МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

“Харківський авіаційний інститут”

Факультет СУЛА

Кафедра математичного моделювання та штучного інтелекту

**Лабораторна робота №5**

з Організація баз даних

на тему: «ADO.NET: LINQ to SQL»

Виконав: студент 3 курсу групи №335a

напряму підготовки(спеціальності):

122 Комп’ютерні науки

(шифр і назва напряму підготовки(спеціальності))

Гринюк М.О.

(прізвище й ініціали студента)

Прийняв: ст. викл., к.т.н. Коробчинський К.П.

(посада, науковий ступінь, прізвище й ініціали)

Харків – 2020

**Цель работы:** Изучить принципы построения отсоединенный уровня, рассмотрел использование DataSet, DataTable, DataColumn, DataRow.

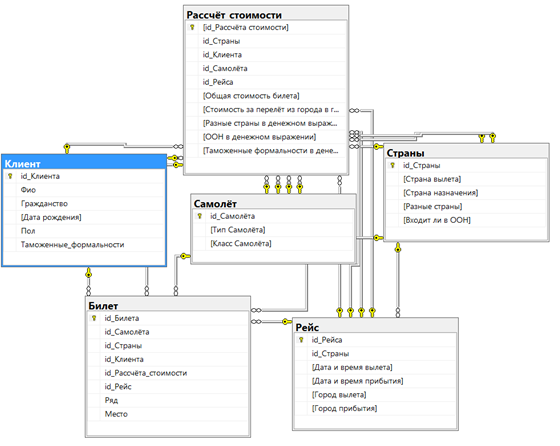
**Ход работы:**

1. Пройти тест
2. Использовать в своём проекте класс логирования **(использовать текстовые файлы)**
3. Показать созданную базу данных
4. Подключить проект C# проверить соединение с БД вывести отладочную информацию в лог файл.
5. **Создать примеры LINQ to SQL**и реализовать действия:
   * Определение контекста данных и моделей
   * Операции с набором в LINQ to SQL
   * Изменение объектов в LINQ to SQL
   * Добавление в LINQ to SQL
   * Удаление в LINQ to SQL
   * Методы ExecuteCommand и ExecuteQuery. Хранимые процедуры
   * Вывод на экран информации по условию своего варианта;
6. Оформить отчёт и сохранить на сервере;
7. **Знать ответы на контрольные вопросы.**

**Содержание работы:**

1. Для своего варианта на основе построенной логической модели своего варианта создать БД;
2. Подключить класс для вывода логирования;
3. Показать пример работы с правильными и не правильными данными;
4. Создать форму для входа и после успешной авторизации запустить основную, если авторизация не успешная, то главная форма не должна создаваться. (повышенная сложность на максимальное количество баллов);
5. Для своего варианта реализовать **LINQ to SQL.**

**Логическая модель БД**



**Выполнение работы**

1. Для своего варианта на основе построенной логической модели своего варианта создать БД;

[Table(Name = "Самолёт")]

public class Airplane

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "id\_Самолёта")]

public int Id { get; set; }

[Column(Name = "[Тип Самолёта]")]

public string Type\_Airplane { get; set; }

[Column(Name = "[Класс Самолёта]")]

public string Class\_Airplane { get; set; }

[Table(Name = "Клиент")]

class Client

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "id\_Клиента")]

public int Id { get; set; }

[Column(Name = "[Фио]")]

public string FullName { get; set; }

[Column(Name = "[Гражданство]")]

public string Nation { get; set; }

[Column(Name = "[Дата\_рождения]")]

public DateTime DateBirth { get; set; }

[Column(Name = "[Пол]")]

public byte Gender { get; set; }

[Column(Name = "[Таможенные\_формальности]")]

public byte Formalities { get; set; }

}

[Table(Name = "Страны")]

class Country

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "id\_Страны")]

public int Id { get; set; }

[Column(Name = "[Страна вылета]")]

public string CountryFrom { get; set; }

[Column(Name = "[Страна назначения]")]

public string CountryTo { get; set; }

[Column(Name = "[Разные страны]")]

public byte DifferentCountries { get; set; }

[Column(Name = "[Входит ли в ООН]")]

public byte IncludeOON { get; set; }

}

[Table(Name = "Customers")]

class Customers

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "idCustomer")]

public int Id { get; set; }

[Column(Name = "[UserName]")]

public string UserName { get; set; }

[Column(Name = "[Pass]")]

public string Pass { get; set; }

}

[Table(Name = "Рейс")]

class Flight

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "id\_Рейса")]

public int Id { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Страны")]

public int IDCountry { get; set; }

[Column(Name = "[Дата и время вылета]")]

public DateTime DateFrom { get; set; }

[Column(Name = "[Дата и время прибытия]")]

public DateTime DateTo { get; set; }

[Column(Name = "[Город вылета")]

public string TownFrom { get; set; }

[Column(Name = "[Город прибытия]")]

public string TownTo { get; set; }

}

[Table(Name = "Managers")]

class Managers

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "idManager")]

public int Id { get; set; }

[Column(Name = "[ManagerName]")]

public string ManagerName { get; set; }

[Column(Name = "[Pass]")]

public string Pass { get; set; }

}

[Table(Name = "Билет")]

class Ticket

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "id\_Билета")]

public int Id { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Самолёта")]

public int IdAirplane { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Страны")]

public int IdCountry { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Клиента")]

public int IdClient { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Рассчёта\_стоимости")]

public int IdCostCalc { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Рейс")]

public int IdFlight { get; set; }

[Column(Name = "[Ряд]")]

public int Row { get; set; }

[Column(Name = "[Место]")]

public int Place { get; set; }

}

[Table(Name = "Users")]

class Users

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "idUser")]

public int Id { get; set; }

[Column(Name = "[LoginName]")]

public string LoginName { get; set; }

[Column(Name = "[Pass]")]

public string Pass { get; set; }

}

[Table(Name = "Рассчёт\_стоимости")]

class Сost\_Сalculation

{

[Column(IsPrimaryKey = true, IsDbGenerated = true, Name = "[id\_Рассчёта стоимости]")]

public int Id { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Страны")]

public virtual Country IdCountry { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Клиента")]

public virtual Country IdClient { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Самолёта")]

public virtual Country IdAirplane { get; set; }

[Column(IsDbGenerated = true, Name = "id\_Рейса")]

public int IdFlight { get; set; }

[Column(IsDbGenerated = true, Name = "Общая стоимость билета")]

public int TotalCost { get; set; }

[Column(Name = "[Стоимость за перелёт из города в город]")]

public int CostFlight { get; set; }

[Column(Name = "[Разные страны в денежном выражении]")]

public int DifferentCountriesInValue { get; set; }

[Column(Name = "[ООН в денежном выражении]")]

public int OONinValue { get; set; }

[Column(Name = "[Таможенные формальности в денежном выражении]")]

public int FormalnostiInValue { get; set; }

}

2. Подключить класс для вывода логирования;

public class ConnectedLayer

{

public string connectionString = System.Configuration.ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;

private SqlCommand oSqlCom;

private SqlDataAdapter oSqlDtAdptr;

public string str1 { get; set; }

public string str2 { get; set; }

public DataTable datatable1;

public DataTable datatable2;

private DataSet ds;

private DataTable dt;

Logger logger = LogManager.GetCurrentClassLogger();

SqlConnectionStringBuilder builder;

public ConnectedLayer()

{

builder = new SqlConnectionStringBuilder(connectionString);

//ds = new DataSet();

var oSqlCon = new SqlConnection(connectionString);

using (SqlConnection connection = new SqlConnection(connectionString))

{

logger.Trace(

"Connection to" + Environment.NewLine +

"Data Source: " + oSqlCon.DataSource + Environment.NewLine +

"Database: " + oSqlCon.Database + Environment.NewLine +

"State: " + oSqlCon.State +

"User: " + builder.UserID +

"Catalog: " + builder.InitialCatalog +

"Server: " + builder.DataSource

);

Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

config.ConnectionStrings.ConnectionStrings.Add(

new ConnectionStringSettings("ConnectionStr", "SomeConnectionString")

);

config.Save();

ConnectionStringsSection section = config.GetSection("connectionStrings") as ConnectionStringsSection;

if (section.SectionInformation.IsProtected)

{

section.SectionInformation.UnprotectSection();

}

else

{

section.SectionInformation.ProtectSection("DataProtectionConfigurationProvider");

}

config.Save();

}

}

3.

**Вывод:** Изучил принципы построения отсоединенный уровня, рассмотрел использование DataSet, DataTable, DataColumn, DataRow.