Achtung! Schwertransporte im SQL Server!

Dateien im SQL Server mit FileStream & FileTable

SNEK Nürnberg, 22.03.2014
Johannes Curio
(Maic Beher)

1 / 25

Zu meiner Person

Johannes Curio

- MCT und MCITP für SQL Server
- Seit etwa 8 Jahren mit dem SQL Server unterwegs

Erreichbar unter
www.curio-consulting.de

2 / 25

Agenda

1. FileStream	• Verfügbar seit SQL Server 2008 (ab Express) • Warum FileStream? • Einrichten von FileStream • Nutzen von FileStream
2. FileTable	• Verfügbar seit SQL Server 2012 (ab Express) • Warum FileTable? • Einrichten von FileTable • Nutzen von FileTable • Berechtigungen

3 / 25

Datenbank und BLOBs?

- Eine Datenbank wurde konzipiert, um **Zeichen** zu speichern.

ID	Nachname	Vorname	Ort
1	Müller	Hans	München
2	Schmidt	Gabi	Köln

Zum schnelleren Verarbeiten werden die Zeichen zusammengefasst, im Bufferpool zu 8 KB-Seiten und in der MDF zu 64 KB-Blöcke.

- Welche Möglichkeit, externe **Dateien** (Dokumente, Videos, …), auch BLOBs (=Binary Large Object) genannt, in einem Feld zu speichern?

ID	Nachname	Vorname	Ort	Dokument
1	Müller	Hans	München	
2	Schmidt	Gabi	Köln	

? Profil_Meier.pdf
Profil_Schmidt.pdf

4 / 25

Dateien 1/3: Speichern in Datenbank

- Möglichkeit 1/3:
  Speichern der **Dateien** direkt in der Datenbank – direkt neben den Daten.

```
Insert into Kunden (Nachname, Vorname, Ort, Dokument)
Select 'Müller', 'Hans', …, * FROM OPENROWSET(BULK
N'X:\pdf\Beschrei_Spüle.pdf', SINGLE_BLOB) AS Dokument;
```

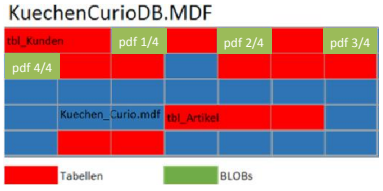
ID	Nachname	Vorname	Ort	Dokument
1	Müller	Hans	München	Profil_Meier.pdf
2	Schmidt	Gabi	Köln	Profil_Schmidt.pdf

Da der SQL Server die Dateien neben den Daten in seinen Tabellen verwaltet, sind die Dateien im transaktional-konsistenten Zustand.

5 / 25

Dateien 1/3: Speichern in Datenbank

- Möglichkeit 1/3:



Datei („Profil_Meier.pdf“) wird beim Insert auf jeden freien 64 KB-Block aufgeteilt und beim Select wieder zusammengesetzt werden incl. Aktualisieren der Systemtabellen. Performance?

6 / 25

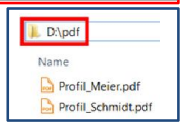
Dateien 2/3: Speichern im Dateisystem

- Möglichkeit 2/3:
  Speichern der **Dateien** im **Dateisystem** (z.B. NTFS)
  mit manueller Pfadangabe in der Datenbank

```
Insert into Kunden (Nachname, Vorname, Ort, Dok)  Values
('Müller', 'Hans', 'München', 'D:\pdf\Profil_Meier.pdf')
```

ID	Nachname	Vorname	Ort	Dokument
1	Müller	Hans	München	"D:\pdf\Profil_Meier.pdf"
2	Schmidt	Gabi	Köln	"D:\pdf\Profil_Schmidt.pdf"

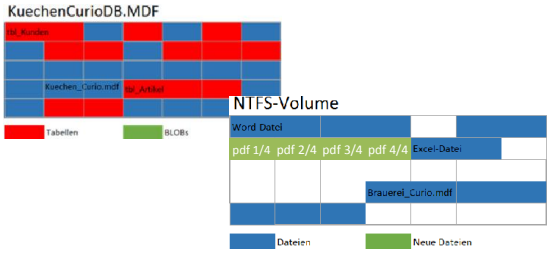
Die Datei befindet sich nicht in der Kontrolle des SQL Servers, sondern des OS. Somit sind die Dateien nicht in einem transaktional-konsistenten Zustand.



7 / 25

Dateien 2/3: Speichern im Dateisystem

- Möglichkeit 2/3:



Beim Speichern versucht NTFS die Datei („Profil_Meier.pdf") in einen zusammenhängenden Dateisystem-Bereich zu speichern.

8 / 25

Dateien 3/3: Speichern über FileStream

- Möglichkeit 3/3:
  Speichern der Dateien im Dateisystem; DB hat aber aktiv Kenntnis von der Datei über `FS_Spalte.PathName()`

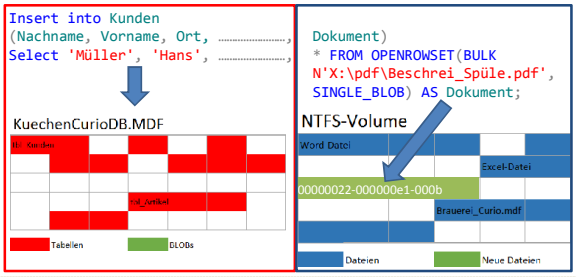
```
Insert into Kunden (Nachname, Vorname, Ort, Dokument)
Select 'Müller', 'Hans', …, * FROM OPENROWSET(BULK
N'X:\pdf\Beschrei_Spüle.pdf', SINGLE_BLOB) AS Dokument;
```

ID	Nachname	Vorname	Ort	Dokument
1	Müller	Hans	München	X:\KuechenCu … 0e1-000b
2	Schmidt	Gabi	Köln	X:\KuechenCu … 136-0003

9 / 25

Dateien 3/3: Speichern über FileStream

- Möglichkeit 3/3:
  Daten und Dateien sind im transaktional-konsistenten Zustand, da der SQL Server Kenntnisse



10 / 25

Fazit: Filestream

Vorteile durch FileStream: Das Beste aus beiden Welten!

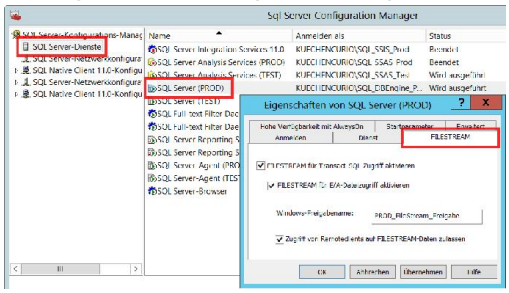
- Dateien unterliegen dem SQL Server, d.h.
  - dem Sicherheitskonzept des SQL Servers,
  - Sicherungsmechanismen des SQL Servers, …
- Dateien werden direkt im Dateisystem gespeichert ohne zusätzliche Zerlegung der Dateien in der MDF.

Vergleichspunkt	SQL Server	File Server	FileStream
Max. BLOB Größe	2 GB	Unbegrenzt (∞)	T-SQL -> 2 GB .Net -> ∞
Transaktional	Ja	Nein	Ja
Performance	Schlecht	Sehr Gut	Gut

11 / 25

Einrichten 1/4: Zugriff aktivieren

- Direkt bei der Installation der Instanz
- Nachträglich über den Konfiguration Manager

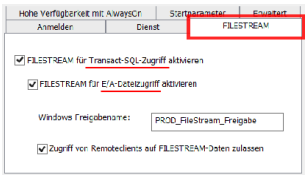


12 / 25

Einrichten 1/4: Sicherheitskonzept

Zugriffsrechte (Reiter „Sicherheit“ beim Windows-Ordner)

- 1. Haken: **Exklusiv**
  Nur der SQL Server darf per T-SQL auf die Dateien zugreifen.
- 2. Haken: **Shared**
  Andere, lokale Anwendungen dürfen auch z.B. per .Net auf die Dateien zugreifen.
- 3. Haken: **Shared**
  Andere, nicht lokale Anwendungen (z.B. Client) dürfen per .Net auf die Dateien zugreifen.

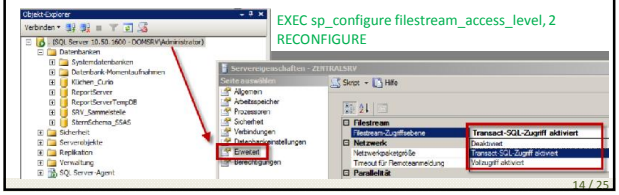


13 / 25

Einrichten 2/4: Instanz scharf schalten

4-Augen-Prinzip:

- Windows-Admin bestimmt durch Haken auf der vorherigen Folie, ob ein SQL Server auf das Dateisystem des Servers zugreifen darf.
- Zusätzlich legt der SQL Server-Admin fest, ob „seine“ Instanz das FileStream-Feature nutzen darf.

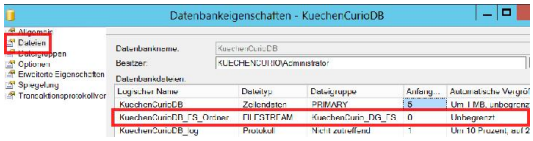



```
EXEC sp_configure filestream_access_level, 2
RECONFIGURE
```

14 / 25

Einrichten 3/4: Dateigruppe FileStream

Aktivierung des Dateityps „FileStream“

- 

- Skript

```
CREATE DATABASE [KuechenCurio] (
NAME=N'KuechCur', FILENAME=N'M:\KuechCur.mdf', SIZE=5120KB, …),
 FILEGROUP [KuechCur_DG_FS] CONTAINS FILESTREAM  DEFAULT
  (NAME = N'KuechCur_FS_Ordner', FILENAME =
   N'X:\KuechCurDB_Dateien\KuechCur_FS_Ordner' , MAXSIZE = UNLIMITED)
 LOG ON …
```

Ein Ordner!

15 / 25

Einrichten 4/4: Dateityp Varbinary(Max)

Erstellen einer Tabelle mit dem Datentyp Varbinary(Max)

```
CREATE TABLE [dbo].[Artikel]
(
[ID_Artikel] [int] IDENTITY PRIMARY KEY NOT NULL,
[Artikel_Name] [nvarchar](50) NULL,
[Artikel_Preis] [decimal](12, 3) NULL,
[Artikel_Dokument] [varbinary](max) FileStream NULL,
[Artikel_Dokument_ID]     [UNIQUEIDENTIFIER]
                          ROWGUIDCOL UNIQUE
                          DEFAULT (newid())  NOT NULL
 )
```

16 / 25

FileStream nutzen

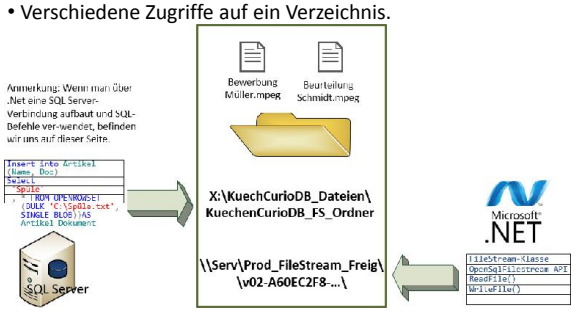
Einfügen eines Artikels mit einem pdf

```
Insert into Artikel
(Artikel_Name, Artikel_Preis, Artikel_Dokument)
Select
  'Schrank,
  , 250
  , * FROM OPENROWSET(BULK
    N'X:\FileStream_Vortrag\1_Beschreibung_2Schrank.pdf',
    SINGLE_BLOB) AS Artikel_Dokument;
```

17 / 25

FileStream – ein Ort

- Die Dateien werden an einem einzigen Ort gespeichert.
- Verschiedene Zugriffe auf ein Verzeichnis.



18 / 25

FileStream: Dateigröße & Kompatibilität

- Mehrzahl der Dateien > 1 MB → FileStream aktivieren. Mehrzahl der Dateien zwischen 256 KB und 1 MB, Test am konkreten System, ob FileStream Performancevorteile bringt. Dateien < 256 KB -> Speichern in DB ohne FileStream.
- FileStream kompatibel mit
  - Volltextsuche
  - Replikation
  - Cluster und Protokollversand und Always on
- FileStream nicht kompatibel mit
  - FileStream-Dateienwerden nicht verschlüsselt, auch dann nicht, wenn die transparente Datenverschlüsselung aktiviert ist.
  - Spiegelung
  - Datenbank-Momentaufnahmen

Quelle: http://technet.microsoft.com/de-de/library/bb895334.aspx

19 / 25

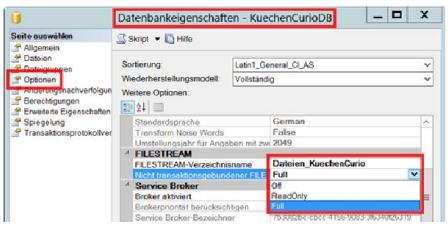
FileTable – Warum?

- Nachteil des **FileStream-Features**: Dateimanagement nur über T-SQL oder die Win32-API
- Hier setzt die FileTable an: Durch die FileTable-Technik können User die Dateien erstellen, ändern und löschen. Diese Modifikationen werden in speziellen Tabellen automatisch vom SQL Server mitprotokolliert.
- Unterscheidung:
  - FileStream ist eine Spalteneigenschaft
  - FileTable ist eine spezielle Tabelle
- FileTable braucht FileStream. Wenn FileStream deaktiviert, funktioniert FileTable nicht mehr.

20 / 25

FileTable Aktivieren

In den DB-Eigenschaften werden die Zugriffsrecht der User auf alle Dateien und Ordner der FileTable festgelegt - **nicht** die Zugriffsrechte des Systems auf die Dateien und Ordner.



21 / 25

FileTable erstellen

- Der passende Befehl

```
Create Table ProduktBeschreibung as Filetable
```

- Was kann ich nicht mitgeben? Die Spalten.
  Die sind intern festgelegt. Wenn man also zusätzliche Informationen speichern möchte, muss man eine 2 Tabelle anlegen und diese mit den zusätzlichen Infos füttern z.B. mit Hilfe eines Triggers.
- Trigger generell: Wenn eine Datei in das Verzeichnis geladen wird, kann der SQL Server auf verschiedene Dateioperationen reagieren.

22 / 25

FileTable nutzen

Demo

23 / 25

FileTable - Berechtigungen

- Berechtigungen auf FileTable-Ordner obliegt SQL Server
- Die Dateien sind in Zeilen gespeichert. Der SQL Server erlaubt keine Berechtigungen auf Zeilenebene
  ➔ Man kann standardmäßig keine Berechtigungen auf verschiedene Dateien / Ordner geben, wenn diese in der gleichen FileTable liegen.

24 / 25


Vielen Dank für Eure Aufmerksamkeit!
Kontakt für Feedback, Anregungen oder Fragen
Johannes Curio
www.curio-consulting.de
25 / 25