

# **MS SQL NEW FUNCTIONS, SYNTAXES, TIPS & TRICKS**

# DAMIR MATEŠIĆ, MAG. INF.

- Senior Database Developer @SPAN.eu
- AD 2018 - Leading Data Events in Croatia
- AD 2019 - Introduced SQL Saturday in Croatia
- AD 2020 - Co-founder & organizer of #Dataweekender...

W: [blog.matesic.info](http://blog.matesic.info)

@: [damir@matesic.info](mailto:damir@matesic.info)

in: [linkedin.com/in/dmatesic](https://www.linkedin.com/in/dmatesic)





# MS SQL NEW FUNCTIONS, SYNTAXES, <sup>3</sup>TIPS & TRICKS

1. STRING(s)
  - STRING\_SPLIT
  - STRING\_AGG
  - UTF8
2. COMPRESS AND DECOMPRESS
3. Other...
4. Something completely new 😊

# STRINGS – SQL 2016

## 1. **STRING\_SPLIT**

- table-valued function
- splitting string values by a separator

## 2. **STRING\_ESCAPE**

- escapes special characters

## 3. **FORMATMESSAGE** (2008)

- `FORMATMESSAGE ( { msg_number | 'msg_string' } , [ param_value [ ,...n ] ] )`



# STRINGS – SQL 2016

## 1. STRING\_SPLIT

- table-valued function
- splitting string values by a separator

## 2. STRING\_ESCAPE

- escapes special characters „by a rule”

## 3. FORMATMESSAGE (2008)

- `FORMATMESSAGE ( { msg_number | 'msg_string' } , [ param_value [ ,...n ] ] )`

# STRINGS – SQL 2016

## 1. STRING\_SPLIT

- table-valued function
- splitting string values by a separator

## 2. STRING\_ESCAPE

- escapes special characters

## 3. **FORMATMESSAGE (2008)**

- `FORMATMESSAGE ( {msg_number | 'msg_string' },`
- `[param_value [ ,...n ]])`





**DEMO**



# STRINGS – SQL 2017

1. **TRIM -> LTRIM(RTRIM(String)) = TRIM(String)**
2. **STRING\_AGG**
  - string aggregation using a separator
3. **TRANSLATE**
  - replaces multiple characters inside the given string value
4. **CONCAT\_WS**
  - concatenating two or more string values



# STRINGS – SQL 2017

1. `TRIM -> LTRIM(RTRIM(String)) = TRIM(String)`
2. **STRING\_AGG**
  - string aggregation using a separator
3. **TRANSLATE**
  - replaces multiple characters inside the given string value
4. **CONCAT\_WS**
  - concatenating two or more string values

# STRINGS – SQL 2017

1. `TRIM -> LTRIM(RTRIM(String)) = TRIM(String)`
2. **STRING\_AGG**
  - string aggregation using a separator
3. **TRANSLATE**
  - replaces multiple characters inside the given string value
4. **CONCAT\_WS**
  - concatenating two or more string values



# STRINGS – SQL 2017

1. `TRIM -> LTRIM(RTRIM(String)) = TRIM(String)`
2. **STRING\_AGG**
  - string aggregation using a separator
3. **TRANSLATE**
  - replaces multiple characters inside the given string value
4. **CONCAT\_WS**
  - concatenating two or more string values





**DEMO**



## STRING OR BINARY DATA WOULD BE TRUNCATED

- Old error message
  - Msg 8152, Level 16, State 30, Line 18
  - String or binary data would be truncated.
- New error message
  - Msg 2628, Level 16, State 1, Line 35
  - String or binary data would be truncated in table  
'SomeDatabase.dbo.SomeTable ', column 'Col'. Truncated value: 'XXX'

## STRING OR BINARY DATA WOULD BE TRUNCATED

- Old error message

- Msg 8152, Level 16, State 30, Line 18
- String or binary data would be truncated.

- New error message

- Msg 2628, Level 16, State 1, Line 35
- String or binary data would be truncated in table  
'SomeDatabase.dbo.SomeTable ', column 'Col'. Truncated value: 'XXX'



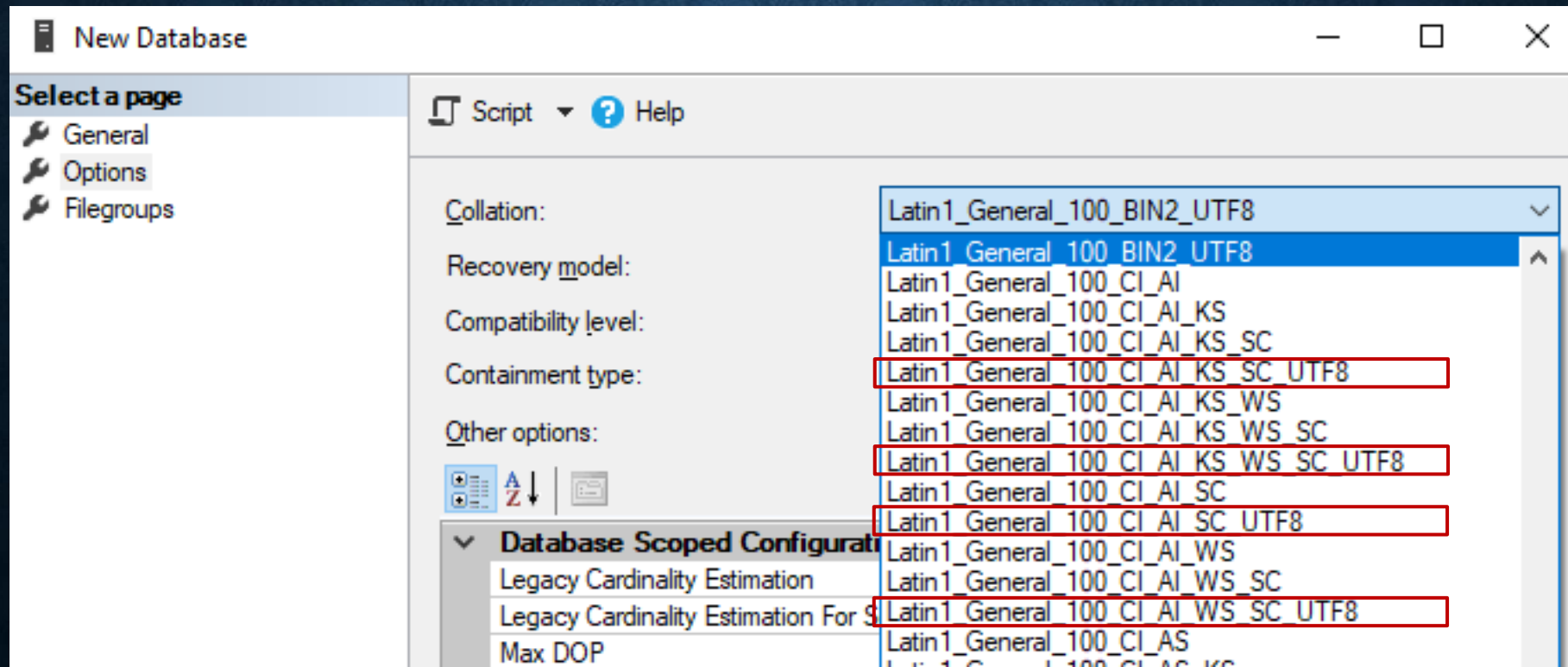
## STRING OR BINARY DATA WOULD BE TRUNCATED

- Old error message
- **Msg 8152**, Level 16, State 30, Line 18
- New error message
- **Msg 2628**, Level 16, State 1, Line 35

```
ALTER DATABASE SCOPED CONFIGURATION SET  
VERBOSE_TRUNCATION_WARNINGS = OFF;
```

# STRINGS – SQL 2019 UTF8

- Allow application(s) internationalization without converting all strings to Unicode
- Implemented as new collation -> 1.553 new collations)





# STRINGS – SQL 2019 UTF8

SQL 2019 supports UTF 8 collation -> SQL 2019 підтримує сортування UTF 8

```
USE NONUTF8  
GO
```

```
DECLARE @v VARCHAR(100) = 'SQL 2019 підтримує сортування UTF 8';  
SELECT @v AS String, DATALENGTH(@v) AS DataLengthValue;  
DECLARE @nv NVARCHAR(100) = N'SQL 2019 підтримує сортування UTF 8';  
SELECT @nv AS String, DATALENGTH(@nv) AS DataLengthValue;  
GO
```

Results	
String	DataLengthValue
SQL 2019 ?????????? ?????????? UTF 8	35
Messages	
String	DataLengthValue
SQL 2019 підтримує сортування UTF 8	70

```
USE UTF8;  
GO
```

```
DECLARE @8v VARCHAR(100) = 'SQL 2019 підтримує сортування UTF 8';  
SELECT @8v AS String, DATALENGTH(@8v) AS DataLengthValue;  
DECLARE @8nv NVARCHAR(100) = N'SQL 2019 підтримує сортування UTF 8';  
SELECT @8nv AS String, DATALENGTH(@8nv) AS DataLengthValue;  
GO
```

Results	
String	DataLengthValue
1 SQL 2019 підтримує сортування UTF 8	54
Messages	
String	DataLengthValue
1 SQL 2019 підтримує сортування UTF 8	70





**DEMO**



# COMPRESS AND DECOMPRESS<sup>10</sup>

## (1/2)

- 2016+
- ROW, PAGE...
- Syntax:
- COMPRESS (expression)
- - **Expression** - nvarchar(n), nvarchar(max), varchar(n), varchar(max), varbinary(n), varbinary(max), char(n), nchar(n), or binary(n) expression.
- - **Return**



# COMPRESS AND DECOMPRESS<sup>20</sup>

## (2/2)

- GZIP
- INDEX
- XML, Log-s, Rarely used data



# COMPRESS AND DECOMPRESS

- Opposite of COMPRESS?
- Syntax:
- DECOMPRESS (expression)
- - **Expression** - Is a varbinary(n), varbinary(max), or binary(n)
- - **Return** -> data in varbinary(max)
- Casting is recommended





**DEMO**



# CREATE OR ALTER

```
CREATE PROCEDURE dbo.sp_SQLNewFunctions AS
BEGIN
    SELECT 'This demo is cool :)' AS Result
END
```

## Messages

Msg 2714, Level 16, State 3, Procedure sp\_SQLNewFunctions, Line 1 [Batch Start Line 0]  
There is already an object named 'sp\_SQLNewFunctions' in the database.

```
IF OBJECT_ID(N'dbo.sp_SQLNewFunctions','P') IS NOT NULL
    EXEC('DROP PROCEDURE dbo.sp_SQLNewFunctions');
CREATE PROCEDURE dbo.sp_SQLNewFunctions AS
BEGIN
    SELECT 'This demo is cool :)' AS Result
END
```

```
CREATE OR ALTER PROCEDURE dbo.sp_SQLNewFunctions AS
BEGIN
    SELECT 'This demo is cool :)' AS Result
END
```

# DROP IF EXISTS (A.K.A. DIE)

```
DROP TABLE dbo.SQLNewFunctions;
```

## Messages

Msg 3701, Level 11, State 5, Line 1

Cannot drop the table 'dbo.SQLNewFunctions', because it does not exist or you do not have permission.

```
IF OBJECT_ID('dbo.SQLNewFunctions','U') IS NOT  
NULL DROP TABLE dbo.SQLNewFunctions
```

```
DROP TABLE IF EXISTS dbo.SQLNewFunctions;
```



# DATEDIFF\_BIG

```
DECLARE
    @StartDate DATETIME = GETDATE()
    , @EndDate DATETIME = DATEADD(day, 1, @StartDate)

SELECT DATEDIFF(MCS, @StartDate, @EndDate ) AS "Microsecond diff"
```

Results Messages

Msg 535, Level 16, State 0, Line 21

The datediff function resulted in an overflow. The number of dateparts separating two date/time instances is too large. Try to use datediff with a less precise datepart.

```
DECLARE
    @StartDate DATETIME = GETDATE()
    , @EndDate DATETIME = DATEADD(day, 1, @StartDate)

SELECT
    ...
    , DATEDIFF_BIG(MCS, @StartDate, @EndDate ) AS "Microsecond diff"
    , DATEDIFF_BIG(NS, @StartDate, @EndDate ) AS "Nanosecond diff"
```

Results Messages

	Week diff	Day diff	Hour diff	Minute diff	Second diff	Millisecond diff	Microsecond diff	Nanosecond diff
1	0	1	24	1440	86400	86400000	86400000000	864000000000000

# HASHBYTES

- SQL 2005 - ~~MD2, MD4, MD5, SHA, SHA1~~
- SQL 2012 - SHA2\_256, SHA2\_512
- - Input: 8 000 bytes





**DEMO**



# GREATEST() & LEAST()

- Azure SQL
  - Azure SQL Database
  - Azure SQL Managed Instance
  - Azure Synapse Analytics - serverless SQL pool only

GREATEST/LEAST (expression1 [ ,...expressionN ])

# GREATEST() & LEAST()

Return types:

- highest precedence
- same data type or implicitly convert
- numeric types



# GREATEST() & LEAST()

Remarks and limitations:

- Comparable data type that can be [implicitly converted](#)
- [Implicit conversion](#)
- NULL
- Types not supported for comparison: varchar(max), varbinary(max) or nvarchar(max) exceeding 8,000 bytes, cursor, geometry, geography, image, non-byte-ordered user-defined types, ntext, table, text, and xml.

# GREATEST() & LEAST()

GREATEST (expression1 [ ,...expressionN ])



# GREATEST() & LEAST()

LEAST (expression1 [ ,...expressionN ])





**DEMO**



**THANKS**