

Curso de C#

Aula 5

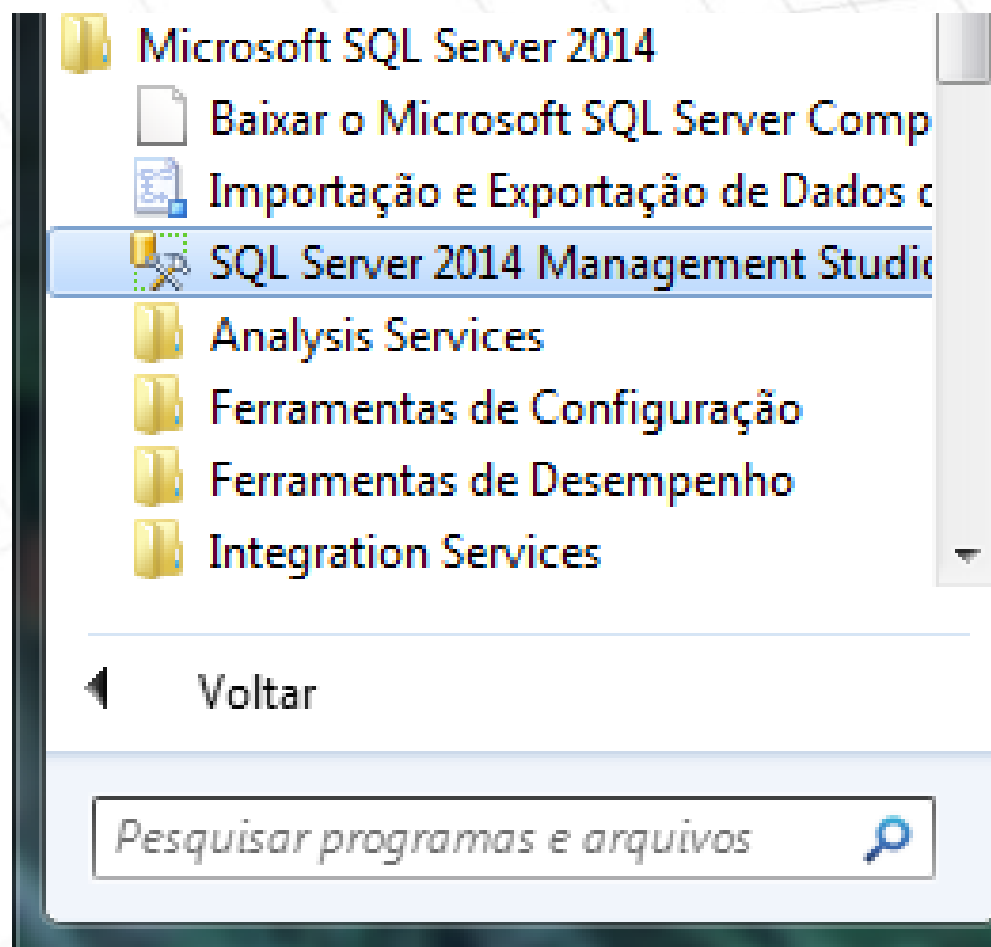
Professores

Célio Alencar de Assis
Fábio Duarte Machado

O que será abordado Hoje

- Aplicação de Gerenciamento do SQLServer
- ADO.NET
- Substituir o Repository de Listas por um de SQLServer

Acessando SQL Server 2014 Management Studio



Efetuando Login

Conectar ao Servidor

Microsoft SQL Server 2014

Tipo de servidor: Mecanismo de Banco de Dados

Nome do servidor: BELGICA\SQLEXPRESS

Autenticação: Autenticação do Windows

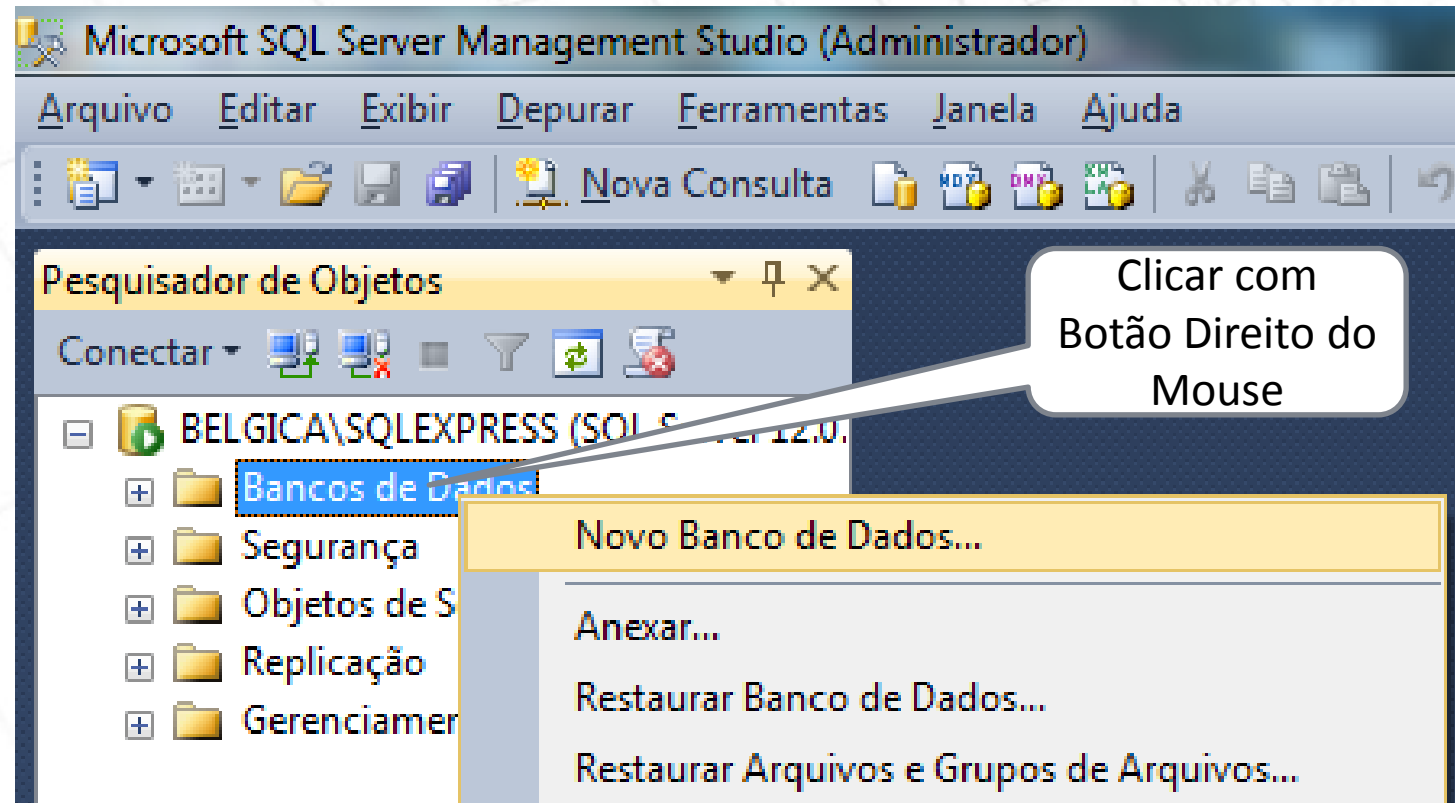
Nome de usuário: BELGICA\Célio

Senha:

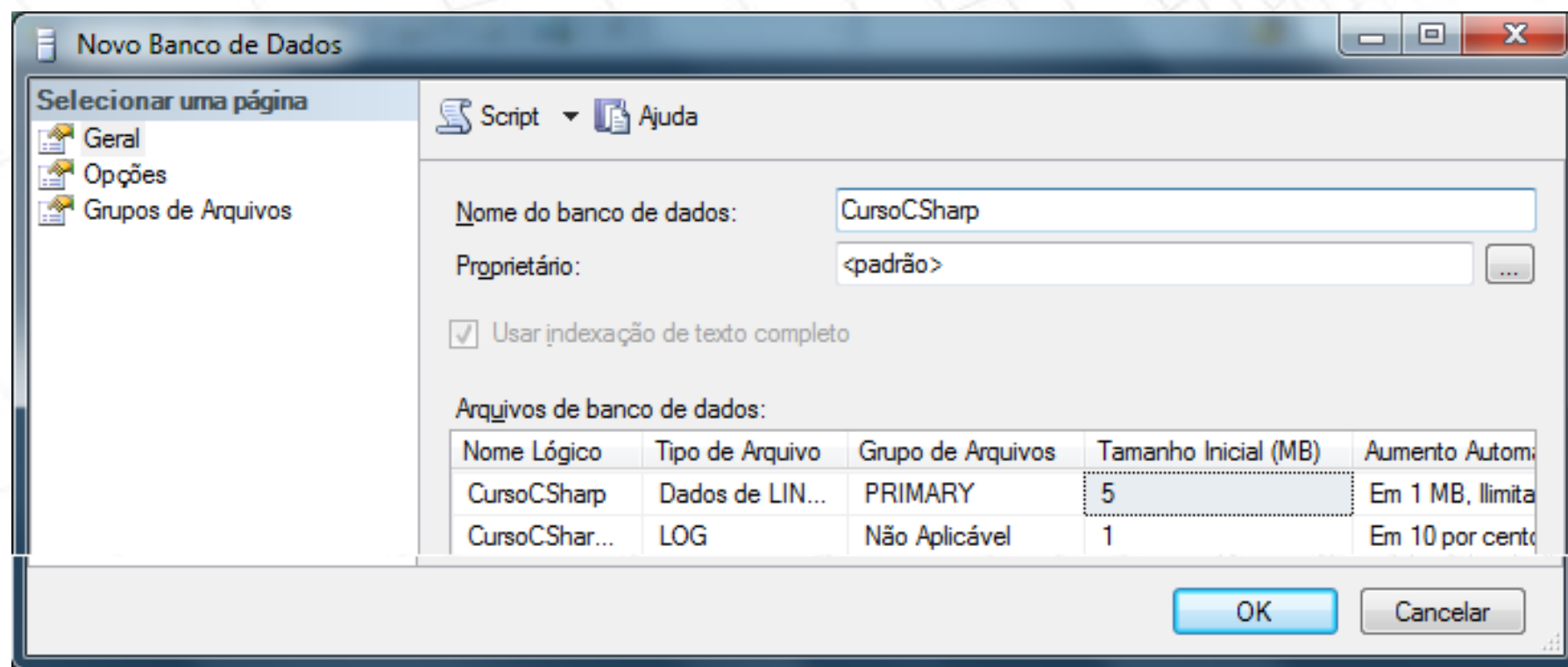
☐ Lembrar senha

Conectar Cancelar Ajuda Opções >>

Criar Banco de Dados



Definir Nome do Banco de Dados



Novo Banco de Dados

Selecionar uma página

- Geral
- Opções
- Grupos de Arquivos

Script ▾ Ajuda

Nome do banco de dados:

Proprietário:

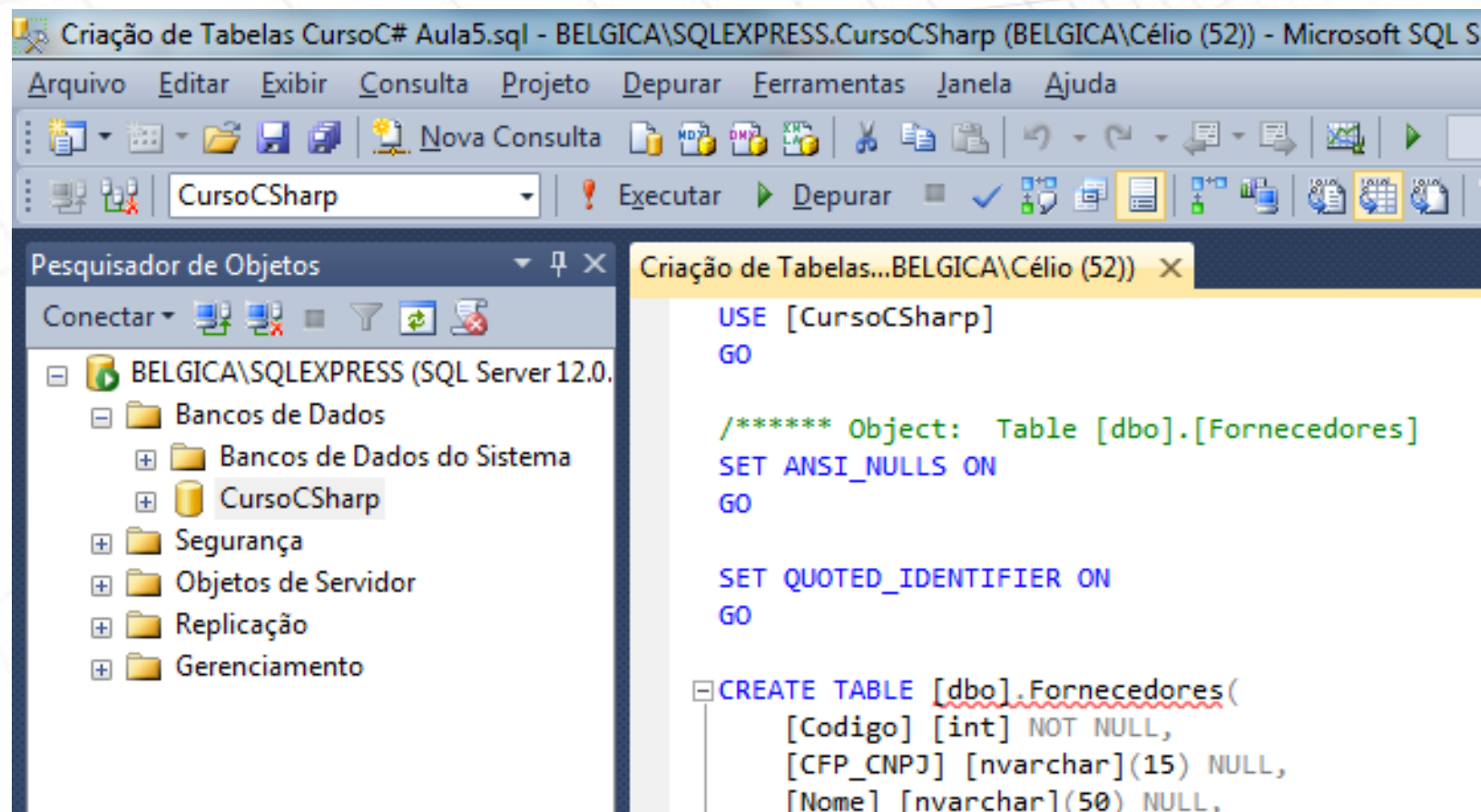
☒ Usar indexação de texto completo

Arquivos de banco de dados:

Nome Lógico	Tipo de Arquivo	Grupo de Arquivos	Tamanho Inicial (MB)	Aumento Automático
CursoCSharp	Dados de LIN...	PRIMARY	5	Em 1 MB, Ilimitado
CursoCSharp...	LOG	Não Aplicável	1	Em 10 por cento

OK Cancelar

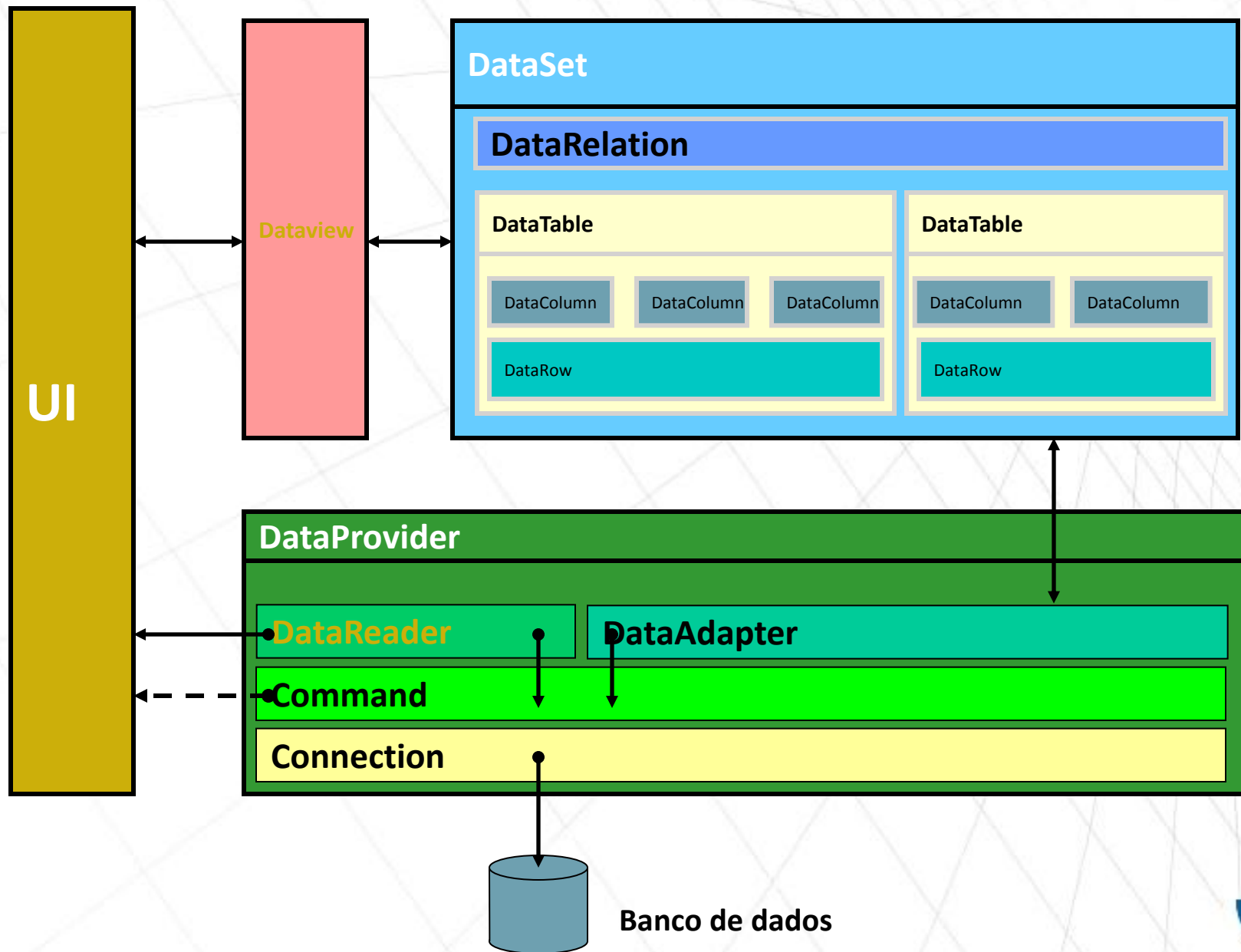
Executar Script para Criação das Tabelas



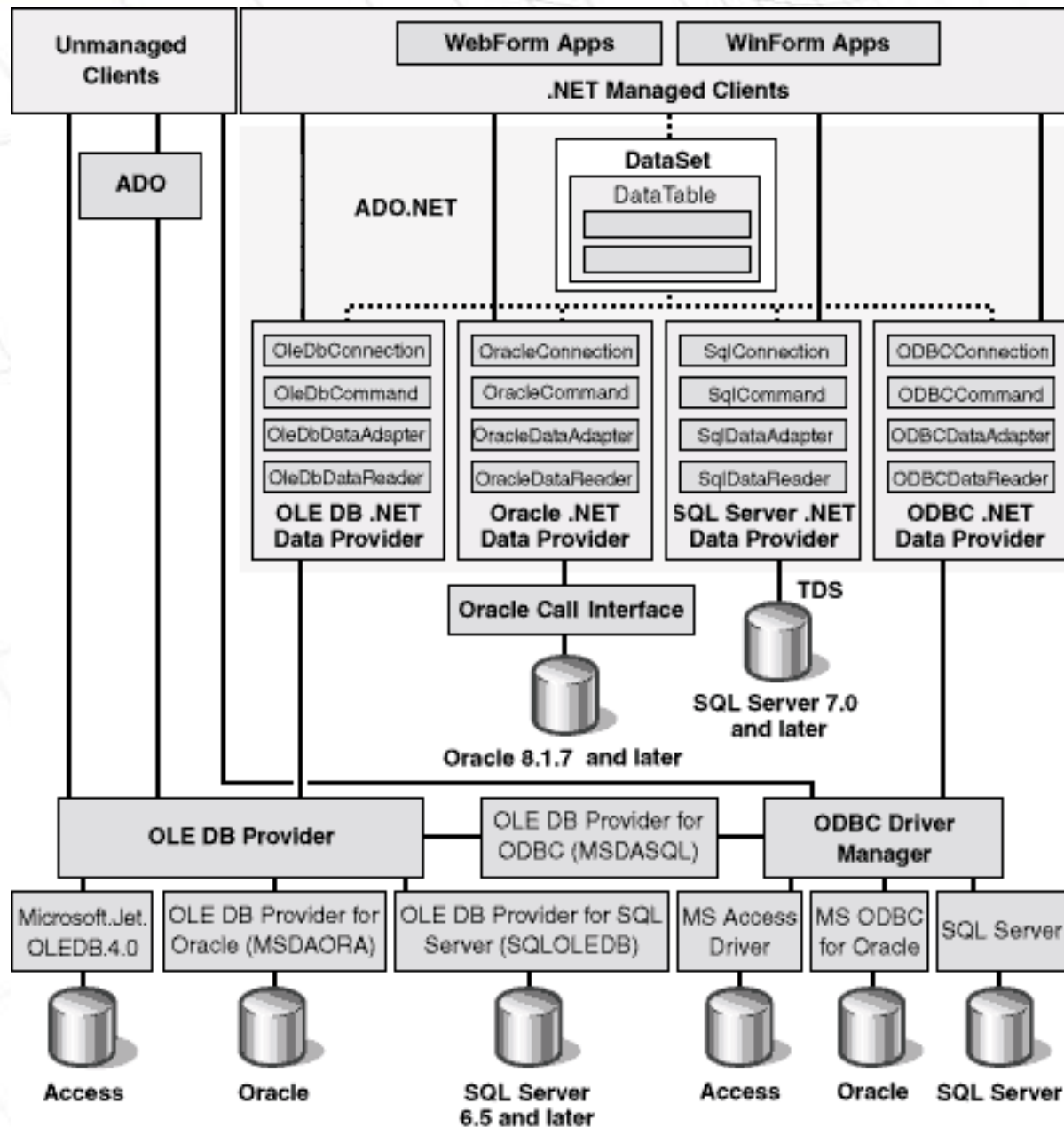
Criação de Tabelas CursoC# Aula5.sql

ADO .net

A plataforma .net oferece uma gama de ferramentas para acesso a dados das mais diversas maneiras.

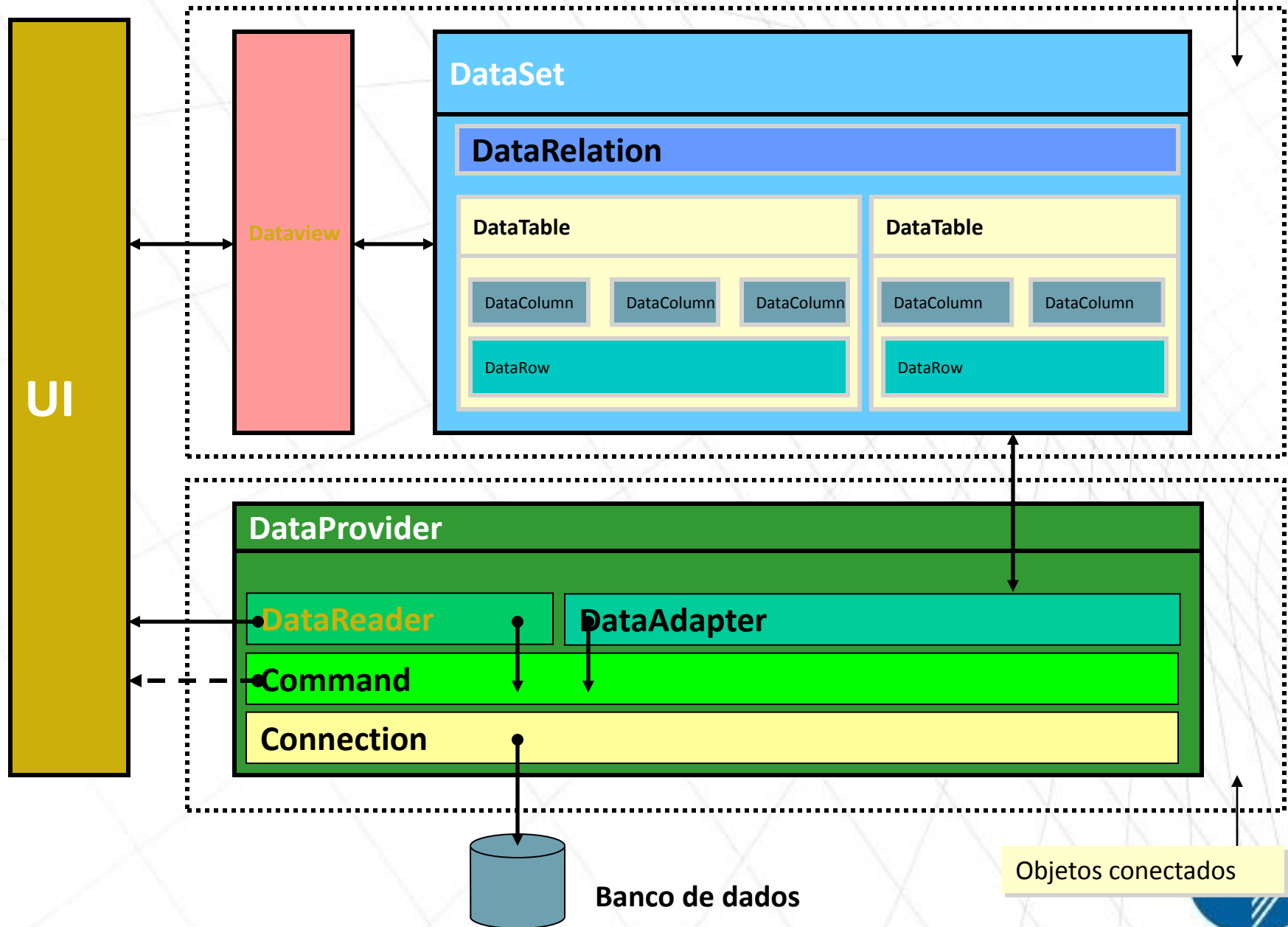


C# Acesso a dados – Modelo ADO .net



É possível trabalhar diretamente conectado ao banco de dados e também de forma desconectada.

Objetos desconectados



Trabalhando de forma conectada



Criando um Banco de Dados

```
protected void btnCriarBanco_Click(object sender, EventArgs e)
```

```
{
```

```
    try
```

```
    {
```

```
        string strConn;
```

```
        //Utilizando Usuário do Banco
```

```
        //strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=master;User ID=sa;Password=123123";
```

```
        //Utilizando Usuário do Windows
```

```
        strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=master;Integrated Security=True";
```

```
        SqlConnection connection = new SqlConnection(strConn);
```

```
        string sql = "CREATE DATABASE TesteAula05";
```

```
        connection.Open();
```

```
        SqlCommand sqlCommand = new SqlCommand(sql, connection);
```

```
        sqlCommand.ExecuteNonQuery();
```

```
    }
```

```
    catch (System.Data.SqlClient.SqlException ex)
```

```
    {
```

```
        System.Diagnostics.Debug.WriteLine(ex.Message);
```

```
    }
```

```
}
```

Abre a conexão

String de conexão

Cria a conexão do SQLServer

Executa a SQL

Cria o Command

Criando uma tabela

```
protected void btnCriarTabela_Click(object sender, EventArgs e)
{
    try
    {
        string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
        SqlConnection connection = new SqlConnection(strConn);
        string sql = "CREATE TABLE Pessoas (" +
            "id          int IDENTITY," +
            "nome        varchar(40) NOT NULL," +
            "endereco    char(80)," +
            "cpf          numeric(11) UNIQUE, " +
            "PRIMARY KEY(id))";
        connection.Open();
        SqlCommand sqlCommand = new SqlCommand(sql, connection);
        sqlCommand.ExecuteNonQuery();
        connection.Close();
    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        System.Diagnostics.Debug.WriteLine(ex.Message);
    }
}
```

String de conexão

Cria a conexão do SQLServer

Abre a conexão

Cria o Command

Executa a SQL

Inserindo dados

```
protected void btnInserirDados_Click(object sender, EventArgs e)
{
    try
    {
        string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
        SqlConnection connection = new SqlConnection(strConn);
        string sql = "INSERT INTO Pessoas (Nome, Endereco, Cpf) " +
            "VALUES ('Nero', 'Av. da Labareda, ROMA',12345678901);" +
            "INSERT INTO Pessoas (Nome, Endereco, Cpf) " +
            "VALUES ('Gensis Khan', 'Av. dos Unos, Persia',12345678902);";
        connection.Open();
        SqlCommand sqlCommand = new SqlCommand(sql, connection);
        sqlCommand.ExecuteNonQuery();
        connection.Close();
    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        System.Diagnostics.Debug.WriteLine(ex.Message);
    }
}
```

Lendo dados

```
protected void btnLerDados_Click(object sender, EventArgs e)
{
    try
    {
        string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
        SqlConnection connection = new SqlConnection(strConn);
        string sql = "SELECT * FROM Pessoas";
        connection.Open();
        SqlCommand sqlCommand = new SqlCommand(sql, connection);
        SqlDataReader sqlDataReader = sqlCommand.ExecuteReader();
        StringBuilder buff = new StringBuilder();
        while (sqlDataReader.Read())
        {
            buff.AppendFormat("Nome: {0} - Endereço: {1} \n",
                sqlDataReader["Nome"].ToString().Trim(),
                sqlDataReader["Endereco"].ToString().Trim());
        }
        connection.Close();
        System.Diagnostics.Debug.WriteLine(buff);
    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        System.Diagnostics.Debug.WriteLine(ex.Message);
    }
}
```

Cria o DataReader

Faz a leitura dos dados

Saída

Mostrar saída de: Depurar

Nome: Nero - Endereço: Av. da Labareda, ROMA

Nome: Gensis Khan - Endereço: Av. dos Unos, Persia

Trabalhando de forma desconectada



Lendo dados

```
protected void btnLerDadosDesconectado_Click(object sender, EventArgs e)
{
    try
    {
        string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
        SqlConnection connection = new SqlConnection(strConn);
        connection.Open();
        SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select * FROM Pessoas", connection);
        System.Data.DataSet dataSet = new System.Data.DataSet();
        sqlDataAdapter.Fill(dataSet);

        connection.Close();

        StringBuilder buff = new StringBuilder();
        foreach (System.Data.DataRow dataRow in dataSet.Tables[0].Rows) {
            buff.AppendFormat("Nome: {0} - Endereço: {1} \n",
                dataRow["Nome"].ToString().Trim(),
                dataRow["Endereco"].ToString().Trim());
        }
        System.Diagnostics.Debug.WriteLine(buff);
    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        System.Diagnostics.Debug.WriteLine(ex.Message);
    }
}
```

Preenche o DataSet

Cria o DataSet

Cria o DataAdapter

Fecha a conexão

Faz a leitura dos dados de maneira desconectada

Saída

Mostrar saída de: Depurar

Nome: Nero - Endereço: Av. da Labareda, ROMA

Nome: Gensis Khan - Endereço: Av. dos Unos, Persia

C# Acesso a dados – Desconectado – Alterando dados

```
string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
SqlConnection connection = new SqlConnection(strConn);
connection.Open();
SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("select * FROM Pessoas", connection);
System.Data.DataSet dataSet = new System.Data.DataSet();

SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(sqlDataAdapter);

sqlDataAdapter.Fill(dataSet, "Pessoas");

connection.Close();

StringBuilder buff = new StringBuilder();
foreach (System.Data.DataRow dr in dataSet.Tables[0].Rows)
{
    buff.AppendFormat("Nome: {0} - Endereço: {1} \n",
        dr["Nome"].ToString().Trim(),
        dr["Endereco"].ToString().Trim());
}
connection.Close();
System.Diagnostics.Debug.WriteLine(buff);

System.Data.DataRow alteraDR = dataSet.Tables[0].Rows[0];
alteraDR["Nome"] = "Cleopatra";

//Abre novamente a conexão com o banco
connection.Open();
sqlDataAdapter.Update(dataSet, "Pessoas");
connection.Close();

buff.Clear();
buff.AppendLine("=== Registros Alterados ===");
foreach (System.Data.DataRow dr in dataSet.Tables[0].Rows)
{
    buff.AppendFormat("Nome: {0} - Endereço: {1} \n",
        dr["Nome"].ToString().Trim(),
        dr["Endereco"].ToString().Trim());
}
System.Diagnostics.Debug.WriteLine(buff);
```

Fecha a
conexão

Envia por parâmetro o
DataAdapter para dentro
de um CommandBuilder.
Essa operação é necessária
para o update funcionar
corretamente.

Seleciona a primeira
linha da primeira tabela
do DataSet

Altera o campo nome

Executa a atualização

Reabre a
conexão

Usando o controle **GridView** com o **DataView**

O GridView é um dos controles mais versáteis do ASP.NET, ele pode facilmente mostrar um conjunto de dados de um DataSet.

GridView

```
protected void btnPreencheGridView_Click(object sender, EventArgs e)
{
    try
    {
        string strConn = "Data Source=localhost\\SQLEXPRESS;Initial Catalog=TesteAula05;Integrated Security=True";
        SqlConnection connection = new SqlConnection(strConn);
        connection.Open();
        SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select * FROM Pessoas", connection);
        System.Data.DataSet dataSet = new System.Data.DataSet();
        sqlDataAdapter.Fill(dataSet);

        connection.Close();

        System.Data.DataView dataView = new DataView(dataSet.Tables[0]);
        gvTestes.DataSource = dataView;
        gvTestes.DataBind();

    }
    catch (System.Data.SqlClient.SqlException ex)
    {
        System.Diagnostics.Debug.WriteLine(ex.Message);
    }
}
```

Cria o DataView com primeira tabela do DataSet

Informa ao GridView que seu source será o dataView

Efetiva a ligação com o source

id	nome	endereço	cpf
1	Cleopatra	Av. da Labareda, ROMA	12345678901
2	Gensis Khan	Av. dos Unos, Persia	12345678902

Criar uma Classe de DAO seguindo o padrão factory que já foi discutido para que encapsule os seguintes comandos:

- Criar DataTable
- Criar DataSet
- ExecuteNoQuery
- ExecuteScalar.

1

2

Criar um Repository para usar banco de dados