

# Erratum



**Korrekturen zu „SQL Server 2017“ von Klemens Konopasek,  
ISBN 978-3-446-44826-1**

© 2018 Carl Hanser Verlag München



## HINWEIS

Durch einen technischen Fehler wurden die Tabellen in Kapitel 3 „Eine neue Datenbank erstellen“ nicht ins Buch übernommen.

Diese Tabellen werden hier zur Verfügung gestellt. Vor den Tabellen steht jeweils der einleitende Text, der auch im Buch enthalten ist. Somit ist erkennbar, zu welcher Stelle die Tabelle gehört.

Der Verlag entschuldigt sich für diesen Fehler und bittet die Leser um Verständnis.

## Fehlende Tabelle in Kapitel 3.1.1 „Bestandteile einer Datenbank“ Abschnitt „Datenbankdateien“

(Seite 106)

Eine Datenbank besteht aus Daten- und Transaktionsprotokolldateien. Eine Übersicht über diese Datenbankdateien liefert Ihnen die nachfolgende Tabelle.

**Tabelle 3.1** Dateien einer Datenbank

Datei	Typ	Beschreibung
Primäre Daten-dateien	MDF	Die primäre Datendatei (Master Data File) gibt es in jeder Datenbank. Bei Datenbanken von kleinerer und mittlerer Größe wird sie auch die einzige Datendatei sein. In dieser werden neben den Benutzerdaten-bankobjekten (Tabellen etc.) auch die Systemobjekte der Datenbank gespeichert. In den Systemtabellen werden zum Beispiel die ganze Struktur der Datenbank, deren Benutzer sowie alle Berechtigungen gespeichert. Die primäre Datendatei wird immer in der Dateigruppe PRIMARY gespeichert.
Weitere Daten-dateien	NDF	Bei großen Datenbanken können weitere Datendateien ergänzt werden, um Datenbankobjekte auf diese zu verteilen.
Transaktions-protokolldateien	LDF	Es können eine oder mehrere Transaktionsprotokolldateien für eine Datenbank festgelegt werden. Im Transaktionsprotokoll werden alle Schreibvorgänge in der Datenbank protokolliert. Diese Informationen dienen der Steuerung von Transaktionen. Fällt der Datenbankserver aus – zum Beispiel durch einen ungesicherten Stromausfall –, werden beim Neustart des Systems alle nicht abgeschlossenen Transaktionen automatisch zurückgesetzt. Des Weiteren wird das Transaktionsproto-koll für Backup- und Recovery-Vorgänge benötigt. (Mehr über die Bedeutung des Transaktionsprotokolls lesen Sie in Kapitel 5 zu den Transaktionen und in Kapitel 9 zum Thema Sichern und Wiederherstel-len von Datenbanken.)

**Fehlende Tabelle in Kapitel 3.2.1 „Tabellenfelder definieren“**

(Seite 124)

Für die Definition der Spalten stellt der SQL Server folgende Datentypen zur Verfügung:

**Tabelle 3.2** SQL Server-Datentypen

Kategorie	Datentyp	Beschreibung
Character	char(Länge) varchar(Länge) nchar(Länge) nvarchar(Länge) varchar(max) nvarchar(max)	Text mit fixer und variabler ( <i>var</i> ) Länge. Als maximale Länge können 8000 Zeichen definiert werden. Typen mit dem Präfix <i>n</i> (für national) verwenden Unicode und belegen den doppelten Speicherplatz. Die maximale Länge beträgt daher 4000 Zeichen. Die Typen mit dem fixen Parameter <i>max</i> können als sogenannte CLOBs (Character Large Objects) maximal 2 GB an Daten aufnehmen. Sie ersetzen den alten Datentyp <i>text</i> , der in der Verarbeitung nicht so flexibel ist.
Datum/ Uhrzeit	datetime smalldatetime date time(länge) datetime2(länge) datetimeoffset(länge)	<p>Datums- und Zeitangaben. Der Datentyp <i>datetime</i> reicht vom 01.01.1753 bis 31.12.9999 auf 3,33 Millisekunden genau und belegt 8 Byte. Der Datentyp <i>smalldatetime</i> kommt mit 4 Byte aus, reicht dafür aber mit Minuten-genauigkeit nur vom 01.01.1900 bis 06.06.2079.</p> <p>Seit dem SQL Server 2008 sind weitere Datentypen in dieser Kategorie verfügbar. Der Datentyp <i>date</i> bietet die Möglichkeit, einen reinen Datumswert ohne eine Uhrzeitkomponente zu verwenden. Er kann in all den Fällen verwendet werden, bei denen immer schon eigentlich nur das Datum benötigt worden ist und die Uhrzeit nur Ballast wäre. Er belegt nur 3 Byte – also weniger als der Datentyp <i>smalldatetime</i> – und hat einen wesentlich weiteren Geltungsbereich: vom 01.01.0001 bis 31.12.9999.</p> <p>Der Datentyp <i>time</i> enthält nur eine Zeit ohne ein Datum. Über den Längenparameter wird die Genauigkeit von Sekundenbruchteilen festgelegt. Dabei entspricht 0 ganzen Sekunden im Format <i>hh:mm:ss</i> und 7 der maximalen Genauigkeit von 100 Nanosekunden (1 Nanosekunde = <math>10^{-9}</math> Sekunden) mit dem Format <i>hh:mm:ss.nnnnnnn</i>. Der belegte Speicherplatz ist vom gewählten Längenparameter abhängig. Bei 0 – 2 werden 3 Byte belegt, bei 3 – 4 sind es 4 Byte und darüber werden 5 Byte benötigt. Wird der Parameter bei der Definition nicht angegeben, entspricht dies der Länge 7.</p> <p>Der Datentyp <i>datetime2</i> ist eine Erweiterung des Datentyps <i>datetime</i>. Der Datumsbereich entspricht dem von <i>date</i>, der enthaltene Zeitbereich dem von <i>time</i>; so als würde man einfach <i>date</i> und <i>time</i> zusammenfügen. Der Längenparameter entspricht eins zu eins dem von <i>time</i>. Auch der</p>

Kategorie	Datentyp	Beschreibung
		<p>Speicherplatzbedarf ergibt sich aus der Addition der beiden. Je nach Genauigkeit der Zeit ergibt sich daher ein Bedarf zwischen 6 und 8 Byte.</p> <p>Wenn Sie auch die Information zur jeweiligen Zeitzone speichern wollen, verwenden Sie den Datentyp <i>datetimeoffset</i>. Dieser entspricht dem um die Zeitzone ergänzten Datentyp <i>datetime2</i> und wird durch das Format <i>YYYY-MM-DD hh:mm:ss.nnnnnnn {+   -}hh:mm</i> dargestellt.</p>
Zahlen	decimal (Genauigkeit, Dezimalstellen) numeric (Genauigkeit, Dezimalstellen) float real bigint int smallint tinyint	<p><i>decimal</i> und <i>numeric</i> sind derselbe Datentyp. Ihre Größe wird durch die Genauigkeit in Stellen (maximal 38) und die darin enthaltenen Dezimalstellen angegeben.</p> <p><i>real</i> und <i>float</i> repräsentieren Gleitkommazahlen. Da nicht alle Werte im Bereich exakt dargestellt werden können, eignen sich diese Datentypen nicht für Primärschlüssel.</p> <p>Die Integer-Typen repräsentieren ganze Zahlen: <i>bigint</i> hat einen Wertebereich von <math>-2^{63}</math> bis <math>2^{63}</math> und benötigt dafür 8 Byte, <i>int</i> kann mit 4 Byte Platzbedarf einen Bereich von <math>-2.147.483.648</math> bis <math>2.147.483.647</math> abdecken. Für den Bereich von <math>-32.768</math> bis <math>32.767</math> kommen Sie mit <i>smallint</i> und 2 Byte je Zahl aus. Der „kleinste“ <i>tinyint</i> benötigt 1 Byte für die Abdeckung des Bereichs 0 bis 255.</p>
Währung	money smallmoney	<p>Währungen speichern Daten auf vier Nachkommastellen genau. Für kleinere Beträge können Sie <i>smallmoney</i> mit einem Wertebereich von <math>-214.748,3648</math> bis <math>214.748,3647</math> verwenden. Dafür werden 4 Byte benötigt. Ist dieser Wertebereich zu gering, müssen Sie <i>money</i> mit einem Speicherbedarf von 8 Byte verwenden. Dafür können Sie einen Bereich von circa <math>-922</math> Billionen bis <math>922</math> Billionen abbilden.</p>
Boolean	bit	<p>Dieser Datentyp kann die Werte <i>Wahr</i> (1), <i>Falsch</i> (0) und <i>NULL</i> darstellen.</p>
Large Objects	text ntext image	<p>BLOBs (Binary Large Objects) und CLOBs zur Speicherung von Daten bis zu 2 GB Größe. <i>Text/ntext</i> (CLOB) und <i>image</i> (BLOB) werden aus Gründen der Abwärtskompatibilität zwar noch unterstützt, sind aber mittlerweile durch <i>varchar(max)</i> und <i>varbinary(max)</i> ersetzt worden. Sie sind auch <i>deprecated</i>, was bedeutet, dass sie bei einer der nächsten Versionen nicht mehr unterstützt werden und daher nicht mehr verwendet werden sollten.</p>
Binär	binary(Länge) varbinary(Länge) varbinary(max)	<p>Datentypen zur Speicherung von Binärdaten mit maximal 8000 Byte. Der Datentyp <i>varbinary(max)</i> kann als BLOB bis zu <math>2^{31} - 1</math> Bytes Daten aufnehmen.</p>

Kategorie	Datentyp	Beschreibung
XML	xml	In diesem Datentyp können XML-Daten bis zu einer Größe von maximal 2 GB gespeichert werden. Diese können in XPath-Syntax durchsucht und an Strukturregeln gebunden werden.
Variant	sql_variant	Mit diesem Datentyp können unterschiedliche Daten wie zum Beispiel <i>varchar</i> oder <i>int</i> gespeichert werden. Er passt sich an den Inhalt an. Allerdings muss dieser Datentyp vor der Verwendung in einem Ausdruck explizit in einen passenden anderen Datentyp konvertiert werden. Zum Beispiel in einen <i>int</i> , um in einer mathematischen Berechnung verwendet werden zu können.
Räumlich	geography geometry	Der Datentyp <i>geography</i> dient dem Speichern von geografischen Daten der Erdkugel. Der Datentyp <i>geometry</i> ist in der Lage, Geometriedaten zu speichern, und kann ähnlich wie <i>geography</i> verwendet werden. Hier wird aber immer von der Ebene und nicht von der Ellipsenform der Erde ausgegangen.
Hierarchisch	hierarchyid	Der Datentyp <i>hierarchyid</i> wird zur Beschreibung von Positionen innerhalb einer Hierarchie verwendet. Der Aufbau der Hierarchie muss allerdings über die Anwendung definiert werden.

### Fehlende Tabelle in Kapitel 3.2.3 „Constraints“, Abschnitt „Gültigkeitsregeln“

(Seite 130)

Wir möchten in der Kundentabelle folgende Gültigkeitsregeln implementieren:

**Tabelle 3.3** Beispiele für Check-Einschränkungen

Regel	Ausdruck
Das Geschlecht darf die Werte 1 bis 5 enthalten.	<code>geschlecht BETWEEN 1 AND 5</code> oder zum Beispiel: <code>geschlecht &gt; 0 AND geschlecht &lt; 6</code> oder zum Beispiel: <code>geschlecht IN(1, 2, 3, 4, 5)</code>
Das Skonto darf nicht negativ sein und nicht über 5% liegen.	<code>skonto BETWEEN 0 AND 5</code> oder zum Beispiel: <code>skonto &gt;= 0 AND skonto &lt;= 5</code>
Die Skontotage dürfen nicht negativ sein und maximal 30 Tage ausmachen.	<code>skontotage BETWEEN 0 AND 30</code>

Regel	Ausdruck
Die E-Mail-Adresse muss gültig sein.	<pre>email LIKE '%_@%_.__' OR email LIKE '%_@%_.__' OR email LIKE '%_@%_.__'</pre>
Ist im Geschlecht <i>Herr/Frau/Familie</i> ausgewählt, müssen Nachname und Vorname erfasst werden. Ist <i>Firma/Sonstiges</i> eingetragen, muss die Firma auch ausgefüllt sein.	<pre>(geschlecht &lt;= 3 AND nachname IS NOT NULL AND vorname IS NOT NULL) OR (geschlecht &gt;= 4 AND firma1 IS NOT NULL)</pre>

### Fehlende Tabelle in Kapitel 3.5.5 „Objekte und Datenbanken skripten“ Abschnitt „Einzelne Objekte skripten“

(Seite 168)

Für alle Datenbankobjekte und die Datenbank selber besteht die Möglichkeit, direkt über das Kontextmenü das Skript zu erzeugen. Die entsprechende Anweisung ändert sich je nach Objekt, da der Name des Objekttyps in der Anweisung enthalten ist. Bei Tabellen lautet die Anweisung SKRIPT FÜR TABELLE ALS. Gefolgt wird die Anweisung von der nächsten Unteranweisung. Nicht alle dieser Subanweisungen stehen bei allen Objekttypen zur Verfügung. Eine Übersicht über alle Möglichkeiten bietet Ihnen Tabelle 3.4.

**Tabelle 3.4** Varianten für das Generieren von Skripten

Anweisung	Bedeutung
CREATE in	Anweisung, um das Objekt zu erstellen.
ALTER in	Bei Sichten, Prozeduren, Triggern und Funktionen ist dies die Basis, um Änderungen an dem jeweiligen Objekt vorzunehmen.
DROP in	Erzeugt eine Anweisung, um das Objekt zu löschen.
DROP und CREATE in	Diese Kombination erzeugt zwei Anweisungen, mit denen das Objekt zuerst gelöscht und danach neu erstellt wird.
SELECT in	Erzeugt für Tabellen und Sichten, aber auch für Skalarwert- und Tabellenwertfunktionen eine SELECT-Musteranweisung, um Daten auszugeben.
INSERT in	Damit kann eine Basisanweisung schnell erzeugt werden, mit der Daten in eine Tabelle oder eine Sicht eingefügt werden können.
UPDATE in	Mit der darüber erzeugten Anweisung bekommen Sie eine Vorlage für das Ändern von Daten in einer Tabelle oder Sicht.
DELETE in	Hiermit bekommen Sie eine Löschanweisung für Tabellen und Sichten.
EXECUTE in	Diese Anweisung wird für das Ausführen von gespeicherten Prozeduren benötigt.

## Fehlende Tabelle in Kapitel 3.6.2 „Objekte in einer FileTable speichern“, Abschnitt „Erstellen einer FileTable“

(Seite 184)

In der Spalte *name* erkennen Sie im Ergebnis die Namen der Dateien und Verzeichnisse. Die genauen Inhalte der Tabelle finden Sie in Tabelle 3.5 aufgeschlüsselt.

**Tabelle 3.5** Struktur einer FileTable

Spalte	Beschreibung
stream_id	Diese stellt, wie schon vom „normalen“ FILESTREAM her bekannt, eine eindeutige ID für jede Datei oder jeden Ordner bereit. Sie ist auch hier als UNIQUEIDENTIFIER ausgeformt.
file_stream	Diese Spalte repräsentiert die Inhalte von Dateien. Diese sind binär hier in einer <i>varbinary(max)</i> -Spalte verfügbar. Für Ordner enthält diese Spalte daher keinen Wert. Auch hier ist der Zusammenhang zu klassischem FILESTREAM zu erkennen.
name	Der Name der Datei oder des Ordners.
path_locator	Diese Spalte vom Datentyp <i>hierarchyid</i> repräsentiert den Primärschlüssel in einer FileTable. Er setzt sich aus den Werten der übergeordneten Pfade und denen der Datei zusammen und ist somit eindeutig.
parent_path_locator	Diese Spalte, ebenfalls vom Datentyp <i>hierarchyid</i> , stellt einen Verweis auf den übergeordneten Ordner dar. Er enthält den Eintrag aus der Spalte <i>path_locator</i> des Ordners, in dem sich die Datei oder das Verzeichnis befindet. Über einen Fremdschlüssel wird hier die Konsistenz gesichert
file_type	Enthält die Dateierweiterung bei Dateien, wie zum Beispiel <i>mp3</i> oder <i>m4a</i> .
cached_file_size	Diese Spalte ist als <i>bigint</i> ausgeformt und zeigt bei Dateien die Größe in Bytes an.
creation_time	Diese Spalte zeigt an, wann die Datei erstellt beziehungsweise in den Ordner kopiert worden ist. Dies ist nicht der Zeitpunkt, den Sie im Explorer als <i>Änderungsdatum</i> angezeigt bekommen. Diese Logik entspricht dem bekannten Verhalten, wenn Sie eine Datei in einen gewöhnlichen Ordner kopieren. Sie sehen denselben Zeitpunkt als <i>Erstellt</i> , wenn Sie sich im Explorer die Eigenschaften der Datei anzeigen lassen. Für diese Spalte wird der Datentyp <i>datetimeoffset</i> verwendet, der auch die Zeitzone mit speichert und anzeigt.
last_write_time	Dieser Wert entspricht dem, der im Explorer als <i>Änderungsdatum</i> angezeigt wird. Sie ist ebenfalls mit dem Datentyp <i>datetimeoffset</i> definiert.
last_access_time	In den Eigenschaften einer Datei finden Sie den Eintrag <i>Letzter Zugriff</i> . In der FileTable finden Sie diesen Wert in dieser Spalte. Wie die beiden zuvor hat diese den Datentyp <i>datetimeoffset</i> .
is_directory	Diese bit-Spalte enthält für Ordner den Eintrag <i>Wahr</i> (1) und für Dateien den Wert <i>Falsch</i> (0).

Spalte	Beschreibung
is_offline	Wird dazu verwendet, um das Dateiattribut <i>Offline</i> innerhalb der FileTable anzuzeigen.
is_hidden	Dieser Wert entspricht dem Dateiattribut <i>Versteckt</i> .
is_readonly	Das Dateiattribut <i>Schreibgeschützt</i> wird in dieser Spalte abgebildet.
is_archive	Das Dateiattribut <i>Archiv</i> wird von dieser Spalte dargestellt.
is_system	Enthält den Wert 1 für <i>Wahr</i> , wenn eine Datei oder ein Ordner als Systemdatei beziehungsweise als Systemordner markiert ist.
is_temporary	Übernimmt das Dateiattribut <i>Temporär</i> .

### Fehlende Tabelle in Kapitel 3.8.3 „Index für speicheroptimierte Tabellen“, Abschnitt „Nicht gruppierter Hash-Index“ (Seite 204)

Die aus unserer Beispieltabelle gelieferten Werte sind nicht wirklich realitätsnah, da in vielen Tabellen eine nur sehr geringe Anzahl an Beispieldaten enthalten ist.

Die Inhalte dieser Ausgabe und die sich daraus ableitbaren Aussagen sind in Tabelle 3.6 zusammengefasst.

**Tabelle 3.6** Statistikinformationen über Hash-Indizes

Ergebnisspalte	Inhalt/Aussage
tabelle	Der Name der betroffenen Tabelle, der mit der Funktion <code>OBJECT_NAME()</code> aus der <i>object_id</i> derselben gebildet wird.
indexname	Der Name des betroffenen Hash-Index.
total_bucket_count	Die Bucketanzahl, die beim Erstellen des Hash-Index angegeben wurde.
empty_bucket_count	Die Anzahl der noch freien Buckets.
empty_bucket_percent	Der Prozentsatz der noch freien Buckets, der aus den beiden vorherigen Werten berechnet ist. Liegt dieser Prozentsatz unter 10 Prozent, ist die aktuelle Bucketanzahl wahrscheinlich zu gering. Es sollten immer mehr als 33 Prozent leer sein, da dieser Wert aufgrund der Hashverteilung erreicht wird, wenn die Anzahl der Buckets der Anzahl der eindeutigen Werte im Index entspricht.
avg_chain_length	Diese Spalte zeigt die durchschnittliche Kettenlänge der Zeilenkette an. Es kommt zur Kettenbildung, wenn idente Spaltenwerte denselben Hashwert liefern oder wenn sich mehrere Hashwerte Buckets teilen müssen, weil zu wenige vorhanden sind. Werden diese Ketten zu lang, beeinflusst das die Performance negativ. Ideal ist hier ein Wert von 1 für eindeutige Indizes, steigt dieser Wert auf über 5-10, obwohl noch genügend Buckets frei sind, ist an dieser Stelle ein nicht gruppierter Index vorzuziehen.
max_chain_length	Zeigt die maximale Kettenlänge an.