Xây dựng Datalayer trong mô hình 3 lớp:

Xây dựng lớp Cls_ReadConnectionString.

Mục đích đọc chuỗi kết nối từ file Ini

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DTOLayer;

namespace DataLayer
{
   public class Cls_ReadConnectionString
    {
//Các thuộc tính của chuỗi kết nối
       string serverName = string.Empty;
       string databaseName = string.Empty;
       string userId = string.Empty;
       string passWord = string.Empty;
       public string connectionstring = string.Empty;
       bool winNT = true;

       public bool WinNT
       {
           get { return winNT; }
           set { winNT = value; }
       }
       public Cls_ReadConnectionString(string path,bool winNT)
       {
           ReadFileINI(path);
           this.winNT = winNT;
       }
//Phương thức đọc chuỗi kết nối từ file ini gán giá trị vào thuộc tính chuỗi kết nối
       public Cls_ReadConnectionString(string path, ref DTOKetNoi _ketNoi)
       {
           ReadFileINI(path);
           _ketNoi.ServerName = serverName;
           _ketNoi.DatabaseName = databaseName;
           _ketNoi.UserID = userId;
           _ketNoi.PassWord = passWord;
           _ketNoi.WinNT = winNT;

       }
//Phương thức ghép các thuộc tính chuỗi kết nối thành chuỗi kết nối hoàn chỉnh
       private string GhepChuoiKetNoi()
       {
           string connnectionstring = string.Empty;
           if(!string.IsNullOrEmpty(serverName))
           {
               connnectionstring += string.Format("server={0}", serverName);
```

```
            }
            else
            {
                return null;
            }
            if (!string.IsNullOrEmpty(databaseName))
            {
                connnectionstring += string.Format("; database={0}", databaseName);
            }
            else
            {
                return null;
            }
            if(winNT==true)
            {
                connnectionstring += ";Integrated security=true";
            }
            else
            {
                if (!string.IsNullOrEmpty(userId))
                {
                    connnectionstring += string.Format("; uid={0}", userId);
                }
                else
                {
                    return null;
                }
                if (!string.IsNullOrEmpty(passWord))
                {
                    connnectionstring += string.Format("; pwd={0}", passWord);
                }

            }
            return connnectionstring;
        }
//Phương thức đọc giá trị trong file INI
        public void ReadFileINI(string path)
        {
            using (StreamReader sr=new StreamReader(path))
            {
                string line = string.Empty;
                while ((line=sr.ReadLine())!=null)
                {
                    if (!string.IsNullOrEmpty(line))
                    {
                        switch (line.Substring(0, line.IndexOf('=')).ToLower())
                        {
                            case "server":
                                serverName = line.Substring(line.IndexOf('=') + 1);
                                break;
                            case "database":
                                databaseName = line.Substring(line.IndexOf('=') + 1);
                                break;
                            case "uid":
                                userId = line.Substring(line.IndexOf('=') + 1);
```

```
                                break;
                            case "pwd":
                                passWord = line.Substring(line.IndexOf('=') + 1);
                                break;
                            case "winnt":
                                WinNT =
Convert.ToBoolean(line.Substring(line.IndexOf('=') + 1));
                                break;
                        }
                    }
                }

                //Ghep thong ket noi

                connectionstring = GhepChuoiKetNoi();
            }
        }
    }
}
```

Xây dựng lớp Cls_Database

Mục đích thức thi các thuộc tính của ADO.NET để làm việc với SQL Server

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;

namespace DataLayer
{
    public class Cls_Database : IDisposable
    {
        //Các thuộc tính ADO.NET
       private SqlConnection cnn;
       private SqlCommand cmd;
       private SqlDataAdapter da;
      //Đối tượng của lớp đọc kết nối từ file
        Cls_ReadConnectionString readconnect;

        public void Dispose()
        {
            if (cnn != null)
            {
                cnn.Dispose();
                cnn = null;
            }
        }
        //properties
        //contructor
```

```
//Hàm tạo của lớp Cls_Database
  public Cls_Database(string path,bool winNT)
        {
            readconnect = new Cls_ReadConnectionString(path, winNT);
            cnn = new SqlConnection(readconnect.connectionstring);
            cmd = cnn.CreateCommand();

        }
         public Cls_Database(string path, ref DTOLayer.DTOKetNoi _ketnoi)
         {
             readconnect = new Cls_ReadConnectionString(path,ref _ketnoi);

             cnn = new SqlConnection(readconnect.connectionstring);
             cmd = cnn.CreateCommand();

         }
         //Phương thức kiểm tra kết nối
         public bool KiemTraKetNoi(ref string err)
         {
             try
             {
                 if (!string.IsNullOrEmpty(cnn.ConnectionString))
                 {
                     cnn.Open();
                     return true;
                 }
                 else
                 {
                     err = "Chuỗi kết nối rỗng";
                     return false;
                 }
             }
             catch (Exception ex)
             {
                 err = ex.Message;
             }
             finally
             {
                 if (cnn.State == ConnectionState.Open)
                     cnn.Close();
             }
             return false;
         }
//Phương thức thực thi câu lệnh truy vấn Insert, Update, delete
         public bool MyExcuteNonQuery(string sql, CommandType ct, ref int
sodong, ref string err, params SqlParameter[] param)
         {

             if (cnn.State == ConnectionState.Open)
                 cnn.Close();
```

```
            cnn.Open();
            cmd.CommandText = sql;

            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
            }
            try
            {
                sodong = cmd.ExecuteNonQuery();

                return true;
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }
            finally
            {
                cnn.Close();
            }
            return false;
        }
//Phương thức nạp chồng lên phương thức phía trên
        public bool MyExcuteNonQuery(string sql, CommandType ct, ref string
err, params SqlParameter[] param)
        {

            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();
            cmd.CommandText = sql;

            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
            }
            try
```

```
            {
               cmd.ExecuteNonQuery();

                return true;
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }
            finally
            {
                cnn.Close();
            }
            return false;
        }
//phương thức thực thi câu lệnh truy vấn trả về 1 giá trị kiểu Object
        public object MyExcuteScalar(string sql, CommandType ct, ref int
sodong, ref string err, params SqlParameter[] param)
        {
            object obj = null;
            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();
            cmd.CommandText = sql;
            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
            }
            try
            {
                obj = cmd.ExecuteScalar();
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }
            finally
            {
                cnn.Close();
            }
            return obj;
        }
//Phương thức thực thi câu lệnh truy vấn trả về 1 kết quả là đối tượng
SqlDataReader
```

```
        public SqlDataReader MyExcuteReader(string sql, CommandType ct,  ref
string err, params SqlParameter[] param)
        {
            SqlDataReader _reader = null;
            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();
            cmd.CommandText = sql;
            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
            }
            try
            {
                _reader = cmd.ExecuteReader();
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }

            return _reader;
        }

//phuong thức thực thi câu lệnh truy cấn trả về 1 đối tượng DataTable
        public DataTable GetDataTable(string sql, CommandType ct, ref string
err, params SqlParameter[] param)
        {

            DataTable dt=new DataTable();
            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();
            cmd.CommandText = sql;

            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
```

```
            }
            try
            {
                da = new SqlDataAdapter(cmd);
                da.Fill(dt);
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }
            finally
            {
                cnn.Close();
            }
            return dt;
        }
//Phương thực thi câu lệnh truy cấn trả về 1 đối tượng DataSet
        public DataSet GetDataSet(string sql, CommandType ct, ref string err,
params SqlParameter[] param)
        {

            DataSet ds = new DataSet();
            if (cnn.State == ConnectionState.Open)
                cnn.Close();
            cnn.Open();
            cmd.CommandText = sql;

            cmd.CommandTimeout = 600;
            cmd.CommandType = ct;
            cmd.Parameters.Clear();
            if (param != null)
            {
                foreach (SqlParameter p in param)
                {
                    cmd.Parameters.Add(p);
                }
            }
            try
            {
                da = new SqlDataAdapter(cmd);
                da.Fill(ds);
            }
            catch (Exception ex)
            {
                err = ex.Message;
            }
            finally
            {
                cnn.Close();
            }
```

```
            return ds;
        }
    }
}
```