

# MICROSOFT .NET ADO.NET

Prof. Dr. Eduardo Chagas de Oliveira

[edu\\_ufu@yahoo.com](mailto:edu_ufu@yahoo.com)

*“Feliz aquele que transfere o que sabe e aprende o que ensina”.*

CORA CORALINA



# Acesso a Dados ADO.NET



# ADO.NET

## ■ Objetivos:

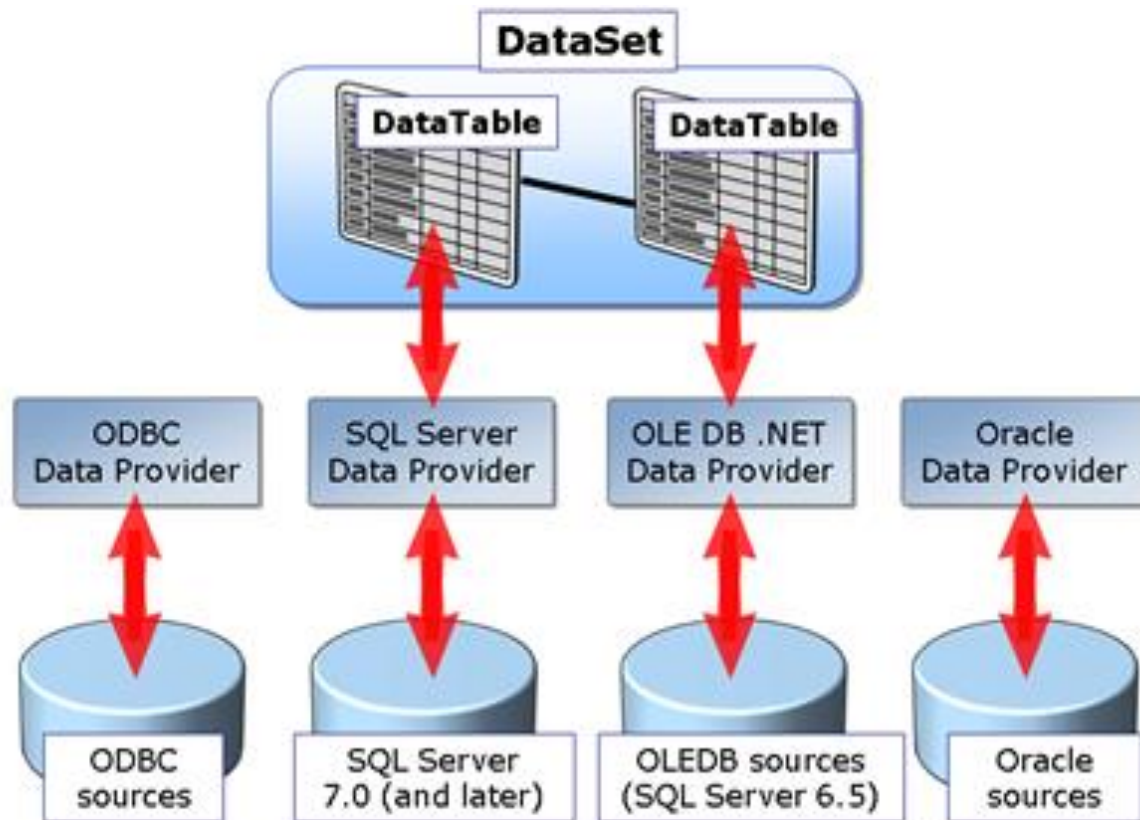
- ❑ Introduzir conhecimentos de Banco de Dados Relacionais
- ❑ Introduzir conceitos de softwares e hardwares
- ❑ Capacitar a manutenção de Banco de Dados
- ❑ Capacitar o desenvolvimento com Banco de Dados
- ❑ Trabalhar problemas que exigem facilidade e rapidez no acesso à informação

# ADO.NET

- O que é ADO.NET?
  - É uma coleção de classes, interfaces, estruturas e tipos de dados que gerenciam o acesso a dados de um banco de dados relacional dentro do .NET *Framework*
  - ADO.NET é uma evolução do ADO
  - Biblioteca de acesso a banco de dados

# ADO.NET

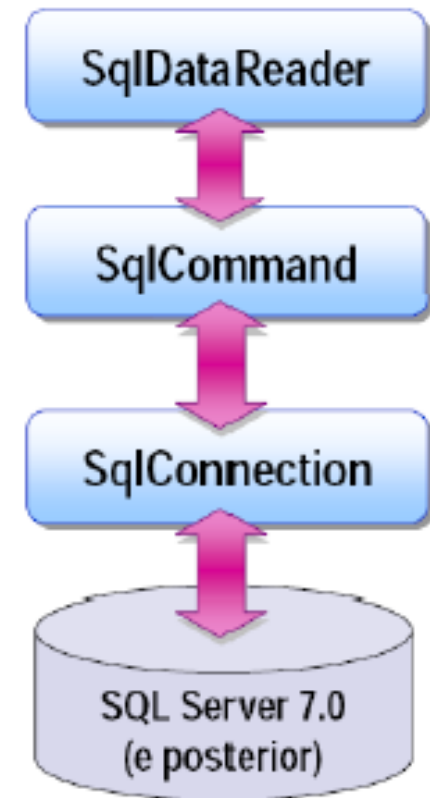
## ■ Modelo de Objetos ADO.NET



# ADO.NET

## ■ Cenário Conectado

- Um cenário conectado é aquele no qual os utilizadores estão permanentemente ligados à base de dados



# ADO.NET

- Cenários Conectados – Criando DataReader
  - ❑ 1. Crie a abra a conexão com o banco de dados
  - ❑ 2. Crie um objeto Command
  - ❑ 3. Crie um DataReader a apartir do Command
  - ❑ 4. Chame o método Execute Reader
  - ❑ 5. Use o DataReader
  - ❑ 6. Feche o objeto DataReader
  - ❑ 7. Feche o objeto Connection

# ADO.NET

## ■ Exemplo - Conectado

### □ 1. Abrir a Conexão

```
SqlConnection conn = new SqlConnection("SERVER=SQLSERVER; INTEGRATED  
SECURITY = TRUE; INITIAL CATALOG=ISEP");  
conn.Open();
```

### □ 2. Executar o Comando

```
SqlCommand cmd = new SqlCommand();  
cmd.Connection = conn;  
cmd.CommandText = "SELECT cod, descricao FROM detalhes WHERE  
zona=42";  
reader = cmd.ExecuteReader();
```



# ADO.NET

## ■ Exemplo - Conectado

### □ 3. Processar Linhas no Reader

```
while(reader.Read())  
{  
    int cod = (int)reader[0];  
    cmbDescricao.Items.Add(reader[1].ToString());  
}
```

### □ 4. Fechar o Reader

```
if(!reader.IsClosed)  
    reader.Close();
```

### □ 5. Fechar a Conexão

```
conn.Close();
```

# ADO.NET

## ■ ADO.NET – Exemplo - Conectado

### □ Métodos do SqlCommand

#### ■ ExecuteReader

- Envia o comando para a conexão e retorna o resultado em um SqlDataReader

#### ■ ExecuteNonQueryReader

- Executar comandos como INSERT, UPDATE, DELETE

#### ■ ExecuteScalar

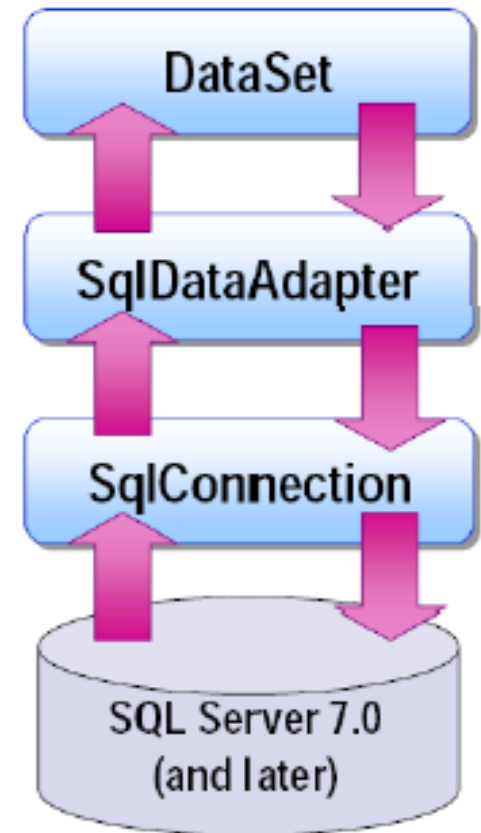
- Executar um comando e retorna um único valor.

#### ■ ExecuteXmlReader

- Executar um comando e retorna um XML

# ADO.NET

- **Cenário Desconectado**
  - Um cenário conectado é aquele no qual um sub-conjunto de dados pode ser copiado e modificado independentemente e mais tarde modificado as alterações podem ser introduzidas de novo na base de dados



# ADO.NET

## ■ DataSet

- ❑ É um banco de dados relacional
  - Tabelas
  - Relacionamentos
- ❑ Tabelas (DataTable)
  - Colunas
  - Linhas
  - Chaves Primárias
- ❑ O Dataset é uma representação desconectada dos dados da memória

# ADO.NET

## ■ Exemplo Desconectado

### □ 1. Abrir a Conexão

```
SqlConnection conn = new SqlConnection("SERVER=SQLSERVER; INTEGRATED  
SECURITY = TRUE; INITIAL CATALOG=ISEP");  
conn.Open();
```

### □ 2. Preencher o DataSet

```
System.Data.DataSet ds = new System.Data.DataSet();  
System.Data.Sql.SqlDataAdapter da =  
    new System.Data.Sql.SqlDataAdapter();  
SqlCommand cmd = new SqlCommand();  
cmd.CommandText = "SELECT * FROM [DETALHES]";  
cmd.Connection = conn;  
da.SelectCommand = cmd;  
da.Fill(ds);
```

# ADO.NET

## ■ Exemplo Desconectado

### ❑ 3. Fechar a Conexão

```
conn.Close();
```

### ❑ 4. Processar o DataSet

```
foreach(DataRow r in ds.Tables[0].Rows)  
    r["preco"] = r["preco"] * 1.05;
```

### ❑ 5. Abrir a Conexão

```
conn.Open();
```

# ADO.NET

## ■ Exemplo Desconectado

### ❑ 6. Atualizar a fonte de dados

```
System.Data.Sql.SqlDataAdapter da =  
    new System.Data.Sql.SqlDataAdapter(  
        "SELECT * FROM [DETALHES]", conn);  
System.Data.Sql.SqlCommandBuilder cb =  
    new System.Data.Sql.SqlCommandBuilder(da);  
da.Update(ds);
```

### ❑ 7. Fechar a conexão

```
conn.Close();
```

# ADO.NET

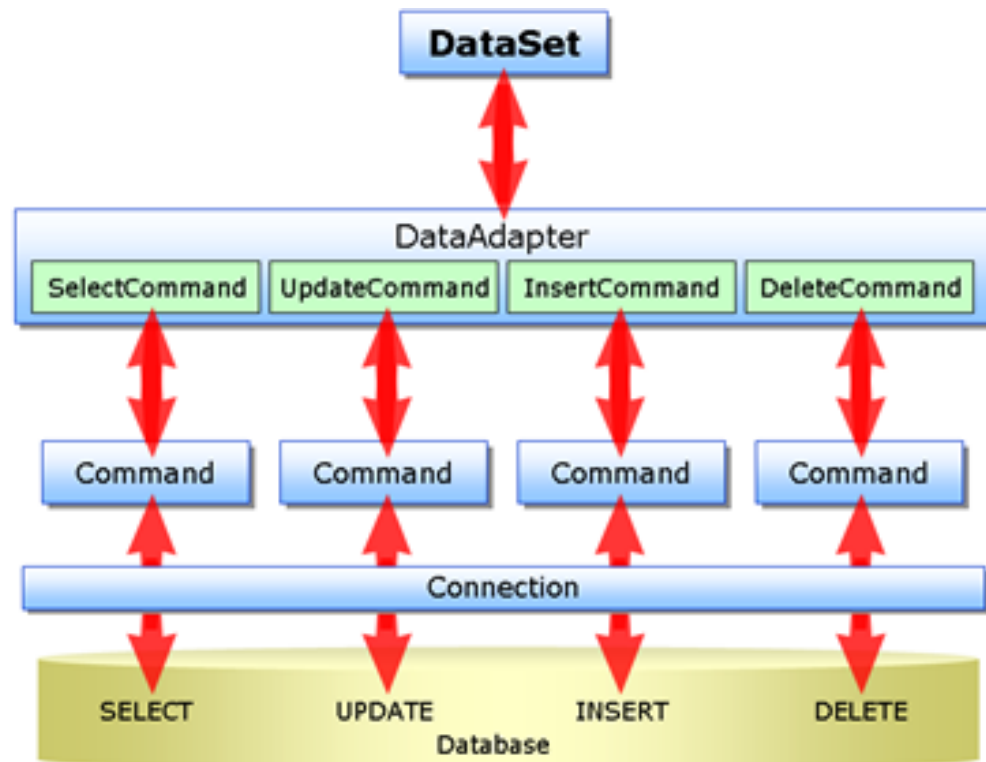
## ■ DataSet x DataReader

<b>DataSet</b>	<b>DataReader</b>
Leitura/Escrita para acesso aos dados	Somente leitura
Inclui múltiplas tabelas de diferentes data sources.	Baseado em uma instrução SQL
Acesso mais lento	Acesso mais rápido
Desconectado	Conectado
Gerado pelo Visual Studio .NET	Manualmente codificado.



# ADO.NET

- ADO.NET – Modelo DataAdapter (DataSet)





# Exemplo ADO.NET Conectado

# Exemplo Completo Conectado

## ■ Exemplo – Cadastro de Alunos

### ❑ 1º PASSO. Criar o Banco de Dados

❑ 1. Clique no menu Iniciar | Programas | Microsoft SQL Server 2005 | Microsoft SQL Server Management Studio Express.

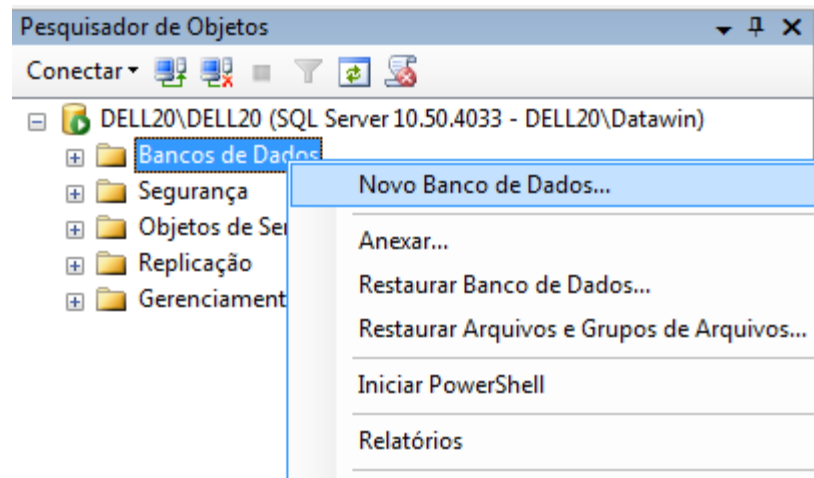
❑ 2. Clique no botão Conectar



# ADO.NET

## ■ Exemplo – Cadastro de Alunos

- ❑ 3. Clique com o botão direito na pasta Database e escolha New Database
- ❑ 4. Dê um nome para o seu banco de dados de db\_curso e clique em OK
- ❑ 5. Na pasta Table, clique com o botão direito e selecione New Table ...



# ADO.NET

## ■ Exemplo – Cadastro de Alunos

- ❑ 6. Dê um nome para a sua tabela de tb\_aluno com os campos
  - cod\_ra int (PK),
  - nom\_aluno varchar(250) (not null)

DELL20\DELL20.db...so - dbo.Table_1* SQLQuery2.sql - SR...rd_0000 (			
	Nome da Coluna	Tipo de Dados	Permitir Nulos
▶	cod_ra	int	<input type="checkbox"/>
	nom_aluno	varchar(250)	<input type="checkbox"/>
			<input type="checkbox"/>

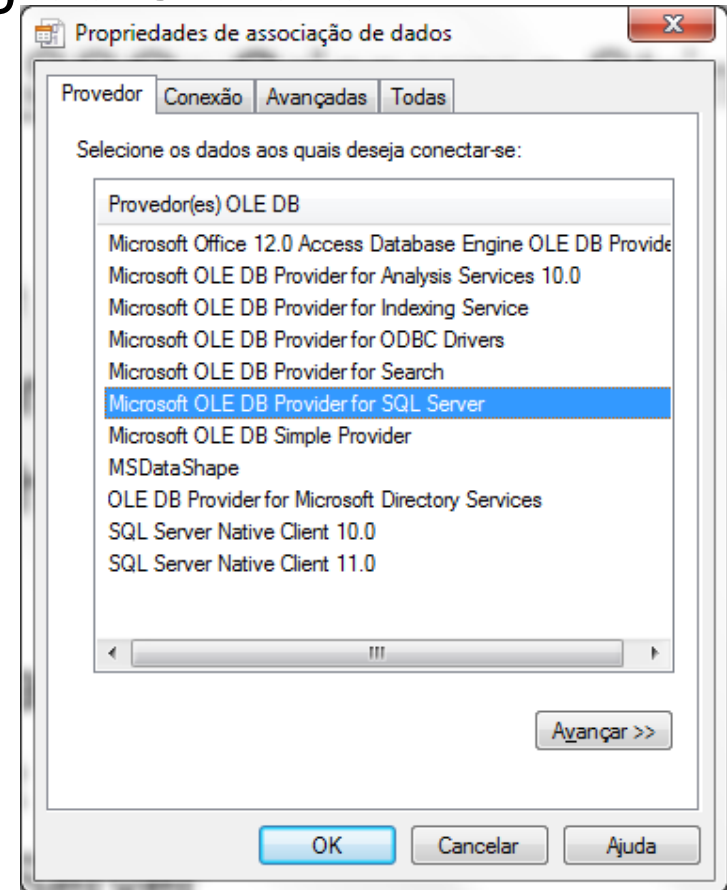
# ADO.NET

## ■ Exemplo - Cadastro de Alunos

### ■ 2º PASSO: Criar uma String de Conexão

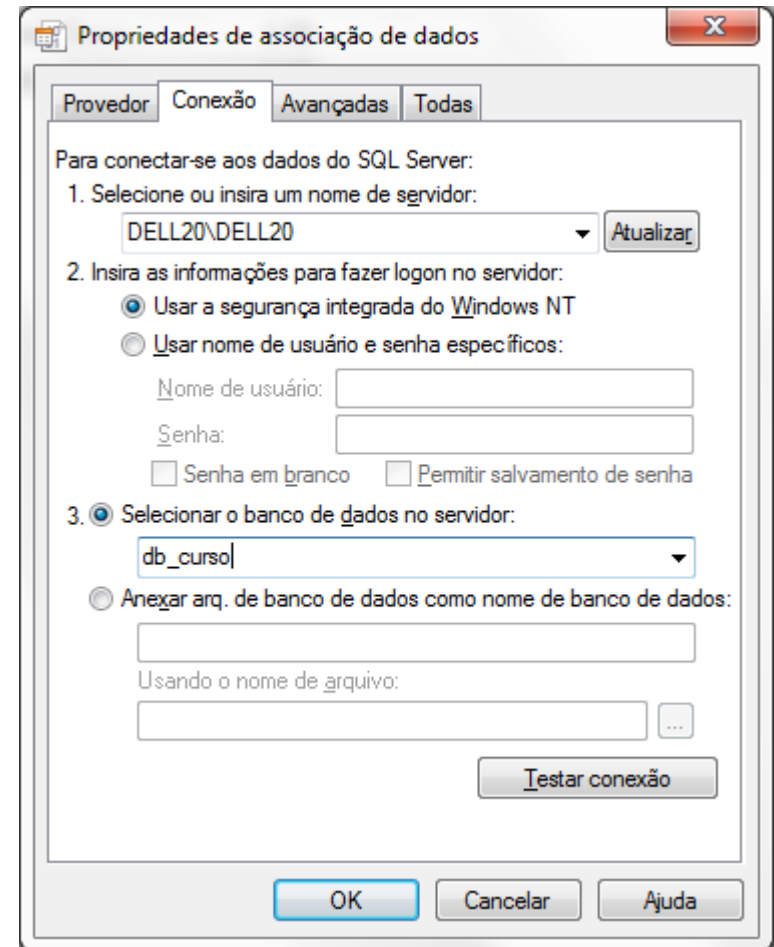
■ 1. Crie um arquivo de nome minhaConexao.txt e renomeie para minhaConexao.udl.

■ 2. Abra o arquivo e escolha Microsoft OLE DB Provider for SQL Server



# ADO.NET

- Cadastro de Alunos – Criando uma string de Conexão
  - 3. Digite o nome do servidor (DELL20\DELL20 ou localhost)
  - 4. Escolha o banco de dados (db\_curso) e teste a conexão.



# ADO.NET

## ■ Cadastro de Alunos – Criando uma string de Conexão

- ❑ 5. Abra o arquivo no bloco de notas e você terá a seguinte string de conexão:

**[oledb]**

**; Everything after this line is an OLE DB initstring**

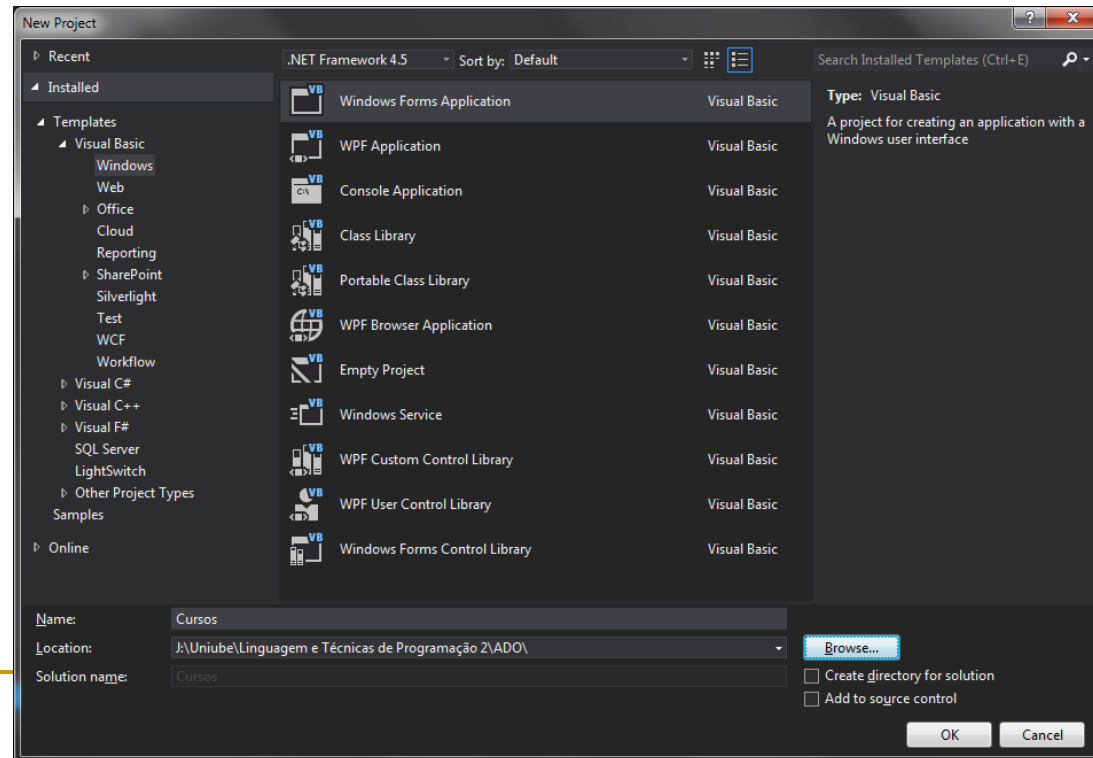
**Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist  
Security Info=False;Initial Catalog=db\_curso;Data  
Source=DELL20\DELL20**



# ADO.NET

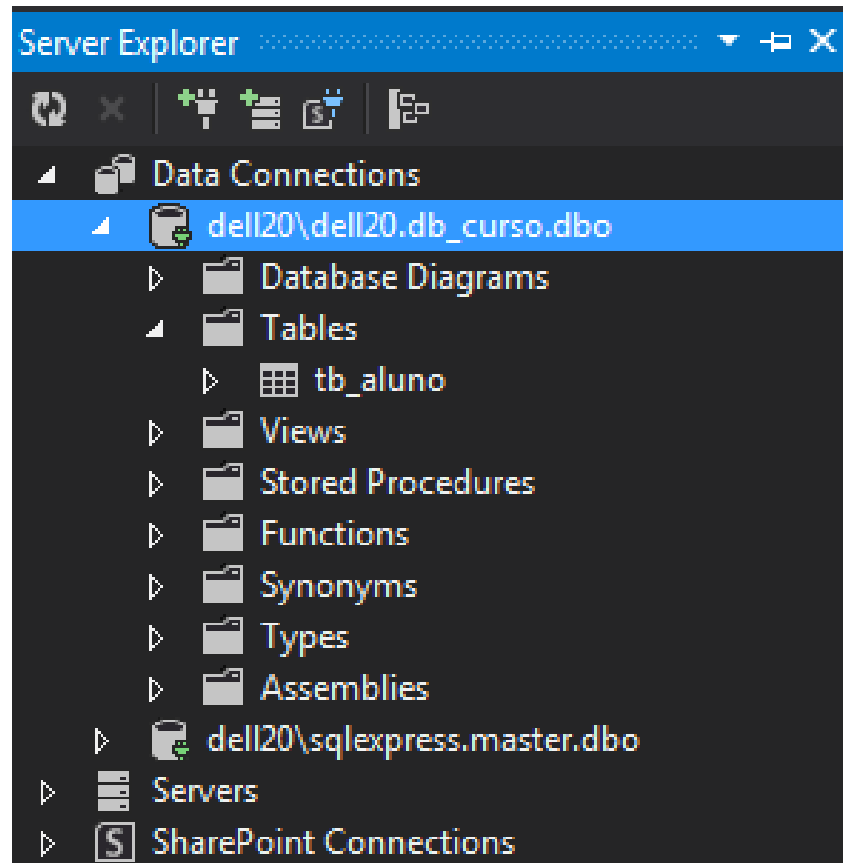
## ■ Exemplo – Cadastro de Alunos

- ❑ 3º PASSO: Crie seu projeto
- ❑ 1. Crie um projeto no Visual Studio para conexão com o Banco de Dados, com nome Cursos



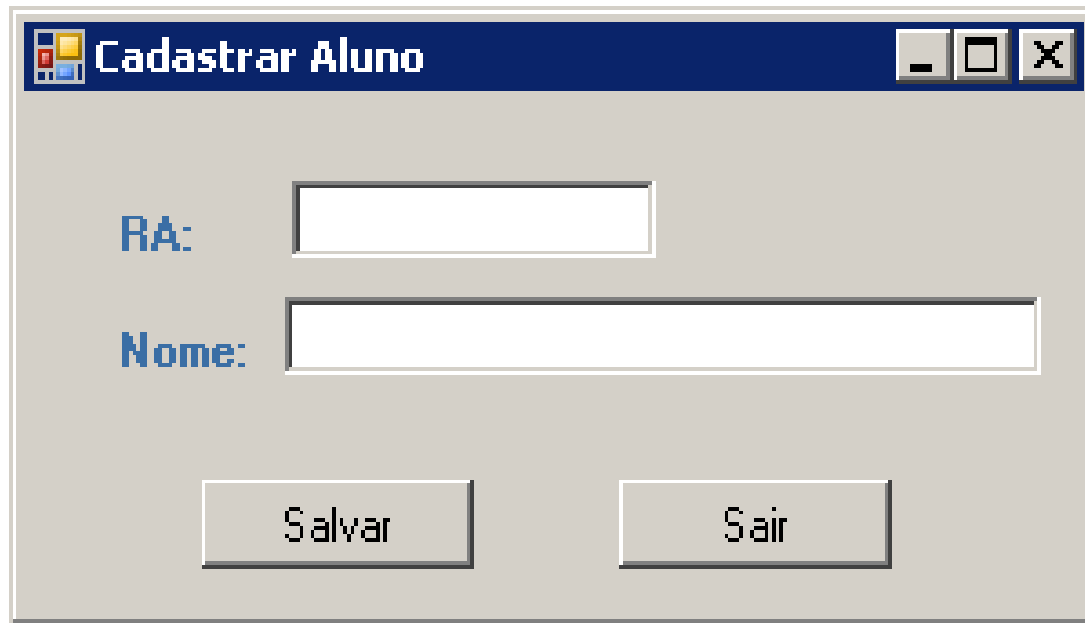
# ADO.NET

- ❑ 2. Crie e verifique a conexão na visão “Server Explorer” ou “Data Sources”



# ADO.NET

- ❑ 2. Crie um formulário para cadastro de alunos



The image shows a screenshot of a Windows application window titled "Cadastrar Aluno". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main area of the window is light gray and contains two text input fields. The first field is preceded by the label "RA:" in blue text. The second field is preceded by the label "Nome:" in blue text. Below the input fields, there are two buttons: "Salvar" (Save) on the left and "Sair" (Exit) on the right. Both buttons have a light gray background and a thin black border.

# ADO.NET

## ■ Exemplo – Cadastro de Alunos

### ■ 4º PASSO: Crie os métodos para acesso ao BD

```
private void btnIncluir_Click(object sender, EventArgs e)
{
    try {
        // pega os campos da tela
        string nomeAluno = txtNome.Text;
        string ra = txtRA.Text;
        // cria os objetos de conexão com o banco
de dados
        OleDbCommand executaComandoSQL = new
OleDbCommand();
        OleDbConnection conexao = new
OleDbConnection();
        string comandoSQL = null;
        string stringConexao = null;
```

# ADO.NET

## ■ Exemplo – Cadastro de Alunos

- Cont. 4º PASSO: Crie os métodos para acesso ao BD

```
//define a string de conexao
stringConexao =
"Provider=SQLNCLI11.1;Integrated Security=SSPI;Initial
Catalog=db_curso;Data Source=DELL20";
//armazena a string de conexao no objeto
conexao.ConnectionString = stringConexao;
//abre a conexao
conexao.Open();
//monta o comando sql
comandoSQL = "INSERT INTO tb_aluno" +
              "(cod_ra, nom_aluno) " +
              "VALUES (" +
                  ra + ", '" +
                  nomeAluno +
                  "' )";
```

# ADO.NET

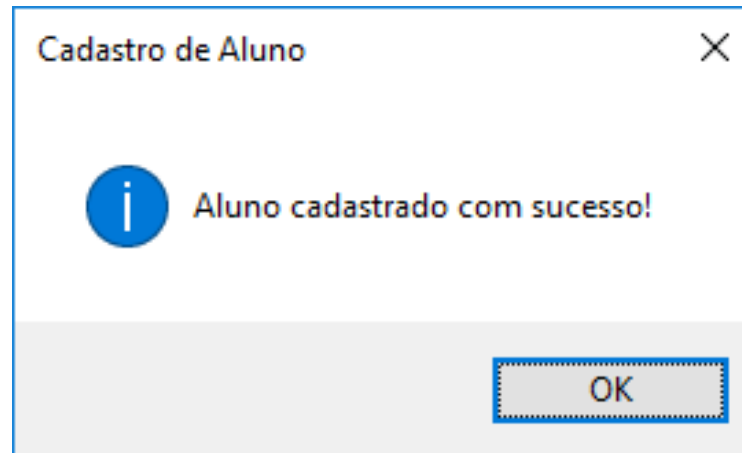
- Exemplo – Cadastro de Alunos
  - Cont. 4º PASSO: Crie os métodos para acesso ao BD

```
        executaComandoSQL.Connection = conexao;
        executaComandoSQL.CommandText =
comandoSQL;

        //Executa o comando sql
        executaComandoSQL.ExecuteNonQuery();
        MessageBox.Show("Aluno cadastrado com
sucesso!", "Cadastro de Aluno", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        conexao.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

# ADO.NET

- Exemplo – Cadastro de Alunos
  - ❑ 5º PASSO: Teste sua tela!!!





# Exemplo ADO.NET Desconectado

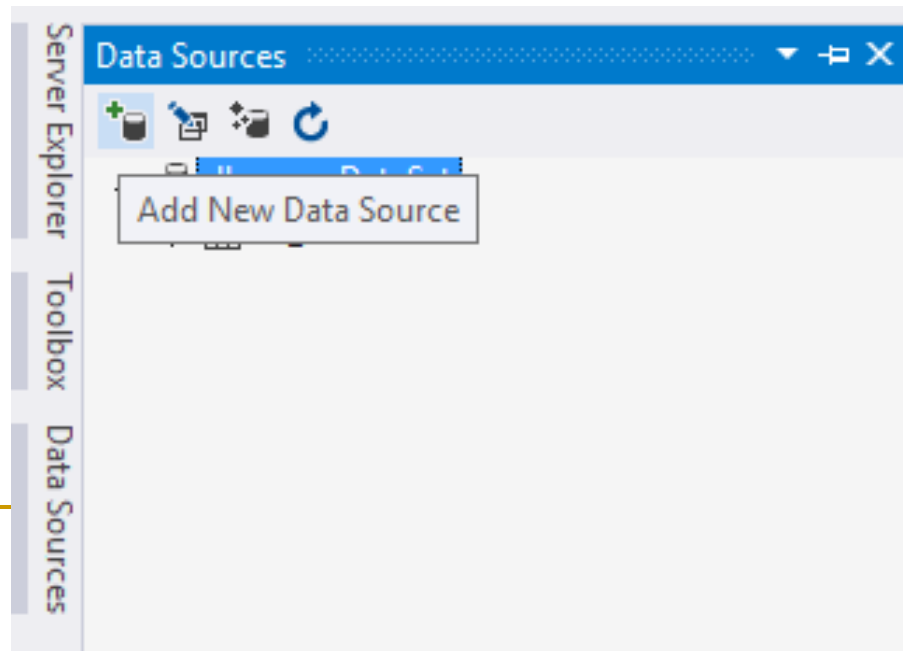


# ADO.NET

- Exemplo – Cadastro de Alunos
  - ❑ 1º PASSO: Criar o banco de dados (feito)
  - ❑ 2º PASSO: Crie o arquivo para configuração da **String de conexão**, chamado app.config
    - 1. Clique com o botão direito sobre o projeto e escolha a opção Add | New Item
    - 2. Em General, selecione Application Configuration File
      - ❑ Arquivo app.config

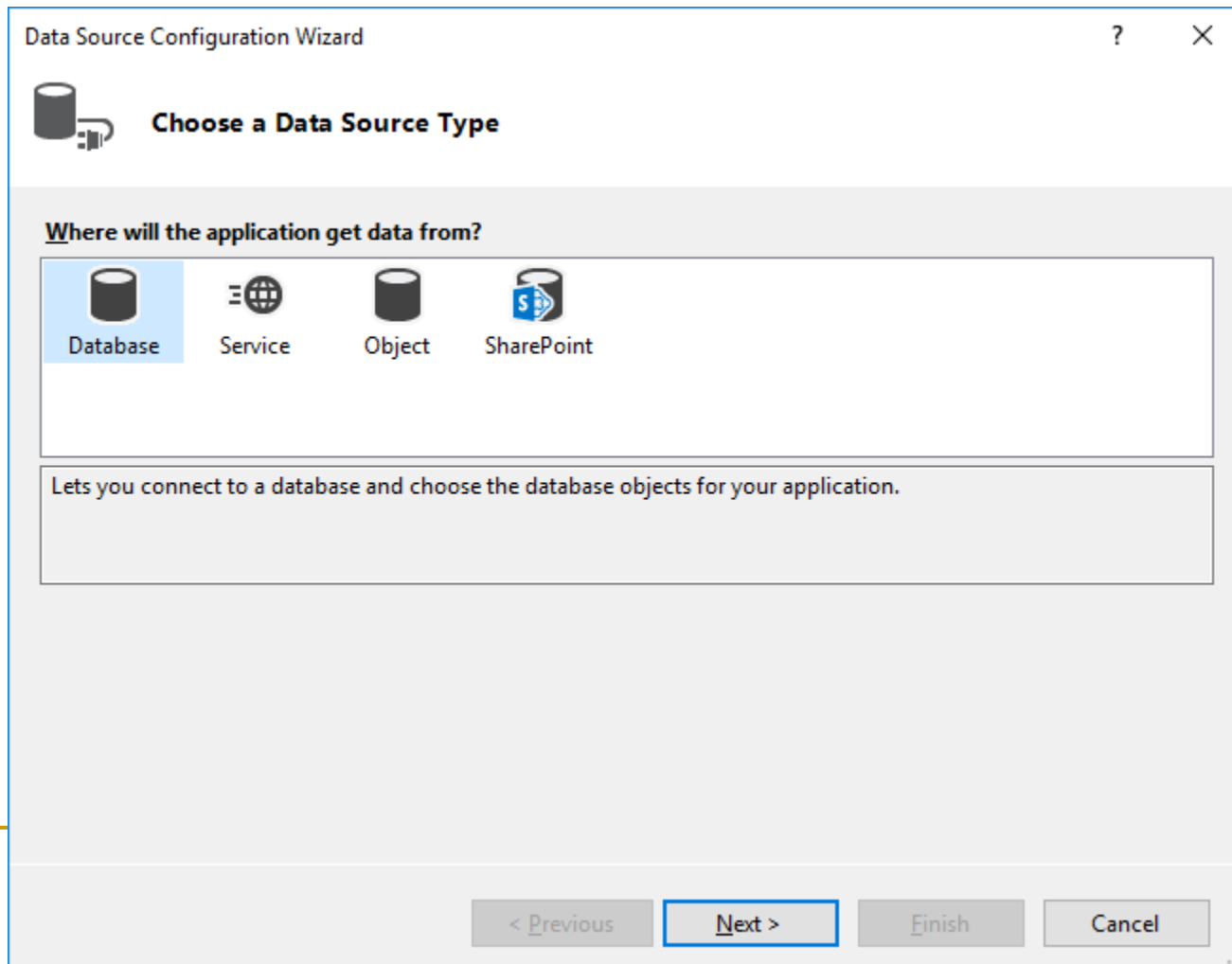
# ADO.NET

- Exemplo – Cadastro de Alunos
  - ❑ 3. Para gerar a string de conexão, crie um Data Source (string de conexão) com o banco de dados
    - Na view “Data Source” clique no botão “Add New Data Source”



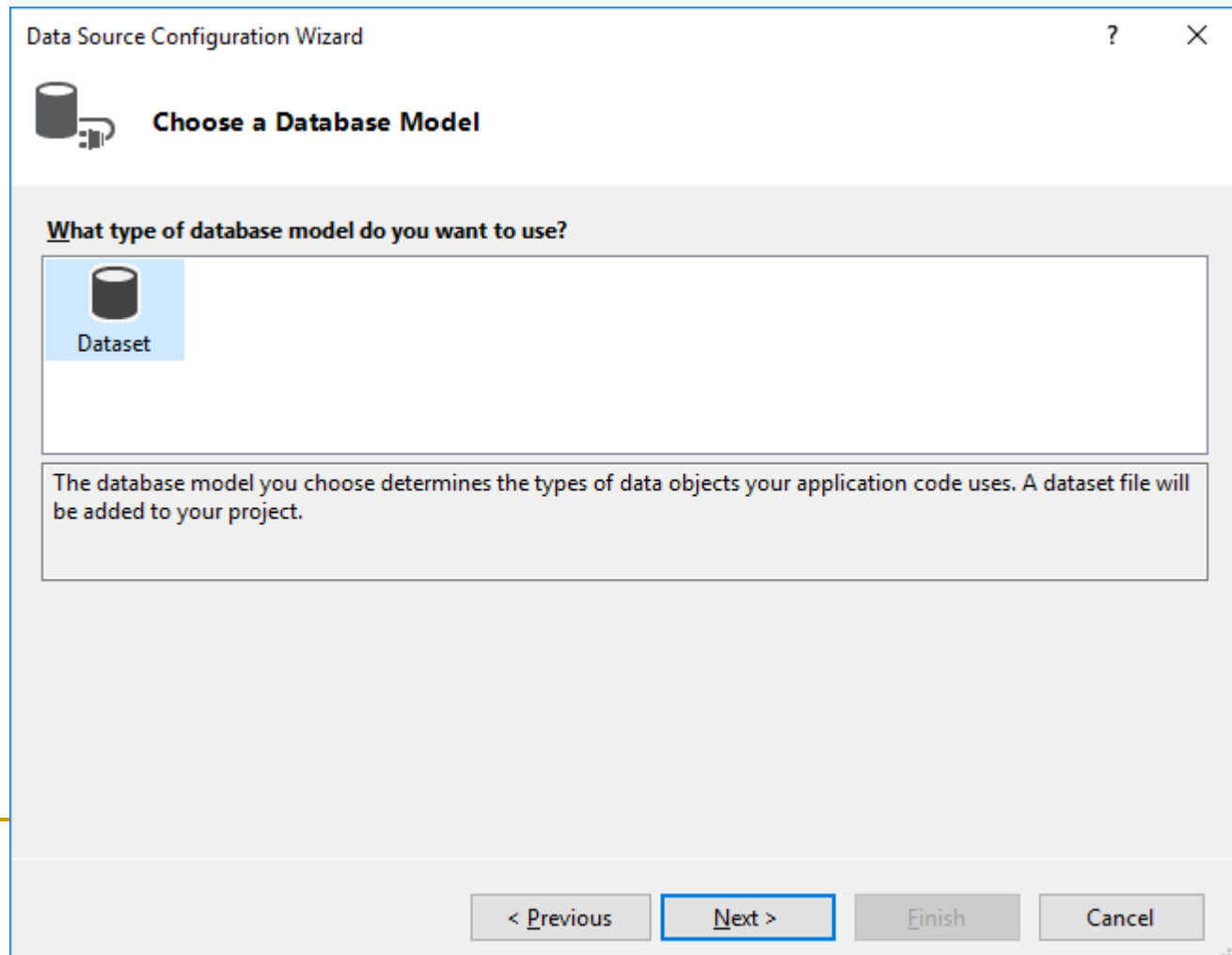
# ADO.NET

- Exemplo – Cadastro de Alunos
  - Selecione “Database”



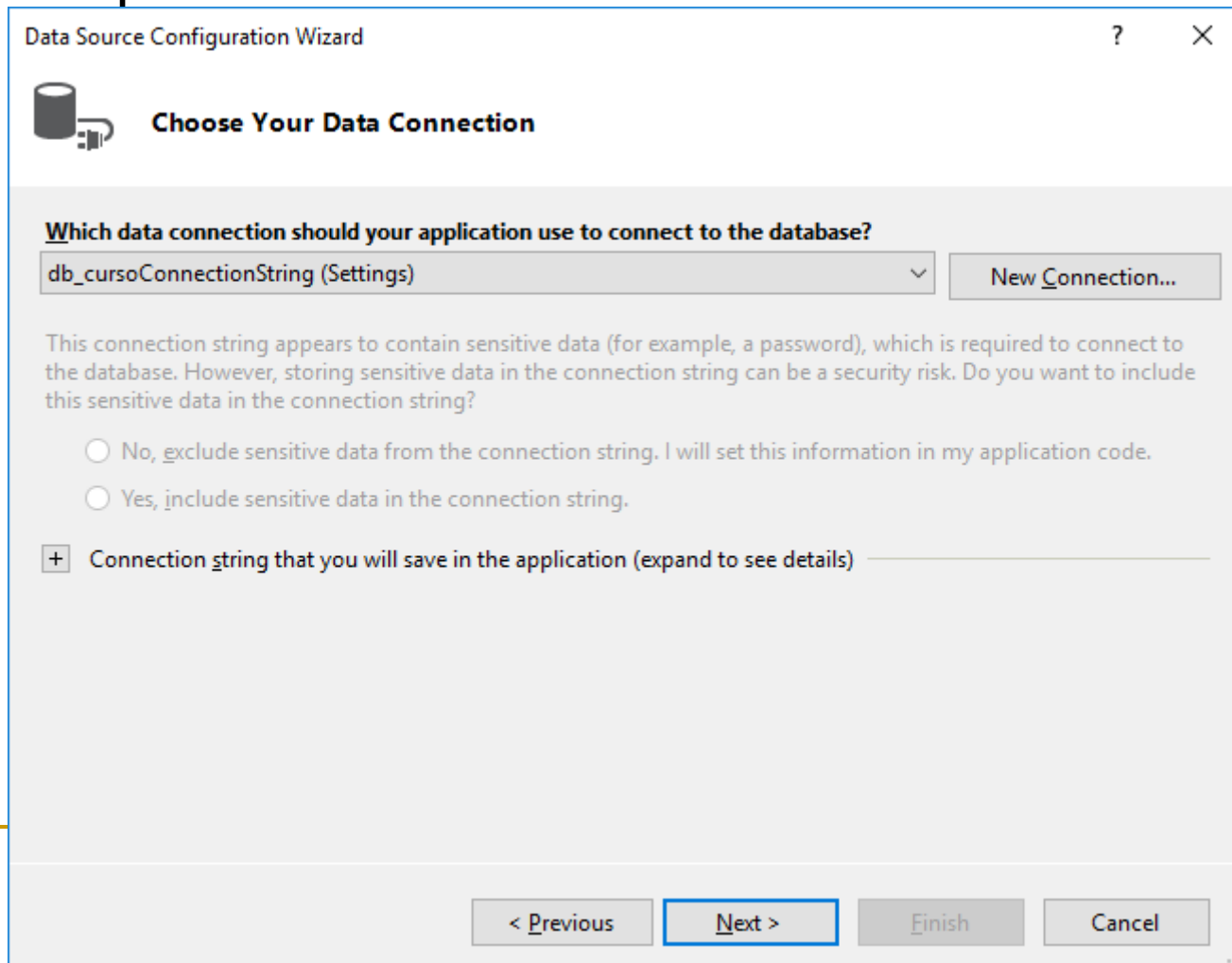
# ADO.NET

- Exemplo – Cadastro de Alunos
  - Selecione “Dataset”



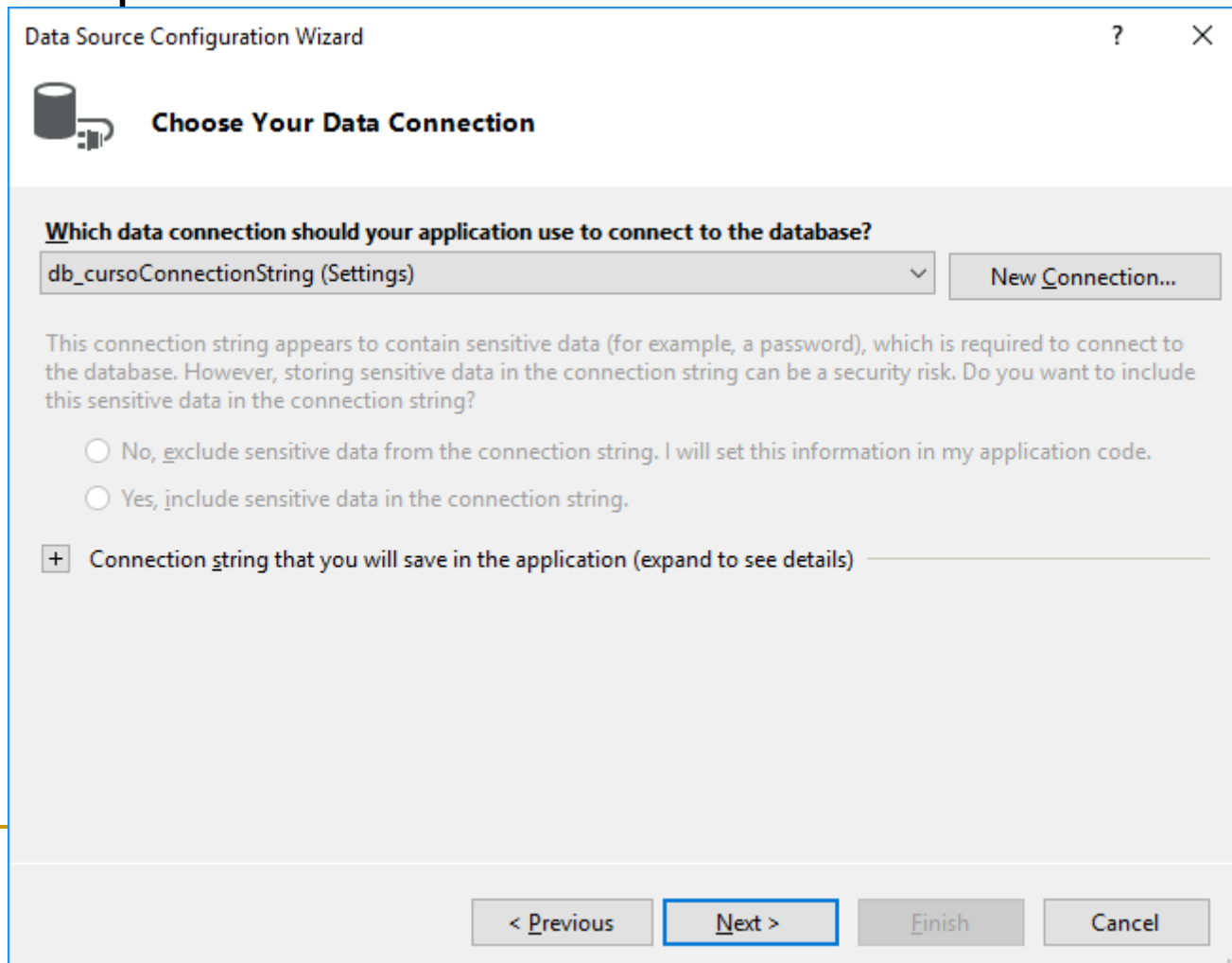
# ADO.NET

- Exemplo – Cadastro de Alunos
  - Clique em “New Connection...”



# ADO.NET

- Exemplo – Cadastro de Alunos
  - Clique em “New Connection...”



# ADO.NET

- Exemplo – Cadastro de Alunos
  - Informe o nome do servidor e base de dados

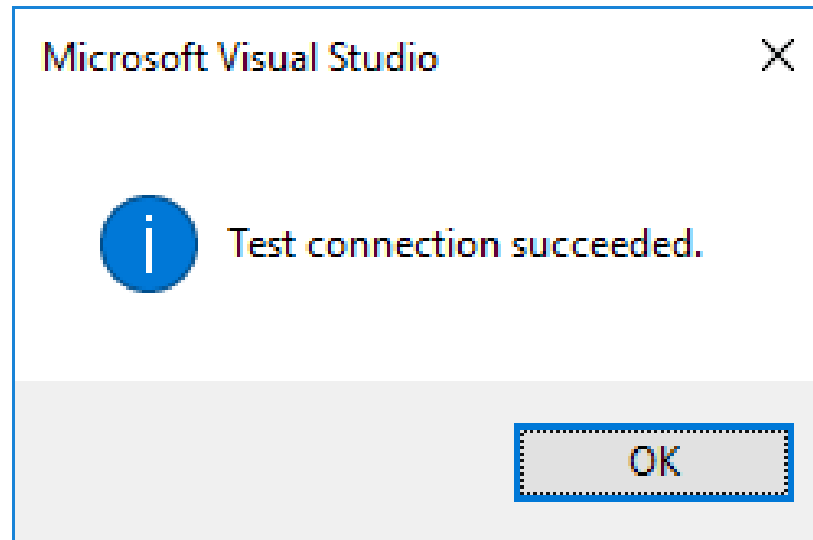
The screenshot shows the 'Add Connection' dialog box with the following fields and options:

- Data source:** Microsoft SQL Server (SqlClient) [Change...]
- Server name:** DELL20 [Refresh]
- Log on to the server:**
  - Authentication:** Windows Authentication
  - User name:** [ ]
  - Password:** [ ]
  - ☐ Save my password
- Connect to a database:**
  - ☒ Select or enter a database name:
    - db\_curso
  - ☐ Attach a database file:
    - [ ] [Browse...]
    - Logical name: [ ]

Buttons at the bottom: Test Connection, OK, Cancel, and an Advanced... button.

# ADO.NET

- Exemplo – Cadastro de Alunos
  - Clique em “Test Connection”

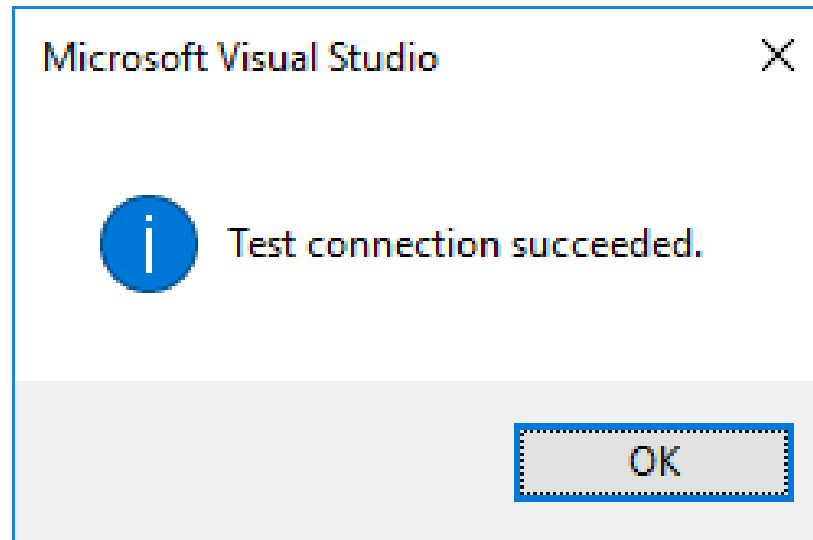




# ADO.NET

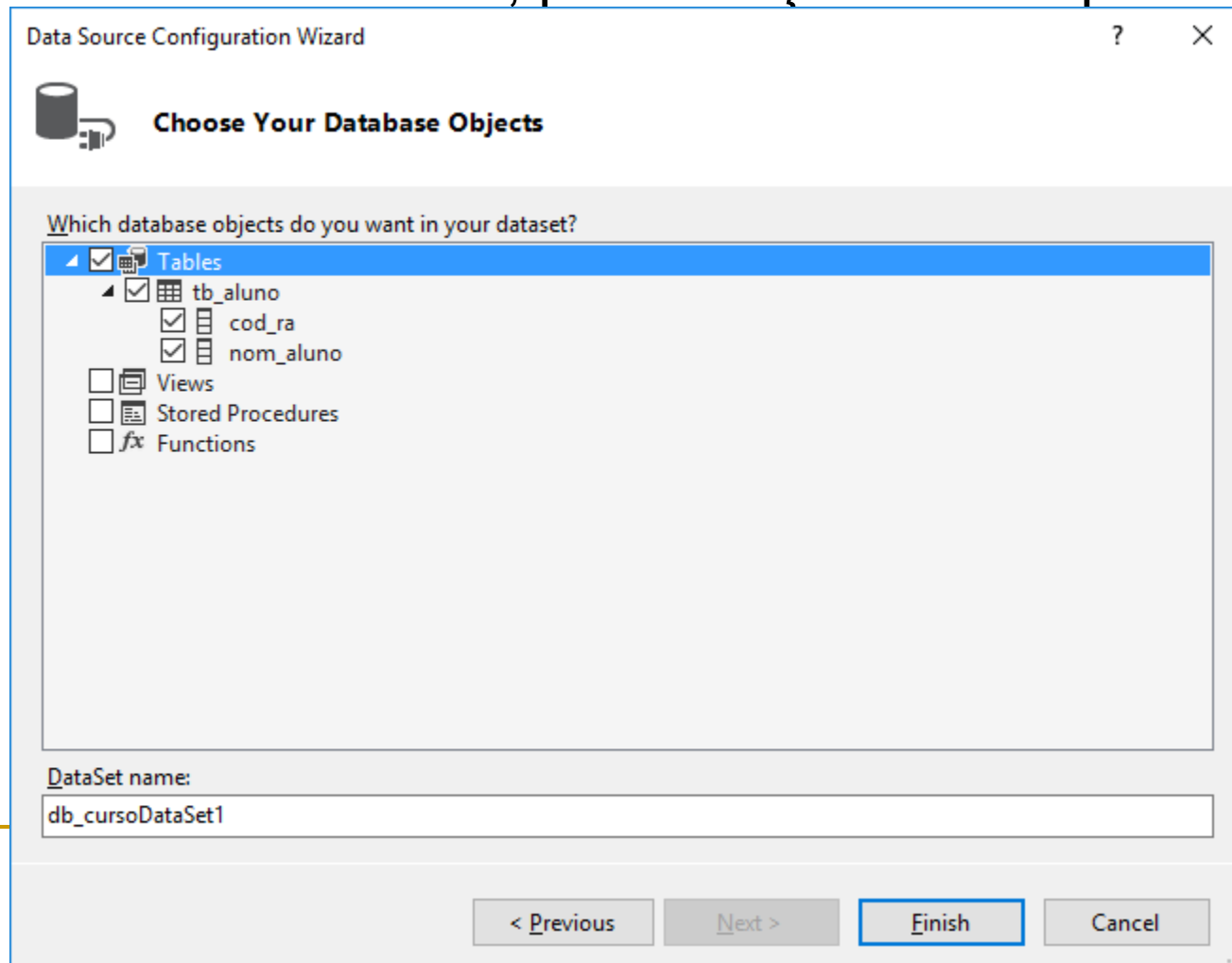
## ■ Exemplo – Cadastro de Alunos

- Clique em “Test Connection” e depois em “OK” e “Next”



# ADO.NET

- Exemplo – Cadastro de Alunos
  - Selecione a tabela, para criação do Adapter



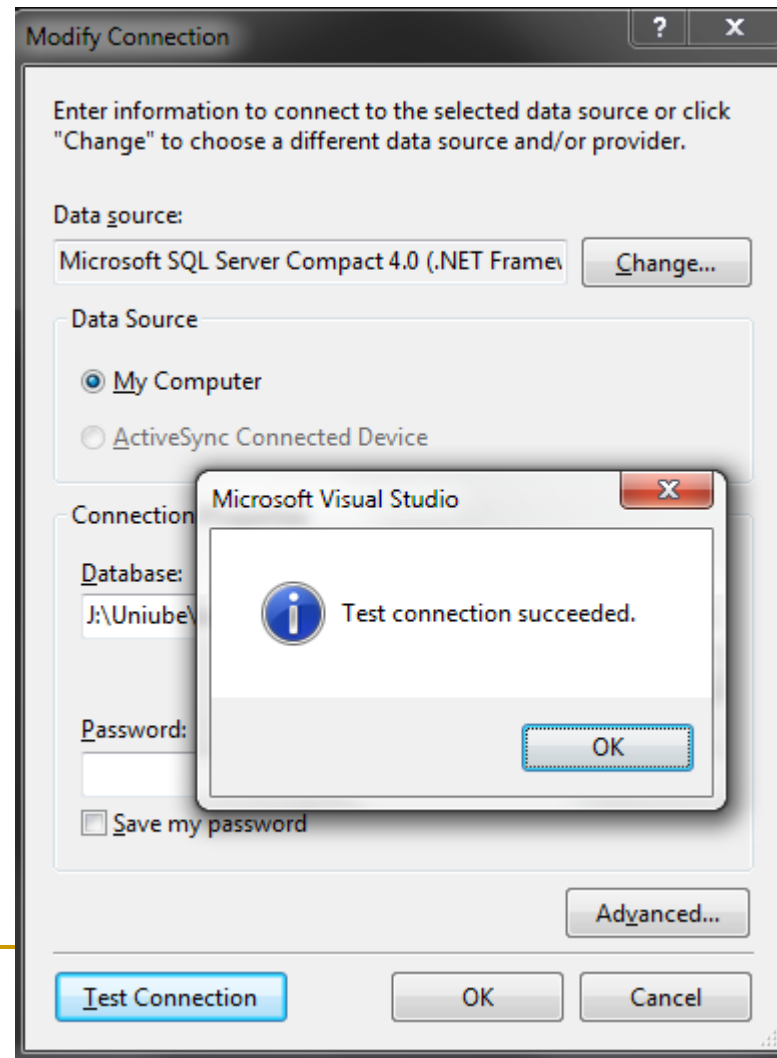
# ADO.NET

## ■ Exemplo – Cadastro de Alunos

- ❑ No menu “Server Explorer”, selecione “Data Connections”
- ❑ Clique com o direito do mouse em “DBCursos” e em “Modify Connection...”
- ❑ Clique em “Test Connection” para testar a conexão com o banco de dados.

# ADO.NET

## ■ Exemplo – Cadastro de Alunos



# ADO.NET

## ■ Exemplo – Cadastro de Alunos

- Clique no arquivo app.config e veja que ele adicionou a seguinte tag

```
<connectionStrings>
```

```
  <add
```

```
    name="Aulas_Programacao_CSharp.Properties.Settings.db_cursoConne  
    ctionString"
```

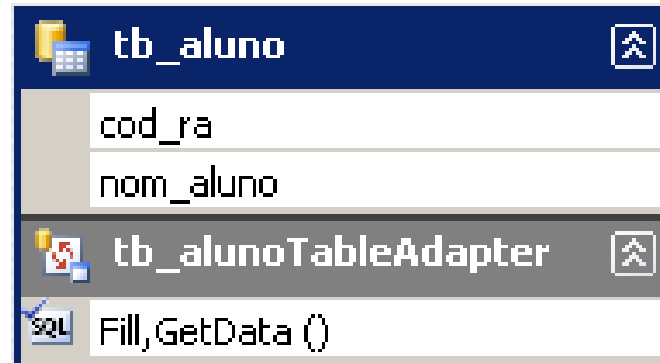
```
      connectionString="Data Source=DELL20;Initial  
      Catalog=db_curso;Integrated Security=True"
```

```
      providerName="System.Data.SqlClient" />
```

```
</connectionStrings>
```

# ADO.NET

- Exemplo – Cadastro de Alunos
  - ❑ 15. Clique no arquivo db\_cursoDataSet.xsd e veja que foi criado um DataSet contendo a tabela tb\_aluno



- ❑ Obs: para usar o DataSet criado no seu código, adicione

**Imports** Cursos.db\_cursoDataSet

**Imports** Cursos.db\_cursoDataSetTableAdapters

# ADO.NET

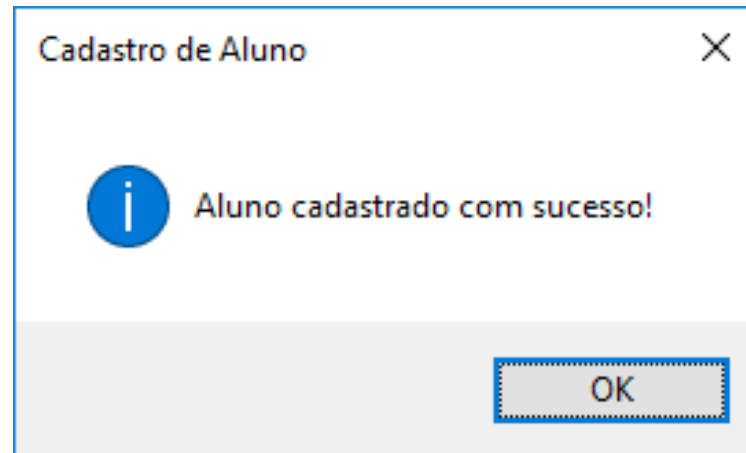
## ■ Exemplo – Cadastro de Alunos

### ■ 3º PASSO: Crie os métodos de acesso a dados

```
private void btnIncluir_Click(object sender, EventArgs e) {  
    try {  
        //Valida se o nome do cliente foi preenchido  
        if ((string.IsNullOrEmpty(txtNome.Text)) |  
            (string.IsNullOrEmpty(txtRA.Text)))  
            MessageBox.Show("Por favor, informe o nome e o RA do aluno!");  
        else {  
            //Cria o adapter  
            tb_alunoTableAdapter adapter = new tb_alunoTableAdapter();  
            //Insere no banco de dados  
            adapter.Insert(Convert.ToInt32(txtRA.Text), txtNome.Text);  
            MessageBox.Show("Aluno cadastrado com sucesso!", "Cadastro de  
Aluno", MessageBoxButtons.OK, MessageBoxIcon.Information);  
        }  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK,  
            MessageBoxIcon.Error);  
    }  
}
```

# ADO.NET

- Exemplo – Cadastro de Alunos
  - ❑ 4º PASSO: Teste sua tela!!!





# ADO.NET

- Observação: Você pode criar métodos para abrir, fechar a conexão e outros, conforme exemplo

- 1. Método para abrir a Conexão com o Banco

```
private System.Data.SqlClient.SqlConnection conn = null;
```

```
public System.Data.SqlClient.SqlConnection  
abreConexao()  
{  
    if ((conn == null))  
        conn = new  
System.Data.SqlClient.SqlConnection(global::Aulas_Programa  
cao_CSharp.Properties.Settings.Default.db_cursoConnectionS  
tring);  
    if (!(conn.State == ConnectionState.Open))  
        conn.Open();  
    return conn;  
}
```

# ADO.NET

- Observação: Você pode criar métodos para abrir, fechar a conexão e outros, conforme exemplo
  - 2. Método para fechar a Conexão com o Banco

```
public void fechaConexao()  
{  
    if ((conn != null)) {  
        conn.Close();  
        conn = null;  
    }  
}
```

# ADO.NET

- 3. Método para executar um comando SQL (SELECT) e popular um DataSet

```
public DataSet executaComando(string SQL) {  
    System.Data.SqlClient.SqlDataAdapter adapter = new  
    System.Data.SqlClient.SqlDataAdapter();  
    DataSet ds = new DataSet();  
    System.Data.SqlClient.SqlCommand comando = new  
    System.Data.SqlClient.SqlCommand();  
    try {  
        comando.Connection = abreConexao();  
        comando.CommandText = SQL;  
        adapter.SelectCommand = comando;  
        adapter.Fill(ds);  
        return ds;  
    }  
    catch (Exception ex) {  
        throw ex;  
    }  
    finally {  
        adapter.Dispose();  
        fechaConexao();  
    }  
}
```

# SQL - Structured Query Language

- Exemplos de comandos SQL que podem ser usados durante o desenvolvimento:

```
CREATE TABLE [dbo].[tb_aluno](  
    [cod_ra] [int] NOT NULL,  
    [nom_aluno] [varchar](250) NOT NULL  
) ON [PRIMARY]
```

```
select * from tb_aluno where nom_aluno like '%Eduardo%'
```

```
update tb_aluno set cod_ra = 25215 where cod_ra = 84922
```

```
delete tb_aluno where cod_ra = 84922
```

```
drop table tb_aluno
```

---

# SQLServer

- ❑ Exercício Extra Classe:
  - ❑ Instalar:
    - ❑ SQLServer DataBase
    - ❑ Management Studio Express

# SQLServer

- ❑ Notas de Instalação:
  - ❑ 32/64bits
  - ❑ SO em inglês/portugues/SP/Win7/Win8
  - ❑ Espaço em disco
  - ❑ Já existe outra versão instalada
  - ❑ Firewall e outros pré-requisitos
  - ❑ Quantidade de features desejadas
  - ❑ Windows Update
- ❑ Boa sorte!

---

# Microsoft .NET

❑ Obrigado pela atenção.

Eduardo Chagas de Oliveira  
eduardo.chagas@uniube.br