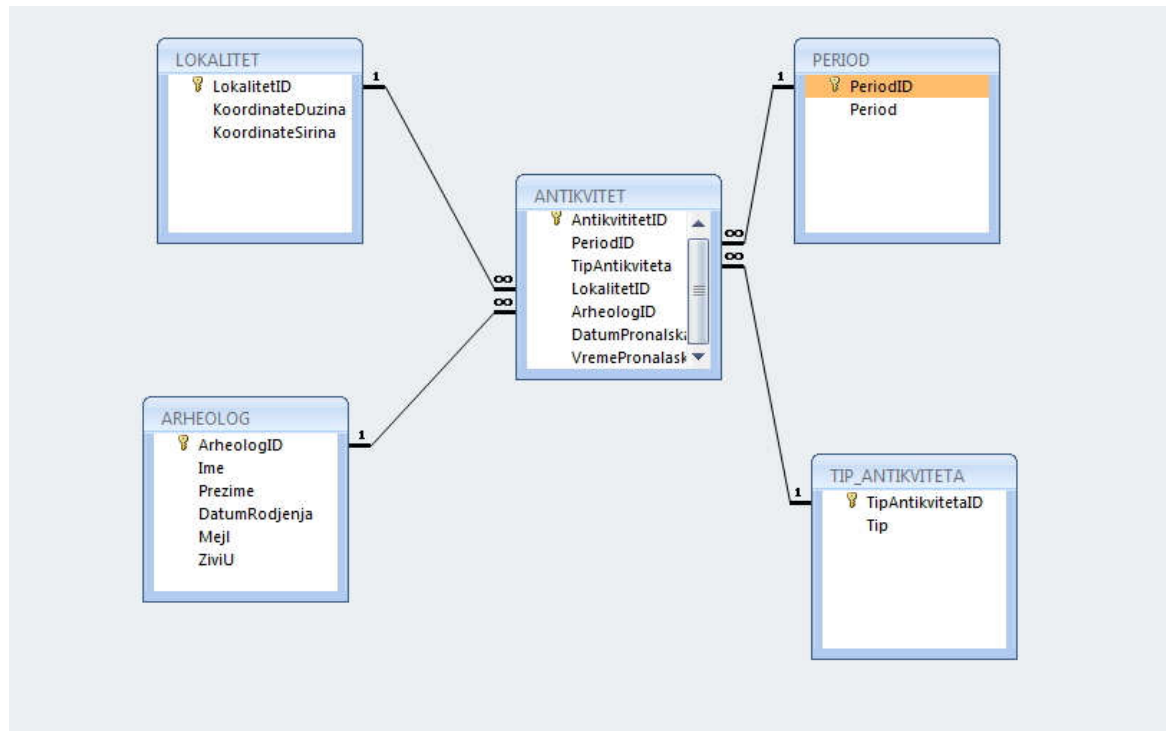


ШИФРА РАДНОГ ЗАДАТКА ЕИТ-Б9

НАЗИВ РАДНОГ ЗАДАТКА АНТИКВИТЕТИ И ЛОКАЦИЈЕ

За потребе археолога који истражује антиквитете на различитим локалитетима у Србији и из различитих периода креирана је база података на основу којих се утврђује како се Србија мењала кроз историју.

Дијаграм базе је дат на слици:

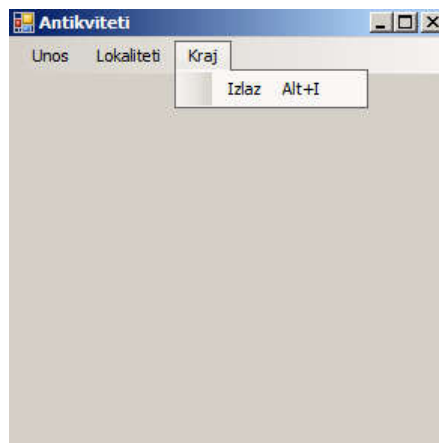
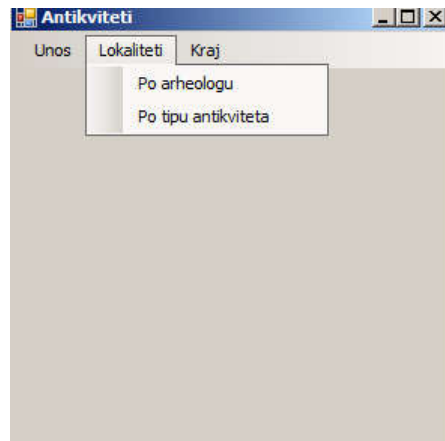
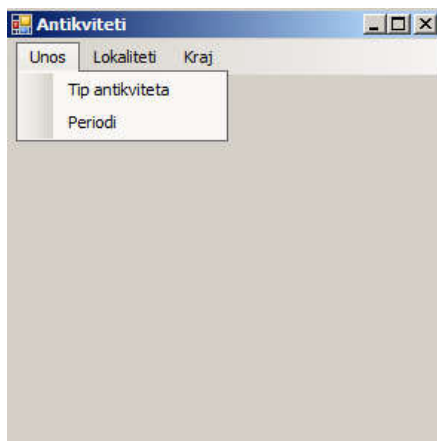


Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

Локалитети се налазе близу градова. Потребно је креирати табелу у којој би се чували подаци о имену града, његовом позивном броју, поштанском броју и броју становника.

Обзиром да су се границе Србије мењале кроз историју и да и ван граница наше земље можемо наћи антиквитете везане за наш народ потребно је чувати податке и о држави у којој се град налази, као и позивни број за државу и њен бој становника.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о типу антиквитета обавља се преко форме чији је изглед дат на слици:

Уколико унета шифра типа антиквитета постоји приказати његов назив.

Дугме **Izadji** затвара отворену форму, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о грешкама.

Ставка менија **Po arheologu**, приказује називе локалитета, координате његове географске дужине и ширине, за оне локалитете у чијем истраживању је учествовао тражени археолог.

Форма треба да изгледа као на слици:

Po arheologu

Sifra arheologa:

	Lokalitet	Geografska duzina	Geografska sirina
	Medijana	43 istocno	21 sevemo
►	Vinca	44 istocno	21 sevemo
	Caricin grad	42 istocno	20 sevemo

Подаци се приказују у *dataGridView* контроли.

За избрани локалитет из *dataGridView* контроле, графички се приказује географски положај локалитета.

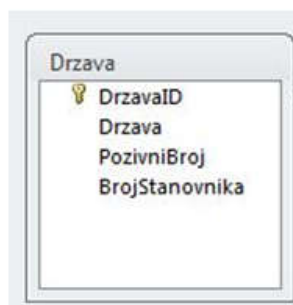
Дугме *Izadji* затвара отворену форму, и враћа на почетни мени.

Објашњење

Прво радимо проширење базе:

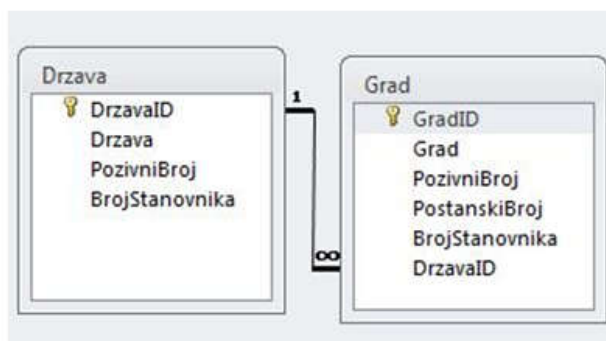
Креирамо табелу држава:

```
CREATE TABLE Drzava  
(  
  DrzavaID number PRIMARY KEY ,  
  Drzava CHAR(60),  
  PozivniBroj CHAR(60),  
  BrojStanovnika CHAR(60)  
)
```



Креирамо табелу Grad и повезујемо DrzavaID из табеле Grad са DrzavaID из табеле Drzava

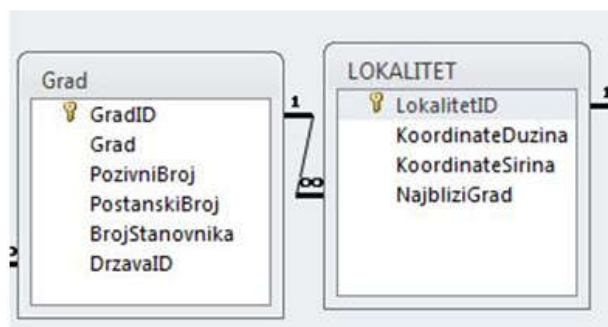
```
CREATE TABLE Grad  
(  
  GradID NUMBER PRIMARY KEY,  
  Grad CHAR(60),  
  PozivniBroj CHAR(60),  
  PostanskiBroj CHAR(60),  
  BrojStanovnika CHAR(60),  
  DrzavaID NUMBER,  
  FOREIGN KEY (DrzavaID)  
  REFERENCES Drzava(DrzavaID)  
)
```



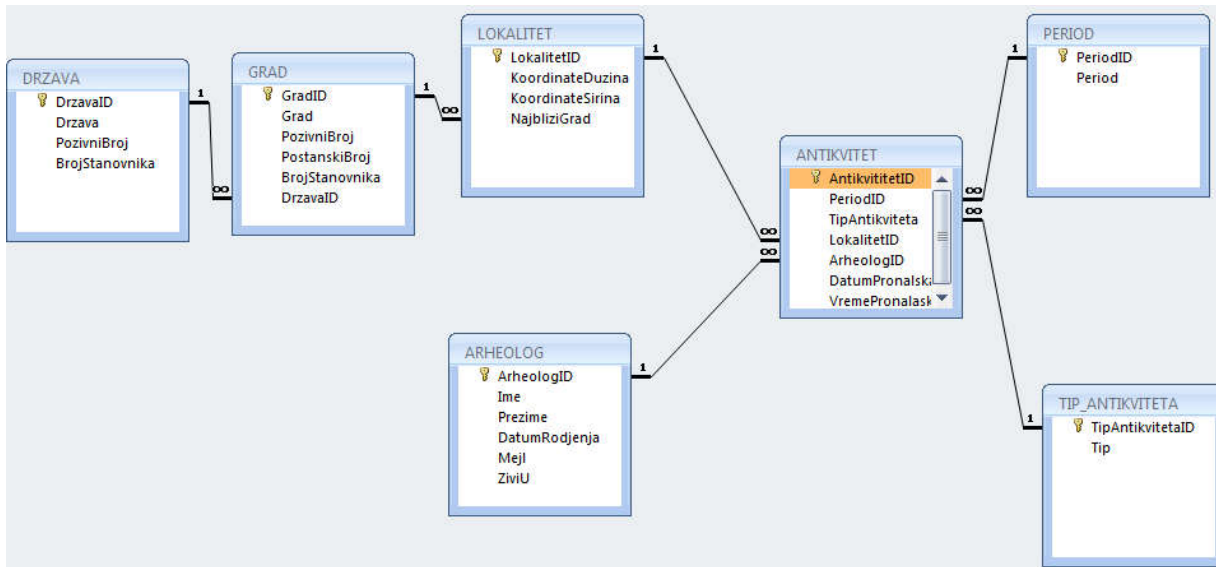
Повезујемо NajblziGrad из табеле Lokalitet са GradID из табеле Grad. При повезивању морамо у табели Grad да преправимо у пољу GradID Field Size из Double у Intiger.

```
ALTER TABLE Lokalitet
```

```
ADD CONSTRAINT Lokalitet FOREIGN KEY (NajblziGrad) REFERENCES Grad (GradID) ;
```



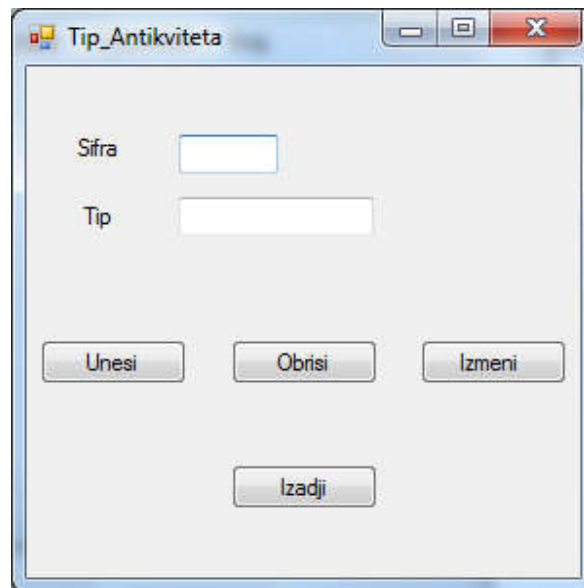
Коначни изглед базе:



Објашњење кода:



Тип антиквитета



The screenshot shows a standard Windows application window with a title bar that says 'Tip_Antikviteta'. Inside the window, there are two text boxes. The first is labeled 'Sifra' and the second is labeled 'Tip'. Below these text boxes, there are four buttons arranged in two rows. The top row contains 'Unesi', 'Obriši', and 'Izmeni'. The bottom row contains 'Izadji'.

```
konekcija.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
    Source=|DataDirectory|\Antikviteti i lokacije - osnovno.accdb";
```

```
OleDbConnection konekcija;
OleDbCommand komanda;
DataTable dt;
OleDbDataAdapter da;
ArrayList lista=new ArrayList();
void Konekcija()
{
    konekcija = new OleDbConnection();
    konekcija.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;
    Data Source=|DataDirectory|\Antikviteti i lokacije.accdb";
    komanda = new OleDbCommand();
    komanda.Connection = konekcija;
    dt = new DataTable();
    da = new OleDbDataAdapter();
}

//metoda za konekciju na bazu TIP_ANTI KVITETA
void Konekcija_na_TIP_ANTI KVITETA()
{
    Konekcija();
    komanda.CommandText = "SELECT * FROM TIP_ANTI KVITETA";
    da.SelectCommand = komanda;
    da.Fill(dt);
    konekcija.Open();
}
```

```
private void Tip_Antikviteta_Load(object sender, EventArgs e)
{
    Konekcija_na_TIP_ANTI KVITETA();
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        textBox1.AutoCompleteCustomSource.Add(dt.Rows[i]["TipAntikvitetaID"].ToString());
        textBox2.AutoCompleteCustomSource.Add(dt.Rows[i]["Tip"].ToString());
    }
}
```

Пошто се у задатку тражи да се приликом уноса прикажу подаци о особи која се налази у бази, податке о тој особи смештамо у листу.

```
public bool postoji_u_bazi()
{
    bool postoji = false;
    Konekcija_na_TIP_ANTI KVITETA();
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        if (textBox1.Text == dt.Rows[i]["TipAntikvitetaID"].ToString())
        {
            postoji = true;
            lista.Add(dt.Rows[i]["TipAntikvitetaID"].ToString());
            lista.Add(dt.Rows[i]["Tip"].ToString());
            break;
        }
    }
    konekcija.Close();
    return postoji;
}

//INSERT
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text != string.Empty && textBox2.Text != string.Empty)
    {
        if (postoji_u_bazi() == false)
        {
            Konekcija();
            string insert = "INSERT INTO TIP_ANTI KVITETA(TipAntikvitetaID,Tip)";
            string values = "VALUES (@TipAntikvitetaID,@Tip)";
            komanda.Parameters.AddWithValue("@TipAntikvitetaID", textBox1.Text);
            komanda.Parameters.AddWithValue("@Tip", textBox2.Text);
            komanda.CommandText = insert + " " + values;

            try
            {
                konekcija.Open();
                komanda.ExecuteNonQuery();
                MessageBox.Show("Podatak dodat u bazu");
                konekcija.Close();
            }
            .
        }
    }
}
```



```

        catch
        {
            MessageBox.Show("Greska");
            textBox1.Text = textBox2.Text = "";
            textBox1.Focus();
        }
    }
else
{
    MessageBox.Show("Podatak: " + lista[0] + "-" + lista[1] + " postoji u bazi");
    lista.Clear();
    textBox1.Text = textBox2.Text = "";
    textBox1.Focus();
}
else
{
    MessageBox.Show("Morate popuniti vrednosti");
    textBox1.Text = textBox2.Text = "";
    textBox1.Focus();
}
}
}
//DELETE
private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text != string.Empty)
    {
        Konekcija();
        komanda.CommandText = @"DELETE FROM TIP_ANTIKVITETA
                                WHERE TipAntikvitetaID=@TipAntikvitetaID";
        komanda.Parameters.AddWithValue("@TipAntikvitetaID", textBox1.Text);
    try
    {
        konekcija.Open();
        komanda.ExecuteNonQuery();
        MessageBox.Show("Podatak obrisan iz baze");
        konekcija.Close();
    }
    catch
    {
        MessageBox.Show("Greska");
    }
    else
    {
        MessageBox.Show("Morate popuniti");
    }
}

//UPDATE
komanda.CommandText = @"UPDATE TIP_ANTIKVITETA SET Tip=@Tip
                        WHERE TipAntikvitetaID=@TipAntikvitetaID";

```

Форма По археологу

	LokalitetID	KoordinateDuzina	Koc
▶	Vinča	21 zapadno	45 jL
	Maglič	20 zapadno	43 s
	Viminacium	22 istočno	45 s
	Vinča	21 zapadno	45 jL

```
//metoda za konekciju na tabelu Arheolog
void Konekcija_na_ARHEOLOG()
{
    Konekcija();
    komanda.CommandText = "SELECT * FROM ARHEOLOG";
    da.SelectCommand = komanda;
    da.Fill(dt);
}

private void Po_Arheologu_Load(object sender, EventArgs e)
{
    Konekcija_na_ARHEOLOG();
    for (int i = 0; i < dt.Rows.Count; i++)
    {
        textBox1.AutoCompleteCustomSource.Add(dt.Rows[i]["ArheologID"].ToString());
    }
}

//UPIT
string select =
@"SELECT Lokalitet.LokalitetID,Lokalitet.KoordinateDuzina,Lokalitet.KoordinateSirina ";

string from =
@"FROM ((Antikvitet INNER JOIN Arheolog ON Antikvitet.ArheologID=Arheolog.ArheologID)
INNER JOIN Lokalitet ON Antikvitet.LokalitetID=Lokalitet.LokalitetID)";

string where = "WHERE (Arheolog.ArheologID=@ArheologID)";

komanda.Parameters.AddWithValue("@ArheologID", textBox1.Text);
komanda.CommandText = select + " "+from + " " + where;
```

//CRTANJE

Изглед података у бази

LokalitetID	LokalitetIme	KoordinateDuzina	KoordinateSirina
1	Caričin grad	21 zapadno	44 severno

```
string duzina, sirina;
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        //rastavljamo stringove koordinata na dva stringa
        string[] du = duzina.Split(' ');
        string[] si = sirina.Split(' ');
        double x = 0;
        double y = 0;

        if (du[1] == "istočno")
        {
            x = Convert.ToDouble(du[0]);
        }
        else
        {
            x = -Convert.ToDouble(du[0]);
        }

        if (si[1] == "severno")
        {
            y = -Convert.ToDouble(si[0]);
        }
        else
        {
            y = Convert.ToDouble(si[0]);
        }

        Pen olovka = new Pen(Color.Red, 3);
        Graphics g = pictureBox1.CreateGraphics();
        g.Clear(Color.White);
        g.DrawRectangle(olovka, 0, 0, pictureBox1.Width - 1, pictureBox1.Height - 1);
        g.DrawLine(olovka, pictureBox1.Width / 2, 0, pictureBox1.Width / 2, pictureBox1.Height);
        g.DrawLine(olovka, 0, pictureBox1.Height / 2, pictureBox1.Width, pictureBox1.Height / 2);
        g.DrawEllipse(olovka, pictureBox1.Width / 2 + ((int)x - 5), pictureBox1.Height / 2 + ((int)y - 5), 5, 5);

        //SELEKTOVANJE REDA U dataGridView
        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            duzina = dataGridView1[1, e.RowIndex].Value.ToString();
            sirina = dataGridView1[2, e.RowIndex].Value.ToString();
        }
    }
}
```