
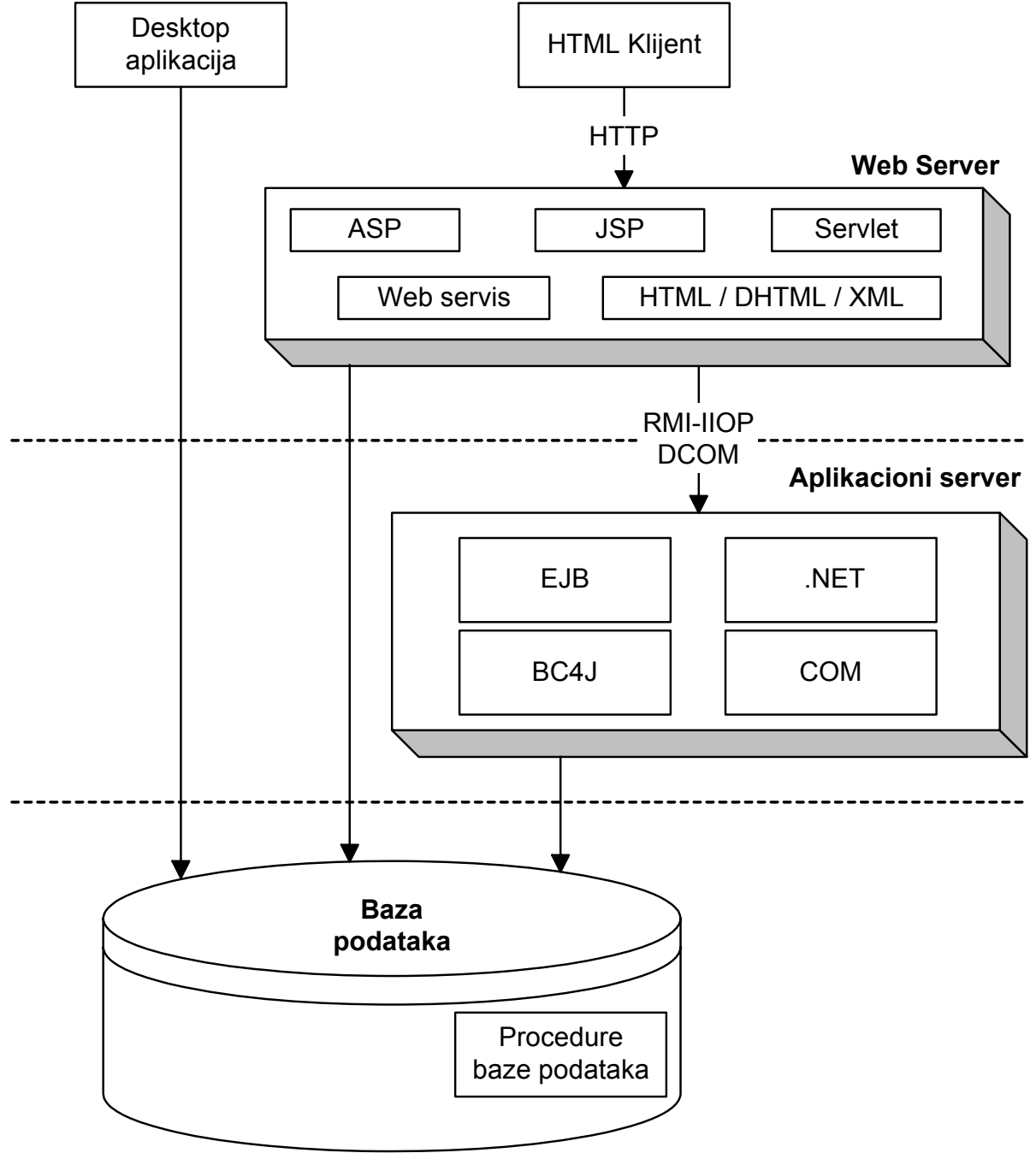


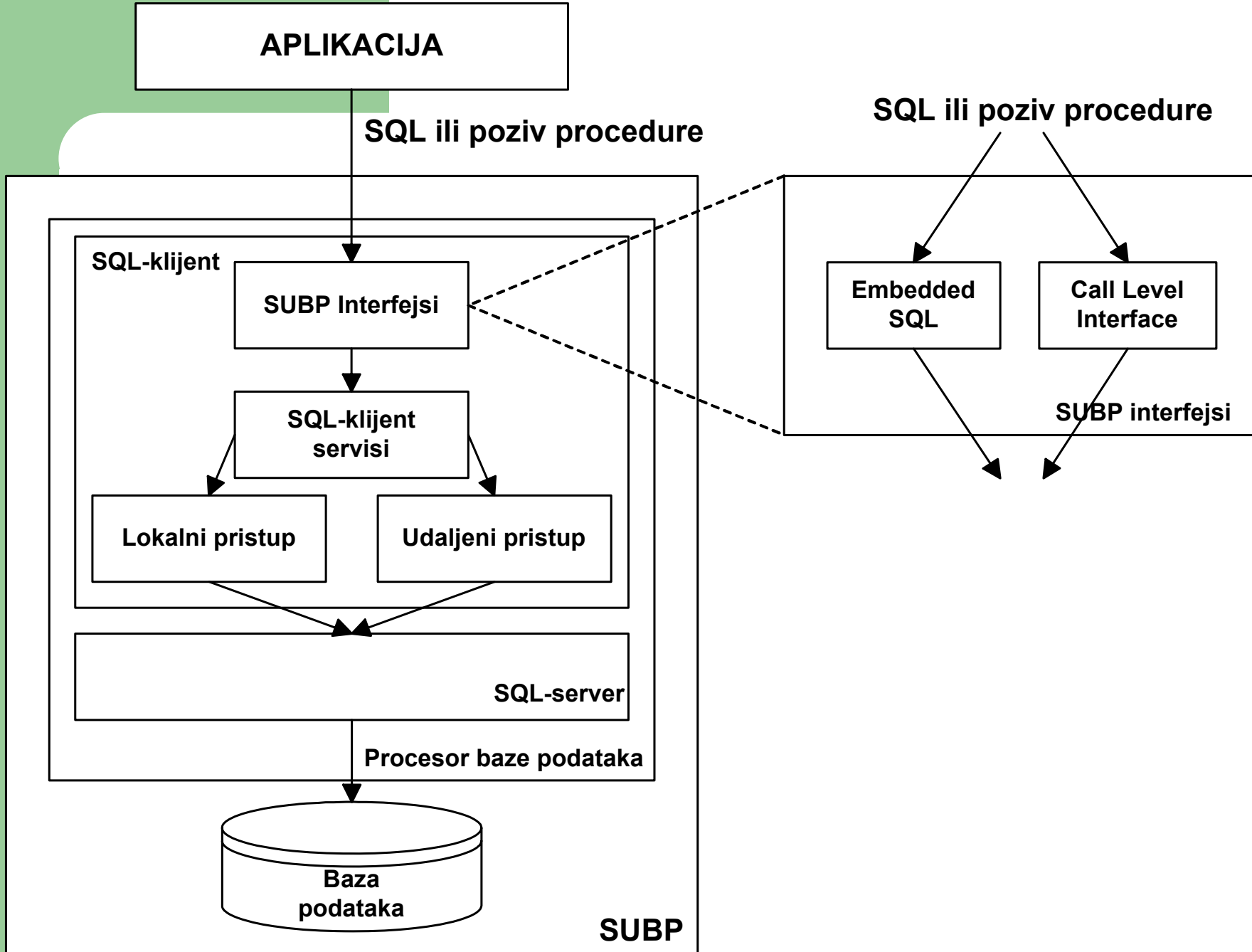
BAZE PODATAKA i RAZVOJ APLIKACIJA



Copyright © 2004 Laboratorija za
informacione sisteme,
Fakultet organizacionih nauka



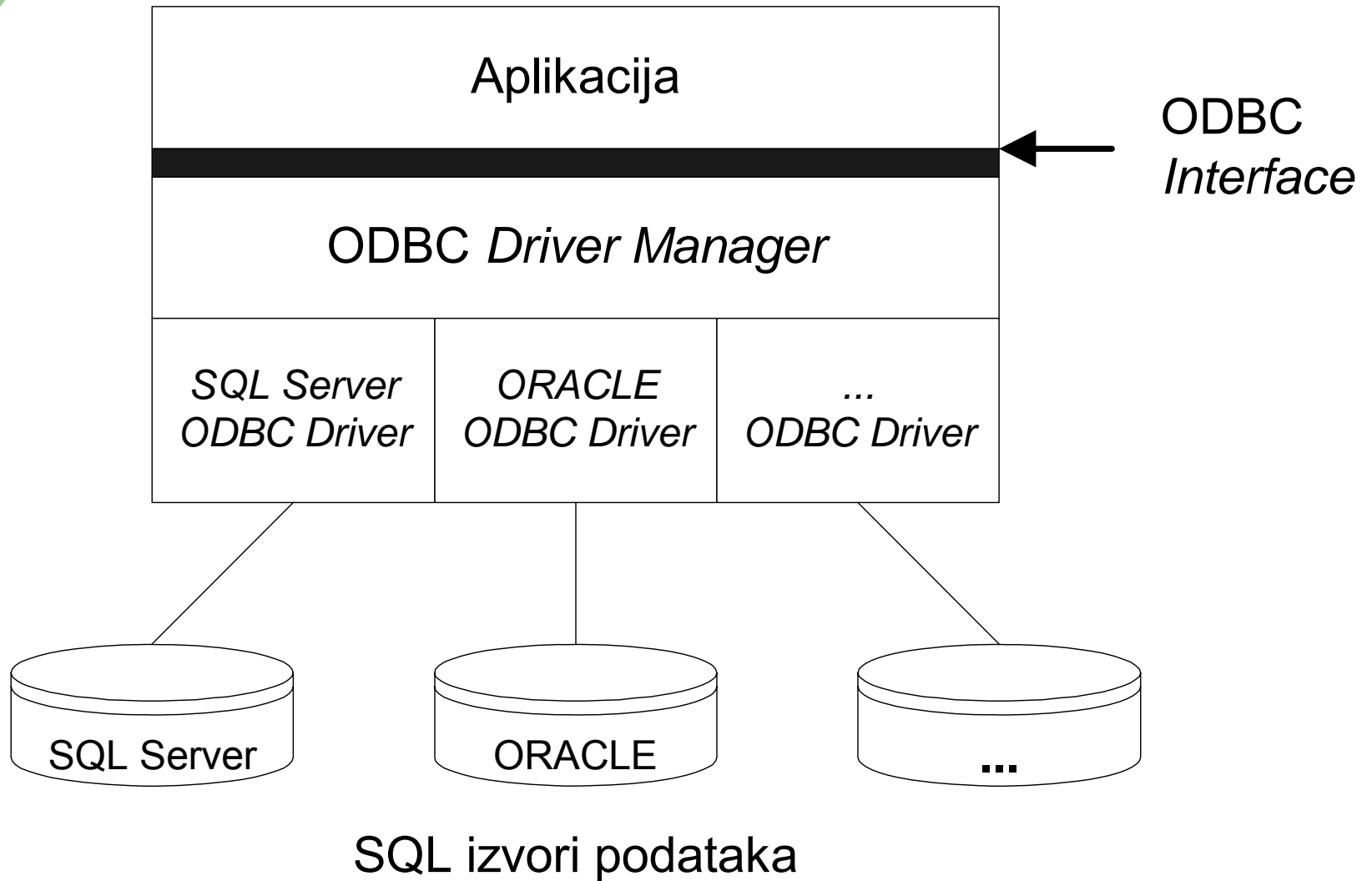
Arhitektura za razvoj aplikacija nad bazom podataka



Open Database Connectivity (ODBC)

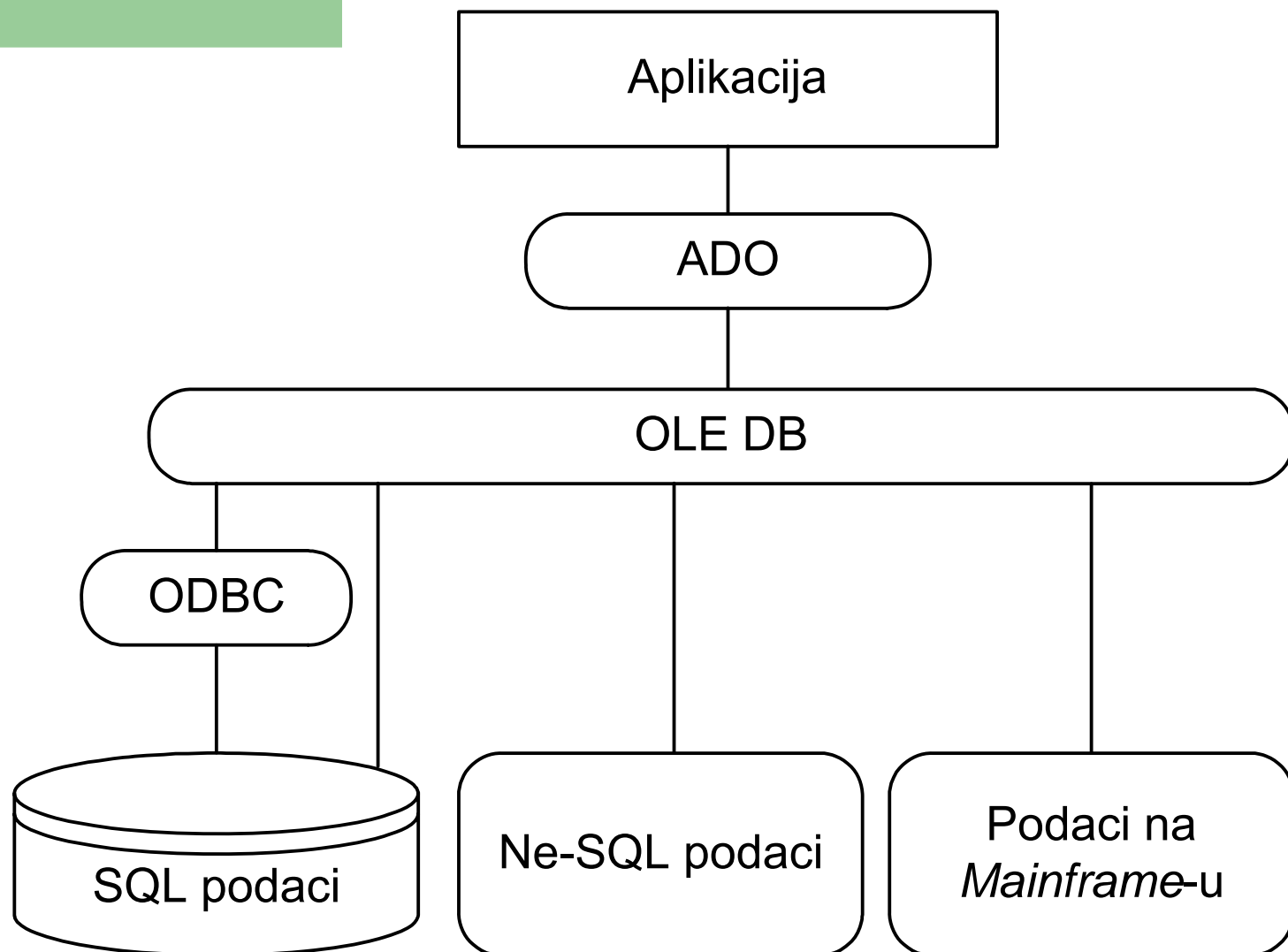
Šta je ODBC?

- ODBC predstavlja biblioteku standardizovanih funkcija (funkcija koje imaju tačno definisano ime, ulazne i izlazne argumente) koje su na raspolaganju programeru da iz programskog koda pristupa bazama podataka.
- ODBC je API za pristup različitim, heterogenim bazama podataka bez izmene izvornog programskog koda.



ODBC arhitektura

ActiveX[®] Data Objects (ADO)



ADO arhitektura

Osnovni objekti ADO komponente su:

- **Connection.** Predstavlja jedinstvenu sesiju ka izvoru podataka. Nad ovim objektom se može pozvati metoda *Execute()* koja direktno izvršava zadatu SQL naredbu.
- **Command.** Ovaj objekat služi za pripremanje i izvršavanje SQL naredbe koja može biti parametrizovana. Objekat Command poseduje kolekciju *Parameters* koja sadrži jedan ili više parametara koji su atributi klase *Parameter*. Vrednosti ovih atributa zadaju se u toku izvršenja programa.
- **Recordset.** Predstavlja objekat ADO komponente koji sadrži rezultate SELECT upita nad bazom podataka i omogućava manipulaciju sa podacima redova iz rezultata.

Osnovne karakteristike ADO komponente

'*** Deklaracija promenljivih

```
Dim objKonekcija As ADODB.Connection
```

```
Dim strKonekcija As String
```

'*** Instanciranje objekta konekcije

```
Set objKonekcija = New ADODB.Connection
```

'*** Postavljanje parametara potrebnih za otvaranje konekcije

```
strKonekcija="Provider=sqloledb;DataSource=DBServer;  
Initial Catalog=Test;User Id=sa;Password=sa;"
```

'*** Otvaranje konekcije

```
objKonekcija.Open strKonekcija
```

Osnovne karakteristike ADO komponente

```
Dim objKomandaRacun As ADO.DB.Command
```

```
Set objKomandaRacun = New ADO.DB.Command
```

```
objKomandaRacun.ActiveConnection = objKonekcija
```

```
objKomandaRacun.CommandType = adCmdText
```

```
objKomandaRacun.CommandText="UPDATE Racun SET
```

```
UkupnaVrednost=" &
```

```
mUkupnaVrednost & " WHERE BrojRacuna=" & brojRacuna
```

***** Izvršenje komande**

```
objKomandaRacun.Execute
```

Osnovne karakteristike ADO komponente

*** Instanciranje i otvaranje recordset-a

```
Dim rstRacun As ADODB.Recordset
```

```
Set rstRacun = New ADODB.Recordset
```

```
Set rstRacun.ActiveConnection = objKonekcija
```

```
rstRacun.CursorLocation = adUseClient
```

```
rstRacun.CursorType = adOpenStatic
```

```
rstRacun.LockType = adLockOptimistic
```

```
SQLupit = "SELECT * FROM Racun"
```

```
rstRacun.Open SQLupit, , , , adCmdText
```

Atribut *CursorType* uzima jednu od četiri moguće vrednosti *adOpenDynamic*, *adOpenKeySet*, *adOpenStatic*, *adOpenForwardOnly*

Osnovne karakteristike ADO komponente

Za pozicioniranje u rekordsetu koriste se metode:
MoveFirst, *MoveLast*, *MoveNext* i *MovePrevious*.

Za pristup vrednostima polja trenutnog rekorda u rekordsetu koristi se kolekcija *Fields*.

```
rstRacun.MoveFirst
```

```
While Not rstRacun.EOF
```

```
    brojRacuna = rstRacun.Fields("BrojRacuna").Value
```

```
    datum = rstRacun.Fields("Datum").Value
```

```
    rstRacun.MoveNext
```

```
Wend
```

Parametrizovane naredbe

- ADO objektni model podržava izvršavanje parametrizovanih SQL naredbi.
- Da bi se označilo mesto gde treba da se ubaci vrednost parametra u vreme izvršenja naredbe koristi se marker parametra (?).
- Za definisanje parametra koristi se objekat *Parameter*.
- Metoda *CreateParameter()* objekta *Command* kao argumente ima naziv parametra, tip podatka polja tabele, tip parametra (ulazni ili izlazni), kao i opciono vrednost parametra.
- Da bi se naglasilo da se radi o parametrizovanoj naredbi atributu *Prepared* se dodeljuje vrednost *True*.

Parametrizovane naredbe - primer

```
Dim brojRacuna As Long
Dim objKomandaRacun As ADODB.Command
Dim param As ADODB.Parameter

Set objKomandaRacun = New ADODB.Command
objKomandaRacun.ActiveConnection = objKonekcija
objKomandaRacun.CommandText="DELETE FROM Racun WHERE
                                BrojRacuna = ?"
objKomandaRacun.CommandType = adCmdText
objKomandaRacun.Prepared = True

Set param= objKomandaRacun.CreateParameter("pBrojRacuna",
    adBigInt, adParamInput)

objKomandaRacun.Parameters.Append param
brojRacuna=893
objKomandaRacun.Parameters("pBrojRacuna").Value=brojRacuna

objKomandaRacun.Execute
```

Poziv procedura baze podataka - primer

```
Set objKomandaRacun = New ADODB.Command
objKomandaRacun.ActiveConnection = objKonekcija
objKomandaRacun.CommandText = "IzracunajIznos"
objKomandaRacun.CommandType = adCmdStoredProc

Set param= objKomandaRacun.CreateParameter("pBrojRacuna",
                                             adBigInt, adParamInput)

objKomandaRacun.Parameters.Append param
objKomandaRacun.Parameters("pBrojRacuna").Value = 750

Set param= objKomandaRacun.CreateParameter("pIznos",
                                             adCurrency, adParamOutput)
objKomandaRacun.Parameters.Append param

objKomandaRacun.Execute
```


Ažuriranje tabele baze podataka preko rezultujućeg seta - primer

```
Set rstRacun = New ADODB.Recordset
Set rstRacun.ActiveConnection = objKonekcija
rstRacun.CursorLocation = adUseClient
rstRacun.CursorType = adOpenStatic
rstRacun.LockType = adLockOptimistic

rstRacun.Open "SELECT * FROM Racun", , , , adCmdText

rstRacun.AddNew
rstRacun("BrojRacuna") = brojRacuna
rstRacun("Datum") = Datum
rstRacun("NazivKupca") = NazivKupca
rstRacun("Napomena") = Napomena
rstRacun.Update
```

Upravljanje transakcijama

*** Otpočinjanje transakcije

```
objKonekcija.BeginTrans
```

... izvršavanje SQL naredbi ...

*** Potvrđivanje transakcije

```
objKonekcija.CommitTrans
```

ili

*** Poništavanje transakcije

```
objKonekcija.RollbackTrans
```

Perzistentnost rekordseta

***** Snimanje sadržaja rekordseta u XML formatu**

```
rstSporniRacuni.Save "C:\Temp\SporniRacuni.xml", adPersistXML
```

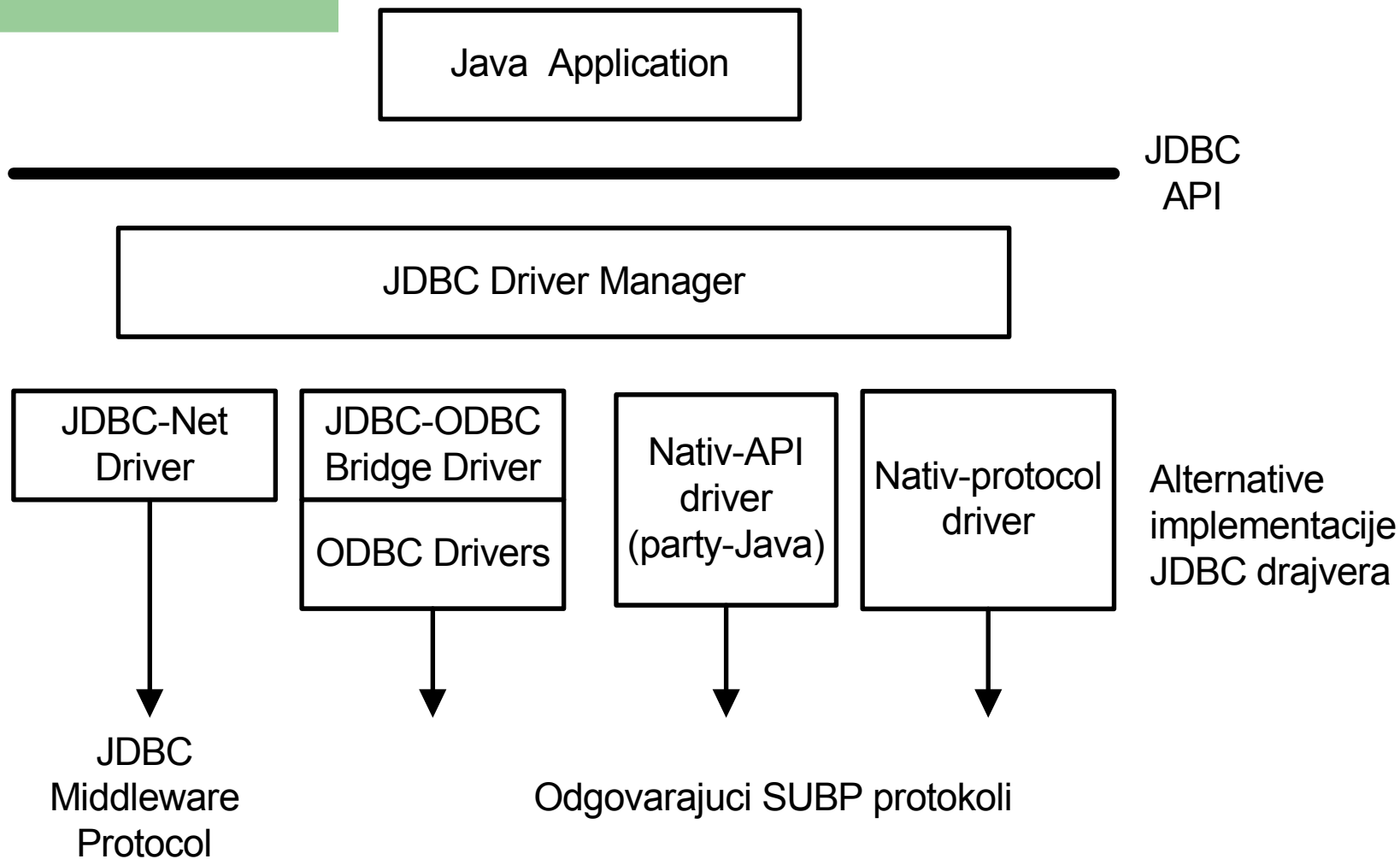
***** Preuzimanje sadržaja XML datoteke u novi rekordset**

```
rstRacuni.Open "C:\Temp\SporniRacuni.xml"
```

Java Database Connectivity (JDBC)

Šta je JDBC?

- JDBC (Java Database Connectivity) je standardni Java interfejs za povezivanje Jave sa relacionim bazama podataka definisan od strane Sun Microsystems.
- JDBC omogućava Java aplikacijama, apletima i servletima pristup bazama podataka kao i drugim tabelarnim izvorima podataka.



JDBC arhitektura

***** Registrovanje ORACLE drajvera**

```
DriverManager.registerDriver(new  
    oracle.jdbc.driver.OracleDriver());
```

***** Uspostavljanje konekcije sa bazom podataka**

```
Connection conn = DriverManager.getConnection(  
    "jdbc:oracle:thin:@zmaj:1521:baza",  
    "korisnik", "lozinka");
```

***** Kreiranje naredbe**

```
Statement stmt = conn.createStatement();
```

***** Izvršenje naredbe**

```
stmt.executeUpdate("UPDATE Racun SET ukupanIznos="+ lUkIzn  
    +" WHERE brojRacuna = "+ broj);
```

***** Rad sa rezultujućim setom**

```
ResultSet rs = stmt.executeQuery("SELECT sifraProizvoda,  
    kolicina FROM StavkaRacuna WHERE brojRacuna = " + broj);  
  
while(rs.next())  
{  
    long lSifPrz = rs.getLong(1);  
    BigDecimal lKol = rs.getDouble(2);  
}
```


***** Parametrizovane naredbe**

```
PreparedStatement pst = conn.prepareStatement(
    "INSERT INTO StavkaRacuna (brojRacuna, rbStavke,
        sifraProizvoda, kolicina) VALUES(?,?,?,?)");
    pst.setBigDecimal(1, new BigDecimal(10));
    pst.setLong(2, (long)1);
    pst.setLong(3, (long)55);
    pst.setDouble(4, (double)3*2.5);
pst.executeUpdate();
```

***** Poziv procedura baze podataka**

```
CallableStatement cstmt =
    con.prepareCall("{call izracunajIznos(?, ?)}");
cstmt.registerOutParameter(1, java.sql.Types.TINYINT);
cstmt.registerOutParameter(2, java.sql.Types.DECIMAL);
ResultSet rs = cstmt.executeQuery();
```

***** Ažuriranje tabele baze podataka preko rezultujućeg seta**

```
Statement stmt = con.createStatement(  
    ResultSet.TYPE_SCROLL_SENSITIVE,  
    ResultSet.CONCUR_UPDATABLE);  
  
ResultSet rs = executeQuery("SELECT brojaRacuna, datum,  
    nazivKupca, ukupanIznos as iznos FROM Racun");  
  
rs.moveToInsertRow();  rs.updateInt("brojRacuna", 15);  
    rs.updateDate("datum", '2003-04-01');  
    rs.updateString("nazivKupca", "Mica");  
    rs.updateFloat("iznos", 10.50f);  
  
rs.insertRow();
```

***** Upravljanje transakcijama**

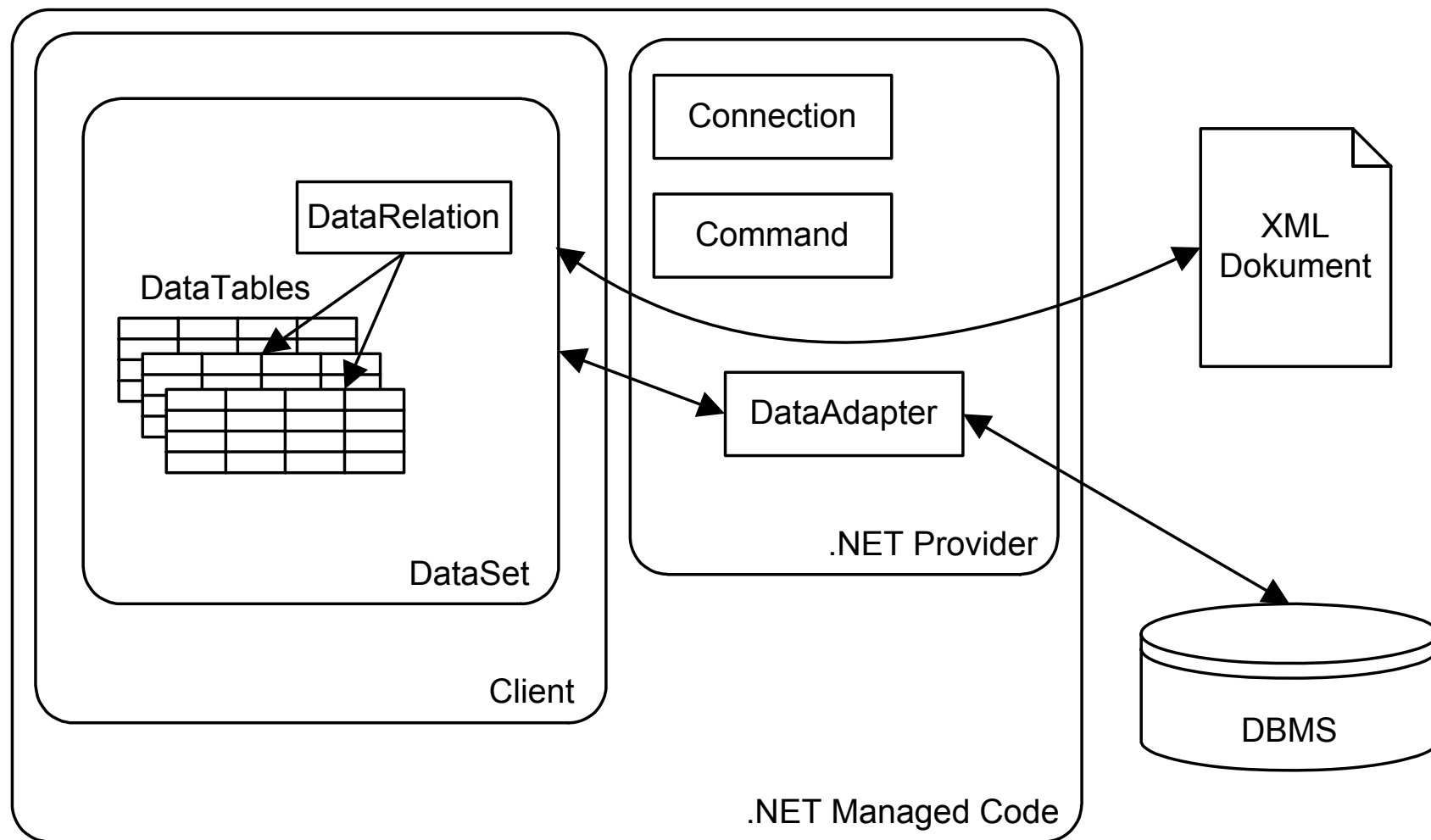
```
conn.setAutoCommit(false);  
...  
    // naredbe  
...  
conn.commit(); // ili conn.rollback();
```

ADO.NET

ADO.NET

- **ADO.NET je dizajniran za korišćenje kako u intranet tako i u Internet aplikacijama**
- **Najprimetnija razlika je između objekata *ADO Recordset-a* i *.NET DataSet-a***
- **Recordset sadrži rezultat jednog upita i najčešće održava otvorenu konekciju ka bazi podataka nad kojom je izvršen upit**
- ***DataSet* može sadržati rezultate više upita, čak iz različitih izvora podataka**
- ***DataSet* je praktično memorijski keš za podatke iz različitih izvora**

ADO.NET



ADO.NET arhitektura

ADO.NET – primer – str. 1

```
using System;  
using System.Collections;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Web;  
using System.Web.SessionState;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Web.UI.HtmlControls;  
using System.Data.SqlClient;  
...
```

ADO.NET – primer – str. 2

```
...
namespace DemoRezervacija
{
    /// <summary>
    /// Summary description for WebForm1.
    /// </summary>
    public class Rezervacija : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.TextBox txt_HotelID;
        protected System.Web.UI.WebControls.TextBox txt_DatumOd;
        protected System.Web.UI.WebControls.TextBox txt_DatumDo;
        protected System.Web.UI.WebControls.TextBox txt_TipSobe;
        protected System.Web.UI.WebControls.TextBox txt_BrojOsoba;
        protected System.Web.UI.WebControls.LinkButton LBtn_Dodaj;
        protected System.Web.UI.HtmlControls.HtmlGenericControl Message;
    }
}
...
```

ADO.NET – primer – str. 3

...

```
#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form
    Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}
private void InitializeComponent()
{
    this.LBtn_Dodaj.Click+=new
    System.EventHandler(this.LBtn_Dodaj_Click);
    this.Load += new System.EventHandler(this.Page_Load);
}
#endregion
```

...

ADO.NET – primer – str. 4

...

```
private void LBtn_Dodaj_Click(object sender, System.EventArgs e)
{
    Message.InnerHtml = "";

    string connstring = "data source=NS1;initial catalog=SrX_db;
    persist security info=False;user id=sa;password=sa;
    workstation id=SMAK;packet size=4096";

    SqlConnection conn = new SqlConnection(connstring);
```

...

ADO.NET – primer – str. 5

...

```
SqlCommand command = new SqlCommand("Dodaj_Rezervaciju", conn);
command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add(new SqlParameter("@DatumOd",
System.Data.SqlDbType.SmallDateTime, 4));
command.Parameters["@DatumOd"].Value = txt_DatumOd.Text;

command.Parameters.Add(new SqlParameter("@DatumDo",
System.Data.SqlDbType.SmallDateTime, 4));
command.Parameters["@DatumDo"].Value = txt_DatumDo.Text;

command.Parameters.Add(new SqlParameter("@HotelID",
System.Data.SqlDbType.Int, 4));
command.Parameters["@HotelID"].Value = txt_HotelID.Text;

command.Parameters.Add(new SqlParameter("@TipSobe",
System.Data.SqlDbType.Int, 4));
command.Parameters["@TipSobe"].Value = txt_TipSobe.Text;

command.Parameters.Add(new SqlParameter("@BrOsoba",
System.Data.SqlDbType.Int, 4));
command.Parameters["@BrOsoba"].Value = txt_BrojOsoba.Text;
```

ADO.NET – primer – str. 6

...

```
command.Connection.Open();
```

```
SqlTransaction trans;
```

```
trans = command.Connection.BeginTransaction();
```

```
command.Transaction = trans;
```

...

ADO.NET – primer – str. 7

```
...
try
{
    command.ExecuteNonQuery();
    trans.Commit();
    Message.InnerHtml = "<b>Rezervacija je dodata!</b>";
}
catch (SqlException sqlex)
{
    if (sqlex.Number == 2627)
        Message.InnerHtml="GRESKA: Zapis vec postoji sa tim
primarnim kljucem!";
    Message.Style["color"] = "red";
    trans.Rollback();
}
...

```

ADO.NET – primer – str. 8

...

```
catch (Exception)
{
    Message.InnerHtml="GRESKA: Ne mozes dodati rekord,
proveri da li si ispravno uneo podatke!";
    Message.Style["color"] = "red";
    trans.Rollback();
}

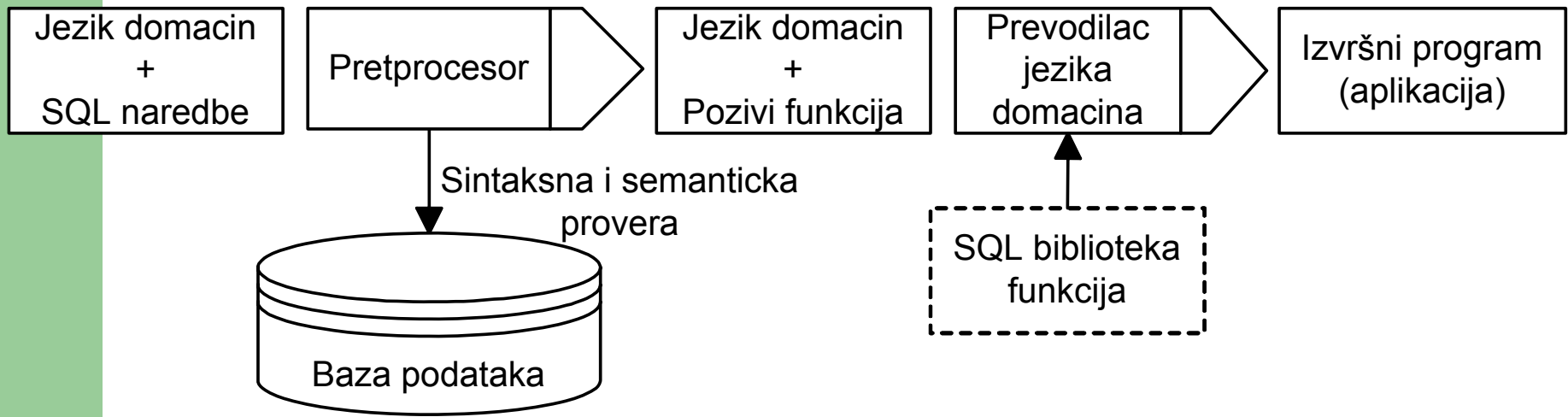
command.Connection.Close();
}
}
}
```

SQL unutar klasičnih programskih jezika:

- Ada, C, Cobol, Fortran, Pascal, PL/I
- Naredbe programskog jezika koje se koriste, tzv. "jezika domaćina" i SQL su "pomešane"
- Kombinovanje naredbi jezika treće generacije i SQL-a omogućava, s jedne strane korišćenje proceduralnih karakteristika jezika treće generacije (sekvenca, selekcija, iteracija, procedura, podprogram), a sa druge strane pristup podacima u bazi podataka pomoću SQL-a.

Programi pisani pomoću ugrađenog SQL-a omogućavaju: :

- Istovremenu obradu podataka u bazi podataka i podataka u datotekama operativnog sistema
- Korišćenje rekurzivnih podprograma, najčešće za obradu hijerarhijskih ili mrežnih struktura podataka (organizaciona struktura, sastavnica proizvoda, itd.)
- Korišćenje dinamičkog SQL-a (SQL naredbe se definišu u vreme izvršavanja programa, a ne u vreme prevođenja)
- Korišćenje DDL (Data Definition Language) naredbi za kreiranje i izmenu objekata baze podataka



Procesiranje aplikacionog programa sa SQL-om unutar programskih jezika