Aula08 – Acesso a Banco de Dados MySQL em C#

1. Introdução

Nesta aula, você aprenderá a realizar o acesso a um banco de dados MySQL em um projeto C# (Console Application). A abordagem será simples, passo a passo, mostrando desde a configuração do ambiente, criação do banco, construção das classes e métodos para realizar operações como incluir, buscar, alterar e excluir clientes, de forma semelhante ao exemplo do Access.

2. Instalando o MySQL Connector/NET

Para conectar o C# ao MySQL, é necessário instalar o MySQL Connector/NET. Baixe gratuitamente em: https://dev.mysql.com/downloads/connector/net/

Após instalar, adicione a referência MySql.Data.dll ao seu projeto no Visual Studio. Clique com o botão direito em 'Dependencies' ou 'Referências', escolha 'Adicionar Referência', e selecione 'MySql.Data'.

No início do seu código, adicione:

using MySql.Data.MySqlClient;

3. Estrutura do Banco de Dados

No MySQL, crie o banco de dados e a tabela de clientes. Utilize qualquer ferramenta de administração de bancos, como phpMyAdmin, DBeaver ou o próprio terminal MySQL:

```
CREATE DATABASE IF NOT EXISTS aulas;
USE aulas;

CREATE TABLE IF NOT EXISTS clientes (
   id INT AUTO_INCREMENT PRIMARY KEY,
   nome VARCHAR(100),
   email VARCHAR(100)
);

INSERT INTO clientes (nome, email) VALUES
('João Silva', 'joao@email.com'),
('Maria Oliveira', 'maria@email.com');
```

4. Classe Cliente

A classe Cliente representa os dados que iremos manipular no banco de dados. Ela também ajuda a organizar o código.

```
public class Cliente
{
    public int Id { get; set; }
    public string Nome { get; set; }
    public string Email { get; set; }

    public void PreencheCliente(MySqlDataReader resultado)
    {
        this.Id = Convert.ToInt32(resultado[0]);
        this.Nome = resultado[1].ToString();
        this.Email = resultado[2].ToString();
    }

    public void Mostrar()
    {
        Console.WriteLine(this.Id + "\t" + this.Nome + "\t" + this.Email);
    }
  // Métodos Incluir, Buscar, Apagar e Alterar serão implementados abaixo
}
```

Com ela, fica mais fácil interagir com os dados em cada operação.

5. Classe de Acesso ao Banco de Dados

A classe a seguir centraliza todos os métodos de acesso ao MySQL. Ela é estática para que os métodos sejam chamados diretamente.

```
public static class bdcomum
{
    public static MySqlConnection fazerconexao()
    {
        MySqlConnection conectar = new MySqlConnection("server=127.0.0.1; database=aulas; uid=root; pwd=;");
        return conectar;
    }
}
```

6. Operações CRUD

Veja os demais métodos implementados para gerenciar os clientes no banco:

```
// Listar todos os clientes
public static void ListarClientes()
{
    MySqlConnection cn = conexao.fazerconexao();
    cn.Open();
    MySqlCommand cmd = new MySqlCommand();
    MySqlDataReader resultado;
    cmd.CommandText = "SELECT * FROM clientes";
```

```
    cmd.Connection = cn;
    resultado = cmd.ExecuteReader();
    Console.WriteLine("Id\tNome\tEmail");
    while (resultado.Read())
    {
        Cliente cliente = new Cliente();
        cliente.PreencheCliente(resultado);
        cliente.Mostrar();
    }
    resultado.Close();
    cn.Close();
}

// Incluir cliente
public static void InserirCliente(Cliente novo)
{
    MySqlConnection cn = conexao.fazerconexao();
    cn.Open();
    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = cn;
    cmd.CommandText = "INSERT INTO clientes (nome, email) VALUES ('" + novo.Nome + "', '" + novo.Email + "')";
    cmd.ExecuteNonQuery();
    cn.Close();
}

// Buscar cliente por nome
public static void BuscarCliente(string nomeProcurado)
{
    MySqlConnection cn = conexao.fazerconexao();
    cn.Open();
    MySqlCommand cmd = new MySqlCommand();
    MySqlDataReader resultado;
    cmd.CommandText = "SELECT * FROM clientes WHERE nome LIKE '%" + nomeProcurado + "%'";
    cmd.Connection = cn;
    resultado = cmd.ExecuteReader();
    if (resultado.HasRows)
    {
        resultado.Read();
        Cliente cliente = new Cliente();
        cliente.PreencheCliente(resultado);
        cliente.Mostrar();
    }
    else
    {
        Console.WriteLine("Nome não encontrado");
    }
    resultado.Close();
    cn.Close();
}
// Apagar cliente
```

```
public static void ExcluirCliente(int id)
{
   MySqlConnection cn = conexao.fazerconexao();
   cn.Open();
   MySqlCommand cmd = new MySqlCommand();
   MySqlDataReader resultado;
   cmd.Connection = cn;
   cmd.CommandText = "SELECT * FROM clientes WHERE id=" + id;
   resultado = cmd.ExecuteReader();
   if (resultado.HasRows)
   {
      resultado.Read();
      Cliente cliente = new Cliente();
      cliente.PreencheCliente(resultado);
      cliente.Mostrar();
      resultado.Close();
      Console.Write("Deseja mesmo apagar este registro? (S/N): ");
      string resp = Console.ReadLine();
      if (resp.ToUpper() == "S")
      {
         MySqlCommand cmdDelete = new MySqlCommand();
         cmdDelete.Connection = cn;
         cmdDelete.CommandText = "DELETE FROM clientes WHERE id=" + cliente.Id;
         cmdDelete.ExecuteNonQuery();
         Console.WriteLine("Cliente excluído!");
      }
   }
   else
   {
      Console.WriteLine("Cliente não encontrado.");
      resultado.Close();
   }
   cn.Close();
}

// Alterar cliente
public static void AlterarCliente(int id)
{
   MySqlConnection cn = conexao.fazerconexao();
   cn.Open();
   MySqlCommand cmd = new MySqlCommand();
   cmd.Connection = cn;
   cmd.CommandText = "SELECT * FROM clientes WHERE id=" + id;
   MySqlDataReader resultado = cmd.ExecuteReader();

   if (resultado.HasRows)
   {
      resultado.Read();
      Cliente cliente = new Cliente();
      cliente.PreencheCliente(resultado);
```

```
        cliente.Mostrar();

        resultado.Close();

        Console.Write("Novo nome: ");
        string nome = Console.ReadLine();
        if (!string.IsNullOrWhiteSpace(nome)) cliente.Nome = nome;

        Console.Write("Novo email: ");
        string email = Console.ReadLine();
        if (!string.IsNullOrWhiteSpace(email)) cliente.Email = email;

        MySqlCommand cmdUpdate = new MySqlCommand();
        cmdUpdate.Connection = cn;
        cmdUpdate.CommandText = "UPDATE clientes SET nome='" + cliente.Nome + "', email='" + cliente.Email + "'
WHERE id=" + cliente.Id;
        cmdUpdate.ExecuteNonQuery();
        Console.WriteLine("Cliente alterado!");
    }
    else
    {
        Console.WriteLine("Cliente não encontrado.");
        resultado.Close();
    }
    cn.Close();
}
```

7. Programa Principal – Exemplo de Uso

Veja como usar todas as funções criadas em um programa principal. Assim, é possível testar todas as operações do CRUD:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;

namespace BancoMySQL25
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int opcao = -1;
            while (opcao != 0)
            {
```

```
            Console.WriteLine("\nMenu:");
            Console.WriteLine("1 - Listar clientes");
            Console.WriteLine("2 - Buscar cliente");
            Console.WriteLine("3 - Inserir cliente");
            Console.WriteLine("4 - Alterar cliente");
            Console.WriteLine("5 - Excluir cliente");
            Console.WriteLine("0 - Sair");
            Console.Write("Escolha uma opção: ");
            opcao = int.Parse(Console.ReadLine());

            if (opcao == 1)
            {
                Cliente.ListarClientes();
            }
            else if (opcao == 2)
            {
                Console.Write("Digite o nome do cliente a buscar: ");
                string nome = Console.ReadLine();
                Cliente.BuscarCliente(nome);
            }
            else if (opcao == 3)
            {
                Cliente novo = new Cliente();
                Console.Write("Digite o nome: ");
                novo.Nome = Console.ReadLine();
                Console.Write("Digite o email: ");
                novo.Email = Console.ReadLine();
                Cliente.InserirCliente(novo);
                Console.WriteLine("Cliente inserido!");
            }
            else if (opcao == 4)
            {
                Console.Write("Digite o id do cliente a alterar: ");
                int id = int.Parse(Console.ReadLine());
                Cliente.AlterarCliente(id);
                Console.WriteLine("Cliente alterado!");
            }
            else if (opcao == 5)
            {
                Console.Write("Digite o id do cliente a excluir: ");
                int id = int.Parse(Console.ReadLine());
                Cliente.ExcluirCliente(id);
                Console.WriteLine("Cliente excluído!");
            }
        }
    }

  }
}
```