

Кафедра «Систем обработки информации и управления»

Лабораторная работа №3
по курсу Постреляционные базы данных

Тема: «Работа с объектно-реляционной БД – сложные типы данных»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-22М

Сметанкин К.И.

" " _____ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.

к.т.н., доцент

" " _____ 2020 г.

Пункты задания для выполнения:

1. Создание пользовательского типа (UDT)

В среде MS Visual Studio на языке C# создать пользовательский тип данных, например, **Area** с полями:

- **W** - ширина, скрытое,
- **H** - длина, скрытое,
- **square()** - метод экземпляра — возвращает площадь фигуры,
- **help()** - метод класса — строка, содержащая пояснение к строке для инициализации объекта.
- Переопределить метод **Parse** — разбор строки инициализации. Исходная строка содержит данные в формате «Ширина-Длина».
- Переопределить метод **toString()** - обратная к Parse, возвращает значение ширины, длины, площади через дефис.
- Вычисляемое поле **L** – периметр (доступны методы Get и Set), изменяет длину и ширину (делает их равными).

Создать сборку, подключить ее к БД на SQL Server, создать тип данных в БД.

2. Работа с переменной пользовательского типа

В среде SQL Server Management Studio в окне запроса создать переменную типа **Circle** и проверить методы работы с ней:

```
declare @c тип
set @c='значение'
select @c.ToString(), @c.поле, @c.метод_экз()
set @c.поле=значение
select @c.ToString(), @c.поле
set @c=NULL
select @c.ToString()
select @c::метод_класса()
```

3. Добавление переменной пользовательского типа в таблицу

Добавить поле **hsize** нового типа к имеющейся таблице, например, **Отелей**. Продемонстрировать в окне запросов выполнение запросов select и update к таблице с обращением к новому полю.

Открыть таблицу на редактирование и заполнить значения нового поля.

В окне запросов составить и выполнить запросы к таблице, например:

- отобразить отели с площадью более 100,
- увеличить периметр в 2 раза для отелей из Москвы.

Реализация

Пользовательский тип C#

```
using System;
using System.Data;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;
using System.Text;

[assembly: System.CLSCompliantAttribute(true)]
[Serializable]
[Microsoft.SqlServer.Server.SqlUserDefinedType(Format.UserDefined,
    IsFixedLength = false, MaxByteSize = -1, IsByteOrdered = true)]
public class UserFullname : INullable, IBinarySerialize
{
    private string _text;
    private char _delimiter = ' ';
    private bool _null;
    public override string ToString()
    {
        return _text;
    }
    public bool IsNull
    {
        get
        {
            return _null;
        }
    }
    public static UserFullname Null
    {
        get
        {
            UserFullname h = new UserFullname();
            h._null = true;
            return h;
        }
    }
    public static UserFullname Parse(SqlString s)
    {
        if (s.IsNull)
            return Null;
        UserFullname text = new UserFullname();
        string str = s.ToString();
        text._text = str;
        return text;
    }
    public char SetDelimiter(char del)
    {
        _delimiter = del;

        return _delimiter;
    }
    public string GetFirstName()
    {
        string[] array = _text.Split(_delimiter);

        return array[0];
    }
}
```

```

public string GetMiddleName()
{
    string[] array = _text.Split(_delimiter);
    if (array.Length < 2) {
        return "";
    }

    return array[1];
}

public string GetLastName()
{
    string[] array = _text.Split(_delimiter);
    if (array.Length < 3)
    {
        return "";
    }

    return array[2];
}

public void Read(System.IO.BinaryReader r)
{
    _null = r.ReadBoolean();
    if (!_null)
        _text = new String(r.ReadChars(r.ReadInt32()));
}

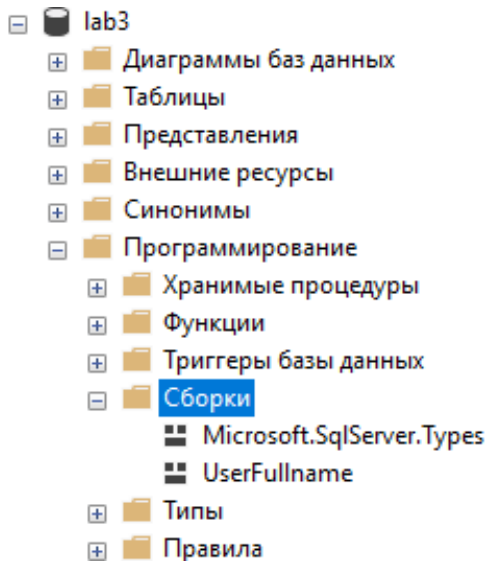
public void Write(System.IO.BinaryWriter w)
{
    w.Write(_null);
    if (!_null)
    {
        w.Write(_text.Length);
        for (int i = 0; i < _text.Length; ++i)
            w.Write(_text[i]);
    }
}
}

```

Создание сборки

Собран проект, полученный файл dll загружен в БД

```
-- создание сборки
CREATE ASSEMBLY UserFullname
FROM 'C:\Users\Кирилл\source\repos\USQLCSharpProject1\USQLCSharpProject1\bin\Debug\USQLCSharpProject1.dll'
WITH PERMISSION_SET = SAFE;
```



После создан тип UserFullname

```
-- объявление типа
CREATE TYPE [UserFullname] EXTERNAL NAME [UserFullname];
```

Проверка методов работы

- ToString - преобразование в строковый тип
- GetFirstName - получение имени
- GetMiddleName - отчества
- GetLastName - фамилии

```
-- использование
declare @tryWithMyText [UserFullname];
set @tryWithMyText = 'Кирилл Игоревич Сметанкин';
select @tryWithMyText;
-- select @tryWithMyText.SetDelimiter('');
select @tryWithMyText.ToString();
select @tryWithMyText.GetFirstName();

set @tryWithMyText = null;
select @tryWithMyText.ToString();
select @tryWithMyText.GetFirstName();
```

00 %

Результаты Сообщения

	(Отсутствует имя столбца)
1	0x0019000000D09AD0B8D180D0B8D0BBD0BB20D098D0B3D0B...

	(Отсутствует имя столбца)
1	Кирилл Игоревич Сметанкин

	(Отсутствует имя столбца)
1	Кирилл

	(Отсутствует имя столбца)
1	NULL

	(Отсутствует имя столбца)
1	NULL

SQL

Создана таблица Test с колонкой пользовательского типа

```
-- создание таблицы с типом
CREATE TABLE Test(
    id int,
    avatar varchar(50),
    fullname UserFullName
);

INSERT INTO TEST (id, avatar, fullname) VALUES
(1, '/static/1.jpg', 'Лука Андреевич Пестов'),
(2, '/static/default.png', 'Зиновий Иванович Бондаренко'),
(3, '', 'Люся Ивановна Сасько');

-- пример использования таблицы
select
    id,
    avatar,
    fullname.ToString() as fullname,
    fullname.GetFirstName() as firstname,
    fullname.GetMiddleName() as middlename,
    fullname.GetLastName() as lastname
from test;
```

00 %

Результаты

Сообщения

	id	avatar	fullname	firstname	middlename	lastname
1	1	/static/1.jpg	Лука Андреевич Пестов	Лука	Андреевич	Пестов
2	2	/static/default.png	Зиновий Иванович Бондаренко	Зиновий	Иванович	Бондаренко
3	3		Люся Ивановна Сасько	Люся	Ивановна	Сасько

Работа с записями таблицы

```
update test set fullname = 'Простая' where fullname.GetLastName() = 'Сасько';
select
    id,
    avatar,
    fullname.ToString() as fullname,
    fullname.GetFirstName() as firstname,
    fullname.GetMiddleName() as middlename,
    fullname.GetLastName() as lastname
from test;
```

6					
Результаты					
id	avatar	fullname	firstname	middlename	lastname
1	/static/1.jpg	Лука Андреевич Пестов	Лука	Андреевич	Пестов
2	/static/default.png	Зиновий Иванович Бондаренко	Зиновий	Иванович	Бондаренко
3		Простая	Простая		

Исходный текст скрипта

```
-- создание сборки
CREATE ASSEMBLY UserFullname
FROM
    'C:
\Users\Кирилл\source\repos\USQLCSharpProject1\USQLCSharpProject1\bin\Debug\USQLC
SharpProject1.dll' WITH PERMISSION_SET = SAFE;

-- объявление типа
CREATE TYPE [UserFullname] EXTERNAL NAME [UserFullname];

-- использование
DECLARE @tryWithMyText[UserFullname];

SET @tryWithMyText = 'Кирилл Игоревич Сметанкин';

SELECT @tryWithMyText;

-- select @tryWithMyText.SetDelimiter('');
SELECT @tryWithMyText.ToString();

SELECT @tryWithMyText.GetFirstName();

SET @tryWithMyText = NULL;

SELECT @tryWithMyText.ToString();

SELECT @tryWithMyText.GetFirstName();

-- создание таблицы с типом
CREATE TABLE Test
(
```



```
    id      int,  
    avatar  varchar(50),  
    fullname UserFullName  
);
```

```
INSERT INTO TEST (id, avatar, fullname)  
VALUES (1, '/static/1.jpg', 'Лука Андреевич Пестов'),  
      (2, '/static/default.png', 'Зиновий Иванович Бондаренко'),  
      (3, '', 'Люся Ивановна Сасько');
```

-- пример использования таблицы

```
SELECT id,  
       avatar,  
       fullname.ToString()      AS fullname,  
       fullname.GetFirstName()  AS firstname,  
       fullname.GetMiddleName() AS middlename,  
       fullname.GetLastName()   AS lastname  
FROM test;
```

```
UPDATE  
    test  
SET fullname = 'Люся Ивановна Простая'  
WHERE fullname.GetLastName() = 'Сасько';
```

```
SELECT id,  
       avatar,  
       fullname.ToString()      AS fullname,  
       fullname.GetFirstName()  AS firstname,  
       fullname.GetMiddleName() AS middlename,  
       fullname.GetLastName()   AS lastname  
FROM test;
```

Литература:

1. Пирогов В. SQL Server 2005. Программирование клиент-серверных приложений. - Спб.: БХВ- Петербург. - 2006 г.
2. Основы программирования на С#. - Электронный ресурс [<http://www.intuit.ru>]