



PresenceSM
Health Partners

SQL Server Technology Enhancements for Developers

Upgrade from SQL Server 2008 to 2012/2014

Wylie Blanchard
SQL Server Consultant

Presentation Summary

Your team is planning to upgrade from your MS SQL Server 2008 environment to a newer platform. Learn what's new in MS SQL Server 2012/2014. Which features and enhancements are really important to the work life of a SQL Server Developer.

In this presentation we'll explore SQL Server 2012/2014 new possibilities, showing you how to use new T-SQL functions, features and enhancements that are only available in SQL Server 2012/2014.

Enhancements - SQL Server 2012/(2014)

+ T-SQL Functions

- T-SQL Analytic Functions (FIRST_VALUE, LAST_VALUE, LEAD, LAG)
- T-SQL String Functions (FORMAT, CONCAT)
- T-SQL Expression (COALESCE - not new)

+ SSMS Engine

- FileTables
- Query Breakpoints enhancements
- Sequence Numbers Objects
- Contained databases
- In-memory OLTP (2014)

T-SQL Functions

T-SQL Analytic Functions

FIRST_VALUE

LAST_VALUE

LEAD

LAG

FIRST_VALUE / LAST_VALUE

- + Returns the first and last value in a list
- + Get info from result set without using self-joins, derived tables, etc
- + Syntax: `FIRST_VALUE ([scalar_expression]) OVER ([partition_by_clause] order_by_clause [rows_range_clause])`
- + Syntax: `LAST_VALUE ([scalar_expression]) OVER ([partition_by_clause] order_by_clause rows_range_clause)`

FIRST_VALUE / LAST_VALUE - Demo

```
/** show first value in preceding rows for list */  
USE AdventureWorks2012  
GO  
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,  
FIRST_VALUE(SalesOrderDetailID) OVER (ORDER BY  
SalesOrderDetailID) FstValue,  
LAST_VALUE(SalesOrderDetailID) OVER (ORDER BY  
SalesOrderDetailID) LstValue  
FROM Sales.SalesOrderDetail s  
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty  
GO
```

FIRST_VALUE / LAST_VALUE - Demo (cont)

```
/** show first value in preceding rows and current value for
    list sectioned by value */
USE AdventureWorks2012
GO
SELECT s.SalesOrderID,s.SalesOrderDetailID,
FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY
    SalesOrderID
ORDER BY SalesOrderDetailID) FstValue,
LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY
    SalesOrderID
ORDER BY SalesOrderDetailID) LstValue
FROM Sales.SalesOrderDetail s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
GO
```

LEAD / LAG

- + Get data from subsequent (LEAD) and previous (LAG) row in same result set
- + Syntax: LAG (scalar_expression [,offset] [,default]) OVER ([partition_by_clause] order_by_clause)
- + Syntax: LAST_VALUE ([scalar_expression]) OVER ([partition_by_clause] order_by_clause rows_range_clause)

LEAD and LAG - Demo

```
/** get data from subsequent and previous row **/  
USE AdventureWorks2012  
GO  
SELECT s.SalesOrderID,s.SalesOrderDetailID,  
LEAD(SalesOrderDetailID) OVER (ORDER BY  
SalesOrderDetailID  
) LeadValue,  
LAG(SalesOrderDetailID) OVER (ORDER BY  
SalesOrderDetailID  
) LagValue  
FROM Sales.SalesOrderDetail s  
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty  
GO
```

LEAD / LAG - Demo (cont)

```
/** get data from subsequent and previous row offset by 3 **/  
USE AdventureWorks2012  
GO  
SELECT s.SalesOrderID,s.SalesOrderDetailID,  
LEAD(SalesOrderDetailID,3) OVER (ORDER BY SalesOrderDetailID  
 ) LeadValue,  
LAG(SalesOrderDetailID,3) OVER (ORDER BY SalesOrderDetailID  
 ) LagValue  
FROM Sales.SalesOrderDetail s  
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty  
GO
```

T-SQL Functions

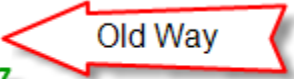
T-SQL String Functions

FORMAT
CONCAT

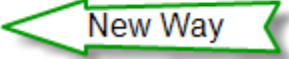
FORMAT

- + Format strings
- + Format dates
- + Format currency
 - and more

```
DECLARE @Revenue MONEY = 314159.26  
SELECT '$' + CONVERT(VARCHAR(32),@Revenue,1);  
SELECT FORMAT(@Revenue, 'C');
```



Old Way



New Way

- + Syntax: FORMAT(value, format, [culture])
 - Value is the thing to be formatted
 - Format specifies how we want it to look
 - Optional Culture specifies the specific language/locale used for formatting.
- + Easier than CONVERT

Formatting Types

- + A valid .NET Framework format string
- + C = Currency
- + D = Date
- + X = hexadecimal

```
-- old way
DECLARE @Revenue MONEY = 314159.26
SELECT '$' + CONVERT(VARCHAR(32),@Revenue,1);

-- now with format
DECLARE @Revenue MONEY = 314159.26
SELECT FORMAT(@Revenue, 'C');
SELECT FORMAT(getdate(), 'd');
SELECT FORMAT(1234, 'X');
```

FORMAT – Culture

- + The culture argument is not provided, then the language of the current session is used
 - Server default
 - SET LANGUAGE
- + Examples
 - En-us, fr-fr, de-de, jp-jp

```
-- using the culture parameter
DECLARE @Revenue MONEY = 314159.26
SELECT FORMAT(@Revenue, 'c', 'en-us') as English;
SELECT FORMAT(@Revenue, 'c', 'fr-fr') as French;
SELECT FORMAT(@Revenue, 'c', 'de-de') as German;
SELECT FORMAT(@Revenue, 'c', 'ja-JP') as Japanese;
```

FORMAT - Demo

```
/** old way **/  
DECLARE @RevenueA MONEY = 314159.26  
SELECT '$' + CONVERT(VARCHAR(32), @RevenueA, 1);
```

```
/** now with format **/  
DECLARE @RevenueB MONEY = 314159.26  
SELECT FORMAT(@RevenueB, 'C');
```

```
/** other examples **/  
SELECT FORMAT(getdate(), 'd');  
SELECT FORMAT(1234, 'X');
```

FORMAT – Demo (cont)

```
/** custom format values **/  
SELECT FORMAT(getdate(), 'MMMM dd, yyyy (dddd)');
```

```
/** format using the culture parameter **/  
DECLARE @Revenue MONEY = 314159.26  
SELECT FORMAT(@Revenue, 'c', 'en-us') as English;  
SELECT FORMAT(@Revenue, 'c', 'fr-fr') as French;  
SELECT FORMAT(@Revenue, 'c', 'de-de') as German;  
SELECT FORMAT(@Revenue, 'c', 'ja-JP') as Japanese;
```


CONCAT

- + Concatenates data
- + Easier than using + because all types are cast to strings
- +
Syntax: `CONCAT (string1, string2 [, stringN])`
- + Output is a string, input is more than one string.
- + Forces conversion to string
 - `PRINT 'Current Time ' + GETDATE()`
 - `PRINT CONCAT('Current Time ', GETDATE())`

CONCAT – Demo

```
/** concat in tsql 2012 **/  
SELECT CONCAT(1, ' two ', 3.0, ' four');
```



```
/** another example **/  
SELECT 'uniqueidentifier = ' + NEWID(); -- fails  
SELECT CONCAT('uniqueidentifier = ', NEWID());
```



```
/** print concat trick **/  
PRINT 'Time ' + GETDATE(); -- fails  
PRINT 'Time ' + CAST(GETDATE() AS VARCHAR(30));  
PRINT CONCAT('Time ', GETDATE());
```

T-SQL Expression COALESCE

- + COALESCE (not new)
 - Introduced with SQL Server 2005
 - Not new but it is a function that should be in every developer's tool belt
- + COALESCE Description
 - Returns the first non-null expression among its arguments
 - If all arguments are NULL, COALESCE returns NULL
- +
Syntax: COALESCE (expression [,...n])

COALESCE – Demo

```
/** select first non null value - returns current date **/  
SELECT COALESCE(NULL, NULL, NULL, GETDATE())  
  
/** easily change column list into comma separated values **/  
USE AdventureWorks2012  
GO  
DECLARE @columnlist VARCHAR(MAX)  
SELECT @columnlist = COALESCE(@columnlist+', ' , '') + Name  
FROM Production.Product  
SELECT @columnlist  
GO
```

COALESCE - Demo (cont)

```
/** create dynamic where clause to pass parameters **/  
USE AdventureWorks2012  
GO  
CREATE PROCEDURE usp.SearchPersons  
@FirstName varchar(20),  
@LastName varchar(20)  
AS  
SELECT BusinessEntityID, FirstName, LastName  
FROM Person.Person  
WHERE FirstName = COALESCE(@FirstName, FirstName)  
AND LastName = COALESCE(@LastName, LastName)  
GO
```

SSMS Engine **FileTable**

wait - let's talk about FILESTREAM

- + FILESTREAM (not new)
 - Introduced with SQL Server 2008
- + Used to store data in file system
- + Similar to storing blob data in varbinary(max) column
- +
Useful when data objects are too large for varbinary(max) datatype
- + Better performance for large data objects
 - use system cache instead of sql server cache

FileTables

- + Significantly enhances FILESTREAM capabilities
 - Store files and documents in special tables called FileTables
- + You can access data, in file system, using T-SQL
- + Applications can access data directly through filesystem
 - No need to change application logic
- +
Use when dealing with data objects too large for varbinary(max) datatype
- + Use when integrating with non transactional applications (middle tier applications)

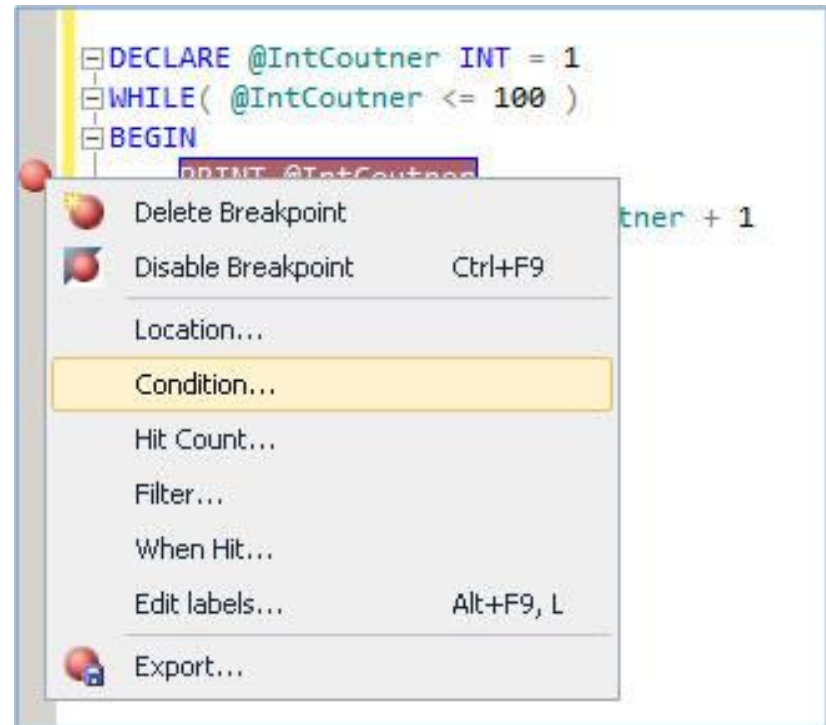
FileTables Demo

Steps:

1. Enable filestream at instance level
2. Create file stream database
3. Create file table
4. Query the file table

Query Breakpoints enhancements

- + SQL Server 2008 allows
 - create breakpoint
 - delete breakpoint
 - disable breakpoint
- + SQL Server 2012 adds
 - Conditions
 - Hit Counts (pause when cycle (x) times)
 - Filter
 - When Hit (do this)
- + Useful to watch the action of parameters



SSMS Engine

Query Breakpoints enhancements

Query Breakpoints enhancements (cont)

- + Conditions – evaluates expression
 - Useful for tracking parameter scenarios
 - Ex.: @IntCounter = 5
- + Hit Counts – track process cycles
 - Useful to pause at loop points
 - Ex.: pause when Breakpoint reached (x) times
- + Filter – searches for active specified computers, operating system processes, and threads
 - Useful for trouble shooting applications
 - Ex.: ProcessID = 123
- + When Hit – (do this)
 - Useful when action is needed once a condition is met

Query Breakpoints enhancements - Demo

```
/** Query Breakpoints demo **/  
  
Declare @IntCounter INT = 1  
WHILE( @IntCounter <= 100)  
BEGIN  
    PRINT @IntCounter --press F9 to create breakpoint and then  
    right click red circle for options  
    SET @IntCounter = @IntCounter + 1  
end
```

SSMS Engine

Sequence Objects

SEQUENCE objects

- + CREATE SEQUENCE - Automatically generates numbers
- + Database object that is an alternative to IDENTITY column
 - found and saved in the database Programmability folder
- + Can be used to get the next sequence of numbers without create a table

Example:

```
CREATE SEQUENCE EmployeeSeq AS tinyint
    START WITH 0
    INCREMENT BY 5;

GO
```

Sequence Objects vs Identity column

	SEQUENCE Object	IDENTITY column
Table independent	Yes	No
Obtain the new value in your application before using it	Yes	No
Generate new values in an UPDATE statement	Yes	No
Obtain a whole range of new sequence values in one effort	Yes	No
Define minimum and maximum values	Yes	No

SEQUENCE objects - Demo

```
/** create sequence with max value **/  
USE FileStreamDB  
CREATE SEQUENCE EmployeeSeqToError AS tinyint  
    START WITH 0  
    INCREMENT BY 5  
    MAXVALUE 100;  
  
GO
```

SEQUENCE objects - Demo (cont)

```
/** create sequence with max and min value and reseed/recycle
**/
USE FileStreamDB
CREATE SEQUENCE EmployeeSeqCycle AS tinyint
    START WITH 1
    INCREMENT BY 5
    MINVALUE 1
    MAXVALUE 100
    CYCLE;

GO
```

SSMS Engine

Contained Databases

Contained Databases

- + A database that is independent from the SQL Server instance

Benefits

- + User authentication can be performed by the database
 - reduces the databases dependency on the logins of the instance of SQL Server
- + Easily move a database from one instance of SQL Server to another
 - Metadata maintained in actual database instead of the master database
- + Give db owner more control over database, without giving the db owner sysadmin permission
 - Errors related to missing users and orphan users are no longer an issue with contained databases

Contained Databases (cont)

Disadvantages

- + DB Owner can create contained db users without the permission of a DBA
 - can lead to security issues & data theft threat
- + Can't use replication, change data capture, change tracking, numbered procedures, schema-bound objects that depend on built-in functions with collation changes
- + A user confined to the contained database may be able to access other databases on the Database Engine
 - if the other databases have enabled the guest account

Contained Databases - Demo

Steps:

1. Enable database at the server/instance level
2. Enable containment at the database level
3. Create a contained user
4. Test connectivity

Contained Databases - Demo (cont)

```
/** enable database containment on server/instance */  
sp_configure 'show advanced options', 1  
GO  
RECONFIGURE WITH OVERRIDE  
GO  
sp_configure 'contained database authentication', 1  
GO  
RECONFIGURE WITH OVERRIDE  
GO  
sp_configure 'show advanced options', 0  
GO  
RECONFIGURE WITH OVERRIDE  
GO
```

Contained Databases - Demo (cont)

```
/** enable contained database on database */  
USE [master]  
GO  
ALTER DATABASE [FileStreamDB] SET CONTAINMENT = PARTIAL WITH  
NO_WAIT  
GO  
/** create a contained user */  
USE [FileStreamDB]  
GO  
CREATE USER [MyContainedUser] WITH PASSWORD=N'!LPpeople!',  
DEFAULT_SCHEMA=[dbo]  
GO  
  
/** test connectivity */
```


Contained Databases - Demo (cont)

+ Test Connectivity

- Close and reopen SSMS
- Click “Options” once the login screen appears
- Select ‘Database Engine’ for “Server type”
- Specify the instance that hosts the database for “Server Name”
- Enter the user login credentials that were created (Do not click Connect)
- Navigate to the “Connection Properties” tab
- Specify the name of the contained database in the “Connect to Database” box
- Click “Connect”

Connect to Server

Microsoft SQL Server 2014

Login | Connection Properties | Additional Connection Parameters

Server _____
Type the server name, or choose it from the drop-down list.

Server type: Database Engine

Server name: MIA-SQL

Authentication: SQL Server Authentication

Login: MyContainedUser

Password:

Remember password

Connect Cancel Help Options <<

Connect to Server

Microsoft SQL Server 2014

Login | Connection Properties | Additional Connection Parameters

Type or select the name of the database for the connection.

Connect to database: FileStreamDB

Network _____

Network protocol: <default>

Network packet size: 4096 bytes

Connection _____

Connection time-out: 15 seconds

Execution time-out: 0 seconds

Encrypt connection

Use custom color:

Connect Cancel Help Options <<

SSMS Engine

In-memory OLTP (2014)

In-memory OLTP

New technology released with SQL Server 2014 database engine

- + Memory Optimized Tables
 - Tables using the new data structures
- + Allow highly used tables to live in memory
 - Remain in memory forever without losing out a single record
- + Designed to significantly reduce blocking and locks
- + High Performance response than disk tables due to data living in memory

In-memory OLTP - Demo

Steps:

- + Create Database Which Creates A File Group Containing Memory_Optimized_Data
- + Create two different tables 1) Regular table and 2) Memory Optimized table
- + Create two stored procedures 1) Regular SP and 2) Natively Compiled SP
- + Compare the performance of two SPs

In-memory OLTP - Demo (cont)

```
/** Create A Table With Setting Memory_Optimized Set To
Enabled **/
USE InMemory
GO
/** create a simple table **/
CREATE TABLE DummyTable (ID INT NOT NULL PRIMARY KEY,
Name VARCHAR(100) NOT NULL)
GO
/** create a memory optimized table **/
CREATE TABLE DummyTable_Mem (ID INT NOT NULL,
Name VARCHAR(100) NOT NULL
CONSTRAINT ID_Clust_DummyTable_Mem PRIMARY KEY NONCLUSTERED
HASH (ID) WITH (BUCKET_COUNT=1000000))
WITH (MEMORY_OPTIMIZED=ON)
GO
```

In-memory OLTP - Demo (cont)

```
/** Create A Stored Procedure Which Is Natively Compiled**/  
/** simple table to insert 100,000 rows **/  
CREATE PROCEDURE Simple_Insert_test  
AS  
BEGIN  
SET NOCOUNT ON  
DECLARE @counter AS INT = 1  
DECLARE @start DATETIME  
SELECT @start = GETDATE()  
WHILE (@counter <= 100000)  
BEGIN  
INSERT INTO DummyTable VALUES (@counter, 'WylieBlanchard')  
SET @counter = @counter + 1  
END  
SELECT DATEDIFF(SECOND, @start, GETDATE() ) [Simple_Insert in  
sec]  
END  
GO
```

Other Features

Not mentioned in this presentation but are worth researching.

- + Data Quality Services
- + Data Master Services
- + SQL Server Integration Services
 - SSIS - Undo and Redo Features
 - SSIS - Project Deployments
- + SQL Server Reporting Services
 - SSRS Power View
 - SSRS Data Alerts
 - PowerPivot (not new)
 - Google Chrome enhancements (2014)
- + Azure integration enhancements

Other Features

Not mentioned in this presentation but are worth researching.

- Data Quality Services
- Data Master Services
- SQL Server Integration Services
 - SSIS - Undo and Redo Features
 - SSIS - Project Deployments
- SQL Server Reporting Services
 - SSRS Power View
 - SSRS Data Alerts
 - PowerPivot (not new)
 - Google Chrome enhancements (2014)
- Azure integration enhancements

Resources

SQL Server Evaluation Downloads:

- + SQL Server 2012: <https://www.microsoft.com/en-us/download/details.aspx?id=29066>
- + SQL Server 2014: <https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2014>

AdventureWorks Sample Database Downloads

- + AdventureWorks 2012:
<http://msftdbprodsamples.codeplex.com/releases/view/55330>
- + AdventureWorks 2014:
<https://msftdbprodsamples.codeplex.com/releases/view/125550>

Questions & Answers