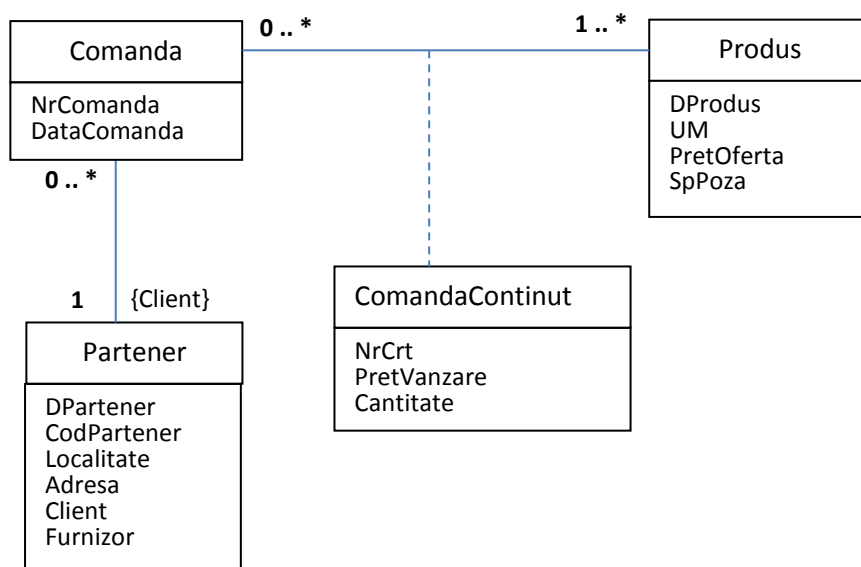


## TEHNICI DE PROGRAMARE AVANSATA – LABORATOR 12

### Un model de vizualizare si actualizare tabele asociere

Consideram diagrama de clase:



Schema bazei de date

Comenzi(IdComanda, NrComanda, DataComanda, IdClient)

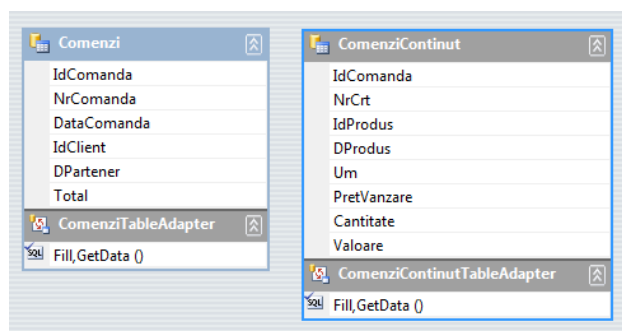
Parteneri(IdPartener, DPartener, CodPartener, Localitate, Adresa, Client, Furnizor))

Produse(IdProdus, DProdus, UM, PretOferta, SpPoză)

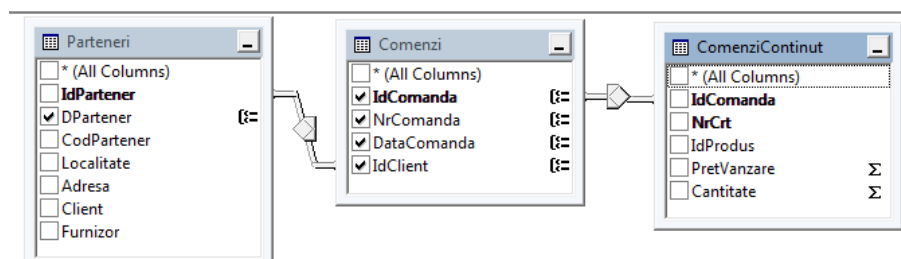
ComenziContinut(IdComanda, NrCrt, IdProdus, PretVanzare, Cantitate)

Obs. Se continua aplicatia cu introducere date in tabela asociere

Pas 1. Se completeaza data set-ul cu doua noi tabele:



Cererea SQL pt. ComenziTableAdapter:



Column	Alias	Table	Outp...	Sort Type	Sort Order	Group By
IdComanda		Comenzi	<input checked="" type="checkbox"/>			Group By
NrComanda		Comenzi	<input checked="" type="checkbox"/>			Group By
DataComanda		Comenzi	<input checked="" type="checkbox"/>			Group By
IdClient		Comenzi	<input checked="" type="checkbox"/>			Group By
DPartener		Parteneri	<input checked="" type="checkbox"/>			Group By
ComenziContinut.PretVanzare * ComenziContinut.Cantitate	Total		<input checked="" type="checkbox"/>			Sum

```

SELECT Comenzi.IdComanda, Comenzi.NrComanda, Comenzi.DataComanda, Comenzi.IdClient, Parteneri.DPartener,
SUM(ComenziContinut.PretVanzare * ComenziContinut.Cantitate) AS Total
FROM ((Comenzi LEFT OUTER JOIN
Parteneri ON Comenzi.IdClient = Parteneri.IdPartener) LEFT OUTER JOIN
ComenziContinut ON Comenzi.IdComanda = ComenziContinut.IdComanda)
GROUP BY Comenzi.IdComanda, Comenzi.NrComanda, Comenzi.DataComanda, Comenzi.IdClient, Parteneri.DPartener

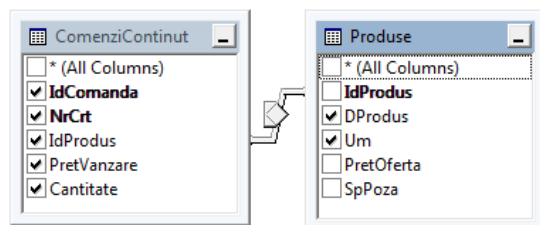
```

Cererea SQL pt. ComenziContinutTableAdapter:

```

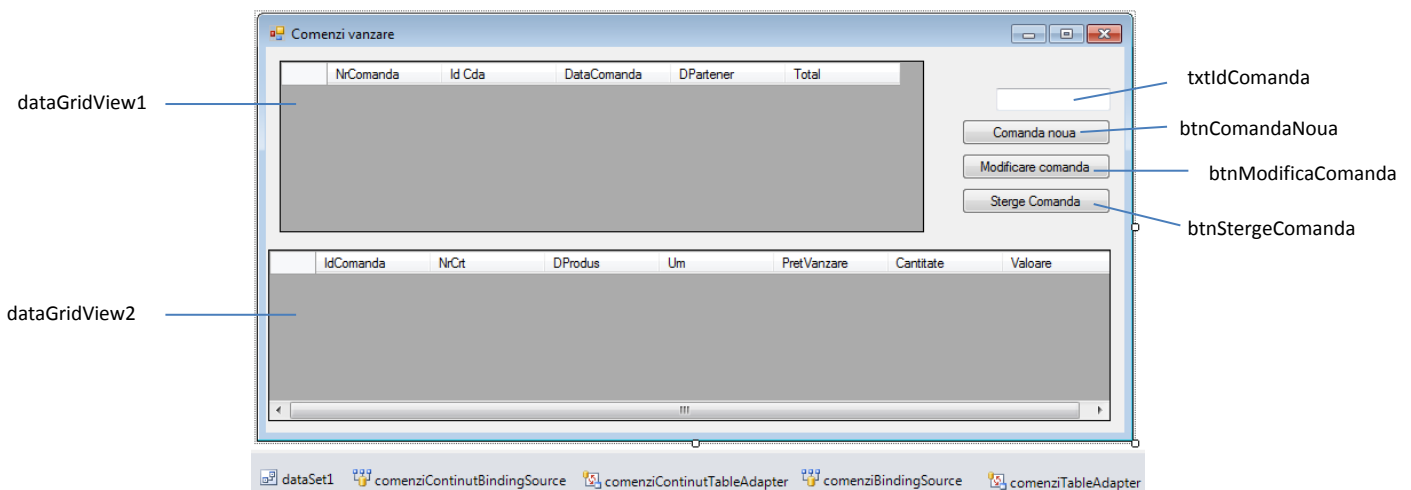
SELECT ComenziContinut.IdComanda, ComenziContinut.NrCrt, ComenziContinut.IdProdus, Produse.DProdus, Produse.Um,
ComenziContinut.PretVanzare, ComenziContinut.Cantitate,
ComenziContinut.PretVanzare * ComenziContinut.Cantitate AS Valoare
FROM (ComenziContinut LEFT OUTER JOIN
Produse ON ComenziContinut.IdProdus = Produse.IdProdus)

```



Column	Alias	Table	Outp...	Sort Type
IdComanda		ComenziContinut	<input checked="" type="checkbox"/>	
NrCrt		ComenziContinut	<input checked="" type="checkbox"/>	
IdProdus		ComenziContinut	<input checked="" type="checkbox"/>	
DProdus		Produse	<input checked="" type="checkbox"/>	
Um		Produse	<input checked="" type="checkbox"/>	
PretVanzare		ComenziContinut	<input checked="" type="checkbox"/>	
Cantitate		ComenziContinut	<input checked="" type="checkbox"/>	
ComenziContinut.PretVanzare * ComenziContinut.Cantitate	Valoare		<input checked="" type="checkbox"/>	

Pas 2 Se creaza un form nou pt. vizualizare comenzi



Obs. Se implementeaza doar vizualizare comenzi, Comanda noua si Stergere comanda

La executie form-ul arata astfel:

NrComanda	Id Cda	DataComanda	DPartener	Total
101	1	30.10.2014	Partener 03	60
102	2	01.10.2014	Partener 03	29.2
103	5	07.12.2017	Partener 05	23
104	8	11.12.2017	Partener 01	
105	9	11.12.2017	Partener 03	431

IdComanda	NrCt	DProdus	Um	PretVanzare	Cantitate	Valoare
1	1	Produs 02	Kg	12	2	24
1	2	Produs 02	Kg	12	3	36

Observatii:

1. Vizualizarea comenzilor se bazeaza pe un mecanism de filtrare al lui comenziContinutBindingSource cu id-ul comenzii curente.
2. txtIdComanda este legat la campul IdComanda din comenziBindingSource
3. dataGridView1 este legat la tabela Comenzi din dataSet
4. dataGridView2 este legat la tabela ComenziContinut din dataSet

Pas 3 Codul:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace AplCom
{
    public partial class FComenzi : Form
    {
        public FComenzi()
        {
            InitializeComponent();
        }

        private void refreshGrid()
        {
            comenziTableAdapter.Fill(dataSet1.Comenzi);
            comenziContinutTableAdapter.Fill(dataSet1.ComenziContinut);
        }

        private void FComenzi_Load(object sender, EventArgs e)
        {
            refreshGrid();

            comenziContinutBindingSource.Filter = "IdComanda=" + txtIdComanda.Text;
        }

        private void comenziBindingSource_PositionChanged(object sender, EventArgs e)
        {
            try
            {
                comenziContinutBindingSource.Filter = "IdComanda=" + txtIdComanda.Text;
            }
        }
    }
}
```

```

    }
    catch { }
}

private void btnComandaNoua_Click(object sender, EventArgs e)
{
    FComenziAct f = new FComenziAct();
    f.ShowDialog();
    refreshGrid();
}

private void btnStergeComanda_Click(object sender, EventArgs e)
{
    const string mesaj = "Confirmati stergerea";
    const string titlu = "Stergere inregistrare";

    var rezultat = MessageBox.Show(mesaj, titlu, MessageBoxButtons.YesNo,
                                    MessageBoxIcon.Warning);
    if (rezultat == DialogResult.No) return;

    OleDbConnection con = new OleDbConnection();
    OleDbCommand cmd = new OleDbCommand();

    con.ConnectionString = comenziTableAdapter.Connection.ConnectionString;

    cmd.Connection = con;

    // Sterg continut comanda
    cmd.CommandText = "Delete From ComenziContinut Where IdComanda = " + txtIdComanda.Text;

    MessageBox.Show(cmd.CommandText);

    con.Open();

    cmd.ExecuteNonQuery();

    // Sterg comanda
    cmd.CommandText = "Delete From Comenzi Where IdComanda = " + txtIdComanda.Text;
    MessageBox.Show(cmd.CommandText);
    cmd.ExecuteNonQuery();

    con.Close();

    // Refresh grid-uri
    refreshGrid();
}
}
}

```

Obs. FcomenziAct este numele form-ului ce efectueaza operatia *Comanda noua*.