

# SQL Server 2016

New Feature Preview

2015-10-10

@RileyMajor



# SQL Saturday #453

- Thank you Sponsors!
  - Please visit the sponsors during the vendor break from 2:45 – 3:15 and enter their end-of-day raffles
- Event After Party
  - Dave and Buster's in Southdale Center. 3rd floor by Macy's starting at 6:15
- Want More Free Training?
  - PASSMN meets the 3rd Tuesday of every month.  
<https://mnssug.org/>

# SQL Saturday #453

- Twitter
  - @PASSMN
  - #SQLSatMN

# Presentation Overview

- Brief History of SQL Server
- New Features in 2016
  - JSON
  - Temporal Tables
  - Interactive Query Plans
- Getting Started with 2016
- Future Thoughts
- Bio

4 | Riley Major | SQL Server 2016 – New Feature Preview | 2015-10-10



Caveats: I talk mostly about the database engine (as opposed to business intelligence), and from the perspective of a developer at a smaller shop.

## Birth of SQL Server

- Started with Ashton Tate (dBase) and Sybase
- Version 1 Shipped in 1989
- Ditched Ashton Tate with dBase IV failure
- Version 1.1 Shipped in 1990 – Sybase adds support for Windows
- Version 4.2 Shipped in 1992 – First significant Microsoft involvement

5 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 | MINNESOTA 2015

My first databases were built with dBase IV. I made a database of states and people in my class. I even had a spot for pictures.

<http://blogs.msdn.com/b/euanga/archive/2006/01/19/sql-mythbusters-sql-server-is-really-a-sybase-product-not-a-microsoft-one.aspx>

[https://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server)

<https://en.wikipedia.org/wiki/Ashton-Tate>

<https://en.wikipedia.org/wiki/Borland>

<https://en.wikipedia.org/wiki/Sybase>

## SQL Server on Windows

- Parted with Sybase in 1993
- Focused on Windows only.
- Version 6 shipped in 1996 – No further Sybase involvement.
- Version 7 shipped in 1998 – DTS Packages, OLAP tools.
- SQL Server 2000 – User-Defined Functions, Reporting and Analysis Services

6 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



<http://blogs.msdn.com/b/euanga/archive/2006/01/19/sql-mythbusters-sql-server-is-really-a-sybase-product-not-a-microsoft-one.aspx>

<https://en.wikipedia.org/wiki/Ashton-Tate>

<https://en.wikipedia.org/wiki/Borland>

<https://en.wikipedia.org/wiki/Sybase>

# SQL Server Grows Up

- SQL Server 2005 (v9)
  - Sybase Code “completely rewritten”.
  - SSMS
  - XML
  - TRY/ CATCH
  - APPLY Operator, Common Table Expressions
  - varchar(max)
  - OUTPUT Clause
  - PIVOT, Ranking Window Functions (ROW\_NUMBER; OVER)
  - Encryption, Hashing
  - Dynamic Management Views (DMVs), DDL Triggers, Query Notifications
  - Database Mirroring, Service Broker, SSIS, Database Mail, CLR
  - Database Snapshots\*, Table Partitioning\*, Online Index Operations\*, Mirrored Backups\*

7 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



“Dave Campbell was talking the other day about some of the aspects of his more than 10 years in SQL Server, there are still folks in the team that have been here longer than him and yet almost all of the 1000 people who now work on SQL Server (caveat, this includes the MDAC, System.Data, System.XML and WinFS teams) have never seen a line of code that was written by anyone other than an MS employee.”  
<http://blogs.msdn.com/b/euanga/archive/2006/01/19/sql-mythbusters-sql-server-is-really-a-sybase-product-not-a-microsoft-one.aspx>

All of these are standard edition except for Database Snapshots, Table Partitioning, Online Index Operations

Features Supported by the Editions of SQL Server 2005

[https://technet.microsoft.com/en-us/library/ms143761\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms143761(v=sql.90).aspx)

Paul Randal survey: as of 2013, 41% still running 2005.

<http://www.sqlskills.com/blogs/paul/build-numbers-survey-results/>

RockSolid SQL survey: as of 2013, vast majority on 2008 R2.

[http://www.rocksolidsql.com/dNews/sql\\_server\\_2012\\_where\\_is\\_the\\_love.aspx](http://www.rocksolidsql.com/dNews/sql_server_2012_where_is_the_love.aspx)

Top 10 Advanced Features of SQL Server 2005

<http://www.techrepublic.com/article/top-10-advanced-features-of-sql-server-2005/>

# The Hits Keep Coming

- SQL Server 2008 (v10)
  - Date and Time Data Types
  - VALUES (Table Value Constructor)
  - Extended Events
  - Table-Valued Parameters
  - MERGE
  - HIERARCHYID
  - Geography and Geometry Spatial Data Types
  - Filtered Indexes
  - Grouping Sets
  - Change Data Capture\*
  - Compression\*
  - Resource Governor\*
  - Transparent Data Encryption\*

8 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



## SQL Server 2008 - What's New

<https://technet.microsoft.com/en-us/magazine/2008.04.overview.aspx>

## Features Supported by the Editions of SQL Server 2008

[https://msdn.microsoft.com/en-us/library/cc645993\(v=sql.100\).aspx](https://msdn.microsoft.com/en-us/library/cc645993(v=sql.100).aspx)

## 10 New SQL Server 2008 Features

<http://www.databasejournal.com/features/mssql/article.php/3816486/10-New-SQL-Server-2008-Features.htm>

## There is no SQL Server 2010

- SQL Server 2008 R2 (v10.5)
  - Compression for Standard (Log/Backup only)
  - BI Tooling (Reporting Services, PowerPivot)
  - Master Data Services\*
  - StreamInsight\*\*
- Well, it was only a point release.
- Last Chance for Slot-based Licensing

9 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



SQL Server 2008 R2 New Features

<http://sqlmag.com/sql-server-2008/sql-server-2008-r2-new-features>

Top 10 Features of SQL 2008 R2

<http://www.databasejournal.com/features/mssql/article.php/3857466/Top-10-Features-of-SQL-2008-R2.htm>

Features Supported by the Editions of SQL Server 2008 R2

[https://msdn.microsoft.com/en-us/library/cc645993\(v=sql.105\).aspx](https://msdn.microsoft.com/en-us/library/cc645993(v=sql.105).aspx)

## That's a little better...

- SQL Server 2012 (v11)
  - TRY\_PARSE, PARSE, TRY\_CONVERT
  - FORMAT
  - CHOOSE, IIF
  - THROW
  - OFFSET / FETCH
  - Window Function Enhancements (LAG, LEAD)
  - User-Defined Server Security Roles
  - Contained Databases
  - AlwaysOn\*
  - Columnstore Indexes\*
  - Analysis Services Tabular Model\*

10 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 | MINNESOTA 2015

Features Supported by the Editions of SQL Server 2012

[https://technet.microsoft.com/en-us/library/cc645993\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/cc645993(v=sql.110).aspx)

My Top 5 SQL Server 2012 Features by Aaron Bertrand, Guest Blogger

<http://blogs.technet.com/b/dataplatforminsider/archive/2011/11/01/my-top-5-sql-server-2012-features-by-aaron-bertrand-guest-blogger.aspx>

The Performance of the T-SQL Window Functions

<https://www.simple-talk.com/sql/t-sql-programming/the-performance-of-the-t-sql-window-functions/>

SQL Server Enterprise Edition Features You Didn't Know You Could Use

<http://www.datavail.com/category-blog/sql-server-enterprise-edition-features/>

THE CASE FOR SQL SERVER 2012 ENTERPRISE (VS. STANDARD) EDITION

<http://www.midnightdba.com/Jen/2014/03/the-case-for-sql-server-2012-enterprise-vs-standard-edition/>

## DENY on T\_SQL

- SQL Server 2014 (v12)
  - Backup to Azure
  - Delayed Durability
  - Encrypted Backups
  - Buffer Pool Extension (SSD Cache)
  - Updatable Columnstore Indexes\*
  - In-Memory OLTP (Hekaton)\*

11 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



I was going to use the phrase “Where’s the beef?” for the title, but would most of the audience even get that? Also, there’s beefy stuff here; just not T-SQL related.

Features Supported by the Editions of SQL Server 2014

[https://msdn.microsoft.com/en-US/library/cc645993\(v=sql.120\).aspx](https://msdn.microsoft.com/en-US/library/cc645993(v=sql.120).aspx)

What's New (Database Engine)

[https://msdn.microsoft.com/en-US/library/bb510411\(v=sql.120\).aspx](https://msdn.microsoft.com/en-US/library/bb510411(v=sql.120).aspx)

Important New Features in SQL Server 2014

<http://sqlmag.com/sql-server-2014/sql-server-2014-important-new-features>

Top Ten: New Features in SQL Server 2014

<http://windowsitpro.com/sql-server-2014/top-ten-new-features-sql-server-2014>

(Almost) Everything You Need to Know About SQL Server 2014

<http://www.brentozar.com/archive/2013/06/almost-everything-you-need-to-know-about-the-next-version-of-sql-server/>

# The New and Shiny

- SQL Server 2016
  - Query Store
  - Dynamic Data Masking
  - JSON
  - Temporal Tables
  - Live Query Statistics
  - Always Encrypted
  - Multiple Secondary Readers for Load Balancing
  - R Algorithms in SQL
  - Stretch Database (Auto-Archive to Azure)
  - Row-Level Security
  - Automatic TempDB Optimization
  - PolyBase (T-SQL for Hadoop)

12 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



## SQL Server 2016 Datasheet

[http://download.microsoft.com/download/F/D/3/FD33C34D-3B65-4DA9-8A9F-0B456656DE3B/SQL\\_Server\\_2016\\_datasheet.pdf](http://download.microsoft.com/download/F/D/3/FD33C34D-3B65-4DA9-8A9F-0B456656DE3B/SQL_Server_2016_datasheet.pdf)

## SQL Server 2016 CTP Technical Deep Dive

<http://www.slideshare.net/idigdata/sql-server-2016-ctp-technical-deep-dive>

## Announcing Row-Level Security in Azure SQL Database

<http://www.brentozar.com/archive/2015/01/announcing-row-level-security-sql-server/>

## Reading the SQL Server 2016 Data Sheet

<http://www.brentozar.com/archive/2015/05/reading-the-sql-server-2016-data-sheet/>

## What's New in SQL Server 2016, September Update

<https://msdn.microsoft.com/en-us/library/bb500435.aspx?f=255&MSPPError=-2147217396>

## What's New in Database Engine

[https://msdn.microsoft.com/en-US/library/bb510411\(v=sql.130\).aspx](https://msdn.microsoft.com/en-US/library/bb510411(v=sql.130).aspx)

SQL Server 2016 Community Technology Preview 2.4 is available

<http://blogs.technet.com/b/dataplatforminsider/archive/2015/09/30/sql-server-2016-community-technology-preview-2-4-is-available.aspx>

10 New Features Worth Exploring in SQL Server 2016

<http://www.databasejournal.com/features/mssql/10-new-features-worth-exploring-in-sql-server-2016.html>

Top 7 Features Coming to SQL Server 2016

<https://redmondmag.com/articles/2015/06/03/features-to-sql-server-2016.aspx>

## Cha-ching!

- Query Store – store ALL the query plans!
- Your plan cache is only so big, but your disks are bigger.
- Why recompute when you can reload?
- Reboot? Would you like me to warm that cache up for you?
- Don't like this new plan? Return it for a full refund!

13 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



Monitoring Performance By Using the Query Store  
<https://msdn.microsoft.com/en-us/library/dn817826.aspx>

## <REDACTED>

- Dynamic Data Masking hides sensitive information (such as personally-identifiable information— PII) from normal users.
- There are a variety of masking functions which turn text into XXXX, dates into 2000-01-01, etc.
- UNMASK permissions are required to see real data.

14 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



Dynamic Data Masking

<https://msdn.microsoft.com/en-us/library/mt130841.aspx>

## Hold this but don't look inside!

- Always Encrypted hides data even from DBAs.
- Encryption keys live with application.
- Performance issues slightly mitigated with repeatable encryption, but that weakens security.
- Suggested for use with external DBA resources.

15 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10

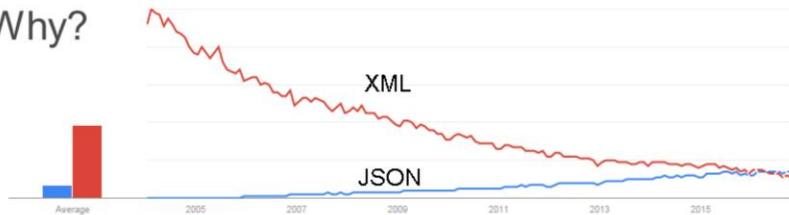


Always Encrypted

<https://msdn.microsoft.com/en-us/library/mt163865.aspx>

# JSON

- Why?



- What?

```
[  
  {  
    "MyField1": "MyValue",  
    "MyField2": 123.45  
  },  
  {  
    "MyField1": "MyValue2",  
    "MyField2": 345.67  
  }]  
]
```

16 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 | MINNESOTA 2015

## Google Search Trend

<https://www.google.com/trends/explore#q=json%2C%20xml&cmpt=q&tz=Etc%2FGMT%2B5>

Look at the timing. SQL Server 2005's XML support came at the apex of XML popularity. SQL Server 2016's JSON support comes just as JSON surpasses XML.

JSON's data format is rooted in JavaScript syntax. It's an array of objects. Each object has properties.

## References:

### JSON Data (SQL Server)

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

### Format Query Results as JSON with FOR JSON (SQL Server)

<https://msdn.microsoft.com/en-us/library/dn921882.aspx>

### SQL Server 2016 : JSON Support

By Aaron Bertrand on May 11, 2015 in T-SQL

<http://blogs.sqlsentry.com/aaronbertrand/sql-server-2016-json-support/>

How the New JSON Support Will Work in SQL Server 2016

<https://visualstudiomagazine.com/blogs/data-driver/2015/05/sql-server-json-support.aspx>

## JSON vs XML – Sample Data

```
DECLARE @Orders TABLE
(
    OrderID bigint IDENTITY,
    OrderDate datetime
);
```

```
DECLARE @OrderDetails TABLE
(
    OrderDetailsID bigint IDENTITY,
    OrderID bigint,
    ProductID varchar(50),
    Qty int
);
```

OrderID	OrderDate	ProductID	Qty
1	2015-10-10	Bike	2
1	2015-10-10	Helmet	2
1	2015-10-10	Wheels	4
2	2015-10-09	Ball	10

17 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



```
DECLARE @Orders TABLE
(
    OrderID bigint IDENTITY,
    OrderDate datetime
);

DECLARE @OrderDetails TABLE
(
    OrderDetailsID bigint IDENTITY,
    OrderID bigint,
    ProductID varchar(50),
    Qty int
);

INSERT INTO @Orders
(
    OrderDate
)
VALUES
    ('2015-10-10'),
    ('2015-10-09');

INSERT INTO @OrderDetails
(
    OrderID,
```

```

        ProductID,
        Qty
)
VALUES
        (1,'Bike',2),
        (1,'Helmet',2),
        (1,'Wheels',4),
        (2,'Ball',10);

SELECT
        Orders.OrderID,
        Orders.OrderDate,
        OrderDetails.ProductID,
        OrderDetails.Qty
FROM @Orders AS Orders
JOIN @OrderDetails AS OrderDetails
ON Orders.OrderID = OrderDetails.OrderID;

SELECT 'Path (Default)';

SELECT
        Orders.OrderID,
        Orders.OrderDate,
        OrderDetails.ProductID,
        OrderDetails.Qty
FROM @Orders AS Orders
JOIN @OrderDetails AS OrderDetails
ON Orders.OrderID = OrderDetails.OrderID
FOR XML PATH;

SELECT
        Orders.OrderID,
        Orders.OrderDate,
        OrderDetails.ProductID,
        OrderDetails.Qty
FROM @Orders AS Orders
JOIN @OrderDetails AS OrderDetails
ON Orders.OrderID = OrderDetails.OrderID
FOR JSON PATH;

SELECT 'Full Auto';

```

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM @Orders AS Orders
JOIN @OrderDetails AS OrderDetails
ON Orders.OrderID = OrderDetails.OrderID
FOR XML AUTO;
```

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM @Orders AS Orders
JOIN @OrderDetails AS OrderDetails
ON Orders.OrderID = OrderDetails.OrderID
FOR JSON AUTO;
```

```
SELECT 'Path (Custom)';
```

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    (
        SELECT
            OrderDetails.ProductID,
            OrderDetails.Qty
        FROM @OrderDetails AS OrderDetails
        WHERE Orders.OrderID =
            OrderDetails.OrderID
        FOR XML PATH('OrderDetail'), TYPE
    ) AS OrderDetails
FROM @Orders Orders
FOR XML PATH('Order'), ROOT('Orders');
```

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    (

```

```
SELECT
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM @OrderDetails AS
OrderDetails
WHERE Orders.OrderID =
OrderDetails.OrderID
FOR JSON PATH
) AS OrderDetails
FROM @Orders Orders
FOR JSON PATH, ROOT('Orders');
```

## JSON vs XML – Production (Path)

### JSON

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM    @Orders AS
        Orders
JOIN   @OrderDetails AS
        OrderDetails
ON     Orders.OrderID =
        OrderDetails.OrderID
FOR    JSON
        PATH;
```

### XML

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM    @Orders AS
        Orders
JOIN   @OrderDetails AS
        OrderDetails
ON     Orders.OrderID =
        OrderDetails.OrderID
FOR    XML
        PATH;
```

## JSON vs XML – Production (Path)

### JSON

```
[  
  {  
    "OrderID":1,  
    "OrderDate":"2015-10-10T00:00:00",  
    "ProductID":"Bike",  
    "Qty":2  
  },  
  {  
    "OrderID":1,  
    "OrderDate":"2015-10-10T00:00:00",  
    "ProductID":"Helmet",  
    "Qty":2  
  },  
  {  
    "OrderID":1,  
    "OrderDate":"2015-10-10T00:00:00",  
    "ProductID":"Wheels",  
    "Qty":4  
  },  
  {  
    "OrderID":2,  
    "OrderDate":"2015-10-09T00:00:00",  
    "ProductID":"Ball",  
    "Qty":10  
  }]
```

### XML

```
<ROW>  
  <OrderID>1</OrderID>  
  <OrderDate>2015-10-10T00:00:00</OrderDate>  
  <ProductID>Bike</ProductID>  
  <Qty>2</Qty>  
</ROW>  
  <ROW>  
  <OrderID>1</OrderID>  
  <OrderDate>2015-10-10T00:00:00</OrderDate>  
  <ProductID>Helmet</ProductID>  
  <Qty>2</Qty>  
</ROW>  
  <ROW>  
  <OrderID>1</OrderID>  
  <OrderDate>2015-10-10T00:00:00</OrderDate>  
  <ProductID>Wheels</ProductID>  
  <Qty>4</Qty>  
</ROW>  
  <ROW>  
  <OrderID>2</OrderID>  
  <OrderDate>2015-10-09T00:00:00</OrderDate>  
  <ProductID>Ball</ProductID>  
  <Qty>10</Qty>  
</ROW>
```

19 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



Formatters:

JSON

<https://jsonformatter.curiousconcept.com/>

XML

<http://www.freeformatter.com/xml-formatter.html#ad-output>

## JSON vs XML – Production (Auto)

### JSON

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM    @Orders AS
        Orders
JOIN   @OrderDetails AS
        OrderDetails
ON      Orders.OrderID =
        OrderDetails.OrderID
FOR    JSON
        AUTO;
```

### XML

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM    @Orders AS
        Orders
JOIN   @OrderDetails AS
        OrderDetails
ON      Orders.OrderID =
        OrderDetails.OrderID
FOR    XML
        AUTO;
```

# JSON vs XML – Production (Auto)

## JSON

```
[  
  {  
    "OrderID": 1,  
    "OrderDate": "2015-10-10T00:00:00",  
    "OrderDetails":  
      [  
        {"ProductID": "Bike", "Qty": 2},  
        {"ProductID": "Helmet", "Qty": 2},  
        {"ProductID": "Wheels", "Qty": 4}  
      ]  
  },  
  {  
    "OrderID": 2,  
    "OrderDate": "2015-10-09T00:00:00",  
    "OrderDetails":  
      [  
        {"ProductID": "Ball", "Qty": 10}  
      ]  
  }]
```

## XML

```
<Orders OrderID="1" OrderDate="2015-10-10T00:00:00">  
  <OrderDetails ProductID="Bike" Qty="2" />  
  <OrderDetails ProductID="Helmet" Qty="2" />  
  <OrderDetails ProductID="Wheels" Qty="4" />  
</Orders>  
<Orders OrderID="2" OrderDate="2015-10-09T00:00:00">  
  <OrderDetails ProductID="Ball" Qty="10" />  
</Orders>
```

## JSON vs XML – Path with Nesting

### JSON

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    (
        SELECT
            OrderDetails.ProductID,
            OrderDetails.Qty
        FROM    @OrderDetails AS OrderDetails
        WHERE   Orders.OrderID= OrderDetails.OrderID
        FOR JSON PATH
    ) AS OrderDetails
FROM      @Orders Orders
FOR       JSON PATH,
        ROOT('Orders');
```

### XML

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    (
        SELECT
            OrderDetails.ProductID,
            OrderDetails.Qty
        FROM    @OrderDetails AS OrderDetails
        WHERE   Orders.OrderID= OrderDetails.OrderID
        FOR XML PATH('OrderDetail'), TYPE
    ) AS OrderDetails
FROM      @Orders Orders
FOR       XML PATH('Order'),
        ROOT('Orders');
```

# JSON vs XML – Path with Nesting

## JSON

```
{  
  "Orders":  
  [  
    {  
      "OrderID": 1,  
      "OrderDate": "2015-10-10T00:00:00",  
      "OrderDetails":  
        [  
          {"ProductID": "Bike", "Qty": 2},  
          {"ProductID": "Helmet", "Qty": 2},  
          {"ProductID": "Wheels", "Qty": 4}  
        ]  
    },  
    ...  
  ]  
}
```

## XML

```
<Orders>  
  <Order>  
    <OrderID>1</OrderID>  
    <OrderDate>2015-10-10T00:00:00</OrderDate>  
    <OrderDetails>  
      <OrderDetail>  
        <ProductID>Bike</ProductID>  
        <Qty>2</Qty>  
      </OrderDetail>  
      <OrderDetail>  
        <ProductID>Helmet</ProductID>  
        <Qty>2</Qty>  
      </OrderDetail>  
      <OrderDetail>  
        <ProductID>Wheels</ProductID>  
        <Qty>4</Qty>  
      </OrderDetail>  
    </OrderDetails>  
  </Order>  
  ...  
</Orders>
```

# Unholy Unions

## XML in JSON

```
{  
    "UnholyUnion":  
        "<DataList  
        DataElement='Yes, you can put  
        XML in JSON!'><DataList  
        DataElement='But why would  
        you do this?'>"  
}
```

## JSON in XML

```
<row>  
    <UnholyUnion>  
        [{"DataElement":"Yes, you  
        can put JSON in  
        XML!"}, {"DataElement":"But why  
        would you do this?"}]  
    </UnholyUnion>  
</row>
```

SELECT

```
(  
    SELECT *  
    FROM (VALUES  
        ('Yes, you can put XML in  
        JSON!'),  
        ('But why would you do  
        this?')) AS DataList(DataElement)  
    FOR XML AUTO  
) AS UnholyUnion  
FOR JSON PATH;
```

SELECT

```
(  
    SELECT *  
    FROM (VALUES  
        ('Yes, you can put JSON in  
        XML!'),  
        ('But why would you do  
        this?')) AS DataList(DataElement)  
    FOR JSON AUTO  
) AS UnholyUnion
```

FOR XML PATH;

## Look ma, no tags!

### XML

```
SELECT  
    'Test'  
FOR XML PATH();
```

Results:

Test

### JSON

```
SELECT  
    'Test'  
FOR JSON PATH;
```

Results:

Msg 13605, Level 16, State 1, Line 1  
Unnamed tables cannot be used as  
JSON identifiers as well as unnamed  
columns cannot be used as key  
names. Add alias to the unnamed  
column/table.

## You're just not my type.

- XML is a data type.
- JSON is \*not\* a data type. Use NVARCHAR.
  - Already being stored as text.
    - But so was XML.
    - And so what? Convert over time. Convert on the fly.
  - Don't have to update other SQL Server tools.
    - Boo hoo. Ok for now, but convert over time.
  - Client apps can handle native XML but not JSON.
    - Wait, what?
    - And so what if it's text to the outside world; what about in-database performance?

26 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



JSON Support in SQL Server 2016 - Jovan Popovic (MSFT) 16 May 2015 7:17 AM  
<http://blogs.msdn.com/b/jocapc/archive/2015/05/16/json-support-in-sql-server-2016.aspx>

MSSQL Server 2016 coming with JSON support (not really)  
<http://www.itworld.com/article/2925117/enterprise-software/mssql-server-2016-coming-with-json-support-not-really.html>

## Nettlesome Nesting

### XML

```
SELECT  
    CONVERT(xml,  
            '<TextXML>I typed  
this.</TextXML>'  
        ) AS 'OuterTag'  
FOR XML PATH();
```

Results:

```
<OuterTag>  
<TextXML>I typed this.</TextXML>  
</OuterTag>
```

### JSON

```
SELECT  
    '{"TextJSON":"I typed this."}'  
AS 'OuterTag'  
FOR JSON PATH;
```

Results:

```
{"OuterTag":"\\"TextJSON\"\\"I  
typed this.\\""}\n}
```

## Nettlesome Nesting - Workaround

```
SELECT
(
    SELECT
        'I typed this.' AS TextJSON
        FOR JSON PATH
    ) AS 'OuterTag'
FOR JSON PATH;
```

Results:

```
{"OuterTag":{"TextJSON":"I typed this."}}
```

## Well is it or isn't it?

- Without JSON type, can't use TRY\_CONVERT() to validate.
- Use ISJSON() instead.
- Can use in CHECK constraint to ensure text field has valid JSON.
- Can then safely create calculated field based off JSON contents.

## Is it “rows” or “records”?

```
OPENXML                                nodes
DECLARE
    @i int,
    @x xml =
'<x><a>1</a><a>2</a></x>';
EXEC sp_xml_preparedocument
@i OUTPUT, @x;

SELECT * FROM
    OPENXML (@i, '/x/a', 2)
WITH (a varchar(10) .);

DECLARE
    @x xml =
'<x><a>1</a><a>2</a></x>';
SELECT
    a.value('.','varchar(10)')
FROM  @x.nodes('/x/a') AS x(a);
```

## But I haven't prepared!

- There is no nodes-style syntax for JSON.
- OPENJSON has similar syntax to OPENXML.
- No prepare statement is needed.
  - Work in user-defined function?
  - Multiple in single SQL statement?
  - Performance?

## OPENJSON

```
DECLARE @j nvarchar(max) = '{"Orders": [  
    {"OrderID":1, "OrderDate": "2015-10-10"},  
    {"OrderID":2, "OrderDate": "2015-10-09"}]}';  
  
SELECT  
    OrderID, OrderDate  
FROM OPENJSON (@j, '$.Orders')  
WITH  
(  
    OrderID bigint,  
    OrderDate datetime  
) AS OrdersArray;
```

## Third time's the charm.

- FOR JSON works in CTP 2.4.
- We have to wait until CTP 3 for:
  - OPENJSON()
  - JSON\_VALUE()
  - ISJSON()

33 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



SQL Server 2016 CTP Technical Deep Dive

<http://www.slideshare.net/idigdata/sql-server-2016-ctp-technical-deep-dive>

SQL Server 2016 OpenJSON Error

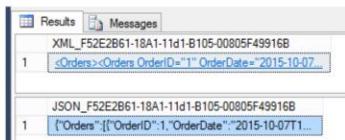
<http://www.kodyaz.com/t-sql/sql-server-2016-openjson-error.aspx>

## Jason who?

- SSMS sees JSON as syntax error.

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    OrderDetails.ProductID,
    OrderDetails.Qty
FROM
    @Orders Orders
JOIN
    @OrderDetails OrderDetails
ON
    Orders.OrderID = OrderDetails.OrderID
FOR
    JSON AUTO, ROOT('Orders');
```

- SSMS has no formatting or special handling for JSON results.



The screenshot shows the SSMS interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active and displays two rows of data. Row 1 shows XML output:

```
XML_F52E2B61-18A1-11d1-B105-00805F49916B
1 <Orders><Orders OrderID="1" OrderDate="2015-10-07T...
```

Row 2 shows JSON output:

```
JSON_F52E2B61-18A1-11d1-B105-00805F49916B
1 {"Orders": [{"OrderID": 1, "OrderDate": "2015-10-07T1...
```

# Yuck.

```
Results Messages
JSON_F52E2B61-18A1-11d1-B105-00805F49916B
1 [{"name": "syscols", "object_id": 3, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:..."}, {"2": "ped" true, "is_published" false, "is_schema_published" false}, {"name": "sysdbfrag", "object_id": 18, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:..."}, {"3": "odify_date": "2015-09-20T03:35:03.367", "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "sysowners", "object_id": 2, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:35:03.367"}, {"4": "pe_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:35:04.470", "modify_date": "2015-09-20T03:35:04.480", "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "sysconstraints", "object_id": 46, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2009-04-13T12:59:08.217", "modify_date": "2015-09-20T03:35:04.470"}, {"5": "ct_id": 46, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2009-04-13T12:59:08.217", "modify_date": "2015-09-20T03:35:04.470"}, {"6": "false", "is_schema_published" false}, {"name": "sysendpt", "object_id": 56, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:35:04.390", "modify_date": "2015-09-20T03:35:04.390"}, {"7": "fy_date": "2015-09-20T03:35:04.390", "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "sysclustobj", "object_id": 64, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2009-04-13T12:59:08.047", "modify_date": "2009-04-13T12:59:08.057"}, {"8": "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "syscatalogobj", "object_id": 81, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:..."}, {"9": "dedrecoveryforks", "object_id": 81, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2015-09-20T03:..."}, {"10": "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "sysnames", "object_id": 90, "schema_id": 4, "parent_object_id": 0, "type": "S", "type_desc": "SYSTEM_TABLE", "create_date": "2009-04-13T12:59:07.577", "modify_date": "2009-04-13T12:59:07.603"}, {"11": "is_ms_shipped" true, "is_published" false, "is_schema_published" false}, {"name": "queue_messages_1035150733", "object_id": 1051150790, "schema_id": 4, "parent_object_id": 0, "type": "IT", "type_desc": "INTERNAL_TABLE", "create_date": "2009-04-13T12:59:07.603", "modify_date": "2009-04-13T12:59:07.603"}, {"12": "false"}, {"name": "sqlagent_jobsteps_log", "object_id": 1339151816, "schema_id": 4, "parent_object_id": 0, "type": "IT", "type_desc": "INTERNAL_TABLE", "create_date": "2009-04-13T12:59:07.603", "modify_date": "2009-04-13T12:59:07.603"}, {"13": "t_id": 1467152272, "schema_id": 1, "parent_object_id": 0, "type": "V", "type_desc": "VIEW", "create_date": "2015-09-20T03:37:24.107", "modify_date": "..."}]
```

## JSON Summary

- FOR JSON
  - Works pretty much like FOR XML.
- OPENJSON
  - Similar to OPENXML, but no need for separate “preparation” step (sp\_xml\_preparedocument).
- ISJSON(@JSON)
  - Similar to TRY\_CONVERT(xml, '<x>xml</x>')
- JSON\_VALUE(@JSON, '\$.Order.Qty')
  - Similar to @XML.value

## Temporal Tables

- Like Apple's Time Machine, but for your tables.
- Can operate entirely transparently to existing applications.
- Main table gets two extra time stamp fields.
- History table:
  - Has same schema and stores old information.
  - Uses compression by default.
  - Can be queried directly.
  - Microsoft suggests using Azure stretch table.

# Temporal Tables – Basic Syntax

```
CREATE TABLE OrderDetails
(
    OrderDetailID bigint IDENTITY PRIMARY KEY,
    OrderID bigint,
    ProductID varchar(50),
    Qty int,
    EffectiveStart datetime2
        GENERATED ALWAYS AS ROW START NOT NULL,
    EffectiveStop datetime2
        GENERATED ALWAYS AS ROW END NOT NULL,
    PERIOD FOR SYSTEM_TIME (EffectiveStart, EffectiveStop)
)
WITH (SYSTEM_VERSIONING =
    ON (HISTORY_TABLE = dbo.OrderDetailsHistory));
```

38 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



## Temporal Tables

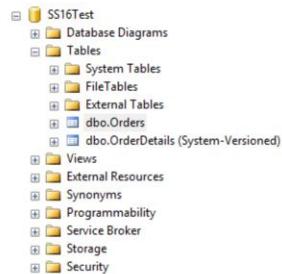
<https://msdn.microsoft.com/en-us/library/dn935015.aspx>

## CREATE TABLE (Transact-SQL)

<https://msdn.microsoft.com/en-us/library/ms174979.aspx>

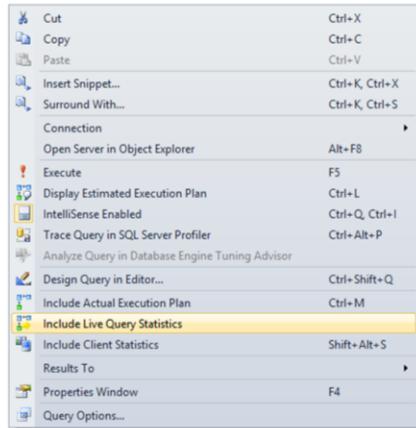
## Temporal Tables – Time Travelling

```
SELECT      *
FROM        OrderDetails
FOR
SYSTEM_TIME
AS OF @dt
ORDER BY
OrderDetailID;
```



# Live Query Statistics

- Boom!
- Wow.
- “The performance impact of turning this on, on a production server, could be significant.” – Russ Thomas
- ☹



40 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



You have to enable this in advance. You can't just start monitoring an existing long-running query.

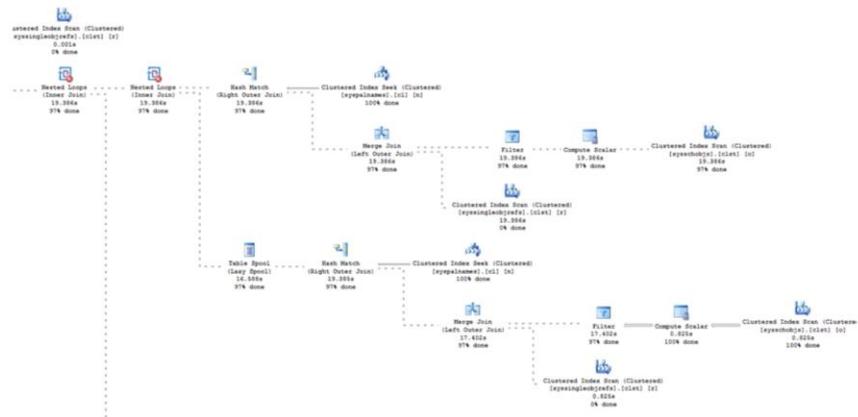
Coming Soon SQL 2016 Live Query Statistics (LQS)

<https://sqljudo.wordpress.com/2015/06/02/coming-soon-sql-2016-live-query-statistics-lqs/>

Live Query Statistics

<https://msdn.microsoft.com/en-us/library/dn831878.aspx>

# Feast your eyes...



41 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 MINNESOTA 2015

Coming Soon SQL 2016 Live Query Statistics (LQS)

<https://sqljudo.wordpress.com/2015/06/02/coming-soon-sql-2016-live-query-statistics-lqs/>

## Can I play too?

- Download and Install
  - Best with Virtualization.
  - 2 gig for SQL Server download. 4 gig for Windows.
  - This can take several hours to download and set up.
  - Free for evaluation for 120 days.
- TechNet Virtual Labs
  - Free, but limited options.
- Azure Virtual Machine
  - SQL Server 2016 Pre-Installed
- Azure SQL Database
  - Eventually

42 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



TechNet Virtual Labs

<https://technet.microsoft.com/en-us/virtuallabs>

Azure Virtual Machine with SQL Server 2016

[https://azure.microsoft.com/en-us/marketplace/partners/microsoft/sqlserver2016ctp2evaluationwindowsserver2012r2/?wt.mc\\_id=sql16\\_vm](https://azure.microsoft.com/en-us/marketplace/partners/microsoft/sqlserver2016ctp2evaluationwindowsserver2012r2/?wt.mc_id=sql16_vm)

Download Link

<https://technet.microsoft.com/en-in/evalcenter/mt130694.aspx>

## Path of Most Resistance

- You need a VM host if you don't want to mess with multiple SQL Server and SSMS versions on your main box.
  - VirtualBox is free and cross-platform.
- You need an OS.
  - Windows 10 isn't supported on VirtualBox (yet). You can get free evaluation version of Windows 8.1; lasts for 120 days.
  - If you have a spare license, you can download normal Windows 8.1 ISO. But you can't use that for the evaluation version.

43 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



You'll need a VM Host:

<https://www.virtualbox.org/wiki/Downloads>

Install VirtualBox server.

You'll need an OS:

If you have a licensed OS to spare:

<http://windows.microsoft.com/en-us/windows-8/create-reset-refresh-media>

Run and create ISO image

If you want an eval version:

<http://www.microsoft.com/en-us/evalcenter/evaluate-windows-8-1-enterprise>

You'll need the CTP bits:

<https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2016>

ISO/CAB/Azure?

<https://technet.microsoft.com/en-gb/evalcenter/dn205290.aspx>

ISO = DVD Image

CAB = “Normal” installation files (.box and .exe)

Azure = coordinate with Azure

Windows 10 ISO filename:

10240.16384.150709-1700.TH1\_CLIENTENTERPRISEVAL\_OEMRET\_X64FRE\_EN-US.ISO

Windows 10 is not yet a recommended Guest OS:

[https://www.virtualbox.org/wiki/Guest\\_OSes](https://www.virtualbox.org/wiki/Guest_OSes)

Windows 8.1 ISO filename:

9600.17050.WINBLUE\_REFRESH.140317-1640\_X64FRE\_ENTERPRISE\_EVAL\_EN-US-IR3\_CENA\_X64FREE\_EN-US\_DV9

SQL Server Install:

SQL Server 2016 Community Technology Preview 2.4

Evaluations | 180 days

SQLServer2016-x64-ENU.exe

SQLServer2016-x64-ENU.box

BOX is a compressed file format like zip.

Not easy to find info.

[http://www.microsoftstore.com/store/msusa/en\\_US/DisplayHelpSoftwareDownloadsPage](http://www.microsoftstore.com/store/msusa/en_US/DisplayHelpSoftwareDownloadsPage)

The EXE appears to just explode the BOX into a folder.

Then run setup.exe.

But first, install .NET 3.5. You will need to download from Microsoft (if you don't have an installation locally).

Windows Install

You don't need a Windows Live (aka Microsoft) account

<http://www.infobyte.hr/blog/337/windows-8-1-preview-how-to-install-without-microsoft-account-skip-microsoft-account/>

Install VirtualBox Guest Additions

Quick Link

SQL Server 2016 CTP 2.4 Announcement

<http://blogs.technet.com/b/dataplatforminsider/archive/2015/09/30/sql-server-2016-community-technology-preview-2-4-is-available.aspx>

Ready-made Azure VM

<https://azure.microsoft.com/en-us/marketplace/partners/microsoft/sqlserver2016ctp2evaluationwindowsserver2012r2/>

## Give me more. I can take it.

- You need the SQL Server 2016 CTP bits.
  - ISO – Can burn to DVD or mount with software.  
In retrospect, this is the best option.
  - EXE + BOX (compressed file like a CAB or ZIP) –  
Can just run installation.
    - However, you need space to host uncompressed and  
compressed files \*in\* the VM.
  - Azure – Mostly useful if you are testing in an  
Azure VM.
- Evaluation lasts for 180 days.

## I've got all day.

- Use VM software to make blank VM.
- Mount ISO image using VM software.
- Install Windows.
  - Enable .NET 3.5.
  - Install Java (if you want Polybase).
  - Install VM tools (to access files outside VM).
- Install SQL Server.
- So many updates!
- Plenty of reboots, too.

## Going for Broke

- Which will require Enterprise edition?
- Look at past enhancements:
  - 2005 was almost all Standard Edition.
  - Since then, headline features are Enterprise.
  - But T-SQL features usually Standard.
- “SQL Server 2014 Standard Edition Sucks, and It’s All Your Fault” – Brent Ozar, 2013-07-29

46 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



SQL Server 2014 Standard Edition Sucks, and It’s All Your Fault [UPDATE]

<http://www.brentozar.com/archive/2013/07/sql-server-2014-standard-edition-sucks-and-its-all-your-fault/>

Editions and Components of SQL Server 2016

<https://msdn.microsoft.com/en-us/library/ms144275.aspx>

## Place your bets...

- Enterprise
  - PolyBase, R Algorithms in SQL
    - Big Data is Big Bucks
  - Dynamic Data Masking
    - Compliance is Big Bucks
  - Always Encrypted & Row-Level Security
    - TDE and Fine-Grained Auditing
  - Multiple Secondary Readers
    - AlwaysOn AGs
  - Temporal Tables
    - Change Data Capture, Uses Compression

“By default, the history table is PAGE compressed.”

Temporal Tables

<https://msdn.microsoft.com/en-us/library/dn935015.aspx>

## The Crumbs

- Standard
  - JSON
  - XML
- Stretch Database
  - Previous Azure Support
- Automatic TempDB Optimization
  - Buffer Pool & Delayed Durability
- Live Query Statistics & Query Store
  - Most “RDBMS Manageability” is in Standard

48 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



Features Supported by the Editions of SQL Server 2014

[https://msdn.microsoft.com/en-US/library/cc645993\(v=sql.120\).aspx](https://msdn.microsoft.com/en-US/library/cc645993(v=sql.120).aspx)

JSON in Standard (and express):

“First of all, this is going to be standard functionality available in all editions, even Express.”

<http://blogs.sqlsentry.com/aaronbertrand/sql-server-2016-json-support>

## Say Goodbye

- SQL Server 2016 will discontinue and/or break some features, but they are “to be determined”.

Discontinued Database Engine Functionality in SQL Server 2016  
<https://msdn.microsoft.com/en-US/library/ms144262.aspx>

## Steer Clear

- Discontinued in next version
  - SET ROWCOUNT for data modification.
    - Use TOP instead.
  - Result Sets from Triggers
  - Remote Servers
    - Use Linked Servers instead.
- Still kicking:
  - Database Mirroring
  - Omitting Semicolons

Deprecated Database Engine Features in SQL Server 2016  
<https://msdn.microsoft.com/en-us/library/ms143729.aspx>

## The People Have Spoken

- JSON Support is #1 Connect Item
  - 1070 Up-Votes
- But...
  - CREATE OR REPLACE
  - 7<sup>th</sup> Most Popular (442 Up-Votes)
  - Created in March of 2005

51 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 | MINNESOTA 2015

Add native support for JSON to SQL Server, a la XML (as in, FOR JSON or FROM OPENJSON) By - bret\_m\_lowery

<https://connect.microsoft.com/SQLServer/Feedback/Details/673824>

## Shut Up and Take My Money

“While this has also been hotly requested, I’d caution you to **be careful what you ask for.** Users demanded XML support inside SQL Server, and then proceeded to use SQL Server as an XML query engine, sending CPU through the roof. **SQL Server is one of the world’s most expensive application servers.**”

– Brent Ozar, 2015-05-04

52 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



Reading the SQL Server 2016 Data Sheet

<http://www.brentozar.com/archive/2015/05/reading-the-sql-server-2016-data-sheet/>

## In the year 2000...

- SQL Server 2018?
- Brent Ozar: VMs mean 2005 & 2008 4eva
- Other popular Connect items:
  - Better Error Info for “String... truncated”
  - Error Table for Big Insert/Update Failures
  - Inline Scalar UDFs
  - Full Regex for LIKE, etc.
  - Built-in Tally Table (Table of Numbers)
  - Read/Write Table-Valued Parameters

53 | Riley Major | SQL Server 2016 – New Feature Preview 2015-10-10



#453 | MINNESOTA 2015

Most of the popular Connect items are T-SQL or developer-centric improvements but they haven't been getting a lot of love recently. It's all about Big Data and big speed.

Databases Five Years from Today

<http://www.brentozar.com/archive/2013/03/databases-five-years-from-today/>

Connect items:

Please fix the "String or binary data would be truncated" message to give the column name By – Dwalker

<https://connect.microsoft.com/SQLServer/Feedback/Details/339410>

new virtual table: errors. It would analogous to the deleted and inserted tables By - danholmes

<https://connect.microsoft.com/SQLServer/Feedback/Details/774754>

The Scalar Expression function would speed performance while keeping the benefits of functions. - by Andrew Novick

<https://connect.microsoft.com/SQLServer/Feedback/Details/273443>

Regex functionality in pattern matching By - Simon Sabin

<https://connect.microsoft.com/SQLServer/Feedback/Details/261342>

Add a built-in table of numbers By - Erland Sommarskog

<https://connect.microsoft.com/SQLServer/Feedback/Details/258733>

Relax restriction that table parameters must be readonly when SPs call each other. By

- Erland Sommarskog

<https://connect.microsoft.com/SQLServer/Feedback/Details/299296>

## Riley Major

- @RileyMajor | PASSMN@RileyMajor.com
- Enterprise Architect
- Manna Freight Systems, Inc.
- Worked with SQL Server since May of 2000
- PASSMN Board – Director of Technology
- Conference speaker
- Father of three girls

## Evaluations Please

- Remember to fill out your online evaluations for the event and any sessions you have attended. They will be online until 10/17/15.

<http://www.sqlsaturday.com/453/eventeval.aspx>

<http://www.sqlsaturday.com/453/sessions/sessionevaluation.aspx>