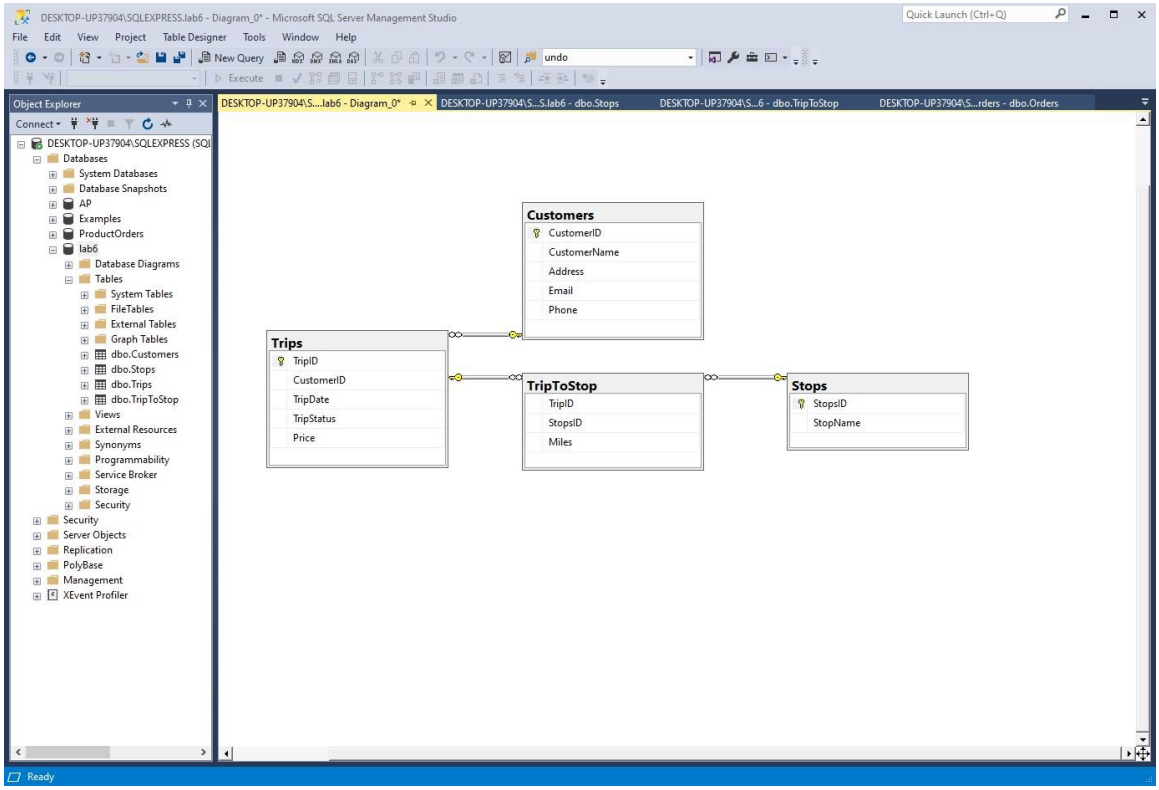
1. I created this database diagram according to the requirements. The tables are not in the 3NF since many columns can be split or put into another table. This is just a demonstration of the relationship between the tables. This applies for the next few questions as well.

Table Name	Customers	Trips	Stops	TripToStop
Primary Key	CustomerID	TripID	StopID	N/A
Foreign Key	N/A	CustomerID	N/A	TripID & StopID

An individual customer can have multiple trips so it is a one-to-many relationship.

In this part, I have not yet included the "car pool" option, where a trip can have multiple different customers. In the Trips table, its foreign key is CustomerID.

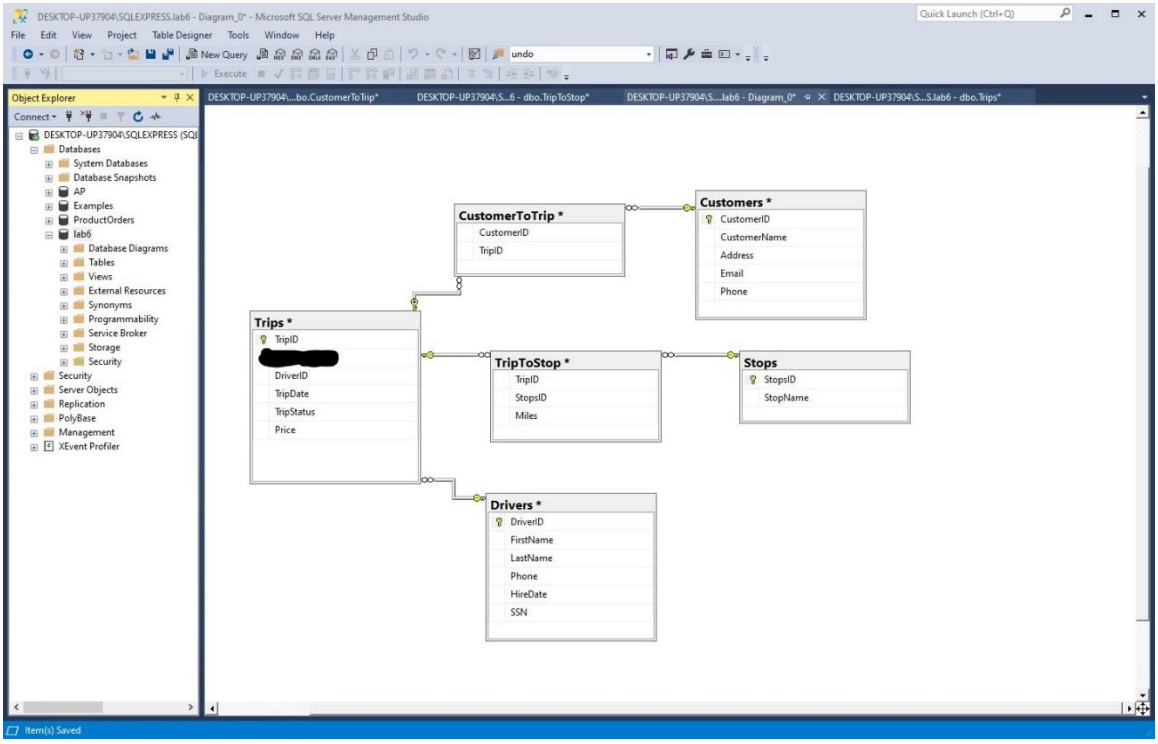
In each trip, there can be multiple stops, and for a stop, it can be in multiple different trips, so it is a many-to-many relationship. Then we need a linking table "TripToStop" to link the two. The TripToStop table have both foreign keys of TripID and StopID



2. With the ‘Car Pool’ options included; a trip can now have multiple customers. The relationship between Customers and Trips becomes many-to-many. The foreign key in the Trip table is now removed (was CustomerID previously). The linking table called CustomerToTrip is created with foreign keys: TripID and CustomerID.

   With a new Drivers table included, its primary key is DriverID. Since a driver can have multiple trips but a trip can have only one driver, then this is a one-to-many relationship. The Trips table now have a foreign key of DriverID.

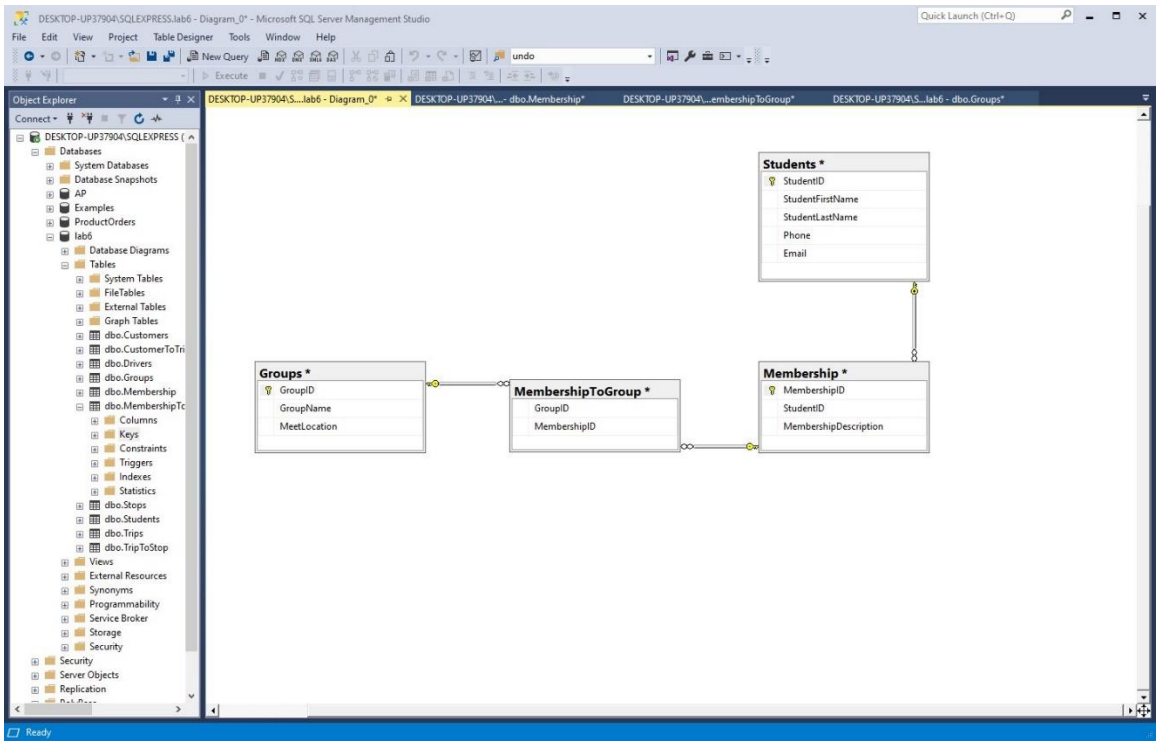
Table Name	Customers	Trips	Stops	TripToStop	Drivers	CsutomerToTrip
Primary Key	CustomerID	TripID	StopID	N/A	DriverID	N/A
Foreign Key	N/A	DriverID	N/A	TripID & StopID	N/A	TripID & CustomerID



3. A student can have multiple memberships, so one-to-many relationship.
   Each membership can be used for multiple groups and each group can have multiple memberships, so many-to-many relationship with linking table MemebershipToGroup.
   You may assume membership as a 'role'. Then a student can be the same or different role in different groups.
   The primary key for Students is StudentID.
   The primary key for Groups is GroupID.
   Membership has a foreign key of StudentID and its own primary key MembershipID.
   The linking table has foreign keys MembershipID and GroupID.

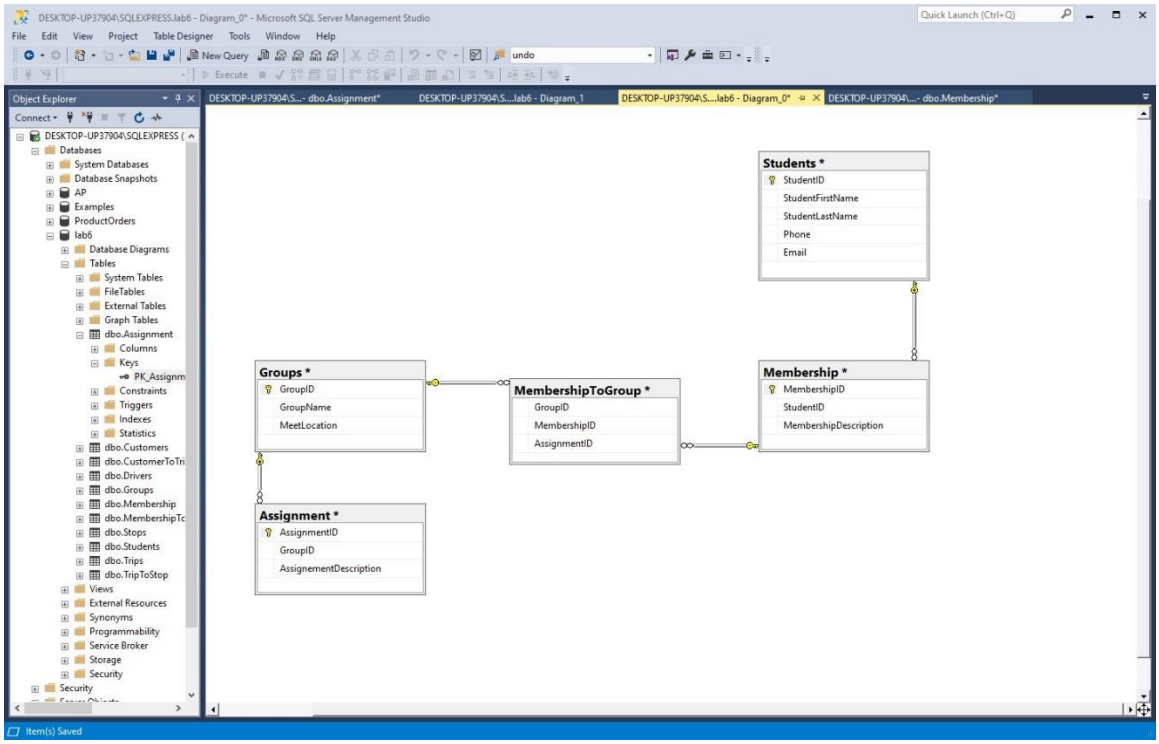
   I may have overcomplicated the question, so I can also have a student to group as many-to-many relationship while not creating the Membership table.

Table Name	Students	Membership	Groups	MemebershipToGroup
Primary Key	StudentID	MembershipID	GroupID	N/A
Foreign Key	N/A	StudentID	N/A	MembershipID & GroupID



4. A group may have many assignments. I created an Assignment table with its foreign key GroupID, and its own primary key AssignmentID. This is a one-to-many relationship. However, each student can only have one unique assignment, so I added the AssigmentID column to the linking table. This way it is impossible for a student to have multiple assignments in a group.

Table Name	Students	Membership	Groups	MemebershipToGroup	Assignments
Primary Key	StudentID	MembershipID	GroupID	N/A	AssignmentID
Foreign Key	N/A	StudentID	N/A	MembershipID & GroupID	GroupID



Remarks

In this lab we learned how tables are related when we convert a real world example into a database system. I think it's a very good practice for the upcoming project. The next step would be getting the tables into 3NF and have more details(tables and columns).