



XML в MS SQL Server 2000 и технологиях доступа к данным

Алексей Шуленин
Microsoft, Moscow



Отдел стратегических платформ .NET
системный инженер



Зачем XML нужен в СУБД?

- **Передача запросов и результатов через Интернет**
 - XML прозрачно проходит сквозь сетевые экраны
 - В отличие, скажем, от СОМ-объектов
- **Взаимодействие в гетерогенных средах**
 - XML сейчас понимают все независимо от платформы и ОС
 - Интеграция не только с другими СУБД, но и с серверами В2В, электронной коммерции и пр.
- **Не требует установки дополнительного ПО на клиента**
 - Клиент может быть сколь угодно тонким
 - Опять же потому, что средства его поддержки присутствуют изначально

Немного истории

- **ADO 2.1 (1999 г.)**
 - Сохранить рез-ты запроса (ADODB.Recordset) в XML-формате
 - До этого в ADO 2.0 поддерживался только ADTG
 - Частный бинарный формат, использовавшийся при передаче recordset'а в удаленном доступе с помощью RDS
 - На выходе – нормальный XML со своей схемой
 - Можно делать все, что и с обычным XML-документом
 - Открыть при пом. DOM, запросить при пом. XPath, преобразовать при пом. XSL и т.д.

[Посмотреть код](#)

Немного истории

- **ADO 2.5**
 - Промежуточное сохранение в файл – лишние затраты
 - Появилась возможность напрямую в программе передавать полученный поток (stream) XML любому объекту, подд. IStream

[Посмотреть код](#)

- Возм. и обр.ситуация - чтение XML из потока и его запись в recordset
 - За счет введения нового сервисного провайдера
 - Microsoft OLE DB Persistence Provider ("Provider=MSPersist")

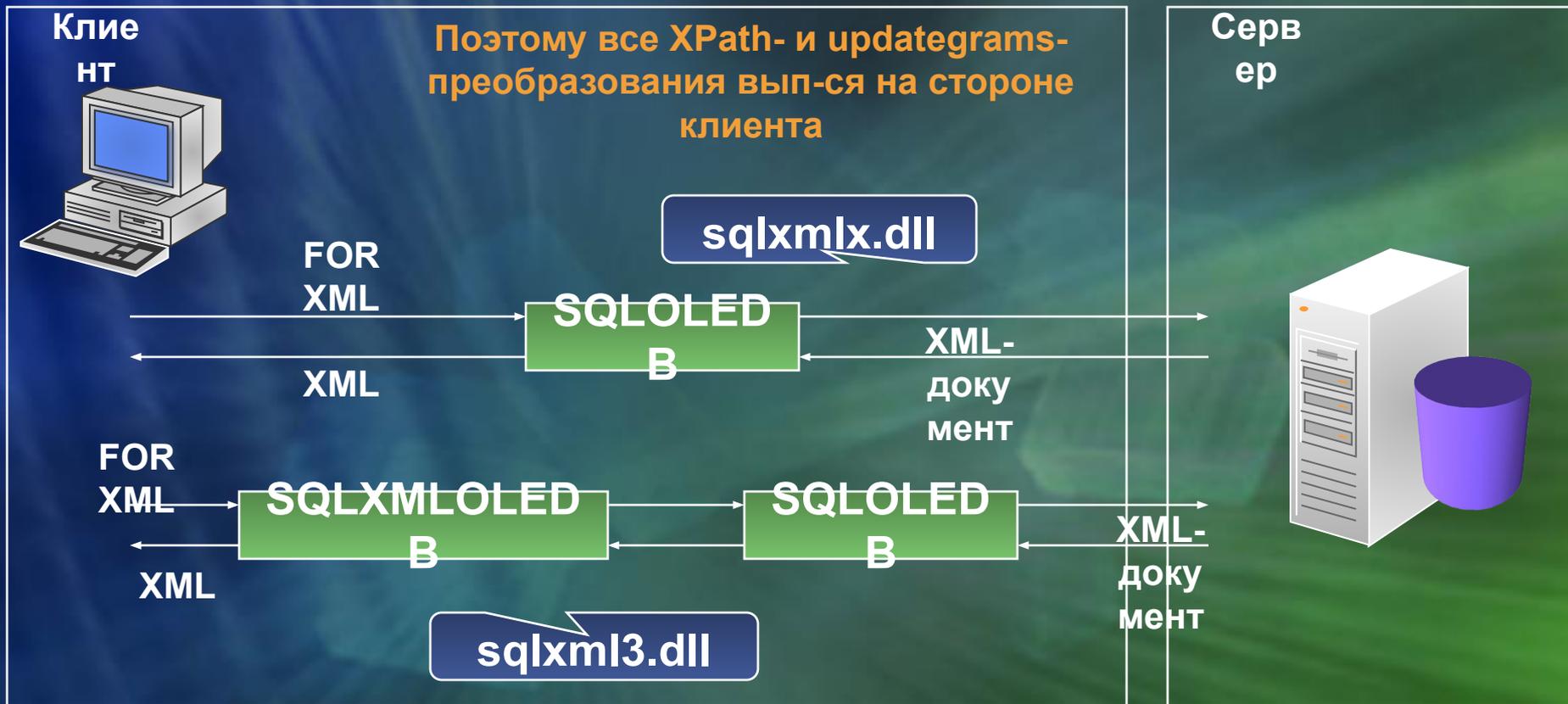
Появление XML в SQL Server

- Рассмотренные ранее подходы имеют две особенности
 - Преобразование в XML происходит на клиенте
 - XML получается по определенной схеме
 - Требуется дополнительное XSL-преобразование
- В SQL Server 2000 появилась возможность получения XML на стороне сервера
 - `SELECT ... FOR XML ...`

[Посмотреть код](#)

SELECT ... FOR XML

[Посмотреть код](#)



- Документ получается на сервере, но ввиду отсутствия встроенного типа XML он может только отдать его клиенту
 - Его нельзя использовать как результат подзапросов, хранимых функций и всего остального, что предполагает дальнейшую обработку на SQL Server

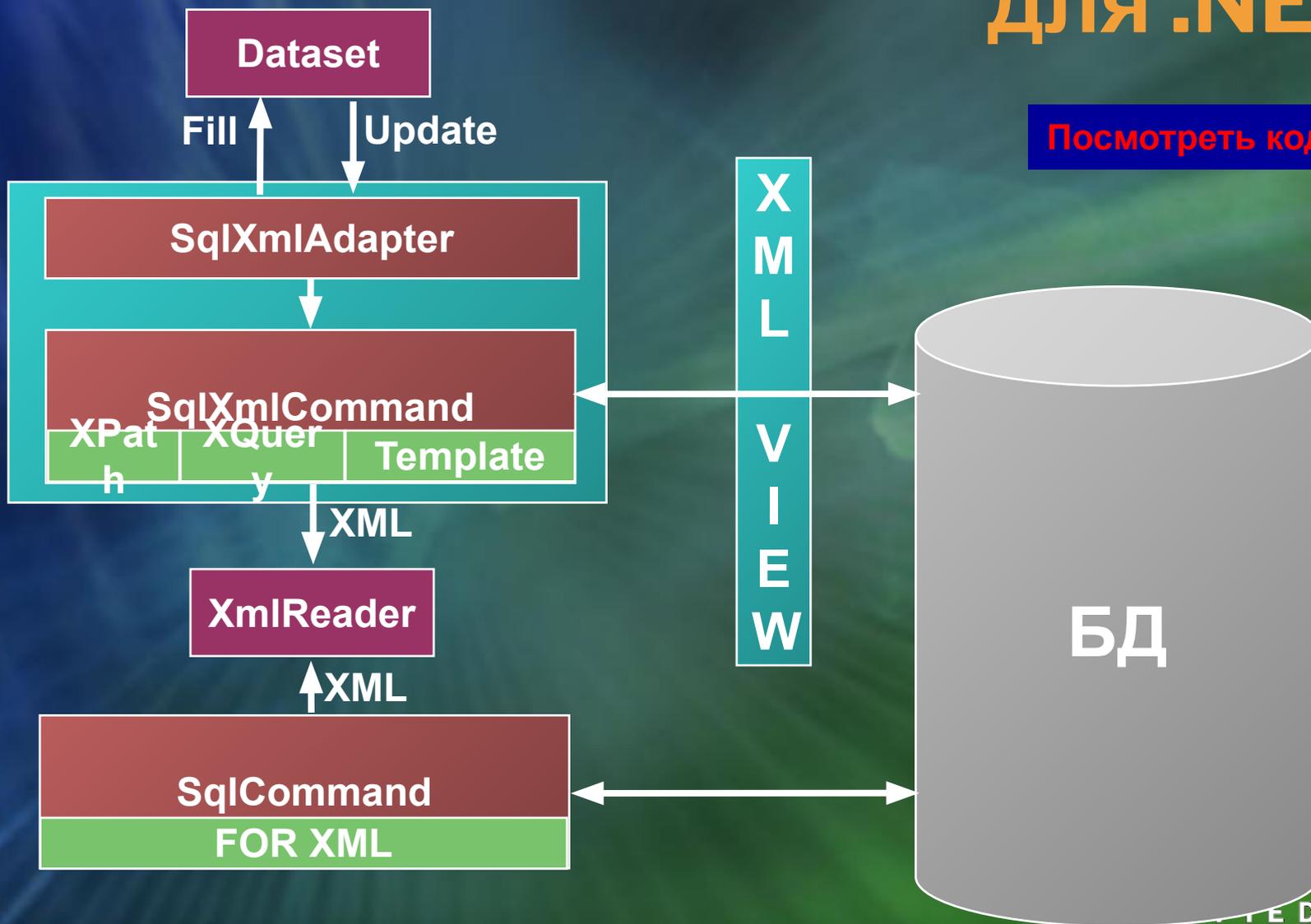
Поддержка XML в SQL Server

- Впервые появилась в составе SQL Server 2000
 - Август 2000 г.
- Дополняется и расширяется SQLXML веб-релизами
 - Выкладываются на сайт SQL Server
 - Доступны для бесплатного скачивания и установки зарегистрированными пользователями
- Текущий на данный момент – SQLXML 3.0
 - В составе Microsoft SQL Server 2000 Web Services Toolkit
 - <http://msdn.microsoft.com/downloads/sample.asp?url=/MSDN-FILES/027/001/872/msdncompositedoc.xml&frame=true>



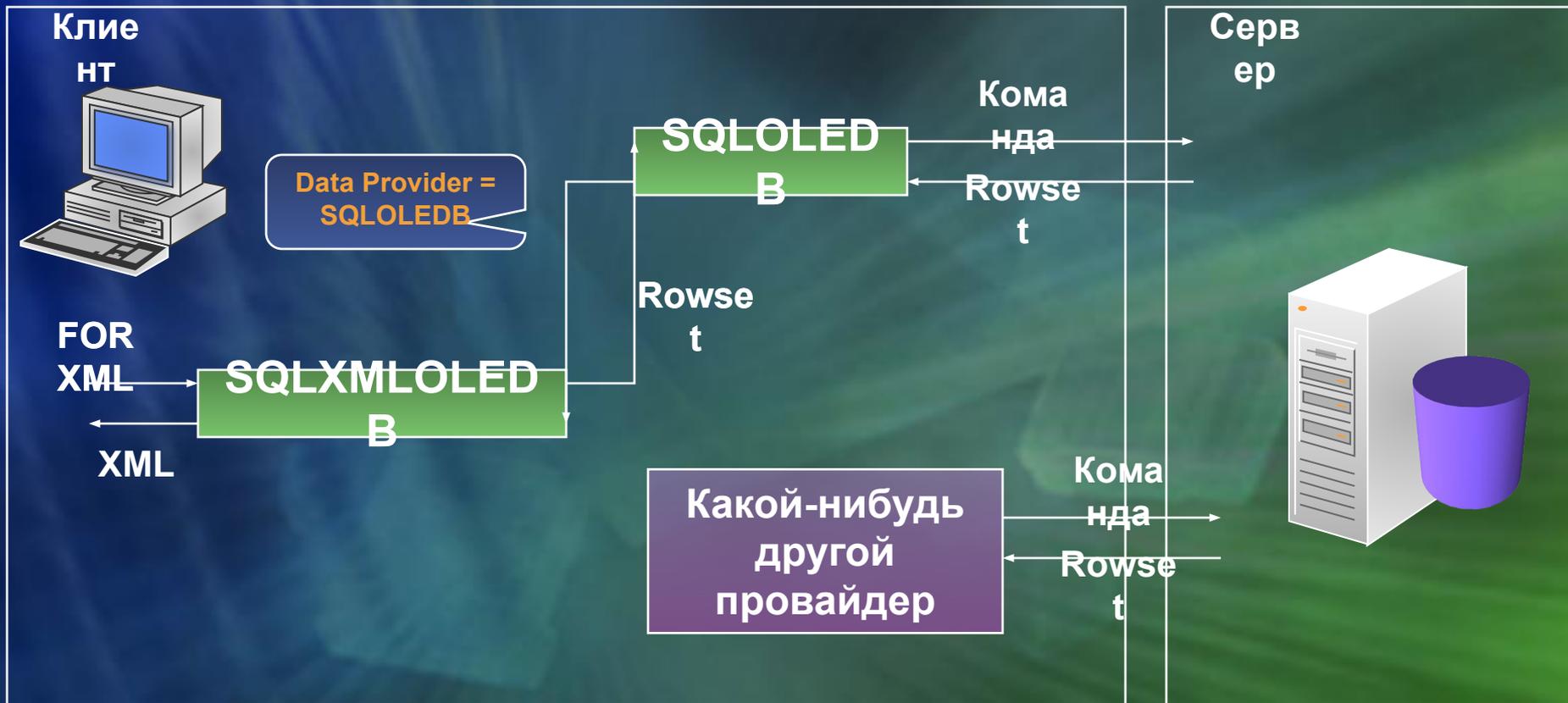
SQLXML Managed Classes для .NET

[Посмотреть код](#)



FOR XML на клиенте

[Посмотреть код](#)



- В перспективе SQLXMLOLEDB можно использовать с любыми источниками данных
 - Provider= SQLXMLOLEDB;DataProvider=...
- Однако в настоящее время в качестве провайдера данных поддерживается только SQL Server

Другие способы получения XML на клиенте. ADO.Net

- Типовой сценарий
 - Получить внутри DataSet таблицы как результаты запросов к источнику данных
 - Возможно, к разным
 - Связать их между собой на основе объектов DataRelation
 - Создать XML-представление DataSet'а при помощи XmlDocument

Microsoft®

С [Посмотреть код](#)

Галерея

Класс XmlDocument

- Производный от DOMовского XmlDocument
 - Позволяет выполнять над DataSet все стандартные XML-операции
 - XPath-запросы, XSL-преобразования
 - в част., данные источника можно модифицировать не только напрямую через DataSet
 - `ds.Tables[<Имя или номер таблицы в коллекции>].Rows[<Номер записи в коллекции>][<Имя или номер поля в коллекции>] = ...`
 - Но и через его XML-представление
 - Задавая координаты узлов в терминах XPath
 - Эти изменения отражаются в DataSet'е, над которым построен данный XmlDocument
 - А далее транслируются в источник
 - Если у DataAdapter'ов таблиц определены UpdateCommand'ы

Microsoft®

С [Посмотреть код](#)

Галерея

Обратное тоже верно

- В DataSet можно загрузить XML-документ, который затем читать и модифицировать реляционными операциями
- На основе XSD-схемы ADO.Net создает DataSet эквивалентной реляционной структуры
 - Можно обращаться и модифицировать XML-документ, как если бы он был совокупностью связанных таблиц.

Microsoft

С [Посмотреть код](#)

Галерея

Прямые XPath-запросы к объектам SQL Server

- Рассмотренные возможности ADO.Net по своей сути результат эволюции простого сохранения `ADODB.Recordset` в XML
- Вернемся к серверной стороне вопроса
- SQL Server понимает и обрабатывает не только SQL, но и XPath
- На сервере XPath превращается в `SELECT ... FOR XML`
 - Можно посмотреть в Profiler
- Ограничения
 - Пользуется схемой по умолчанию: строки – элементы, столбцы – атрибуты
 - Результат возвращается в виде строкового стрима в силу уже упоминавшегося отс-я типа XML

Microsoft

[Смотреть код](#)

Галерея

Аннотированные схемы

- Позволяют снять оба ограничения
 - Задают XSD-описание реляционной информации
 - С пом. аннотированных схем XPath-запросы к SQL Server возвращают нормальный XML
- Аннотации привязывают реляционку к схеме
 - `<sql:relation>`
 - Сопоставляет элемент таблице БД
 - `<sql:field>`
 - Указывает на поле таблицы
 - `<sql:key-fields>`
 - Задает набор полей, однозначно определяющих запись
 - `<sql:relationship>`
 - Определяет связь «первичный/внешний ключ» между таблицами в БД
 - ...
- Определены в `urn:schemas-microsoft-com:mapping-schema`

XSD-схема

Аннотированная схема

Еще один пример

XPath-запросы через аннотированные схемы

- Таким образом, аннотированные схемы – это инструмент для адресации XPath-запросов к SQL Server
- Аннотации позволяют задать соответствия между реляционной и XML-схемами
- Через призму аннотированной схемы реляционная информация видится XPath-запросу, как обычный XML-документ

Microsoft

[Посмотреть код](#)

Галерея

Шаблоны

Посмотреть живой шаблон

Посмотреть пример вызова

- Аналог хранимых процедур SQL Server с той лишь разницей, что в них могут сочетаться SQL- и XPath-запросы

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql"
      sql:xsl='XSL FileName' >
  <sql:header>
    <sql:param>..</sql:param>
    <sql:param>..</sql:param>...n
  </sql:header>
  <sql:query>
    sql statement(s)
  </sql:query>
  <sql:xpath-query mapping-schema="SchemaFileName.xml">
    XPath query
  </sql:xpath-query>
</ROOT>
```

Возможность преобразования XML-
результатов запроса SQL

SQL-запросы (включая
вызовы процедур)

Возможность передачи
параметров SQL-запросу и
установки их значений по
умолчанию

Наряду с SQL поддерживаются XPath-запросы

Шаблоны

- Аннотированные схемы, на которые ссылается `<sql:xpath-query mapping-schema=...>`, могут находиться в теле шаблона

[Посмотреть шаблон](#)

- Шаблоны – это не только хранимые процедуры
 - Шаблон можно приготовить в ходе выполнения и подать как Stream

[Посмотреть код](#)

- Частный случай – как строку

[Посмотреть код](#)

Кэширование шаблонов

- **Шаблоны остаются в памяти после первого выполнения**
 - Что улучшает производ-ть при интенсивной работе с шаблонами
- **Емкость кэша задается здесь**
 - HKLM\SOFTWARE\Microsoft\MSSQLServer\Client\SQLXML3\TemplateCacheSize
 - В штуках шаблонов
 - По умолч. 31
- **Отмена кэширования**
 - Virtual Directory Properties -> Advanced -> Disable Caching of template
- **Аналогично настраивается кэш для аннотированных схем**
 - HKLM\SOFTWARE\Microsoft\MSSQLServer\Client\SQLXML3\SchemaCacheSize

XML-обновление на сервере

- До сих пор мы только читали данные в виде XML
- Возможность обновления косвенно затрагивалась 2 раза
 - Когда разбирали квантово-волновой дуализм ADO.Net (XmlDataDocument) 😊
 - Это неинтересно, потому что происходит через клиента
 - И шаблоны, на с.д., если прописать INSERT/UPDATE/DELETE в <sql:query>
- **Шаблон** **Код**
 - Это тоже неинтересно, п.ч. по сути делается ср-ми SQL, а не XML

XML-запросы на обновление

Посмотреть живую Updategrams

- **UpdateGrams**
 - По своей природе – шаблон
 - Содержит updgr-аннотации, при помощи которых описываются модификации в БД

```
<ROOT xmlns:updg="urn:schemas-microsoft-com:xml-updategram">  
  <updg:sync [mapping-schema="XDRSchemaFile.xml"]>  
    <updg:before>  
      ...  
    </updg:before>  
    <updg:after>  
      ...  
    </updg:after>  
  </updg:sync>  
</ROOT>
```

Аннотированная схема. Определяет связь между SQL- и XML-схемами

Все удаления / вставки внутри одного sync происходят в одной транзакции

XML BulkLoad

- Не грузит весь XML целиком в память
- Считывает поэлементно, определяя область действия элемента по аннотированной схеме
- Как и в случае UpdateGrams, возможно наполнение связанных таблиц

Документ

Схема

Загрузка

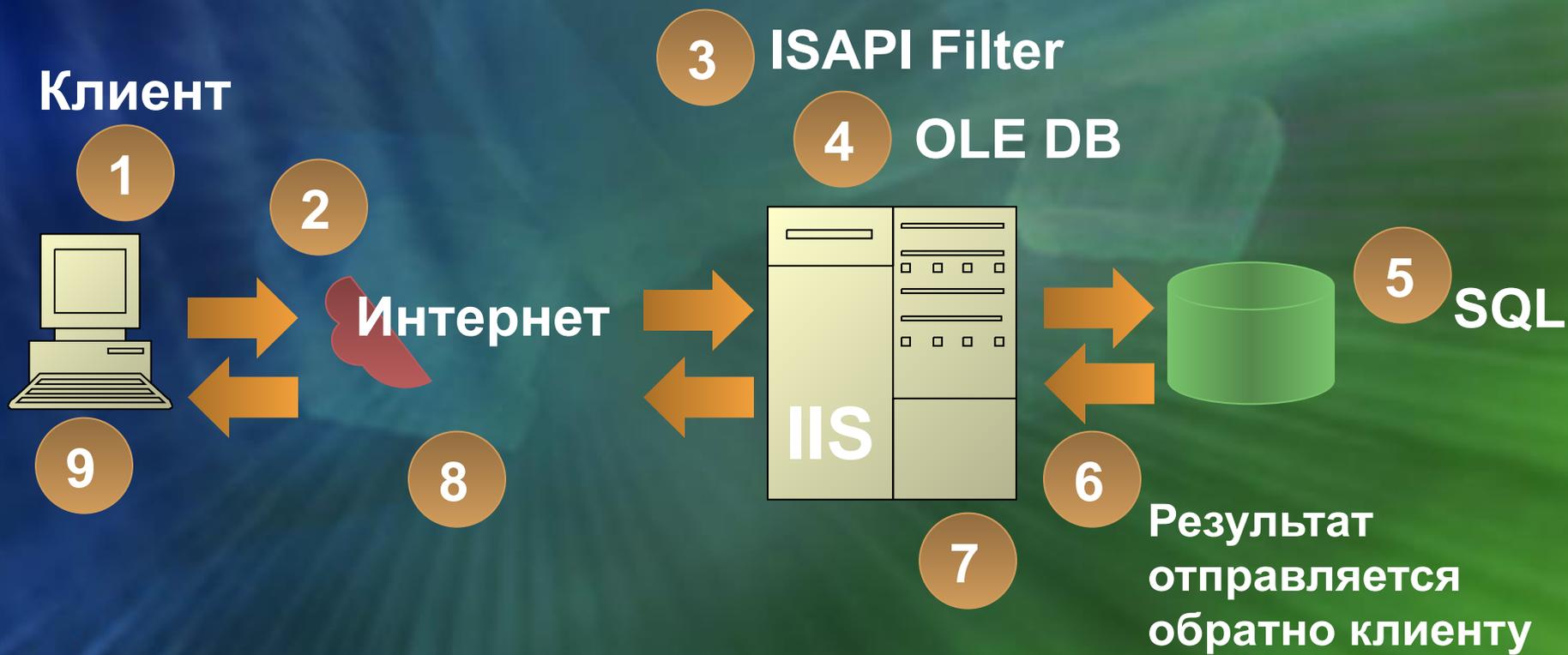
А как насчет доступа по HTTP?

- В самом начале я называл его одним из трех ключевых обоснований поддержки XML в СУБД
- Однако до сих пор в примерах разбирались по сути сценарии локальной сети
- Настало время исправить эту несправедливость

Организация взаимодействия с SQL Server по HTTP

- Напрямую по HTTP с SQL Server общаться нельзя
 - Во всяком случае, пока
- Между клиентом и SQL Server требуется IIS
- ISAPI-фильтр sqlis3.dll (входит в SQLXML 3.0) принимает вызовы GET, HEAD, POST, вытаскивает запрос к SQL Server и передает ему
 - Т.о. участок IIS – SQL Server относится к локальному взаимодействию и сводится к задаче, к-ю мы умеем решать
 - Чтобы разобраться участком клиент – IIS, нужно понять две вещи
 - Как упаковать запрос в GET и POST
 - И как его отправить фильтру

То же на картинке – схема публикации БД



Результат
отправляется
обратно клиенту

Microsoft
CERTIFIED

Partner

Как задействовать ISAPI-фильтр?

- Не мудрствуя лукаво, создать вирт.дир. и повесить его на нее
 - Все, кто к ней обращается, пропускаются через sqlis3.dll
- Создается программным путем или графически через Configure IIS Support в SQLXML 3.0
- Что указывается при ее создании
 - Connectivity
 - Имя SQL Server и базы, на к-ю через нее собираемся ходить
 - Аутентификация
 - Windows Integrated, базовая на SQL Server или анонимная (Anonymous д.б. привязан к учетной записи на SQL Server)
 - Права
 - На выполнение SELECT ... FOR XML или ad hoc-templates
 - XPath
 - Templates

Как упаковать запрос

- **Типы подкаталогов виртуальной директории**
 - **dboject**
 - Позволяет адресовать запросы напрямую к объектам БД
 - `http://localhost/Northwind/DBObj/Employees[@EmployeeID=1]/@LastName`
 - **schema**
 - Позволяет задавать аннотированную схему, преобразующую SQL в XML
 - `http://localhost/Northwind/Schema/AnnotatedSchemaExample.xsd/Customer[@CustomerID="ALFKI"]`
 - **template**
 - Позволяет оборачивать SQL/XPath-запросы в XML-шаблон
 - `http://localhost/Northwind/Temp1/GeneralTemplateExample.xml?Cntry=USA`
 - **soap**
 - Превращает SQL Server в Web-сервис
 - Позволяет из клиентского приложения обращаться к процедурам, функциям и шаблонам как к методам

GET

Или из браузера

POST

Или из формы

Что такое Web-сервис с точки зрения СОМ-разработчика?

- Это DCOM, где компоненты выходят за пределы локальной сети и могут быть распределены по Интернету
 - IDL / Type Library = WSDL
 - HKEY_CLASSES_ROOT / IUnknown = UDDI
 - RPC = SOAP
- Сейчас по HTTP гоняется в основном HTML
 - Он подходит для визуального восприятия, но малопригоден, если из URL нужно забрать информацию и передать в приложение

SQL Server как Web-сервис

- Виртуальные поддиректории типа SOAP позволяют обращаться к SQL Server как к Web-сервису
 - Хранимые процедуры / функции и шаблоны видны как методы Web-сервиса
 - wsdl-файл генерируется автоматически
 - Результат возвращается как единственный DataSet, массив объектов DataSet, либо массив XmlElement

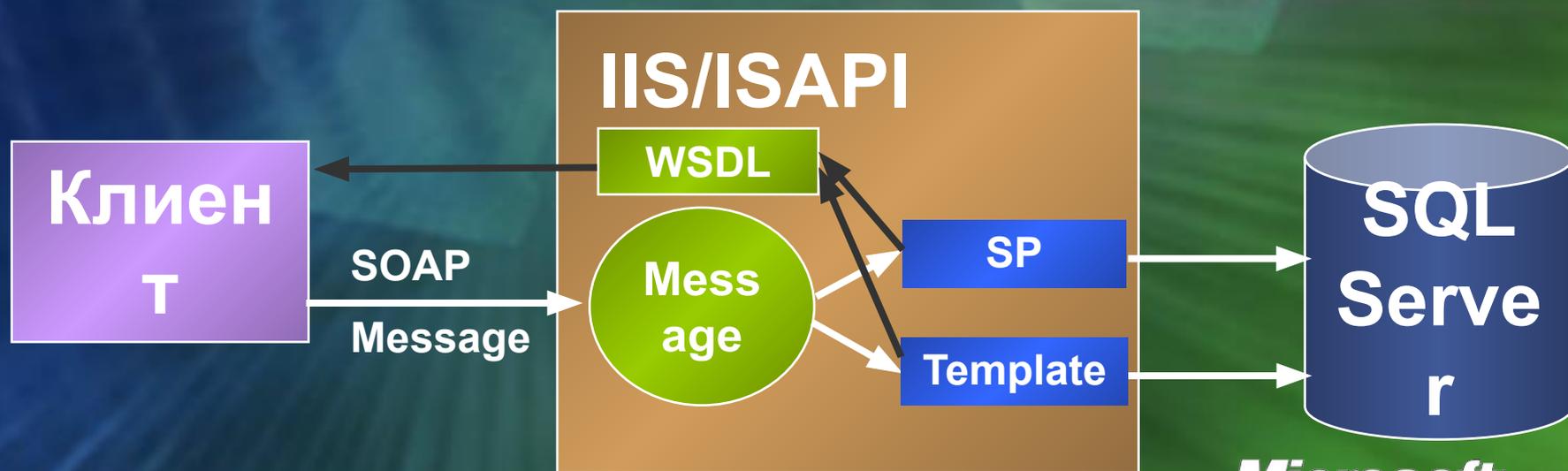
Microsoft®

С Е

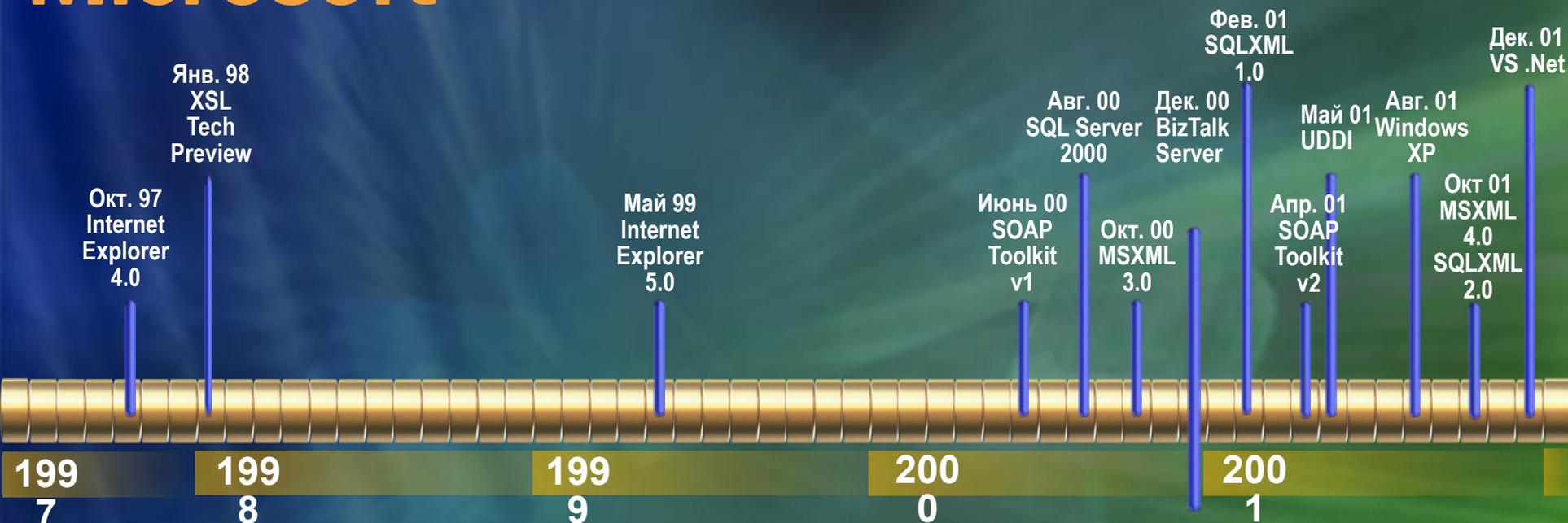
Пример

SOAP и Web Services

- Генерируется WSDL-файл, описывающий «засвеченные наружу» хранимые процедуры и шаблоны
 - Утилита для их выбора



XML в продуктах и технологиях Microsoft



- **SQLXML 1** – XML Updategrams, XML Bulkload, улучшения в производительности
- **SQLXML 2** – поддержка аннотированных XSD, FOR XML на стороне клиента (поддержка в SQL 7.0), SQLXMLOLEDB-провайдер, интеграция с ADO.Net (SQLXML Managed Classes)
- **SQLXML 3** – SOAP для SQL Server
- **SQLXML 4** – XQuery, гетерогенные запросы

XQuery

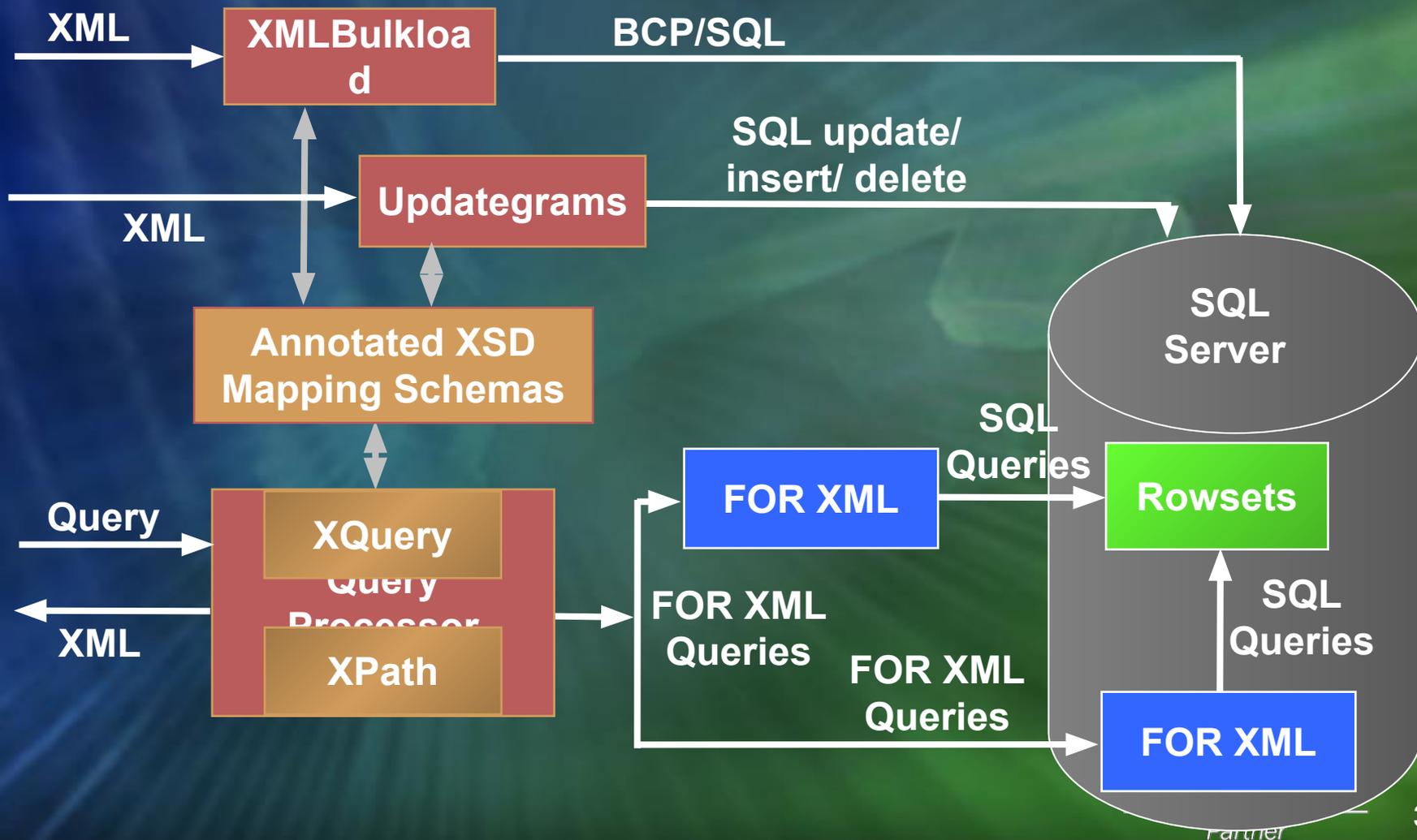
- Язык запросов XML, приходит на смену XPath
- Отличительные черты – явное конструирование элементов, возможность декомпозиции, более пригоден для работы оптимизатора
- Планируется ввести в SQL Server 2000 в SQLXML 4.0
 - Полная поддержка – в MS SQL Server «Yukon»

```
FOR $b IN //book WHERE
    $b/publisher = "Morgan Kaufmann" AND $b/year = "1998"
RETURN $b/title
```

```
FOR $p IN //procedure[1],
    $e IN //* AFTER ($p//incision)[1]
    BEFORE ($p//incision)[2]
RETURN $e
```

```
FOR $d IN /dept
WHERE EVERY $e IN $d/emp SATISFIES count($e/skill) >= 2
UPDATE $d/skill_audit WITH <skill_audit> "OK" </skill_audit>
```

Выводы: XML в SQL Server 2000



Вопросы?

Microsoft[®]

Microsoft[®]
CERTIFIED

Partner