

T-SQL-Abfragen für Microsoft SQL Server 2022

» Hier geht's
direkt
zum Buch

DAS VORWORT



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 möchte, 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.

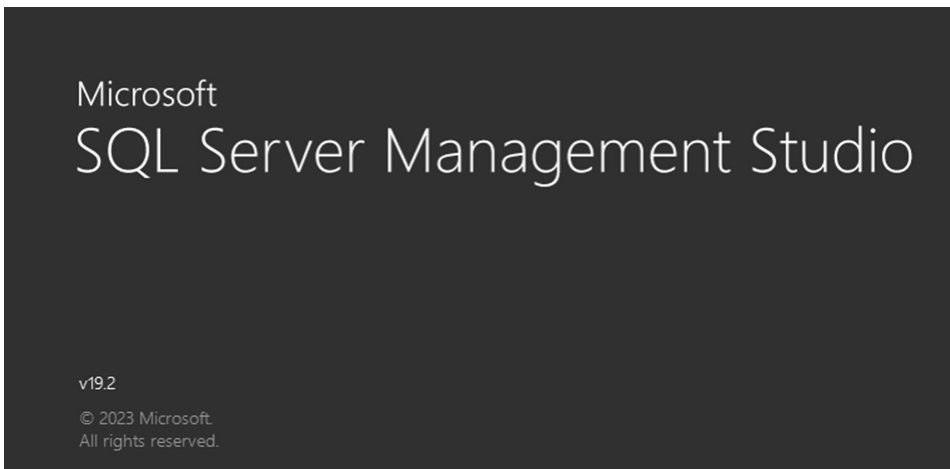
Begleitmaterial

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.

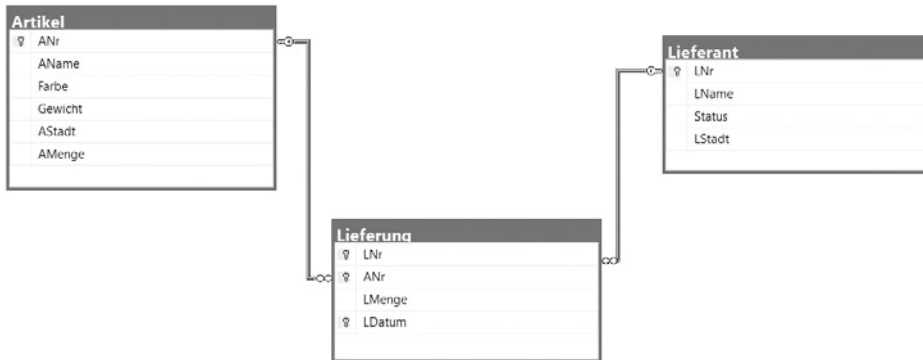
Informationen zum System

```
-----  
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:

```

-- Beispielskript 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
,ASadt 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 'Lieferantenname'
,Status AS 'Status des Lieferanten'
,LStadt AS 'Firmensitz'
FROM dbo.Lieferant;
go

SELECT
Anr AS 'Artikelnummer'
,AName AS 'Artikelname'
,Farbe
,Gewicht
,ASStadt 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.