

Dirk Angermann

# T-SQL-Abfragen für Microsoft SQL Server 2022



# Inhaltsverzeichnis

<b>Einleitung</b> .....	<b>11</b>
Aufbau des Buches .....	11
Begleitmaterial .....	12
Danksagung .....	19
Über den Autor .....	19
<b>1 Arbeiten mit dem SQL Server Management Studio</b> .....	<b>21</b>
1.1 Starten von SQL Server Management Studio .....	21
1.2 Herstellen einer Verbindung mit SQL Server .....	21
1.3 Einstellungen im SQL Server Management Studio .....	22
1.4 Arbeiten mit Skriptdateien und Projekten .....	34
<b>2 Einführung in T-SQL-Abfragen</b> .....	<b>39</b>
2.1 Einführung in T-SQL .....	39
2.1.1 Über T-SQL .....	39
2.1.2 Kategorien von T-SQL-Anweisungen .....	41
2.1.3 T-SQL-Sprachelemente .....	41
2.2 Grundlegendes zu Mengen .....	48
2.2.1 Mengenlehre und SQL Server .....	48
2.2.2 Auf SQL Server-Abfragen angewendete Mengenlehre .....	50
2.3 Grundlegendes zur Prädikatlogik .....	50
2.3.1 Prädikatlogik und SQL Server .....	50
2.3.2 Auf SQL Server angewendete Prädikatlogik .....	51
2.4 Grundlegendes zur logischen Reihenfolge der Vorgänge in SELECT-Anweisungen .....	51
2.4.1 Elemente einer SELECT-Anweisung .....	52
2.4.2 Logische Abfrageverarbeitung .....	52
2.4.3 Anwenden der logischen Reihenfolge von Vorgängen zum Schreiben von SELECT-Anweisungen .....	54
<b>3 Schreiben von SELECT-Abfragen</b> .....	<b>55</b>
3.1 Übersicht .....	55
3.1.1 Schreiben von einfachen SELECT-Anweisungen .....	55
3.1.2 Entfernen von Duplikaten mit DISTINCT .....	63
3.1.3 Verwenden von Spaltenaliasen .....	64
3.1.4 Schreiben von einfachen CASE-Ausdrücken .....	67

## Inhaltsverzeichnis

<b>4</b>	<b>Filtern von Daten</b>	<b>71</b>
4.1	Filtern von Daten in der WHERE-Klausel	71
4.2	Arbeiten mit unbekannten Werten	77
<b>5</b>	<b>Gruppieren und Aggregieren von Daten</b>	<b>79</b>
5.1	Verwenden von Aggregatfunktionen	79
5.2	Einsatz von DISTINCT und ALL	82
5.3	Verwenden der GROUP BY-Klausel	84
5.4	Filtern von Gruppen mit HAVING	88
<b>6</b>	<b>Daten sortieren</b>	<b>91</b>
6.1	Sortieren von Daten	91
6.2	Filtern von Daten mithilfe der TOP-Option	93
6.3	Filtern von Daten mithilfe der OFFSET FETCH-Option	96
<b>7</b>	<b>Arbeiten mit SQL Server-Datentypen</b>	<b>99</b>
7.1	Einführung in SQL Server-Datentypen	99
7.1.1	SQL Server-Datentypen	99
7.1.2	Numerische Datentypen	100
7.1.3	Datentypen der Binärzeichenfolge	102
7.1.4	Sonstige Datentypen	102
7.1.5	Datentypenrangfolge	103
7.1.6	Wann werden Dateitypen konvertiert?	104
7.2	Arbeiten mit Zeichendaten	106
7.2.1	Zeichendatentypen	106
7.2.2	Sortierung	107
7.2.3	Zeichenfolgenverkettungen	108
7.2.4	Zeichenfolgenfunktionen	110
7.2.5	Arbeiten mit Datum- und Uhrzeitdaten	114
7.3	Datums- und Uhrzeitdatentypen	115
7.3.1	Datums- und Uhrzeitdatentypen: Literale	115
7.3.2	Getrenntes Arbeiten mit Datum und Uhrzeit	116
7.3.3	Abfragen von Datums- und Uhrzeitwerten	117
7.3.4	Datums- und Uhrzeitfunktionen	117
<b>8</b>	<b>Verwenden von integrierten Funktionen</b>	<b>123</b>
8.1	Schreiben von Abfragen mit integrierten Funktionen	123
8.1.1	Typen von in SQL Server integrierten Funktionen	123
8.1.2	Skalarfunktionen	124
8.1.3	Aggregatfunktionen	126
8.1.4	Fensterfunktionen	126
8.1.5	Rowsetfunktionen	127

8.2	Verwenden von Konvertierungsfunktionen . . . . .	128
8.2.1	Implizite und explizite Datentypenkonvertierung . . . . .	128
8.2.2	Konvertierung mit CAST . . . . .	129
8.2.3	Konvertierung mit CONVERT . . . . .	130
8.2.4	Konvertieren von Zeichenfolgen mit PARSE . . . . .	132
8.2.5	Konvertieren mit TRY_PARSE und TRY_CONVERT . . . . .	133
8.3	Verwenden von logischen Funktionen . . . . .	135
8.3.1	Schreiben von logischen Tests mit Funktionen . . . . .	135
8.3.2	Ausführen von Bedingungsüberprüfungen mit IIF . . . . .	136
8.3.3	Auswählen von Elementen aus einer Liste mit CHOOSE . . . . .	137
8.4	Verwenden von Funktionen, um mit NULL zu arbeiten . . . . .	138
8.4.1	Konvertieren von NULL mit ISNULL . . . . .	138
8.4.2	Verwenden von COALESCE, um Nicht-NULL-Werte zurückzugeben . . . . .	139
8.4.3	Verwenden von NULLIF, um NULL zurückzugeben, wenn Werte zusammenpassen . . . . .	140
9	<b>Verwenden von Unterabfragen</b> . . . . .	141
9.1	Übersicht . . . . .	141
9.2	Schreiben von eigenständigen Unterabfragen . . . . .	142
9.2.1	Schreiben von mehrwertigen Unterabfragen . . . . .	144
9.2.2	Schreiben von korrelierten Unterabfragen . . . . .	148
9.2.3	Verwenden von EXISTS-Prädikat mit Unterabfragen . . . . .	150
10	<b>Abfragen mehrerer Tabellen</b> . . . . .	153
10.1	Grundlegendes zu Joins . . . . .	153
10.1.1	Das kartesische Produkt – CROSS JOIN . . . . .	153
10.1.2	Abfragen mit INNER JOIN . . . . .	155
10.1.3	Abfrage mit SELF JOIN . . . . .	163
10.1.4	Abfragen mit OUTER JOIN . . . . .	164
11	<b>Verwenden von Tabellenwertausdrücken</b> . . . . .	169
11.1	Verwenden von Sichten . . . . .	169
11.1.1	Schreiben von Abfragen, die Ergebnisse von Sichten zurückgeben . . . . .	169
11.1.2	Erstellen einfacher Sichten . . . . .	171
11.2	Verwenden von Inline-Tabellenwertfunktionen . . . . .	173
11.2.1	Schreiben von Abfragen, die Inline-Tabellenwertfunktionen verwenden . . . . .	173
11.2.2	Erstellen von einfachen Inline-Tabellenwertfunktionen . . . . .	173
11.2.3	Abrufen von Inline-Tabellenwertfunktionen . . . . .	174

## Inhaltsverzeichnis

11.3	Verwenden von abgeleiteten Tabellen . . . . .	175
11.3.1	Schreiben von Abfragen mit abgeleiteten Tabellen . . . . .	175
11.3.2	Richtlinien für abgeleitete Tabellen . . . . .	177
11.3.3	Verwenden von Aliassen für Spaltennamen in abgeleiteten Tabellen . . . . .	177
11.4	Verwenden allgemeiner Tabellenausdrücke . . . . .	179
11.4.1	Schreiben von Abfragen mit allgemeinen Tabellenausdrücken . . . . .	179
11.4.2	Erstellen und Abfragen von allgemeinen Tabellenausdrücken . . . . .	180
<b>12</b>	<b>Verwenden von Mengenoperatoren . . . . .</b>	<b>183</b>
12.1	Schreiben von Abfragen mit dem UNION-Operator . . . . .	183
12.1.1	Interaktion zwischen Mengen . . . . .	183
12.1.2	Verwenden des UNION-Operators . . . . .	184
12.1.3	Verwenden des UNION ALL-Operators . . . . .	186
12.2	Verwenden von EXCEPT und INTERSECT . . . . .	186
12.2.1	Verwenden des INTERSECT-Operators . . . . .	186
12.2.2	Verwenden des EXCEPT-Operators . . . . .	187
<b>13</b>	<b>Verwenden von Fensterrangfolge-, Offset- und Aggregatfunktionen . . . . .</b>	<b>189</b>
13.1	Erstellen von Fenstern mit OVER . . . . .	189
13.1.1	SQL-Fensterfunktionen . . . . .	189
13.1.2	Fensterkomponenten . . . . .	190
13.1.3	Verwenden von OVER . . . . .	190
13.1.4	Partitionieren von Fenstern . . . . .	191
13.2	Kennenlernen von Fensterfunktionen . . . . .	192
13.2.1	Fensterfunktionen definieren . . . . .	193
13.2.2	Fensteraggregatfunktionen . . . . .	193
13.2.3	Fensterrangfolgefunktionen . . . . .	194
13.2.4	Fensterverteilungsfunktionen . . . . .	202
13.2.5	Fensteroffsetfunktionen . . . . .	203
<b>14</b>	<b>Pivotieren und Groupingsets . . . . .</b>	<b>205</b>
14.1	Schreiben von Abfragen mit PIVOT und UNPIVOT . . . . .	205
14.1.1	Was ist Pivotieren? . . . . .	205
14.1.2	PIVOT-Elemente . . . . .	207
14.1.3	Schreiben von Abfragen mit UNPIVOT . . . . .	208
14.2	Arbeiten mit Groupingsets . . . . .	210
14.2.1	Schreiben von Abfragen mit Groupingsets . . . . .	210
14.2.2	CUBE und ROLLUP . . . . .	213

<b>15</b>	<b>Abfragen von SQL Server-Metadaten</b>	217
15.1	Abfragen von Systemkatalogsichten und Systemfunktionen	217
	15.1.1 Systemkatalogsichten	217
	15.1.2 Informationsschemasichten	219
	15.1.3 Systemmetadatenfunktionen	220
15.2	Ausführen von gespeicherten Systemprozeduren	222
	15.2.1 Ausführen von gespeicherten Prozeduren	222
	15.2.2 Ausführen von gespeicherten Systemprozeduren	223
	15.2.3 Häufig verwendete gespeicherte Systemprozeduren	224
15.3	Abfragen von dynamischen Verwaltungssichten	224
	15.3.1 Dynamische Verwaltungsobjekte	224
	15.3.2 Kategorisieren von dynamischen Verwaltungssichten	225
	15.3.3 Abfragen von dynamischen Verwaltungssichten und -funktionen	226
<b>16</b>	<b>Ausführen von gespeicherten Prozeduren</b>	229
16.1	Abfragen von Daten mit gespeicherten Prozeduren	229
	16.1.1 Untersuchen von gespeicherten Prozeduren	229
	16.1.2 Ausführen von gespeicherten Prozeduren	230
16.2	Übergeben von Parametern an gespeicherte Prozeduren	231
	16.2.1 Übergeben von Eingabeparametern	231
	16.2.2 Arbeiten mit OUTPUT-Parametern	231
16.3	Erstellen von einfachen gespeicherten Prozeduren	232
	16.3.1 Erstellen von Prozeduren, die Zeilen übergeben	232
	16.3.2 Erstellen von Prozeduren, die Parameter annehmen	234
16.4	Arbeiten mit dynamischem SQL	236
	16.4.1 Erstellen von dynamischem SQL	236
	16.4.2 Schreiben von Abfragen mit dynamischem SQL	237
<b>17</b>	<b>Programmieren mit T-SQL</b>	239
17.1	T-SQL-Programmierelemente	239
	17.1.1 Einführung in T-SQL-Batches	239
	17.1.2 Arbeiten mit Batches	240
	17.1.3 Einführung in T-SQL-Variablen	240
	17.1.4 Arbeiten mit Variablen	242
	17.1.5 Arbeiten mit Synonymen	243
17.2	Steuern des Programmflusses	244
	17.2.1 T-SQL-Sprache zur Ablaufsteuerung	244
	17.2.2 Arbeiten mit IF ... ELSE	244
	17.2.3 Arbeiten mit WHILE	246

<b>18</b>	<b>Implementieren der Fehlerbehandlung</b>	249
18.1	Verwenden von TRY/CATCH-Blöcken	249
18.1.1	Strukturierte Ausnahmebehandlung	249
18.1.2	Erstellen von TRY- und CATCH-Blöcken	250
18.2	Arbeiten mit Fehlerinformationen	251
18.2.1	Abfragen eines ERROR-Objekts	251
18.2.2	Verwenden der THROW-Anweisung	252
<b>19</b>	<b>Implementieren von Transaktionen</b>	255
19.1	Transaktionen und das Datenbankmodul	255
19.1.1	Definieren von Transaktionen	255
19.1.2	Die Notwendigkeit von Transaktionen: Probleme mit Batches	256
19.1.3	Transaktionen erweitern Batches	256
19.2	Steuern von Transaktionen	258
19.2.1	BEGIN TRANSACTION	258
19.2.2	COMMIT TRANSACTION	259
19.2.3	ROLLBACK TRANSACTION	259
<b>20</b>	<b>Verbessern der Abfrageleistung</b>	263
20.1	Faktoren der Abfrageleistung	263
20.1.1	Schreiben von leistungsstarken Abfragen	263
20.1.2	Indizieren in SQL Server	264
20.1.3	Grundlagen zu SQL Server-Index: Gruppierter Index	265
20.1.4	Grundlagen zu SQL Server-Index: nicht gruppierter Index	266
20.1.5	SQL Server-Indizes: Leistungsaspekte	266
20.2	Anzeigen der Abfrageleistung	267
20.2.1	Was ist ein Ausführungsplan?	267
20.2.2	Tatsächliche und geschätzte Ausführungspläne	268
20.2.3	Anzeigen grafischer Ausführungspläne	269
20.2.4	Interpretieren des Ausführungsplanes	270
20.2.5	Anzeige von Abfragestatistiken	270
	<b>Index</b>	273

# Einleitung

- Aufbau des Buches
- Begleitmaterial
- Danksagung
- Über den Autor

## Aufbau des Buches

In der SQL-Fachliteratur findet man zum Thema »Abfragen« die unterschiedlichsten Informationen. Und immer wird darauf hingewiesen, dass dies bei Oracle-Datenbanken so zu schreiben, bei MySQL-Datenbanken so zu schreiben und bei T-SQL von Microsoft so zu schreiben ist. Das ist in vielen Fällen verwirrend. Deswegen basiert dieses vorliegende Buch auf reiner T-SQL-Syntax. Auch ist dieses Buch für blutige Einsteiger gedacht. Administratoren, die diese Abfragesprache auch in einigen administrativen Themen benötigen, Entwickler, die die Syntax kennenlernen möchten, da viel davon sich in Datenbankobjekten widerspiegelt. Aber auch Benutzer, die sich mit der Berichtserstellung auseinandersetzen müssen. Natürlich gibt es zu jedem Thema des Buches noch tiefer greifende Informationen. Das soll hier nicht das Ziel sein. Es soll den ersten Einstieg geben. Die Datenbank, auf die sich alle Beispiele beziehen, ist eine sehr kleine und vereinfachte Datenbank. Der Sinn dahinter besteht darin, die Informationen, die die Abfragen liefern, auch zu vergleichen und nicht einfach hinzunehmen. Das würde sich als schwierig erweisen, wenn die Tabellen Tausende von Zeilen hätten. Auch soll es einfach sein, den Aufbau dieser Datenbank zu verstehen. Denn nur dann gelingt es, die Abfragen auch besser lesen zu können. Microsoft bietet schon seit SQL Server 2005 eine Beispieldatenbank »AdventureWorks« an, die sowohl Entwickler als auch Administratoren zu Testzwecken nutzen können. Aber sich in die Struktur der Datenbank mit den Tausenden von Datensätzen hineinzudenken, benötigt viel Zeit. Dieses Buch soll schnell und kompakt einen Einstieg bringen. Denn mit dem dann aufgebauten Wissen und der Überzeugung, dass die Daten wirklich dem entsprechen, was die Abfrage sucht, kann man sich dann mit jeder anderen Datenbank auseinandersetzen. Egal, welche Größe sie hat. Sie finden die Datenbank zum Buch mit vielen anderen Materialien unter [www.mitp.de/0631](http://www.mitp.de/0631).

## Begleitmaterial

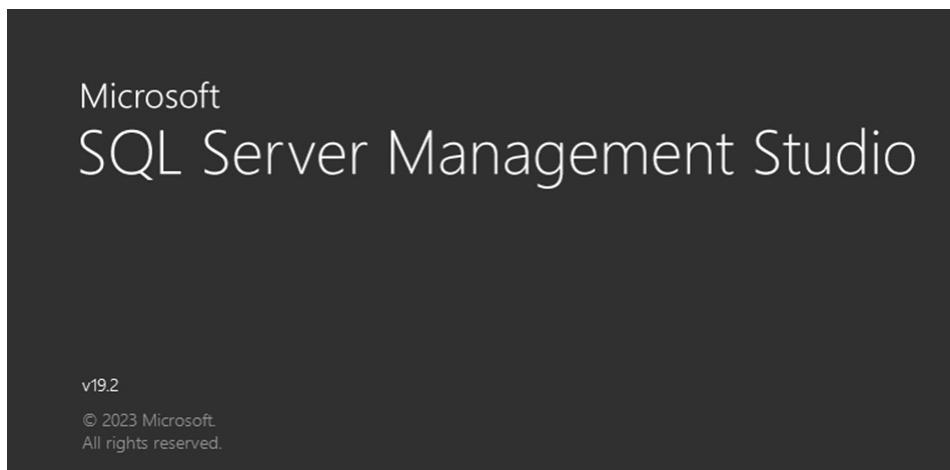
Das im Buch benutzte Material basiert auf der neuesten Version der Software von Microsoft: SQL Server 2022. Im Laufe der Erstellung des Buches kamen einige kumulativen Updates, die hier aber keine Rolle spielen. Natürlich kann die Datenbank auch auf früheren Versionen angelegt und abgefragt werden. Sollten diverse Möglichkeiten erst ab einer bestimmten Version im Abfrage-Code zur Verfügung stehen, wird im Kapitel explizit darauf hingewiesen. Microsoft stellt zwei Editionen zur Verfügung, die kostenlos genutzt werden können: die »SQL Server Express Edition« und die »SQL Server Developer Edition«. Diese können bei Microsoft heruntergeladen und problemlos ohne größeren Aufwand installiert werden.

### Informationen zum System

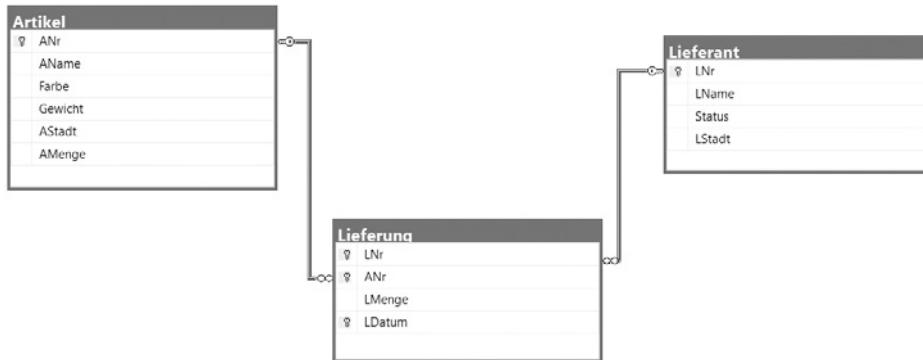
---

```
Microsoft SQL Server 2022 (RTM-CU10) (KB5031778) - 16.0.4095.4 (X64)
Oct 30 2023 16:12:44
Copyright (C) 2022 Microsoft Corporation
Developer Edition (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build
17763: ) (Hypervisor)
```

Auch das Verwaltungstool »SQL Server Management Studio« ist kostenlos und kann mit der jeweils aktuellsten Version bei Microsoft heruntergeladen werden. Natürlich ist es auch möglich, die Beispiele mit früheren Versionen nachzuvollziehen.



Den grundsätzlichen Aufbau der im Buch benutzten Beispieldatenbank zeigt das nachfolgende Schaubild.



Um die Datenbank anzulegen, kann nachfolgendes Skript genutzt werden. Allerdings wird aus drucktechnischen Gründen nicht der gesamte Pfad gezeigt und muss dem eigenen System angepasst werden. In der Datenbank werden nicht nur drei Tabellen angelegt, sondern auch einige Trigger. Diese sind für die automatische Datenaktualisierung verantwortlich. Sollte eine Lieferung hinzugefügt werden, wird automatisch der Status eines Lieferanten um einen Punkt nach oben gezählt. Das heißt, der Status eines Lieferanten in der Tabelle »Lieferant« ergibt sich automatisch aus der Anzahl der Lieferungen. Des Weiteren wird der Lagerbestand automatisch korrigiert. Beispielsweise ist der Artikel »A01« zweimal mit jeweils 300 Stück geliefert worden. Demzufolge ist der Lagerbestand in der Spalte »AMenge« 600. Natürlich werden auch bei dem Löschen einer Lieferung beide Werte angepasst und – sollte die Liefermenge im Nachhinein korrigiert werden – wiederum der Lagerbestand.

### Datenbank:

```

-- Beispieldaten zum Erstellen einer Datenbank mit SQL Server 2022
-- Copyright 2024 Dirk Angermann IT-Training & Consulting

USE master;
go

-- eventuell vorhandene Datenbank löschen
DROP DATABASE IF EXISTS db_Test;
go

-- Neue Datenbank 'db_Test' erstellen
-- Bitte die Pfadangabe Ihrem System anpassen.
CREATE DATABASE db_Test
ON PRIMARY

```

```
(  
    NAME = 'db_Test_Data'  
    ,FILENAME = 'C:\Program Files\...\DATA\db_Test_Data.mdf'  
    ,SIZE = 10 MB  
    ,MAXSIZE = 50 MB  
    ,FILEGROWTH = 10 MB  
)  
  
LOG ON  
(  
    NAME = 'db_Test_Log'  
    ,FILENAME = 'C:\Program Files\...\DATA\db_Test_Log.ldf'  
    ,SIZE = 1 MB  
    ,MAXSIZE = 25 MB  
    ,FILEGROWTH = 1 MB  
);  
go  
  
-- Wechsel in die Datenbank db_Test  
USE db_Test;  
go  
  
-- Tabellen erstellen  
DROP TABLE IF EXISTS dbo.Artikel;  
go  
  
CREATE TABLE dbo.Artikel  
(  
    ANr VARCHAR(3) NOT NULL  
    ,AName VARCHAR(20) NOT NULL  
    ,Farbe VARCHAR(7) NULL  
    ,Gewicht DECIMAL(9,2) NULL  
    ,AStadt VARCHAR(20) NULL  
    ,AMenge INT NULL  
    ,CONSTRAINT PK_ANr PRIMARY KEY(ANr)  
);  
go  
  
DROP TABLE IF EXISTS dbo.Lieferant;  
go
```

```
CREATE TABLE dbo.Lieferant
(
LNr VARCHAR(3) NOT NULL
,LName VARCHAR(20) NOT NULL
,Status INT NOT NULL CONSTRAINT DF_Zahl DEFAULT(0)
,LStadt VARCHAR(20) NULL
,CONSTRAINT PK_LNr PRIMARY KEY(LNr)
);
go

DROP TABLE IF EXISTS dbo.Lieferung;
go

CREATE TABLE dbo.Lieferung
(
LNr VARCHAR(3) NOT NULL
,ANr VARCHAR(3) NOT NULL
,LMenge INT NOT NULL
,LDatum DATE NOT NULL
,CONSTRAINT ZPK_Lief PRIMARY KEY(LNr,ANr,LDatum)
,CONSTRAINT FK_LNr FOREIGN KEY(LNr) REFERENCES dbo.Lieferant(LNr)
,CONSTRAINT FK_ANr FOREIGN KEY(ANr) REFERENCES dbo.Artikel(ANr)
);
go

-- Trigger erstellen
DROP TRIGGER IF EXISTS dbo.tr_INSERT;
go

CREATE TRIGGER dbo.tr_INSERT
ON dbo.Lieferung
AFTER INSERT
AS
UPDATE dbo.Lieferant
SET Status = Status + 1
FROM dbo.Lieferant AS a INNER JOIN INSERTED AS b ON a.LNr = b.LNr;

UPDATE dbo.Artikel
SET AMenge = AMenge + b.LMenge
FROM dbo.Artikel AS a INNER JOIN INSERTED AS b ON a.ANr = b.ANr;
go
```

```
DROP TRIGGER IF EXISTS dbo.tr_DELETE;
go

CREATE TRIGGER dbo.tr_DELETE
ON dbo.Lieferung
AFTER DELETE
AS
DECLARE @Cur_LNr VARCHAR(5)
    ,@Cur_ANr VARCHAR(5)
    ,@cur_LMenge INT;

DECLARE cur_Lief_Lösch CURSOR
FOR
SELECT
    LNr
    ,ANr
    ,LMenge
FROM DELETED;

OPEN cur_Lief_Lösch;

FETCH NEXT FROM cur_Lief_Lösch INTO @cur_LNr, @cur_ANr, @cur_LMenge;

WHILE @@FETCH_STATUS = 0
BEGIN
UPDATE dbo.Artikel
SET AMenge = AMenge - @cur_LMenge
WHERE ANr = @cur_ANr

UPDATE dbo.Lieferant
SET Status = Status - 1
WHERE LNr = @cur_LNr

FETCH NEXT FROM cur_lief_lösch INTO @cur_LNr, @cur_ANr, @cur_LMenge
END;

CLOSE cur_Lief_Lösch;
DEALLOCATE cur_Lief_Lösch;
go
```

```
DROP TRIGGER IF EXISTS dbo.tr_UPDATE;
go

CREATE TRIGGER dbo.tr_UPDATE
ON dbo.Lieferung
AFTER UPDATE
AS
IF UPDATE (LMenge)
BEGIN
UPDATE dbo.Artikel
SET AMenge = AMenge + (SELECT SUM(LMenge) FROM INSERTED)
- (SELECT SUM(LMenge) FROM DELETED)
FROM dbo.Artikel AS a INNER JOIN INSERTED AS b ON a.ANr = b.ANr
INNER JOIN DELETED AS c ON a.ANr = c.ANr
END;
go

SET NOCOUNT ON;
go

--Tabellen mit Anfangsdaten füllen
INSERT INTO dbo.Lieferant VALUES('L01','Schmidt',0,'Hamburg'),
('L02','Jonas',0,'Ludwigshafen'),
('L03','Blank',0,'Ludwigshafen'),
('L04','Clark',0,'Hamburg'),
('L05','Adam',0,'Aachen');

go

INSERT INTO dbo.Artikel VALUES('A01','Mutter','rot',12,'Hamburg',0),
('A02','Bolzen','grün',17,'Ludwigshafen',0),
('A03','Schraube','blau',17,'Mannheim',0),
('A04','Schraube','rot',14,'Hamburg',0),
('A05','Nockenwelle','blau',12,'Ludwigshafen',0),
('A06','Zahnrad','rot',19,'Hamburg',0);

go

INSERT INTO dbo.Lieferung VALUES('L01','A01',300,'20190518');
INSERT INTO dbo.Lieferung VALUES('L01','A02',200,'20190713');
INSERT INTO dbo.Lieferung VALUES('L01','A03',400,'20190101');
INSERT INTO dbo.Lieferung VALUES('L01','A04',200,'20190725');
INSERT INTO dbo.Lieferung VALUES('L01','A05',100,'20190801');
```

```
INSERT INTO dbo.Lieferung VALUES('L01','A06',100,'20190723');
INSERT INTO dbo.Lieferung VALUES('L02','A01',300,'20190802');
INSERT INTO dbo.Lieferung VALUES('L02','A02',400,'20190805');
INSERT INTO dbo.Lieferung VALUES('L03','A02',200,'20190806');
INSERT INTO dbo.Lieferung VALUES('L04','A02',200,'20190809');
INSERT INTO dbo.Lieferung VALUES('L04','A04',300,'20190820');
INSERT INTO dbo.Lieferung VALUES('L04','A05',400,'20190821');
go

-- Abfragen der Tabellen
SELECT
LNr AS 'Lieferantennummer'
,LName AS 'Lieferantename'
,Status AS 'Status des Lieferanten'
,LStadt AS 'Firmensitz'
FROM dbo.Lieferant;
go

SELECT
Anr AS 'Artikelnummer'
,AName AS 'Artikelname'
,Farbe
,Gewicht
,AStadt AS 'Lagerort'
,AMenge AS 'Lagermenge'
FROM dbo.Artikel;
go

SELECT
LNr AS 'Lieferantennummer'
,ANr AS 'Artikelnummer'
,LMenge AS 'Gelieferte Menge'
,CONVERT(VARCHAR(10),LDatum,104) AS 'Lieferdatum'
FROM dbo.Lieferung;
go

SET NOCOUNT OFF;
go
```

## Danksagung

Als der Verlag an mich herantrat und mich bat, dieses Buch zu schreiben, dachte ich mir: »Cool, ich schreibe einfach ein paar Zeilen um meine Skripte herum, die ich seit 20 Jahren in meinen Seminaren benutze.« Schnell stellte sich heraus, dass es doch gar nicht so einfach ist. Danke an das Verlagshaus für die Geduld. Aber auch danke an alle, die mich in der Zeit des Schreibens ertragen haben. Einen speziellen und lieben Dank an die tolle Frau an meiner Seite.

## Über den Autor

Dirk Angermann wurde 1969 in Halle/Saale geboren. Bei seinem Einstieg in das Berufsleben erlangte er erste Erfahrungen im Bereich der EDV im VEB Datenverarbeitungszentrum Halle. PL/1- und Assemblerprogrammierung für Großrechenanlagen zählten zu seinen ersten Programmiersprachen. Aber auch dBase gehörte dazu. Weitere Kenntnisse im Bereich von Netzwerkinfrastruktur und Aktive Directory erlernte er als Netzwerkadministrator bei der Bundeswehr in Potsdam beim IV. Korps. Seit über 20 Jahren ist er Microsoft Certified Trainer und unterrichtet europaweit alle Themen rund um SQL Server. Auch als Berater ist er in vielen Unternehmen, Ämtern, Behörden und Institutionen anzutreffen und unterstützt in allen Bereichen rund um die Datenbanksoftware von Microsoft. Diese praktischen Erfahrungen versucht er, in seine Seminare einzubinden.

# Arbeiten mit dem SQL Server Management Studio

- Starten von SQL Server Management Studio
- Herstellen einer Verbindung mit SQL Server
- Einstellungen im SQL Server Management Studio
- Arbeiten mit Skriptdateien und Projekten

## 1.1 Starten von SQL Server Management Studio

Das SQL Server Management Studio (SSMS) ist eine integrierte Verwaltungs-, Entwicklungs- und Abfrageanwendung mit vielen Funktionen zum Durchsuchen von und Arbeiten mit Datenbanken. SQL Server Management Studio basiert auf der Visual Studio-Shell.

Sie können das SQL Server Management Studio folgendermaßen starten:

- Verwenden der Verknüpfung im Windows-Startmenü
- Eingabe des Dateinamens SSMS.EXE in einem Eingabeaufforderungsfenster
- Eingabe von SSMS im Feld **SUCHBEGRIFF HIER EINGEBEN** in der Taskleiste

## 1.2 Herstellen einer Verbindung mit SQL Server

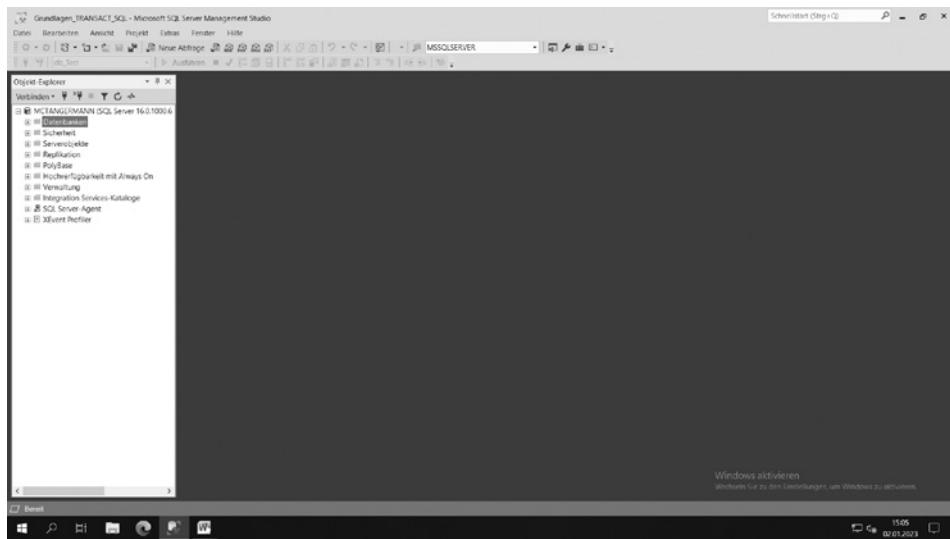
Standardmäßig wird in SSMS ein Dialogfeld **VERBINDUNG MIT SERVER HERSTELLEN** angezeigt. Mithilfe dieses Auswahlfensters kann man den Namen der jeweiligen Instanz auswählen oder auch manuell eingeben, mit der man sich verbinden möchte.

## Kapitel 1

### Arbeiten mit dem SQL Server Management Studio



Das Fenster präsentiert sich seit vielen Versionen des Management Studio in nachfolgend gezeigter Ansicht.

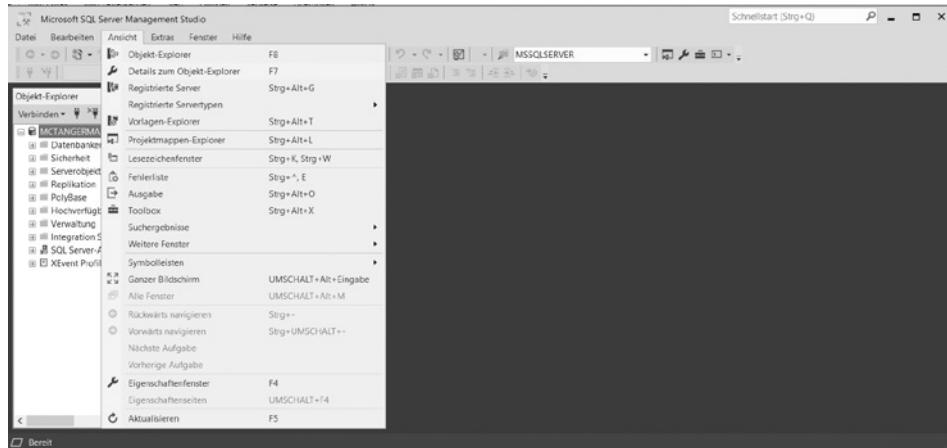


## 1.3 Einstellungen im SQL Server Management Studio

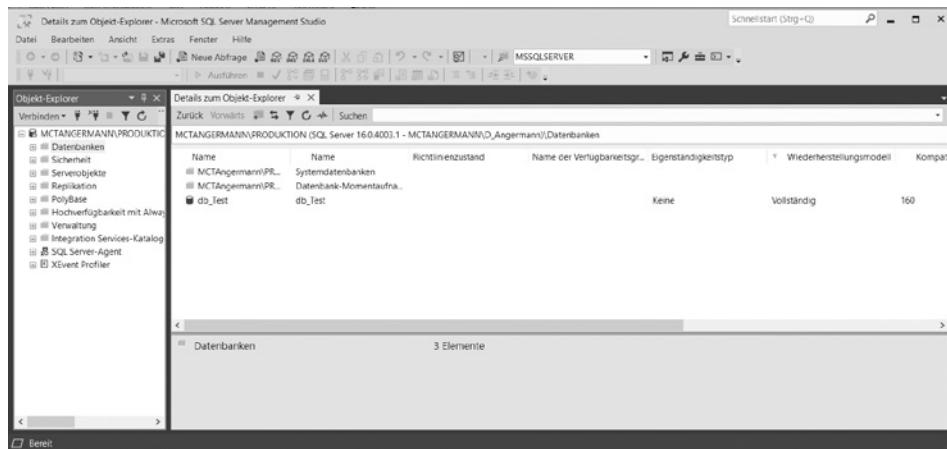
Sobald man sich mit einer Instanz verbunden hat, kann man einige Einstellungen tätigen, die man über den Menüpunkt **ANSICHT** oder **TOOLS – OPTIONEN** findet.

Im Nachfolgenden werden einige Einstellungen gezeigt.

### Registerkarte DETAILS ZUM OBJEKT-EXPLORER



Nachfolgende Darstellung wird präsentiert:

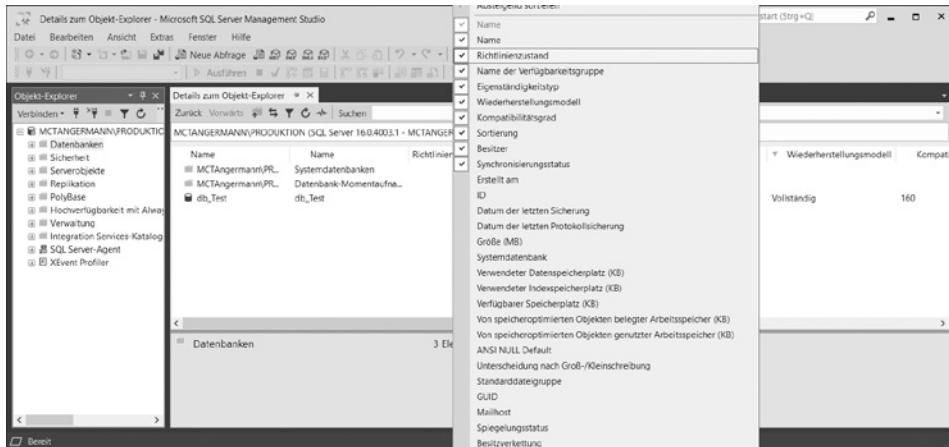


Nun kann es sein, dass gerade in der Tabelle der aufgelisteten Datenbanken einige Spalten existieren, die nicht benötigt werden. Diese können entfernt werden, indem man mit der rechten Maustaste auf den Spaltenkopf klickt und das Häkchen im angezeigten Kontextmenü entfernt. Wie z.B. die Spalten

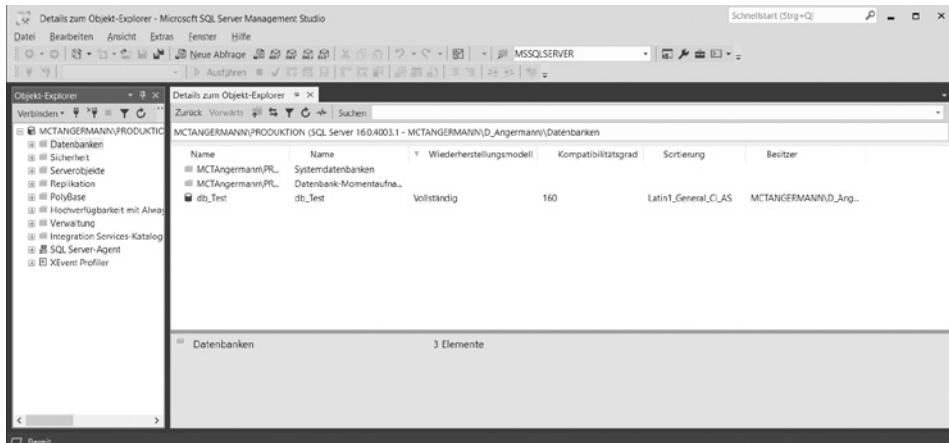
- Richtlinienzustand
- Name der Verfügbarkeitsgruppe
- Eigenständigkeitstyp
- Synchronisierungsstatus

## Kapitel 1

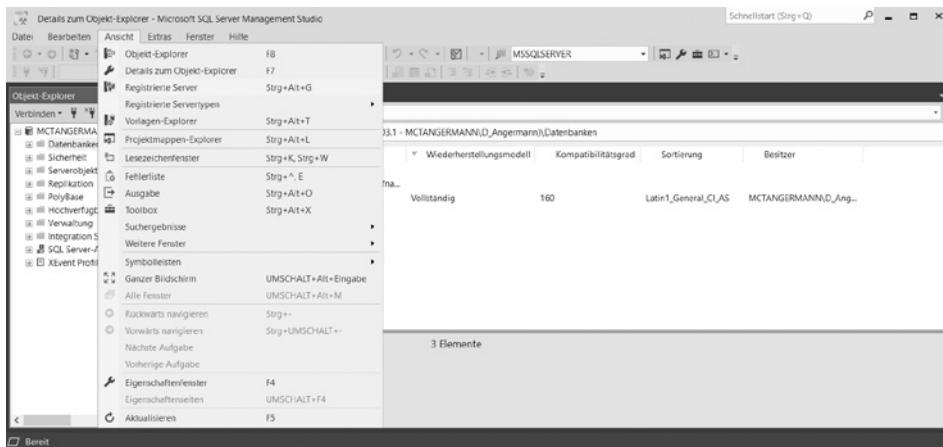
### Arbeiten mit dem SQL Server Management Studio



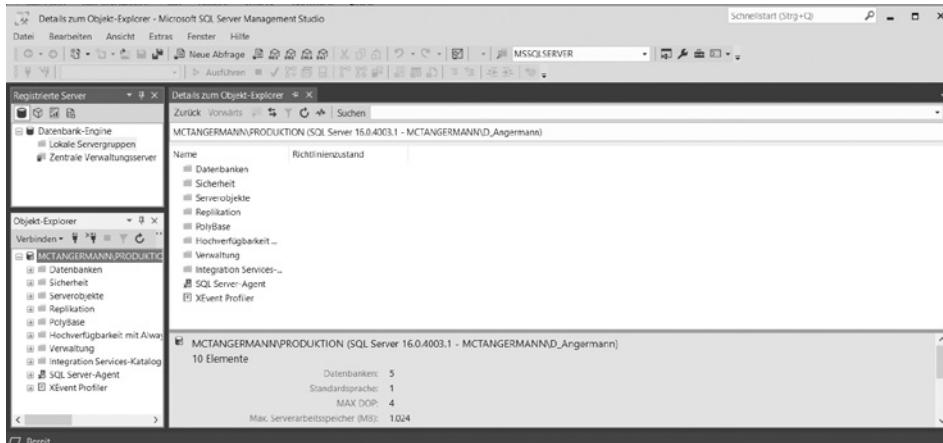
Nun könnte die Darstellung der Daten so präsentiert werden:



Natürlich kann man das Management Studio auch benutzen, um mehrere Instanzen zu administrieren. Hier wäre es hilfreich, ohne das Verbindungsfenster öffnen zu müssen, schnellstmöglich auf die vorhandenen Instanzen zugreifen zu können. Hier stehen im Menüpunkt **ANSICHT** die Auswahl **REGISTRIERTE SERVER** zur Verfügung.



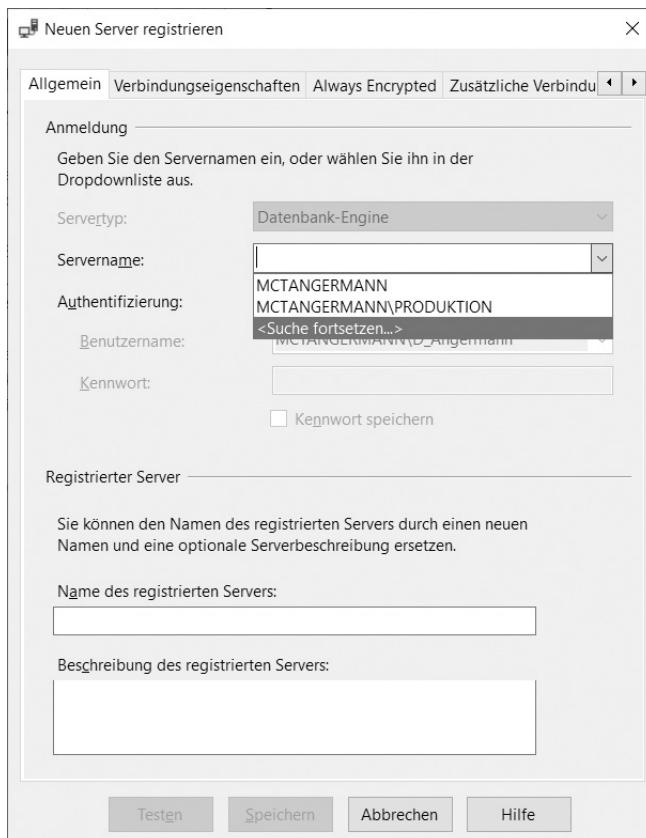
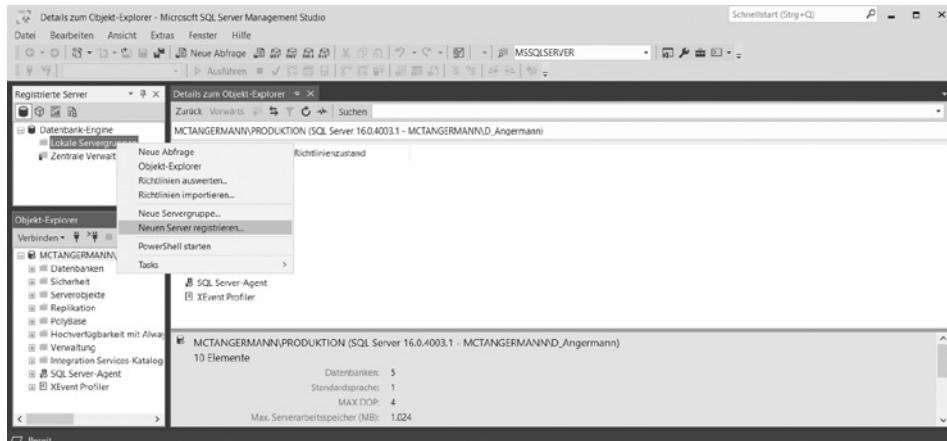
Nachfolgende Darstellung wird präsentiert:



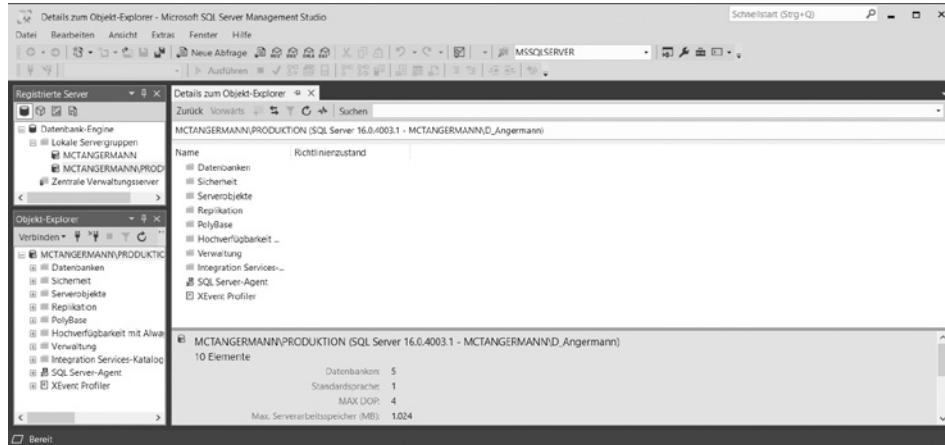
Über das Kontextmenü lassen sich nun beliebig viele Instanzen über den Menüpunkt NEUEN SERVER REGISTRIEREN hinzufügen.

## Kapitel 1

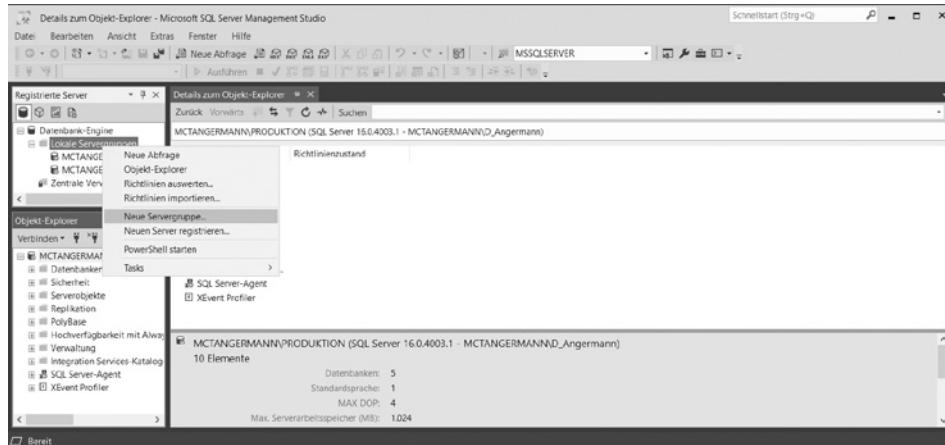
### Arbeiten mit dem SQL Server Management Studio



Nachfolgende Darstellung wird präsentiert:

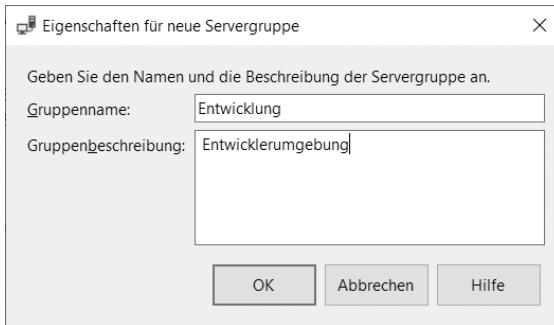


Auch kann man Instanzen nach bestimmten Funktionalitäten oder vielleicht auch nach Standorten gruppieren. Da hilft wieder das Kontextmenü über NEUE SERVERGRUPPE:

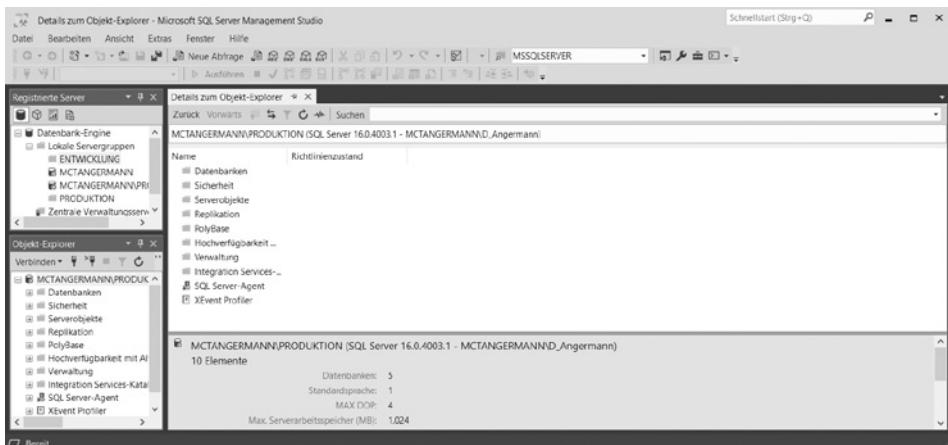


## Kapitel 1

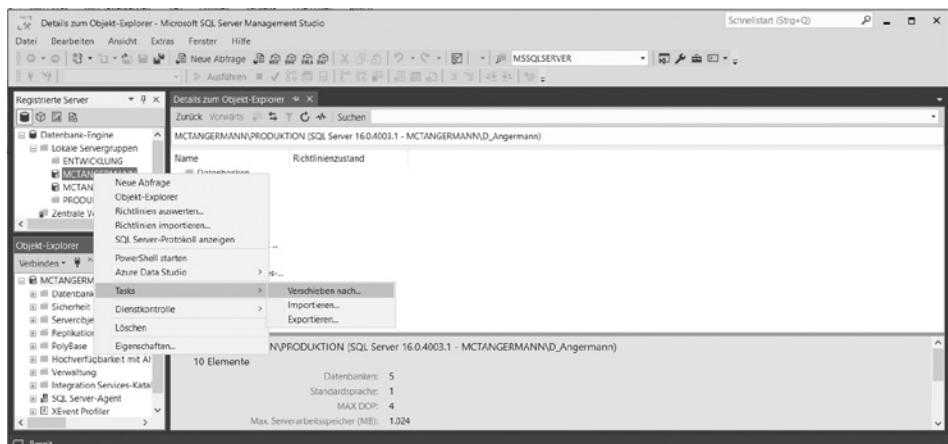
### Arbeiten mit dem SQL Server Management Studio



Nachfolgende Darstellung wird präsentiert:



Wenn man die Instanzen nicht direkt in der Gruppe registriert, kann man diese natürlich auch verschieben. Im Kontextmenü steht der Punkt **TASKS|VERSCHIEBEN NACH** zur Verfügung.



# Index

## A

Abfrageleistung 263  
Anzeigen 267  
Faktoren 263  
Verbessern 263  
Abfragestatistiken 270  
Abgeleitete Tabellen 177  
Richtlinien 177  
Ablaufsteuerung 244  
Ablaufsteuerungsanweisungen 45  
BEGIN ... END 46  
BEGIN TRANSACTION 46  
COMMIT TRANSACTION 46  
IF ... ELSE 45  
ROLLBACK TRANSACTION 46  
TRY ... CATCH 46  
WHILE 45  
Aggregatfunktionen 43, 79, 123, 126  
Aliasse 177  
ALL 82  
Allgemeine Tabellenausdrücke  
Erstellen und abfragen 180  
Schreiben 179  
ALTER 41  
Arbeiten mit Synonymen 243  
Arbeiten mit Variablen 242  
ASC 91  
Ausführen von Bedingungsüberprüfungen 136  
Ausführungsplan 267  
geschätzt 268  
grafisch 269  
Interpretieren 270  
Tatsächlich 268  
Auswählen von Elementen  
aus einer Liste 137  
AVG 79, 126

## B

Batches 256  
Probleme 256  
BEGIN TRANSACTION 257, 258  
Bereiche 73  
BETWEEN 41  
BIGINT 100  
Binärzeichenfolge 102  
Datentypen 102  
BIT 101

**C**

CASE 44  
CASE-Ausdrücke 67  
CAST 128, 129  
CHAR 106  
CHOOSE 135, 137  
COALESCE 139  
COLLATE-Klausel 107  
COMMIT 255  
COMMIT TRANSACTION 257, 259  
CONVERT 128, 130  
COUNT 79, 126  
COUNT(\*) 79  
CREATE 41  
CREATE FUNCTION 173  
CREATE VIEW 169, 171  
CROSS JOIN 153  
CUBE 213  
CURSOR 103

**D**

DANSE\_RANK 194  
DATE 115  
DATEADD() 119  
DATEFROMPARTS 115  
Datenabfragesprache 41

**DATENAME()** 121  
 Datenbearbeitungssprache 41  
 Datendefinitionssprache 41  
 Datenkontrollsprache 41  
 Datenpivotisierung 205  
 Datentypen  
     Andere 100  
     Binärzeichenfolgen 100  
     Genaue numerische 100  
     Konvertierung 105  
     Rangfolge 103  
     Sonstige 102  
     Ungefähr numerische 100  
 Datentypenkonvertierung  
     Implizite und explizite 128  
**DATEPART()** 120  
**DATETIME** 115  
**DATETIME2** 115  
**DATETIMEOFFSET** 115  
 Datums- und Uhrzeitdatentypen  
     Literale 115  
 Datums- und Uhrzeitfunktionen 124  
 Datum-/Uhrzeitfunktionen 43  
 Datum und Uhrzeit  
     Abfragen 117  
     Funktionen 117  
     Getrenntes Arbeiten 116  
 Datum- und Uhrzeitdaten 114  
**DCL** 41  
**DDL** 41  
**DECIMAL** 101  
**DELETE** 41  
**DENY** 41  
**DESC** 91  
 Determinismus 124  
**DISTINCT** , 63  
**DML** 41  
**DQL** 41  
**DROP** 41  
 Dynamisches SQL 236  
     Erstellen 236  
     Schreiben von Abfragen 237  
 Dynamische Verwaltungssichten 224, 225  
     Abfragen 226  
     Kategorisieren 225

**E**  
 Effiziente T-SQL-Abfragen schreiben 50  
 Entfernen von Duplikaten 63  
**EOMONTH** 115  
**EOMONTH()** 121  
 Erstellen von Prozeduren 232  
**EXCEPT** 186, 187  
**EXISTS**-Prädikat 150  
**F**  
 Fehlerbehandlung 249  
     Implementieren 249  
     Strukturierte Ausnahmehandlung 249  
 Fehlerinformation 251  
     Abfragen eines ERROR-Objekts 251  
     THROW-Anweisung 252  
 Fenster 189  
     Erstellen 189  
     Partitionieren 191  
 Fensteraggregatfunktionen 193  
 Fensterfunktionen 123, 126, 192  
     Definieren 193  
 Fensterkomponenten 190  
 Fensteroffsetfunktionen 203  
 Fensterrangfolgefunktionen 194  
 Fensterverteilungsfunktionen 202  
 Filtern von Daten 71  
 Filtern von Gruppen 88  
**FIRST\_VALUE** 203  
**FLOAT** 101  
**FORMAT ()** 110  
**G**  
 Gespeicherte Prozeduren 229  
     Ausführen 230  
     Übergeben von Parametern 231  
     Untersuchen 229  
 Gespeicherte Systemprozeduren 222  
     Ausführen 222  
     Häufig verwendete 224  
**GETDATE()** 118  
**GRANT** 41  
**GROUP BY**-Klausel 84  
 Groupingsets 210

Gruppieren und Aggregieren von Daten 79  
 Gruppiertes Index 265  
 Gruppierungsschlüssel 265

**H**

HAVING 88  
 Heap 265  
 HIERARCHYID 103

**I**

IF ... ELSE 244  
 IIF 135, 136  
 IN 41  
 Indexscan 265  
 Indexsuche 265  
 Indizieren in SQL Server 264  
 Informationsschemasichten 219  
 Inline-Tabellenwertfunktionen 173  
     Abrufen 174  
     Einfache 173  
 INNER JOIN 155  
 INSERT 41  
 INTEGER 100  
 Integrierte Funktionen 123  
     Typen 123  
 INTERSECT 186  
 ISNULL 138

**J**

Joins 153

**K**

Kartesisches Produkt 153  
 Kategorien von T-SQL-Anweisungen 41  
 Konvertieren von NULL 138  
 Konvertieren von Zeichenfolgen 132  
 Konvertierung 129  
 Konvertierungsfunktionen 124, 128

**L**

LAG 203  
 LAST\_VALUE 203  
 LEAD 203  
 LEFT() 110

Leistungsstarke Abfragen 263  
 Schreiben 263  
 LEN () 112  
 LIKE 41  
 Logische Funktionen 135  
 Verwenden 135  
 LOWER() 113

**M**

Mathematische Funktionen 43, 124  
 MAX 79, 126  
 Mengenlehre und SQL Server 48  
 Mengenoperatoren 183  
 MIN 79, 126  
 MONEY 101

**N**

NCHAR 106  
 Nicht gruppiertes Index 266  
 NTILE 195  
 NULL 138  
 NULLIF 140  
 NULL-Wert 77  
 NUMERIC 101  
 Numerische Datentypen 100  
 NVARCHAR(MAX) 106

**O**

OFFSET-FETCH 96  
 OFFSET-FETCH-Option 91  
 OPENDATASOURCE 127  
 OPENQUERY 127  
 OPENROWSET 127  
 OPENXML 127  
 ORDER BY-Klausel 91  
 OUTER JOIN 164  
 OVER 190

**P**

PARSE 128, 132  
 PIVOT 205  
 PIVOT-Elemente 207  
 Prädikatlogik 50  
     Auf SQL Server angewendet 51  
 Projekt 34  
 Projektmappen 34

**R**

RANK 127, 194  
 REAL 101  
 Relationales Modell 48  
 REPLACE() 114  
 REPLICATE() 114  
 RETURN 173  
 RETURN TABLE 173  
 REVERSE() 113  
 REVOKE 41  
 RIGHT() 110  
 ROLLBACK 255  
 ROLLBACK TRANSACTION 259  
 ROLLUP 213  
 ROW\_NUMBER 195  
 Rowsetfunktionen 123, 127  
 ROWVERSION 102

**S**

Schreiben von logischen Tests  
 mit Funktionen 135  
 SELECT-Anweisung 51, 55  
 Elemente 52  
 SELF JOIN 163  
 SET SHOWPLAN\_XML ON 268  
 SET STATISTICS IO ON/OFF 271  
 SET STATISTICS TIME ON/OFF 271  
 Sicht 169  
 einfache 171  
 Skalarfunktion 44  
 Skalarfunktionen 123, 124  
 SMALLDATETIME 115  
 SMALLMONEY 101  
 Sortieren von Daten 91  
 Sortierung 107, 124  
 Spaltenaliasse 64  
 SQL-Fensterfunktionen 189  
 SQL Server-Datentypen 99  
 SQL Server Management Studio 21  
 SQL Server-Metadaten 217  
 SQL\_VARIANT 103  
 Steuern des Programmflusses 244  
 STUFF() 112  
 SUBSTRING() 111  
 SUM 79, 126

SYSDATETIME() 118  
 SYSDATETIMEOFFSET() 119  
 Systemkatalogsichten 217  
 Systemmetadatenfunktionen 124, 220

**T**

Tabellen 153  
 Abfragen 153  
 Tabellenausdrücke 179  
 Tabellenwertausdrücke 169  
 TABLE 103  
 Tablescan 265  
 TIME 115  
 TINYINT 100  
 TOP-Option 91, 93  
 Transaktionen 255  
 Autocommit-Transaktionen 255  
 Explizite 255  
 Steuern 258  
 TRY/CATCH 249  
 TRY\_CONVERT 128, 133  
 TRY ... PARSE 128  
 TRY\_PARSE 133  
 TRY- und CATCH-Blöcke 250  
 Erstellen 250  
 T-SQL-Batches 239  
 Einführung 239  
 T-SQL-Programmierelemente 239  
 T-SQL-Sprachelemente  
 Ausdrücke 44  
 Batchtrennzeichen 46  
 Kommentare 46  
 Prädikate und Operatoren 41  
 Steuerung von Abläufen, Fehlern und  
 Transaktionen 45  
 Variablen 43  
 T-SQL-Sprachelemente 41  
 T-SQL-Variablen 240

**U**

Unbekannte Werte 77  
 UNION ALL 186  
 UNION-ALL-Operator 183  
 UNION-Operator 183  
 UNIQUEIDENTIFIER 102  
 UNPIVOT 205, 208

Unterabfragen 141  
Eigenständige 142  
Korrelierte 148  
Mehrwertige 144  
UPDATE 41  
UPPER() 113

## V

VARCHAR 106  
VARCHAR(MAX) 106  
Vergleichsoperatoren 71  
Verknüpfungsoperatoren 76  
Verwaltungsobjekte 224  
Dynamische 224

## W

WHERE-Klausel 71  
WHILE 244, 246  
WITH TIES 94, 95

## X

XML 103

## Z

Zeichendatentypen 106  
Zeichenfolgenfunktionen 42, 110  
Zeichenfolgenverkettungen 108  
Zeichenfolgeübereinstimmungen 74