**ATIVIDADE 10-1. ACESSANDO DADOS DE UM BANCO DE DADOS SQL SERVER**

Nesta atividade, você irá começar a familiarizar com o seguinte:

* Estabelecendo uma conexão para o Banco de Dados SQL Server
* Executando *Queries* através de um objeto *Command*.
* Acessando os dados com um objeto *DataReader*.
* Executando uma *Stored Procedure* usando um objeto *Command*.

**Criando uma conexão e executando as Queries SQL**

Para criar uma conexão e executar as *Queries SQL*, siga estes passos:

1. Inicie o *Visual Studio*. Selecione *File* -> *New* -> *Project*.
2. Escolha um projeto *Console Application*. Nomeie o projeto para *Activity10\_1*.
3. Depois do projeto aberto, adicione uma nova classe para o nome do projeto *Author*.
4. Abra o código da classe *Author* no editor de código. Adicione a seguinte instrução *using* no topo do arquivo:

*using System.Data;*

*using System.Data.SqlClient;*

1. Adicione o código para declarar uma variável de nível de classe *(class-level) private* contendo uma *connection string*:

*Class Author*

*{*

*string \_connString = “Data Source=localhost; Initial Catalog=pubs; Integraste Security=True”;*

*}*

1. Adicione um método para a classe que irá ser usado um objeto *Command* para executar uma *query* (consulta) para o contador de número de autores na tabela *Authors*. Por causa você verá retornando apenas um único valor, você irá usar o método *ExecuteScalar* do objeto *Command*.

*public int CountAuthors()*

*{*

*using (SqlConnection pubConnection = new SqlConnection(\_connString))*

*{*

*using (SqlCommand pubCommand = new SqlCommand())*

*{*

*pubCommand.Connection = pubConnection;*

*pubCommand.CommandText = “Select Count(au\_id) from authors”;*

*pubConnection.Open();*

*return (int)pubCommand.ExecuteScalar();*

*}*

*}*

*}*

1. Adicione o seguinte código para o método *Main* da classe *Program*, com isso irá executar o método *GetAuthorCount* definido na classe *Author*:

*static void Main (string[] args)*

*{*

*try*

*{*

*Author author = new Author();*

*Console.WriteLine(author.CountAuthors());*

*Console.ReadLine();*

*}*

*catch (Exception ex)*

*{*

*Console.WriteLine(ex.Message).*

*Console.ReadLine();*

*}*

*}*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *Console* deve lançar o números de autores mostrados. Depois de visualizar a saída, feche o depurador.

**Usando o objeto DataReader para acessar os registros**

Para usar o objeto *DataReader* para acessar os registros, siga estes passos:

1. Abra o código da classe *Author* no editor de código.
2. Adicione um método público para a definição de classe chamada *GetAuthorList* para retorna uma *List* genérica de strings.

*public List<string> GetAuthorList()*

*{*

*}*

1. Adicione o seguinte código, com isso executa uma instrução SQL Server para acessar os últimos nomes dos autores. Um objeto *DataReader* então volta através de registros e criação de listas de nomes que obtém retornado para o cliente.

*List<string> nameList = new List<string>();*

*using(SqlConnection pubConnection = new SqlConnection(\_connString))*

*{*

*using(SqlCommand authorsCommand = new SqlCommand())*

*{*

*authorsCommand.Connection = pubConnection;*

*authorsCommand.CommandText = “Select au\_lname from authors”;*

*pubConnection.Open();*

*using(SqlDataReader authorDataReader = authorsCommand.ExecuteReader())*

*{*

*while (authorDataReader.Read() == true)*

*{*

*nameList.Add(authorDataReader.GetString(0));*

*}*

*return nameList;*

*}*

*}*

*}*

1. Troque o código no método *Main* da classe *Program* para mostrar a lista de nomes na janela console.

*static void Main(string[] args)*

*{*

*try*

*{*

*Author author = new Author();*

*foreach (string name in author.GetAuthorList())*

*{*

*Console.WriteLine(name);*

*}*

*Console.ReadLine();*

*}*

*catch (Exception ex)*

*{*

*Console.WriteLine(ex.Message);*

*Console.ReadLine();*

*}*

*}*

1. Selecione Debug -> Start para rodar o projeto. A janela Console deve lançar mostrando os nomes de autores. Depois de visualizar a saída, pare o depurador.

**Executando uma Stored Procedure usando um objeto Command**

Para executar uma *stored procedure* usando um objeto *Command*, siga estes passos:

1. Abra o código na classe *Author* no editor de código.
2. Adicione um método *public* que irá sobre sobrecarregar o método *GetAuthorList* aceitando um parâmetro do tipo *integer* (inteira)nomeado como *Royalty*. Esta função irá chamar a *stored procedure* pelos *(royalty)* direitos autorais no banco de dados *Pubs*. O procedimento levar uma saída do tipo *integer* (inteira)de porcentagem de direitos autorais e retorna uma lista de Id de autores com as porcentagens.

*public List<string> GetAuthorList(int royalty)*

*{*

*List<string> nameList = new List<string>();*

*using (SqConnection pubConnection = new SqlConnection(\_connString))*

*{*

*using (SqlCommand authorsCommand = new SqlCommand())*

*{*

*authorsCommand.Connection = pubConnection;*

*authorsCommand.CommandType = CommandyTpe.StoredProcedure;*

*authorsCommand.CommandText = “byroyalty”;*

*SqlParameter inputParameter = new SqlParameter();*

*inputParameter.ParameterName = “@percentage”;*

*inputParameter.Direction = ParameterDirection.Input;*

*inputParameter.SqlDbType = SqlDbType.Int;*

*inputParameter.Value = royalty;*

*authorsCommand.Parameters.Add(inputParameter);*

*pubsConnection.Open();*

*using (SqlDataReader authorDataReader = authorsCommand.ExecuteReader())*

*{*

*while( authorDataReader.Read() == true)*

*{*

*nameList.Add(authorDataReader.GetString(0));*

*}*

*}*

*return nameList;*

*}*

*}*

*}*

1. No método *Main* da classe *Program*,prover um parâmetro de saída de 25 para o método *GetAuthorList*.

*foreach (string name in author.GetAuthorList(25))*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *Console* deve lançar mostrando os Ids dos autores. Depois de visualizar a saída, pare o depurador.
2. Quando finalizamos os testes, saia do Visual Studio.