

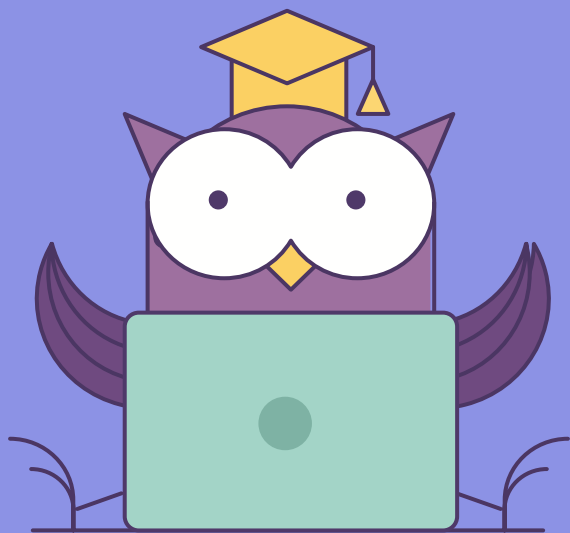


ОНЛАЙН-ОБРАЗОВАНИЕ


Не забыть включить
запись!



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо
Или напишите, какие есть проблемы

CLR

Курс “MS SQL Server разработчик”
Группа 2021-03



Модуль 1

Использование БД

- Тема 1. Основные компоненты и архитектура SQL Server. ДЗ
- Тема 2. Оператор SELECT и простые фильтры, GROUP BY, HAVING. ДЗ
- Тема 3. Подзапросы, CTE, временные таблицы. ДЗ
- Тема 4. Введение в индексы и планы запросов
- Тема 5. Операторы CROSS APPLY, PIVOT, CUBE. ДЗ
- Тема 6. Оконные функции. ДЗ
- Тема 7. Выборки из xml и json полей. ДЗ
- Тема 8. Динамический SQL. ДЗ
- Тема 9. Итоговое занятие по SELECT
- Тема 10. Операторы изменения данных. ДЗ
- Тема 11. Уровни изоляции транзакций
- Тема 12. Хранимые процедуры, функции, триггеры, курсоры. ДЗ
- Тема 13. Создание и использование CLR. ДЗ



Модуль 2

Проектирование БД



Модуль 3

Оптимизация и Администрирование БД



Варианты ДЗ (сделать любой один):

1. Взять готовую dll, подключить ее и продемонстрировать использование. Например, <https://sqlsharp.com>
2. Взять готовые исходники из какой-нибудь статьи, скомпилировать, подключить dll, продемонстрировать использование.
Например,
 - <https://www.sqlservercentral.com/articles/xlsexport-a-clr-procedure-to-export-proc-results-to-excel>
 - <https://www.mssqltips.com/sqlservertip/1344/clr-string-sort-function-in-sql-server/>
 - <https://www.sqlservercentral.com/articles/practical-usage-of-sqlclr-building-querytohtml-function>
 - http://tool-bar.ru/CLR_SQL/index.htm
3. Написать полностью свое:
 - Тип: JSON с валидацией, IP / MAC - адреса, ...
 - Функция: работа с JSON, архивация (сжатие), шифрование ...
 - Агрегат: аналог STRING_AGG, ...
 - (любой ваш вариант)

01

Что такое .NET, CLR

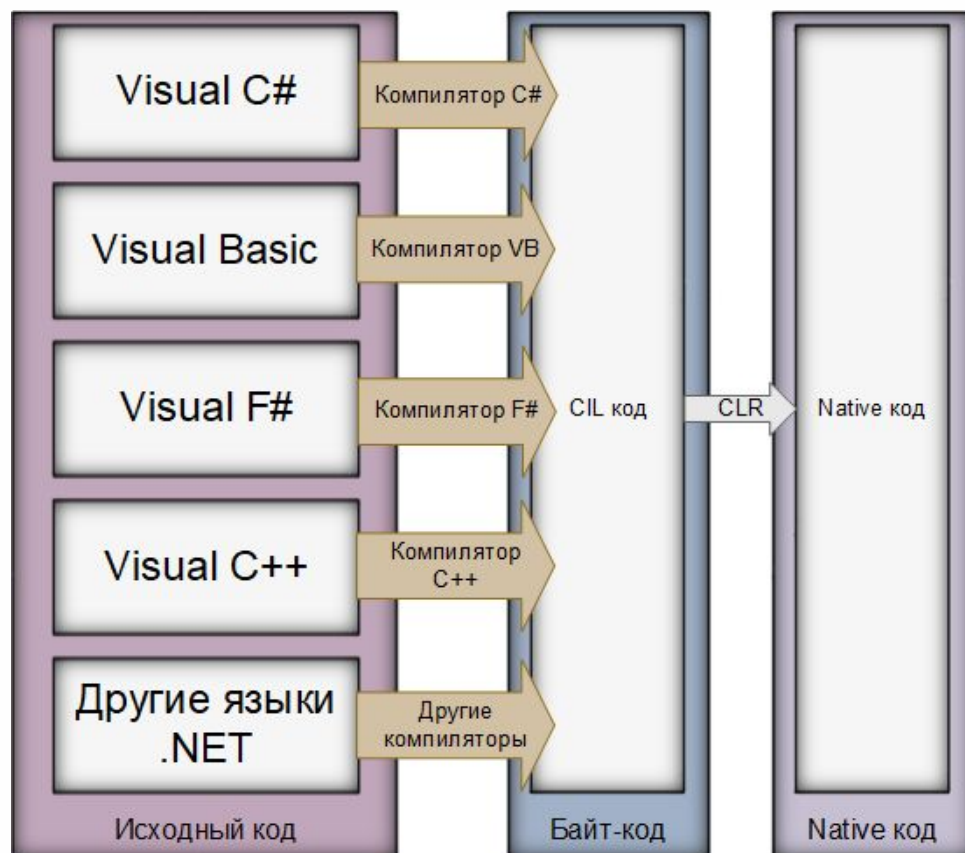
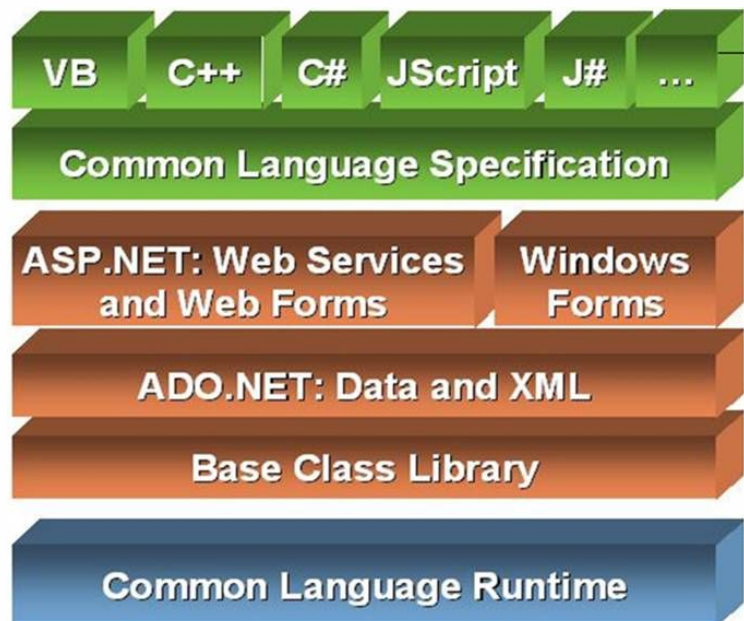
- Разработка ХП, функций и др объектов на С# (.NET)
- Напишите в чат степень вашего знакомства с С#

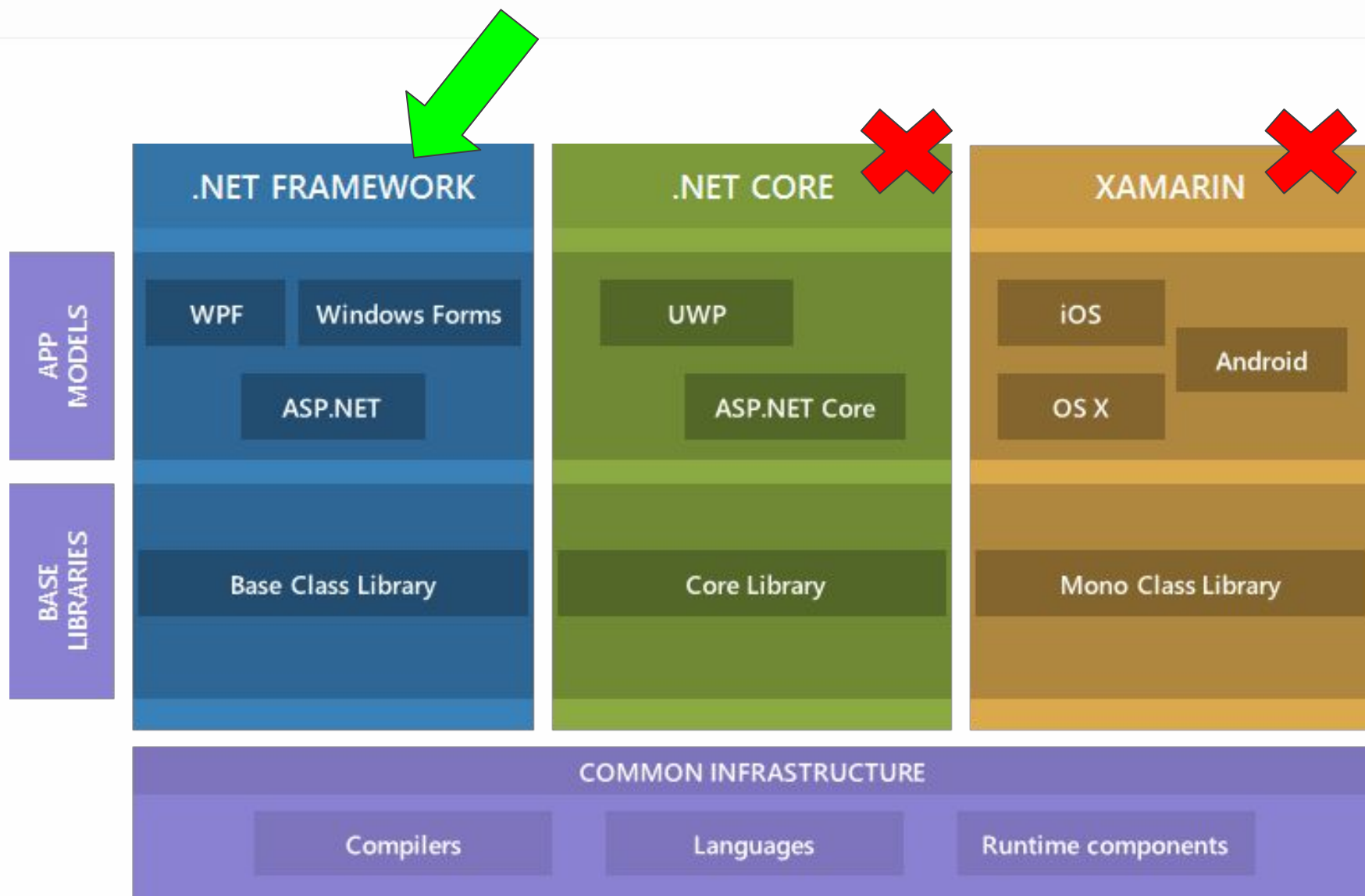
от 0 до 5

6 - писали CLR для SQL Server

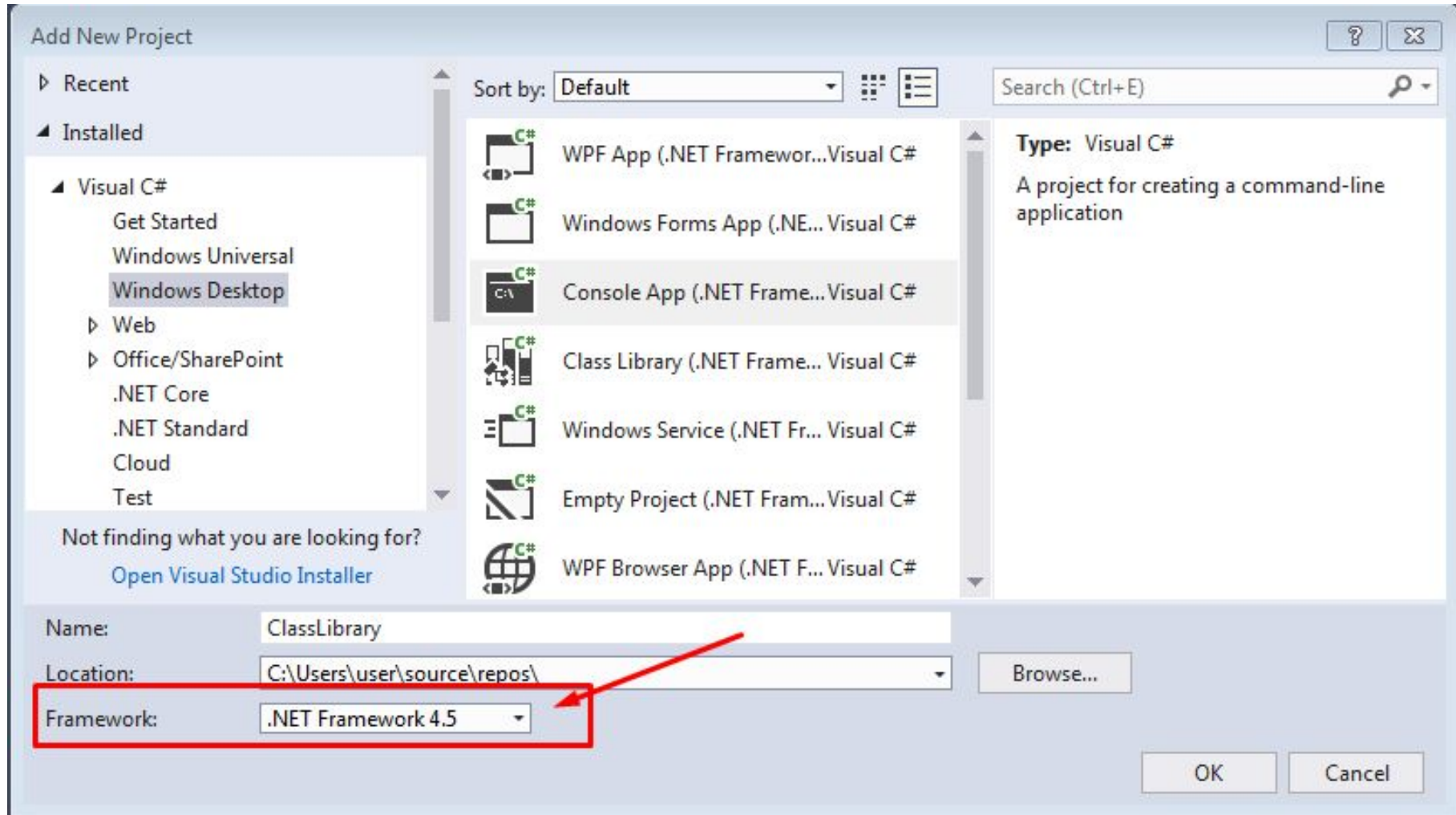


- Managed vs native code
- **Сборка, assembly** - exe, dll
- **CLR** - Common Language Runtime, Общезыковая среда исполнения (виртуальная машина)
- **SQLCLR** - реализация CLR в SQL





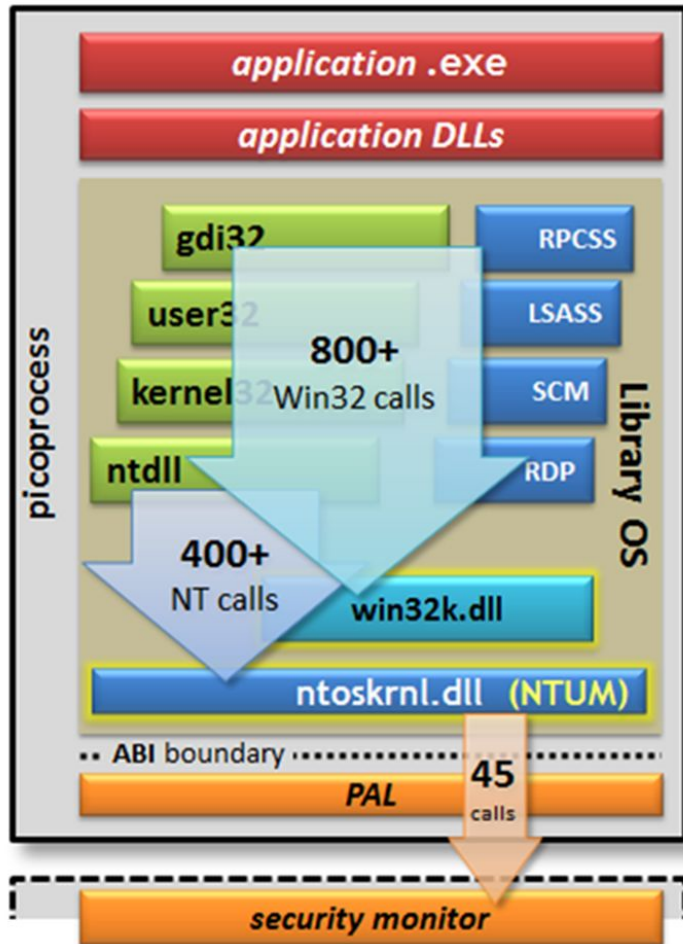
.NET 5 объединил .NET Framework и .NET Core



Версии .NET Framework и CLR

SQL Server —» Category (below)	2005	2008 / 2008 R2	2012 / 2014
CLR version	2.0		4.0
.NET Framework versions	2.0 / 3.0 / 3.5		4.0 / 4.5 / 4.5.x
.NET Framework version installed with SQL Server ¹	2.0	3.5 SP1	4.0

- [Stairway to SQLCLR Level 5: Development \(Using .NET within SQL Server\)](#)
- [Версии .NET Framework и ОС Windows](#)



Поддерживаются сборки CLR с набором разрешений SAFE

Library OS was capable of hosting other Windows components. Enough of the Win32 and NT layers were implemented to host **CLR**, MSXML, and other APIs that the SQL suite depends on. This meant that we could get more functionality to work without rewriting whole features.

<https://cloudblogs.microsoft.com/sqlserver/2016/12/16/sql-server-on-linux-how-introduction/>

<https://github.com/MicrosoftDocs/sql-docs/issues/1594>

Зачем использовать CLR в SQL Server?
Почему T-SQL не достаточно?

Ваши варианты (напишите в чат).



- Реализация доп. функционала (например, регулярные выражения)
- Сложная логика (математика, обработка текста)
- Доступ к ресурсам, файлам, удаленным службам (*)
- Производительность (*)
- ...

- Хранимые процедуры
- Функции
 - Скалярные
 - Табличные
- Триггеры
- Агрегаты
- Типы (hierarchyid, geometry, geography)

- SAFE

Доступ к данным внутри SQL Server. Нет доступа к внешним ресурсам.

- EXTERNAL_ACCESS

+ Доступ к файлам, сети, переменным окружения, реестру.

- UNSAFE

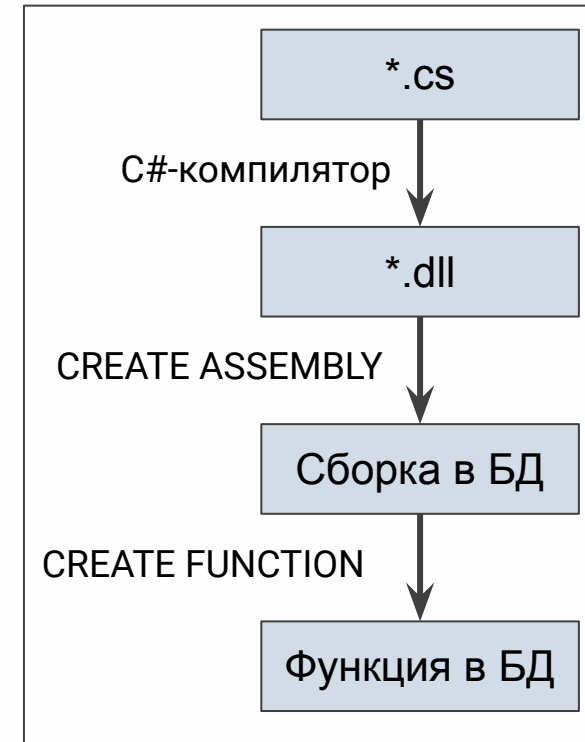
Неограниченный доступ внутри SQL Server и снаружи. Может также запускать неуправляемый код.

В Linux только SAFE

- Включение CLR

```
sp_configure 'show advanced options', 1  
sp_configure 'clr enabled', 1
```

```
sp_configure 'clr strict security', 0  
ALTER DATABASE SET TRUSTWORTHY ON
```



- `CREATE ASSEMBLY assemblyName FROM 'c:\file.dll'`
- `CREATE [PROCEDURE | FUNCTION | ...]
AS EXTERNAL NAME
AssemblyName.[ExampleNamespaceName.ClassName].MethodName`

ДЕМО

Ручная регистрация CLR



02

SQLCLR

New Project

Recent

Installed

- Visual C#
- Other Languages
 - Visual Basic
 - SQL Server**
 - Visual F#
- Other Project Types
- Online

.NET Framework 4.7.1 Sort by: Default Search (Ctrl+E)

SQL Server Database Project SQL Server

Type: SQL Server

Должны быть установлены SSDT [SQL Server Data Tools](#)

Add New Item - C# Demo

Installed

- SQL Server
 - Programmability
 - Security
 - Service Broker
 - Server Objects
 - SQL CLR
 - SQL CLR C#**
 - Storage
 - Tables and Views
 - User Scripts
 - Visual Studio Templates
- Online

Sort by: Default Search (Ctrl+E)

Assembly Information File	SQL Server	Type: SQL Server
Class	SQL Server	Transact-SQL script for a SQL CLR C# user defined type
Code File	SQL Server	
SQL CLR C# Aggregate	SQL Server	
SQL CLR C# Stored Procedure	SQL Server	
SQL CLR C# Trigger	SQL Server	
SQL CLR C# User Defined Function	SQL Server	
SQL CLR C# User Defined Type	SQL Server	

Name: SqlUserDefinedType1.cs

Add Cancel

- Как создать проект
- Скрипт
- Publish

- Name
Имя функции
- IsDeterministic
Детерминированная или нет.
- IsPrecise
Выполняет ли функция неточные вычисления, например, с плавающей точкой.
- DataAccess
Есть доступ к данным или нет
- Для доступа к данным используется ADO.NET
connection string = "context connection = true"

ДЕМО

Функции



Табличная функция

О U S

```
[SqlFunction(  
    TableDefinition = "item nvarchar(100), num int",  
    FillRowMethodName = "MakeRow")]
```

0 references

```
public static IEnumerable Split(string str, string separator)
```

```
{
```

```
    var items = str.Split(separator.ToCharArray());
```

```
    // В result будет итоговое значение
```

```
    var result = new List<object[]>();
```

object[]
это одна строка

```
    for (int i = 0; i < items.Length; i++)
```

```
    {
```

```
        var row = new object[2];
```

```
        row[0] = items[i];
```

```
        row[1] = i + 1;
```

```
        result.Add(row);
```

```
    }
```

```
    return result;
```

```
}
```

```
select *  
from dbo.Split('a,ab,abc', ',')
```

	item	num
1	a	1
2	ab	2
3	abc	3

Основная логика.
Формируем
"таблицу" result

Табличная функция

```
[SqlFunction(  
    TableDefinition = "item nvarchar(100), num int",  
    FillRowMethodName = "MakeRow")]
```

0 references

```
public static IEnumerable Split(string str, string separator)
```

```
{
```

```
    var items = str.Split(separator.ToCharArray());
```

```
    // В result будет итоговое значение
```

```
    var result = new List<object[]>();
```

```
    for (int i = 0; i < items.Length; i++)
```

```
    {
```

```
        var row = new object[2];
```

```
        row[0] = items[i];
```

```
        row[1] = i + 1;
```

```
        result.Add(row);
```

```
    }
```

```
    return result;
```

```
}
```

object[]
это одна строка

Основная логика.
Формируем
"таблицу" result

```
select *  
from dbo.Split('a,ab,abc', ',')
```

	item	num
1	a	1
2	ab	2
3	abc	3

```
public static void MakeRow(  
    object obj, out string item, out int num)  
{  
    var row = obj as object[];  
    item = (string)row[0];  
    num = (int)row[1];  
}
```

SQL	Boolean	Byte	Int16	Int32	Int64	SByte	UInt16	UInt32	UInt64	Decimal	Single	Double	Enum	Char	String	Char[]	XDocument	XElement	custom types
BIT	✓	→	→	→	→	→	→	→	→	→	→	→							
TINYINT	←	✓	→	→	→	↔	↔	↔	↔	→	→	→	↔	↔					
SMALLINT	←	←	✓	→	→	↔	↔	↔	↔	→	→	→	↔	↔					
INT	←	←	←	✓	→	↔	↔	↔	↔	→	→	→	↔	↔					
BIGINT	←	←	←	←	✓	↔	↔	↔	↔	→	→	→	↔	↔					
SMALLMONEY	←	←	←	→	→	↔	↔	↔	↔	→	→	→							
MONEY	←	←	←	←	→	↔	↔	↔	↔	→	→	→							
DECIMAL(20)/NUMERIC(20)	←	←	←	←	←	↔	↔	↔	↔	→	→	→							
DECIMAL(28)/NUMERIC(28)	←	←	←	←	←	↔	↔	↔	↔	→	→	→							
DECIMAL(29)/NUMERIC(29)	←	←	←	←	←	↔	↔	↔	↔	→	→	→							
DECIMAL(38)/NUMERIC(38)	←	←	←	←	←	↔	↔	↔	↔	→	→	→							
REAL/FLOAT(24)	←	←	←	←	←	↔	↔	↔	↔	→	✓	→							
FLOAT/FLOAT(53)	←	←	←	←	←	↔	↔	↔	↔	→	→	✓							
CHAR(1)													↔	✓	→	→			→
NCHAR(1)													↔	✓	→	→			→
CHAR(50)														✓	→	→			
NCHAR(50)														✓	→	→			
VARCHAR(50)																			
NVARCHAR(50)																			
VARCHAR(MAX)																			
NVARCHAR(MAX)																			
TEXT																			
NTEXT																			
XML																			

Legend

- ✓ Data should roundtrip without overflow/data loss exceptions
- ← Possible overflow/data loss exception getting data FROM the database
- Possible overflow/data loss exception saving data TO the database
- ↔ Possible overflow/data loss exception FROM or TO the database
- Mapping not supported / run-time behavior not defined

Note: silent precision loss may occur, even for mappings without overflow

На слайде не вся таблица. Полностью - [SQL-CLR Type Mapping](#)

ДЕМО

Типы, Null

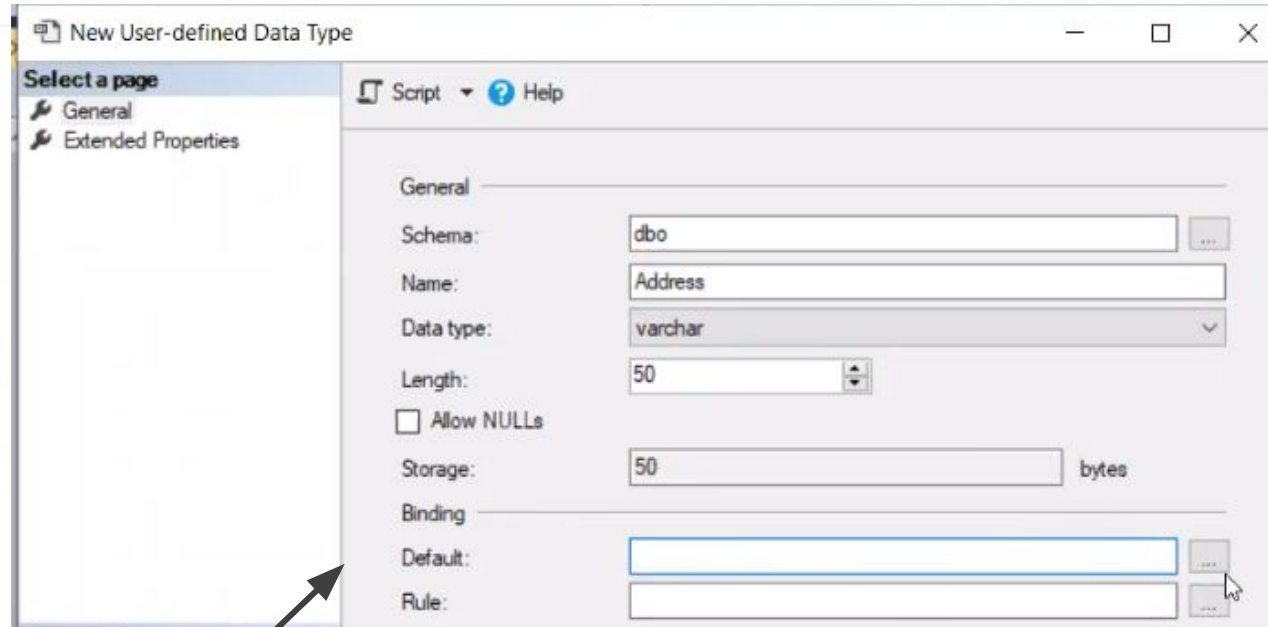
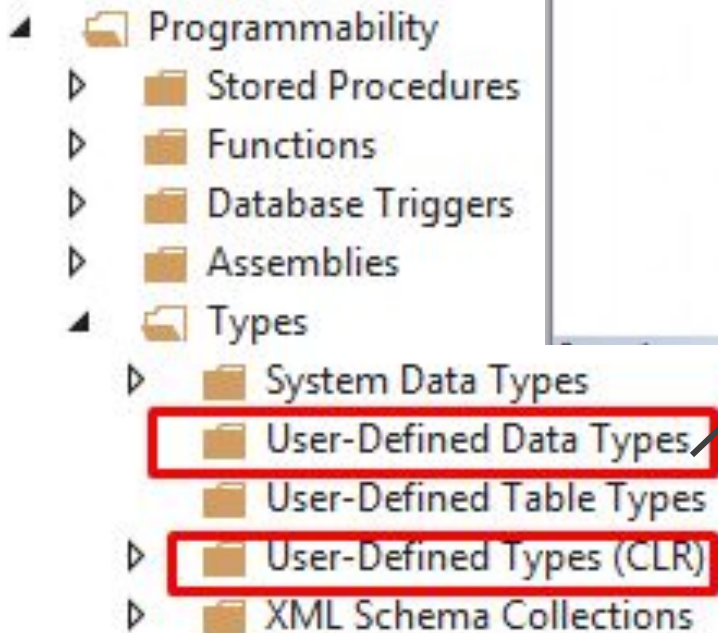


- Возвращают void или статус
- SqlContext, SqlPipe
 - Messages
 - ResultSets
 - Errors
- Генерирование ResultSet
 - Генерация результата без SQL
- Триггеры
 - DDL, DDL
 - Почти то же самое, что и ХП (нет параметров)
 - SqlTriggerContext

ДЕМО

Хранимые процедуры





Это не то, о чем мы будем говорить.
Это просто псевдонимы для существующих
типов. Новую функциональность не сделать.

Мы говорим об этом.

ДЕМО

Пользовательские типы



```
CREATE TYPE [dbo].[PhoneNumber]  
EXTERNAL NAME [DatabaseProjectDemo].[PhoneNumber]
```


- Скалярное значение из набора данных
 - AVG, MIN, MAX, COUNT и другие
- На TSQL написать нельзя
- Обычно быстрее, чем курсоры (**)
- Initialize -> Accumulate -> Merge -> Terminate

ORDER BY не поддерживается в OVER для CLR-агрегатов

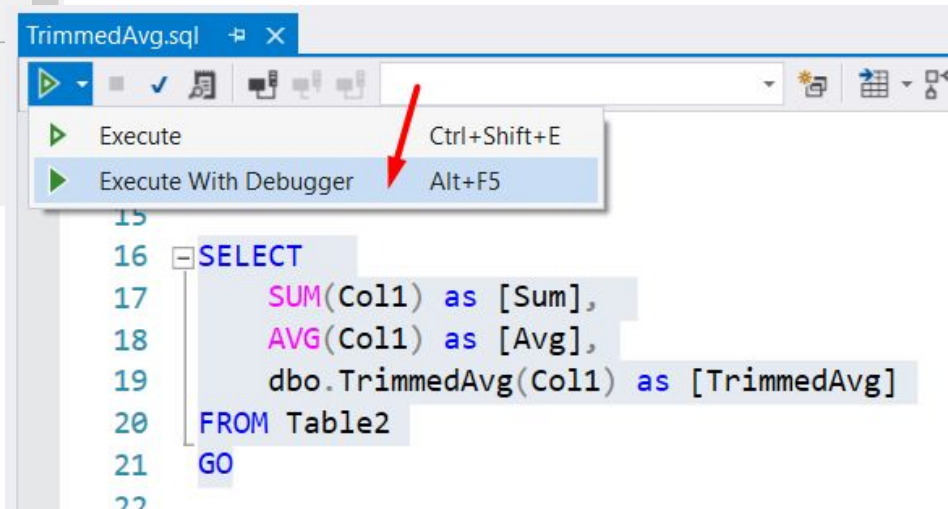
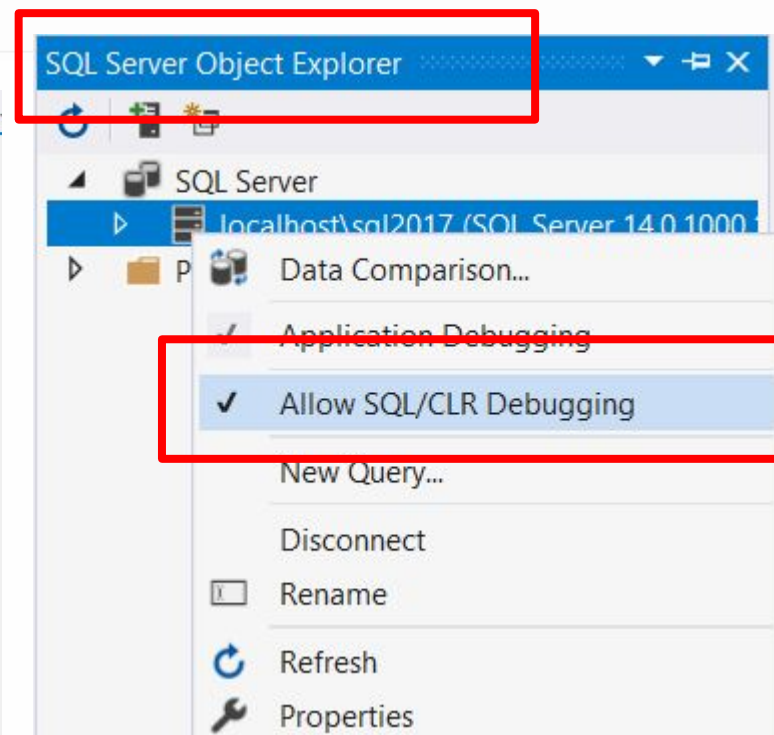
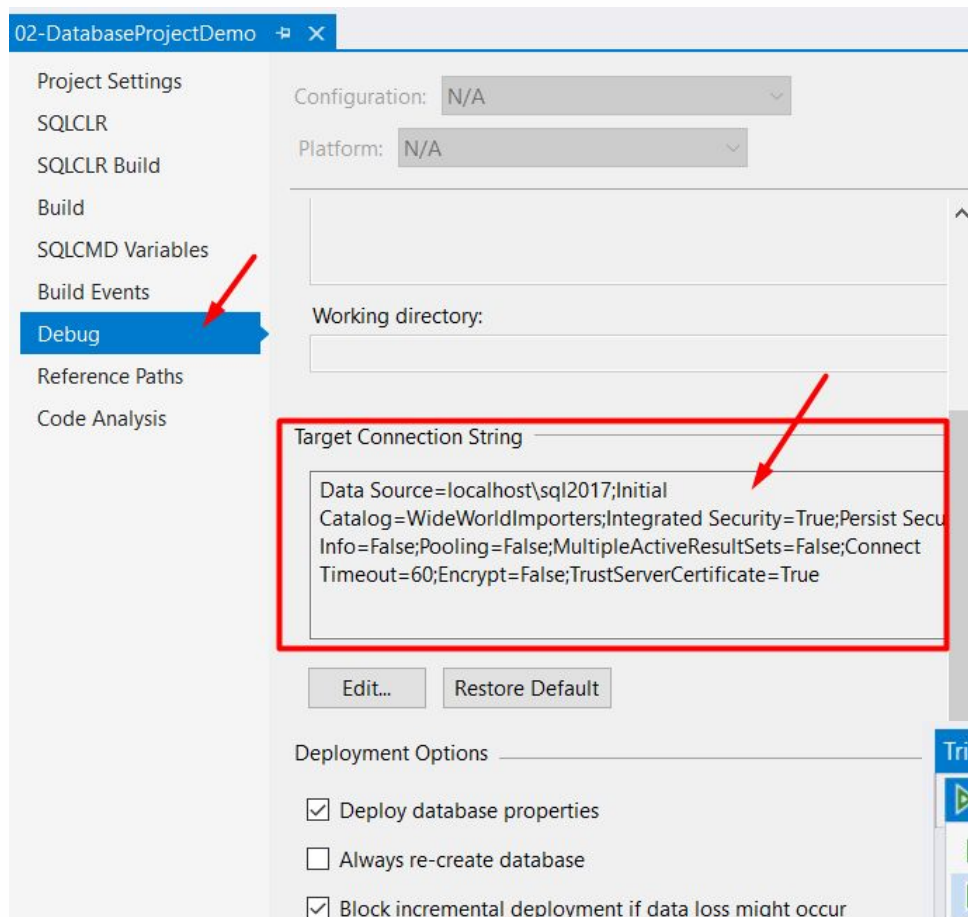
- <https://feedback.azure.com/forums/908035-sql-server/suggestions/32910790-add-support-for-over-order-by-for-clr-aggregate>
- <https://stackoverflow.com/questions/24165879/sql-order-by-within-over-clause-incompatible-with-clr-aggregation>

ДЕМО

Агрегаты



```
CREATE AGGREGATE [dbo].[TrimmedAvg] (@value [float])  
RETURNS[float]  
EXTERNAL NAME [ClrDemo].[TrimmedAvg]
```



Visual Studio запустить "Run as administrator..."

В SSMS отладку убрали

03

Language Extensions (расширения языка)

- Расширения языка — это функция SQL Server, используемая для выполнения внешнего кода
- mssql-server-extensibility, mssql-server-extensibility-java

```
CREATE EXTERNAL LANGUAGE Java
FROM (CONTENT = N'/opt/mssql-extensibility/lib/java-lang-extension.tar.gz',
      FILE_NAME = 'javaextension.so',
      ENVIRONMENT_VARIABLES = N'{"JRE_HOME":"/opt/mssql/lib/zulu-jre-11"}')
```

```
CREATE EXTERNAL LIBRARY regex
FROM (CONTENT = '<path>/regex.jar')
WITH (LANGUAGE = 'Java');
GO
```

<https://docs.microsoft.com/en-us/sql/language-extensions/language-extensions-overview>

<https://docs.microsoft.com/en-us/sql/language-extensions/install/windows-java>

```
CREATE OR ALTER PROCEDURE [dbo].[java_regex] @expr nvarchar(200), @query nvarchar(400)
AS
BEGIN
--Call the Java program by giving the package.className in @script
--The method invoked in the Java code is always the "execute" method
EXEC sp_execute_external_script
    @language = N'Java'
    , @script = N'pkg.RegexSample'
    , @input_data_1 = @query
    , @params = N'@regexExpr nvarchar(200)'
    , @regexExpr = @expr
with result sets ((ID int, text nvarchar(100)));
END
GO

--Now execute the above stored procedure and provide the regular expression and an input query
EXECUTE [dbo].[java_regex] N'[Jj]ava', N'SELECT id, text FROM testdata'
GO
```

```
DROP PROC IF EXISTS generate_iris_model;
GO
CREATE PROC generate_iris_model
AS
BEGIN
    EXEC sp_execute_external_script
        @language = N'R'
        , @script = N'
            library(e1071);
            irismodel <-naiveBayes(iris_data[,1:4], iris_data[,5]);
            trained_model <- data.frame(payload = as.raw(serialize(irismodel, connection=NULL)));
        '
        , @input_data_1 = N'select "Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species"
        , @input_data_1_name = N'iris_data'
        , @output_data_1_name = N'trained_model'
        WITH RESULT SETS ((model varbinary(max)));
END;
GO
```



```
CREATE PROCEDURE [dbo].[py_generate_customer_scores]
AS
BEGIN

-- Input query to generate the customer data
DECLARE @input_query NVARCHAR(MAX) = N'SELECT customer, orders, items, cost FROM dbo.Sales.Orders'

EXEC sp_execute_external_script @language = N'Python', @script = N'
import pandas as pd
from sklearn.cluster import KMeans

# Get data from input query
customer_data = my_input_data

# Define the model
n_clusters = 4
est = KMeans(n_clusters=n_clusters, random_state=111).fit(customer_data[["orders","items","cost"]])
clusters = est.labels_
customer_data["cluster"] = clusters

OutputDataSet = customer_data
'

, @input_data_1 = @input_query
, @input_data_1_name = N'my_input_data'
WITH RESULT SETS (( "CustomerID" int, "Orders" float, "Items" float, "Cost" float, "ClusterResult" float ))
END;
GO
```


04

Домашнее задание

Варианты ДЗ (сделать один любой):

1. Взять готовую dll, подключить ее и продемонстрировать использование. Например, <https://sqlsharp.com>
2. Взять готовые исходники из какой-нибудь статьи, скомпилировать, подключить dll, продемонстрировать использование.
Например,
 - <https://www.sqlservercentral.com/articles/xlsexport-a-clr-procedure-to-export-proc-results-to-excel>
 - <https://www.mssqltips.com/sqlservertip/1344/clr-string-sort-function-in-sql-server/>
 - <https://www.sqlservercentral.com/articles/practical-usage-of-sqlclr-building-querytohtml-function>
 - http://tool-bar.ru/CLR_SQL/index.htm
 - или любые другие
3. Написать полностью свое:
 - Тип: JSON с валидацией, IP / MAC - адреса, ...
 - Функция: работа с JSON, архивация (сжатие), шифрование ...
 - Агрегат: аналог STRING_AGG, ...
 - или любой ваш вариант

- .NET, CLR / SQL CLR
- Функции, процедуры в CLR
- Типы, агрегаты CLR

- Что запомнилось?
- Узнали ли что-то новое?



**Заполните,
пожалуйста,
опрос в ЛК о занятии**

