

Тема лекції 14:

Використання XML та JSON в SQL Server

- ☐ Вибір моделі даних: реляційна чи XML
 - ☐ Технології зберігання XML-даних в SQL Server
 - ☐ Тип даних xml в SQL Server
 - ☐ Типізований і нетипізований xml
 - ☐ Завантаження XML даних
 - ☐ Керування колекціями XML-схем
 - ☐ Метадані про використання XML
 - ☐ Мова XQuery
 - ☐ Мова XML DML
 - ☐ JSON в SQL Server
-

Платформа для керування частково-або неструктурованими даними

- Від **SQL Server 2005**: реалізовано широкий діапазон обробки XML-даних:
 - значення XML можна зберігати у стовпці, який можна типізувати у відповідності з колекцією XML-схем або залишити нетипізованим; XML-стовпець можна індексувати
 - підтримка деталізованої обробки даних з використанням мов XQuery і XML DML
 - введено розширення мови Transact-SQL, які реалізують SQL-орієнтований підхід для співставлення XML-даних результатам реляційних запитів за допомогою інструкції FOR XML
 - створення на основі XML-даних реляційних представлень за допомогою інструкції OPENXML
 - підтримка власного XML
-

Доцільність використання XML-засобів в SQL Server

- ❑ Необхідно ефективно робити запити до XML-даних, змінювати, розповсюджувати їх на основі транзакцій. Велике значення має висока деталізація доступу до даних. Наприклад, потрібно вибирати деякі розділи XML-документу або вставляти в нього нові розділи без заміни всього документу.
 - ❑ Необхідно працювати як з реляційними, так і XML-даними, і забезпечити їх сумісність в прикладній програмі.
 - ❑ Необхідно забезпечити мовну підтримку запитів і модифікації даних в додатках, які охоплюють декілька доменів.
-

Доцільність використання XML-засобів в SQL Server

- ❑ Вимагається, щоб сервер гарантував істинність структури даних і, по можливості, перевіряв дані у відповідності з XML-схемами.
 - ❑ Необхідно проіндексувати XML-дані для оптимізації запитів і покращення масштабованості.
 - ❑ Необхідно звертатись до XML-даних, використовуючи технології SOAP, ADO.NET і OLE DB.
 - ❑ Необхідно для керування XML-даними використовувати засоби адміністрування, реалізовані в сервері баз даних. Прикладами таких задач керування є резервне копіювання даних, їх відновлення і реплікація.
 - ❑ Якщо жодна з цих умов не виконується, то для зберігання даних доцільніше використовувати відмінний від XML тип великих об'єктів (наприклад, [n]varchar(max) або varbinary(max))
-

Технології зберігання XML-даних в SQL Server

- ❑ Зберігання у вигляді типу xml
 - ❑ Співставлення XML-даних і реляційних даних
 - ❑ Зберігання великих об'єктів, а також [n]varchar(max) і varbinary(max)
-

Поєднання технологій зберігання XML-даних в SQL Server

Як правило, використовується поєднання цих підходів. Наприклад,

- ❑ XML-дані можна зберігати у стовпці типу XML, проводячи розповсюдження його властивостей до рівня реляційних стовпців
 - ❑ можна використовувати технологію співставлення для зберігання нерекурсивних фрагментів у стовпцях, не XML типу, а в стовпцях XML зберігати рекурсивні фрагменти
-

Технологія XML-представлень

- ❑ Визначивши відповідність між XML-схемами і таблицями бази даних, можна створити XML-представлення даних, які зберігаються
 - ❑ Щоб заповнити базові таблиці за допомогою XML-представлення, можна використовувати операцію масового завантаження XML-даних
 - ❑ Запити до XML-даних можна будувати за допомогою технології XPath, при цьому запит перетворюється в SQL-запит до таблиць
 - ❑ Операції оновлення також розповсюджуються на ці таблиці
-

Приклад 1. Моделювання XML-даних з використанням типу xml

- Припустимо, що мова йде про **інструкцію по продукції** у форматі XML, яка охоплює ряд тем, які поділені на декілька глав, і які включають декілька розділів. Розділ може містити підрозділи. Таким чином, елемент <section> є рекурсивним.
 - Інструкції з використання продукції містять великий об'єм інформації, схем і технічних характеристик. Ці дані **структуровані частково**.
 - Користувачам таких інструкцій може знадобитись можливість контекстного пошуку певних тем.
 - Для зберігання таких документів доцільно використовувати стовпець типу xml. Це дозволяє зберегти InfoSet-вміст XML-даних.
 - Індекссування XML-стовпця дозволить підвищити ефективність обробки запитів.
-

Приклад 2. Зберігання точних копій XML-даних

- ❑ Припустимо, що державні закони вимагають, щоб зберігались точні текстові копії XML-документів. До такої категорії відносяться підписані документи, юридичні постанови, звіти про біржові операції тощо. Такі дані можна зберігати у стовпці типу `[n]varchar(max)`.
- ❑ При виконанні запитів необхідно перетворити такі дані в тип XML і виконати для них запит XQuery.
- ❑ Однак перетворення типів під час виконання запитів може призвести до значної трати ресурсів, особливо, якщо документ великий.
- ❑ У випадку високої частоти запитів документи можна додатково зберігати у стовпці типу XML та індексувати саме його. А для повернення точних копій документу використовувати стовпець `[n]varchar(max)`.
- ❑ Зауважимо, що XML-стовпець може бути обчислюваним стовпцем на основі стовпця `[n]varchar(max)`. Однак для обчислюваного XML-стовпця, а також для стовпців типу `[n]varchar(max)` або `varbinary(max)` не можна створити XML-індекс.

Приклад 3. Моделювання даних з використанням анотованої XML-схеми (AXSD)

- ❑ Нехай у наявності є реляційні дані (наприклад, відомості про замовників, товар і замовлення), які треба обробляти як XML.
- ❑ У цьому випадку треба визначити XML-представлення, застосувавши схему AXSD до реляційних даних.
- ❑ Така модель особливо ефективна, якщо потрібно обмінюватись даними, які містять XML-розмітку, з іншими програмами без зупинки роботи програм SQL.
- ❑ Досить часто для моделювання даних найкраще підходить **комбінація** реляційних стовпців і стовпців типу xml. Деякі значення XML даних можна зберігати в реляційних стовпцях, а решта – в XML-стовпці. Це може привести до підвищення продуктивності за рахунок повного контролю над індексами, які створено для реляційних стовпців.

Тип даних xml в SQL Server

- ❑ Тип xml (як інші вбудовані типи даних) можна використовувати як тип стовпця, змінної або функції
 - ❑ Тип даних xml можна використовувати в інструкціях CAST і CONVERT
 - ❑ Тип даних xml і зв'язані з ним методи полегшують інтеграцію XML в реляційну платформу SQL Server.
-

Обмеження на тип даних xml в SQL Server

- ❑ розмір збереженого екземпляру типу даних xml не повинен перевищувати 2 ГБ;
 - ❑ тип xml не може використовуватись в якості підтипу екземпляру sql_variant;
 - ❑ перетворення до типів text або ntext не підтримується; замість цього використовуються типи varchar(max) і nvarchar(max);
 - ❑ не допускаються операції порівняння і сортування – це означає, що дані типу xml не можуть використовуватись у фразі GROUP BY;
 - ❑ тип xml не може використовуватись в якості параметрів будь-яких скалярних вбудованих функцій, крім ISNULL, COALESCE та DATALENGTH;
-

Обмеження на тип даних xml в SQL Server

- ❑ стовпець типу xml не може використовуватись як ключовий стовпець індексу, однак може включатись в кластерний індекс у вигляді даних або явно додаватись в некластерний індекс за допомогою ключового слова INCLUDE;
 - ❑ тип даних xml не підтримує наступних обмежень стовпця і таблиці: PRIMARY KEY і FOREIGN KEY, UNIQUE, RULE, COLLATE
 - ❑ XML використовує своє власне кодування. Параметри сортування застосовуються лише до рядкових типів, а тип даних xml до таких не відноситься. Однак у нього є рядкове представлення, і його можна приводити до рядкових типів даних і навпаки
-

Типізований і нетипізований xml

- ❑ У SQL Server можна створити змінні, параметри і стовпці типу xml.
 - ❑ При необхідності можна зв'язати колекцію XML-схем зі змінною, параметром або стовпцем типу xml. У такому випадку екземпляр типу даних xml називається **типізованим**.
 - ❑ В іншому випадку екземпляр XML називається **нетипізованим**.
-

Нетипізовані змінні і параметри типу xml

- ❑ У нетипізованому XML-стовпці тип даних XML дозволяє зберігати синтаксично коректні XML-документи, а також так звані фрагменти XML-вмісту з текстовими вузлами і довільним числом елементів верхнього рівня
 - ❑ Система, яка перевіряє правильність формату даних, не вимагає, щоб стовпець був пов'язаний з XML-схемами, і відхиляє, які мають неправильний формат в загальноприйнятому сенсі
-

Типізовані змінні і параметри типу xml

- ❑ **Обмеження перевірки.** SQL Server перевіряє типізований екземпляр XML після кожної операції присвоєння чи заміни
 - ❑ **Відомості про тип даних.** Схеми надають відомості про типи атрибутів та елементів в екземплярі типу даних XML. Відомості про тип дозволяють точніше визначити семантику операцій над значеннями, які містяться в екземплярі, у порівнянні з нетипізованим XML
 - Наприклад, десяткові арифметичні дії можуть виконуватись над десятковими значеннями, але не можуть виконуватись над рядковим типом. З цієї причини типізоване XML-сховище може займати значно менше місця, ніж нетипізоване.
-

Доцільність використання нетипізованого xml

- ❑ Немає схеми XML-даних
 - ❑ Є схеми, але не бажано, щоб сервер перевіряв дані. Такі випадки мають місце, коли
 - програма перед збереженням даних на сервері перевіряє їх на стороні клієнта
 - програма тимчасово зберігає XML-дані, які не відповідають схемі
 - програма використовує компоненти схеми, які не підтримуються сервером
-

Доцільність використання типізованого xml

- ❑ Існують схеми XML-даних і необхідно, щоб сервер перевіряв відповідність даних цим схемам
 - ❑ Потрібно оптимізувати зберігання даних і обробку запитів на основі інформації про типи даних
 - ❑ Потрібно у повній мірі використовувати інформацію про типи при компіляції запитів
-

Завантаження XML даних

- ❑ Масове завантаження XML-даних на сервер можна виконати за допомогою засобу масового завантаження **bcp**
 - ❑ Інструкція OPENROWSET дозволяє завантажити дані в XML-стовпець з файлів
-

Приклад завантаження XML даних

- У таблицю T завантажуються значення XML-стовпця з файлу як об'єкт CLOB, а в цілочисельний стовпець вводиться значення 10.

```
INSERT INTO T
```

```
  SELECT 10, xCol FROM
```

```
(SELECT * FROM OPENROWSET
```

```
  (BULK 'C:\MyFile\xmlfile.xml',
```

```
  SINGLE_CLOB) AS xCol) AS R(xCol) ;
```

Системи кодування XML-даних

- ❑ SQL Server зберігає XML-дані в кодуванні Юнікод (UTF-16)
 - ❑ Якщо потрібні дані в іншій системі кодування, то вибрані XML-дані з бази даних сервера можна представити в іншій системі.
 - ❑ Якщо текст XML представлено в Юнікодi (UCS-2, UTF-16), то можна без додаткових заходів занести його в стовпець або змінну XML.
 - ❑ Якщо текст XML представлено не в Юнікодi, можна **НЕЯВНО** задати кодування, використавши рядково-символьну кодову сторінку і фразу COLLATE.
 - ❑ Якщо такої кодової сторінки на сервері немає, то потрібно додати **ЯВНУ** XML-декларацію з коректним кодуванням. Для цього потрібно використати тип `varbinary()`, оскільки він не працює з кодовими сторінками, або символьний тип для відповідної кодової сторінки. Після цього треба назначити ці дані XML-стовпцю, змінній або параметру.
-

Приклад явного вказання системи кодування

- ❑ Нехай XML-документ `vcdoc`, зберігається в типі даних `varchar(max)`, тобто не оголошений явно як XML.
- ❑ Наступна інструкція додає оголошення XML з кодуванням `iso8859-1`, приєднує до нього XML-документ, приводить результат до типу `varbinary(max)` (щоб зберегти двійковий вигляд) і, нарешті, приводить його до типу XML.
- ❑ Це дозволяє процесору XML виконувати синтаксичний аналіз даних у відповідності з вказаним кодуванням `iso8859-1` і створювати для рядково-символьних значень відповідне представлення UTF-16:

```
SELECT CAST(  
    CAST (('<?xml version="1.0"  
        encoding="iso8859-1"?>' + vcdoc)  
    AS VARBINARY (MAX)) AS XML);
```

Керування колекціями XML-схем на SQL сервері

- За допомогою колекцій XML-схем можна зв'язати XSD-схеми зі змінною або стовпцем типу xml.
 - Колекція XML-схем зберігає імпортовані XML-схеми і використовується для вирішення наступних задач:
 - перевірка екземплярів XML;
 - типізація XML-даних, які зберігаються в базі даних.
-

Імпорт колекції XML-схем

- ❑ Інструкція CREATE XML SCHEMA COLLECTION імпортує компоненти схеми в базу даних і зберігає їх в різних системних таблицях (метаданих).
 - ❑ До компонент схеми відносяться її елементи, атрибути і оголошення типів.
 - ❑ Зауважимо, що при імпорті схеми в базу даних SQL Server не зберігає саму схему, а зберігає різні окремі компоненти, які розуміє SQL Server.
 - ❑ Після створення колекції схеми можна створювати змінні або стовпці типу XML і зв'язати з ними колекцію схем, тобто створити типізований XML.
-

Приклад зв'язування колекції схем зі змінною типу xml

```
DECLARE @x xml  
(Production.ProductDescriptionSchemaCollection);
```

де

- Production – це ім'я схеми (*schemaName*),
 - ProductDescriptionSchemaCollection – ім'я колекції XML-схем (*XmlSchemacollectionName*).
-

Приклад вказання схеми для стовпця xml

- Використовуються інструкції
 - CREATE TABLE або
 - ALTER TABLE

```
CREATE TABLE T1(  
    Col1 int,  
    Col2 xml  
        (schemaName.XmlSchemacollectionName)  
);
```

Зберігання XML-документів або XML-вмісту

- ❑ Зберігати XML-документи або XML-вміст можна у **типізованих** XML-стовпцях, параметрах і змінних
 - ❑ При оголошенні колекції XML-схеми необхідно вказати параметр DOCUMENT, якщо кожен екземпляр XML має рівно один елемент верхнього рівня
 - ❑ В іншому випадку треба вказати параметр CONTENT (за замовчуванням)
 - ❑ Компілятор запитів використовує прапорець DOCUMENT при перевірці типів під час компіляції запитів для визначення одинарних елементів верхнього рівня.
-

Приклад зберігання XML-документів або XML-вмісту

```
CREATE TABLE T1 (  
  Col1 xml  
  (DOCUMENT schemaName.XmlSchemacollectionName)  
);
```

```
CREATE TABLE T2 (  
  Col_1 xml  
  (CONTENT schemaName.XmlSchemacollectionName)  
);
```

Інші інструкції керування колекціями XML-схем

- ❑ Додавати великі компоненти в існуючі схеми або нові схеми в існуючу колекцію можна за допомогою інструкції
`ALTER XML SCHEMA COLLECTION`
 - ❑ Видалити усю колекцію XML-схем, включаючи її компоненти можна за допомогою інструкції
`DROP XML SCHEMA COLLECTION`
-

Метадані про використання XML в SQL Server

- ❑ Відомості про колекції XML-схем містить представлення каталогу **sys.xml_schema_collections**. Це інформація про імена колекцій, дату створення, власника колекції. Також це представлення містить визначений простір імен, який можна використовувати у всіх користувацьких колекціях XML-схем, не завантажуючи їх явно. Цей список містить простір імен xml, xs, xsi, fn і xdt.
 - ❑ У представленні каталогу **sys.xml_schema_namespaces** вказуються усі простори імен кожної колекції XML-схем.
 - ❑ У представленні каталогу **sys.xml_components** вказані усі компоненти кожної XML-схеми.
-

Метадані про використання XML в SQL Server

- У представленні каталогу **sys.xml_indexes** входить інформація про XML-індекси. Це представлення містить усі стовпці `sys.indexes` і деякі специфічні стовпці, корисні при роботі з XML-індексами.
 - Зауважимо, що XML-індекси входять в `sys.indexes` з індексом типу 3, а стовпець `name` при цьому містить ім'я XML-індексу.
 - Інформацію про простір, який займають XML-індекси, можна отримати за допомогою функції **sys.dm_db_index_physical_stats**. Вона подає таку інформацію в табличному вигляді (для всіх індексів): кількість сторінок на диску, які займає індекс; середній розмір рядка в байтах; кількість записів.
-

Методи типу даних XML

- ❑ query ('XQuery')
 - ❑ value (XQuery, SQLType)
 - ❑ exist (XQuery)
 - ❑ modify (XML_DML)
 - ❑ nodes (XQuery) as Table(Column)
 - ❑ Функції розширення XQuery
 - ❑ Інструкції розширення XML DML
-

Мова XQuery

- ❑ Саме завдяки тому, що в SQL Server підтримується тип даних xml, XML-документи можуть зберігатись в базі даних і з них може вибиратись інформація за допомогою мови XQuery.
 - ❑ XQuery є мовою, яка може виконувати запити до структурованих або напівструктурованих XML-даних.
 - ❑ Розробляється консорціумом W3C за участю всіх основних поставщиків СУБД.
 - ❑ XQuery базується на мові запитів XPath з додатковою покращеною підтримкою ітерацій, результатів сортування і можливості конструювання необхідних структур XML.
 - ❑ XQuery являється строго типізованою мовою для типів схеми і слабо типізованою для нетипізованих даних.
-

Приклад виконання запиту до екземпляра XML

- ❑ Якщо екземпляр XML зберігається у змінній або стовпці типу xml, то можна використовувати методи типу даних xml.
- ❑ Наприклад, можна оголосити змінну типу xml і виконати запит, використовуючи метод query():

```
DECLARE @x xml
```

```
SET @x='<ROOT><a>111</a></ROOT>';
```

```
SELECT @x.query('/ROOT/a');
```

Стандартні типи даних мови XQuery

- ❑ вбудовані типи XML-схем в просторі імен XMLSchema, які позначаються стандартним префіксом `xs`. Наприклад, `xs:integer` та `xs:string`. Вбудовані типи можуть використовуватись при створенні колекції схем.
 - ❑ Типи, визначені у просторі імен XPath-datatypes, які позначаються стандартним префіксом `xdt`. Такі типи не можуть використовуватись при створенні колекції XML-схем.
-

Запит XQuery

Складається з прологу і тексту запиту:

- ❑ Пролог XQuery – це набір оголошень і визначень, які створюють необхідне середовище для обробки запиту. В SQL Server в пролозі XQuery містяться оголошення простору імен.
 - ❑ Текст запиту XQuery містить послідовність виразів, які визначають бажаний результат вибірки XML-даних.
-

Ідентифікатор в XQuery

- ❑ Ідентифікатор в XQuery є іменем QName.
 - ❑ Ім'я QName складається з префікса простору імен і локального імені.
-

Простори імен, які використовуються в SQL Server для підтримки XQuery

Префікс

xs

xsi

xd

fn

(без префікса)

sqltypes

XML

Простір імен

XMLSchema

XMLSchema-instance

xpath-datatypes

xpath-datatypes

xml-sql

sqltypes

namespace

Вираз у запиті XQuery

- Результат виразу в XQuery є послідовністю XML-вузлів (таких як елемент, атрибут, текст, інструкція з обробки, коментар чи документ) і екземплярів атомарних типів XSD.
 - Контекст виразу являє собою дані, які використовуються для аналізу і оцінки. Існує дві фази оцінки XQuery:
 - статичний контекст – це фаза компіляції запиту;
 - динамічний контекст – це фаза виконання запиту.
-

Статичний контекст

Ініціалізація статичного контексту відноситься до процесу об'єднання даних для статичного аналізу. При цьому виконуються наступні операції:

- ❑ Ініціалізується зв'язування префікса з простором імен
 - ❑ Дозволяються зв'язування простору імен у процесі ініціалізації статичного контексту.
 - ❑ У запиті, який використовує типізований стовпець або змінну `xml`, компоненти колекції XML-схем імпортуються в статичний контекст.
 - ❑ Функція приведення також стає доступною в статичному контексті для кожного атомарного типу в імпортованих схемах
 - ❑ Після ініціалізації статичного контексту аналізується (компілюється) вираз запиту
-

Приклад приведення для вбудованого типу XML int

- у виразі вказується функція приведення для вбудованого типу XML int:

```
declare @x xml
```

```
set @x=
```

```
select @x.query('xs:int(5)')
```

Статичний аналіз

включає наступні операції:

- ❑ аналіз запиту;
 - ❑ дозвіл функції та імен типу, вказаного у виразі;
 - ❑ статичну типізацію запиту – це гарантує безпеку запиту.
-

Динамічний контекст

- пов'язаний з даними, які повинні бути доступними під час виконання виразу
 - тому, окрім статичного контексту відбувається ініціалізація фокусу виразу, такого як елемент контексту, положення контексту і розмір контексту.
-

Тип даних xml і елемент контексту

- ❑ Тип даних xml встановлює для вузла документа елемент контексту і вузол, що обробляється.
 - ❑ Положення елементу контексту, який відноситься до вузла, що обробляється, встановлюється в 1.
 - ❑ Розмір контексту – це кількість елементів в послідовності – встановлюється в 1, оскільки завжди існує один вузол документу.
 - ❑ Всі ці значення можна відмінити за допомогою методу **nodes()**.
-

Мова модифікації XML-даних (XML DML)

- є розширенням мови XQuery, що стосується маніпулювання даними
 - включає наступні інструкції (з врахуванням регістра), які відсутні в мові XQuery:
 - insert;
 - delete;
 - replace value of.
-

Модифікація змінних і стовпців типу xml

- ❑ Щоб модифікувати або оновити екземпляри XML, необхідно використати метод **modify()** для типу даних xml і задати відповідні інструкції всередині цього методу.
 - ❑ не можна модифікувати
 - атрибути xmlns, xmlns:* і xml:base для типізованого і нетипізованого xml
 - атрибути xsi:nil і xsi:type для типізованого xml.
 - ❑ При модифікації типізованого екземпляра XML кінцевий формат повинен бути допустимим екземпляром типу, інакше повертається помилка перевірки.
-

JSON формат

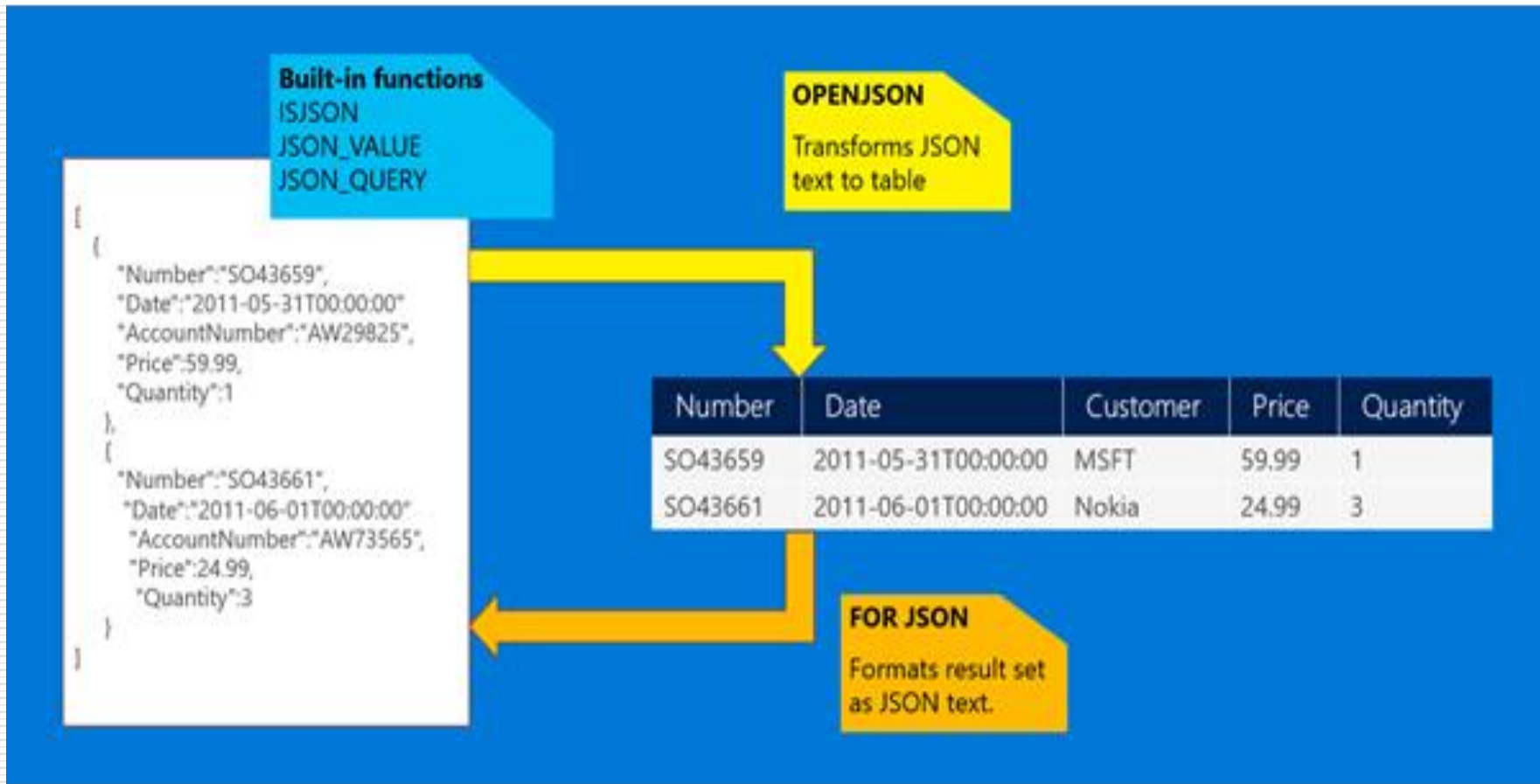
- ❑ це популярний формат текстових даних, який використовується для обміну даними в сучасних веб- і мобільних додатках
- ❑ JSON - це також основний формат обміну даними між веб-сторінками та веб-серверами за допомогою викликів AJAX
- ❑ використовується для зберігання неструктурованих даних в файлах журналів або базах даних NoSQL
- ❑ багато веб-служб REST повертають результати в форматі тексту JSON або приймають дані в форматі JSON.

JSON в SQL Server

Функції JSON з SQL Server 2016

- ❑ об'єднують принципи NoSQL і реляційних баз даних в одній базі даних
 - ❑ об'єднують в одній таблиці класичні реляційні стовпці за допомогою стовпців, які містять документи в форматі тексту JSON
 - ❑ аналізують і імпортуєть документи JSON в реляційні структури або форматовувати реляційні дані в текст JSON
-

Дані JSON в SQL Server



Дякую за увагу
