



Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

## **Постреляционные базы данных**

### ***Отчет по лабораторной работе № 3***

#### **«Работа с объектно-реляционной БД – сложные типы данных»**

Выполнил:  
студент группы ИУ5 – 23М

Крутов Т.Ю.

Преподаватель:

Виноградова М.В.

2020г.

Цель работы:

- Изучить различные технологии работы с ОР СУБД.
- Освоить языки и технологии SQL\PSM.
- Получить навыки программирования на стороне сервера.

## Создание пользовательского типа (UDT)

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

namespace lab3
{
    [Serializable]
    [Microsoft.SqlServer.Server.SqlUserDefinedType(Format.Native,
        IsByteOrdered = true, ValidationMethodName = "ValidateImage")]
    public struct Images : INullable
    {
        private bool is_Null;
        private Int32 _height;
        private Int32 _width;
        private Int32 _depth;

        public bool IsNull
        {
            get
            {
                return (is_Null);
            }
        }

        public static Images Null
        {
            get
            {
                Images img = new Images();
                img.is_Null = true;
                return img;
            }
        }

        public override string ToString()
        {
            if (this.IsNull)
                return "NULL";
            else
            {
                StringBuilder ImageBuilder = new StringBuilder();
                ImageBuilder.Append(_height);
                ImageBuilder.Append(",");
                ImageBuilder.Append(_width);
                ImageBuilder.Append(",");
                ImageBuilder.Append(_depth);
                return ImageBuilder.ToString();
            }
        }
    }
}
```

```

[SqlMethod(OnNullCall = false)]

public static Images Parse(SqlString str )
{
    if (str.IsNull)
        return Null;

    Images img = new Images();
    string[] hw = str.Value.Split(",".ToCharArray());
    img.height = Int32.Parse(hw[0]);
    img.width = Int32.Parse(hw[1]);
    img.depth = Int32.Parse(hw[2]);

    if (!img.ValidateImage())
        throw new ArgumentException("Invalid XY coordinate values.");
    return img;
}

private bool ValidateImage()
{
    // Валидация размеров изображений
    if ((_height >= 0) && (_width >= 0) && (_depth >= 0))
    {
        return true;
    }
    else
    {
        return false;
    }
}

public Int32 height
{
    get { return this._height; }
    set
    {
        Int32 temp = _height;
        _height = value;
        if (!ValidateImage())
        {
            _height = temp;
            throw new ArgumentException("Invalid height.");
        }
    }
}

public Int32 width
{
    get { return this._width; }
    set
    {
        Int32 temp = _width;
        _width = value;
        if (!ValidateImage())
        {
            _width = temp;
            throw new ArgumentException("Invalid width.");
        }
    }
}

public Int32 depth
{
    get { return this._depth; }
    set
    {

```

```

        Int32 temp = _depth;
        _depth = value;
        if (!ValidateImage())
        {
            _depth = temp;
            throw new ArgumentException("Invalid depth.");
        }
    }

    [SqlMethod(OnNullCall = false)]
    public int Size/Images img)
    {
        return (img._height * img._width * img._depth);
    }
    [SqlMethod(OnNullCall = false)]

    public Images transpose/Images img)
    {
        Images img_trans = new Images();
        img_trans._width = img._height;
        img_trans._height = img._width;
        img_trans.depth = img.depth;
        return img_trans;
    }

    public Images compress/Images img, Int32 k_ratio)
    {
        Images img_compress = new Images();
        img_compress._width = img._height * (1 / k_ratio);
        img_compress._height = img._width * (1 / k_ratio);
        img_compress.depth = img.depth * (1 / k_ratio);
        return img_compress;
    }
}
}
}

```

Работа с переменной пользовательского типа:

```

declare @i Images;
set @i = '2560, 1440 ,16';
select @i.height,@i.width, @i.depth;

select @i.transpose(@i).height, @i.transpose(@i).width, @i.transpose(@i).depth
select cast(@i as varchar), @i.ToString(), @i.Size(@i), @i;

set @i.height=20;

select @i.height,@i.width

set @i=NULL

select @i.height,@i.width, @i.ToString();

-----Метод транспонирования -----

declare @i Images;
declare @it Images;

```

```

set @i = '2560, 1440 ,16';
set @it = @i.transpose(@i);
select @it.height,@it.width, @it.depth;
select @it

```

-----Метод сжатия -----

```

declare @i Images;
declare @it Images;
set @i = '2560, 1440 ,16';
set @it = @i.compress(@i,2);
select @it.height,@it.width, @it.depth;

```

Добавление переменной пользовательского типа в таблицу

```

GO
sp_configure 'clr enabled', 1;
GO
RECONFIGURE;
GO
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO

```

```

CREATE ASSEMBLY Images
FROM 'C:\\testDB\\lab3.dll'
WITH PERMISSION_SET = SAFE;

```

```

CREATE TYPE dbo.Images
EXTERNAL NAME Images.[lab3.Images];
go
go

```

```

CREATE TABLE [dbo].[imgOrder](
  id int IDENTITY(1,1) NOT NULL PRIMARY KEY,
  imgOwner nvarchar(120),
  img dbo.Images,
  size integer
);
GO
insert into imgOrder(imgOwner,img) values ('Иванов Константин','1920, 1080, 16')
insert into imgOrder(imgOwner,img) values ('Иванов Иван Петрович','1920, 1080,
32');
insert into imgOrder(imgOwner,img) values ('Михайлов Пётр Сергеевич','2560, 1440,
16');
insert into imgOrder(imgOwner,img) values ('Николаенко Виктор
Александрович','2560, 1440, 32');
insert into imgOrder(imgOwner,img) values ('Петрова Екатерина','720, 480, 32');
insert into imgOrder(imgOwner,img) values ('Роскосмос Плутон Аркадьевич','200,
300, 8');

```

```

select id, imgOwner, img.height, img.width, img.depth, size from imgOrder;

```

```

update imgOrder
set size = img.Size(img)

```

```
where id =5
```

```
select * from imgOrder
```

```
select id, imgOwner, img.height ,img.width, img.depth, size from imgOrder where  
img.height > 1950
```

### **Список используемой литературы:**

1. Методические указания к лабораторной работе.