Сборка (assembly) — это логическая единица, содержащая скомпилированный код для .NET Framework, т.е. это полностью самодостаточный и скорее логический, нежели физический элемент.

**SQL CLR (Common Language Runtime)**

**CLR** – общеязыковая исполняющая среда. Позволяет выполнять управляемый код .NET внутри SQL Server.

**Поддерживаемые языки:** C# и VB.NET.

Используется для реализации сложной логики, недоступной в T-SQL.

**Основные концепции SQL CLR**

1. Управляемая среда: Автоматическое управление памятью (Garbage Collection). Управление потоками. Обеспечение безопасности кода (Code Access Security – CAS).
2. Домены приложений (AppDomains): Изолированные среды выполнения кода внутри SQL Server. Позволяют загружать и выгружать сборки без перезапуска сервера.
3. Интеграция .NET и SQL Server: Возможность использовать мощные алгоритмы из .NET внутри SQL Server. Расширение возможностей SQL Server с использованием внешних API.

**Применение SQL CLR**

1. Сложная логика, которую трудно реализовать на T-SQL.
2. Решение математических задач, таких как сложные вычисления, статистика и обработка данных.
3. Подключение к удаленным службам (например, веб-сервисам REST/SOAP).
4. Работа с удаленными хранилищами данных (например, файлы в облачных сервисах).

Управление внешними файлами (чтение/запись файлов на сервере).

**SQL CLR Объекты**

1. Процедуры (CLR Stored Procedures): Аналог T-SQL-хранимых процедур, но с возможностью выполнения сложной логики на C# или VB.NET.
2. Функции

- Скалярные функции (Scalar Functions) – возвращают одно значение.

- Табличные функции (Table-Valued Functions) – возвращают набор строк.

1. Типы (User-Defined Types, UDTs): Позволяют создавать собственные типы данных в SQL Server.
2. Триггеры: Позволяют реализовывать триггеры с расширенной логикой на C#.

**SQL CLR: Пользовательские агрегатные функции**

**SqlUserDefinedAggregate** Позволяет создавать кастомные агрегатные функции (аналог SUM, AVG, COUNT):

Init – инициализация значений агрегата.

Accumulate – обработка каждой строки в наборе.

Merge – объединение нескольких агрегированных значений.

Terminate – завершение вычислений и возврат результата.

Пример: создание агрегатной функции для конкатенации строк.

**SQL CLR: Определяемые пользователем типы (UDT)**

**SqlUserDefinedType**: Позволяет создавать собственные типы данных в SQL Server.

INullable – интерфейс для работы с NULL.

IsNull – проверяет, является ли значение NULL.

Null – возвращает новый экземпляр типа с NULL-значением.

ToString – строковое представление объекта.

Parse – преобразует строку в экземпляр типа.

Пример: создание типа для хранения IP-адресов в SQL Server.

**Безопасность SQL CLR**

**Code Access Security (CAS)**: Управляет уровнем доступа к ресурсам системы:

SAFE – доступ только к внутренним ресурсам SQL Server.

EXTERNAL\_ACCESS – разрешен доступ к файлам, сети, службам, реестру.

UNSAFE – полный доступ к системным ресурсам.

⚠ CLR strict security включен по умолчанию (с SQL Server 2017)

Все сборки по умолчанию считаются UNSAFE.

Требуется вручную разрешить выполнение кода CLR (опция sp\_configure 'clr strict security', 0).

**Дополнительные возможности SQL CLR**

- Использование регулярных выражений в SQL Server.

- Реализация парсинга JSON/XML при помощи .NET-классов.

- Работа с многопоточностью и асинхронными операциями.

- Интеграция с машинным обучением и AI-библиотеками.