

Online on-demand Project Support



BASTA!
.NET, WINDOWS, JAVASCRIPT

Folien & Code auf
www.dotnetconsulting.eu/blog

Thorsten Kansy | dotnetconsulting.eu

SQL Server 2016

die spannenderen Neuerungen im Überblick



Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor








MCSE: Data Plattform & Business Intelligence

MCPD * MCTS * MCAD * MCSD

MCDBA * MCSE+I * MCT





Agenda I

- Dynamic Data Masking 
- Row Level Security 
- Always Encrypted 
- JSON Support 
- Temporal Table 

 Auch als Kurzvideo verfügbar

Agenda II

- SESSION_CONTEXT 
- DROP IF EXISTS/ CREATE OR ALTER
- Live Statistics 
- Stretch Database
- Reporting Services

 Auch als Kurzvideo verfügbar

Unser Plattformen

- SQL Server 2016
- SQL vNext (Linux)
- SQL Azure

Wenn SQL 2016, dann SP1!

Feature	RTM				SP1			
	Standard	Web	Express	Local DB	Standard	Web	Express	Local DB
Row level security	Yes	No	No	No	Yes	Yes	Yes	Yes
Dynamic Data Masking	Yes	No	No	No	Yes	Yes	Yes	Yes
Change data capture*	No	No	No	No	Yes	Yes	No*	No*
Database snapshot	No	No	No	No	Yes	Yes	Yes	Yes
Columnstore	No	No	No	No	Yes	Yes	Yes	Yes
Partitioning	No	No	No	No	Yes	Yes	Yes	Yes
Compression	No	No	No	No	Yes	Yes	Yes	Yes
In Memory OLTP	No	No	No	No	Yes	Yes	Yes	No**
Always Encrypted	No	No	No	No	Yes	Yes	Yes	Yes
PolyBase	No	No	No	No	Yes	Yes	Yes	No
Fine grained auditing	No	No	No	No	Yes	Yes	Yes	Yes
Multiple filestream containers	No	No	No	No	Yes	Yes	Yes	Yes



Row Level Security



Zeilenweiser, funktionsbasierter Filter

```
CREATE SECURITY POLICY SecurityFilter
  ADD FILTER PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter],
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] AFTER INSERT,
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] BEFORE DELETE,
  ADD BLOCK PREDICATE dbo.fn_AccessFilter([SecurityDescriptor]) ON
    [dbo].[Mitarbeiter] BEFORE UPDATE;
```



Dynamic Data Masking



Automatisch Anonymisierung

```
SELECT * FROM dbo.Mitarbeiter;
```



UNMASK?



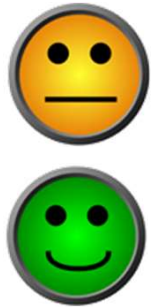
Results		Messages			
	ID	Name	Gehalt	Telefon	EMail
1	1	T-sy	695.54	xxxx	tXXX@XXXX.com
2	2	J-nd	863.30	xxxx	jXXX@XXXX.com
3	3	D-er	420.50	xxxx	dXXX@XXXX.com

	ID	Name	Gehalt	Telefon	EMail
1	1	Thorsten Kansy	999.90	1234567	tkansy@dotnetconsulting.eu
2	2	James Bond	12000.00	234567	jbond@dotnetconsulting.eu
3	3	Doc Snyder	9.99	3456789	dsnyder@dotnetconsulting...

Filter, Sortierungen, etc. funktionieren!



Always Encrypted



Verschlüsselung von Daten

- Auf den Massenspeichern des Servers
- Im Transit (Netzwerk Server ↔ Client)

- .NET 4.6

```
Data Source=.; Initial Catalog=dotnetconsulting;  
Integrated Security=true; Column Encryption  
Setting=enabled;
```



JSON Support

Integrierter JSON Support

```
SELECT TOP 3
    name, database_id, source_database_id
FROM sys.databases
ORDER BY database_id
FOR JSON PATH, ROOT('Databases'), INCLUDE_NULL_VALUES;
```

- FOR JSON-Klausel
- ISJSON-Funktion
- JSON_VALUE-Funktion
- OPENJSON-Funktion
- JSON_QUERY-Funktion

JSON Path als Abfragesprache (das XPATH des JSON)



Temporal Table

Kein Cascading Delete ☹



Automatisch Historisierung von Daten

```
CREATE TABLE dbo.Werte
(
    ID INT IDENTITY(1,1) NOT NULL,
    Wert1 NVARCHAR(10) NULL,
    Wert2 NVARCHAR(10) NULL,
    StartTime datetime2(7) GENERATED ALWAYS AS ROW START HIDDEN NOT NULL,
    EndTime datetime2(7) GENERATED ALWAYS AS ROW END HIDDEN NOT NULL,
    PERIOD FOR SYSTEM_TIME(StartTime, EndTime),
    CONSTRAINT Werte_PK PRIMARY KEY (ID)
)
WITH
(SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.Werte_History));

SELECT * FROM dbo.Werte
[FOR SYSTEM_TIME AS OF '2016-01-25 18:21:24.0738473'];
```



SESSION_CONTEXT



Verbindungsweites Dictionary

```
EXEC sp_set_session_context  
    @key = 'Department',  
    @value='Sales';
```

```
SELECT SESSION_CONTEXT(N'Department');
```

„Schalter“ für Trigger, Views, etc.



DROP IF EXISTS

Sorgloses löschen von Objekten

```
DROP <Typ> IF EXISTS <Name>;
```

```
DROP PROCEDURE IF EXISTS dbo.uspProc1;
```

```
DROP TABLE IF EXISTS dbo.table1;
```



CREATE OR ALTER

Anlegen oder modifizieren von Objekten

```
CREATE OR ALTER <Typ> <Name> AS ...;
```

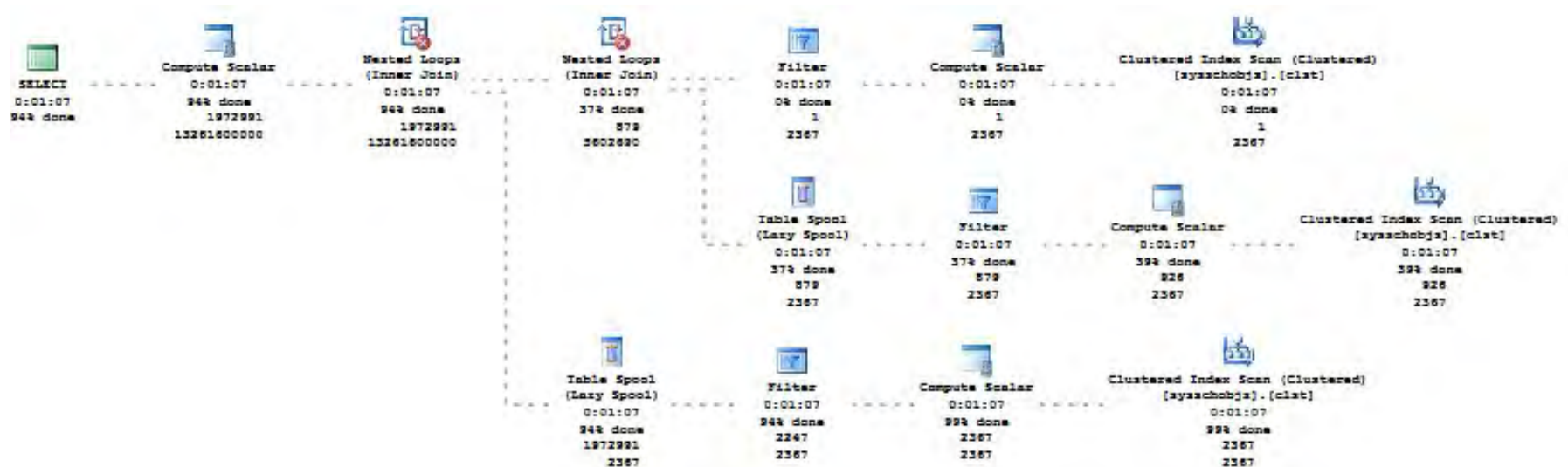
```
CREATE OR ALTER PROCEDURE dbo.uspProc1 AS ...;
```

```
CREATE OR ALTER dbo.view1 AS ...;
```



Live Statistics

Echtzeit Auswertung von Abfragen





Stretch Database

Historische Daten in Azure SQL
Datenbank auslagern

- Warm Storage (kein Cold Storage)
=> stehen für Abfragen bereit

On premise => Clustered Columnstore Index



In-Memory Verbesserungen

- ALTER TABLE
- Persistierte Memory Optimized Tables > 2 TB
- LOBs (`varbinary(max)`,
`[n]varchar(max)`) werden unterstützt
- DML Trigger
- Fremdschlüssel
- Check/Unique Constraints
- Transparent Data Encryption (TDE)

Reporting Services

- Drucken ohne ActiveX
- Benutzerdefinierter Parameterbereich



Drucken ohne ActiveX



Benutzerdefinierter Parameterbereich



Parameters

Category:	<input type="text"/>	Start Date:	<input type="text"/>
Subcategory:	<input type="text"/>	End Date:	<input type="text"/>

1 2 3 4 5 6

Parameters

Category:	<input type="text"/>	Subcategory:	<input type="text"/>
Start Date:	<input type="text"/>	End Date:	<input type="text"/>

1 2 3 4 5 6

- Parameter im Raster frei platzierbar

Fragen?

Links



<http://dotnetconsulting.eu/blog/>



@Tkansy



tkansy@dotnetconsulting.eu



www.dotnetconsulting.eu