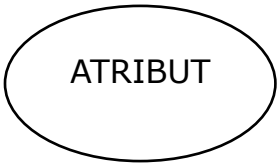
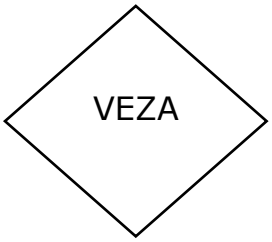
PRIRUČNIK ZA IZRADU PROJEKTA DESKTOP APLIKACIJE

ER dijagram

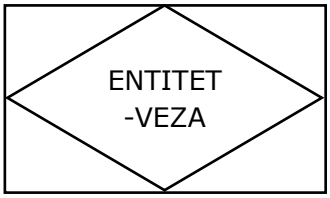
Predstavlja tabelu u bazi podataka, ono što opisujemo.



Predstavlja polje unutar tabele, odnosno entiteta.



Povezanost različitih entiteta, uglavnom se opisuje glagolom.



Predstavlja vezu sa atributima.

Vrste veza (kardinaliteta) između entiteta

1:1 – npr. jedna osoba može da ima samo jedan pasoš određene države.

1:M – npr. jedna osoba može da ima jedan ili više brojeva telefona.

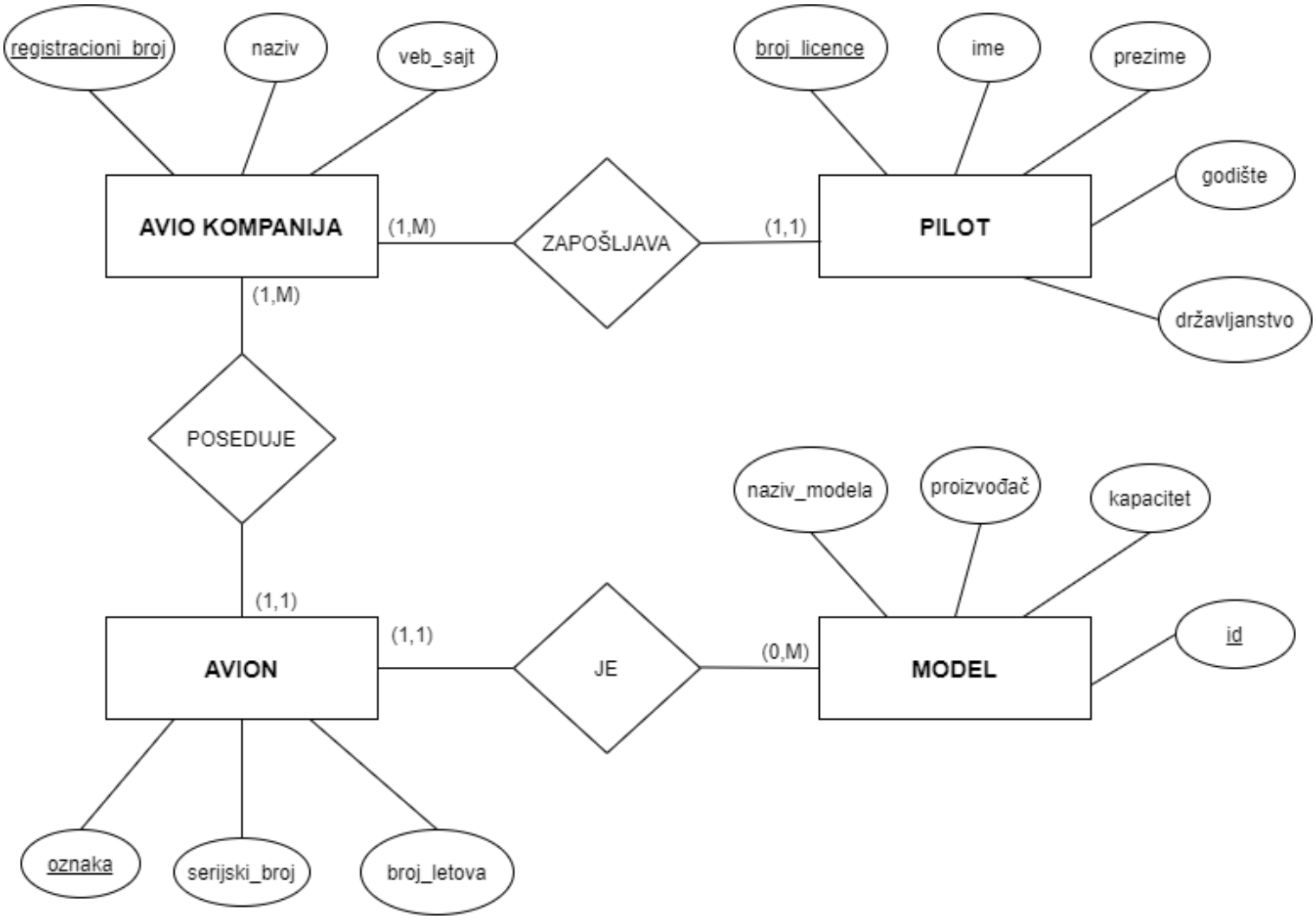
M:N – npr. jedan profesor može da predaje više predmeta, a istovremeno i taj predmet može da predaje više profesora.

Primarni ključ – Atribut koji jedinstveno identifikuje vrste u koloni, npr. JMBG za osobu. Na ER dijagramu je potrebno podvući atribut koji predstavlja primarni ključ.

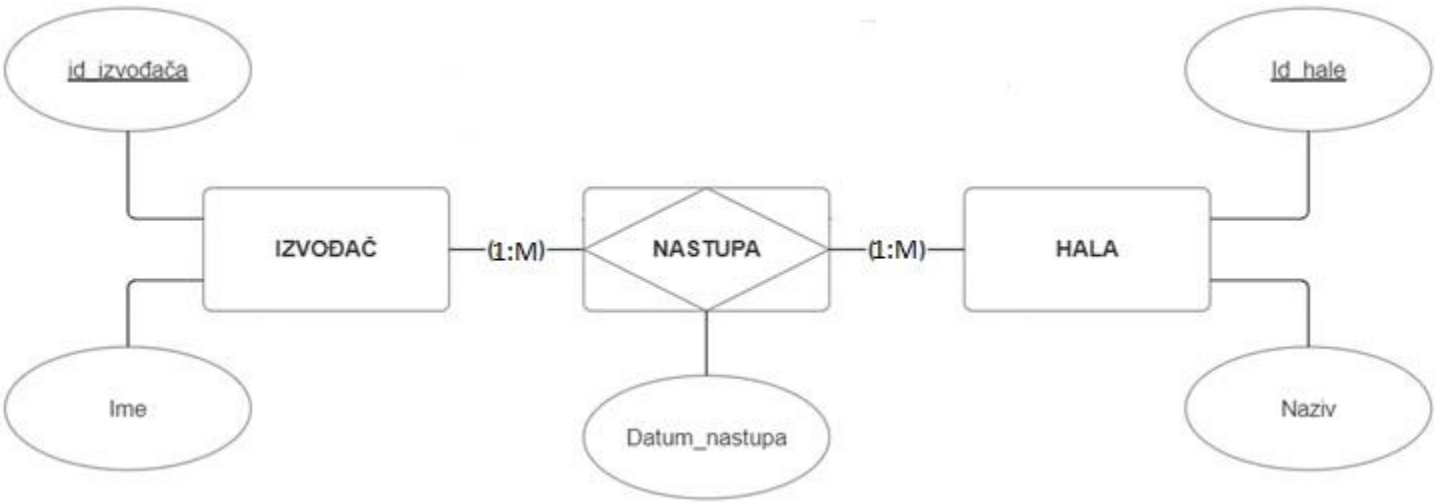
Strani ključ – atribut koji entitet preuzima od drugog entiteta s kojim je povezan (primarni ključ druge tabele bitan za trenutnu tabelu). Strani ključevi se ne navode na ER dijagramu.

Primeri ER dijagrama

Avio kompanija



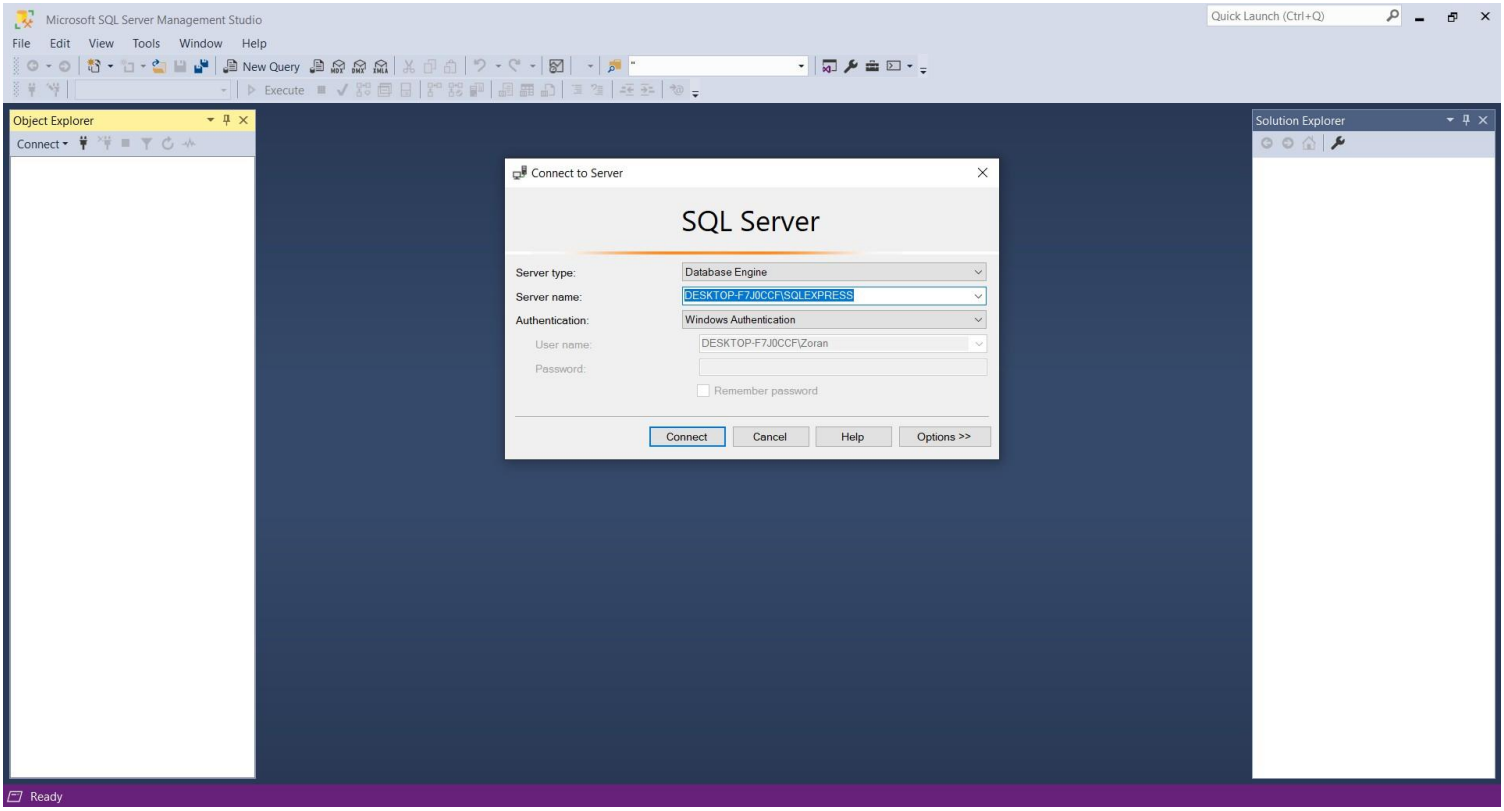
Izvođači

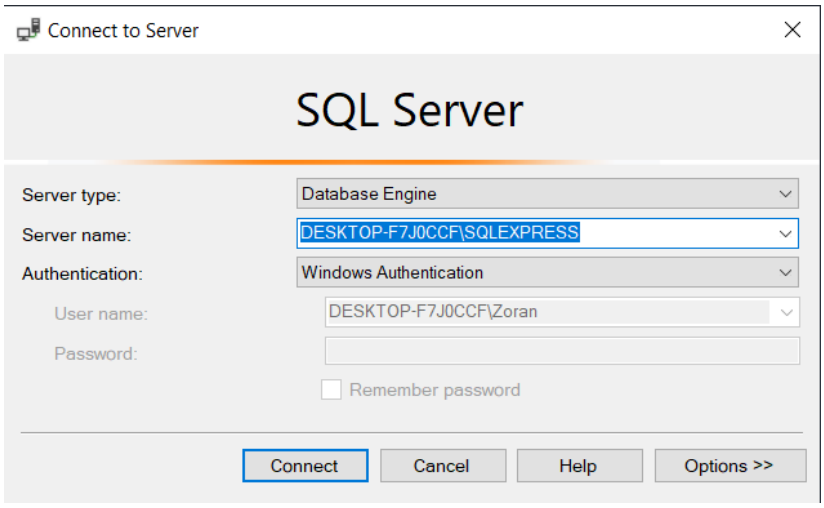


SQL Server Management Studio – početni koraci

Pokretanje Microsoft SQL Server Management Studio okruženja:

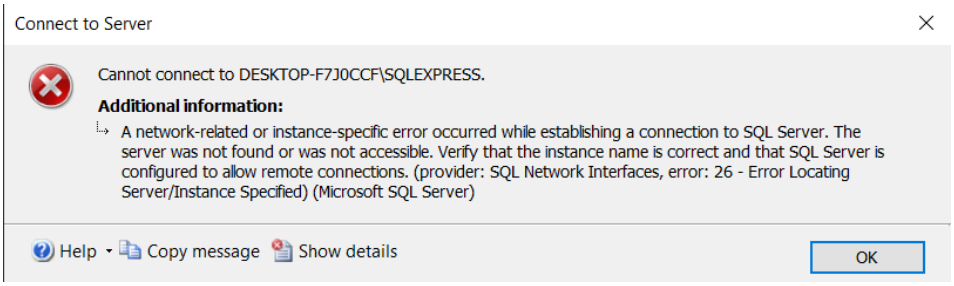
Nakon pokretanja programa dobija se radno okruženje i otvoren prozor *Connect to Server*:

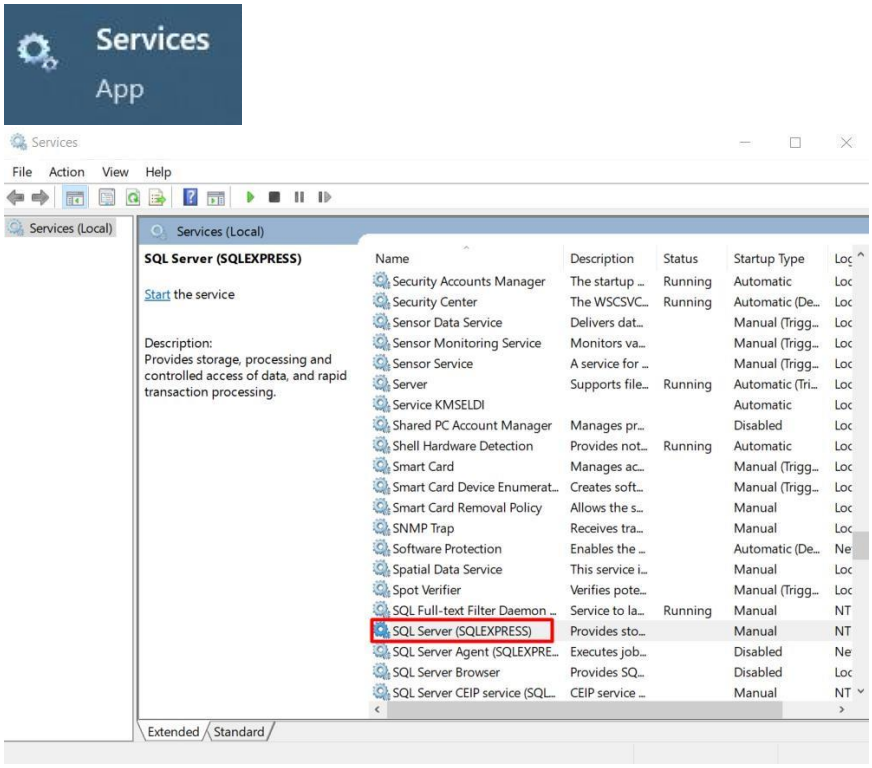




U prozoru *Connect to Server* potrebno je izabrati opciju *Connect*. Na mestu *Server name* će stojati ime_računara\SQLEXPRESS ako prilikom instalacije nije drugačije naznačeno.

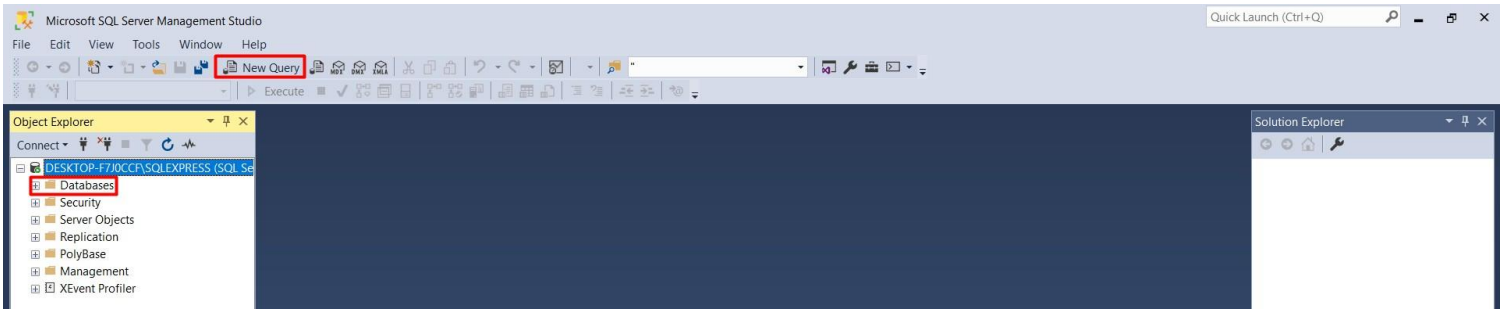
Ukoliko prilikom konektovanja na server dođe do greške prikazane na slici ispod, to je znak da SQL Server nije pokrenut na računaru.





Da bi se SQL Server pokrenuo pristupa se servisima (*Services*) i pronalazi se servis *SQL Server (SQLEXPRESS)* i pokreće se desnim klikom na njega i izborom opcije *Start*.

Nakon pokretanja potrebnog servisa ponovo biramo opciju *Connect* u prozoru *Connect to Server* i dobijamo radno okruženje sa *Object Explorer*-om na levoj, *Solution Explorer*-om na desnoj strani i praznim mestom u sredini. Upite kreiramo izborom opcije *New Query* iz gornje trake sa opcijama (ili kombinacijom tastera *CTRL* i *N*). Unutar *Object Explorer*-a će se nalaziti sve baze podataka koje kreirate, zajedno sa njihovim tabelama, pogledima itd.



Nakon kreiranja novog upita možemo birati nad kojom bazom podataka želimo da ga izvršimo, izborom baze iz padajućeg menija u gornjoj traci sa alatima. Izborom opcije Execute izvršavamo kreirani upit.

SQL podsetnik – komande i funkcije

Koristićemo tabelu *Student(StudentID, Ime, Prezime, Grad, GodinaUpisa, DatumRodjenja)* kao primer.

SELECT – čitanje podataka

Prikazuje kolone iz tabele.

```sql
SELECT * FROM Student;
SELECT Ime, Prezime FROM Student;
```

WHERE – filtriranje

Prikazuje samo redove koji ispunjavaju uslov.

```sql
SELECT Ime, Prezime FROM Student WHERE Grad = 'Beograd';
```

INSERT – unos novih podataka

Dodaje novi red.

```sql
INSERT INTO Student (StudentID, Ime, Prezime, Grad)
VALUES (1, 'Petar', 'Petrović', 'Beograd');
```

UPDATE – izmena podataka

Menja vrednosti u postojećim redovima.

```sql
UPDATE Student
SET Grad = 'Novi Sad'
WHERE StudentID = 1;
```

DELETE – brisanje redova

```sql
DELETE FROM Student WHERE StudentID = 1;
```

ORDER BY – sortiranje rezultata

```sql
SELECT Ime, Prezime FROM Student ORDER BY Prezime DESC;
```

BETWEEN – opseg vrednosti

Filtrira po vrednosti između dva kraja (uključivo).

```sql
SELECT Ime, GodinaUpisa
FROM Student
WHERE GodinaUpisa BETWEEN 2019 AND 2021;
```

COUNT – brojanje redova

```sql
SELECT COUNT(*) AS [Broj studenata] FROM Student;
```

DISTINCT – jedinstvene vrednosti

```sql
SELECT DISTINCT Grad FROM Student;
```

Bez dupliranja istih vrednosti.

GROUP BY – grupisanje

Grupiše redove po koloni i koristi se sa COUNT, SUM, AVG... Sve kolone koje nisu deo agregatne funkcije (COUNT, MAX, MIN…) se moraju naći u GROUP BY klauzuli.

```sql
SELECT Grad, COUNT(*) AS BrojStudenata
FROM Student
GROUP BY Grad;
```

HAVING – uslov nad grupama

Slično WHERE, ali posle grupisanja.

```sql
SELECT Grad, COUNT(*) AS BrojStudenata
FROM Student
GROUP BY Grad
HAVING COUNT(*) > 10;
```

YEAR() – izdvajanje godine iz datuma

```sql
SELECT YEAR(DatumRodjenja) AS [Godina rođenja] FROM Student;
```

GETDATE() – trenutni datum i vreme

U SQL Server-u vraća trenutni datum/vreme servera.

```sql
SELECT GETDATE() AS Sada;
```

Može se koristiti za filtriranje:

```sql
WHERE YEAR(DatumUpisa) >= YEAR(GETDATE()) - 5
```

CONVERT – promene tipova (npr. broj u tekst)

Korisno za spajanje teksta i broja.

```sql
SELECT CONVERT(varchar(10), StudentID) + '-' + Ime AS Kod FROM Student;
```

CASE – uslov unutar SELECT-a

Koristi se za pravljenje kolona sa uslovima (kao if/else).

```
SELECT Ime,
       CASE WHEN GodinaUpisa < 2020 THEN 'Stari'
            WHEN GodinaUpisa = 2020 THEN 'Srednji'
            ELSE 'Novi'
       END AS Status
FROM Student;
```

JOIN – spajanje tabela

```
SELECT s.Ime, g.Naziv
FROM Student s
JOIN Grad g ON s.GradID = g.GradID;
```

Kombinovani primer

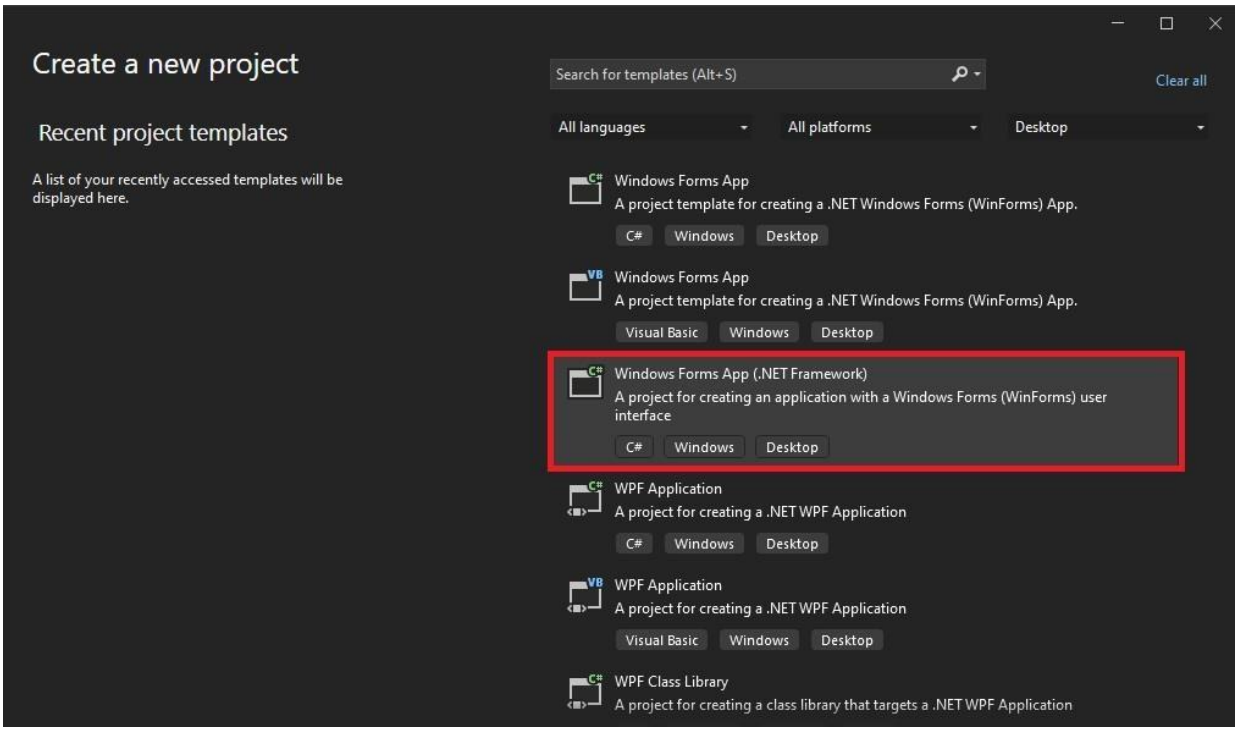
Prikaz broja studenata po gradu, samo za poslednje 5 godina:

```
SELECT g.Naziv AS Grad,
       COUNT(*) AS Broj
FROM Student s, Grad g
WHERE s.GradID = g.GradID
  AND YEAR(s.GodinaUpisa) BETWEEN YEAR(GETDATE())-5 AND YEAR(GETDATE())
GROUP BY g.Naziv
HAVING COUNT(*) > 10
ORDER BY Broj DESC;
```

Windows forme

Potrebno je kreirati forme sa svim potrebnim kontrolama (TextBox-ovi, Label-i itd.) za 7 maturskih zadataka iz programiranja.

Prilikom kreiranja projekta birati opciju Windows Forms App (.NET Framework).



NAPOMENA:

Kontrola sa vidljivim kolonama je *ListView*. Kolone se dodaju preko svojstva *Columns*, a da bi bile vidljive potrebno je svojstvo *View* promeniti na *Details*. Ukoliko je potrebno obezbediti da se selektuje ceo red vrednost svojstva *FullRowSelect* se menja na *True*.

Kontrola sa sivom pozadinom je *DataGridView*.

Kontrola sa grafikom je *Chart*. Tip grafika se bira unutar svojstva *Series* i opcije *ChartType*.

Kontrola sa tabovima je *TabControl*. Tabovi se dodaju i imenuju preko svojstva *TabPages*.

Kontrola sa dugmadima u vrhu je *ToolStrip*. Bitna svojstva za postavljanje i pozicioniranje teksta i slike su *DisplayStyle* i *TextImageRelation*. Za veličinu slike *ImageScalingSize*. Za pozicioniranje samog *ToolStrip*-a svojstvo *Dock*.

Kontrola za brojeve je *NumericUpDown*. Bitna svojstva su *Minimum*, *Maximum* i *Value*.

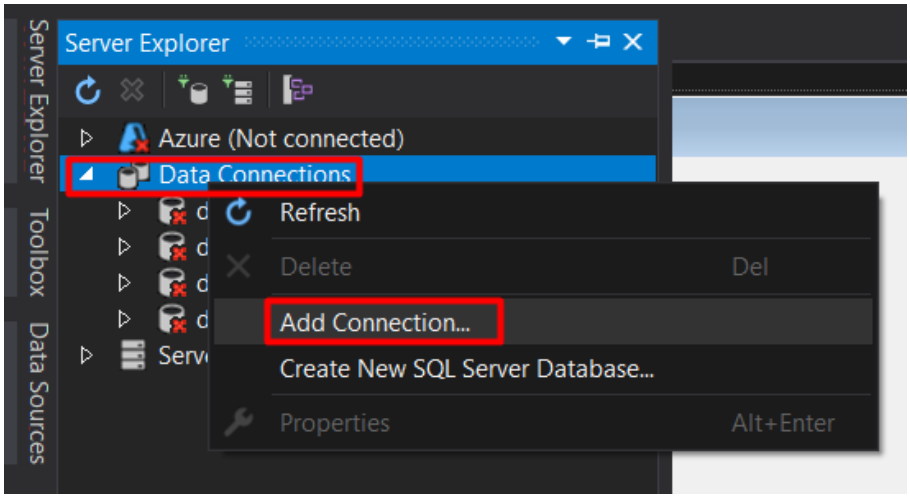
Kontrola padajućeg menija je *ComboBox*, a kontrola za unos je *TextBox* (svojstvo za maksimalan broj karaktera za unos je *MaxLength*).

Dodavanje nove forme u projekat se vrši desnim klikom na ime projekta u *Solution Explorer*-u i izborom opcije *Add – New – Form*.

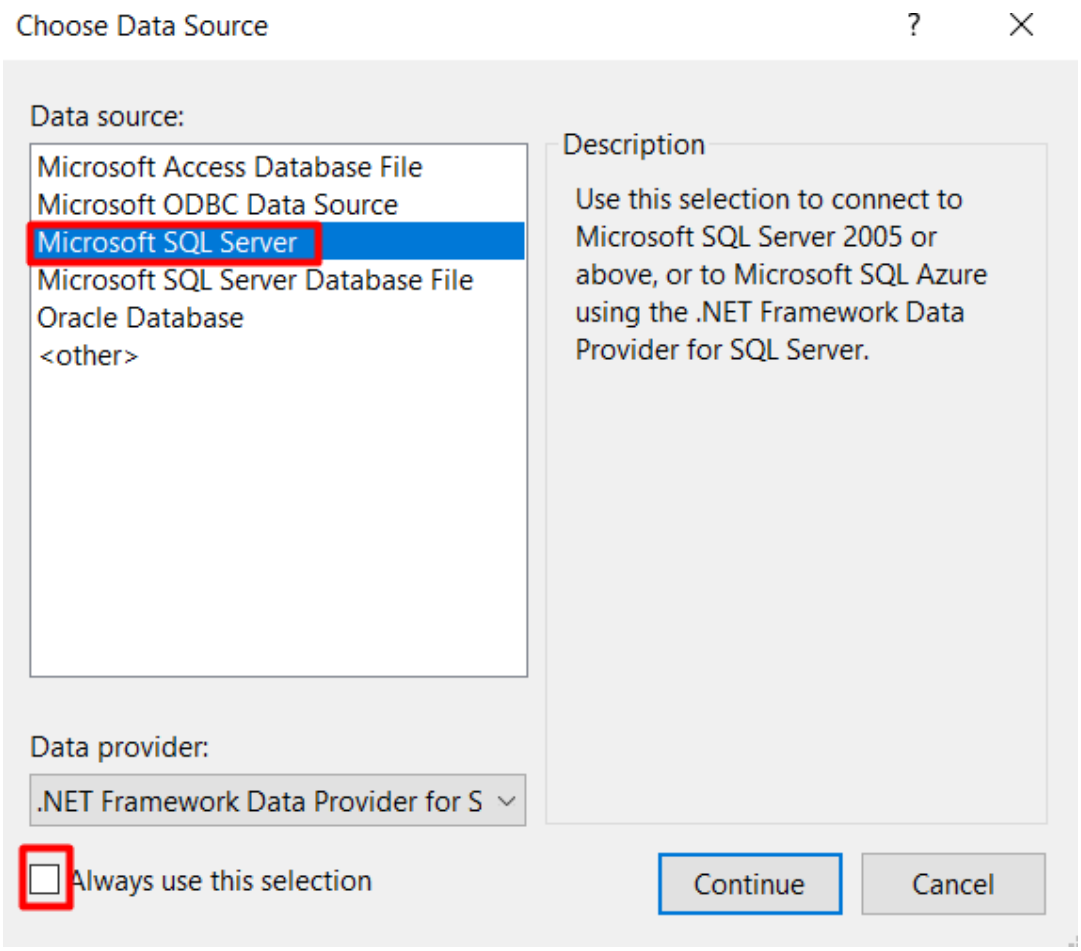
Slike koje se koriste (preko svojstava *BackgroundImage* i *Image*) ubacuju se u resurse programa izborom opcije *Project resource file* i *Import...*. Podešavanja veličine slike i uklapanja u kontrolu se vrše preko svojstava *BackgroundImageLayout* (za pozadinsku sliku) i *SizeMode* za običnu sliku.

Povezivanje baze podataka sa projektom – početni koraci

Unutar *Server Explorer*-a desnim klikom na *Data Connections* biramo opciju *Add Connection…* (ukoliko se *Server Explorer* ne nalazi na levoj strani radne površine otvaramo ga iz gornjeg menija biranjem opcija *View -> Server Explorer*).

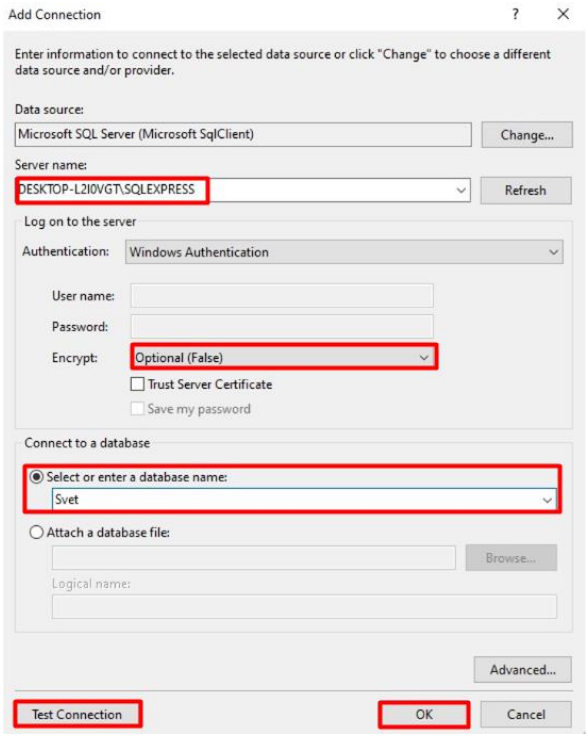


U prozoru *Choose Data Source* biramo opciju *Microsoft SQL Server*, a polje *Always use this selection* ostavljamo nečekirano. Zatim biramo opciju *Continue*.

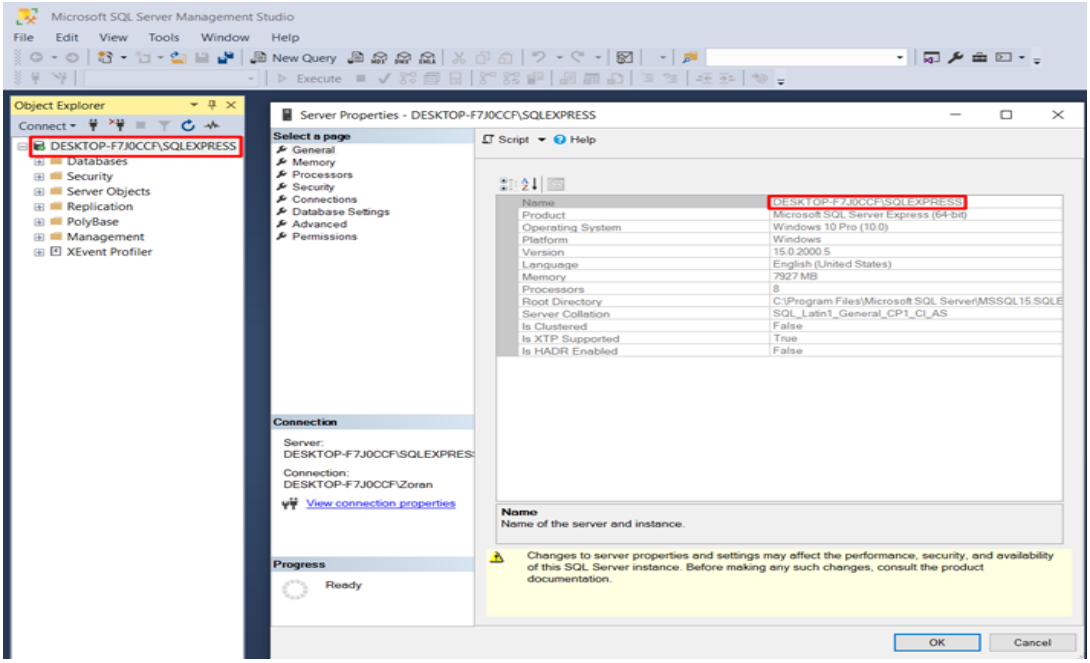


U polje *Server name:* upisujemo ime servera (isto koje je i u SQL Management Studio-u).

Obavezno je izabrati opciju *Optional (False)* unutar *Encrypt* dela. Zatim u padajućem meniju biramo bazu podataka za koju želimo da dodamo konekciju. Nakon toga pritiskom na dugme *Test Connection* dobijamo povratnu informaciju da li je konekcija uspešna (*Test connection succeeded* ako jeste). Ukoliko je konekcija uspešna biramo dugme *OK*.



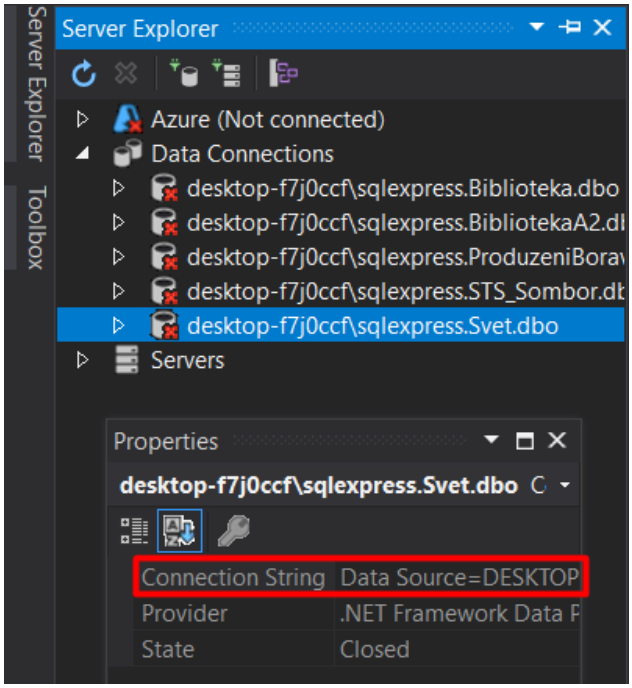
Desnim klikom na server u *Object Explorer*-u (leva slika) i izborom opcije *Properties* možemo doći do imena servera.



Konekcioni string

Konekcioni string koji nam je kasnije potreban za povezivanje sa bazom možemo naći u *Properties*-u od kreirane konekcije na bazu podataka.

Konekcioni string nam koristi da bi projekat znao na koji server i koju bazu podataka je potrebno da se poveže.



Kopirani tekst konekcionog stringa čuvamo unutar tekstualne promenljive proizvoljnog naziva (u primerima i priručniku promenljiva će biti nazivana *konekcioniString*).

```
string konekcioniString = @"Data Source=DESKTOP-F7J0CCF\SQLEXPRESS;Initial Catalog=Biblioteka;Integrated Security=True";
```

Rad sa kontrolama

Rad sa ListView-om

ListView kontrola se na formu dodaje preuzimanjem iz *Toolbox*-a.

- ***Dodavanje kolona*** - Kolone se unutar ListView-a mogu dodavati ili preko svojstva **Columns** unutar

Properties-a ili unutar koda forme (imeListView.Columns.Add("ImeKolone");).

- ***Prikaz kolona*** – Da bi videli naslove i podatke u kreiranim kolonama svojstvo **View** menjamo na vrednost **Details** (takođe ili preko *Properties*-a ili unutar koda (imeListView.View = View.Details;).
- ***Selektovanje celog reda*** – Ukoliko želimo da se klikom na red u ListView-u selektuju sve kolone tog reda menjamo svojstvo **FullRowSelect** u vrednost **True**.
- ***Selektovanje većeg broja redova*** – Ukoliko želimo da omogućimo/onemogućimo selektovanje većeg broja redova unutar ListView-a menjamo svojstvo **MultiSelect** u vrednost **True/False**.

Ukoliko želimo da određeni red bude selektovan:

```
imeListViewa.SelectedIndices.Add(indeks od željenog reda);
```

Ukoliko želimo da određeni red bude vidljiv (ukoliko je lista dugačka i potrebno je skrolovanje metoda

EnsureVisible() će automatski dovesti do željenog reda):

```
imeListViewa.Items[indeks od željenog reda].EnsureVisible();
```

Popunjavanje kolona podacima

Da bi popunili kolone podacima koristimo ranije pomenute klase za rad sa bazama podataka. Primer :

```
// Dok čitač prolazi kroz podatke na osnovu prosleđenog upita kod unutar petlje će se izvršavati
            while (citac.Read())
            {
                /* Kreira se stavka ListView-a, a u konstruktor se postavlja prva željena vrednost u
                tekstualnoj vrednosti */
                ListViewItem prikaz = new ListViewItem(citac[0].ToString());
/* Na kreiranu stavku se dodaju SubItem-i, onoliko njih koliko podataka želimo da izvučemo iz upita
                Broj polja ne sme biti veći od broja polja koje vraća upit */
                prikaz.SubItems.Add(citac[1].ToString());
                prikaz.SubItems.Add(citac[2].ToString());
                prikaz.SubItems.Add(citac[3].ToString());
                prikaz.SubItems.Add(citac[4].ToString());
                // Kreirana stavka sa svojim SubItem-ima se dodaje ListView-u
                imeListViewa.Items.Add(prikaz);
            }
```

Rad sa ComboBox-om

Popunjavanje *ComboBox*-a podacima iz baze podataka se vrši pomoću korišćenja klasa *SqlDataAdapter* i *DataTable*.

```
            SqlDataAdapter adapter = new SqlDataAdapter(SQLupit, konekcija);

            DataTable podaci = new DataTable();
            adapter.Fill(podaci);

            // Popunjena tabela se dodaje kao DataSource
            imeComboBoxa.DataSource = podaci;

            // Za ValueMember se uzima unikatna vrednost iz upita
            imeComboBoxa.ValueMember = "UnikatnoPoljeIzTabele";
```

Rad sa DataGridView-om

DataGridView kontrola se na formu dodaje preuzimanjem iz *Toolbox*-a.

```
SqlDataAdapter dataAdapter = new SqlDataAdapter(SQLupit);

DataTable podaci = new DataTable();
dataAdapter.Fill(podaci);

// Podatke iz tabele dodeljujemo DataGridView-u
imeDataGridViewa.DataSource = podaci;
```

Preko svojstva *ReadOnly* se omogućava/onemogućava da se vrši unos u polja *DataGridView*-a.

Rad sa Chart-om

Chart kontrola se na formu dodaje preuzimanjem iz *Toolbox*-a. U zavisnosti od tipa *Chart*-a (Line, Pie, Column...) zavisiće i izgled koda.

```
// Dodeljujemo izvor podataka našem chart-u
imeCharta.DataSource = dataAdapter;

// Na x osu postavljamo kolonu po kojoj želimo da se mere podaci

imeCharta.Series[0].XValueMember = "Kolona X ose";

// Na y osu postavljamo kolonu koja po kojoj će se meriti podaci sa x ose

imeCharta.Series[0].YValueMembers = "Kolona Y ose";

// Unosimo podatke unutar chart-a
imeCharta.DataBind();
```