustered index on the GroupID column and a nonclustered index on the IndividualID column of the GroupMembership table.

```
USE Membership;
CREATE CLUSTERED INDEX IX_GroupMembership_GroupID
    ON GroupMembership(GroupID)
CREATE INDEX IX_GroupMembership_IndividualID
    ON GroupMembership(IndividualID);
```



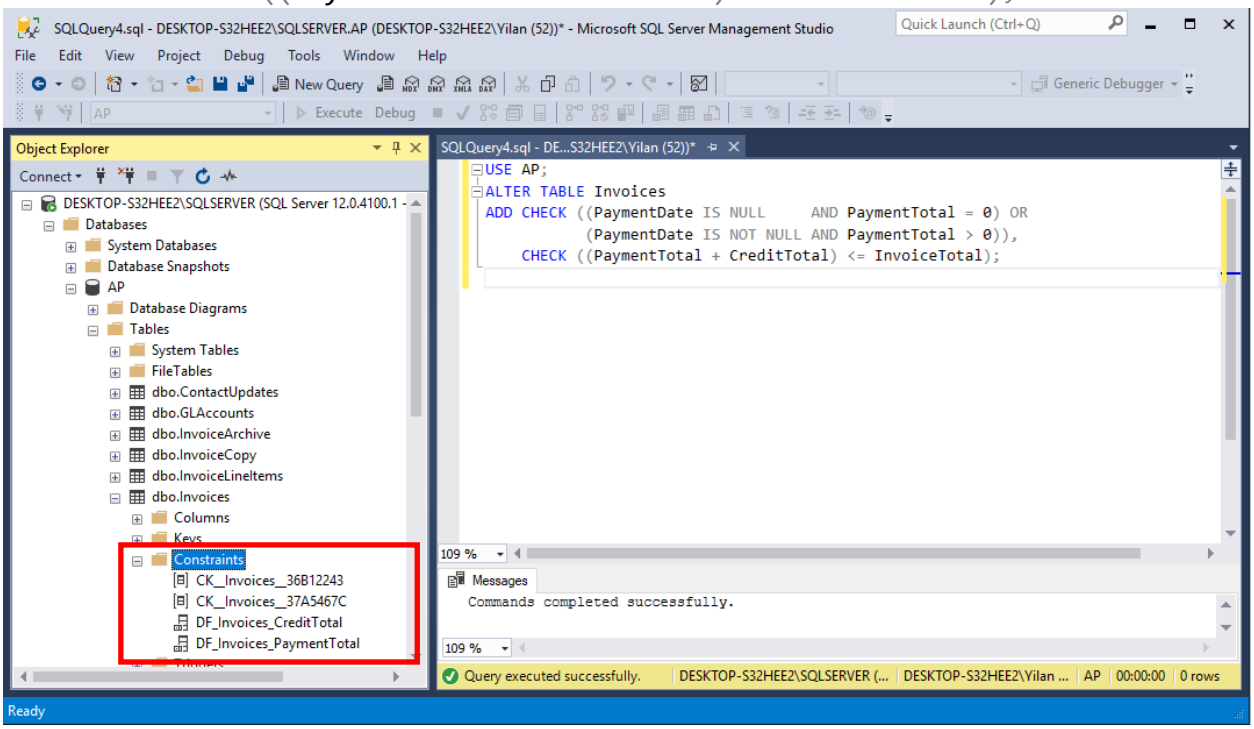
4. Write an ATER TABLE statement that adds a new column, DuesPaid, to the Individuals table. Use the bit data type, disallow null values, and assign a default Boolean value of False.

```
USE Membership;
ALTER TABLE Individuals
ADD DuesPaid bit NOT NULL DEFAULT 0;
```



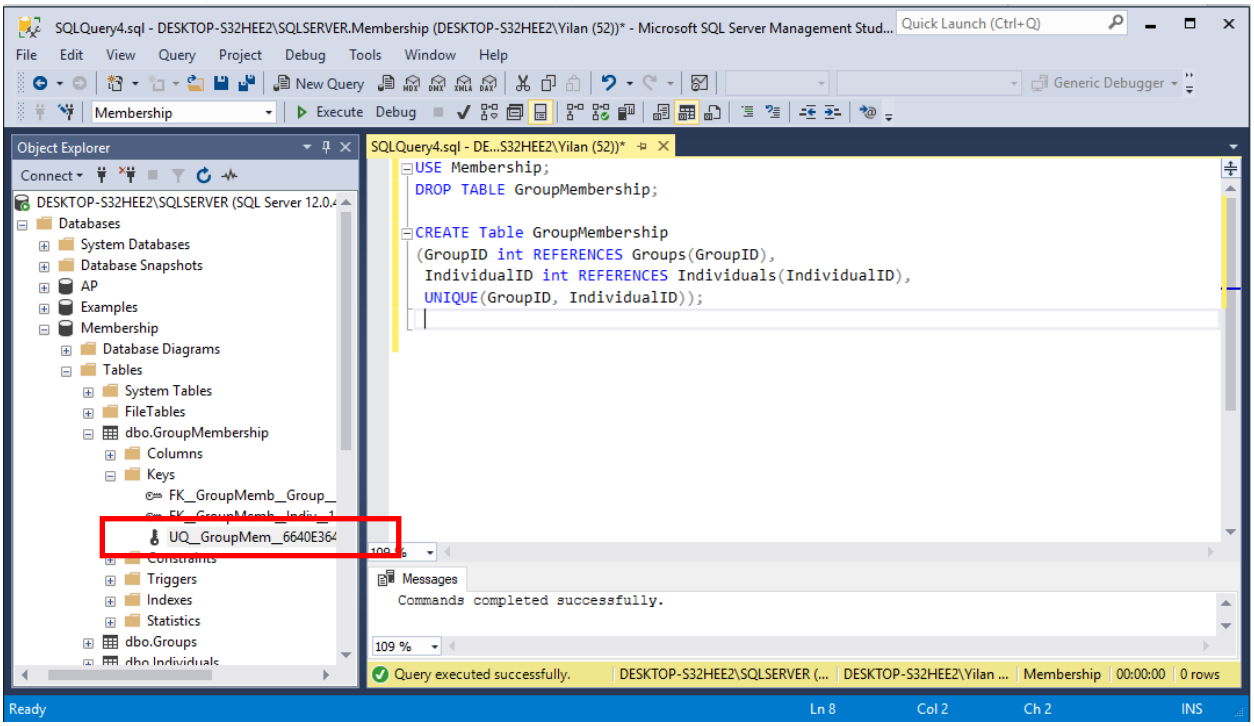
5. Write an ALTER TABLE statement that adds two new check constraints to the Invoices table (in AP database) of the AP database. The first should allow (1) PaymentDate to be null only if PaymentTotal is zero and (2) PaymentDate to be not null only if PaymentTotal is greater than zero. The second constraint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

```
USE AP;
ALTER TABLE Invoices
ADD CHECK ((PaymentDate IS NULL     AND PaymentTotal = 0) OR
           (PaymentDate IS NOT NULL AND PaymentTotal > 0)),
    CHECK ((PaymentTotal + CreditTotal) <= InvoiceTotal);
```



6. Delete the GroupMembership table from the Membership database. Then, write a CREATE TABLE statement that recreates the table, this time with a unique constraint that prevents an individual from being a member in the same group twice.

```
USE Membership;
DROP TABLE GroupMembership;

CREATE Table GroupMembership
(GroupID int REFERENCES Groups(GroupID),
 IndividualID int REFERENCES Individuals(IndividualID),
UNIQUE(GroupID, IndividualID));
```



7. Use the Management Studio to create a new database called Membership2 using the default settings.