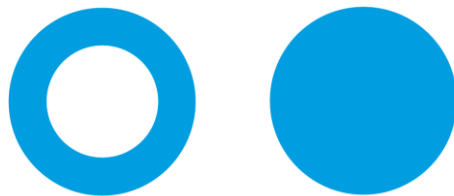


JSON & XML

jetzt mit der Version **SQL Server 2016**
gilt es die jeweiligen Vorteile abzuwägen



Organizer



**Hochschule
Bonn-Rhein-Sieg**

Bronze Sponsor



Silver Sponsor



Gold Sponsor



about me



Alexander Karl

.net - CDE 

SQL + BI Consultant

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
Solutions Associate
SQL Server 2012/2014

Microsoft
CERTIFIED
Solutions Expert
Data Platform

my previous slides

www.net-CDE.de/presentations

about me



Ralph Kemperdick

Digital Business Architect, Data

Ralph Kemperdick ist seit vielen Jahren bei Microsoft Deutschland für die Systemberatung im Bereich Data Plattform tätig. Er arbeitet gemeinsam mit dem weltweit agierenden Customer Advisory Team an den größten Big Data Projekten in Deutschland. Vor seiner Zeit bei Microsoft hat er in Erfahrung in den Bereichen Softwareentwicklung und Marketing gesammelt sowie Softwareentwicklungsprojekte in Telekommunikation, Handel und Industrie geleitet. Seit 2011 beschäftigte er sich schwerpunktmäßig mit Big Data, Analytische Applikationen, Massiv Parallele Datawarehouse Infrastrukturen und Data Mining Verfahren. Er versteht sich als Sprachrohr der Microsoft Kunden zu der Produkt- und Services Entwicklungsorganisation, die derzeit mit der Weiterentwicklung der Microsoft Data Plattform beschäftigt ist.



www.microsoft.com/sqlserver

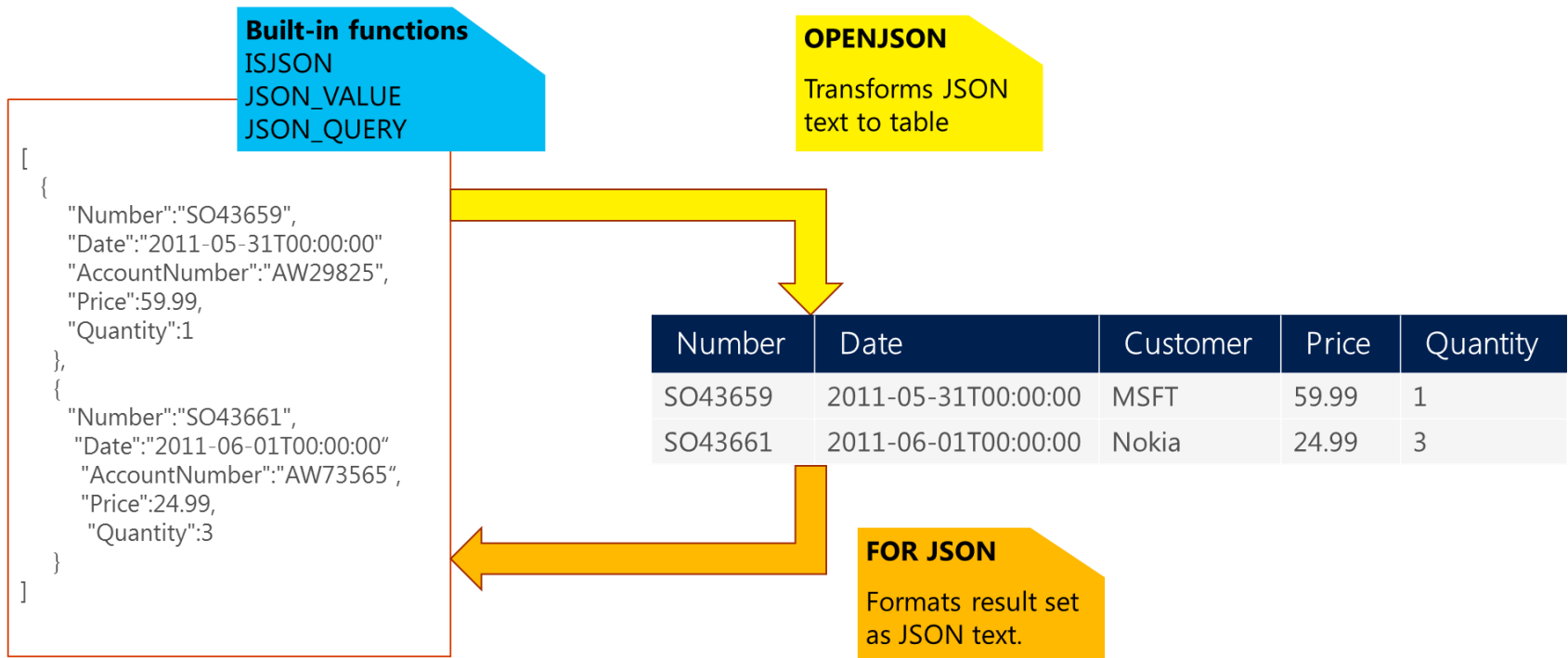
Agenda

- JSON „new feature in SQL Server 2016“
- appropriate tools
- syntax For JSON / For XML
- syntax OpenJSON / OpenXML
- datatype ? JSON vs. XML
- indexes on datatype

Agenda

JSON	→ raus aus der Datenbank	XML
JSON	← rein in die Datenbank	XML
??	speichern in der Datenbank	nativer Datentyp XML

{JSON} „round trip“



<https://msdn.microsoft.com/en-us/library/dn921897.aspx>

<xml /> tooling

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure, with the `xmlValue` column of the `dbo.CarManufacturer_xml` table selected. The main window shows a SQL query in `SQLQuery4.sql` and its results in a table. A pop-up window shows the XML content for the first row.

```
SELECT [ID], [key], xmlValue
FROM dbo.CarManufacturer_xml
```

	ID	key	xmlValue
1	1	Key01	<Customer CustomerID="1021"><!-- info 1021 custo...
2	2	Key02	<Customer CustomerID="1022"><!-- info 1022 custo...
3	3	Key03	<Customer CustomerID="1023"><!-- info 1023 custo...
4			
5			
6			
7			
8			
9			
10			
11			
...			

```
<Customer CustomerID="1021">
  <!-- info 1021 customer since 2010-->
  <CompanyName>Alfa Romeo</CompanyName>
  <CustomerSince>2010</CustomerSince>
</Customer>
```

{JSON}

tooling

```
1 SELECT name          as 'name'
2     , database_id    as 'database_id'
3     , create_date    as 'create_date'
4     , is_published   as 'replication.is_published'
5     , is_subscribed  as 'replication.is_subscribed'
6     , is_merge_published as 'replication.is_merge_published'
7     , is_distributor as 'replication.is_distributor'
8 FROM sys.databases
9 FOR JSON PATH, ROOT('databases'), Include_NULL_Values;
10
```

100 % <

Results Messages

JSON_F52E2B61-18A1-11d1-B105-00805F49916B

1 {"databases":{"name":"master","database id":1,"create date":...

JSON_F52E2B61-18...00805F49916B1.xml × demo01.sql

```
1 {"databases":{"name":"master","database_id":1,"create_date":"2003-04-08T09:13:36.390"}}
```

Sign up for our free weekly **Web Developer Newsletter**.



ENTER THE **DOLBY AUDIO CHALLENGE**

GRAND PRIZE **\$10,000 USD**

[CLICK HERE](#)

Search for articles, questions, tips

Articles » Web Development » Client side scripting » General



Tip/Trick

[Browse Code](#)

[Stats](#)

[Revisions \(9\)](#)

[Alternatives](#)

[Comments \(11\)](#)

Add your own alternative version

Tagged as

- Javascript
- IE8
- IE10
- IE11
- IE7
- Browser

View JSON in Internet Explorer

Coding 101, 21 May 2014 CPOL

Rate this: ★★★★★

★★★★★ 4.88 (17 votes)

A simple Registry change will enable IE to display JSON responses.

Need to view JSON responses in IE?

[Hide](#) [Copy Code](#)

```
Windows Registry Editor Version 5.00;
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00
```

1. Open Notepad and paste the following:
2. Save document as *IE-Json.reg* and then run it.

ENTER THE **DOLBY AUDIO CHALLENGE**

Internet Explorer address bar: <http://www.codeproject.com/Tips/216175/View-JSON-in-Internet-Explorer>

Sign up for our free weekly **Web Developer Newsletter**.

CODE PROJECT For those who code

DOLBY AUDIO

ENTER THE **DOLBY AUDIO CHALLENGE**

GRAND PRIZE **\$10,000 USD**

CLICK HERE

Internet Explorer address bar: <http://jsonview.com/example.json>

```

{"hey": "guy","anumber": 243,"anobject": {"whoa": "nuts","anarray": [1,2,"thre
ee"], "more":"stuff"},"awesome": true,"bogus": false,"meaning": null,
"japanese":"明日がある。","link": "http://jsonview.com", "notLink":
"http://jsonview.com is great"}

```

Tagged as: Javascript, IE8, IE10, IE11, IE7, Browser

```

Windows Registry Editor Version 5.00;
; Tell IE 7,8,9,10,11 to open JSON documents in the browser on Windows XP and later.
; 25336920-03F9-11cf-8FD0-00AA00686F13 is the CLSID for the "Browse in place" .
;
[HKEY_CLASSES_ROOT\MIME\Database\Content Type\application/json]
"CLSID"="{25336920-03F9-11cf-8FD0-00AA00686F13}"
"Encoding"=hex:08,00,00,00

```

1. Open Notepad and paste the following:
2. Save document as *IE-Json.reg* and then run it.

Hide Copy Code

ENTER THE **DOLBY AUDIO CHALLENGE**

Firefox Add-ons-Manager | about:addons

Suchen

Alle Add-ons durchsuchen

Add-ons suchen
Erweiterungen
Erscheinungsbild
Plugins
Dienste





JSONView 1.1.0

Von Ben Hollis

```

{
  hey: "gu",
  anumber: 243,
  anobject: {
    whoa: "nuts",
    anarray: [
      1,
      2,
      "threelive"
    ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "素晴らしい。",
  link: http://jsonview.com,
  notlink: "http://jsonview.com is great"
}

```

JSON-Dokumente im Browser anzeigen.

Normalerweise bietet Firefox beim Aufrufen eines JSON-Dokuments (Content-Type "application/json") das Abspeichern der Datei an. Die JSONView-Erweiterung stellt JSON-Dokumente analog wie XML-Dokumente dar. Das Dokument verfügt über Formatierungen sowie Hervorhebungen und Felder und Objekte können erweitert und zusammengefasst werden. Enthält das JSON-Dokument Fehler, so zeigt JSONView nur den Quelltext an.

Um JSONView in Aktion zu sehen, rufen Sie nach der Installation <http://benhollis.net/software/jsonview/example.json> auf.

Automatische Updates	<input type="radio"/> Standard <input checked="" type="radio"/> Ein <input type="radio"/> Aus
Zuletzt aktualisiert	Sunday, February 21, 2016
Homepage	http://jsonview.com/
Bewertung	★★★★★ 143 Bewertungen

The screenshot shows a Firefox browser window with the 'Add-ons-Manager' tab active. The address bar displays 'http://jsonview.com/example.j...'. The page content is a JSON object rendered by the JSONView extension, showing a hierarchical structure with expandable sections. The JSON data is as follows:

```
{
  hey: "guy",
  anumber: 243,
  anobject: {
    whoa: "nuts",
    anarray: [
      1,
      2,
      "thr<h1>ee"
    ],
    more: "stuff"
  },
  awesome: true,
  bogus: false,
  meaning: null,
  japanese: "明日がある。",
  link: http://jsonview.com,
  notLink: "http://jsonview.com is great"
}
```

{JSON} Formatter & Validator

JSON FORMATTER & VALIDATOR

About Learn Bookmarklet Changelog Support Contact

JSON Data/URL

```
{ "Header": [ { "symbol": "MSFT", "issuerName": "MICROSOFT CORP", "trdPrc": "51.82", "netChg": "-0.37", "pcntChg": "-0.71", "bid": "51.80", "ask": "51.81", "bidSize": "0", "askSize": "0", "cumVol": "33559073", "sectyType": "0" }, { "putcallInd": "C", "symbol": "MSFT1619B25", "exch": "US", "expr": "2016-02-20T00:00:00", "occExprDate": "2016-02-19T00:00:00", "strikePrc": "25.00", "trdPrc": "27.61", "netChg": "0.0", "bid": "26.70", "ask": "27.10", "cumVol": "0", "opnInt": "1", "shortDatedInd": "0", "putcallIndX": "P", "symbolX": "MSFT1619N25", "exchX": "US", "exprX": "2016-02-20T00:00:00", "occExprDateX": "2016-02-19T00:00:00", "strikePrcX": "25.00", "trdPrcX": "0.02", "netChgX": "0.0", "bidX": "0.0", "askX": "0.04", "cumVolX": "0", "opnIntX": "2", "shortDatedIndX": "0" }, { "putcallInd": "C", "symbol": "MSFT1619B28", "exch": "US", "expr": "2016-02-20T00:00:00", "occExprDate": "2016-02-19T00:00:00", "strikePrc": "28.00", "trdPrc": "20.30", "netChg": "0.0", "bid": "22.20", "ask": "22.70", "cumVol": "0", "opnInt": "1", "shortDatedInd": "0", "putcallIndX": "P", "symbolX": "MSFT1619N28", "exchX": "US", "exprX": "2016-02-20T00:00:00", "occExprDateX": "2016-02-19T00:00:00", "strikePrcX": "28.00", "trdPrcX": "20.30", "netChgX": "0.0", "bidX": "22.20", "askX": "22.70", "cumVolX": "0", "opnIntX": "1", "shortDatedIndX": "0" } ] }
```

Paste in JSON or a URL and away you go.

JSON Standard
RFC 4627

JSON Template
3 Space Tab

Process

{JSON} Formatter & Validator

The screenshot shows a web browser window with the URL `https://jsonformatter.curiousconcept.com/`. The page title is "JSON FORMATTER & VALIDATOR". The navigation menu includes "About", "Learn", "Bookmarklet", "Changelog", "Support", and "Contact". A green banner at the top of the main content area reads "VALID JSON (RFC 4627)". Below this, the text "Formatted JSON Data" is displayed above a code block containing the following JSON:

```
{
  "Header": [
    {
      "symbol": "MSFT",
      "issuerName": "MICROSOFT CORP",
      "trdPrc": "51.82",
      "netChg": "-0.37",
      "pcntChg": "-0.71",
      "bid": "51.80",
      "ask": "51.81",
      "bidSize": "0",
      "askSize": "0",
      "cumVol": "33559073",
    }
  ]
}
```

Demo

Agenda

JSON ✓	→ raus aus der Datenbank	XML ✓
JSON ✓	← rein in die Datenbank	XML ✓
??	speichern in der Datenbank	nativer Datentyp XML

{JSON} FOR JSON

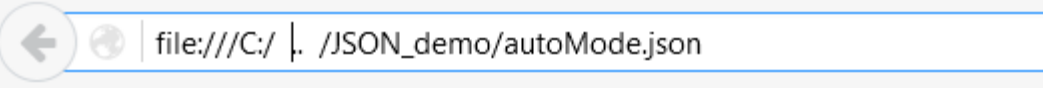
■ AUTO Mode

```
SELECT name, database_id, create_date
FROM   sys.databases
FOR    JSON Auto
```

■ PATH Mode

```
SELECT name                as 'name',
       , database_id       as 'database_id',
       , create_date       as 'create_date',
       , is_published      as 'replication.is_published',
       , is_subscribed     as 'replication.is_subscribed',
       , is_distributor    as 'replication.is_distributor',
FROM   sys.databases
FOR    JSON PATH, ROOT('databases'), Include_NULL_Values;
```

{JSON} FOR JSON

- AI 

```
SEI [
FRC  ▼ {
FOF   name: "master",
       database_id: 1,
       create_date: "2003-04-08T09:13:36.390"
     },
     ▼ {
SEI   name: "tempdb",
       database_id: 2,
       create_date: "2016-02-24T11:40:54.827"
     },
     ▼ {
FRC   name: "model",
FOF   database_id: 3,
       create_date: "2003-04-08T09:13:36.390"
     },
     .s_published,
     .s_subscribed,
     .s_distributor,
     .....de_NULL_Values;
```


{JSON} FOR JSON

file:///C:/.../JSON_demo/pathMode.json

```
■ AI {
  SEI [
  FRC  ▼ {
  FOF   name: "master",
        database_id: 1,
        create_date: "2003-04-08T09:13:36.390",
        replication: {
          is_published: false,
          is_subscribed: false,
          is_merge_published: false,
          is_distributor: false
        }
      },
  P/  ▼ {
  SEI   name: "tempdb",
        database_id: 2,
        create_date: "2016-02-24T11:40:54.827",
        replication: {
          is_published: false,
          is_subscribed: false,
          is_merge_published: false,
          is_distributor: false
        }
      },
  FRC  },
  FOF. }
}
```

<xml /> FOR XML

■ AUTO Mode

```
SELECT name, database_id, create_date
FROM   sys.databases
FOR    XML Auto ;
```

■ PATH Mode

```
SELECT name                as 'name',
       , database_id       as 'database_id',
       , create_date       as 'create_date',
       , is_published      as 'replication/is_published',
       , is_subscribed     as 'replication/is_subscribed',
       , is_distributor    as 'replication/is_distributor',
FROM   sys.databases
FOR    XML PATH ('db'), ROOT ('databases');
```

{JSON} OpenJSON

```
DECLARE @json NVARCHAR(MAX)
SET @json = '{
    "Name"      : "PASS Deutschland e.V.",
    "addInfo"   : null,
    "ID"        : 828,
    "Current"   : true,
    "Skills"    : ["SQL", "SSIS", "SSRS", 42, "MDX"],
    "Region"    : {"Country": "Germany", "Territory": "Hessen"}
}';

SELECT * -- [key], [value], type
FROM   OpenJSON (@json);

SELECT *
FROM   OpenJSON (@json, '$.Skills')

SELECT *
FROM   OpenJSON (@json, '$.Region')
```



OpenJSON

```

DECLARE @json
SET @json = '{
    "Name"
    "addInf
    "ID"
    "Curren
    "Skills
    "Region
    }';

```

```

SELECT * -- [k
FROM OpenJSON

```

```

SELECT *
FROM OpenJSON

```

```

SELECT *
FROM OpenJSON

```

Results		Messages	
	key	value	type
1	Name	PASS Deutschland e.V.	1
2	addInfo	NULL	0
3	ID	828	2
4	Current	true	3
5	Skills	["SQL","SSIS","SSRS",42,"MDX"]	4
6	Region	{'Country':"Germany","Territory':"Hessen"}	5

	key	value	type
1	0	SQL	1
2	1	SSIS	1
3	2	SSRS	1
4	3	42	2
5	4	MDX	1

	key	value	type
1	Country	Germany	1
2	Territory	Hessen	1

essen"}

{JSON}

OpenJSON Data Type

```
1 DECLARE @json NVARCHAR(MAX)
2 SET @json = '{
3     "Name"      : "PASS Deutschland e.V.",
4     "addInfo"   : null,
5     "ID"        : 828,
6     "Current"   : true,
7     "Skills"    : ["SQL", "SSIS", "SSRS", 42, "MDX"],
8     "Region"    : {"Country": "Germany", "Territory": "Hessen"}
9 }';
10
```

100 %

Results

Messages

	key	value	type	Data Type
1	Name	PASS Deutschland e.V.	1	string
2	addInfo	NULL	0	null
3	ID	828	2	int
4	Current	true	3	true/false
5	Skills	["SQL", "SSIS", "SSRS", 42, "MDX"]	4	array
6	Region	{"Country": "Germany", "Territory": "Hessen"}	5	object

{JSON} Storing JSON

- NVARCHAR (MAX)
- Index
„abgeleitete Spalte“
mit `JSON_VALUE (@JsonCol, '$. ')`

>> CREATE INDEX ...



JSON Namespaces

The screenshot shows a web browser window with the URL `https://msdn.microsoft.com/de-de/library/system.runtime.serialization.json(v=vs.110).aspx`. The breadcrumb navigation shows `System.Runtime Namespaces > System.Runtime.Serialization.Json`. A yellow notice states: "Dieser Artikel wurde maschinell übersetzt. Bewegen Sie den Mauszeiger über die Sätze im Artikel, um den Originaltext anzuzeigen. [Weitere Informationen](#)". Below this, there are radio buttons for "Übersetzung" (selected) and "Original". The main heading is "System.Runtime.Serialization.Json-Namespace". Underneath, it says ".NET Framework (current version) | [Andere Versionen](#)". The main text reads: "Stellt Klassen bereit, die mit Json-Serialisierung verknüpft sind." On the right side, there is a sidebar with options: "Vorschläge?", "Drucken", "Exportieren (0)", and "Teilen". Below that, it says "IN DIESEM ARTIKEL" followed by a link for "Klassen".



JSON schema validation

JSON Schema Generator

jsonschema.net/#/

Home About Contact Resources Previous Version

JSON

URL:

JSON:

```
{
  "Header": [
    {
      "symbol": "MSFT",
      "issuerName": "MICROSOFT CORP",
      "trdPrc": "51.82",
      "netChg": "-0.37",
      "pcntChg": "-0.71",
      "bid": "51.80",
      "ask": "51.81",
      "bidSize": "0",
      "askSize": "0",
      "cumVol": "33559073",
    }
  ]
}
```

Well done! You provided valid JSON.

Metadata Include metadata keywords

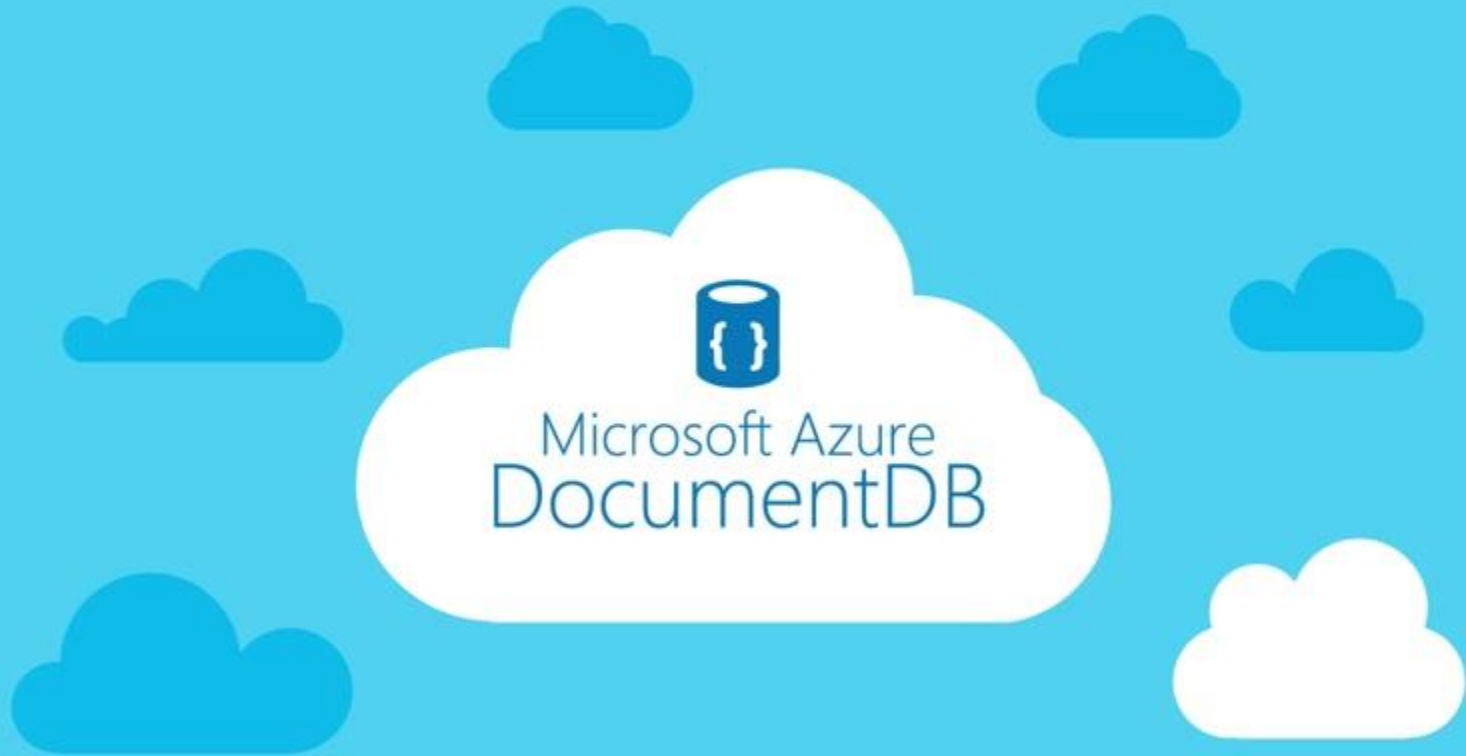
Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "",
  "type": "object",
  "properties": {
    "Header": {
      "id": "/Header",
      "type": "array",
      "items": {
        "id": "/Header/0",
        "type": "object",
        "properties": {
          "symbol": {
            "id": "/Header/0/symbol",
            "type": "string"
          },
          "issuerName": {
            "id": "/Header/0/issuerName",
            "type": "string"
          },
          "trdPrc": {
            "id": "/Header/0/trdPrc",
            "type": "string"
          }
        }
      }
    }
  }
}
```


{JSON} JSON functions

- **ISJSON()**
return 1 if JSON snippet is valid
- **JSON_VALUE()**
extract a scalar value from JSON snippet
2nd option [lax | strict] to handle missing values
- **JSON_QUERY()**
extract an object or an array from JSON snippet

.. and Microsoft ❤️ to store JSON data here



parallel zur DocumentDB
hier ein Lösungsvorschlag *on premises*

SQL Server
mit Tables + Indexes
und Volltextsuche

JSON

Ralph.Kemperdick@microsoft.com

Digital Business Architect, Data
Microsoft Deutschland GmbH

Data Loading: Log analysis

```
SELECT c.CountryRegion, min_income, COUNT(*) as Anzahl
FROM OPENROWSET (BULK N'C:\demo\JSON\customers.csv', FORMATFILE =
'C:\demo\JSON\customers.xml',CODEPAGE = '65001') as c
    CROSS APPLY OPENJSON(IncomeBracket, '$.IncomeBracket')
        WITH(CountryRegion varchar(50), min_income varchar(20))
WHERE c.customer_id >= 500
GROUP BY c.CountryRegion, min_income
Order by Anzahl desc, min_income, c.CountryRegion
```

Performance

Text processing

Indexes

NONCLUSTERED indexes on computed columns
Full text indexes on arrays or (key,value) pairs

In-Memory OLTP

Memory optimized, lock free, optimistic concurrency
Native compilation

Clustered Column Store Integration

NVARCHAR(MAX) support
10-30x compression, batch mode processing

Function	Execution time (microsec)
@xml.value()	30-50
JSON_VALUE	11-13
CHARINDEX	7-9
PATINDEX	7-9
SUBSTRING	1-2
SUBSTRING+CHARINDEX	18-22

NCI*: Index JSON path

```
SELECT ProductID, Name  
FROM ProductCatalog  
WHERE JSON_VALUE(Data, '$.Color') = 'Silver'
```

```
ALTER TABLE ProductCatalog  
ADD vColor AS JSON_VALUE(Data, '$.Color')
```

```
CREATE INDEX idx_Color  
ON ProductCatalog(vColor)
```

*NCI = Non Clustered Index

FTS*: Index any array elements

```
["Red", "Gold", "White"]
```

```
SELECT ProductID, Name  
FROM ProductCatalog  
WHERE CONTAINS(Keywords, 'Silver OR Red')
```

*FTS = Full Text Search

FTS*: Index any key:value

```
{  
  "Color": "Silver",  
  "Count": 4  
}
```

```
SELECT ProductID, Name  
FROM ProductCatalog  
WHERE CONTAINS(Data,  
  '(Color~Silver) AND NEAR((Count,4),1)')  
AND JSON_VALUE(Data, '$.Color') = 'Silver'  
AND JSON_VALUE(Data, '$.Count') = '4'
```

*FTS = Full Text Search

User defined Compression

```
CREATE TABLE IF EXISTS Person
```

```
CREATE TABLE Person(  
    _id int identity primary key,  
    data varbinary(max),  
    value AS CAST(DECOMPRESS(data) AS nvarchar(max))  
)
```

```
INSERT INTO Person(data)  
    VALUES (COMPRESS(@json))
```

```
SELECT COMPRESS((SELECT * FROM tab FOR JSON PATH))
```

New features

Better string processing

STRING_SPLIT

FORMATMESSAGE

User defined compression

COMPRESS/DECOMPRESS

Standard GZIP compression

BULK IMPORT with UTF8

CODEPAGE = '65001'

Language enhancements

DROP IF EXISTS

AT TIME ZONE

DATEDIFF_BIG

<xml /> Select & Ins/Upd/Del

- T-SQL conformer Ins/Upd/Del
- T-SQL XQuery
 - <column>.query (return xml)
 - <column>.value (return sql-Type)
 - <column>.exist (return bit)
 - <column>.nodes

<xml /> Select & Ins/Upd/Del

- T-SQL Xquery Navigation im <xml>
 - child
 - descendant
 - parent
 - Attribut
 - self
 - descendant-or-self

- .node()
- .text()

<xml /> CRUD Samples

```
SQLQuery1.sql x
1  ---- 1st Query
2  SELECT [key], xmlValue.query('*') as Complete_Sequence
3  FROM   dbo.CarManufacturer_xml
4
5  ---- 2nd Query
6  SELECT [key], xmlValue.query('data(*)') as Complete_Data
7  FROM   dbo.CarManufacturer_xml
8
9  ---- 3rd Query /node() & /comment()
10 SELECT [key]
11        , xmlValue.query('/Customer/CompanyName/node()') as CompanyName
12        , xmlValue.query('/Customer/comment()')
13 FROM   dbo.CarManufacturer_xml
14
```

<xml /> Indexing

- PRIMARY XML INDEX IX_primaryXml
- CREATE XML INDEX IX_name
ON sch.table (xmlCol)
USING XML INDEX IX_primaryXml
 - FOR PATH;
 - FOR VALUE;
 - FOR PROPERTY;

<xml /> Indexing

```
xml_Indexes.sql x
1 CREATE PRIMARY XML INDEX IX_primaryXml
2 ON dbo.eConsumption_xml ( xmlValue );
3 GO
4
5
6 CREATE XML INDEX IX_Xml_PATH
7 ON dbo.eConsumption_xml ( xmlValue )
8 USING XML INDEX IX_primaryXml
9 FOR PATH;
10
11 CREATE XML INDEX IX_Xml_PROPERTY
12 ON dbo.eConsumption_xml ( xmlValue )
13 USING XML INDEX IX_primaryXml
14 FOR PROPERTY;
15
16 CREATE XML INDEX IX_Xml_VALUE
17 ON dbo.eConsumption_xml ( xmlValue )
18 USING XML INDEX IX_primaryXml
19 FOR VALUE;
20 GO
```


<xml /> Schema Validation

- XML SCHEMA COLLECTION schemaName
- (table) xmlColumn
xml (DOCUMENT schemaName)

<xml /> Schema Validation

```
xml_Schema.sql x
1 CREATE XML SCHEMA COLLECTION dbo.bookSchemaCollection
2 AS
3 N'<?xml version="1.0"?>
4 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
5   targetNamespace="http://www.test-fabrik.de/BooksSchema"
6   xmlns="http://www.test-fabrik.de/BooksSchema"
7   elementFormDefault="qualified">
8   <xs:element name="Book">
9     <xs:complexType>
10      <xs:attribute name="Title" type="xs:string"/>
11      <xs:attribute name="Price" type="xs:decimal"/>
12    </xs:complexType>
13  </xs:element>
14 </xs:schema>';
15 GO
16
17 CREATE TABLE dbo.Books_xml(
18   ID int Identity(1,1)
19   , [key] nvarchar(50)
20   , xmlRaw xml
21   , xmlValue xml(DOCUMENT bookSchemaCollection)
22 );
23 GO
24
```

<xml /> Schema Validation

xml_Schema.sql

```
1  INSERT dbo.Books_xml
2      ( [key] , xmlValue )
3  VALUES ( 'key02', '<Book xmlns="http://www.test-fabrik.de/BooksSchema"
4              Title="the booksample2"
5              Price="49.90" />' )
6
7  -----
8  -- !! decimal ,
9  INSERT dbo.Books_xml
10     ( [key] , xmlValue )
11  VALUES ( 'key03', '<Book xmlns="http://www.test-fabrik.de/BooksSchema"
12              Title="! v ! Book"
13              Price="49,90" />' )
14
15  --Msg 6926, Level 16, State 1, Line 40
16  --XML Validation: Invalid simple type value: '49,90'. Location: /*:Book[1]/*:Price
```

<xml /> queries & transformation

- XML „*FLWOR*“
for let where order return
- Aggregat functions
 - ✓ {count(\$j)}
 - ✓ {sum(\$j)}
 - ✓ {avg(\$j)}

<xml /> queries & transformation

```
xml_FLWOR.sql x
1  -- sample FLWOR
2  SELECT ID, [key], xmlValue
3      , xmlValue.value('(CustomerData/Customer/CustID)[1]', 'nvarchar(40)' ) as 'nvarchar_CustID'
4      , xmlValue.query(' for $i in CustomerData/Customer/CustID
5                          let $j := CustomerData/Customer/Consumption/consumptionValue
6                          where avg($j) >= 5
7                          return
8                          <result>
9                              {data($j)}
10                         </result>
11                         ') as 'data'
12      , xmlValue.query(' for $i in CustomerData/Customer/CustID
13                          let $j := CustomerData/Customer/Consumption/consumptionValue
14                          where avg($j) >= 5
15                          return
16                          <result>
17                              <count> {count($j)} </count>
18                              <sum> {sum($j)} </sum>
19                              <avg> {avg($j)} </avg>
20                          </result>
21                         ') as 'summary'
22  FROM  dbo.eConsumtion_xml;
```



Save the date!



#555 | MUNICH 2016

8. October 2016 – New Microsoft Office

*The next level of **SQL Server***

Neues vom SQL Server für DBA's, Developer und BI Spezialisten

PASS Camp 2016

2016
07.-09. Dezember

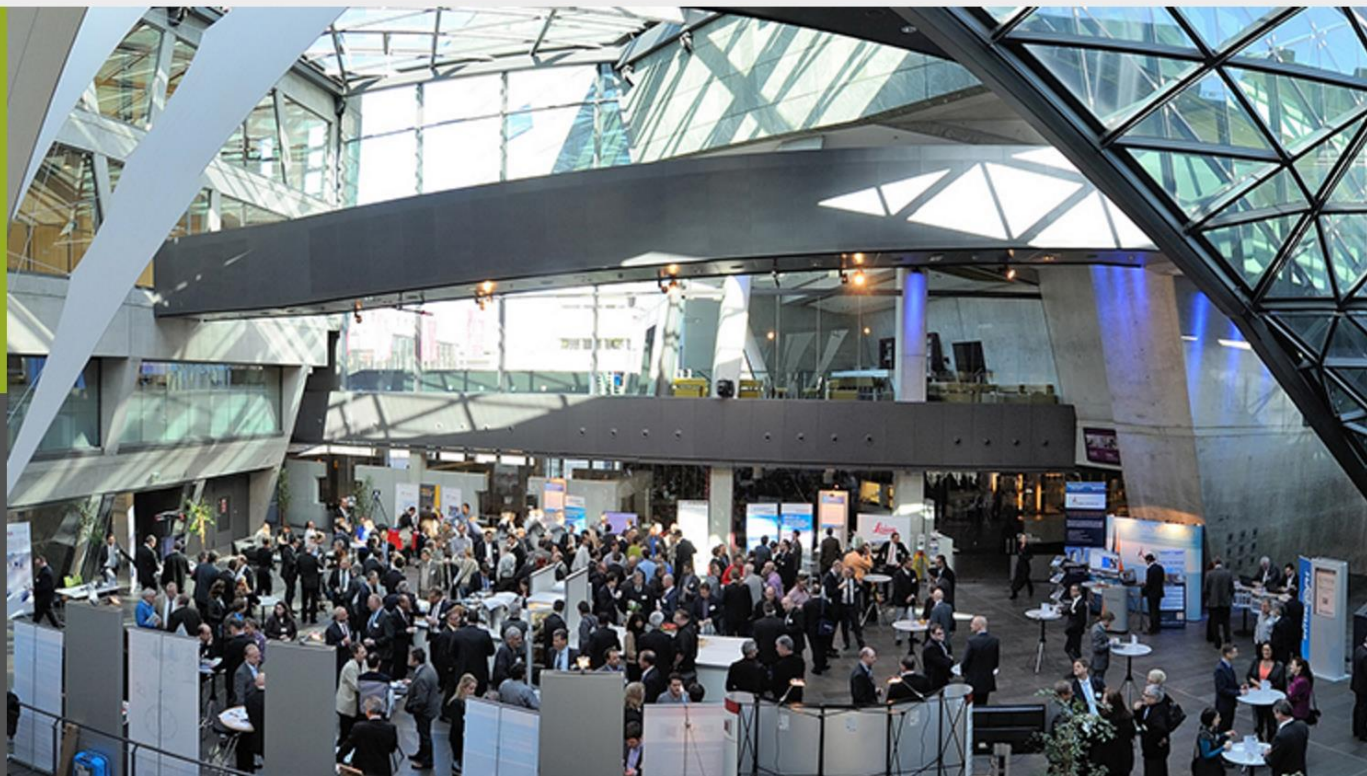
Lufthansa Conference Center
Seeheim bei Darmstadt

Save the date!

 SQL Server Konferenz 2017
powered by PASS Deutschland e.V.

FROM 14 TO 16 FEB 2017

KONGRESSCENTER
DARMSTADT
GERMANY



 Microsoft
Data Platform
Community
PASS
DEUTSCHLAND e.V.

11.06.2016

SQLSaturday Rheinland 2016

 **SQL**
saturday

#525 | RHEINLAND 2016