



Colégio
Pedro II

PROGRAMAÇÃO O.O. (C#)



Desenvolvendo aplicação CRUD
Professor: João Luiz Lagôas



Roteiro da Aula



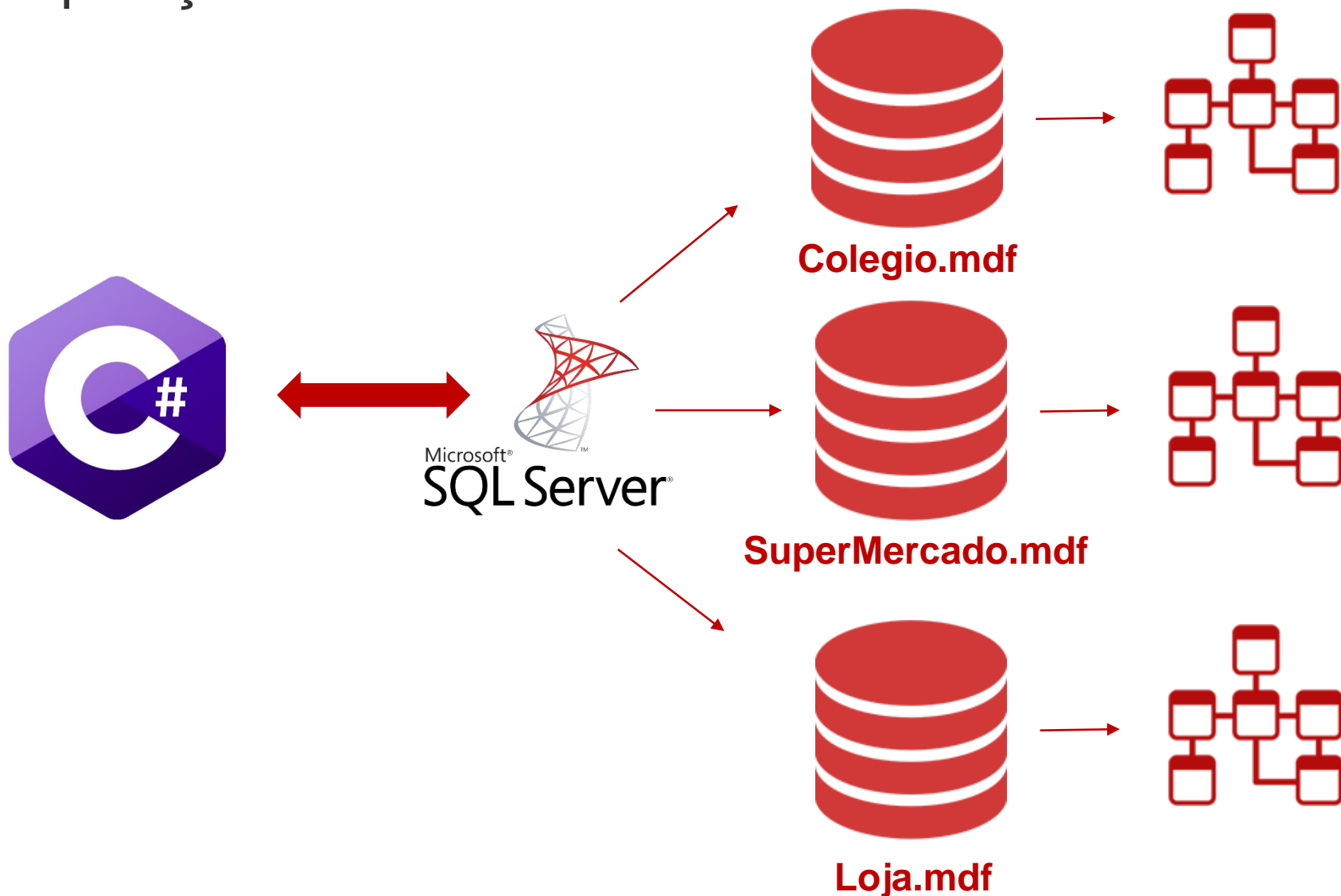
Parte 01:

- Lembrar a estrutura de uma aplicação de banco de dados;
- Lembrar a criação de um projeto de banco de dados;
- Fazer e corrigir a Lista 1 – Exercícios 02-03;

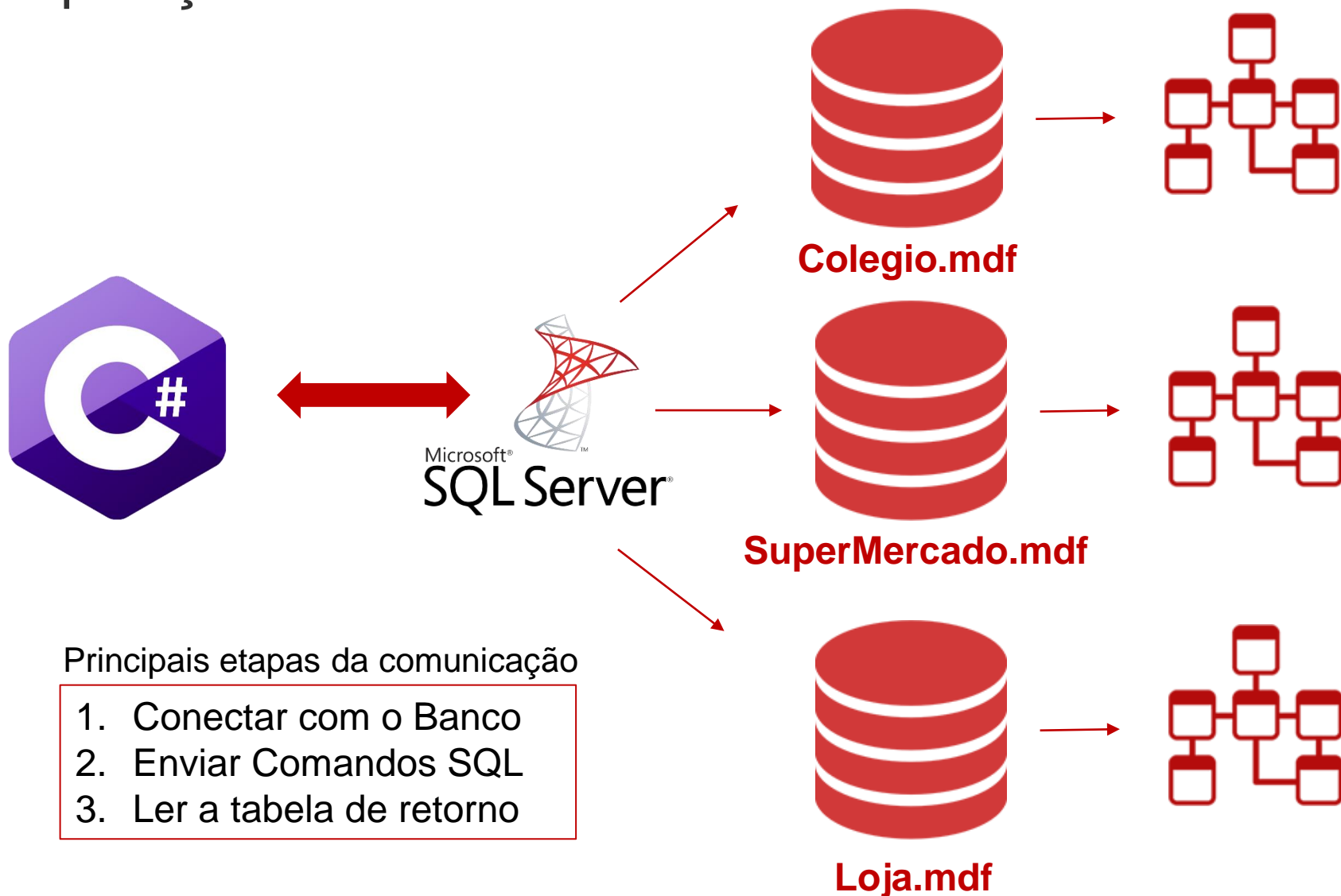
Parte 02:

- Aprender o necessário para se criar uma aplicação CRUD.
- Fazer a Lista 2 – Exercícios 04-06;

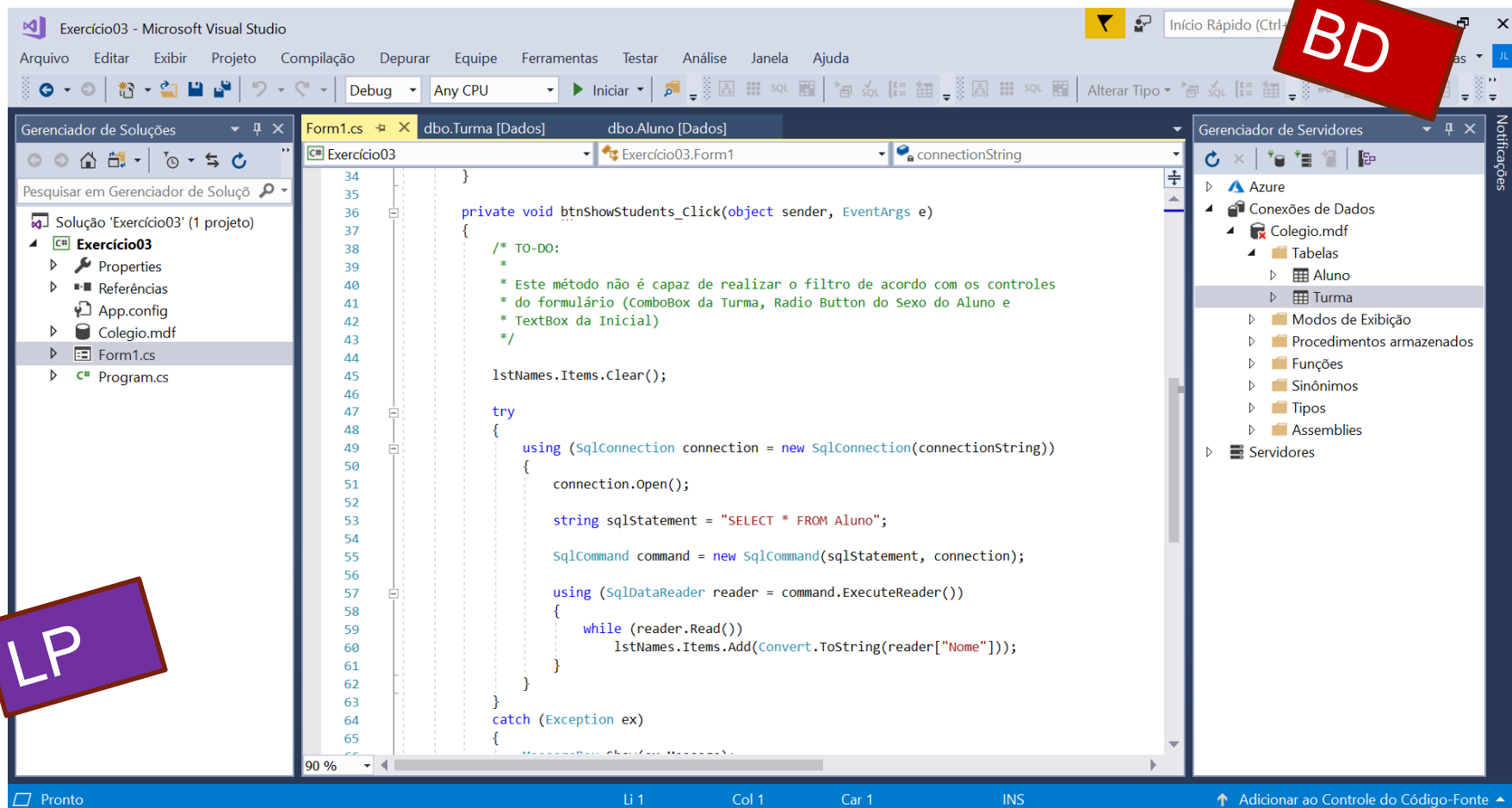
Estrutura de uma aplicação de banco de dados



Estrutura de uma aplicação de banco de dados



Estrutura de uma aplicação de banco de dados



Exemplo



Dado um formulário como mostrado abaixo e um banco de dados denominado Colégio com apenas uma tabela Aluno, o processo de conexão, envio de comando e leitura seria dado através do código a seguir:

Form1

Aluno

	Matricula	Nome	Sexo
▶ 1	1	Pedro	M
	2	Mari	F
	3	Antonio	F
	4	Sergio	M
	5	Eduardo	M
	6	João	M
	7	Leonardo	M
	8	Felipe	M
	9	Pamela	F
	10	Mariah	F
	11	Regina	F
	12	Leticia	F
	13	Leandro	M

SqlConnection: Realizando a conexão

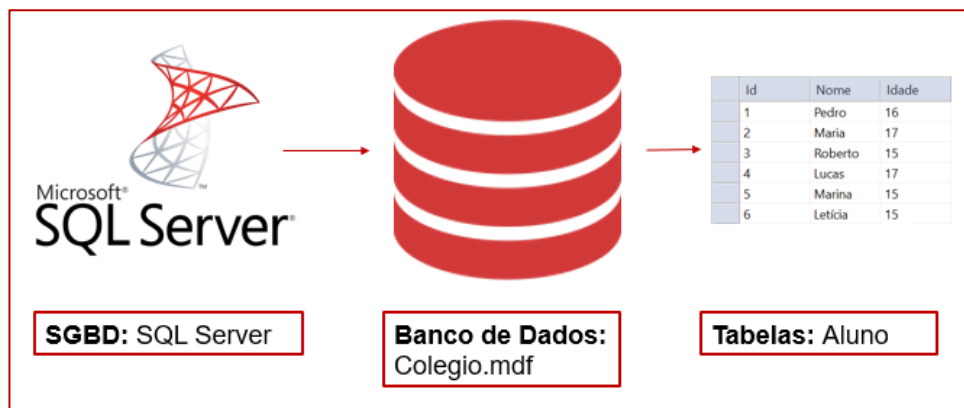
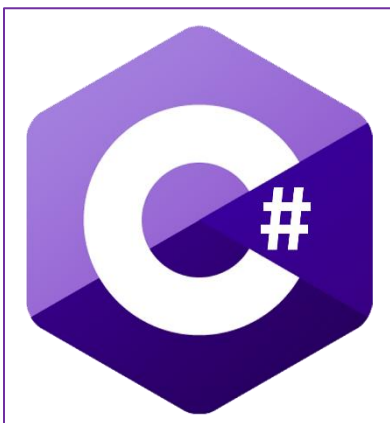


```
string stringDeConexao = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\João
Lagôas\source\repos\ExemploPratico\ExemploPratico\Colegio.mdf';Integrated
Security=True";

SqlConnection conexao = new SqlConnection(stringDeConexao);

conexao.Open();
```

Exemplo



SqlConnection: Realizando a conexão

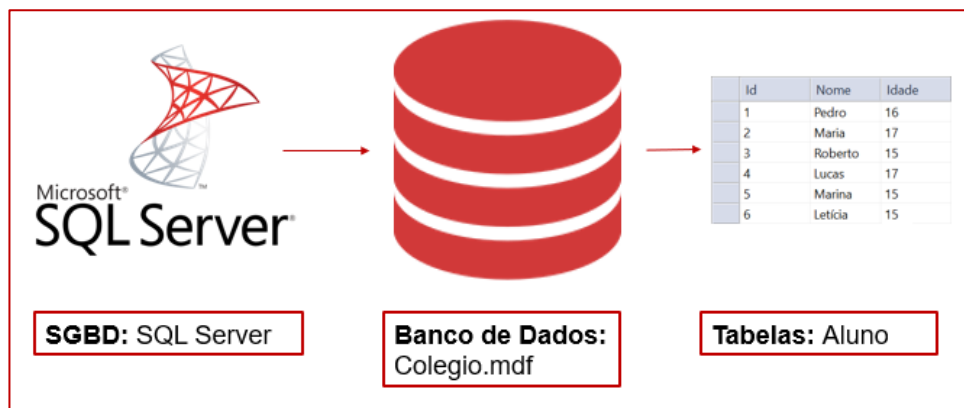
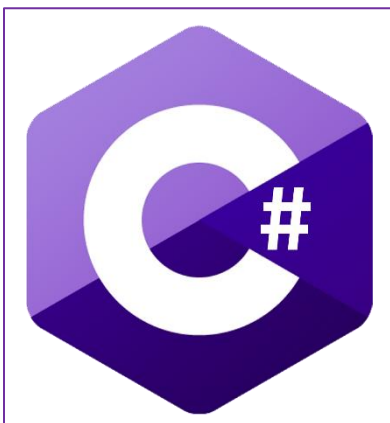


Exemplo

```
string stringDeConexao = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\João
Lagôas\source\repos\ExemploPratico\ExemploPratico\Colegio.mdf';Integrated
Security=True";

SqlConnection conexao = new SqlConnection(stringDeConexao);

conexao.Open();
```



SqlConnection: Realizando a conexão



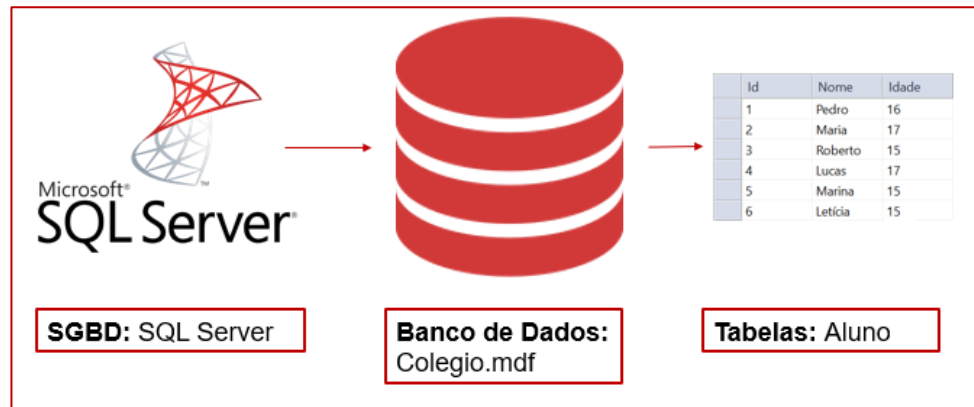
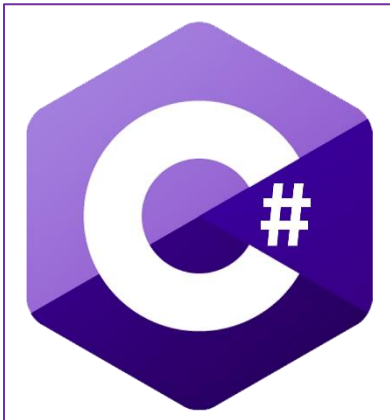
Exemplo

```
string stringDeConexao = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\João
Lagôas\source\repos\ExemploPratico\ExemploPratico\Colegio.mdf';Integrated
Security=True";
```



SqlConnection conexao = new SqlConnection(stringDeConexao);

conexao.Open();



SqlConnection: Realizando a conexão

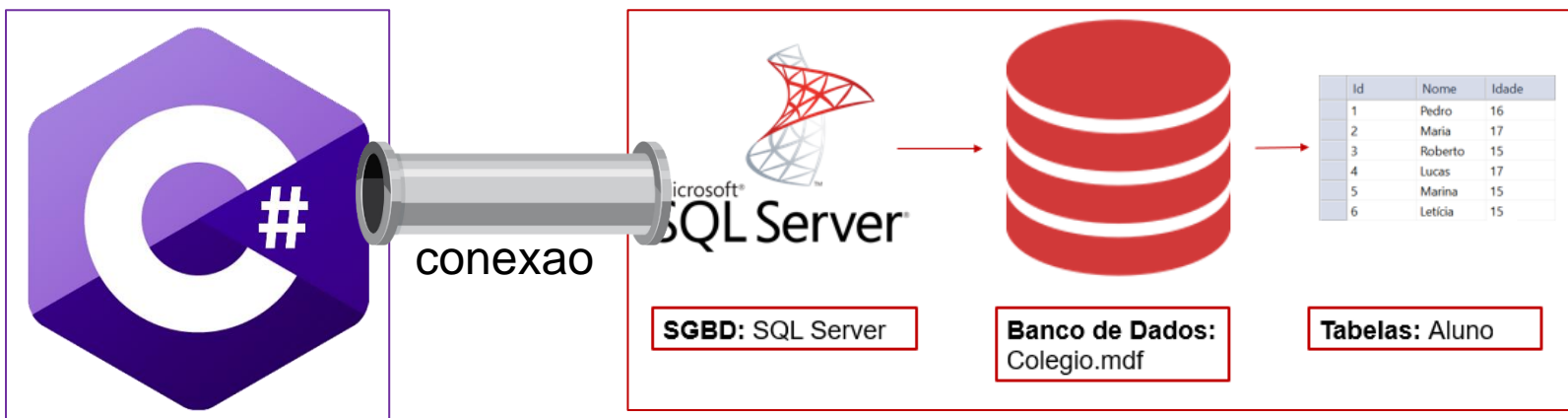


Exemplo

```
string stringDeConexao = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\João
Lagôas\source\repos\ExemploPratico\ExemploPratico\Colegio.mdf';Integrated
Security=True";

SqlConnection conexao = new SqlConnection(stringDeConexao);

conexao.Open();
```

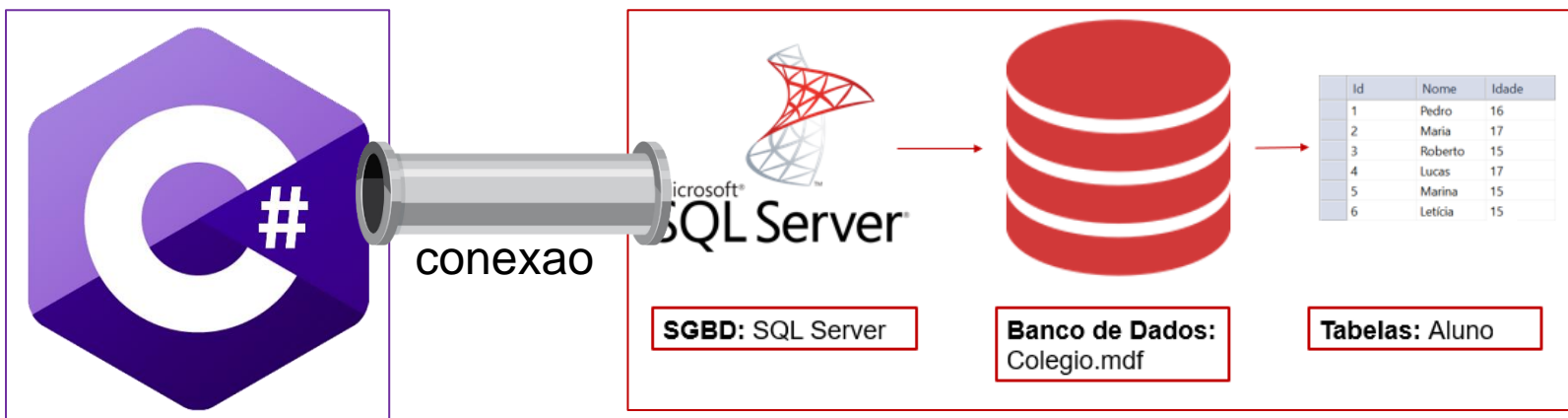


SqlCommand: Enviando um comando



```
string consulta = "SELECT * FROM Aluno";  
  
SqlCommand comando = new SqlCommand(consulta, conexao);  
  
SqlDataReader leitor = comando.ExecuteReader();
```

Exemplo

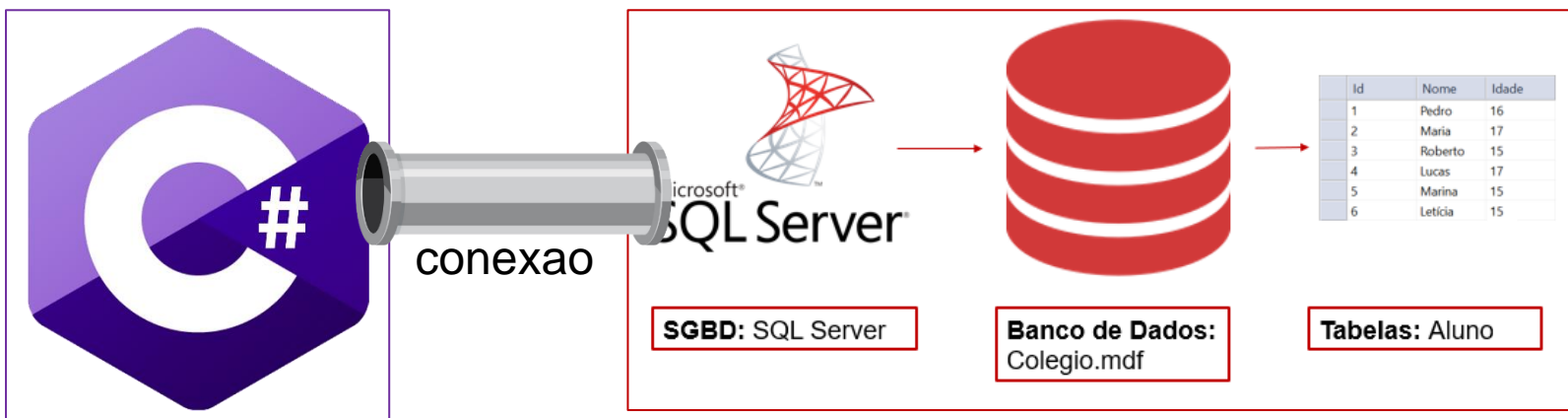


SqlCommand: Enviando um comando



```
string consulta = "SELECT * FROM Aluno";  
  
SqlCommand comando = new SqlCommand(consulta, conexao);  
  
SqlDataReader leitor = comando.ExecuteReader();
```

Exemplo



SqlCommand: Enviando um comando



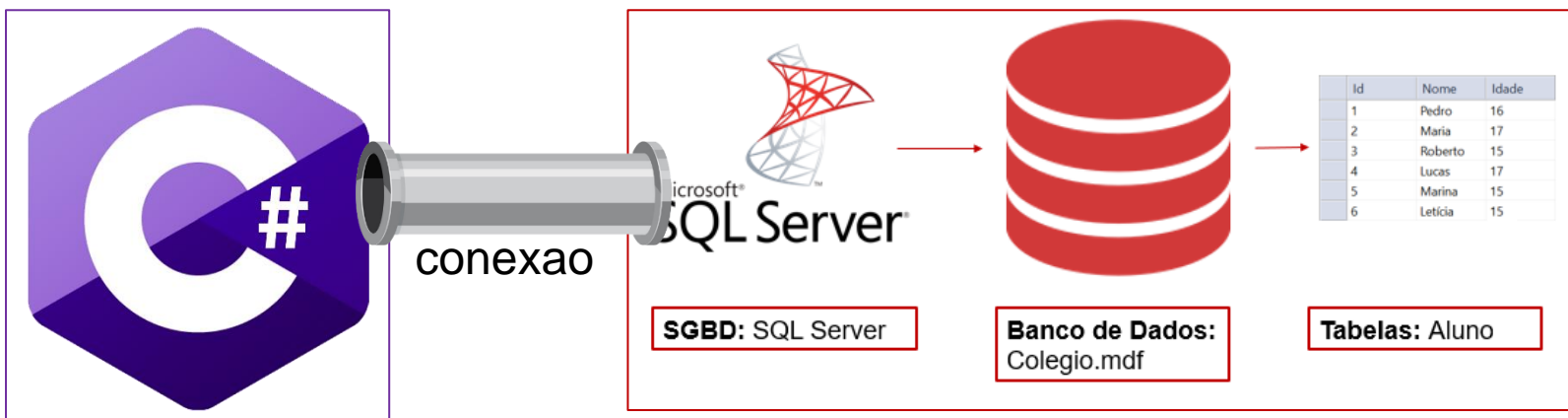
```
string consulta = "SELECT * FROM Aluno";
```



```
SqlCommand comando = new SqlCommand(consulta, conexao);
```

```
SqlDataReader leitor = comando.ExecuteReader();
```

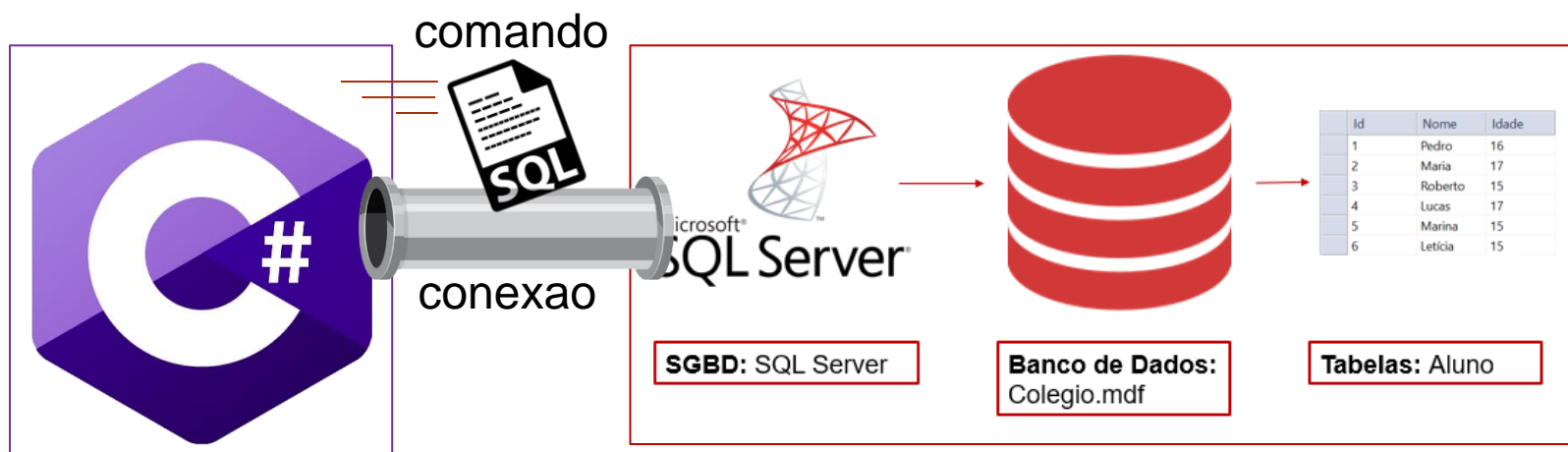
Exemplo



SqlCommand: Enviando um comando

```
string consulta = "SELECT * FROM Aluno";  
  
SqlCommand comando = new SqlCommand(consulta, conexao);  
→ SqlDataReader leitor = comando.ExecuteReader();
```

Exemplo

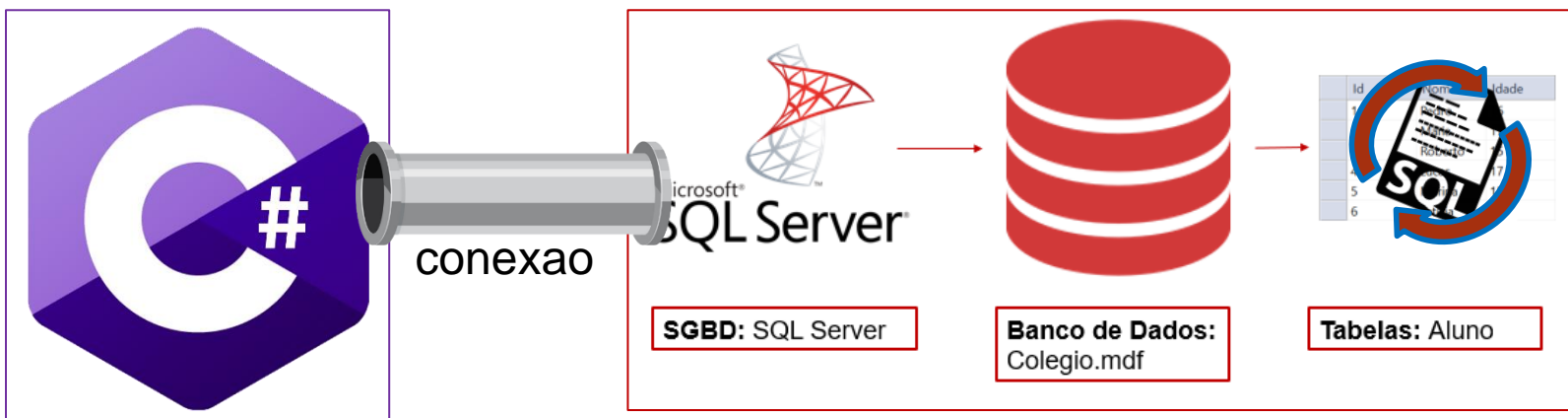


SqlCommand: Enviando um comando



```
string consulta = "SELECT * FROM Aluno";  
  
SqlCommand comando = new SqlCommand(consulta, conexao);  
SqlDataReader leitor = comando.ExecuteReader();
```

Exemplo



SqlCommand: Enviando um comando

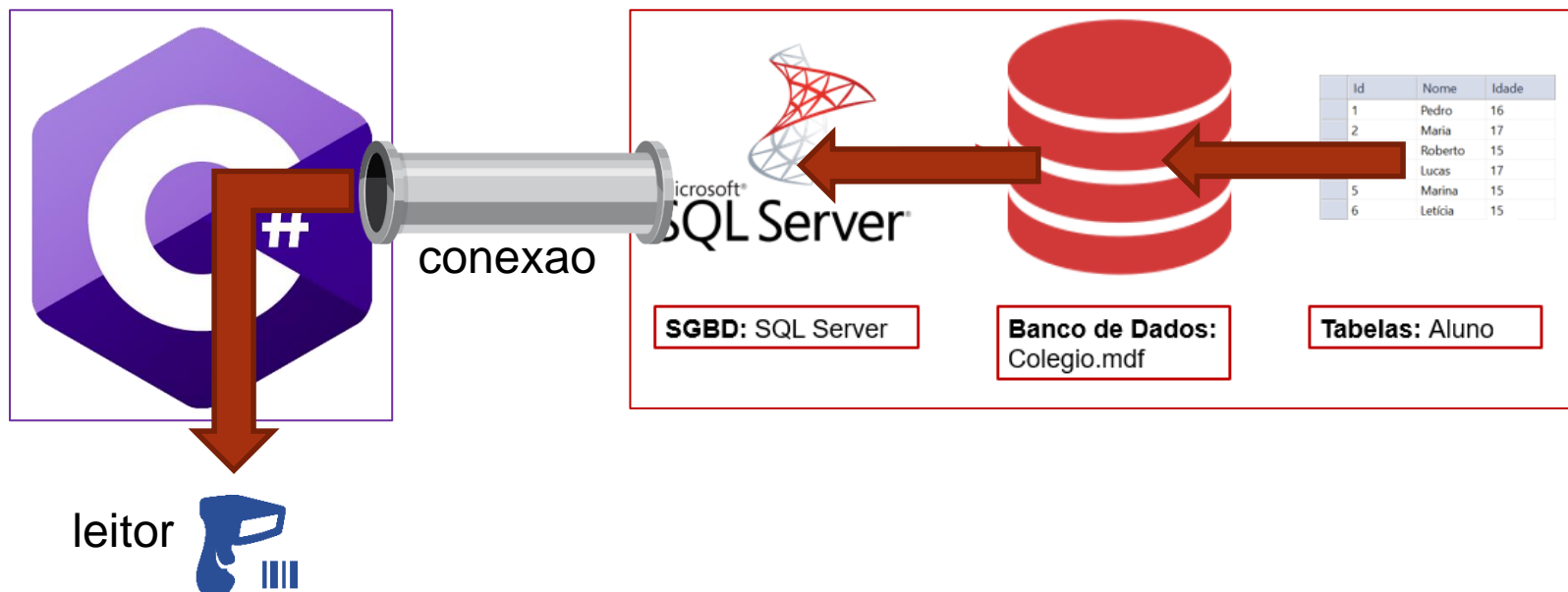


```
string consulta = "SELECT * FROM Aluno";

SqlCommand comando = new SqlCommand(consulta, conexao);

SqlCommandReader leitor = comando.ExecuteReader();
```

Exemplo



SqlDataReader: Lendo a resposta do Banco de Dados



Colégio
Pedro II



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor




	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

SqlDataReader: Lendo a resposta do Banco de Dados



Colégio
Pedro II



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor



leitor["Id"]

leitor["Nome"]

leitor["Idade"]

	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

SqlDataReader: Lendo a resposta do Banco de Dados



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor




	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

SqlDataReader: Lendo a resposta do Banco de Dados



Colégio
Pedro II



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor["Id"] leitor["Nome"] leitor["Idade"]

leitor



	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

SqlDataReader: Lendo a resposta do Banco de Dados



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor



	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

SqlDataReader: Lendo a resposta do Banco de Dados



Colégio
Pedro II



```
while(leitor.Read())  
{  
    listBox1.Items.Add(leitor["Nome"].ToString());  
}  
leitor.Close();
```

leitor["Id"] leitor["Nome"] leitor["Idade"]

leitor



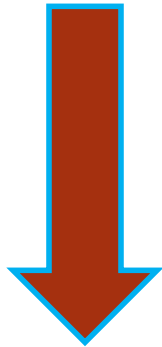
	Id	Nome	Idade
	1	Pedro	16
	2	Maria	17
	3	Roberto	15
	4	Lucas	17
	5	Marina	15
	6	Letícia	15

Lista 1 Banco de Dados

Exercício 02 – 03



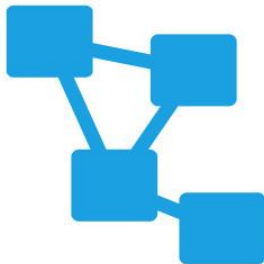
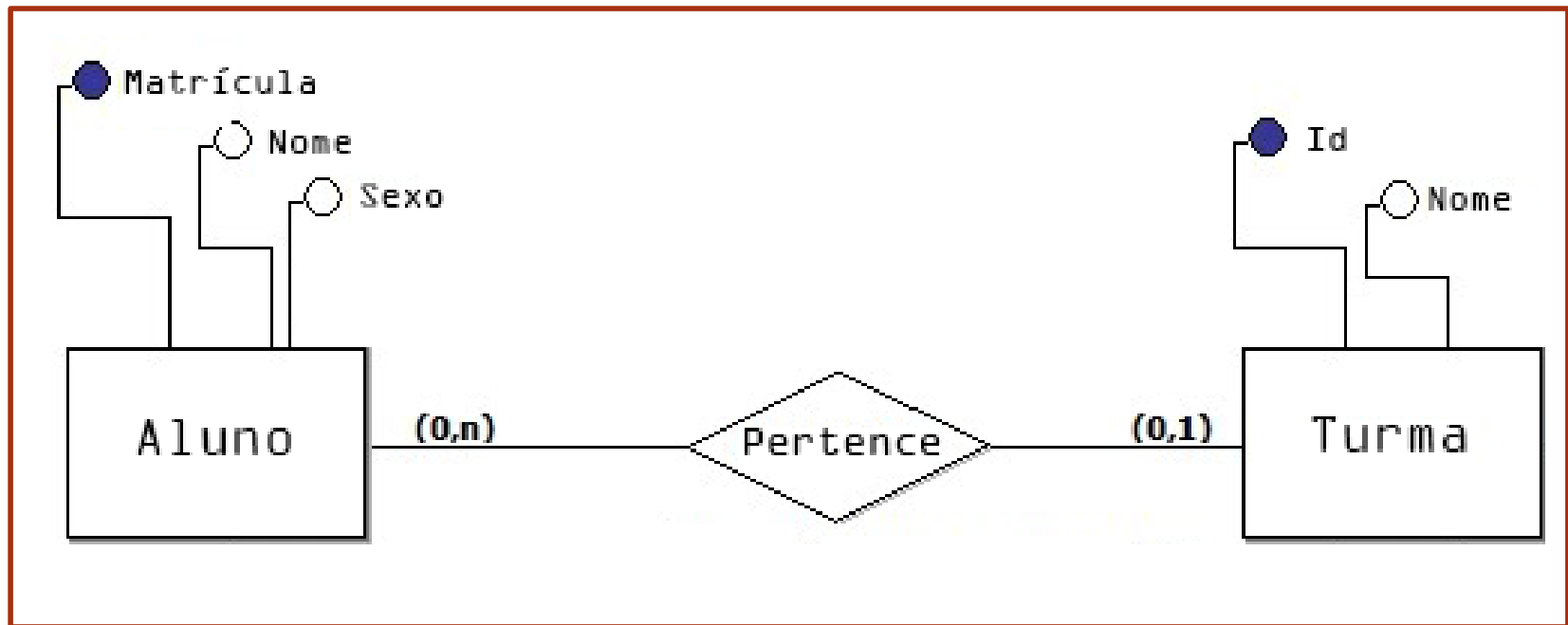
1. Criação do **Projeto Conceitual**;
2. Criação do **Projeto Lógico**;
3. Criação do **Projeto Físico**;



4. Criação da **Aplicação C#**;

Exemplo

Projeto **Conceitual** de Banco de Dados



Geralmente representado através de um **Diagrama Entidade Relacionamento (DER)**.

Exemplo

Projeto **Lógico** de Banco de Dados



Aluno(Matricula, Nome, Sexo, Turma)

Turma REF Turma(Id)

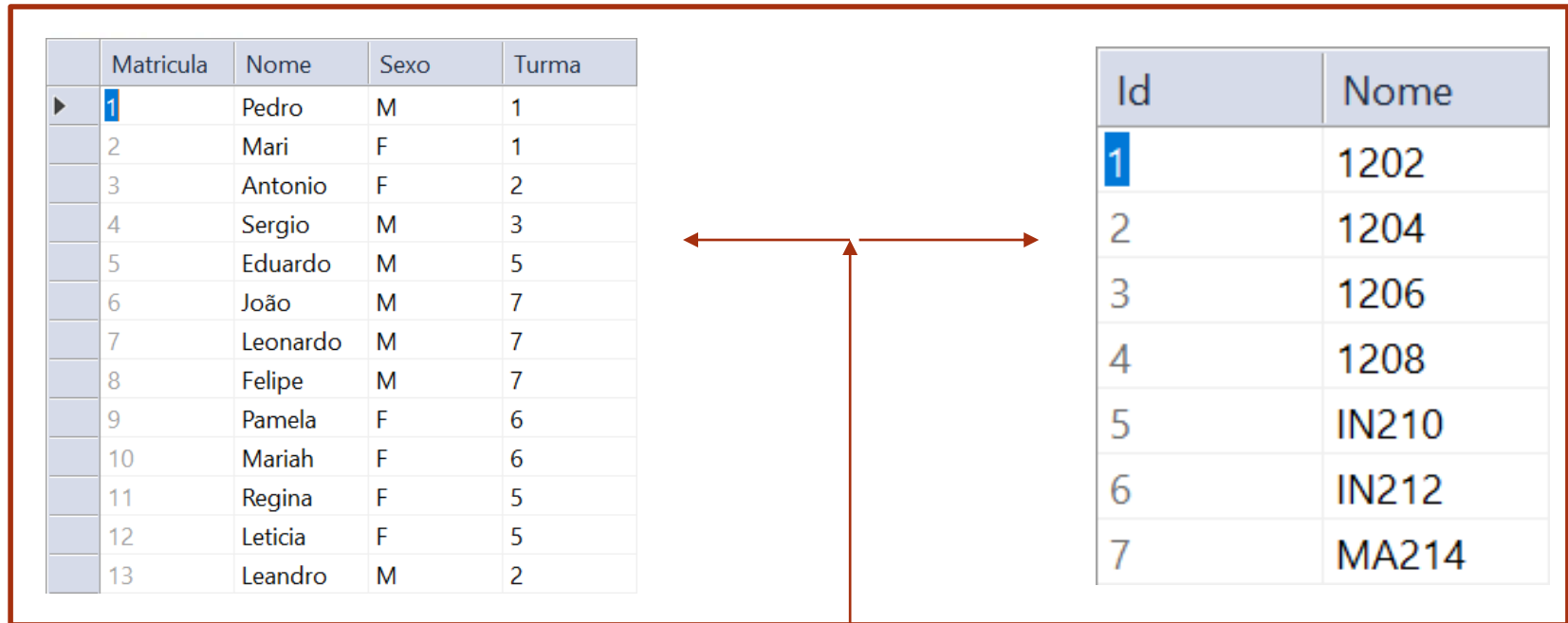
Turma(Id, Descricao)



Geralmente representado através de
Relações/Tabelas.

Exemplo

Projeto Lógico de Banco de Dados



Um conjunto de relações/tabelas preenchidas com alguns registros é denominado uma instância do modelo relacional.

Geralmente representado através de **Relações/Tabelas**.

Exemplo

Projeto **Físico** de Banco de Dados



```
CREATE TABLE [dbo].Turma
(
  [Id] INT NOT NULL PRIMARY KEY
  IDENTITY,
  [Nome] NVARCHAR(50) NOT NULL
)
```

```
CREATE TABLE [dbo].Aluno
(
  [Matricula] INT NOT NULL PRIMARY KEY,
  [Nome] NVARCHAR(50) NULL,
  [Sexo] char(1) NOT NULL CHECK (Sexo
  IN('F', 'M')),
  [Turma] INT NULL FOREIGN KEY
  REFERENCES Turma(Id)
)
```



Geralmente representado através de **Comandos SQL** que realizam o modelo relacional.

Exemplo

Aplicação C#



Colégio
Pedro II

Ao se criar Formulários com muitos controles, é uma boa prática nomear os controles de acordo com algum padrão.

Exemplo:

textBox1 – txtInitial
comboBox1 – cmbClass
checkBox1 – chkFilter
radioButton1 – radFemale
radioButton2 – radMale
Button1 – btnShowStudents
listBox1 - lstNames

The screenshot shows a Windows application window titled "Exercício 03". Inside the window, there is a list box containing the following names: Bernardo, Christiano, Eduardo, Hercília, João, João, Jorge, Juliana, Karine, Larissa, Leonardo, Philipi, Rodrigo, Saulo, and Victor. Below the list box, there is a checkbox labeled "Filtrar Dados". To the right of the checkbox, there is a section labeled "Sexo" with two radio buttons: "Feminino" (selected) and "Masculino". Below the "Filtrar Dados" checkbox, there is a text box labeled "Inicial" and a dropdown menu. At the bottom of the window, there is a button labeled "Exibir Alunos".

Red arrows point from labels to the controls:

- lstNames** points to the list box.
- radFemale** points to the "Feminino" radio button.
- radMale** points to the "Masculino" radio button.
- btnShowStudents** points to the "Exibir Alunos" button.
- chkFilter** points to the "Filtrar Dados" checkbox.
- txtInitial** points to the "Inicial" text box.
- cmbClass** points to the dropdown menu.

Exemplo

Aplicação C#



```
private void btnShowStudents_Click(object sender, EventArgs e)
{
    lstNames.Items.Clear();

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            string sqlStatement = "SELECT * FROM Aluno";

            SqlCommand command = new SqlCommand(sqlStatement, connection);

            using (SqlDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    lstNames.Items.Add(Convert.ToString(reader["Nome"]));
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

O restante do código foi enviado por e-mail para melhor verificação.

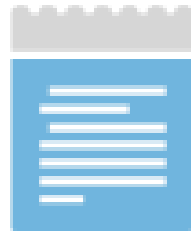
Aplicação CRUD

Parte 02



CREATE

C



READ

R



UPDATE

U



DELETE

D

SQL

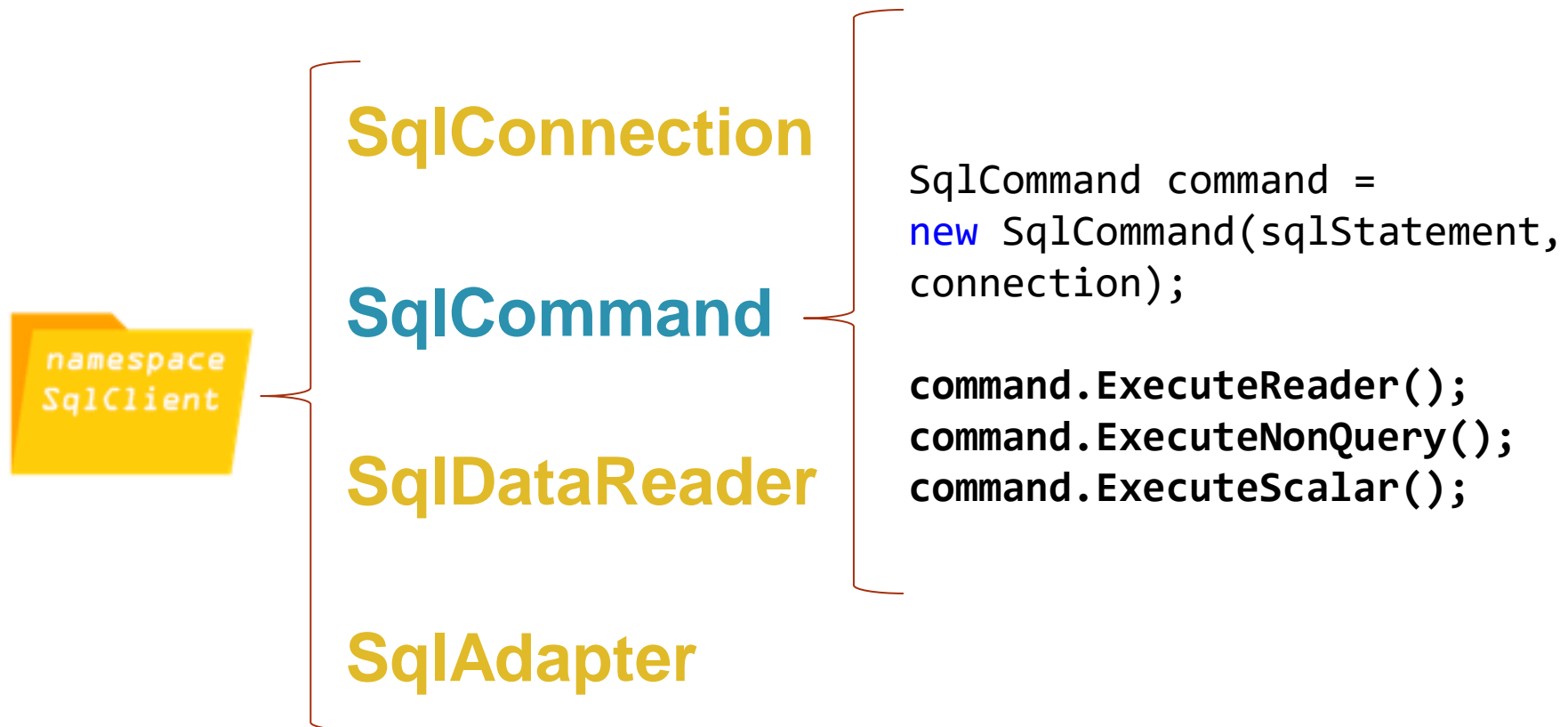
Tipos de comandos



- Os comandos SQL são classificados de acordo com seu propósito. Há comandos de **Definição**, **Consulta/Recuperação**, **Atualização/Manipulação**, **Transação** e **Controle de Permissão**.

Linguagem	Propósito	Exemplo
DDL (Data Definition Language)	Definição	CREATE TABLE, ALTER TABLE, DROP TABLE
DQL (Data Query Language)	Consulta/Recuperação	SELECT
DML (Data Manipulation Language)	Atualização/Manipulação	INSERT, UPDATE, DELETE
TCL (Transaction Control Language)	Transação	COMMIT, SAVEPOINT, ROLLBACK
DCL (Data Control Language)	Controle de Permissão	GRANT, REVOKE

Como enviar comandos **DML** através da aplicação?



```
using System.Data.SqlClient;
```


Propósito dos métodos



ExecuteReader():

- É indicado para comandos SQL do tipo DQL (SELECT).
- Retorna um objeto do tipo **SqlDataReader** que pode ser usado para iterar no resultado do comando DQL.

ExecuteNonQuery():

- É indicado para comandos SQL do tipo DML (INSERT, UPDATE, DELETE)
- Retorna apenas um **int** indicando a quantidade de linhas que foram afetadas pelo comando DML. Para qualquer outro tipo de instrução SQL, o retorno é -1.

ExecuteScalar():

- É indicado para comandos SQL do tipo DQL (SELECT) mas que utilizam alguma função de agregação para obter um resultado específico.
- Retorna apenas o valor da primeira coluna da primeira linha do SELECT. O tipo de retorno é **object**, o que torna comum convertê-lo para **string** ou **int** caso necessário.

Exemplo

Comando **INSERT**



```
try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlStatement = "INSERT INTO Aluno VALUES (50, 'Mateus', 'M', 7)";
        SqlCommand command = new SqlCommand(sqlStatement, connection);

        int affectedRows = command.ExecuteNonQuery();

        if (affectedRows == 1)
            MessageBox.Show("Aluno inserido com sucesso!");
        else
            MessageBox.Show("Não foi possível inserir o aluno...");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Exemplo

Comando UPDATE



```
try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlStatement = "UPDATE Aluno SET Nome=Mateus WHERE
                               Matricula = 1";

        SqlCommand command = new SqlCommand(sqlStatement,
                                             connection);

        int affectedRows = command.ExecuteNonQuery();

        if (affectedRows == 1)
            MessageBox.Show("Aluno atualizado com sucesso!");
        else
            MessageBox.Show("Não foi possível atualizar o
                             aluno...");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Exemplo

Comando **DELETE**



```
try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlStatement = "DELETE FROM Aluno
                                WHERE Matricula = 18";

        SqlCommand command = new SqlCommand(sqlStatement, connection);

        int affectedRows = command.ExecuteNonQuery();

        if (affectedRows == 1)
            MessageBox.Show("Aluno deletado com sucesso!");
        else
            MessageBox.Show("Não foi possível deletar o aluno...");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Exemplo

Função de Agregação



```
try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlStatement = "SELECT COUNT(*) FROM Aluno";

        SqlCommand command = new SqlCommand(sqlStatement, connection);

        int quantidadeDeAlunos = Convert.ToInt32(command.ExecuteScalar());

        MessageBox.Show("Há no colégio: " + quantidadeDeAlunos + " alunos");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Exemplo

Função de Agregação



```
try
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlStatement = "SELECT COUNT(*) FROM Alu
        SqlCommand command = new SqlCommand(sqlStatement, connection);

        int quantidadeDeAlunos = Convert.ToInt32(command.ExecuteScalar());

        MessageBox.Show("Há no colégio: " + quantidadeDeAlunos + " alunos");
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

object

Como a resposta do método ExecuteScalar() vem no tipo object, torna-se necessário convertê-la para um tipo adequado (geralmente int, decimal ou string).

Lista 2 Banco de Dados

Exercício 05 – 06



Exercício 03

1 - Bernardo - M - IN210
 2 - Christiano - M - IN210
 3 - Eduardo - M - IN210
 4 - Hercília - F - IN210
 5 - João - M - IN210
 6 - João - M - IN210
 7 - Jorge - M - IN210
 8 - Juliana - F - IN210
 9 - Karine - F - IN210
 10 - Larissa - F - IN210
 11 - Leonardo - M - IN210
 12 - Philipi - M - IN210
 13 - Rodrigo - M - IN210
 14 - Saulo - M - IN210
 15 - Victor - M - IN210

Matricula:

Nome:

Sexo:

Turma:

numRegistry:

txtName:

cmbSex:

cmbClass:

btnNewStudent:

btnEditStudent:

btnDeleteStudent:

btnLoadStudents:

O restante do código foi enviado por e-mail para melhor verificação.