Solution Lab 2 – on DataGridView

SQL Server:

-- 1-n: Groups-Students

create table Groups(
Gid int primary key,
NoOfStudents int,
LeaderName varchar(50))

create table Students(
Sid int primary key identity,
SName varchar(50),
SSurname varchar(50),
Gender varchar(20),
Dob date,
Gid int foreign key references Groups(Gid))

insert into Groups values (923, 28, 'Andrei'), (925, 29, 'Alina'), (921, 30, 'Dan'), (927, 29, 'Mihaela')
insert into Students values ('Paul', 'Damian', 'male', '12/11/2000', 923), ('Paula', 'Maria', 'female', '02/01/2001', 923), ('Cristi', 'Suciu', 'male', '07/11/2000', 927), ('Daniel', 'Irina', 'female', '04/01/2000', 925), ('Claudiu', 'Mihalache', 'male', '07/11/2001', 921)

select * from Groups
select * from Students

	Gid	NoOfStudents	LeaderName
1	921	30	Dan
2	923	28	Andrei
3	925	29	Alina
4	927	29	Mihaela

	Sid	SName	SSurname	Gender	Dob	Gid
1	5	Paul	Damian	male	2000-12-11	923
2	6	Paula	Maria	female	2001-02-01	923
3	7	Cristi	Suciu	male	2000-07-11	927
4	8	Daniel	Irina	female	2000-04-01	925
5	9	Claudiu	Mihalache	male	2001-07-11	921

Groups
- Gid
- NoOfStudents
- LeaderName

Students
- Sid
- SName
- SSurname
- Gender
- Dob
- Gid

-- 1-n: CoursesTypes-Courses

create table CoursesTypes(
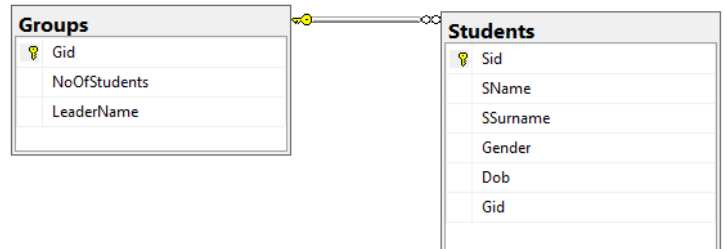Tid int primary key identity,
Type varchar(50),
Specialization varchar(50))

create table Courses(
Cid varchar(20) primary key,
CName varchar(50),
Credits int,
Semester int,
Tid int foreign key references CoursesTypes(Tid))

insert into CoursesTypes values ('mandatory', 'CS'), ('mandatory', 'MCS'), ('optional', 'CS'), ('optional', 'IR')
insert into Courses values ('MO9893', 'Databases', 6, 3, 1), ('MJ0574', 'Database Management Systems', 4, 4, 1), ('MO9783', 'SGBD', 4, 4, 4), ('MI6793', 'DB', 6, 3, 3)
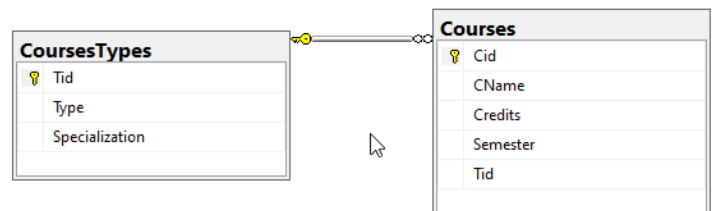
select * from CoursesTypes
select * from Courses

	Tid	Type	Specialization
1	1	mandatory	CS
2	2	mandatory	MCS
3	3	optional	CS
4	4	optional	IR

	Cid	CName	Credits	Semester	Tid
1	MI6793	DB	6	3	3
2	MJ0574	Database Management Systems	4	4	1
3	MO9783	SGBD	4	4	4
4	MO9893	Databases	6	3	1

CoursesTypes
- Tid
- Type
- Specialization

Courses
- Cid
- CName
- Credits
- Semester
- Tid

This is just an example. If you choose this option to solve your problem, you must explain every step that is perform in DataGridView. Also, you should be able to make one modification in the child table (INSERT, UPDATE, DELETE) and then save it; BUT also, to make multiple modifications in the child table (for example, 2 insert's, 3 update's, 1 delete) and then save them.

Application:
Form1.cs [Design]
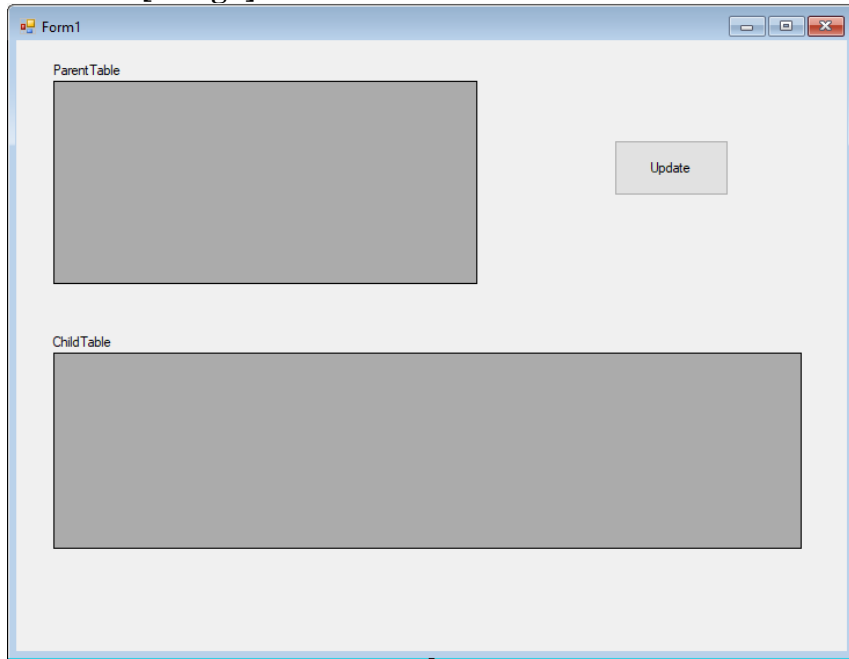
App.config - prepare the both senarios (one active and one commented)

```xml
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
 <connectionStrings>
   <add      name="cn"      connectionString="Data      Source=DESKTOP-VB4PGBM\SQLEXPRESS;Initial
Catalog=Lab1C;Integrated Security=True" providerName="System.Data.SqlClient" />
 </connectionStrings>

 <startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
 </startup>

      <appSettings>
            <add key="parentTable" value="Groups"/>
            <add key="parentPrimaryKey" value="Gid"/>
            <add key="childTable" value="Students"/>
            <add key="childForeignKey" value="Gid"/>
            <add key="server" value="DESKTOP-VB4PGBM\SQLEXPRESS"/>
            <add key="database" value="Lab2_CS_DGV"/>
      </appSettings>

      <!--
      <appSettings>
            <add key="parentTable" value="CoursesTypes"/>
```
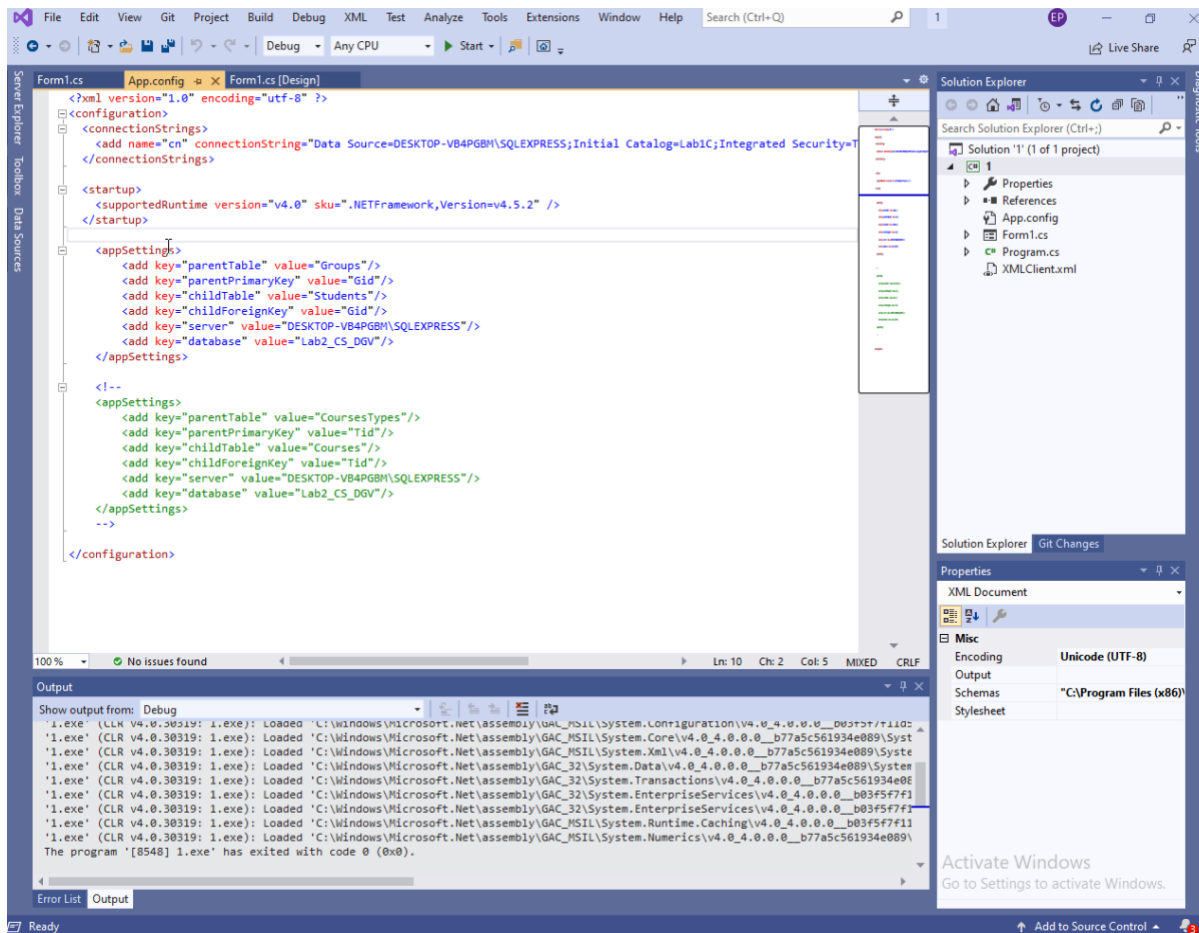
```xml
                <add key="parentPrimaryKey" value="Tid"/>
                <add key="childTable" value="Courses"/>
                <add key="childForeignKey" value="Tid"/>
                <add key="server" value="DESKTOP-VB4PGBM\SQLEXPRESS"/>
                <add key="database" value="Lab2_CS_DGV"/>
        </appSettings>
-->
</configuration>
```


Form1.cs

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Configuration;
using System.Collections.Specialized;
using System.Xml;
```

```
namespace _1
{
  public partial class Form1 : Form
  {
    public static string server = ConfigurationManager.AppSettings.Get("server");
    public static string database = ConfigurationManager.AppSettings.Get("database");
    public static string parentTable = ConfigurationManager.AppSettings.Get("parentTable");
    public static string childTable = ConfigurationManager.AppSettings.Get("childTable");
    public static string parentPrimaryKey = ConfigurationManager.AppSettings.Get("parentPrimaryKey");
    public static string childForeignKey = ConfigurationManager.AppSettings.Get("childForeignKey");

    //    SqlConnection    sqlConnection    =    new    sqlConnection(@"Data    Source=DESKTOP-
VB4PGBM\SQLEXPRESS; Initial Catalog=Lab2_CS_DGV; Integrated Security=True");
    SqlConnection sqlConnection = new SqlConnection("Data Source=" + server + ";Database=" + database +
";Integrated Security=SSPI");

    DataSet dataSet = new DataSet();

    SqlDataAdapter parentDataAdapter = new SqlDataAdapter();
    SqlDataAdapter childDataAdapter = new SqlDataAdapter();

    BindingSource parentBindingSource = new BindingSource();
    BindingSource childBindingSource = new BindingSource();

    SqlCommandBuilder parentBuilder = new SqlCommandBuilder();
    SqlCommandBuilder childBuilder = new SqlCommandBuilder();

    public Form1()
    {
      InitializeComponent();
    }

    public void populate()
    {
      parentDataAdapter = new SqlDataAdapter("SELECT * FROM " + parentTable, sqlConnection);
      childDataAdapter = new SqlDataAdapter("SELECT * FROM " + childTable, sqlConnection);

      SqlCommandBuilder parentBuilder = new SqlCommandBuilder(parentDataAdapter);
      SqlCommandBuilder childBuilder = new SqlCommandBuilder(childDataAdapter);

      // dataSet.Clear();
      parentDataAdapter.Fill(dataSet, parentTable);
      childDataAdapter.Fill(dataSet, childTable);

      DataColumn parentPK = dataSet.Tables[parentTable].Columns[parentPrimaryKey];
      DataColumn childFK = dataSet.Tables[childTable].Columns[childForeignKey];

      DataRelation relation = new DataRelation("fk_parent_child", parentPK, childFK);
      dataSet.Relations.Add(relation);
```

```
        parentBindingSource.DataSource = dataSet;
        parentBindingSource.DataMember = parentTable;

        childBindingSource.DataSource = parentBindingSource;
        childBindingSource.DataMember = "fk_parent_child";

        parentDGV.DataSource = parentBindingSource;
        childDGV.DataSource = childBindingSource;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        populate();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            parentDataAdapter.Update(dataSet, parentTable);
            childDataAdapter.Update(dataSet, childTable);
            MessageBox.Show("Updated with succes!");
            // populate();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
            sqlConnection.Close();
        }
    }
}
}
```

Execution:
Relation 1:

Add

Update

Delete

Relation 2:

Add

Update

Delete