

15 Bilder und Dateien im SQL Server

Eines der großen Probleme von Access-Datenbanken ist der vergleichsweise geringe Speicherplatz. Sicher, wenn Sie nur reine Textinformationen oder Zahlen verwalten, kommen Sie damit eine Weile aus. Wenn die Datenbank aber auch noch Elemente wie Bilder oder Dateien aufnehmen soll, geraten Sie schnell ans Limit. Unter Access gibt es dabei die Möglichkeit, nur den Pfad zur jeweiligen Datei zu speichern und diese im Dateisystem zu belassen, aber damit sind einige Einschränkungen verbunden – zum Beispiel müssen Sie die Pfadangaben ändern, wenn die Datenbank samt referenzierter Dateien einmal verschoben werden soll und Sie müssen sich selbst darum kümmern, dass die Daten immer an Ort und Stelle bleiben und gesichert werden.

15.1 FILESTREAM

Mit SQL Server 2008 hat Microsoft ein neues Feature namens *FILESTREAM* eingeführt, mit dem Sie Daten über die Datenbank in einer bestimmten Verzeichnisstruktur ablegen und verwalten können. Mit dem SQL Server 2012 wurde dieses Feature noch ausgebaut – um die *FileTable*, die wir weiter unten betrachten. Wir schauen uns in diesem Kapitel zunächst an, wie der *FILESTREAM* funktioniert und wie Sie Dateien von Access aus speichern und diese wieder auslesen. Dabei bauen wir auf dem Stand von SQL Server 2012 auf.

15.1.1 FILESTREAM aktivieren

Bevor Sie *FILESTREAM* nutzen können, müssen Sie das Feature zunächst aktivieren. Dazu starten Sie den SQL Server-Konfigurations-Manager (Startmenü, *Alle Programme/SQL Server 2012/Konfigurationstools/SQL Server-Konfigurations-Manager*). Klicken Sie doppelt auf den Eintrag *SQL Server-Dienst* und dann mit der rechten Maustaste rechts auf *SQL Server (MSSQLSERVER)*. Wählen Sie den Eintrag *Eigenschaften* aus dem Kontextmenü aus. Im nun erscheinenden Dialog wechseln Sie zur Registerseite *FILESTREAM* (siehe Abbildung 15.1). Dort finden Sie folgende Optionen:

- » *FILESTREAM für Transact-SQL-Zugriff aktivieren*: Aktiviert den *FILESTREAM* auf SQL Server-Basis.
- » *FILESTREAM für E/A-Datenzugriff aktivieren*: Aktiviert *FILESTREAM* für den Zugriff über das Dateisystem.
- » *Windows-Freigabename*: Gibt den Namen des Verzeichnisses an, in dem die Dateien gespeichert werden.
- » *Zugriff von Remoteclients auf FILESTREAM-Daten zulassen*: Erlaubt den Zugriff von anderen Rechnern im Netzwerk.

Kapitel 15 Bilder und Dateien im SQL Server

Die hier vorgenommenen Einstellungen können Sie auch über das SQL Server Management Studio vornehmen. Dazu zeigen Sie die Eigenschaften des Eintrags für den SQL Server an und wechseln dort zur Seite *Erweitert*. Wenn Sie *FILESTREAM* für den Zugriff über das Dateisystem aktiviert haben, finden Sie im Windows Explorer eine entsprechende Freigabe vor (siehe Abbildung 15.2). Wozu diese benötigt wird, erfahren Sie weiter unten.

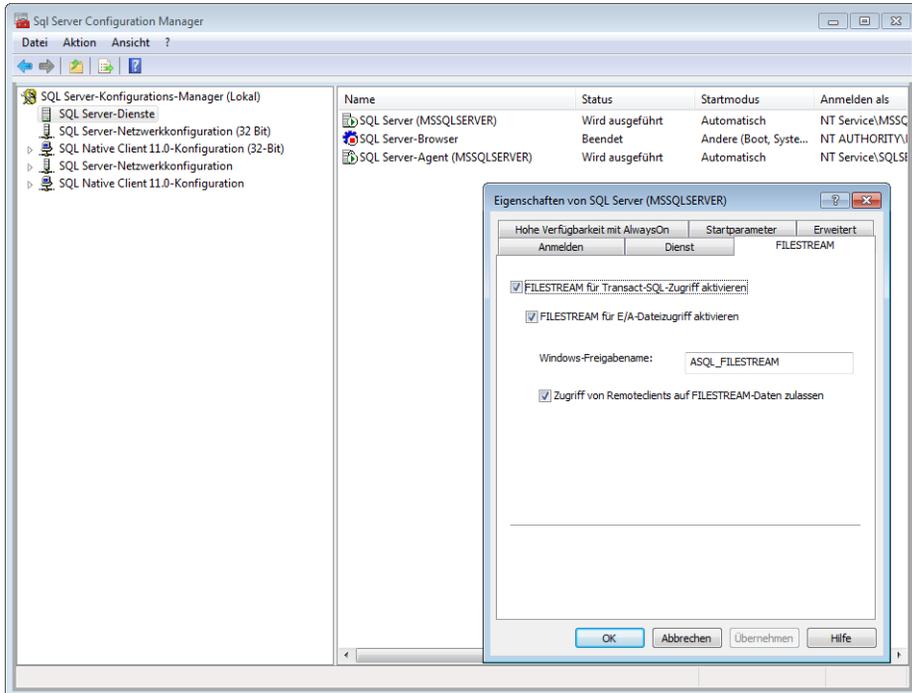


Abbildung 15.1: Aktivieren von *FILESTREAM*

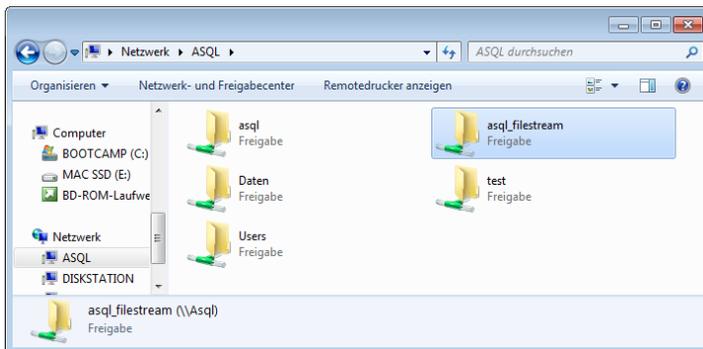


Abbildung 15.2: Freigabe für die über den SQL Server gespeicherten Dateien

Noch schneller erledigen Sie die Aufgabe allerdings mit den folgenden, nacheinander im Abfragefenster des SQL Server Management Studios abgesetzten Befehlen:

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

15.1.2 Datenbank für FILESTREAM erstellen

Nun schauen wir uns die notwendigen Arbeiten auf Seite der Datenbank an. Dazu legen wir zunächst eine neue Datenbank namens *ASQL_FILESERVER* an. Dies erledigen wir diesmal direkt per T-SQL, und zwar wie folgt:

```
CREATE DATABASE ASQL_FILESTREAM
ON
PRIMARY (NAME=ASQL_FILESTREAM_DB,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\FILESTREAM_DB.mdf'),
FILEGROUP ASQL_FILESTREAM_GROUP
CONTAINS FILESTREAM(NAME=FILESTREAM_FS,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\ASQL_FILESTREAMS')
LOG ON (NAME=ASQL_FILESTREAM_LOG,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\FILESTREAM_DB.ldf')
```

Der erste Teil der Anweisung erstellt die Daten-Datei und legt einen logischen Namen für die Dateigruppe fest. *CONTAINS FILESTREAM* legt fest, dass die Datenbank Dateien im Dateisystem speichert. Das dahinter mit *FILENAME* angegebene Verzeichnis wird ebenfalls erstellt.

Die neue Datenbank kommt im SQL Server Management Studio wie jede andere daher – kein Wunder: Die Möglichkeiten für den Umgang mit diesem Feature beschränken sich auch auf T-SQL, wie die folgenden Abschnitte zeigen werden.

15.1.3 Tabelle mit FILESTREAM-Feld erstellen

Eine Tabelle, die Dateien in einem *FILESTREAM*-Feld speichern soll, weist zwei Besonderheiten gegenüber herkömmlichen Tabellen auf:

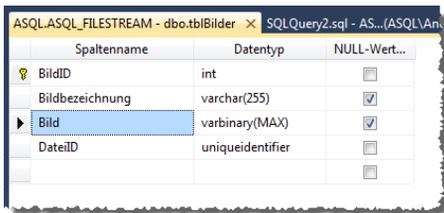
- » Das Feld für die *FILESTREAM*-Daten muss den Datentyp *varbinary(max)* aufweisen und mit dem Schlüsselwort *FILESTREAM* gekennzeichnet werden.
- » Die Tabelle muss ein Feld mit einer GUID enthalten (dazu können Sie ein zusätzliches Feld hinzufügen, wie im nachfolgenden Beispiel geschehen, oder das GUID-Feld direkt als Primärschlüsselfeld definieren).

Die folgende Anweisung erstellt eine Tabelle namens *tblBilder* mit den Feldern *BildID* (Primärschlüsselfeld), *Bildbezeichnung*, *Bild* (als *FILESTREAM*-Feld) und *DateiID* (als *GUID*-Feld mit *newid()* als Standardwert):

Kapitel 15 Bilder und Dateien im SQL Server

```
CREATE TABLE dbo.tblBilder (  
    BildID int IDENTITY(1,1) CONSTRAINT pkBildID PRIMARY KEY,  
    Bildbezeichnung varchar(255),  
    Bild varbinary(max) FILESTREAM,  
    DateiID UNIQUEIDENTIFIER ROWGUIDCOL DEFAULT newid()  
    NOT NULL CONSTRAINT ukDateiID UNIQUE  
);
```

Wenn Sie sich die Tabelle im Entwurf ansehen, finden Sie keinerlei Hinweis darauf, dass es sich bei Bild um ein *FILESTREAM*-Feld handelt (siehe Abbildung 15.3).

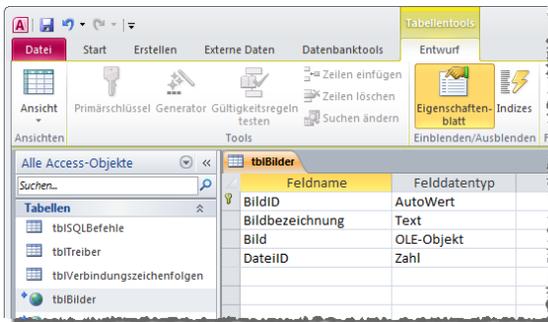


Spaltenname	Datentyp	NULL-Wert...
BildID	int	<input type="checkbox"/>
Bildbezeichnung	varchar(255)	<input checked="" type="checkbox"/>
Bild	varbinary(MAX)	<input checked="" type="checkbox"/>
DateiID	uniqueidentifier	<input type="checkbox"/>

Abbildung 15.3: Tabelle mit *FILESTREAM*-Feld

15.1.4 FILESTREAM-Feld unter Access

Nach dem Einbinden der Tabelle in eine Access-Datenbank liefert diese ebenfalls keine besonderen Eigenschaften – das *FILESTREAM*-Feld wird als herkömmliches Feld mit dem Datentyp *OLE-Objekt* angezeigt (siehe Abbildung 15.4).



Feldname	Felddatentyp
BildID	AutoWert
Bildbezeichnung	Text
Bild	OLE-Objekt
DateiID	Zahl

Abbildung 15.4: Unter Access eingebundene Tabelle mit *FILESTREAM*-Feld

Wenn Sie die Tabelle als Datenherkunft eines Formulars anzeigen, können Sie Dateien ganz einfach per Drag and Drop vom Windows Explorer in das OLE-Feld ziehen (siehe Abbildung 15.5). Das Bild wird dann allerdings als Symbol angezeigt und nicht als Bild. Ein Doppelklick auf die das OLE-Feld öffnet die Datei dann auch mit der dafür vorgesehenen Anwendung.

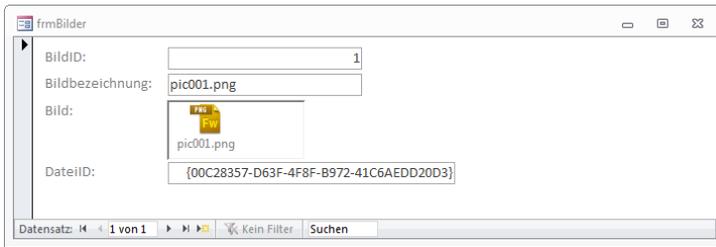


Abbildung 15.5: Formular mit OLE-Feld

Nun interessiert uns aber zunächst einmal, was nun mit der im Feld *Bild* gespeicherten Bilddatei geschehen ist. Die Abfrage des entsprechenden Datensatzes im Abfragefenster des SQL Server Management Studios liefert das Ergebnis aus Abbildung 15.6, also die binäre Darstellung der Datei.

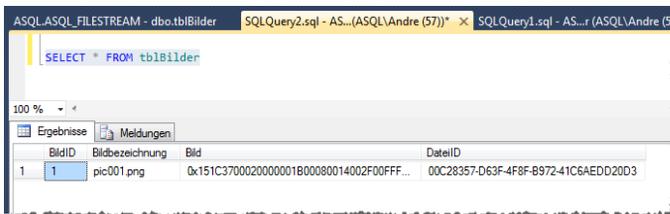


Abbildung 15.6: Bild als BLOB im SQL Server

Ein Blick in das für die *FILESTREAM*-Dateien angelegte Verzeichnis liefert auch nicht viel Erhellendes (siehe Abbildung 15.7). Dort befinden sich zwar einige Ordner, von denen einer auch eine Datei enthält, deren Speichergröße ungefähr mit der des gespeicherten Bildes übereinstimmt, aber öffnen lässt sich die Datei nicht – auch nicht nach dem Anfügen der Original-Dateiendung.

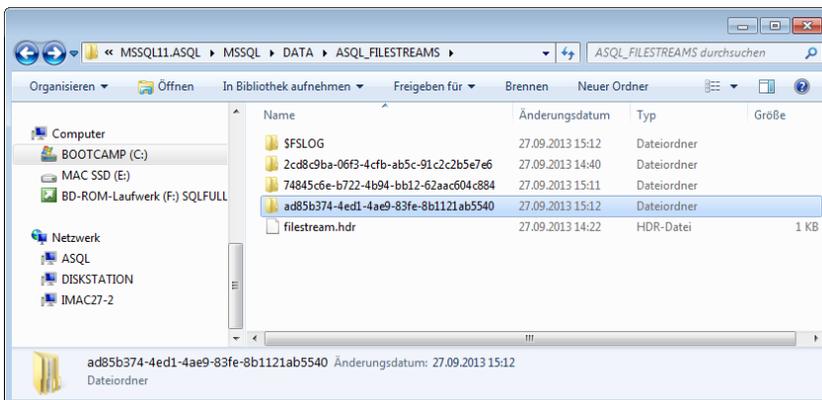


Abbildung 15.7: Verzeichnisse im *FILESTREAM*-Ordner

15.2 Tabellen mit FileTable

Die *FileTable*-Technik basiert auf *FILESTREAM*. Damit können Sie sowohl Dateien über die Datenbank speichern und diese dann über das Dateisystem öffnen als auch Dateien in dem für die Datenbank vorgesehenen Verzeichnis speichern und diese damit unter den Zugriff der SQL Server-Datenbank stellen.

15.2.1 FILESTREAM-Beispieldatenbank erstellen

Zu Beispielzwecken erstellen wir zunächst eine neue Datenbank, die mit normalem *FILESTREAM* ausgestattet ist:

```
CREATE DATABASE ASQL_FileTable
ON
PRIMARY (NAME=ASQL_FileTable_DB,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\FileTable_DB.mdf'),
FILEGROUP ASQL_FileTable_GROUP CONTAINS FILESTREAM(NAME=FileTable_FS,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\ASQL_FileTable')
LOG ON (NAME=ASQL_FileTable_LOG,
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.ASQL\MSSQL\DATA\FileTable_DB.ldf')
```

15.2.2 FILESTREAM-Verzeichnis festlegen

Weiter oben haben wir bei der Anpassung des SQL Servers für die Verwendung von *FILESTREAM* und *FileTable* bereits einen Windows-Freigabennamen festgelegt, unter dem die vom SQL Server verwalteten Dateien aufzufinden sein sollen.

Diese Freigabe wird von der kompletten SQL Server-Instanz genutzt, was bedeutet, dass nicht nur eine, sondern auch mehrere Datenbanken darauf zugreifen. Deshalb müssen Sie pro Datenbank noch ein Unterverzeichnis erstellen. Dieses legen Sie in den Eigenschaften der Datenbank auf der Seite *Optionen* unter *FILESTREAM* fest.

Stellen Sie für die Eigenschaft *FILESTREAM-Verzeichnisname* einen entsprechenden Wert wie etwa *ASQL_FileTable* ein und legen Sie mit der Eigenschaft *Nicht transaktionsgebundener FILESTREAM-Zugriff* fest, wie weit der Zugriff von außerhalb des SQL Servers zugelassen werden soll – also beispielsweise über den Windows Explorer (siehe Abbildung 15.8).

Das Verzeichnis können Sie auch mit folgender Anweisung festlegen:

```
ALTER DATABASE ASQL_FILETABLE SET FILESTREAM(DIRECTORY_NAME='ASQL_FILETABLE');
```

Und diese Anweisung stellt die Zugriffsart auf *FULL*, *READONLY* oder *OFF* ein:

```
ALTER DATABASE ASQL_FILETABLE SET FILESTREAM(NON_TRANSACTED_ACCESS=FULL);
```

Wenn Sie nun noch einmal den Windows Explorer bemühen, werden Sie unter der bereits vorhandenen Freigabe einen neuen Ordner vorfinden, der den angegebenen Namen trägt (siehe Abbildung 15.9).



Abbildung 15.8: FILESTREAM-Einstellungen für eine Datenbank

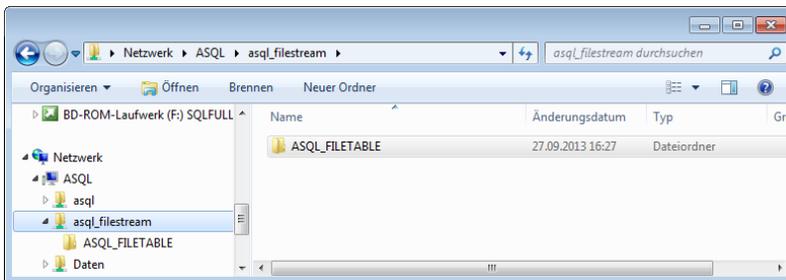


Abbildung 15.9: Ein neues Verzeichnis unterhalb der FILESTREAM-Freigabe

15.2.3 FileTable erstellen

Zum aktuellen Zeitpunkt können Sie allerdings noch keine Dateien über das Dateisystem in diesem Verzeichnis ablegen. Um dies zu ändern, müssen Sie eine Tabelle zum Speichern der Dateien erstellen. Für *FileTables* sieht der SQL Server einen eigenen Bereich im Objekt-Explorer vor, und zwar unterhalb des Eintrags *Tabellen*. Dort finden Sie im Kontextmenü auch die Möglichkeit, den grundlegenden Code zum Erstellen einer neuen *FileTable* in einer neuen Abfrage abzubilden (siehe Abbildung 15.10). Im Gegensatz zum FILESTREAM-Feld, das Sie einfach mit anderen Feldern in einer herkömmlichen Tabelle unterbringen können, ist die *FileTable* prinzipiell ein eigener Tabellentyp, deren Struktur bestimmten Vorgaben unterliegt. Eine einfache Anweisung zum Erstellen einer *FileTable* ist die folgende:

```
CREATE TABLE dbo.tb1FileTable AS FILETABLE
WITH
(FILETABLE_DIRECTORY = 'tb1FileTable', FILETABLE_COLLATE_FILENAME = database_default)
```

Kapitel 15 Bilder und Dateien im SQL Server

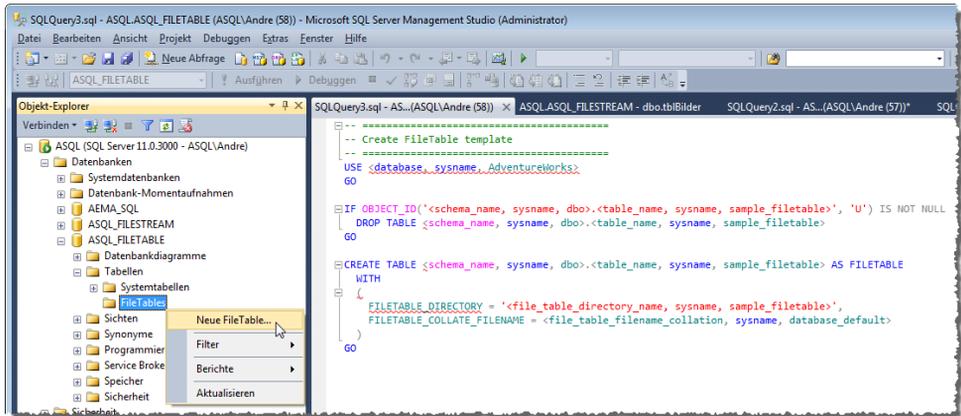


Abbildung 15.10: Erstellen einer neuen FileTable

Dabei geben Sie keine Feldnamen an, sondern nur ein paar Eigenschaften. Die erste hier abgebildete heißt `FILETABLE_DIRECTORY` und gibt an, in welchem Verzeichnis unterhalb des Verzeichnisses für die aktuelle Datenbank die Daten dieser Tabelle gespeichert werden sollen. Die zweite heißt `FILETABLE_COLLATE_FILENAME` und legt die Sortierreihenfolge für die Tabelle fest.

Nach dem Anlegen erscheint die Tabelle in dem für *FileTables* vorgesehenen Bereich im Objekt-Explorer (siehe Abbildung 15.10). Wie Sie dem Kontextmenü für diesen Eintrag entnehmen können, sehen die möglichen Aktionen etwas anders aus als für herkömmliche Tabelle – zum Beispiel fehlt die Möglichkeit zum Editieren der Tabelle.

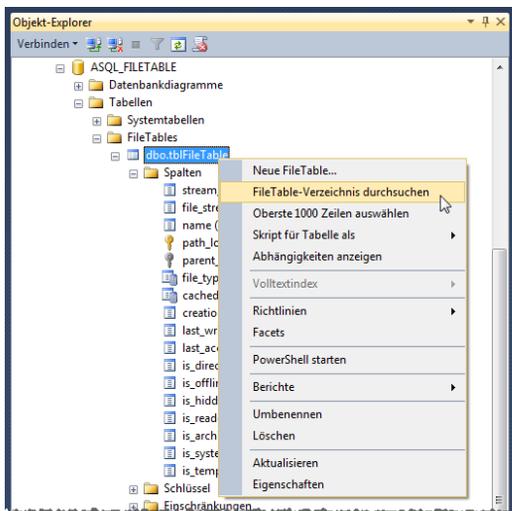


Abbildung 15.11: Die neue FileTable im Objekt-Explorer