Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Методы сбора, хранения, обработки и анализа данных»

Студент: Жук С.С.

ФИТ 3 курс 2 группа

Преподаватель: Комкова А.В.

Содержание

[**Лабораторная работа № 1** 3](#_Toc190648900)

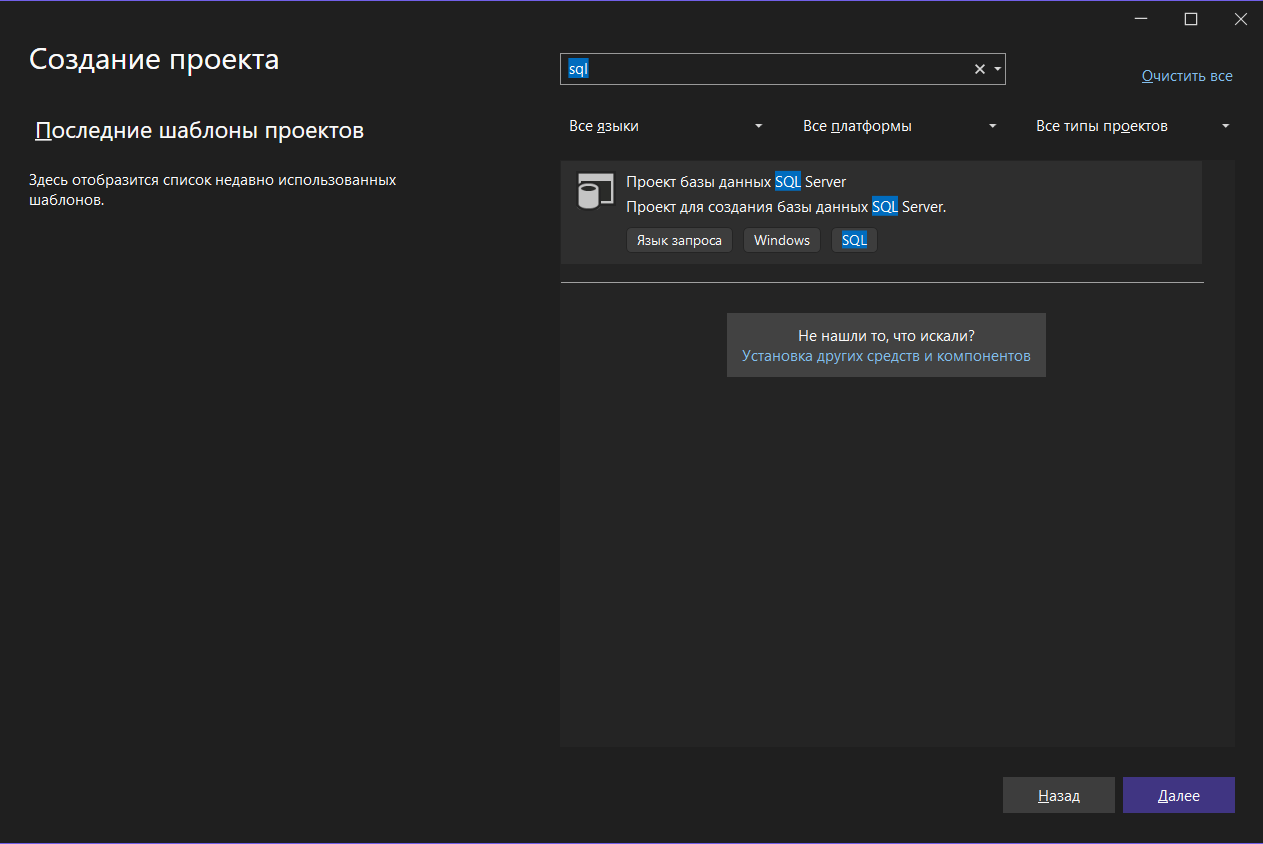
[**Лабораторная работа № 2** 11](#_Toc190648901)

# **Лабораторная работа № 1**

Задание:

* + - 1. Создать хранимую процедуру/функцию CLR по варианту.
      2. Создать определяемый пользователем тип CLR по варианту.
      3. Добавить все вышеперечисленное в сборку CLR и продемонстрировать выполнение.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Услуги аренды | Вычисление среднего значения без максимального и минимального значений | Паспортные данные |

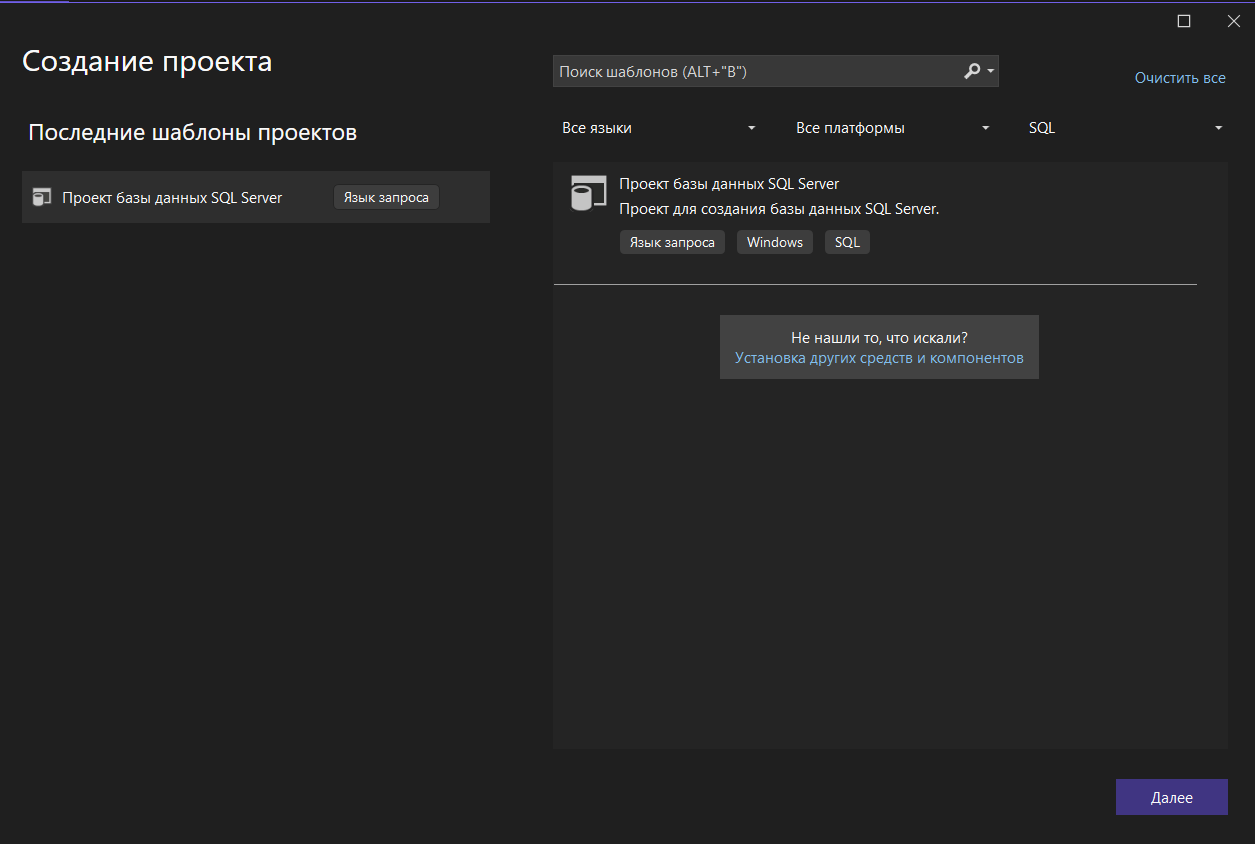


1. Создать хранимую процедуру/функцию CLR по варианту;

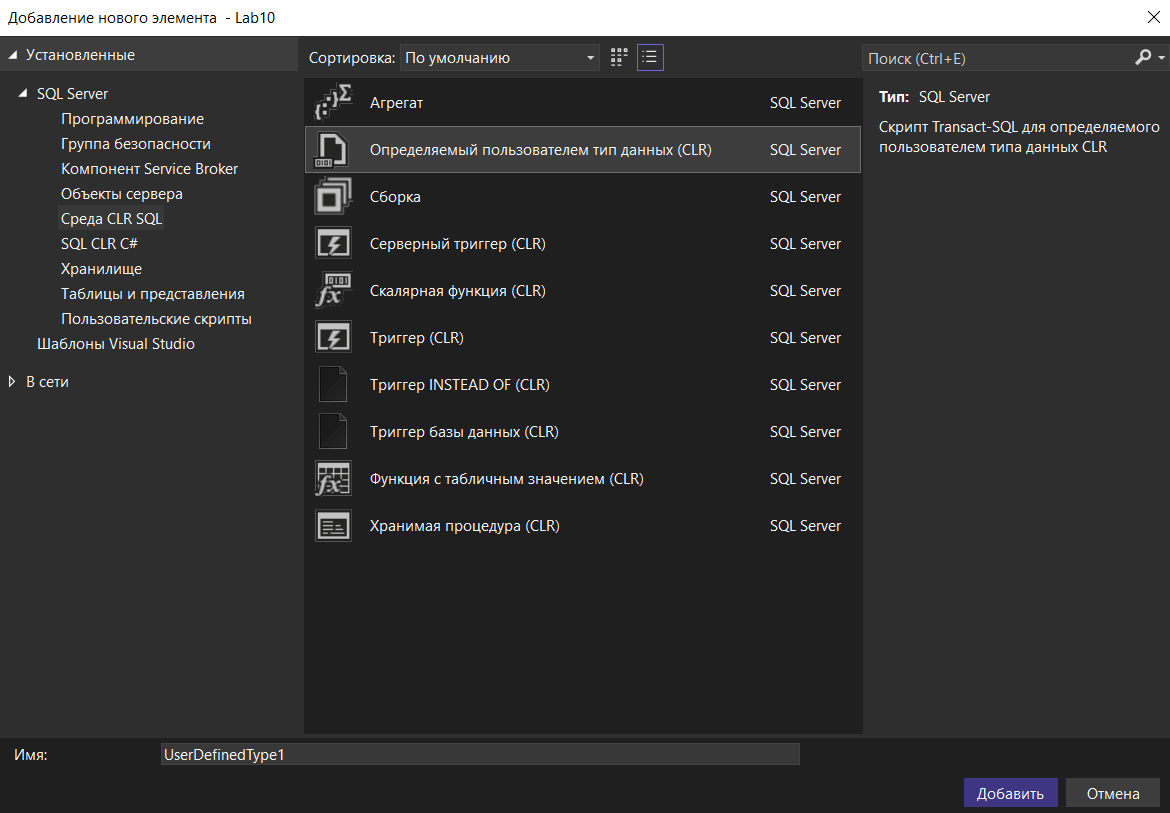
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.SqlClient;  using System.Data.SqlTypes;  using System.Windows.Markup;  using Microsoft.SqlServer.Server;  public partial class UserDefinedFunctions  {  [Microsoft.SqlServer.Server.SqlFunction(DataAccess = DataAccessKind.Read)]  public static SqlDouble CalculateAverageWithoutMinMax(SqlString values)  {  if (values.IsNull)  {  return SqlDouble.Null;  }  string[] stringValues = values.Value.Split(',');  if (stringValues.Length < 3)  {  return SqlDouble.Null;  }  double[] numbers = new double[stringValues.Length];  for (int i = 0; i < stringValues.Length; i++)  {  if (!double.TryParse(stringValues[i].Trim(), out numbers[i]))  {  return SqlDouble.Null;  }  }  double min = numbers[0], max = numbers[0], sum = 0;  int count = numbers.Length;  foreach (double num in numbers)  {  if (num < min)  {  min = num;  }  if (num > max)  {  max = num;  }  sum += num;  }  sum -= (min + max);  count -= 2;  if (count <= 0) return SqlDouble.Null;  double avg = sum / (numbers.Length - 2);  return new SqlDouble(avg);  }  } |

1. Создать определяемый пользователем тип CLR по варианту;

* Создадим проект VS по шаблону:



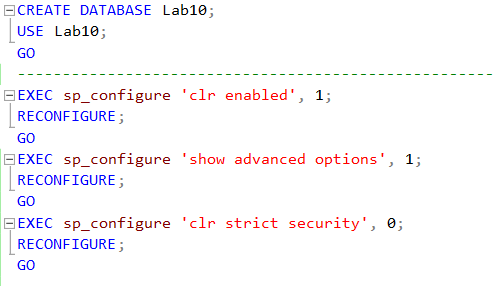
* Добавим шаблонные классы для типа и функции:



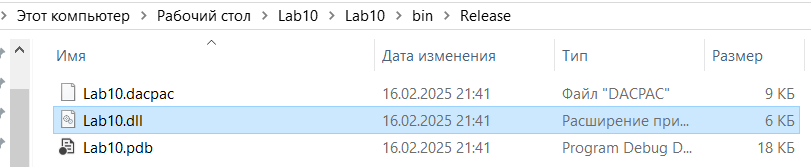
* Напишем реализуем необходимые интерфейсы и логику функции и поля типа

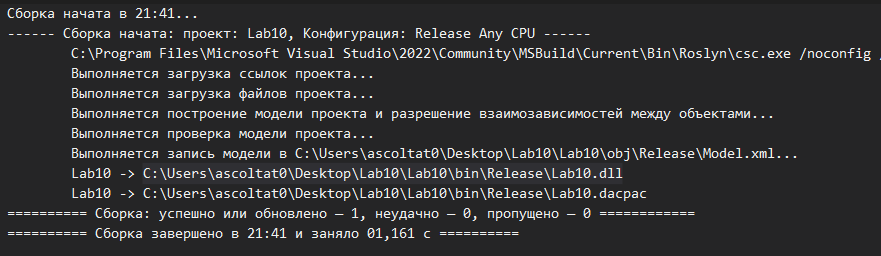
|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Data.SqlTypes;  using System.IO;  using Microsoft.SqlServer.Server;  [Serializable]  [Microsoft.SqlServer.Server.SqlUserDefinedType(Format.UserDefined, IsByteOrdered = true, MaxByteSize = 8000)]  public struct PassportData : INullable, IBinarySerialize  {  public override string ToString()  {  return $"{PassportSeries} {PassportNumber}";  }  public bool IsNull  {  get  {  return \_null;  }  }  public static PassportData Null  {  get  {  PassportData h = new PassportData();  h.\_null = true;  return h;  }  }  public void Read(System.IO.BinaryReader r)  {  if (r != null)  {  PassportSeries = r.ReadString();  PassportNumber = r.ReadInt32();  }  }  public void Write(System.IO.BinaryWriter w)  {  if (w != null)  {  w.Write(PassportSeries);  w.Write(PassportNumber);  }  }  public static PassportData Parse(SqlString s)  {  if (s.IsNull)  return Null;  string[] parts = s.Value.Split(' ');  if (parts.Length != 2)  throw new ArgumentException("Неверный формат. Используйте: 'Series Number'.");  PassportData passportData = new PassportData();  passportData.PassportSeries = parts[0].Trim();  passportData.PassportNumber = int.Parse(parts[1].Trim());  return passportData;  }  public string PassportSeries { get; set; }  public int PassportNumber { get; set; }  private bool \_null;  } |

* Изменим конфиг бд для предотвращения ошибок CLR



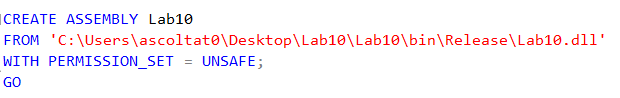
* Скомпилируем библиотеку





3.Добавить все вышеперечисленное в сборку CLR и продемонстрировать выполнение.

* Добавим сборку

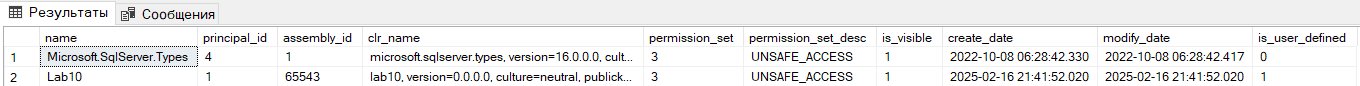


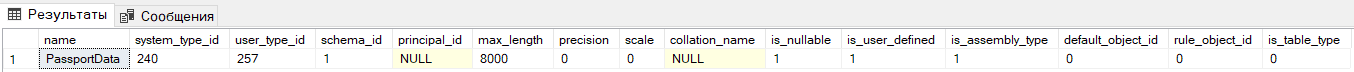
* Создадим тип из сборки в БД



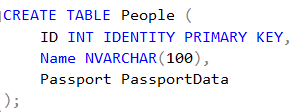
* Проверим тип и сборку соответственно:

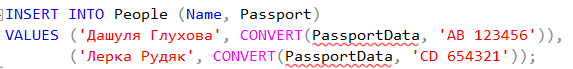
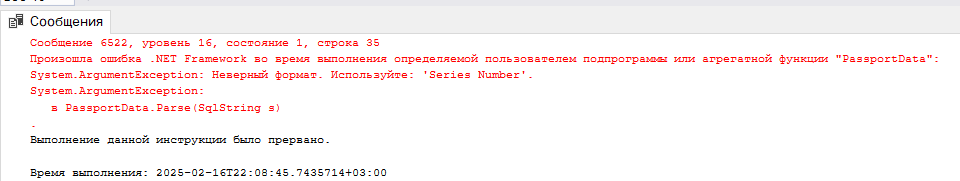




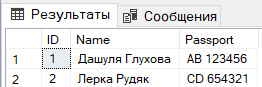


* Создадим таблицу с типом данных PassportData:

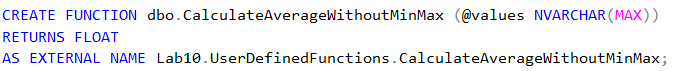


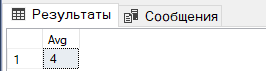


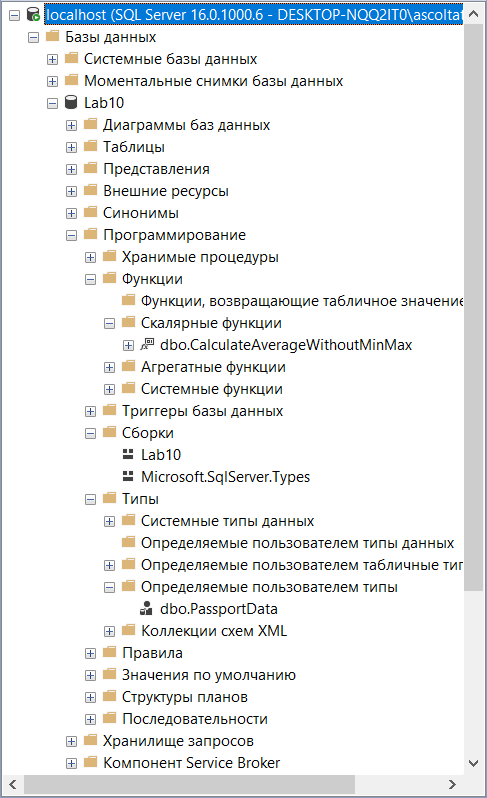


* Создадим функцию из сборки в БД









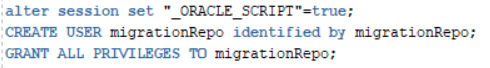
# **Лабораторная работа № 2**

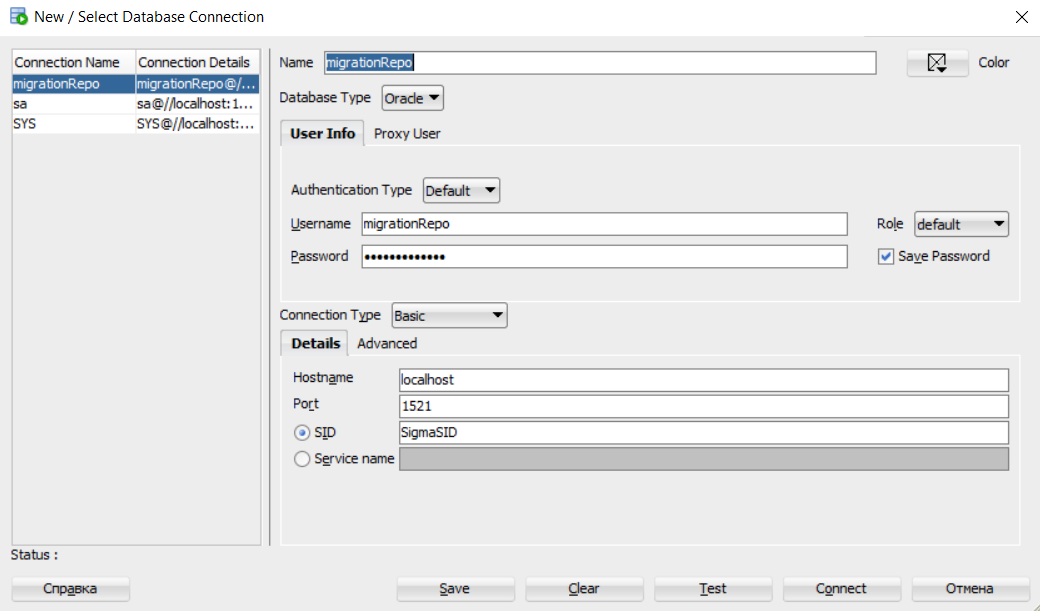
**Миграция базы данных из MS SQL Server в Oracle**

Задание:

1. Произведите миграцию структуры, программного кода и данных базы данных из СУБД MS SQL Server в СУБД Oracle любым удобным для себя способом.
2. В отчете опишите процесс миграции, возникшие проблемы и способы их решения.

От SYS:





2 способа создания репозитория:

