

# SQL Server 2019

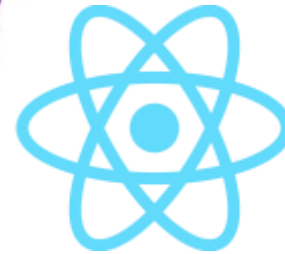
## Durch die Augen eines Programmierers



Thorsten Kansy (tkansy@dotnetconsulting.eu)

# Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,  
Entwickler, Trainer & Fachautor



Azure Cosmos DB



# Mein Service- Ihr Benefit

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
  - Problemanalyse und Lösungen
  - Technologieentscheidungen



# Agenda

- SQL Server & Tooling
- Intelligent Database
  - Intelligent Query Processing
  - In-Memory Database
  - Intelligent performance
- Neue Graph Features
- Java-Unterstützung
- SQL Server auf Linux
- Sicherheit
- Sonstiges



# SQL Server & Tooling

The background image shows a wide city square under a cloudy sky. In the center stands a tall, dark monument of a soldier holding a flag, with a plaque at its base containing Russian text. To the left is a large, light-colored building with a sign that reads 'ШМАКОВСКАЯ'. To the right is a modern building with a glass facade. People are walking in the square, and several pigeons are on the ground in the foreground.

# SQL Server 2019

- Windows
- Red Hat Enterprise Server, SUSE Linux Enterprise Server & Ubuntu
- Docker

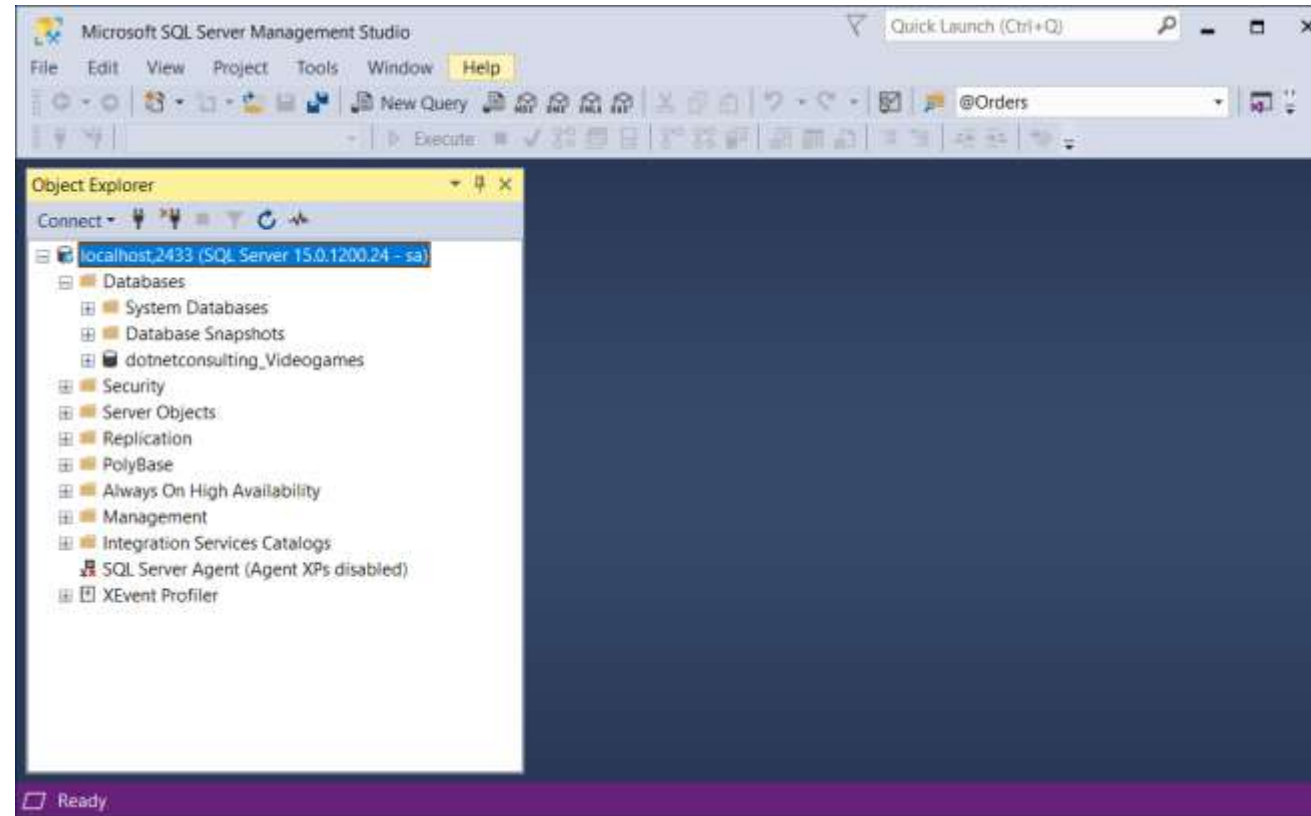
```
docker run -e ACCEPT_EULA=Y  
           -e SA_PASSWORD=P@ssw0rd99  
           -p 2433:1433  
           -d mcr.microsoft.com/mssql/server:2019-GA-ubuntu-16.04
```



<https://hub.docker.com/r/microsoft/mssql-server-linux/>

# SQL Server Management Studio 18

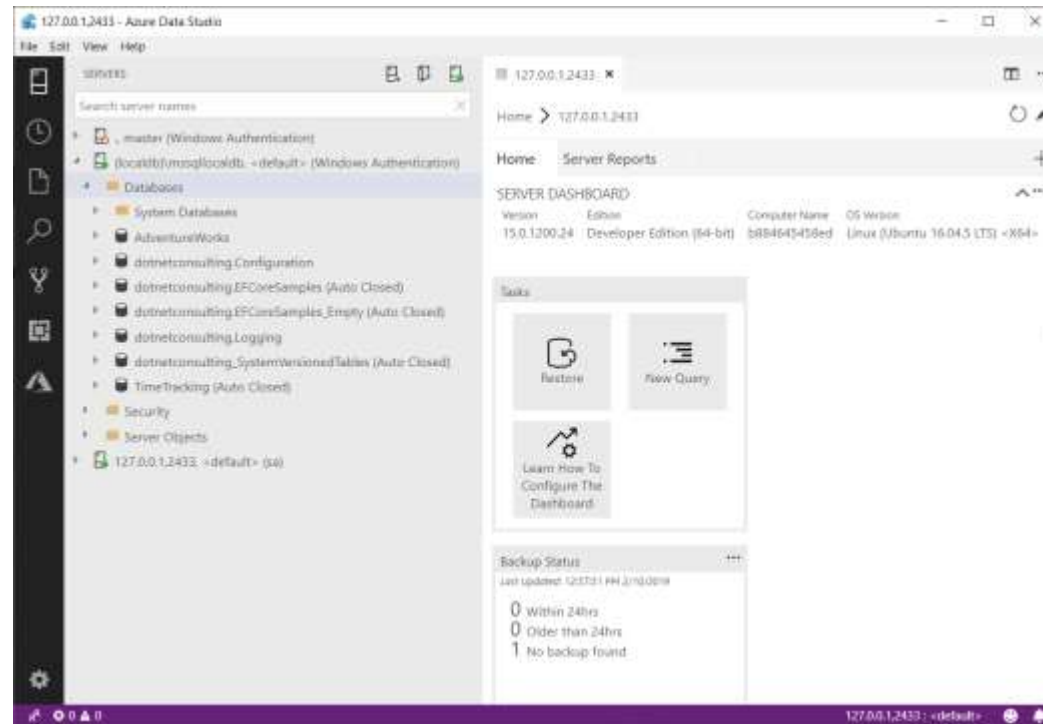
Windows only



<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

# Azure Data Studio

- Basiert auf Visual Studio Code
  - Windows
  - macOS
  - Linux



<https://docs.microsoft.com/en-us/sql/azure-data-studio>



A photograph of a brown monkey with a pinkish face, sitting on a reddish-brown rock. The monkey is looking towards the camera. The background consists of a clear blue sky and a dense green forest. A dark blue horizontal band is overlaid across the middle of the image, containing the title and subtitle text.

# Intelligent Database

Intelligent Query Processing

# Intelligent Query Processing

- Approximate query processing
  - Schnellere Anzahl eindeutiger Werte, wenn diese nicht absolut präzise sein müssen
- Scalar UDF Inlining
  - Automatisches Einbetten von skalaren UDF in Abfrage

## Nicht Entwickler Features:

Row mode memory grant feedback, Batch mode on rowstore, Table variable deferred compilation

# Approximate query processing

Liefert schnellere Anzahl eindeutiger Werte, wenn diese nicht absolut präzise sein müssen

```
-- Eindeutige Werte  
SELECT APPROX_COUNT_DISTINCT([OrderQty]) AS Approx_Distinct_OrderKey  
FROM [Sales].[SalesOrderDetail];
```

“The function implementation guarantees up to a 2% error rate within a 97% probability”



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

 Demo 

# Scalar UDF Inlining

## Automatisches Einbetten von skalaren UDF in Abfragen

```
CREATE OR ALTER FUNCTION [dbo].[fnDivScore]
    ( @Punkte INT, @Div INT )
RETURNS INT
WITH INLINE = ON -- Dies ist der Standard, OFF deaktiviert Inlining
-- Für die gesamte DB ausschalten: ALTER DATABASE SCOPED CONFIGURATION SET TSQL_SCALAR_UDF_INLINING = OFF;
AS
BEGIN
    RETURN @Punkte / @Div;
END

-- Abfrage mit UDF
SELECT [dbo].[fnDivScore](Punkte, 1000), * FROM [dbo].[Highscores];
-- wird intern zu: SELECT Punkte / 1000, * FROM [dbo].[Highscores];
```



<https://blogs.msdn.microsoft.com/sqlserverstorageengine/2018/11/07/introducing-scalar-udf-inlining/>



 Demo 



# Intelligent Database

In-Memory Database

# In-Memory Database

## Nicht Entwickler Features:

Memory-optimized TempDB metadata, Database Snapshots mit Memory-optimized filegroups, Hybrid Buffer Pool



The image is a composite of two photographs. The top photograph shows a dense forest of trees with light-colored, bare branches in the foreground and green foliage in the background. The bottom photograph shows a wide, stone-paved path or plaza, bordered by low stone walls, with green grass on the left. A dark blue horizontal band is superimposed over the middle of the image, containing white text.

# Intelligent Database

Intelligent performance

# Intelligent performance

- OPTIMIZE\_FOR\_SEQUENTIAL\_KEY
  - Optimierung für sequenzielle Indizes (Identity, Sequenzen, Datetime, ...)

## Weitere (nicht Entwickler Features):

Forcing fast forward and static cursors, Resource governance, Resource governance, Resource governance, Concurrent PFS updates, Scheduler worker migration



# OPTIMIZE\_FOR\_SEQUENTIAL\_KEY

Optimierung für sequenzielle Indizes (verhindert „last page insert contention“)

```
CREATE TABLE [dbo].[ErrorLog](  
[ErrorLogID] [int] IDENTITY(1,1) NOT NULL  
...  
CONSTRAINT [PK_ErrorLog_ErrorLogID] PRIMARY KEY CLUSTERED  
( [ErrorLogID] ASC )  
    WITH ( OPTIMIZE_FOR_SEQUENTIAL_KEY = ON )  
)
```



<https://techcommunity.microsoft.com/t5/SQL-Server/PAGELATCH-EX-waits-and-heavy-inserts/ba-p/384289>

 Demo 



# Graph Features



# Neue Graph Features

- CONNECT-Einschränkung
  - Einschränken, welche Nodes mit Edges verbunden werden können
- SHORTEST\_PATH-Prädikat
  - Kürzester Pfad zwischen zwei Nodes
- MERGE-Anweisung (Upsert)
  - Merge-Statement für Nodes & Edges

# Connect-Einschränkung

```
-- People dürfen People folgen
CREATE TABLE [dbo].[Follows](
    Since DateTime2 NOT NULL DEFAULT(GETDATE()),
    CONSTRAINT People_to_People CONNECTION ([dbo].[People] TO [dbo].[People])
)
AS EDGE;

-- People dürfen Pets besitzen (aber nicht umgekehrt)
CREATE TABLE [dbo].[Owns]
(
    [Since] DATETIME2 NOT NULL DEFAULT(GETDATE()),
    CONSTRAINT People_to_Pets CONNECTION ([dbo].[People] TO [dbo].[Pets])
) AS EDGE;
```



 Demo 

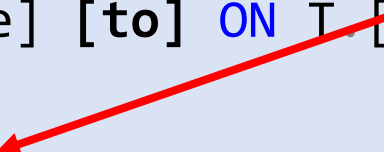
# SHORTEST\_PATH-Prädikat

```
SELECT
    pFrom.Id,
    pFrom.[Name]
    LAST_VALUE(pTo.[Name]) WITHIN GROUP (GRAPH PATH)
    STRING_AGG(pTo.[Name], '->') WITHIN GROUP (GRAPH PATH)
    COUNT(pTo.Id) WITHIN GROUP (GRAPH PATH)
    AS StartNode,
    AS FinalNode,
    AS [Edges Path],
    AS [Level]
FROM
    [dbo].[People] pFrom,
    [dbo].[People] FOR PATH pTo,
    [dbo].[Follows] FOR PATH follows
WHERE
    MATCH(SHORTEST_PATH(pFrom(-(follows)->pTo)+))
    -- '+': Maximale Pfadlänge, '{1,m}': Pfadtiefe 1 bis m
AND pFrom.[Name] = @form;
```

 Demo 

# Merge-Anweisung

```
MERGE [dbo].[Follows]
USING
(
    (SELECT * FROM @follower) AS T
    JOIN [dbo].[people] [from] ON T.[fromName] = [from].[Name]
    JOIN [dbo].[people] [to] ON T.[toName] = [to].[Name]
)
ON MATCH ([from]-(Follows)->[to])
WHEN NOT MATCHED THEN INSERT ($from_id, $to_id)
    VALUES ([from].$node_id, [to].$node_id)
WHEN NOT MATCHED BY SOURCE THEN DELETE
```



# Demo



A photograph of a forest scene with large, gnarled tree trunks and dense green foliage. A solid blue horizontal band is superimposed across the middle of the image, containing the text 'Java Unterstützung' in white. The background image shows a variety of trees, some with thick, textured bark and others with thinner, smoother bark, all set against a backdrop of lush green leaves.

# Java Unterstützung



# Java Unterstützung

SQL Server 2016: R

SQL Server 2017: Python

SQL Server 2019: Java

OS	Java Version
Windows	1.10
Linux	1.8

Datenaustausch (SQL Server => Java) via Array

# Java language programmability extensions

```
SQL Copy

DECLARE @myClassPath nvarchar(30)
DECLARE @param1 int

SET @myClassPath = N'<my path>/program.jar'
SET @param1 = 3

EXEC sp_execute_external_script
    @language = N'Java'
    , @script = N'<packageName>.<ClassName>.<methodName>'
    , @input_data_1 = N'<Input Query>'
    , @params = N'@CLASSPATH nvarchar(30), @param1 INT'
    , @CLASSPATH = @myClassPath
    , @param1 = @param1
```



<https://docs.microsoft.com/en-us/sql/advanced-analytics/java/extension-java?view=sqlallproducts-allversions>



<https://docs.microsoft.com/en-us/sql/advanced-analytics/java/howto-call-java-from-sql?view=sqlallproducts-allversions>

A photograph of a blue steam locomotive with orange wheels and a tall black smokestack, positioned on a gravel track. The locomotive is partially covered by a wooden canopy. In the background, there is a red building and a clear blue sky. A dark blue horizontal band is overlaid across the middle of the image, containing the text "SQL Server auf Linux" in white.

# SQL Server auf Linux

# Datenbank Replikation

- Replication Support
  - Transactional
  - Snapshot
  - Merge
- Rollen
  - Publisher
  - Distributor
  - Subscriber



# Sonstiges

- Microsoft Distributed Transaction Coordinator
- Always On-Verfügbarkeitsgruppe in Docker-Containern mit Kubernetes
- OpenLDAP-Unterstützung für AD-Drittanbieter
- Machine Learning unter Linux



Sicherheit



# SQL Server-Konfigurations-Manager

- SSL/TLS-Zertifikate verwalten
- Anzeigen und Überprüfen der auf einer SQL Server-Instanz installierten Zertifikate
- Anzeigen von bald ablaufenden Zertifikaten
- Bereitstellen von Zertifikaten auf Computern, die Always On-Verfügbarkeitsgruppen angehören (auf dem Knoten, auf dem sich das primäre Replikat befindet)
- Bereitstellen von Zertifikaten auf Computern, die einer Failoverclusterinstanz angehören (auf dem aktiven Knoten)



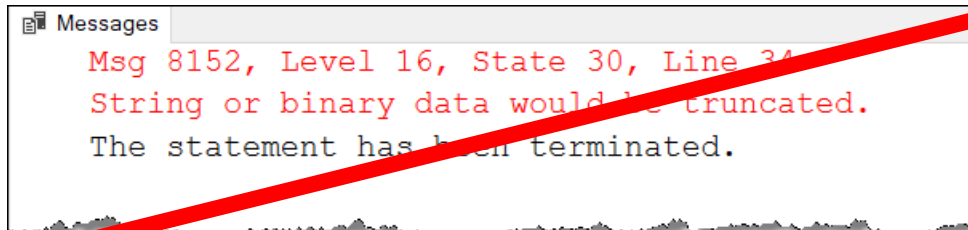
A photograph of three giraffes in a savanna landscape. The giraffes are standing in a grassy field, with a dense line of green bushes and trees in the background. The sky is blue with scattered white clouds. A semi-transparent blue horizontal band is overlaid across the middle of the image, containing the word "Sonstiges" in white text.

Sonstiges

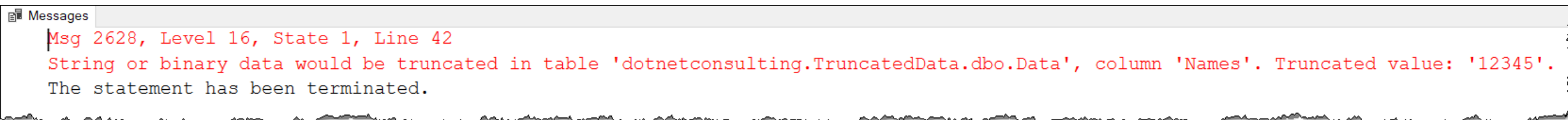
Endlich!!!

# Truncation error message

Brauchbare Meldung WO und WELCHE Daten abgeschnitten wurden



Msg 8152, Level 16, State 30, Line 34  
String or binary data would be truncated.  
The statement has been terminated.



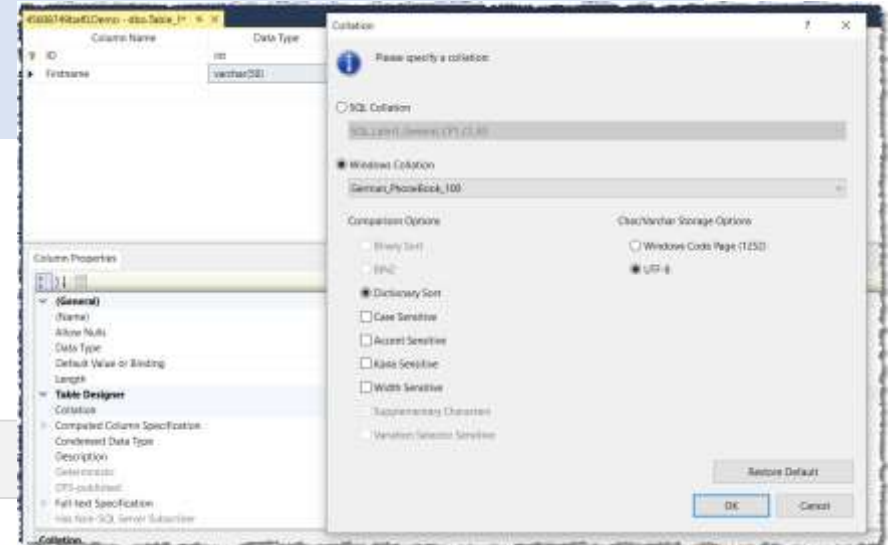
Msg 2628, Level 16, State 1, Line 42  
String or binary data would be truncated in table 'dotnetconsulting.TruncatedData.dbo.Data', column 'Names'. Truncated value: '12345'.  
The statement has been terminated.

 Demo 

# UTF-8 Unterstützung

Seit 2009 die meistverwendete Codierung im WWW

```
CREATE TABLE dbo.DemoTable  
(  
    ID int NOT NULL IDENTITY (1, 1),  
    Firstname varchar(50) COLLATE German_PhoneBook_100_CI_AI_SC_UTF8 NULL  
) ON [PRIMARY];
```



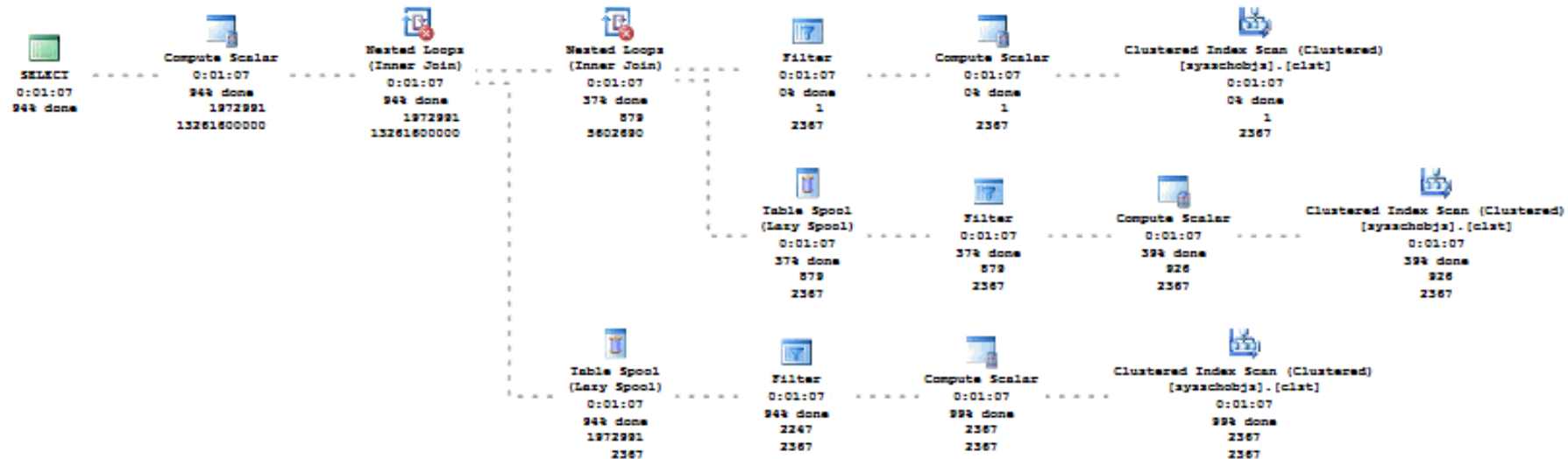
<https://en.wikipedia.org/wiki/UTF-8>

 Demo 



# Lightweight query profiling infrastructure

- Seit SQL Server 2014 SP2 // SQL Server 2016 SP1
  - Aktivierung via Trace Flag 7412
- 2% Overhead
- Ab SQL Server 2019 automatisch aktiv



 Demo 

# sys.dm\_db\_page\_info DMV

## Details über Seiten in der Datenbank abrufen

```
-- Pre SQL Server 2019
DBCC TRACEON(3604);
GO
DBCC PAGE(dotnetconsulting_Videogames, /* Database */ 1, /* Page */ 1, 0);

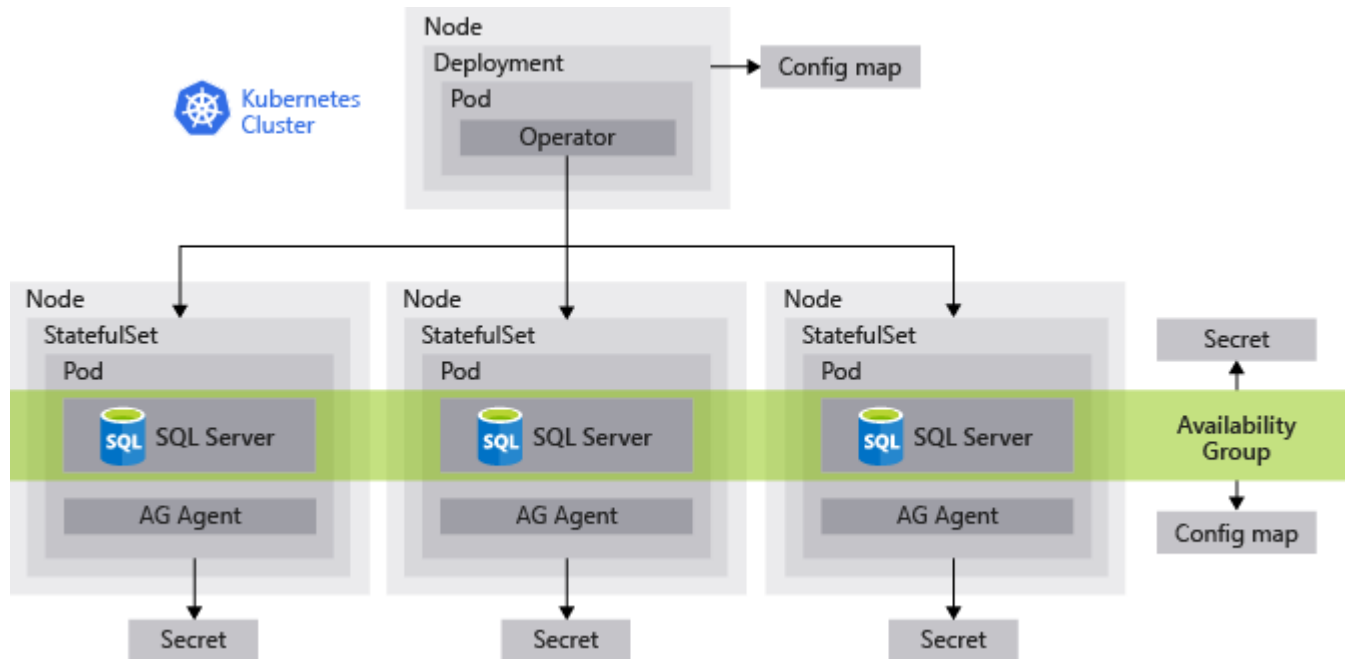
-- SQL Server 2019
SELECT * FROM sys.dm_db_page_info(/* Database */ 1, /* Page */ 1, 0, 'DETAILED');
```



<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-db-page-info-transact-sql?view=sqlallproducts-allversions>

# Hochverfügbarkeit

## High Availability & Disaster Recovery on containers in Kubernetes



<https://cloudblogs.microsoft.com/sqlserver/2018/12/10/availability-groups-on-kubernetes-in-sql-server-2019-preview/>

# Secure Enclaves

- Spezieller geschützter Speicherbereich (RAM)
  - Windows 10 Ent, Windows Server DataCenter, Windows Server 2019
- Always Encrypted
  - Ermöglicht mehr als Gleichheit (bei deterministic Encryption)
    - LIKE
    - Gleichheit (bei randomized Encryption)
  - In-place Encryption

Daten werden serverseitig in der Secure Enclave verarbeitet



<https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-enclaves?view=sqlallproducts-allversions>

# Fragen? Jetzt oder später!

## Kontakt

---

### E-Mail

[tkansy@dotnetconsulting.eu](mailto:tkansy@dotnetconsulting.eu)

### Telefon

+49 (0) 6187 / 2009090

### Microsoft Teams

[Meet now](#)

### LinkedIn

[Link me](#)

### XING

[Xing me](#)

### X (Twitter)

[@tkansy](#)





# www.dotnetconsulting.eu

SQL Server meets .NET (Core)- professionally!



Ich berate, coache und trainiere im Bereich Entwicklung von .NET (Core) Anwendungen mit Microsoft SQL Server- mit Allem, was dazu gehört- und was man vielleicht weglassen sollte.

