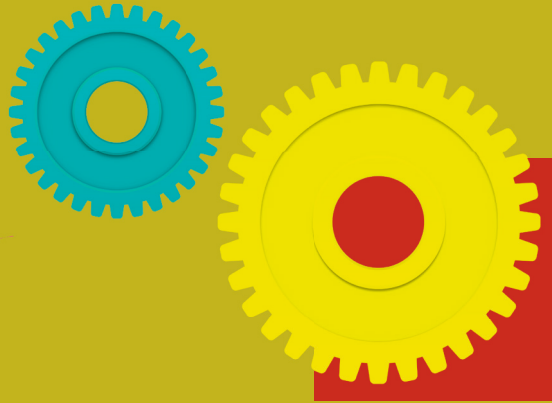


klemens KONOPASEK



SQL SERVER 2017

DER
SCHNELLE
EINSTIEG

HANSER



Im Internet: Alle Beispiele aus dem Buch

Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



www.hanser-fachbuch.de/newsletter



Hanser Update ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



www.hanser-fachbuch.de/update



Klemens Konopasek

SQL Server 2017

Der schnelle Einstieg

HANSER

Der Autor:

Klemens Konopasek, Gössendorf/Graz
klemens@konopasek.at

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen - oder Teilen davon - entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) - auch nicht für Zwecke der Unterrichtsgestaltung - reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2018 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sylvia Hasselbach

Copy editing: Walter Saumweber, Ratingen

Umschlagdesign: Marc Müller-Bremer, München, www.rebranding.de

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-44826-1

E-Book-ISBN: 978-3-446-44916-9

Inhalt

Vorwort	XIII
1 Der SQL Server 2017 stellt sich vor	1
1.1 SQL Server – wer ist das?	2
1.1.1 Der SQL Server im Konzert der Datenbanksysteme	2
1.1.2 Entscheidungsszenarien für Datenbanksysteme	5
1.1.3 Komponenten einer Datenbankanwendung	7
1.1.4 SQL Server – das Gesamtkonzept	10
1.2 Versionen und Editionen des SQL Servers	12
1.3 SQL Server 2017 installieren	17
1.4 Datenbanken installieren und nutzen	41
1.5 Gratis: die Express Edition	47
1.6 SQL Server Feature Pack	49
2 Die grafischen Tools des SQL Server 2017	53
2.1 Die Tools im Überblick	53
2.2 Das Management Studio	56
2.3 Das Kommandozeilentool: SQLCMD	76
2.4 Der Konfigurations-Manager	78
2.5 Das SQL Server-Installationscenter	81
2.6 Der Profiler	82
2.7 Der Datenbankoptimierungsratgeber	83
2.8 Die SQL Server Data Tools	85
2.9 Der Import/Export-Assistent	89
2.10 Der SQL Server Migration Assistant	99
2.11 SQL Operations Studio	103
3 Eine neue Datenbank erstellen	105
3.1 Erstellen einer neuen Datenbank	105
3.1.1 Bestandteile einer Datenbank	105

3.1.2	Datenbank mit dem grafischen Tool anlegen	109
3.1.3	Datenbank über eine SQL-Anweisung erstellen	117
3.1.4	Datenbank mit Filestream ausstatten	118
3.2	Tabellen in der Datenbank erstellen	123
3.2.1	Tabellenfelder definieren	124
3.2.2	Spalteneigenschaften	128
3.2.3	Constraints	131
3.2.4	Indizierung	141
3.2.5	Erste Daten erfassen	149
3.3	Datenbankdiagramme einsetzen	152
3.4	Richtlinien für Benennungsregeln einsetzen	156
3.5	Was Sie noch wissen sollten	161
3.5.1	Tabellen in anderen Dateigruppen speichern	161
3.5.2	Tabellen direkt mit DDL-Anweisungen erstellen	163
3.5.3	Gefahren der grafischen Oberfläche	164
3.5.4	Berechnete Spalten integrieren	168
3.5.5	Objekte und Datenbanken skripten	171
3.6	Tabelle mit Filestream und FileTable	174
3.6.1	Tabelle mit Filestream erstellen	175
3.6.2	Objekte in einer FileTable speichern	181
3.7	Beispieldatenbank generieren	195
3.8	Speicheroptimierte Tabellen	197
3.8.1	Datenbank mit In-Memory-Filegroup erstellen	197
3.8.2	Speicheroptimierte Tabelle anlegen	199
3.8.3	Index für speicheroptimierte Tabellen	203
3.8.4	Speichernutzung beschränken	208
4	SQL – Zugriff auf Daten	211
4.1	Einsatz des Abfrage-Designers	213
4.1.1	Die Bereiche des Abfrage-Designers	213
4.1.2	Erstellen einer Abfrage	219
4.2	Sichten für den Datenzugriff gestalten	233
4.2.1	Gründe für den Einsatz von Sichten	234
4.2.2	Erstellen einer Sicht	235
4.2.3	Daten aus einer Sicht abrufen	237
4.3	SQL-Anweisungen verwenden	239
4.3.1	Data Query Language (DQL)	240
4.3.2	Data Manipulation Language (DML)	250
4.3.3	Die MERGE-Anweisung	251
4.3.4	Den Abfrage-Designer im Abfrageeditor einsetzen	256
4.4	Abfragen mit Geodaten	258
4.4.1	Typen im Geodatenmodell	259
4.4.2	Geodaten in Tabellen speichern und verwenden	265
4.4.3	Index für räumliche Daten	282

5	Transact-SQL – die Sprache zur Serverprogrammierung	287
5.1	Bestandteile und Funktionalität von Transact-SQL	289
5.1.1	Variablen und Datentypen	289
5.1.2	Benutzerdefinierte Tabellentypen	297
5.1.3	Funktionen	299
5.1.4	Kontrollstrukturen	336
5.1.5	Cursor für Datenzugriffe einsetzen	353
5.2	Transaktionen gezielt steuern	359
5.2.1	Automatische Transaktionen	360
5.2.2	Explizite und implizite Transaktionen	361
5.2.3	Benannte Transaktionen	367
5.3	SET-Optionen verwenden	368
5.4	Fehlerbehandlung in den Code einbauen	376
5.5	Sequenzen	382
5.6	Paging mit OFFSET und FETCH	385
5.7	Window-Funktionen	386
6	Gespeicherte Prozeduren, Funktionen und Trigger	389
6.1	Gespeicherte Prozeduren programmieren	390
6.1.1	Aufbau einer gespeicherten Prozedur	392
6.1.2	Erzeugen einer gespeicherten Prozedur	393
6.1.3	Einfache gespeicherte Prozeduren	403
6.1.4	Gespeicherte Prozeduren mit Eingabeparametern	406
6.1.5	Ergebnisrückgabe von Prozeduren	409
6.1.6	Cursor in gespeicherten Prozeduren nutzen	420
6.1.7	Transaktionen in Prozeduren	428
6.1.8	Table-Valued Parameter einsetzen	433
6.1.9	Systemintern kompilierte gespeicherte Prozeduren	438
6.1.10	Gespeicherte Prozeduren aus Client-Anwendungen heraus aufrufen	449
6.2	Mit Triggern automatisieren	464
6.2.1	DML-Trigger: Insert, Update, Delete	465
6.2.2	Triggerreihenfolge festlegen	485
6.2.3	INSTEAD OF-Trigger	487
6.2.4	Rekursive Trigger	490
6.2.5	Trigger löschen	502
6.2.6	Systemeigen kompilierte Trigger	503
6.2.7	DDL-Trigger	508
6.3	Benutzerdefinierte Funktionen implementieren	515
6.3.1	Skalarwertfunktionen	515
6.3.2	Inline-Funktionen	521
6.3.3	Tabellenwertfunktionen	523
6.3.4	Systemintern kompilierte benutzerdefinierte Funktionen	529

6.4	Debuggen	532
6.4.1	Voraussetzungen für das Debuggen	532
6.4.2	Debuggen einer gespeicherten Prozedur	534
6.4.3	Debuggen von Triggern	539
6.4.4	Debuggen von Funktionen	542
6.5	Praxistipps	543
6.5.1	Fehleranalyse mit ERROR_MESSAGE()	544
6.5.2	Fehler gezielt zur Ablaufsteuerung einsetzen	546
6.5.3	Fehlerprotokoll führen	549
6.5.4	Über Fehler benachrichtigen lassen	551
6.5.5	Automatisierte Importe mit BULK INSERT	555
7	SQL Server CLR-Integration	561
7.1	Mit im Boot: .NET Framework	562
7.1.1	Integration mit dem Visual Studio	564
7.2	CLR-Aktivierung	567
7.2.1	Code auf den Server bringen: Assembly	570
7.3	.NET User-Defined Functions	573
7.4	.NET Stored Procedures	581
7.4.1	Datenzugriff aus der CLR heraus	581
7.4.2	Prozeduren mit Werterückgabe	582
7.4.3	Zugriff auf externe Daten	589
7.5	.NET-Trigger	596
7.6	User-Defined Aggregates (UDA)	605
7.7	Externe Assemblys verwenden	612
7.8	CLR-Sicherheitseinstellungen	619
7.8.1	Assembly als vertrauenswürdig erklären	620
7.8.2	Assembly signieren	624
7.9	Verwalten des Servers mit SMO	632
7.10	Übrigens: Debuggen	638
7.10.1	Debuggen einer T-SQL Stored Procedure	639
7.10.2	Debuggen einer .NET-Stored Procedure	641
8	Data Tier Applications und SQL Server Data Tools	645
8.1	Datenebenenanwendungen	645
8.1.1	DAC über Management Studio erstellen	646
8.1.2	Eine DAC auf dem SQL Server bereitstellen	649
8.1.3	Aktualisieren einer DAC	651
8.1.4	Entfernen einer DAC	654
8.1.5	Von DACPAC zu BACPAC	654
8.1.6	Erstellen einer DAC mit dem Visual Studio	659
8.2	Die SQL Server Data Tools	660
8.2.1	Ein neues Datenbankprojekt erstellen	660

8.2.2	Datenbankobjekte erstellen	662
8.2.3	Datenbankprojekt bereitstellen	667
8.2.4	Schemavergleich	670
8.2.5	Datenbank in ein Datenbankprojekt importieren	675
8.2.6	Ersatz für das Management Studio?	678
9	Client-Server-Datenbank verwalten	681
9.1	Anfügen und Trennen von Datenbanken	681
9.1.1	Trennen einer Datenbank	682
9.1.2	Anfügen einer Datenbank	685
9.1.3	Option „Automatisch schließen“	691
9.2	Datenbank sichern	692
9.2.1	Sicherungsvarianten	692
9.2.2	Sicherungsziele	694
9.2.3	Sicherung mit dem Management Studio	697
9.2.4	Sicherung über TRANSACT-SQL	703
9.2.5	Zeitgesteuerte Sicherung mit dem SQL Server-Agent	707
9.2.6	Zeitgesteuerte Sicherung mit der Express Edition	711
9.2.7	Datenbank wiederherstellen	715
9.2.8	Einsatz der Zeitachse beim Wiederherstellen	719
9.2.9	Wiederherstellung über Transact-SQL	724
9.2.10	Desaster Recovery	725
9.2.11	Recovery mit FILESTREAM	732
9.3	Datenänderungen protokollieren	735
9.3.1	Change Data Capture	735
9.3.2	Temporale Tabellen	741
9.4	Mit mehreren Instanzen arbeiten	767
9.4.1	Standardinstanzen und benannte Instanzen	768
9.4.2	Zugriff auf Instanzen steuern	770
10	Sicherheit und Zugriffsberechtigungen	775
10.1	Authentifizierungsmodi - Anmeldungen und Benutzer	775
10.1.1	Windows-Authentifizierung	777
10.1.2	Gemischter Modus	778
10.1.3	Anmeldung und Benutzer	778
10.2	Berechtigungen	780
10.3	Rollen	781
10.3.1	Serverrollen	781
10.3.2	Datenbankrollen	784
10.3.3	Anwendungsrollen	785
10.4	Anmeldeinformationen (Credentials)	786
10.5	Schema	788
10.6	Verwaltung im Management Studio	790

10.6.1	Serveranmeldung hinzufügen	790
10.6.2	Schema anlegen	796
10.6.3	Datenbankbenutzer hinzufügen	796
10.6.4	Rollen in einer Datenbank anlegen	800
10.7	Berechtigungen vergeben	802
10.7.1	Berechtigungen auf Datenbankebene	802
10.7.2	Berechtigungen auf Serverebene	810
10.8	Lösungen mit T-SQL	811
10.8.1	Sicherheitsobjekte anlegen	812
10.8.2	Generische Skripte	818
10.9	Contained Databases	818
10.10	Administratorzugriff wiederherstellen	826
10.11	Indirekte Zugriffe verwalten	830
10.11.1	Datenzugriffe über Sichten	831
10.11.2	Sicherheit mit Prozeduren erhöhen	832
10.12	Sicherheit auf Zeilenebene	839
10.12.1	Bestandteile von Row Level Security (RLS)	840
10.12.2	Sicherheitsfunktion erstellen	843
10.12.3	Security Policy definieren	846
10.12.4	Ändern von Sicherheitsrichtlinien	849
10.13	Zugriff auf andere Server	852
10.13.1	SQL Server als Verbindungsserver	854
10.13.2	Verbindungsserver mit Fremdprodukten	861
10.14	Daten verschlüsseln mit Always Encrypted	865
10.14.1	Voraussetzungen für Always Encrypted	866
10.14.2	Konfiguration von Always Encrypted	867
10.14.3	Vorhandene Daten verschlüsseln	874
10.14.4	Abfragen von verschlüsselten Daten	877
10.14.5	Erstellen von Tabellen mit verschlüsselten Spalten	881
10.14.6	Einfügen von Daten mit Verschlüsselung	884
10.14.7	Treibereinsatz am Client	887
11	Erweiterte Funktionalitäten	891
11.1	Datenbank-E-Mail	891
11.1.1	Einrichten von Datenbank-E-Mail	892
11.1.2	E-Mails aus der Anwendung heraus versenden	899
11.1.3	Varianten des E-Mail-Versands	901
11.1.4	Konfiguration über Systemprozeduren	908
11.1.5	Mailbenachrichtigung für Agent-Aufträge	914
11.2	Integration Services	922
11.2.1	Datenabgleich mit IS	923
11.2.2	Pakete ausführen und auf den Server bringen	946
11.2.3	SSIS-Projekte auf den Server bringen	948

12	SQL Server 2017 auf Linux	955
12.1	Installation des SQL Servers	957
12.2	Kommandozeilentools installieren	962
12.2.1	SQLCmd mit ODBC	962
12.2.2	mssql-cli	966
12.3	Server-Agent ergänzen	970
12.4	Integration Services	971
12.5	Serverdienst starten	972
12.6	Updates installieren	973
12.7	Weitere Konfiguration	975
12.8	Windows-Authentifizierung	980
12.9	Linux auch am Client: SQL Operations Studio	988
A	Anhang	993
A.1	Die Tabellen der Datenbank WAWI	993
Index	1005

Vorwort

Eine neue SQL Server-Version ist da! Dies bedeutet einerseits viel Freude, wieder mit neuen Features Aufgabenstellungen aus der Praxis noch besser lösen zu können, und andererseits aber auch, dass ich mich wieder hinsetzen muss, um dieses Buch für diese neue Version zu schreiben. Aber das mache ich gerne für Sie!

Und da es viele spannende Neuerungen vorzustellen gibt, ist die Seitenanzahl bei dieser Neuauflage ordentlich angestiegen. Sie werden sich vielleicht fragen, ob der Untertitel *Der schnelle Einstieg* zu einem Buch passt, das eine Stärke von über tausend Seiten aufweist. Die Antwort ist: und ob! Denn selbst in unserer schnelllebigen Zeit hat das Attribut *schnell* auch noch andere Bedeutungen als *rasch* oder *kurz*. Schlägt man den Duden auf, findet man unter dem Begriff *schnell* als erstes die beiden Verwendungen *schnellstens* und *so schnell wie möglich* vor. Diese beiden passen perfekt zum Charakter des Buches. Der Microsoft SQL Server ist ein so umfangreiches Produkt, dass ein rascher oder kurzer Einstieg gar nicht möglich sein kann. Ich bin vielmehr bemüht, durch die Auswahl der Themen und die Fokussierung auf in der Praxis relevante Schwerpunkte Sie so zu unterstützen, dass Ihr Einstieg schnellstens und so schnell wie möglich, und damit verbunden auch effizient, erfolgreich und angenehm erfolgen kann.

Der SQL Server 2017 kommt ja in sehr kurzem Abstand nach dem SQL Server 2016. Daher kommen in dieser Ausgabe Neuerungen beider Versionen zum Zug. Soweit es den SQL Server 2017 betrifft, ist die wohl unangefochten größte Neuerungen die Verfügbarkeit unter Linux. Dies ist auf Entwicklungen der letzten Jahre zurückzuführen, die zuvor absolut unvorstellbar und als Paradigmenbruch gegolten haben. So hat Microsoft in den letzten Jahren eine unheimlich umfassende Öffnung zu anderen Systemen vollzogen. Ist man vor einigen Jahren schon glücklich gewesen, wenn man auf einem Apple- oder Android-Smartphone ein Word-, Excel- oder PowerPoint-Dokument irgendwie zum Lesen anzeigen konnte, sind mittlerweile die Office-Anwendungen für viel Plattformen Standard. Mit der Öffnung der Produkte für nicht Windows-Plattformen ist ein Meilenstein in der Microsoft-Geschichte gesetzt worden, bei dem die Funktionalität und der Service im Vordergrund stehen. Parallel dazu sind die Angebote in der Cloud mit Azure derart umfangreich geworden, dass damit viele Anforderungen ohne Abstriche umgesetzt werden können. Gleichzeitig hat sich die Funktionalität und Usability von Web-Oberflächen – stelle man sich an dieser Stelle Office 365 vor – dermaßen verbessert, dass hier kaum noch Abstriche gegenüber einer Windows-Anwendung gemacht werden müssen.

Ende 2016 ist Microsoft Platinum-Mitglied der Linux-Foundation geworden und bekennt sich damit offiziell zu diesem Betriebssystem. Der SQL Server für Linux ist meiner Ansicht nach bislang die Krönung dieser Mitgliedschaft und der zuvor beschriebenen Entwicklungen. Schon davor ist der SQL Server die führende Datenbankplattform unter Windows gewesen, mit der Ausweitung auf Linux wird der Einsatzbereich noch einmal ganz deutlich vergrößert. Die Form der Implementierung ist insbesondere bemerkenswert, als der SQL Server nicht einfach für Linux nachgebaut worden ist. Die codegleiche Basisengine ist über eine neue Abstraktionsschicht auch unter dem freien Betriebssystem lauffähig geworden.

Servervirtualisierung ist unabhängig vom Betriebssystem State of the Art geworden und auch der Weg in die Cloud ist für Datenbanken an der Schwelle zur breiten Anerkennung. Die Virtualisierung und die Cloud sind endgültig auch bei der Datenbank angekommen. Dies hängt auch damit zusammen, dass sich die Virtualisierungsprodukte derart weiterentwickelt haben, dass Vorbehalte speziell für Datenbankserver nicht mehr bestehen. Es gibt es keine Nachteile mehr gegenüber einem physischen Server. Damit ist mit den Datenbanken eine der letzten Virtualisierungslücken bereits geschlossen. Ausnahmslos alle SQL Server bei meinen Kunden sind längst virtualisierte Server. Anwendungen in die Cloud auszulagern verliert langsam an Schrecken und Vorbehalte verschwinden. Mit dem Inkrafttreten der europäischen Datenschutzgrundverordnung (DSGVO) im Mai 2018 ist es besonders wichtig, dass die großen Cloud-Anbieter europäische Serverstandorte anbieten und die hundertprozentige Datenhaltung in Europa garantieren können.

Mit Windows Azure SQL-Datenbank steht eine einfach zu verwendende und leistungsstarke Cloud-Plattform für den SQL Server zur Verfügung, der Unternehmen den Betrieb eines Datenbankservers in kostengünstiger und effizienter Form ermöglicht. Um Themen wie Verfügbarkeit, Hardware und Skalierbarkeit müssen Sie sich dann keine Gedanken machen. Die Themen Virtualisierung und Cloud trennen die Entscheidungen für eine neue Server-Hardware und das Update der Datenbankversion voneinander. Ist der Umstieg auf eine neue Datenbankversion in der Vergangenheit mit dem Tausch der Server-Hardware einhergegangen, kann aufgrund der beschriebenen Entwicklungen ein Umstieg wesentlich zügiger vorstattengehen. Sie müssen nicht so lange auf den Einsatz der tollen neuen Features warten.

Die Neuerungen

Die Neuerungen des SQL Server 2016/2017 gegenüber ihrer Vorversion sind im Bereich der Datenbankengine auf mehrere Schwerpunkte fokussiert. Daten unter dem Schlagwort „In-Memory OLTP“ zur Gänze im Arbeitsspeicher zu halten ist als Feature ganz enorm verbessert worden, temporale Tabellen ermöglichen das Abfragen der Daten und deren Veränderung über die Zeit und die Datensicherheit steigt mit der Verschlüsselung von Daten und der Möglichkeit, Zugriffe auf Datensebene zu steuern. Zusätzlich bekommt der SQL Server 2017 mit dem SQL Server Management Studio 17 eine neue Version, welche auch die Nutzung des SQL Server unter Linux unterstützt. Auch wenn die altbekannten Clientprogramme auf Windows beschränkt bleiben, drängen neue Werkzeuge nach, die auch unter Linux und MacOS verfügbar sind. Diese sind das Visual Studio Code, das SQL Operations Studio und das Kommandozeilentool mssql-cli.

Verbesserte Werkzeuge für die Entwicklung unterstützen die Arbeit in einheitlicher Form für alle Plattformen. Die einheitliche Entwicklungsoberfläche stellt eines der Schwerpunkt-

themen dar. Die Bereiche Datenbank- und Anwendungsentwicklung wachsen immer näher zusammen. Sehen Sie sich das an, Sie werden sicher auch begeistert sein.

Für wen ist das Buch gedacht

Dieses Buch richtet sich an all diejenigen, die sich in SQL Server 2017 einarbeiten möchten. Es sind nicht nur Einsteiger in dieses Thema und dieses Produkt, sondern auch Umsteiger von MS Access und Softwareentwickler, die Datenbankkenntnisse für die Umsetzung ihrer Projekte benötigen. Das Buch ist bemüht, aus der Vielzahl an Möglichkeiten jene Themen herauszufiltern, die für das Arbeiten mit dem Produkt besonders wichtig sind und am häufigsten in der Praxis benötigt werden. Insofern habe ich für Sie mit der Auswahl der Inhalte eine Vorentscheidung getroffen, die Ihnen durch die Konzentration auf das Wesentliche den schnellen Einstieg erleichtern soll. Mit den in diesem Buch vermittelten Kenntnissen werden Sie in die Lage versetzt, effizient und umfassend mit dem neuen SQL Server zu arbeiten. Auch Umsteiger von früheren SQL Server-Versionen werden hier wertvolle Informationen für ihre weitere Arbeit mit dem Produkt finden. Schließlich sind nicht nur neue Features hinzugekommen, auch so manche altbekannte Funktionalität ist nun an einer anderen Stelle und manchmal unter einem neuen Namen anzutreffen. Dies ist vor allem für viele, die eine oder mehrere Versionen des SQL Servers übersprungen haben, eine wertvolle Hilfe.

Unter der Systemumgebung Windows hat der SQL Server mittlerweile die absolute Marktführerschaft bei Client-Server-Datenbanken erlangt. Ein großer Vorteil ist: Um auch anspruchsvolle Anwendungen zu realisieren, kann ein und dasselbe Datenbankmodul des SQL Servers plattformübergreifend verwendet werden: angefangen bei Notebooks unter Microsoft Windows 10 bis hin zu großen Multiprozessor-Servern unter Microsoft Windows Server 2016 Datacenter Edition. Mit dem SQL Server 2017 für Linux fällt für viele ein letzter Nachteil für den SQL Server bei der Auswahl eines Datenbanksystems weg.

Aufbau des Buches

Die Abschnitte des Buches sind so aufgebaut, dass Sie direkt an Ihrem Computer arbeiten und die Anwendungen unmittelbar durch Nutzung des SQL Servers ausprobieren und realisieren können. Zum Aufbau des Buches im Einzelnen:

Im **ersten Kapitel** gebe ich Ihnen einen Einstieg in die Leistungsmerkmale und Anwendungspotenziale des SQL Server 2017. Neben der Vorstellung der Editionen sowie der Erläuterung der Vorgehensweise zur Installation erfahren Sie, welche Voraussetzungen Ihr System für den Einsatz von SQL Server 2017 erfüllen muss.

Im **zweiten Kapitel** lernen Sie die Tools kennen, mit denen Sie auf den SQL Server zugreifen können. Sie benötigen diese, um den SQL Server zu verwalten und auf ihm Datenbanken zu erstellen, aber auch um mit ihm Anwendungen optimal entwickeln zu können. Hier kommen Sie erstmals mit dem SQL Server Management Studio in Kontakt, welches das wichtigste dieser Tools ist und sowohl für die Programmierung als auch die Administration eingesetzt wird.

Das **dritte Kapitel** befasst sich mit der Erstellung einer Datenbank, dem Anlegen von Tabellen und dem Einrichten von Beziehungen. Sie erfahren dabei, aus welchen Komponenten eine SQL Server-Datenbank besteht, und lernen gleichzeitig, Datenintegrität durch den Einsatz von Constraints zu implementieren. Der Einsatz von Datenbankdiagrammen, die nicht

nur zum Erstellen von Tabellen und Beziehungen dienen, sondern auch ein ideales Tool zur Dokumentation einer Datenbank sind, wird ebenso beschrieben. Die FileTables kommen in diesem Kapitel auch nicht zu kurz. Kopieren Sie Dateien in einen Ordner auf einem Netzwerk-Share, und schon tauchen diese automatisch wie von Geisterhand in der Datenbank auf.

Im Regelfall wollen Sie nicht ausschließlich Daten in eine Datenbank einpflegen, sondern natürlich Informationen auch wieder aus dem System entnehmen. Zu diesem Zweck erfahren Sie im **vierten Kapitel**, wie Sie effizient durch den Einsatz von Abfragen, Sichten und SQL-Anweisungen auf Daten zugreifen. Sie erhalten dabei auch einen kompakten Überblick über die wichtigen Sprachbereiche und Anweisungen von SQL (Structured Query Language).

Kapitel 5 bietet Ihnen einen Überblick über die Datenbanksprache Transact-SQL, die Ihnen sowohl bei der Datenbankprogrammierung als auch bei der Verwaltung von Datenbanken wertvolle Dienste leistet. So können alle Aufgaben, die Sie mit einem grafischen Verwaltungstool erledigen, auch direkt über diese Sprache realisiert werden. Dadurch können Sie solche Aufgaben in Ihre Applikationen einbauen oder sich Ihre eigenen Verwaltungstools zusammenstellen. Dieses Kapitel erläutert Ihnen die Sprachkomponenten und die dabei verwendeten Strukturen. In der Übersicht der wichtigsten Funktion finden Sie auch jene, die beim SQL Server 2017 neu hinzugekommen sind.

Nach der allgemeinen Einführung in Transact-SQL lesen Sie in **Kapitel 6**, wie Sie diese Sprache zur Programmierung von gespeicherten Prozeduren (Stored Procedures) einsetzen. Durch den gezielten Einsatz solcher Prozeduren bilden Sie die datenbezogenen Vorgänge Ihrer Datenbankapplikation auf dem Server ab. Diese müssen dann von den verschiedenen Client-Programmen nur noch aufgerufen werden. So realisieren Sie effiziente Client-Server-Applikationen.

Transact-SQL wird aber auch zur Programmierung von Triggern verwendet, die es Ihnen erlauben, Automatismen in Ihre Datenbank zu integrieren, die auf das Einfügen, Ändern und Löschen von Datensätzen reagieren. Besonders interessant für die Praxis sind mittlerweile auch Datenbanktrigger, mit denen Sie sowohl Änderungen an der Datenbankstruktur überwachen als auch bei Bedarf unterbinden können. Des Weiteren lernen Sie die benutzerdefinierten Funktionen (User-Defined Functions, UDFs) kennen. Diese Funktionen können im Gegensatz zu gespeicherten Prozeduren auch in SQL-Anweisungen eingesetzt werden und erweitern dadurch den Einsatzbereich in der Programmierung von Transact-SQL. Sie können sie darüber hinaus auch verwenden, um die Standardfunktionen vom SQL Server zu erweitern.

Das **Kapitel 7** beschäftigt sich mit dem Thema .NET im Zusammenhang mit dem SQL Server. Sie lesen hier nicht nur, wie Sie Prozeduren, Funktionen und Trigger mit einer .NET-Programmiersprache für die SQL Server CLR (Common Language Runtime) entwickeln, sondern auch, wie Sie Aggregatfunktionen selbst programmieren. Diese stehen Ihnen dann innerhalb von SQL-Anweisungen wie andere Aggregatfunktionen zur Verfügung. Ein wesentliches Augenmerk habe ich dabei auf die neuen Sicherheitsanforderungen, die für das Ausführen von CLR-Code ab dem SQL Server 2017 erfüllt sein müssen, gelegt. Lesen Sie dazu in diesem Kapitel, wie Sie Ihren Programmcode mit Zertifikaten signieren.

Die Server Management Objects (SMO), mit denen Sie auf so gut wie alle Funktionalitäten des SQL Servers programmatischen Zugriff haben, runden das Kapitel ab. Durch die SQL Server Data Tools wird die Programmierung für die SQL Server CLR interessant, da dazu ein extrem leistungsstarkes und dazu noch freies Werkzeug verwendet werden kann.

Die SQL Server Data Tools revolutionieren für Entwickler die Arbeit mit der Datenbank. Daher sind sie es mir wert, gemeinsam mit den Datenebenenanwendungen in **Kapitel 8** behandelt zu werden. Datenebenenanwendungen, oder Data Tier Applications, wie sie im Original genannt werden, sind mittlerweile schon fast schon integraler Bestandteil für viele Phasen der Datenbankentwicklung. Sie sind das Werkzeug, um Datenbanken auszurollen und Aktualisierungen und Versionierung zu organisieren. Sie sind in die SQL Server Data Tools fest integriert. Die Data Tools sind ein Werkzeug, mit dem es für Programmierer möglich ist, unter dem Dach des Visual Studios mit einem Werkzeug alle Entwicklungsaufgaben von der Datenbank bis zum Frontend zu erledigen.

Da Sie von einer Datenbank nicht viel haben, wenn Ihre wertvollen Daten nicht sicher sind, erfahren Sie in **Kapitel 9**, wie Sie eine SQL Server-Datenbank regelmäßig sichern und im Ernstfall auch wiederherstellen können. Datenbanksicherungen haben ihre Bedeutung aber nicht nur in einem Störfall, sondern sind auch in der täglichen Arbeit mit der Datenbank wichtig, weil sie zum Beispiel auch dafür verwendet werden, eine Datenbank von einem Server auf einen anderen zu übertragen. Besonders begeistert bin ich von den temporalen Tabellen, die ohne Programmieraufwand Veränderungen der Daten über die Zeit verfügbar machen.

In **Kapitel 10** finden Sie alle Informationen, die Sie für die Herstellung der Sicherheit Ihrer Datenbank benötigen. Sie lesen in diesem Kapitel, wie Sie auf Ihrem SQL Server Benutzer anlegen und diesen verschiedene Berechtigungen zuweisen. Sie erfahren, wie Sie Contained Databases einsetzen und nutzen können. Lesen Sie in diesem Kapitel auch, wie Sie den Zugriff auf Zeilenebene einschränken und Ihre Daten mit Always Encrypted verschlüsseln können. Damit können Sie den Anforderungen der europäischen Datenschutzgrundverordnung (DSGVO) mit dem SQL Server noch besser gerecht werden.

In **Kapitel 11** erläutere ich Ihnen zwei erweiterte Funktionalitäten, die Ihnen ergänzend zur Verfügung stehen, falls Sie nicht die Gratis-Edition des SQL Servers 2017 verwenden. Ich stelle Ihnen hierbei Datenbank-E-Mail sowie die Integration Services etwas genauer vor.

In **Kapitel 12** zeige ich Ihnen am Beispiel von Ubutu, wie Sie einen SQL Server unter Linux installieren und einsetzen.

Mit diesem Buch lernen Sie anhand von problembezogenen Aufgabenstellungen in anschaulicher und systematischer Form die zahlreichen Möglichkeiten des SQL Server 2017 für die Datenbankentwicklung kennen. Das Buch eignet sich sowohl zum Selbststudium als auch als begleitende Unterlage für Schulungen.



www.downloads.hanser.de

Hier finden Sie sämtliche Dateien aller im Buch verwendeten Beispiele. Diese enthalten u. a. die Beispiel-Datenbanken, SQL-Skripte zu jedem Kapitel sowie Visual Studio-Projekte.

Ich möchte mich an dieser Stelle bei meinem Dreimäderlhaus – Petra, Alina und Lea – für ihre immense Geduld bedanken.

Und nun viel Erfolg beim schnellen Einstieg in die Arbeit mit dem SQL Server 2017.

Klemens Konopasek, Gössendorf/Graz

Icons

In diesem Buch werden verschiedene Icons verwendet, deren Bedeutung Sie hier finden.



HINWEIS: Mit diesem Symbol soll auf interessante Informationen besonders hingewiesen werden.



PRAXISTIPP: Mit diesem Symbol sind Informationen gekennzeichnet, mit denen Sie sich das Leben leichter machen können.



ACHTUNG! Sehen Sie dieses Icon, finden Sie Informationen, wie Sie etwas nicht machen oder worauf Sie ein ganz besonderes Augenmerk legen sollten.

1

Der SQL Server 2017 stellt sich vor

Der SQL Server 2017 ist da – wieder eine tolle neue Version. Auch wenn diese Version sehr rasch nach dem SQL Server 2016 erschienen ist, der als Hauptrelease bezeichnet wird, ist der SQL Server 2017 als vermeintliche Zwischenversion alles andere als unbedeutend. Während der Entwicklung als SQL Server vNext bezeichnet, ist es doch erstmals in der Geschichte des MS SQL Server soweit, dass dieser neben Windows auch für Linux als Betriebssystem verfügbar ist. Und das ist eine echte Revolution. Da diese Versionen so rasch aufeinanderfolgen, gehe ich in dieser Auflage auch auf Features ein, die mit dem SQL Server 2016 eingeführt worden sind.

Ich bin von den Neuerungen der Versionen 2016 und 2017 begeistert und ertappe mich immer wieder, wenn ich bei einem Kunden noch eine der Vorversionen vorfinde, bei dem Gedanken: „Oje, jetzt muss ich wieder auf dieses und jenes verzichten.“ Mein persönliches Highlight ist die stark verbesserte Möglichkeit, Tabellen einer Datenbank im Arbeitsspeicher zu halten. Sehr spannend finde ich auch die neuen Sicherheitsfunktionen. Wird die Version 2016 von Microsoft selbst doch als das „Performance und Sicherheits-Release“ bezeichnet. Und nun ist dies alles mit dem SQL Server 2017 auch für Linux verfügbar. Viele vermeintliche kleinere Features sind aber groß in der Auswirkung, so ist für Entwickler die native Unterstützung von JSON von großer Bedeutung.

Ich hoffe, auch Sie gehen mit Freude an den SQL Server 2017 und an dieses Buch heran!

Im ersten Kapitel möchte ich Ihnen einen Überblick über das Produkt und seine Komponenten geben. Anschließend stelle ich Ihnen die Editionen vor, in denen der SQL Server 2017 verfügbar ist, und zeige Ihnen, wie Sie bei der Installation vorgehen. Darüber hinaus werden Sie erfahren, wie Sie mit dem SQL Server 2017 arbeiten, um vorhandene Datenbanken und darin enthaltene Datenbankobjekte zu nutzen. Ebenso zeige ich Ihnen, wie eine Integration zu Client-Umgebungen erfolgen kann. Den Abschluss dieses Kapitels bilden die Besonderheiten der freien Express-Version.

■ 1.1 SQL Server – wer ist das?

Eigentlich wollte ich diesen Abschnitt ursprünglich mit „SQL Server – was ist das?“ betiteln. Aber das kam mir dann so plump vor, dass ich das „was“ durch ein „wer“ ersetzt habe. Dies klingt besser, auch wenn ich den SQL Server dadurch nicht personifizieren will.

1.1.1 Der SQL Server im Konzert der Datenbanksysteme

Wenn wir heutzutage von einer Datenbank sprechen, meinen wir in der Regel – ohne dies explizit zu erwähnen – eine relationale Datenbank. Andere Datenbanksysteme, wie zum Beispiel objektorientierte Datenbanken, konnten sich nie wirklich auf breiter Front durchsetzen oder haben ihre beste Zeit bereits hinter sich. Neue moderne Ansätze, die sich unter dem Begriff NoSQL finden, sind für sehr spezielle Anwendungsbereiche ausgerichtet und zielen darauf ab, relationale Datenbanken zu verdrängen oder gar zu ersetzen. Vielmehr wollen sie eine Ergänzung in Nischenbereichen sein, für die relationale Strukturen nicht die ideale Form sind. Daher steht das *No* auch nicht für *kein*, sondern für *not only*. Vielfach kommen diese auch aufs Tapet, wenn von Big Data die Rede ist. Microsoft trägt dem durch die Möglichkeit von hybriden Lösungen mit Hadoop oder R Server Rechnung.



HINWEIS: Relationale Datenbanksysteme entwickeln sich in die Richtung weiter, dass wichtige Features aus dem nicht-relationalen Bereich als Zusätze in die Produkte integriert werden. So bietet der SQL Server 2017 als neues Feature erstmals die Unterstützung von Graph-Daten, eine wichtige Datenart bei NoSQL-Systemen.

Beim „schnellen Einstieg in den SQL Server“ gehen wir von relationalen Datenbanksystemen aus und unterteilen diese in

- *Desktop-Datenbanksysteme* und
- *Server-Datenbanksysteme*.

Eine Datenbankanwendung besteht aus drei Komponenten:

- *Data Layer:* Der Data Layer hat die Aufgabe, Daten zu verwalten und zu speichern. Hier werden außerdem die Strukturen der Datenspeicherung definiert. Diese Aufgabe wird von der Datenbank-Engine wahrgenommen.
- *Program Layer:* Im Program Layer werden die Logiken und Abläufe des Datenzugriffs abgebildet. Hier kommen unterschiedliche Entwicklungsumgebungen zum Einsatz.
- *Presentation Layer:* Aufgabe des Presentation Layers ist es, Ausgaben aus der Datenbank darzustellen. Hierzu gehören insbesondere Benutzeroberflächen und Frontend-Komponenten, mit denen der Benutzer interagiert.

Das Hauptmerkmal eines Desktop-Datenbanksystems besteht darin, dass alle drei Komponenten auf dem Desktop anzutreffen sind. Insbesondere läuft auch die Datenbank-Engine auf dem Desktop. Werden Datenbanken eines desktopbasierten Systems auf dem Server

abgelegt, wird vom Server lediglich der File-Service genutzt, um die Daten remote zur Verfügung zu stellen.

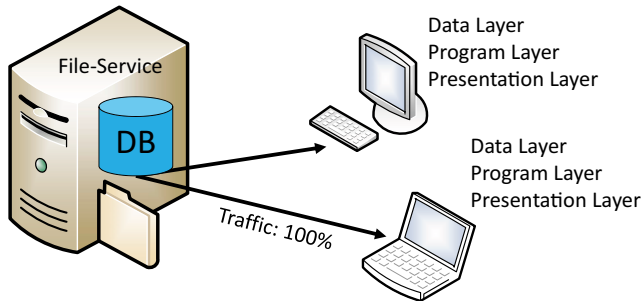


Bild 1.1 Konzept von Desktop-Datenbanken

Ein wesentliches Merkmal eines desktopbasierten Datenbanksystems ist, dass alle datenbankrelevanten Vorgänge auf dem Client ablaufen. Dazu müssen alle Daten vom Server auf den Client transferiert werden, damit die Daten von der lokalen Datenbank-Engine verarbeitet werden können.

Server-Datenbanksysteme hingegen verwenden eine Datenbank-Engine auf dem Server. Von den Clients werden Anfragen an diesen Dienst gestellt, die auf dem Server verarbeitet werden. Dadurch werden nicht alle Rohdaten, sondern nur die Ergebnisse der Anfrage an den Client gesendet. Es findet sozusagen eine Spezialisierung der Aufgaben der Datenverwaltung auf dem Server statt.

In der Abbildung ist der *Program Layer* beiden Komponenten zugeordnet, da Elemente von diesem auch in beiden Komponenten auftreten können. Wir werden später in diesem Buch zwischen serverseitiger und clientseitiger Datenbankprogrammierung unterscheiden.

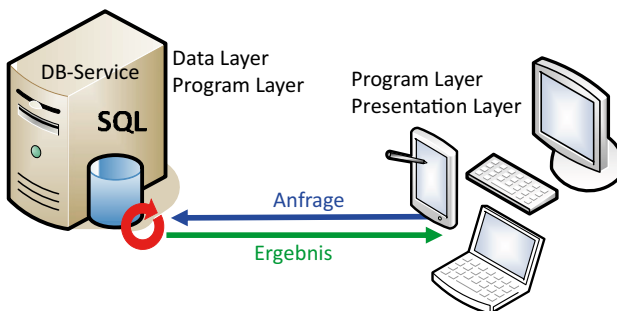


Bild 1.2 Konzept von Server-Datenbanksystemen

In der Kategorie der Desktop-Datenbanksysteme ist vor allem Microsoft Access weit verbreitet. Der SQL Server ist auch als Desktop-Datenbanksystem mit der *LocalDB* genannten Edition vertreten. Diese kann in lokal installierte Anwendungen integriert und weitergegeben werden und ist daher vor allem bei Visual Studio-Entwicklern bekannt. Ebenso in diese Kategorie einzuordnen ist SQLite, die als Embedded-DB ebenso in unzählige Anwendungen lokal integriert ist.

In der Kategorie der Server-Datenbanksysteme sind neben dem Microsoft SQL Server vor allem folgende Produkte von Bedeutung:

- Oracle
- DB2 von IBM
- SAP ASE (früher Adaptive Server Enterprise von Sybase, noch früher Sybase SQL Server)

Als Open-Source-Datenbanksysteme sind zusätzlich von Bedeutung:

- PostgreSQL
- MySQL/MariaDB

Der SQL Server ist das führende serverbasierte Datenbanksystem auf Windows-Plattformen. Bisher sind ja nur die anderen genannten Systeme auch für diverse andere Plattformen verfügbar gewesen.



HINWEIS: Der SQL Server 2017 ist erstmals für Linux verfügbar. Dies läutet einerseits eine neue Ära für den SQL Server ein, andererseits ist das die logische Konsequenz der generell von Microsoft vollzogenen Öffnung der letzten Jahre zu anderen Systemen. Wie weit der SQL Server auch auf diesen Plattformen eine bedeutende Stellung erhalten wird, wird die Zukunft zeigen.

Informationen über den SQL Server 2017 für Linux finden Sie in Kapitel 12.

ACID – das Konsistenzmodell relationaler Datenbanken

Relationale Datenbanken verwenden das Konsistenzmodell ACID. Bei diesem Modell steht die Datenkonsistenz absolut im Vordergrund und ist somit die oberste Maxime. Wenn wir uns die vier Säulen dieses Modells ansehen, werden wir feststellen, dass die Forderungen dieses Modells bei relationalen Desktop-Datenbanken wie Microsoft Access allerdings nicht erfüllt sind. Bei serverbasierten Datenbanken wie dem Microsoft SQL Server sind sie natürlich erfüllt. Die vier Säulen dieses Konsistenzmodells zeigt Bild 1.3.

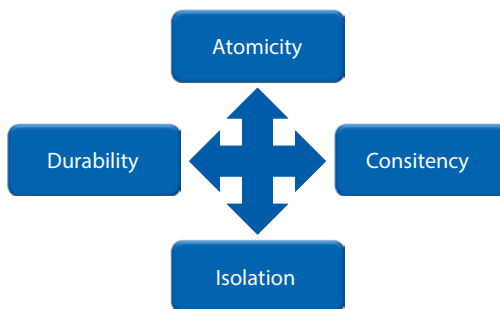


Bild 1.3 Das ACID-Konsistenzmodell

Was bedeuten diese Begriffe im Einzelnen und durch welche Mechanismen werden sie umgesetzt?

- **A – Atomicity:** Zusammenhängende Vorgänge werden entweder zur Gänze oder gar nicht durchgeführt. Gehören mehrere Schreibzugriffe zu einem gemeinsamen Vorgang, werden alle Änderungen erst übernommen, wenn auch der letzte Teilschritt erfolgreich abgeschlossen worden ist. Ist dies aus welchem Grund auch immer nicht möglich, müssen alle bisher vorgenommenen Schritte vollständig wieder rückgängig gemacht werden. Das Werkzeug, um diese Vorgabe zu erreichen, sind *Transaktionen*.
- **C – Consistency:** Die Vorgabe der Konsistenzhaltung legt fest, dass der Übergang von einem konsistenten Zustand immer nur in einen anderen konsistenten Zustand erfolgen darf. Daten müssen also immer in einem vollständigen Zustand vorliegen, es darf nie Verweise auf nicht vorhandene Daten geben. Die *Referenzielle Integrität* sorgt dafür, dass dieses Ziel erreicht wird.
- **I – Isolation:** Die Forderung der Isolation besagt, dass alle Vorgänge von anderen unbeeinflusst abgegrenzt ablaufen dürfen. Die gleichen Daten können nie zeitgleich von mehreren Personen oder Prozessen geändert werden. Solange Änderungen nicht abgeschlossen sind, sind die betroffenen Daten zumindest für den Schreibzugriff für andere gesperrt. Die Änderungen sind für den Durchführenden sofort sichtbar, für alle anderen erst nach Abschluss des Vorgangs. Auch dafür sind *Transaktionen* zuständig.
- **D – Durability:** Unter der Dauerhaftigkeit versteht man, dass Daten, die einmal festgeschrieben worden sind, dauerhaft verfügbar sind und auch Strom- und andere Systemausfälle überstehen. Dieses Ziel kann durch den Einsatz von *Protokollierung* erfolgen.

Die erwähnten Mechanismen Transaktion, referenzielle Integrität und Protokollierung werden Sie in den entsprechenden Kapiteln dieses Buches im Detail erläutert finden. Besonders interessant ist ACID im Zusammenhang mit den speicheroptimierten Tabellen, die im RAM des Servers gehalten werden. Auf den ersten Blick würde man vermuten, dass diese vor allem im Hinblick auf die Dauerhaftigkeit problematisch sind. Allerdings werden Sie lesen, dass diese Tabellen über eine entsprechende Option bei der Erstellung auch ACID-konform eingesetzt werden können. Dies wird durch das zusätzliche Ablegen der Daten auf den Disks erzielt.

1.1.2 Entscheidungsszenarien für Datenbanksysteme

Wenn Sie vor der Entscheidung stehen, ein Datenbanksystem auszuwählen, gilt es, verschiedene Gesichtspunkte zu berücksichtigen. Ich möchte Ihnen in einem kurzen Überblick die aus meiner Sicht wichtigsten Entscheidungsgründe nennen.

- *Preis (TCO):* Bei der Betrachtung der Kosten werden häufig fälschlicherweise lediglich die direkten Lizenzkosten angesetzt. Wesentlich zielführender wäre es allerdings, den Ansatz *TCO (Total Cost of Ownership)* zu wählen; denn neben den Lizenzkosten fallen zum Beispiel auch die folgenden Kosten an:
 - Kosten für Hardware
 - Kosten für Schulungen. Hierbei ist auch die Anzahl der zu schulenden Personen zu berücksichtigen. Sollen viele Personen mit einem System umgehen können oder sollen es Spezialisten für Sie erledigen?

- Kosten aufgrund von Ineffizienz, da Personen, ohne entsprechend geschult zu sein, sich statt mit ihrer eigentlichen Arbeit mit Lösungen im Desktopbereich beschäftigen.

Man kann hier keine generelle Empfehlung für ein desktop- oder serverbasiertes System aussprechen. Dies muss in der speziellen Situation beurteilt und entschieden werden.

- **Datenmenge:** Serverbasierte Systeme sind in der Lage, wesentlich größere Datenmengen zu speichern und effizienter zu verwalten als desktopbasierte Systeme.
- **Benutzeranzahl:** Nicht nur die theoretische Benutzeranzahl ist bei Serversystemen höher. Können bei Access beispielsweise theoretisch 255 Benutzer gleichzeitig auf eine Datenbank zugreifen, würde ich die tatsächliche Grenze mit 20 bis 30 gleichzeitig angemeldeten Benutzern schon als hoch angesetzt sehen. Dies ist aus der Topologie leicht zu erklären. Stellen Sie sich vor, in einem Lokal würden sich alle Kellner um einen Zapfhahn scharen und versuchen, Bier zu zapfen. Das entspricht der Logik eines Desktopsystems. Wesentlich effizienter wäre es, nur eine Person an den Zapfhahn zu stellen, die Bestellungen bearbeitet und die gezapften Biere dann an alle Kellner verteilt. Dies würde ungefähr einem serverbasierten Datenbanksystem entsprechen. Wahrscheinlich werden bei der zweiten Variante mehr Biere in der gleichen Zeit in durstigen Kehlen landen. Daher sehe ich hier klare Vorteile für ein serverbasiertes System.
- **Portabilität:** Eine Desktop-Datenbank, die oft aus einer einzigen Datei besteht, kann sehr leicht beispielsweise auf ein Notebook transferiert werden. Dies funktioniert bei einem serverbasierten System nicht so ohne Weiteres. Ersetzt man allerdings den Begriff Portabilität durch *Zugriff von überall*, könnte man darunter verstehen, auf eine Datenbank remote über eine Webapplikation zuzugreifen. Dafür wäre wiederum eine Serverdatenbank besser geeignet.
- **Flexibilität:** Eine besondere Stärke eines Desktop-Datenbanksystems liegt in der Flexibilität und Einfachheit der Anwendung. Daher wird es gerne verwendet für:
 - Auswertungen (zum Beispiel werden häufig von großen Server-Datenbanksystemen Daten importiert und danach in einem Desktop-Datenbanksystem ausgewertet),
 - Prototyping oder
 - Klein- und Kleinstlösungen.
- **Transaktionen:** Transaktionen sind für konsistente Daten unerlässlich. In der Regel werden diese nur von serverbasierten Systemen geboten.
- **Sicherheit:** Sicherheit ist unter zwei Gesichtspunkten zu betrachten.
 - Die *Zugriffssicherheit* legt fest, wer mit welchen Daten was tun darf.
 - Die *Datensicherheit* legt fest, wie sicher Daten vor Verlust geschützt sind.In beiden Bereichen liegen die Vorteile ganz klar und eindeutig bei Server-Datenbanksystemen, die hierzu spezielle Features anbieten.
- **Backup und Recovery:** Server-Datenbanksysteme ermöglichen Sicherungen im Vollbetrieb und häufig auch das verlustfreie Wiederherstellen exakt bis zum Zustand vor einem Crash. Dies gilt nicht für eine Desktop-Datenbank, bei der diese zunächst alle Anwender verlassen müssen.
- **Netzlaster:** Aufgrund der Topologie, dass nur das Ergebnis einer Anfrage vom Server an den Client übertragen wird, der diese Daten dann anzeigt und verarbeitet, können serverba-

sierte Systeme auch über schwächere Leitungen performant betrieben werden. Eine vorgegebene Bandbreite erlaubt eine größere Anzahl an Benutzern.

- *Stabilität und Verfügbarkeit:* Serversysteme verfügen über Mechanismen, welche die Verfügbarkeit der Datenbank nach dem Prinzip 24-7-365 (24 Stunden am Tag, 7 Tage die Woche und 365 Tage im Jahr verfügbar) ermöglichen.
- *Skalierbarkeit:* Durch den Einsatz unterschiedlicher Editionen ermöglichen Server-Datenbanken ein stufenloses Skalieren einer Lösung von einer kleinen Abteilungslösung bis hin zu Konzernlösungen.

Analysieren Sie Ihre Anforderungen an ein Datenbanksystem anhand dieser Kriterien und treffen Sie dann Ihre Entscheidung.



HINWEIS: Der Microsoft SQL Server bietet ein professionelles Server-Datenbanksystem zu einem vergleichsweise günstigen Preis. Mit den Editionen von Express bis Enterprise werden alle Bedürfnisse bedient; daneben erlauben sie ein uneingeschränktes Wachsen der Datenbank. Bereits ab der Express Edition können Sie die Vorteile von Sicherheit, Stabilität, Transaktionen und geringer Netzlast nutzen. Zudem ist Microsoft SQL Server ein Tool, das einfach und flexibel in der Handhabung ist wie kaum ein vergleichbares System. Außerdem sind in den letzten Versionen immer mehr Features, die nur in lizenzierten Editionen verfügbar gewesen sind, auch in die Express Edition integriert worden. Neue Features wie temporale Tabellen sind beispielsweise von Beginn an auch in der Express Edition verfügbar.

Wenn Sie sich nicht mit der Konfiguration und Wartung des SQL Servers selber auseinandersetzen möchten, können Sie den SQL Server auch in der Cloud mit Azure DB nutzen. Achten Sie aber aufgrund der Europäischen Datenschutzgrundverordnung (DSGVO) darauf, dass Ihre Daten dabei stets ausschließlich in einem Rechenzentrum innerhalb der EU gehostet werden.

1.1.3 Komponenten einer Datenbankanwendung

In der Praxis benötigen Sie keine Datenbank, sondern eine Datenbankanwendung. Auch wenn die Datenbank als „Motor“ einer Anwendung oft die wichtigste Komponente darstellt, ist ein Motor ohne ein Chassis oft nur wenig von Nutzen. Das Chassis ist die Anwendung, die aus einer Datenbank eine Datenbankanwendung macht. Eine Anwendung wird mit einer Entwicklungsumgebung erstellt und greift über standardisierte Schnittstellen mithilfe von SQL auf ein Datenbanksystem zu. Einen Überblick über einsetzbare Programmiersprachen und Schnittstellen zeigt Bild 1.4.

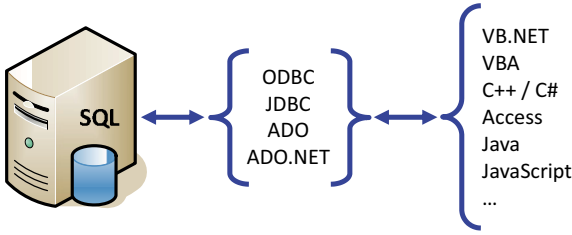


Bild 1.4 Zugriff auf eine Datenbank über Standardschnittstellen

Eine Datenbankanwendung besteht in der Regel aus folgenden Komponenten:

- Datenbankmanagementsystem als Backend für die Verwaltung der Daten
- User-Interface als Frontend für die Bedienung der Anwendung
- Server- und/oder clientseitige Programmierung für die Abbildung von Logiken

Bild 1.5 zeigt eine schematische Darstellung der einzelnen Komponenten und ihr Zusammenspiel.

HINWEIS: Der SQL Server übernimmt in diesem Szenario die Rolle des Datenbankmanagementsystems, auf das mithilfe der Abfragesprache SQL über standardisierte Schnittstellen zugegriffen wird. Für performante Lösungen ergänzt serverseitige Programmierung mittels Transact-SQL und .NET die Datenbankentwicklung mit dem SQL Server.

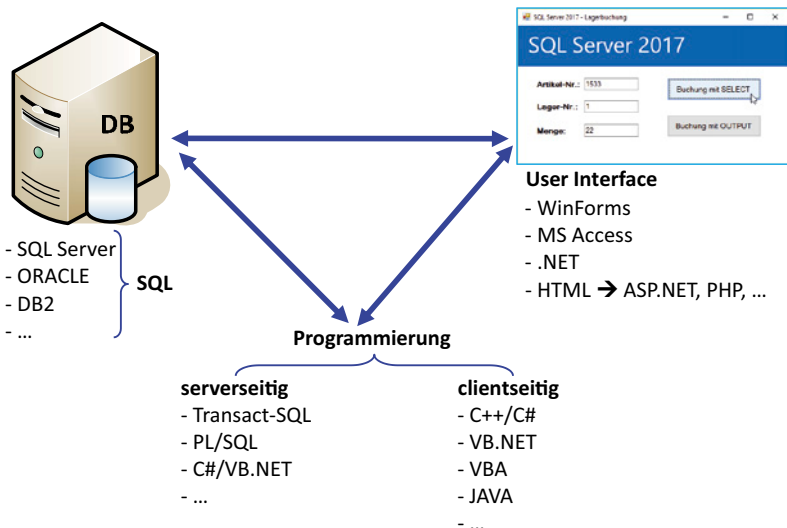


Bild 1.5 Datenbankanwendung und ihre Bestandteile

So lernen Sie den SQL Server in diesem Buch kennen:

- Den SQL Server installieren und konfigurieren
- Datenbanken und Datenbankobjekte mit dem SQL Server erstellen
- Den Zugriff auf Daten mit der Structured Query Language (SQL) vollziehen
- Serverseitig mit Transact-SQL und .NET programmieren
- Die Benutzerverwaltung zur Vergabe von Berechtigungen nutzen
- Sicherung und Wiederherstellung von Datenbanken durchführen
- Erweiterte Features einsetzen

Programmierung im Frontend und Backend

In einer Datenbankanwendung kann sowohl eine Programmierung im Frontend als auch im Backend erfolgen. Im Frontend müssen sämtliche Vorgänge im Zusammenhang mit der Benutzerführung programmiert werden. Manche Vorgänge können aber wahlweise im Frontend oder im Backend programmiert werden. Dies sind vor allem Vorgänge mit Datenbezug.

Die beiden nachfolgenden Abbildungen zeigen die Unterschiede beim Programmablauf von Programmcode, der auf dem Client oder auf dem Server läuft.

Bei clientseitiger Programmierung ist die gesamte Programmlogik im Frontend untergebracht. Werden im Ablauf Informationen aus der Datenbank benötigt oder sind Daten in die Datenbank zu schreiben, werden SQL-Anweisungen zum Datenbankserver geschickt. Mit den Ergebnissen dieser Anweisungen arbeitet der Programmcode anschließend weiter. Ein Programmablauf kann oft aus sehr vielen Einzelschritten bestehen, bei denen mitunter auch sehr viele Datenzugriffe nötig sind.

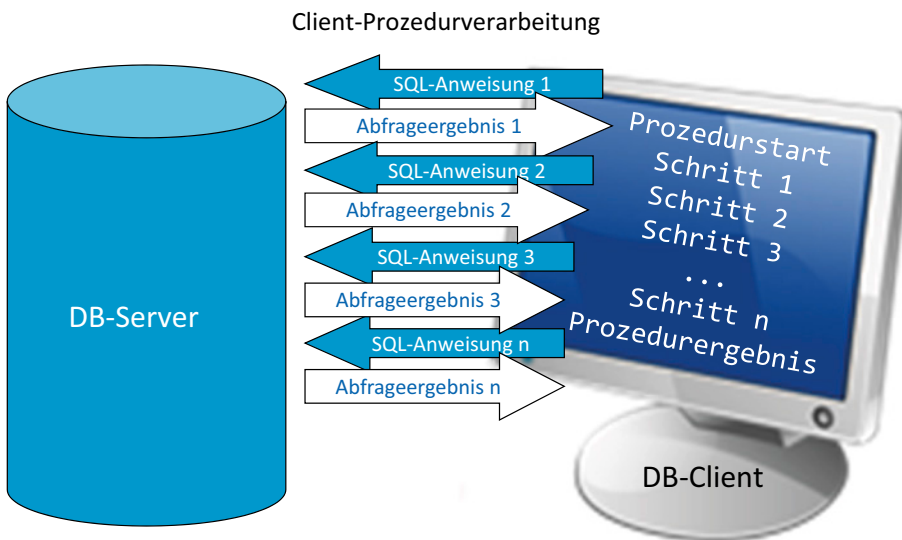


Bild 1.6 Programmlogik im Frontend

Bei serverseitiger Programmierung wird die Programmlogik beispielsweise mithilfe gespeicherter Prozeduren (Stored Procedures) im Backend umgesetzt. Der Vorteil besteht darin, dass das „Hin und Her“ zwischen Frontend und Backend entfällt. Im Frontend wird lediglich die am Server hinterlegte Funktionalität aufgerufen und das Ergebnis abgearbeitet.

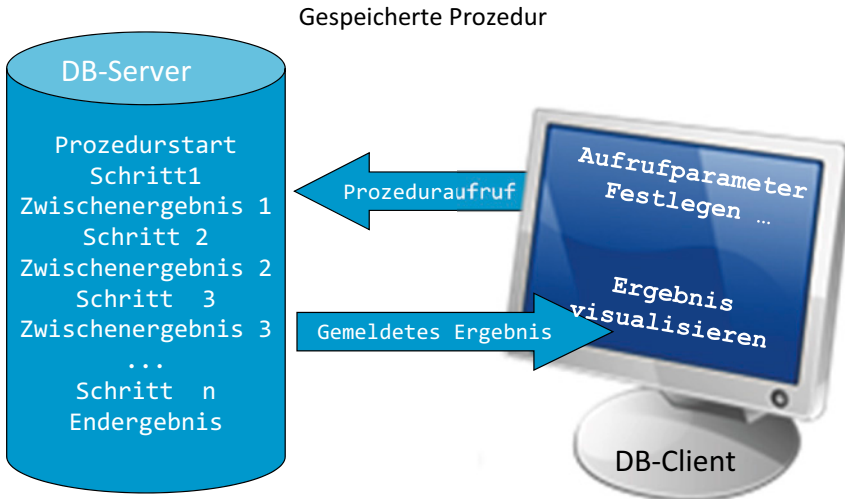


Bild 1.7 Serverseitige Programmierung

In den Kapiteln 4 bis 6 wird das Thema „Serverseitige Datenbankprogrammierung“ im Detail behandelt und auf Vor- sowie Nachteile eingegangen. Clientseitige Programmierung ist nicht Thema dieses Buches, da sie nicht vom SQL Server, sondern von der eingesetzten Programmiersprache und Entwicklungsumgebung abhängt. Anhand praktischer Beispiele, die zeigen, wie Programmierelemente des Servers von clientseitigem Code aufgerufen werden, streifen wir jedoch die clientseitige Programmierung.

1.1.4 SQL Server – das Gesamtkonzept

Der SQL Server beschränkt sich keinesfalls auf die Datenbank-Engine. SQL Server ist mittlerweile eine komplette Produktfamilie, die sich um den Kern schart. Damit ist der SQL Server nicht nur ein reines Datenbanksystem. Er bietet auch Lösungen für viele Anwendungen im Datenbanksystem.

Zur Datenbank-Engine selber zählen folgende Features:

- Volltextsuche
- Datenbankreplikation

Die Zusatzprodukte, oft unter dem Begriff *Business Intelligence (BI)* zusammengefasst, sind folgende Dienste:

- *Integration Services*: Die Integration Services (IS) sind ein umfassendes Werkzeug, um zum Beispiel Daten von A nach B zu transferieren. Dabei sind komplexe Workflows mit Verzweigungen und unzähligen Möglichkeiten realisierbar.

- *Reporting Services*: Aufgabe dieser Services ist es, Berichte, die auf Daten aus der Datenbank basieren, in verschiedenen Formen zur Verfügung zu stellen. Das kann zum Beispiel eine HTML-Seite oder ein PDF-Dokument sein, das per E-Mail verschickt wird. Ziel ist es, das gesamte Berichtswesen eines Unternehmens abbilden zu können. Daher sind diese Berichte auch nicht statisch. Vielmehr erlauben sie es einem Benutzer, durch die Eingabe von Parametern das Ergebnis zu verändern oder über einen definierten Drill-Down immer detailliertere Daten abzurufen. Ein wichtiger Bestandteil der Reporting Services ist neben der Berichtserstellung die Berichtsverteilung. Reporting Services lassen sich sehr gut in Share Point integrieren.
- *Analysis Services*: Diese dienen der Realisierung von Data-Warehouse-Lösungen. Geschäftsleitung, Controller und Marketingmanager benötigen immer anspruchsvollere Analysen und Trendinformationen. Die Basis dafür liegt zu einem Großteil in den bereits auf Servern gespeicherten Unternehmensdaten. In der Praxis werden zur Lösung dieser Aufgabenstellung OLAP-Systeme (Online Analytical Processing; deutsch: analytische Online-Verarbeitung) benötigt, indem auf einfache Weise Informationszusammenstellungen aus OLTP-Daten erstellt werden, die dann für anspruchsvolle Datenanalysen genutzt werden können. Die Analysis Services bieten diese Funktionalität auf einem sehr hohen Niveau und haben den SQL Server in diesem Bereich zu einem der führenden Produkte gemacht.
- *Service Broker*: Dieser Service zielt auf große verteilte Anwendungen ab. Der Service Broker verwaltet Warteschlangen, die mit SQL-Anweisungen „gefüttert“ werden können. Die Inhalte der Warteschlange werden dann der Reihe (englisch: queue) nach abgearbeitet. Diese Warteschlangen können nicht nur am lokalen Server positioniert sein, sondern auch remote abgearbeitet werden.

Anwendungen, die auf dem Prinzip von Warteschlangen basieren, setzen auf einem anderen Anwendungsverständnis auf, als wir es in der Regel gewohnt sind. Schauen wir uns folgendes Beispiel an: Viele von Ihnen haben sicher schon einmal eine Domänenregistrierung vorgenommen. Wenn Sie eine Domäne registrieren möchten, ist der erste Schritt üblicherweise, dass Sie ermitteln, ob die gewünschte Domäne noch verfügbar ist. In einer Online-Applikation würden Sie eine Schaltfläche anklicken, und die Domäne würde Ihnen gehören. So einfach ist es aber bekanntlich nicht. Sie reichen stattdessen den Antrag bei einer akkreditierten Registrierungsstelle ein. Und hier kommt die Warteschlange ins Spiel. Alle Ihre Eingaben (unter Umständen auch Zusatzinformationen) werden in eine Warteschlange eingereiht. Ihr Antrag steht in der Warteschlange und wird, sobald er an der Reihe ist, bearbeitet. Falls Sie der Erste in der Reihe für diese Domäne gewesen sind, werden Sie die Domäne zugeteilt bekommen.

- *Master Data Services*: Darunter versteht man, wenn Organisationen ihre Stammdaten unternehmensweit zentralisiert vereinheitlichen und für gezielte Analysen bereitstellen.
- *Data Quality Services*: Dies ist ein in dieser Version neues Tool, mit dessen Hilfe die Datenqualität in bestehenden Systemen verbessert werden kann. Lücken in Datenbeständen können damit besser aufgefunden und bereinigt werden. Dies können Fragestellungen sein wie: „Sind alle notwendigen Relationen vorhanden und gesetzt?“

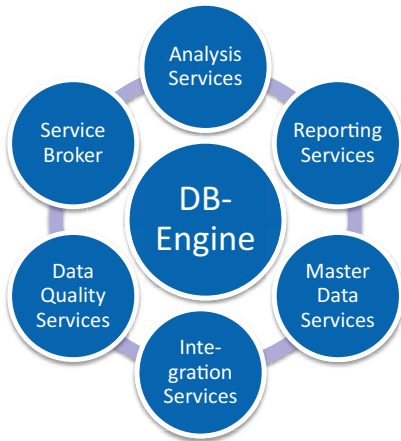


Bild 1.8 SQL Server und seine Zusatzkomponenten

Natürlich stehen diese Features nicht in jeder der verschiedenen SQL Server 2016-Editionen zur Verfügung:

- *Integration Services* stehen ab der Standard Edition zur Verfügung, manche spezielle Formen der Datentransformation erst mit der Enterprise Edition. In allen Editionen ist der SQL Server Import und Export Wizard enthalten.
- *Reporting Services* sind bis zu einem bestimmten Grad bereits ab der Express Edition integriert. Volle Integration findet erst ab der Standard Edition statt.
- *Analysis Services* sind teilweise ab der Standard Edition verfügbar, eine volle Integration ist erst mit der Enterprise Edition gegeben.

■ 1.2 Versionen und Editionen des SQL Servers

Dem Buch liegt die aktuelle Version SQL Server 2017 zugrunde. Diese Version weist gemeinsam mit dem SQL Server 2016 gegenüber deren Vorgängersystem SQL Server 2014 wesentliche Neuerungen auf. Dies betrifft nicht nur die eigentliche relationale Datenbank-Engine, die den Kern des Produkts ausmacht, sondern umfangreiche Erweiterungen der Rahmenprodukte. Diese unter dem Begriff BI (Business Intelligence) zusammengefassten Produkte enthalten beispielsweise die Analysis Services, Integration Services oder Reporting Services.

Erneuerungen gibt es in fast allen Bereichen des SQL Servers. Drei große Schwerpunkte, die ich in meiner täglichen Arbeit nutze, sind:

- *Speicheroptimierte Tabellen*: Ein wahrer Performanceboost sind die mit dem SQL Server 2014 eingeführten speicheroptimierten Tabellen. Hinter diesem Begriff verbirgt sich die Möglichkeit, ganze Tabellen vollständig im RAM zu halten. Da der Zugriff auf Festplatten in der Regel das ist, was eine Datenbank am meisten bremst, bietet diese Möglichkeit

ungeahnte Performancesteigerungen gegenüber herkömmlichen Tabellen. Hat es in der ersten Implementierung noch einige Einschränkungen in der Verwendung gegenüber herkömmlichen Tabellen gegeben – was in der ersten Implementierung eines komplexen und bahnbrechenden Features nichts Ungewöhnliches ist –, sind die meisten dieser Limitationen nun ausgeräumt. Damit ist dies nun ein extrem gutes und praxistaugliches Feature. Und ein weiteres Plus aus meiner Sicht: Es ist nun nicht mehr auf die Enterprise Edition beschränkt.

- *Always Encrypted*: Sensible Daten werden durch Zugriffsrechte auf dem Server vor unbefugtem Zugriff geschützt. Schon bisher hat es die Möglichkeit gegeben, Zugriffsrechte dermaßen einzuschränken, dass Daten vor unbefugten Zugriffen bestmöglich geschützt worden sind. Daten im Backup zu verschlüsseln, ergänzte diese Möglichkeiten, da die Zugriffsrechte, sobald die Datenbank auf einem eigenen Server eingespielt werden kann, selbst definiert werden können. Mit den neuen Möglichkeiten sind sie auch auf dem Server in der Datenbank verschlüsselbar, sodass selbst jemand, der sich der gesamten Datenbank bemächtigt, diese Daten nicht mehr einsehen kann.
- *Temporale Tabellen*: Änderungsprotokollierung ist in der Praxis sehr oft ein Thema. Trotz mehrerer Ansätze in der Vergangenheit, die in der Praxis kein voll befriedigendes Ergebnis geliefert haben, dies einfach vom System her zur Verfügung zu stellen, werden saubere und zufriedenstellende Lösungen mit Triggern ausprogrammiert. Temporale Tabellen liefern aus meiner Sicht erstmals eine sehr brauchbare Lösung, ohne eigene Programmierung eine Lösung für diese Aufgabenstellung zu erhalten. Mit dem SQL Server 2016 eingeführt, unterstützen sie mit dem SQL Server 2017 auch Änderungs- und Löschweitergaben bei Beziehungen (FOREIGN KEY mit DELETE CASCADE und UPDATE CASCADE).
- *Strict Security für Common Language Runtime (CLR)*: Neue Erweiterungen für die Sicherheit von CLR-Code erhöhen mit dem SQL Server 2017 zwar den Aufwand, um CLR-Code auf dem SQL Server ausführen zu können, bieten aber deutlich höhere Sicherheit. Unter CLR-Code versteht man mit C# oder VB.NET programmierte Prozeduren, die mittels der Common Language Runtime (CLR) direkt auf dem SQL Server ausgeführt werden können.

Ein paar weitere Erweiterungen, die teilweise auch in diesem Buch behandelte Themen betreffen, habe ich exemplarisch in der nachfolgenden Tabelle angeführt.

Tabelle 1.1 Einige Neuerungen in SQL Server 2016 und 2017

Thema	Beschreibung
Windows Azure	Daten können, anstelle lokal auf dem Server gehalten zu werden, in Windows Azure-BLOBs abgelegt werden. Das Hosten von SQL Server-Datenbanken auf einem virtuellen Computer in Windows Azure wird über eigene Bereitstellungsassistenten unterstützt. Die Sicherung einer SQL Server-Datenbank kann über eine URL in Windows Azure-BLOBs erfolgen.
Erstellen von Ausführungsplänen	Durch die überarbeitete Logik der Kardinalitätsschätzung werden die Qualität und damit die Effizienz von Ausführungsplänen verbessert. Das wiederum steigert die Abfrageleistung. (Ausführungspläne legen fest, wie der SQL Server intern eine von uns getätigte Abfrage abarbeitet.)
Live-Abfragestatistik	Abfragestatistiken zur Laufzeit bieten Administratoren einen tieferen Einblick in die Verarbeitung von Anweisungen.
Transact-SQL-Erweiterungen	Es gibt zahlreiche Erweiterungen in Form von neuen Funktionen und Anweisungen in der Sprache zur Bearbeitung von Daten. Beispielsweise sind dies die Funktionen DATEDIFF_BIG(), STRING_SPLIT() und STRING_ESCAPE() mit dem SQL Server 2016 und die Funktionen STRING_AGG(), TRANSLATE(), CONCAT_WS() und TRIM() mit dem SQL Server 2017.
Sicherheits-erweiterungen	Berechtigungen können nun auf Datensatzebene vergeben werden. Für diese Funktionalität ist bisher ein Workaround über gefilterte Sichten notwendig gewesen.
Temporäre Datenbank <i>tempdb</i>	Das Aufteilen der temporären Datenbank <i>tempdb</i> auf mehrere Dateien, die auch auf unterschiedlichen Datenträgern verwaltet werden können, ermöglicht eine bessere interne Nutzung und damit eine bessere Performance.
Native JSON-Unterstützung	Für Entwickler interessant ist die native Unterstützung von JSON als Ausgabeformat mit der Klausel FOR JSON, die mit der bisherigen Klausel FOR XML vergleichbar ist.
UTF-8 Unterstützung	Ich bin geneigt zu sagen „endlich“! Die Anweisung BULK INSERT, mit der Textdateien (CSV) einfach und schnell importiert werden können, unterstützt ab dem SQL Server 2016 nun auch UTF-8. Das ist mir in der Vergangenheit sehr oft abgegangen.
GRAPH-Unterstützung	Die Unterstützung von Graph-Datenbanken des SQL Server 2017 ist ein weiterer Schritt zur Integration von NoSQL-Elementen.

Editionen des SQL Server 2017

Microsoft liefert den SQL Server 2017 in einer Reihe unterschiedlicher Editionen aus. Ziel dieser Produktdifferenzierung ist es, dem Kunden ein Angebot zu unterbreiten, das es ermöglicht, den jeweiligen Anforderungen in Hinblick auf Leistungsfähigkeit, Laufzeit und Preise gerecht zu werden. Darüber hinaus werden zahlreiche Zusatzkomponenten angeboten. Welche dieser Komponenten im Einzelfall für eine Installation ausgewählt werden, hängt von den konkreten Anforderungen ab. Die früher verfügbare Business Intelligence Edition ist seit dem SQL Server 2016 nicht mehr verfügbar.

Einen schnellen Überblick über die Varianten gibt die folgende Tabelle:

Tabelle 1.2 SQL Server 2017-Editionen

Edition	Bedeutung/Anwendung
Enterprise Edition	Dies ist die von Featureseite umfangreichste SQL Server-Edition, die optimal für große Unternehmen und hochkomplexe Anforderungen geeignet ist und mit der hohe Anforderungen an das Datenbankmanagement erfüllt werden können. Dazu zählt die Unterstützung sehr großer OLTP-Systeme, hochkomplexer Datenanalysen, von Data-Warehousing-Systemen und von Webanwendungen für Unternehmen.
Standard Edition	Dieses Angebot richtet sich an kleine und mittelständische Unternehmen, welche die erweiterten Leistungsoptionen der SQL Server 2017 Enterprise Edition nicht benötigen. Im Lieferumfang der Edition sind die wesentlichen Funktionen von Business Intelligence enthalten.
Web Edition	Die Web Edition unterscheidet sich von der Standard Edition in erster Linie durch die Lizenzierung, die ausschließlich auf CPU-Basis erfolgt. Sie ist auf den Einsatz mit Webanwendungen ausgelegt.
Express Edition	Die Express Edition ist eine frei erhältliche Datenbank, die einfach zu verwenden und zu verwalten ist. Sie wird oftmals gemeinsam mit dem Microsoft Visual Studio verwendet und ist für die Entwicklung von datengesteuerten Anwendungen geeignet. Die Datenbank kann sowohl als eine Clientdatenbank als auch als einfache Serverdatenbank eingesetzt werden. SQL Server Express wendet sich an kleinere Softwarehersteller und Serverbenutzer sowie Entwickler, die nicht hauptberuflich Webanwendungen entwickeln. Die Express Edition gibt es auch als Variante mit <i>Advanced Services</i> . Die früher verfügbare <i>Express Edition mit Tools</i> , die bereits zusätzlich mit dem Management Studio eine grafische Entwicklungs- und Verwaltungsumgebung hatte, gibt es in dieser Form nicht mehr. Dies ist darin begründet, dass die Clienttools generell separat und frei erhältlich sind. Die Express Edition mit <i>Advanced Services</i> erlaubt zusätzlich noch, eine Volltextsuche einzusetzen, und enthält ein Frontend, um Berichte für die SQL Server Reporting Services zu erstellen.
LocalDB	Die LocalDB wird eingesetzt, wenn SQL Server in eine andere Anwendung eingebettet werden soll. Sie kommt mit minimalen Ressourcen aus und läuft nicht als Service, sondern ist in den Prozess eingebettet. Daher ist sie für den Einsatz von mobilen Anwendungen geeignet, bei denen kein Multiuser-Zugriff auf die Datenbank benötigt wird, denn dieser ist nicht möglich.
Developer Edition	Die Developer Edition ist eine freie Edition für Entwickler. Früher war sie für unter 100 € erhältlich, ab der Version 2016 ist sie überhaupt frei verfügbar. Diese Edition darf keine Produktivsysteme hosten; sie entspricht aber ihrem Umfang nach der Enterprise Edition.



HINWEIS: Der SQL Server 2017 bietet bis hinunter zur Express Edition nur mehr 64-Bit-Versionen an, 32-Bit werden ab der Version 2016 nicht mehr unterstützt. Dies ist dem Umstand geschuldet, dass auch im Desktopbereich, in dem auch die Express Edition oft eingesetzt wird, inzwischen Großteils nur mehr 64-Bit-Systeme im Einsatz sind.

Dass es keine 32-Bit-Versionen für den Server mehr gibt, dürfte in der Praxis kein Problem darstellen. Ich selber kann mich schon kaum mehr daran erinnern, wann ich zuletzt eine solche eingesetzt habe.



ACHTUNG! Problematischer ist es allerdings, dass mit der Version 17 das Management Studio gemeinsam mit den anderen Client-Anwendungen nun auch nur mehr als 64-Bit-Version verfügbar ist. Damit ist das SQL Server Management Studio in der Version 16.5.3 die letzte Version, die noch für 32-Bit-Systeme verfügbar ist. Ich selber habe zum Beispiel bei Kunden manchmal einen virtuellen Client für Remotearbeiten im Einsatz, der, seinerzeit als 32 Bit installiert, auch auf Windows 10 upgedatet nicht ohne erheblichen Aufwand auf 64 Bit gewechselt werden kann.

Die **Enterprise Edition** ist das Flaggschiff unter den Versionen. Sie bietet eine skalierbare Datenbankserverumgebung für Unternehmen jeglicher Größe. Die Enterprise Edition besitzt keine Beschränkungen im Hinblick auf CPU, RAM oder Datenbankgröße. Sie ermöglicht Multimode-Clustering, Online-Indizierung, Oracle-Replikation und viele weitere Funktionen. Die SQL Server Enterprise Edition passt sich den erforderlichen Leistungsebenen problemlos an, sodass die größten OLTP-Systeme und Websites sowie umfassende Data-Warehouse-Systeme unterstützt werden. Sie bietet dazu die notwendigen Optionen für Skalierbarkeit und Zuverlässigkeit mit verteilten und partitionierten Sichten, Protokollversand und Failover-Clusterunterstützung, die für unternehmensweite, branchenspezifische und Internetszenarien erforderlich sind. Alle diese Hochverfügbarkeitstechnologien sind unter dem Begriff *AlwaysOn* zusammengefasst. SQL Server Enterprise Edition enthält außerdem erweiterte Analysetools bzw. OLAP-Features für den Umgang mit sehr großen Cubes mit vielen Dimensionen für Data-Warehouse-Anwendungen.

Die **Standard Edition** enthält bereits Funktionen, die für eine Unternehmensdatenbank erforderlich sind, wie zum Beispiel 2-Node-Clustering, unbegrenzte RAM-Unterstützung, bis zu 24 Prozessorkerne, unbegrenzte Datenbankgröße sowie eine Auswahl an Business-Intelligence-Funktionen. Die SQL Server Standard Edition stellt eine erschwingliche Option für kleine und mittelständische Unternehmen dar, die keine erweiterten Features für Skalierbarkeit und Zuverlässigkeit oder Analysefeatures von SQL Server Enterprise Edition benötigen.

Die **Web Edition** ist speziell für den Einsatz auf Webanwendungen ausgerichtet. Sie verzichtet zwar auf die meisten Business-Intelligence-Funktionalitäten, unterstützt dafür aber bis zu 16 Prozessorkerne und weist keine Begrenzung des nutzbaren Arbeitsspeichers auf.

Mit der SQL Server **Developer Edition** lassen sich komfortable Anwendungen auf Basis des SQL Servers erstellen. Sie ist quasi eine besondere Option für Entwickler von Datenbanken, die SQL Server als Datenspeicher der zu entwickelnden Anwendungen verwenden. Die Edition umfasst alle Funktionen der Enterprise Edition, beinhaltet jedoch einen besonderen Endbenutzer-Lizenzvertrag (EULA) für Entwicklungen und Tests. Obwohl die Developer Edition alle Funktionen der Enterprise Edition unterstützt, um es Entwicklern zu ermöglichen, Anwendungen zu schreiben und zu testen, die diese Funktionen verwenden können, wird die Developer Edition nur für den Einsatz als Entwicklungs- und Testsystem, jedoch nicht für den Einsatz als Produktionsserver lizenziert.



Kostenlos aus dem Web kann die SQL Server 2017 Evaluation Edition heruntergeladen werden. Diese Edition ist eine mit allen Funktionen ausgestattete Version, die ausschließlich zur Evaluierung der Funktionen des SQL Servers dient und maximal 180 Tage ab dem Installationsdatum ausgeführt werden kann.

Die drei Editionen Evaluation, Express und Developer können Sie von folgender Adresse herunterladen:

<https://www.microsoft.com/de-de/sql-server/sql-server-downloads>

Sie erhalten hier einen Installer, mit dem Sie entweder direkt installieren oder aber auch wahlweise die aktuellen Installationsmedien vollständig herunterladen können.

Weit verbreitet ist die frei verfügbare **Express Edition**. Diese weist zwar einige Einschränkungen auf, ist aber für kleinere Anwendungen und als Einstiegssystem bestens geeignet.

Schließlich sei noch auf **Microsoft Azure SQL-Datenbank** hingewiesen. Diese in der Cloud von Microsoft gehostete Version bietet die Möglichkeit, Datenanwendungen auszulagern und ohne einen eigenen Server auszukommen. Vor allem auch für kleinere Anwendungen wird dies immer interessanter, da man sich um Dinge wie die Verfügbarkeit nicht mehr kümmern muss. Arbeiten können Sie mit einer derartigen Version beinahe so, als stünde sie bei Ihnen im Haus. Weitere Informationen zu Microsoft Azure SQL-Datenbank finden Sie unter <https://azure.microsoft.com/de-de/services/sql-database>.

■ 1.3 SQL Server 2017 installieren

In diesem Abschnitt gehe ich mit Ihnen Schritt für Schritt die Installation eines SQL Servers unter Windows durch, damit Sie sich ohne Probleme eine funktionierende Arbeitsumgebung einrichten können.



HINWEIS: Die Installation eines SQL Server 2017 unter Linux finden Sie in Kapitel 12 beschrieben.

Voraussetzungen für die Installation

Für eine fehlerfreie Installation des Microsoft SQL Server 2017 muss Ihr Computer bestimmte Systemanforderungen bezüglich Hardware und Betriebssystem erfüllen.

- **Speicherplatz:** Der benötigte Festplattenspeicher hängt von den zu installierenden Komponenten ab. Das Datenbankmodul schlägt dabei mit Replikation, Volltextsuche und Data Quality Services mit ca. 1,5 GB zu Buche. Weitere 306 MB benötigen jeweils die Integration Services und 967 MB sind für die Reporting Services erforderlich. Die Clientkomponenten beanspruchen mit ca. 500 MB viel Kapazität. Beachten Sie, dass beim SQL Server 2017 das Management Studio nicht mehr berücksichtigt ist, da dieses separat herunterladen und zu installieren ist. Weitere 200 MB werden von der Online-Dokumentation benötigt. Während des Setups sind auf dem Systemlaufwerk mindestens 6 GB freier Platz nötig.
- **Arbeitsspeicher:** Hier unterscheiden sich die Werte der Express Edition von denen der übrigen Editionen. Die Express Edition (Datenbankmodul alleine!) benötigt minimal 512 MB Arbeitsspeicher, alle anderen Konfigurationen mindestens 1 GB. Empfohlen werden für die Express Edition wenigstens 1 GB, für alle anderen Editionen mindestens 4 GB. Dies sind allerdings Untergrenzen und mehr Arbeitsspeicher macht auf jeden Fall Sinn, vor allem bei steigender Datenbankgröße.
- **Prozessor:** Der SQL Server 2017 unterstützt nur mehr die 64-Bit-Architektur. Sie benötigen mindestens einen 64-Bit-Prozessor mit 1,4 GHz, empfohlen werden 2 GHz oder mehr.
- **Betriebssystem:** Je nach Edition werden unterschiedliche Anforderungen an das Betriebssystem gestellt. Einen groben Überblick gibt die folgende Tabelle. Einen detaillierteren Überblick mit den genauen Angaben über die jeweiligen Editionen der angegebenen Betriebssysteme erhalten Sie unter <https://docs.microsoft.com/de-de/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server>.

Tabelle 1.3 Betriebssystemanforderungen für die Installation

Edition	Betriebssystem
Enterprise, Web	Windows Server 2012, Windows Server 2012 R2, jeweils Datacenter, Standard, Essentials und Foundation Windows Server 2016 Datacenter und Standard (nicht: Essentials)
Standard, Developer, Express	Wie Enterprise, zusätzlich: Windows 8, Pro und Enterprise Windows 8.1, Pro und Enterprise Windows 10 Home, Professional und Enterprise, IoT Enterprise

Für die Installation von SQL Server 2017 ist außerdem das .NET Framework 4.6 erforderlich. Sowohl für den SQL Server 2016 als auch für den neuen SQL Server 2017 werden neben dem Windows Server 2008 und 2008 R2 auch Windows 7 nicht mehr als Betriebssystem unterstützt.



HINWEIS: Die nachfolgenden Screenshots habe ich beim Setup einer Enterprise Edition auf einem Windows Server 2016 angefertigt.

Wie müssen Sie vorgehen, um den SQL Server 2017 auf Ihrem Computer zu installieren?



ACHTUNG! Wenn Sie eine deutschsprachige Version des SQL Servers installieren, achten Sie darauf, dass in der Systemsteuerung die Spracheneinstellung auf *Deutsch (Deutschland)* eingestellt ist. Sie werden sich nun fragen, warum ich das anmerke. Für mich als Österreicher ist dies nicht selbstverständlich. Denn mit der Einstellung *Deutsch (Österreich)* erhalten Sie eine Fehlermeldung. Diese besagt, dass Sie nicht die passenden Installationsmedien für die gewählte Sprache verwenden. Es ist ausreichend, die Einstellung *Deutsch (Deutschland)* während des Starts des Setups aktiv zu haben, danach können Sie gegebenenfalls sofort wieder auf die vorherige Einstellung zurückkehren.

Legen Sie die Microsoft SQL Server 2017-DVD in das Laufwerk ein. In der Regel werden Sie eine ISO-Datei verwenden, die Sie auf Ihrem Rechner mit der Option **BEREITSTELLEN** als Laufwerk mounten. Haben Sie eine Express-Edition heruntergeladen, liegt diese zumeist als EXE-Datei vor, die nach dem Aufruf in ein anzugebendes Verzeichnis extrahiert wird. Starten Sie die Datei *Setup.exe* für die Installation.

Der angezeigte Startbildschirm liefert eine Vielzahl an Möglichkeiten und kann auch nach einer Installation jederzeit über das Startmenü als *SQL Server-Installationscenter* geöffnet werden.

Vom SQL Server-Installationscenter aus können sämtliche Aufgaben im Zusammenhang mit einer Installation erledigt und umfassende Informationen dazu eingeholt werden. Auf der linken Seite des Dialogs finden Sie die folgenden Optionen vor:

- *Planen*
- *Installation*
- *Wartung*
- *Extras*
- *Ressourcen*
- *Erweitert*
- *Optionen*

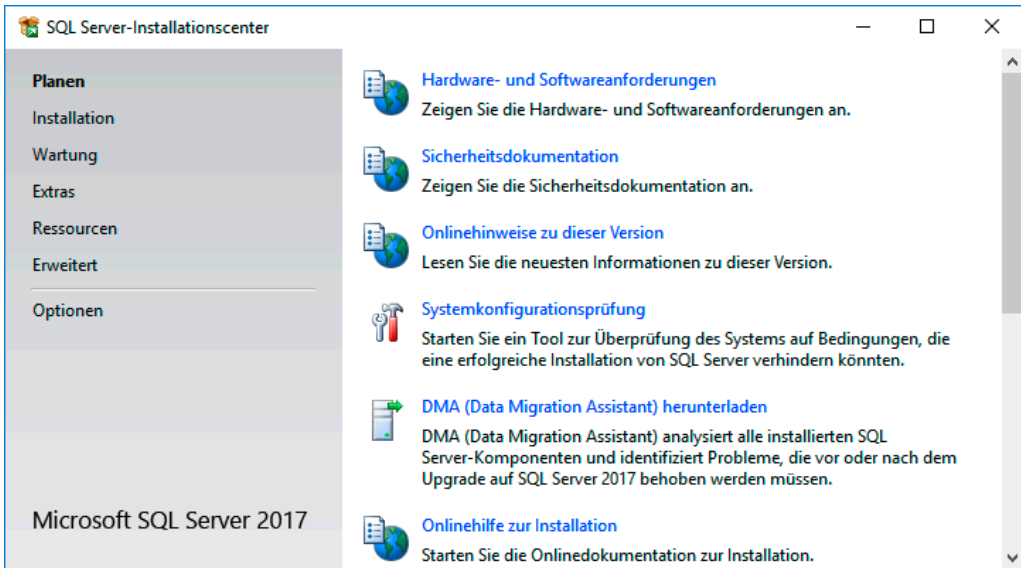
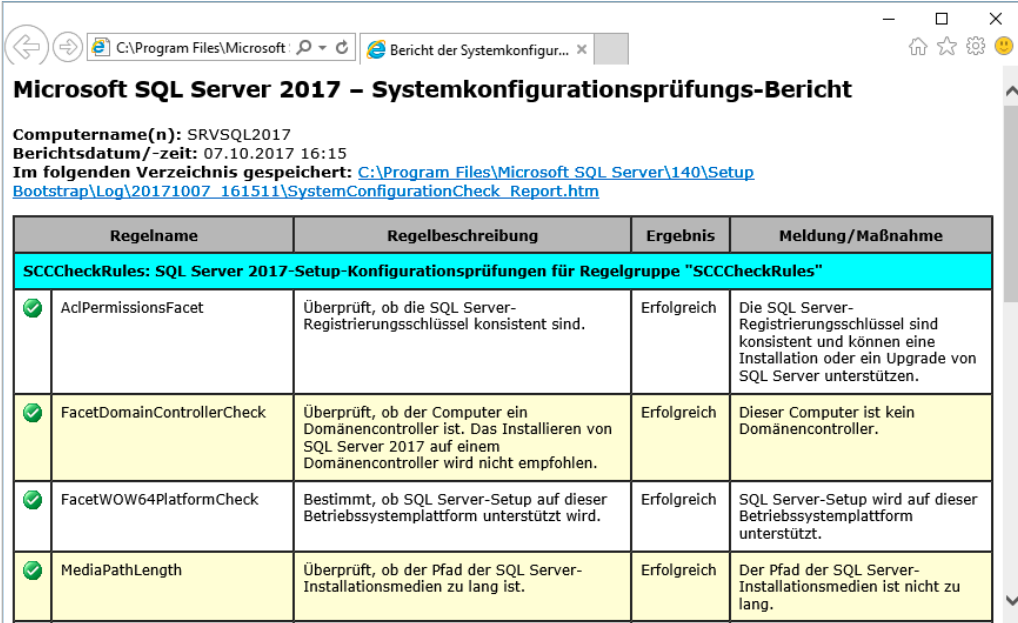


Bild 1.9 Startbildschirm von *SQL Server-Installationscenter*

Unter diesen Optionen sind jeweils mehrere Programmpunkte zusammengefasst. Manche Programmpunkte sind auch unter mehreren Optionen zu finden, da sie inhaltlich zu mehreren Optionen passen.

Unter *Planen* haben Sie direkten Zugriff auf Informationsquellen wie zum Beispiel auf *Hard- und Softwareanforderungen* oder auch die Onlinehilfe für die Installation. Mit der *Systemkonfigurationsprüfung* können Sie schon im Vorfeld einer Installation prüfen, ob die Hard- und Softwarevoraussetzungen erfüllt sind. Das Ergebnis können Sie sich danach in übersichtlicher Form anzeigen lassen.



Microsoft SQL Server 2017 – Systemkonfigurationsprüfungs-Bericht

Computername(n): SRVSQL2017
 Berichtsdatum/-zeit: 07.10.2017 16:15
 Im folgenden Verzeichnis gespeichert: C:\Program Files\Microsoft SQL Server\140\Setup Bootstrap\Log\20171007_161511\SystemConfigurationCheck_Report.htm

Regelname	Regelbeschreibung	Ergebnis	Meldung/Maßnahme
SCCCheckRules: SQL Server 2017-Setup-Konfigurationsprüfungen für Regelgruppe "SCCCheckRules"			
✓ AclPermissionsFacet	Überprüft, ob die SQL Server-Registrierungsschlüssel konsistent sind.	Erfolgreich	Die SQL Server-Registrierungsschlüssel sind konsistent und können eine Installation oder ein Upgrade von SQL Server unterstützen.
✓ FacetDomainControllerCheck	Überprüft, ob der Computer ein Domänencontroller ist. Das Installieren von SQL Server 2017 auf einem Domänencontroller wird nicht empfohlen.	Erfolgreich	Dieser Computer ist kein Domänencontroller.
✓ FacetWOW64PlatformCheck	Bestimmt, ob SQL Server-Setup auf dieser Betriebssystemplattform unterstützt wird.	Erfolgreich	SQL Server-Setup wird auf dieser Betriebssystemplattform unterstützt.
✓ MediaPathLength	Überprüft, ob der Pfad der SQL Server-Installationsmedien zu lang ist.	Erfolgreich	Der Pfad der SQL Server-Installationsmedien ist nicht zu lang.

Bild 1.10 Bericht der Systemkonfigurationsprüfung



PRAXISTIPP: Planen Sie ein Upgrade, sollten Sie unbedingt wie hier angeboten den *DMA (Data Migration Assistant)*, früher *Upgrade Advisor*, herunterladen und installieren. Dieser prüft sämtliche Voraussetzungen für ein erfolgreiches Upgrade und teilt Ihnen mit, was Sie erledigen müssen, bevor Sie mit dem Upgrade beginnen können. Dies ist ein sehr praktisches Tool, mit dem ich selber schon erfolgreich in kürzester Zeit eine Migration durchgeführt habe.

Sollten Sie planen, eine Microsoft Access-Datenbank auf den neuen SQL Server zu migrieren, ist die Option *SSMA (SQL Server Migration Assistant)* *herunterladen* hilfreich. Der SQL Server Migration Assistent ist ein sehr praktisches grafisches Werkzeug, das für verschiedene Ausgangsdatenbanksysteme angeboten wird und die Migration vorhandener Datenbanken sehr gut unterstützt. Vor allem kann dieses Tool die vorgenommenen Migrationsdefinitionen als Projekt speichern und ermöglicht es dadurch, den Vorgang so lange zu verfeinern und immer wieder durchzuführen, bis alles optimal vonstattengeht. Verfügbar ist der SSMA für folgende Datenbanksysteme:

- Microsoft Access
<https://www.microsoft.com/en-us/download/details.aspx?id=54255>
- Oracle
<https://www.microsoft.com/en-us/download/details.aspx?id=54258>
- MySQL
<https://www.microsoft.com/en-us/download/details.aspx?id=54257>

- SAP Adaptive Server Enterprise (früher Sybase ASE)
<https://www.microsoft.com/en-us/download/details.aspx?id=54256>
- DB2
<https://www.microsoft.com/en-us/download/details.aspx?id=54254>

Da der Link für SSMA im Installationscenter im Moment ins Leere geht, habe ich die tatsächlichen Links für die im Moment aktuelle Version 7.6 jeweils ergänzt.



HINWEIS: Durch die Umbenennung des früheren *Upgrade Advisors* in den *Data Migration Assistant* (DMA) kann es an dieser Stelle leicht zu einer Verwechslung mit dem *SQL Server Migration Assistant* (SSMA) kommen. Allerdings haben die beiden zuvor beschriebenen nichts miteinander zu tun, achten Sie daher auf die jeweils verwendete Begrifflichkeit.

Die eigentliche Installation startet unter der Option *Installation*.

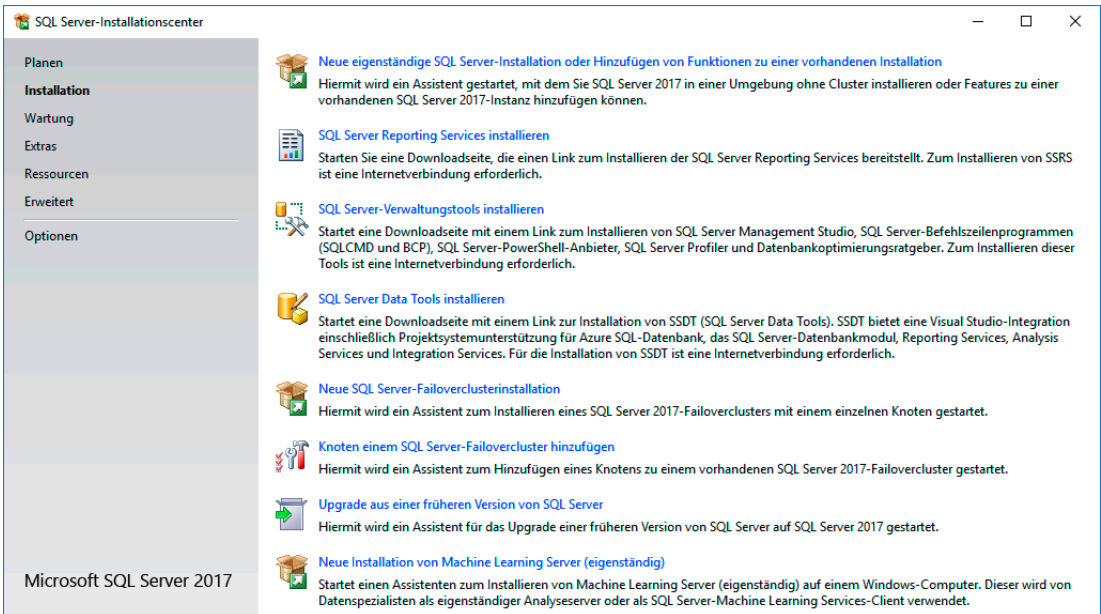


Bild 1.11 Optionen zur Installation

Die wichtigste Option ist die erste mit der Bezeichnung *Neue eigenständige SQL Server-Installation oder Hinzufügen von Funktionen zu einer vorhandenen Installation*. Diese Option werden Sie üblicherweise verwenden. Das Upgrade einer älteren Version – zuvor sollten Sie den Data Migration Assistant verwendet haben – ist ebenfalls hier über *Upgrade aus einer früheren Version von SQL Server* zu starten.



ACHTUNG! Da der SQL Server 2017 unter Windows Server 2008 und 2008 R2 nicht mehr installiert werden kann, kann ein Upgrade auf dieser Plattform nicht direkt erfolgen. Sie müssen stattdessen SQL Server 2017 auf einen neuen Server übertragen.



HINWEIS: Da die SQL Server-Verwaltungstools wie das SQL Server Management Studio nicht mehr in den Setup-Dateien enthalten sind, findet sich hier auch unter der Option *SQL Server-Verwaltungstools installieren* ein Link zum Herunterladen derselben.

Ebenso finden Sie hier einen Link zu den *SQL Server Data Tools*. Diese bieten in eine Visual Studio Shell verpackt die Möglichkeiten der .NET-Programmierung für den SQL Server sowie die Projektunterstützung für Reporting Services, Analysis Services und Integration Services. Ist eine Visual Studio-Installation bereits vorhanden, werden diese Tools in diese Installation integriert.

Unter der Option *Wartung* können Sie eine Editionsaktualisierung vornehmen. Hier können Sie beispielsweise eine Standard Edition im Nachhinein in eine Enterprise Edition umwandeln, ohne eine Neuinstallation durchführen zu müssen. Ebenso können Sie hier eine beschädigte Installation über die entsprechende Option reparieren.



Bild 1.12 Option *Wartung* im Installationscenter

Unter der Option *Extras* finden Sie die Systemkonfigurationsprüfung, die schon unter *Planen* vorhanden war. Außerdem können Sie sich hier einen Überblick über bereits installierte Features und Instanzen verschaffen. Nicht mehr zu finden ist an dieser Stelle die Option, die *SQL Server Data Tools - Business Intelligence*, die noch früher als *Business Intelli-*

gence *Development Studio* bekannt gewesen sind, direkt zu installieren. Diese dienen zum Beispiel dem Erstellen von Integration Services-Paketen. Sie sind nun kein eigenes Installationspaket mehr und in die zuvor schon erwähnten SQL Server Data Tools integriert. Der Zusatz *Business Intelligence* ist damit verschwunden.

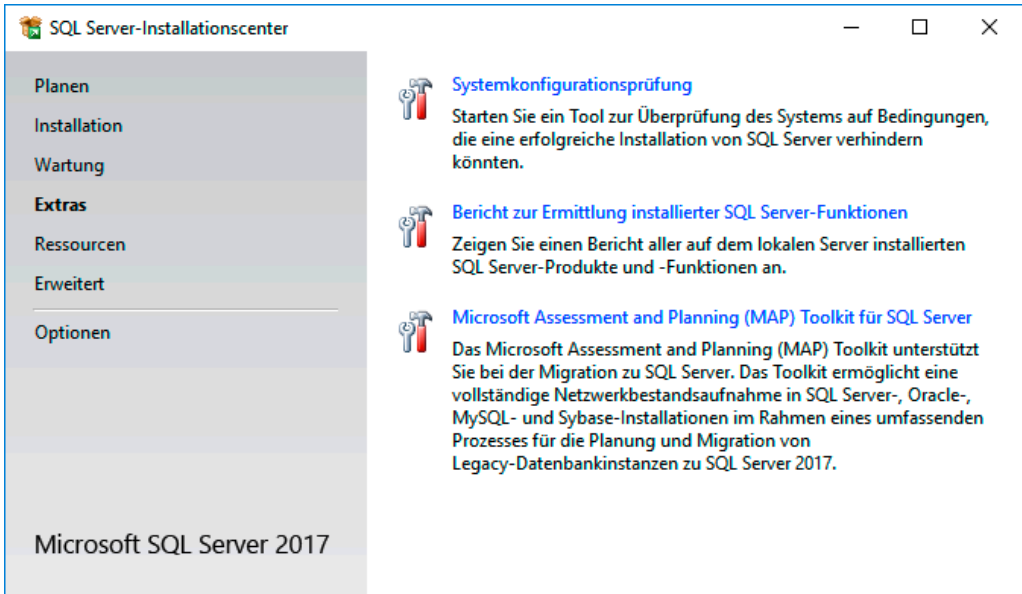


Bild 1.13 *Extras* des Installationscenters

Unter *Ressourcen* finden Sie Links zu unterschiedlichen Online-Informationsquellen. Unter *Erweitert* können Sie eine unbeaufsichtigte Installation mittels einer Konfigurationsdatei vornehmen, einen Failover-Cluster vorbereiten und dessen Konfiguration abschließen. Unter *Optionen* wählen Sie aus, ob Sie 32- oder 64-Bit-Komponenten installieren möchten. Da die Hauptkomponenten nur mehr als 64-Bit verfügbar sind, sehe ich diesen Dialogpunkt lediglich als vergessenen Rest aus den Vorversionen an.

Installation ausführen

Gestartet wird die Installation über den ersten Programmpunkt aus der Option *Installation* des SQL Server-Installationscenters. Grundsätzlich ähnelt der Vorgang sehr dem der Vorgängerversionen, lediglich die Reihenfolge mancher Schritte hat sich geändert. Der erste Schritt ist die Eingabe des *Product Key*.

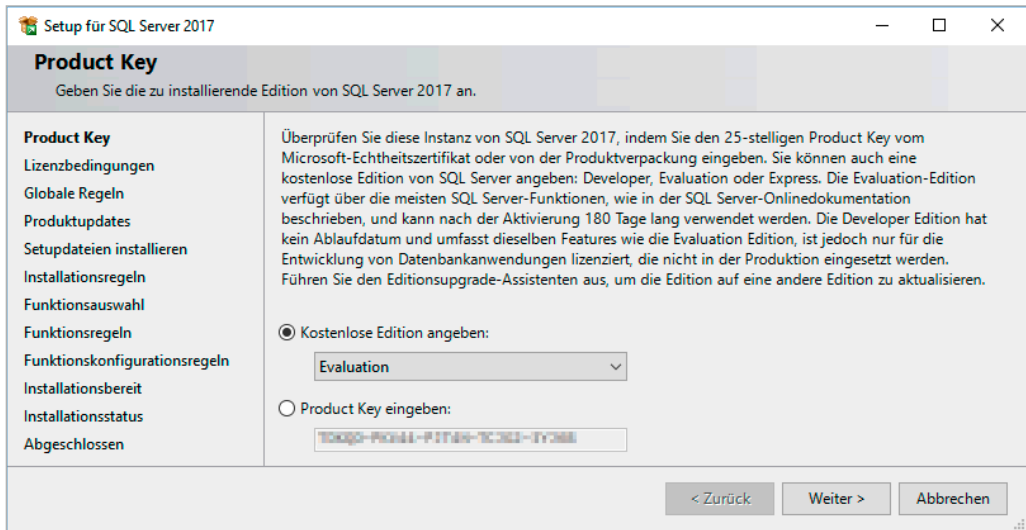


Bild 1.14 Product Key oder freie Edition

Sehr praktisch finde ich die Option, dass man beim Setup mit den Installationsmedien einer Vollversion auch eine der freien Editionen installieren kann. Damit müssen Sie zum Beispiel, wenn Sie die Installationsmedien einer Standard Edition besitzen, für die Installation einer zusätzlichen Express Edition keine separaten Dateien herunterladen. Auch können Sie eine Evaluation oder Developer mit manchen Installationsmedien auswählen. Umgekehrt ist es auch möglich, mit einer heruntergeladenen Evaluation Edition durch Eingabe eines Product Keys direkt auch eine lizenzierte Vollversion zu installieren. Eine Evaluation Edition können Sie 180 Tage lang nutzen und danach bei Bedarf mit einem Editionsupgrade direkt in eine lizenzierte umwandeln.

Nach dem Akzeptieren der Lizenzbestimmung werden die globalen Regeln überprüft. Sie entsprechen großteils der zuvor beschriebenen Systemkonfigurationsprüfung. Sie stellen den ersten Schritt von mehreren Prüfungen dar. Hier werden Voraussetzungen geprüft, die unabhängig von den später gewählten Setup-Optionen stets erfüllt sein müssen. Werden hier Mängel festgestellt, können Sie diese, ohne das Setup zu unterbrechen, beheben und die Prüfung erneut ausführen. Allerdings ist dies nur möglich, wenn das Beheben eines Fehlers keinen Neustart des Rechners erfordert.

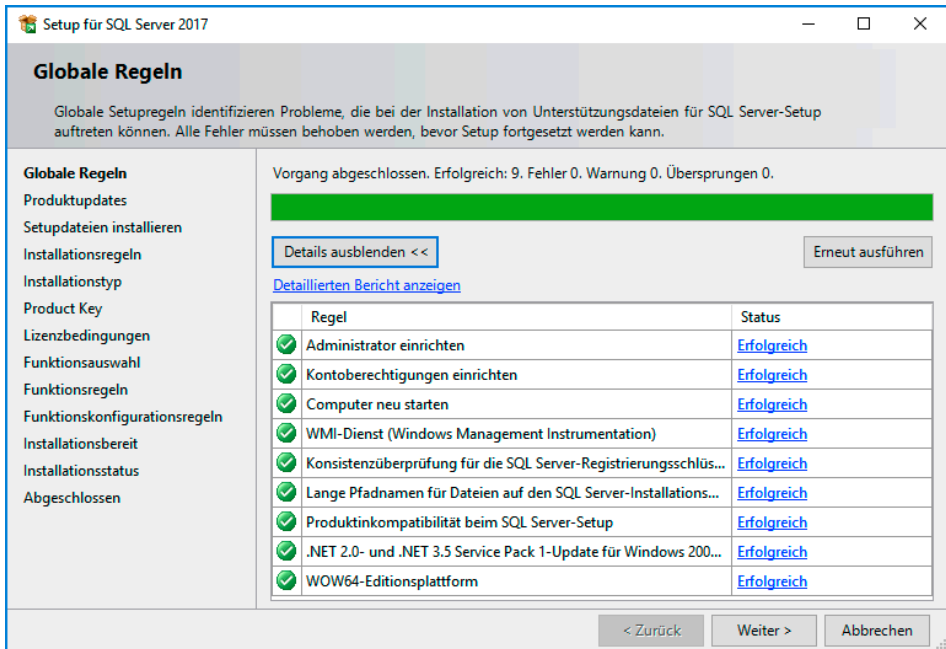


Bild 1.15 Prüfen der globalen Regeln

Sind alle Anforderungen erfüllt, wird das Ergebnis gar nicht mehr angezeigt, sondern direkt zum nachfolgenden Schritt gewechselt. Im linken Dialogbereich sehen wir ja während des gesamten Vorgangs den Status der Gesamtinstallation. So sehen wir, dass als Nächstes der Schritt *Produktupdates* ansteht. Um sich später das Nachinstallieren von Updates zu ersparen, bietet Ihnen das Setup die Möglichkeit, online nach Updates zu suchen und diese gegebenenfalls gleich mit zu installieren.

Nach der Suche und dem unter Umständen notwendigen Herunterladen von *Produktupdates* werden nun die *Setupdateien installiert*. Da diese in der Regel auf dem System verbleiben, muss dies nur bei der ersten auf einem Gerät durchgeführten Installation mit dieser Version erfolgen.

Direkt danach erfolgt die Prüfung weiterer *Installationsregeln*. Hier haben diese eine Warnung im Bereich der Windows-Firewall ergeben. Um die Details der Warnung anzuzeigen, müssen Sie auf die Warnung klicken. Im Dialog werden Ihnen dann weiterführende Informationen angeboten. In unserem Fall ist es die Warnung, dass die Windows-Firewall aktiv ist und bestimmte Programmooptionen des SQL Servers von dieser geblockt werden, sofern nicht bestimmte Ports geöffnet werden. Durch die Windows-Firewall müssen folgende Zugriffe erlaubt werden:

- TCP Port 1433 für die Datenbank-Engine. Bei einer weiteren Instanz muss ein anderer Port verwendet werden; dann ist dieser anzugeben.
- Das SQL Server Management Studio ist in die Liste derjenigen Programme, die durch die Firewall hindurch kommunizieren dürfen, aufzunehmen.

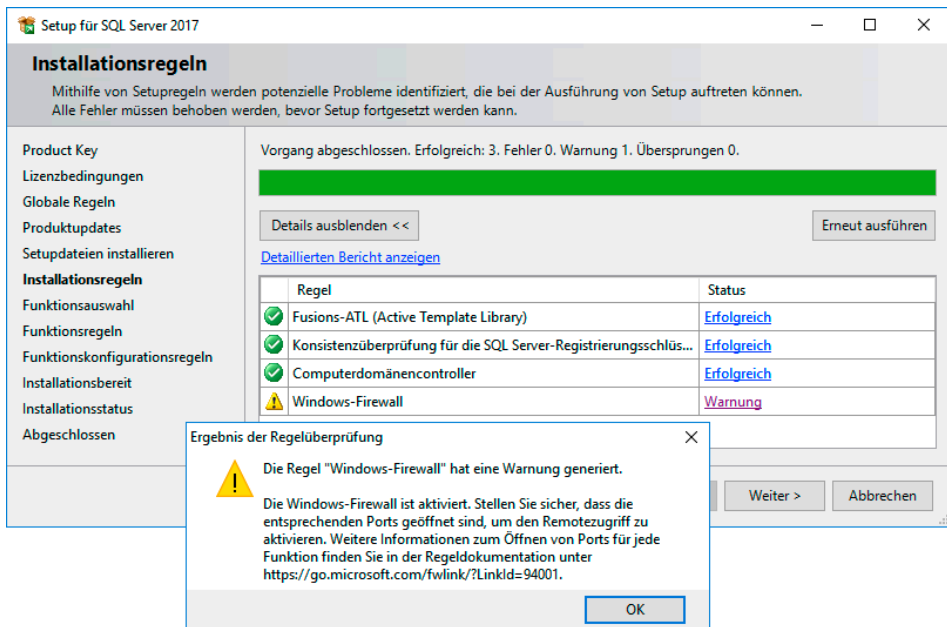


Bild 1.16 Prüfung von Installationsregeln

Eine *Warnung* führt nicht dazu, dass das Setup nicht fortgesetzt werden darf. Lediglich wenn eine der Prüfungen *Nicht erfolgreich* ergibt, kann das Setup vorerst nicht weiter fortgesetzt werden.

Nach der Prüfung der Installationsregeln gelangen Sie zur Funktionsauswahl. Unter diesem Punkt werden *Instanzfunktionen* sowie *Freigegebene Funktionen* gewählt.

- *Instanzfunktionen* gehören zu einer speziellen Instanz und werden für jede Instanz separat installiert. Zu ihnen gehören, je nach gewählter Edition:
 - *Datenbankmoduldienste (Database Engine Services)*: Bei der Auswahl dieser Option wird das Datenbankmodul installiert, wobei separat
 - SQL Server-Replikation,
 - Machine Learning-Dienste (datenbankintern): R, Python
 - Volltext- und semantische Extraktion für die Suche,
 - Data Quality Services und
 - PolyBase-Abfragedienst für externe Daten

gewählt werden können. Unter Replikation versteht man den Abgleich von mehreren Kopien einer Datenbank untereinander. Bei der Volltextsuche ist auch die semantische Suche dabei. Hier geht es darum, nach der Bedeutung zu suchen. Zum Beispiel suchen Sie nach „Stuhl“ und finden den Begriff „Sessel“. Die Data Quality Services bieten die Möglichkeit, die inhaltliche und strukturelle Qualität von Daten zu verbessern. Darunter können Sie sich zum Beispiel die Überprüfung und Korrektur von Adressen nach den Eingabegepflogenheiten des jeweiligen Landes vorstellen. R-Services und PolyBase

sind beim SQL Server 2016 neu hinzugekommen, mit dem SQL Server 2017 ist hier auch Python verfügbar. Hierbei geht es um Tools zur Big Data Integration.

- *Analysis Services*: Mit dieser Variante werden die Analysis Services installiert, die es ermöglichen, OLAP-Cubes zu bilden, mit denen eine Data-Warehouse-Lösung aufgebaut werden kann.
- *Reporting Services – Systemeigen*: Dieses Feature installiert einen Berichtsserver, der das Erstellen und Verteilen von Berichten in unterschiedlichsten Formaten ermöglicht. Bei den Reporting Services wird zwischen zwei Varianten unterschieden: zum einen in der Ausprägung *systemeigen*, hierbei handelt es sich um die eigenständige Variante. Unter den freigegebenen Funktionen finden Sie eine weitere Variante, die in Share-Point integriert ist. Diese benötigen Sie beispielsweise, um Power View zu nutzen.
- *Freigegebene Funktionen* müssen auf einem Rechner nur einmal installiert werden und werden von mehreren Instanzen gemeinsam genutzt.
 - *Machine Learning Server (eigenständig)*: Dieses Feature ist sowohl als Instanzfunktion als auch als freigegebene Funktion verfügbar.
 - R
 - Python
 - *Data Quality Client*: Installiert den Client für die Data Quality Services.
 - *Konnektivität der Clienttools*: Installiert den SQL Server Native Client mit Treibern – zum Beispiel für ODBC.
 - *Integration Services*: Dienste zur Datenintegration und zum Erstellen von Workflows. Neu im SQL Server 2017 sind die Zusatzoptionen des Scale Out-Masters und Scale Out Workers.
 - *Abwärtskompatibilität der Clienttools*: Damit können Clienttools auch für ältere SQL Server-Versionen eingesetzt werden.
 - *Clienttools SDK*: Stellt das Software Development Kit mit Ressourcen für Entwickler zur Verfügung.
 - *Distributed Replay Controller, Distributed Replay Client*: Mit diesen Werkzeugen lassen sich Auslastungs- und Änderungsszenarien simulieren.
 - *SQL Client Connectivity SDK*: Bietet das Software Development Kit für den SQL Server Native Client für Entwickler.
 - *Master Data Services*: Installiert die Master Data Services. Dies ist ein Werkzeug, mit dem Stammdaten aus den verschiedensten Quellen und Systemen zu einem gemeinsamen Stand zusammengeführt und konsolidiert werden können.



HINWEIS: Falls Ihnen an dieser Stelle die *Verwaltungstools* fehlen, die in den Vorversionen jeweils in zwei Varianten auswählbar gewesen sind, sei nochmals erwähnt, dass diese ab dem SQL Server 2016 separat heruntergeladen und installiert werden müssen.

<https://docs.microsoft.com/de-de/sql/ssms/download-sql-server-management-studio-ssms>

Dasselbe gilt ab dem SQL Server 2017 für die Reporting Services. Sind diese bisher noch in zwei Varianten verfügbar gewesen, sind sie nun auch als eigenes Tool ausgelagert. Sie können es für die separate Installation unter dieser Adresse herunterladen:

<https://www.microsoft.com/de-DE/download/details.aspx?id=55252>

Nett finde ich die Anzeige im oberen Dialogbereich mit dem Link hinter der Frage *Auf der Suche nach den Reporting Services?*.

Ebenso verschwunden sind in dieser Version die Dokumentationskomponenten. Dies trägt dem Umstand Rechnung, dass diese in erster Linie online genutzt werden.

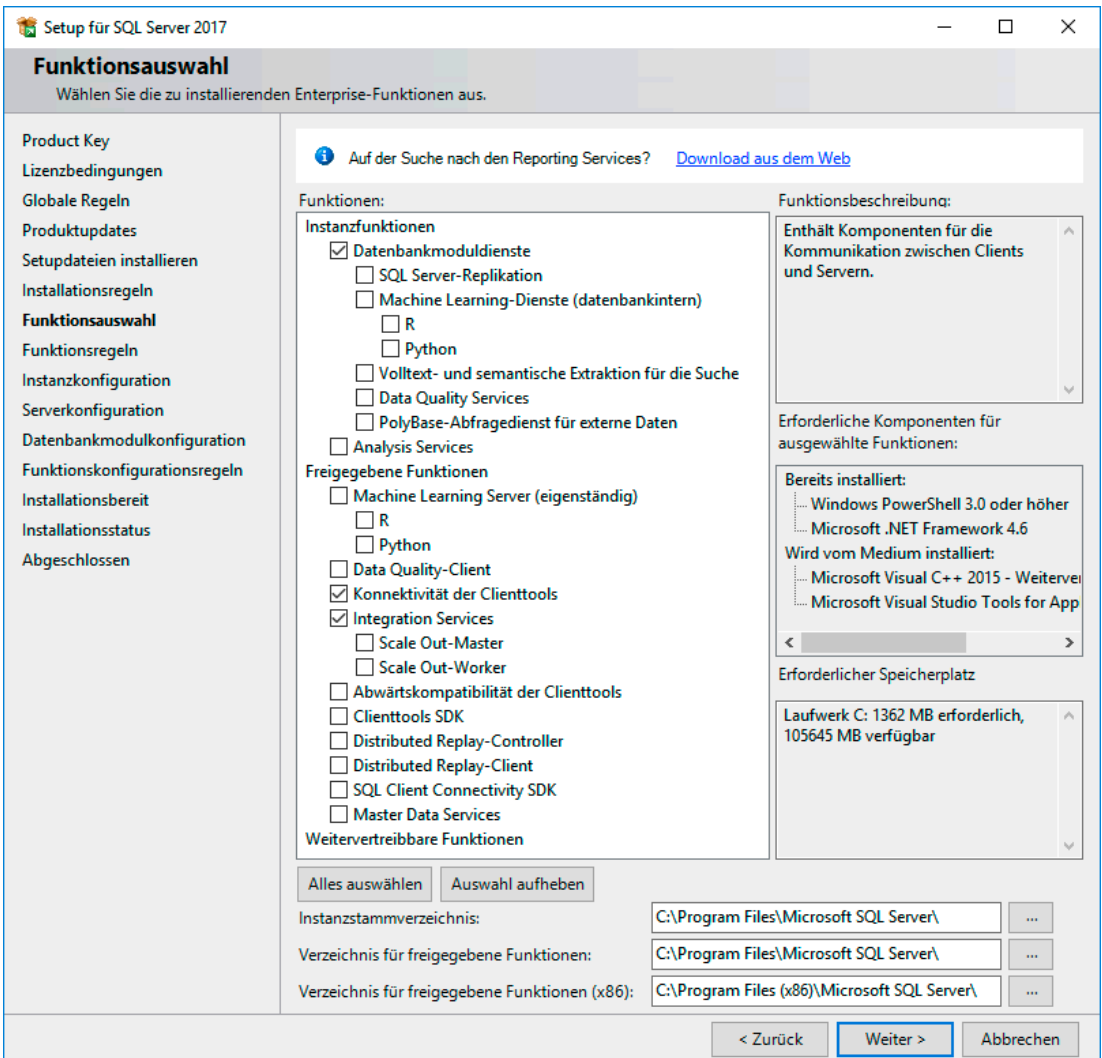


Bild 1.17 Auswahl der zu installierenden Komponenten

Für manche Features kann der Installationspfad an dieser Stelle angepasst werden, ebenso wird an dieser Stelle das *Instanzstammverzeichnis* abgefragt. Dieses gilt für die Programmdateien und muss aus meiner Sicht nicht unbedingt angepasst werden. Den Speicherort für Datendateien können Sie etwas später im Setup noch eigens festlegen.



HINWEIS: Das Verzeichnis für die freigegebenen Features kann nicht mehr geändert werden, wenn auf dem Rechner bereits frühere Installationen vorhanden sind.

Funktionen können auch später in einem weiteren Durchgang dazu installiert werden. Wählen Sie vorerst die in Bild 1.17 markierten Funktionen für die Erstinstallation aus.

Die Überprüfung der *Funktionsregeln* erfolgt im nächsten Schritt. Hier werden noch Voraussetzungen geprüft, die sich aufgrund der zuvor getroffenen Auswahl ergeben. Auch hier gilt, dass bei einem Fehler zuerst nachinstalliert werden muss, bevor das Setup über diesen Punkt hinaus fortgesetzt werden kann. Bei erfolgreicher Prüfung wird ebenfalls unmittelbar zum nächsten Schritt gewechselt.

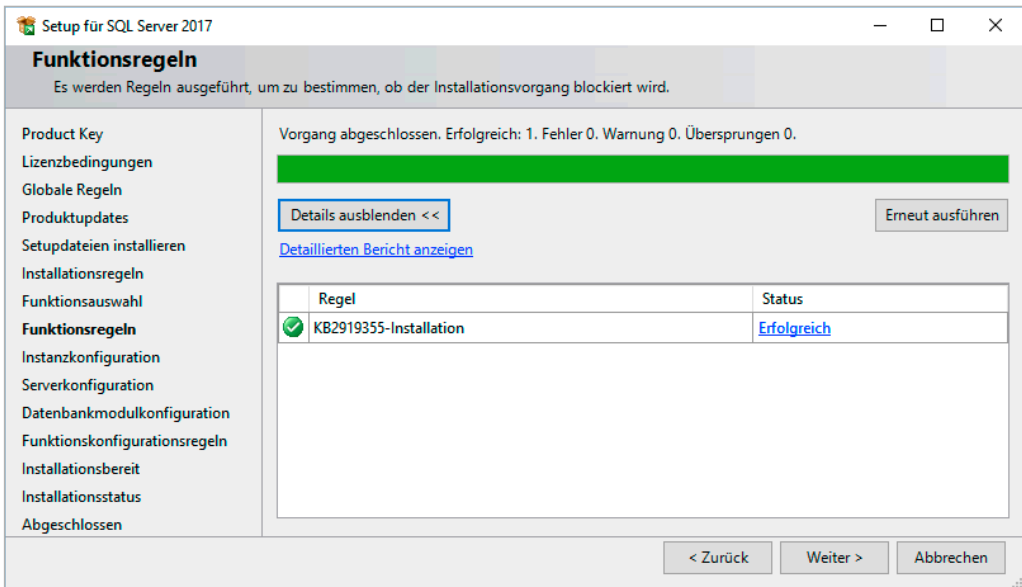


Bild 1.18 Prüfung der Funktionsregeln

Die *Instanzkonfiguration* ist der nächste Schritt. Dabei haben Sie die Wahl zwischen einer *Standardinstanz* oder einer *benannten Instanz*. Auf einem Rechner kann nur eine Standardinstanz installiert werden. Daher kann diese Option nicht mehr gewählt werden, falls eine solche bereits existiert. Bereits installierte Instanzen werden im unteren Bereich des Dialogs angezeigt.



HINWEIS: Bei der Installation der SQL Server Express Edition ist standardmäßig eine benannte Instanz mit dem Instanznamen SQLEXPRESS voreingestellt. Diese Vorgabe können Sie selbstverständlich nach Belieben abändern.

Falls Sie eine benannte Instanz installieren möchten, geben Sie bitte den Namen für die neue Instanz ein. Der Name darf bis zu 16 Zeichen umfassen und muss mit einem Buchstaben oder einem der Zeichen &, _ (Unterstrich) oder # (Raute) beginnen. Der Name darf nicht „Default“ oder „MSSQL Server“ lauten.

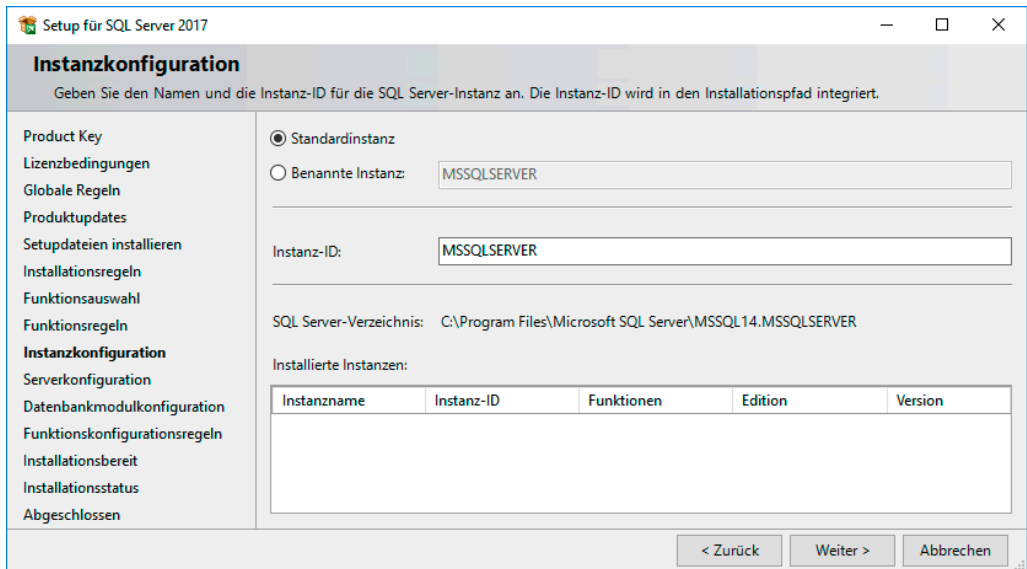


Bild 1.19 Konfiguration der Instanz

Zur Erläuterung sei darauf hingewiesen, dass beim SQL Server die Möglichkeit besteht, mehrere Instanzen auf einer Maschine zu installieren. Deshalb kann auch der Name der Instanz frei gewählt werden. Falls Sie einen besonderen Instanznamen verwenden möchten, müssen Sie auch bedenken, dass sich Clients dann stets mit dem Server gemäß der Syntax *Rechnername\Instanzname* verbinden.

Ich habe für meine neue Instanz – die erste auf dieser Servermaschine – die Option *Standardinstanz* gewählt und den Vorgang fortgesetzt.

Im folgenden Dialog geht es mit der *Serverkonfiguration* weiter. Der erste Schritt ist das Festlegen der Dienstkonten für die gewählten SQL Server-Dienste.

Bei der Verwendung des integrierten Systemkontos *Lokales System* oder eines der vorgegebenen internen Konten für die Dienste, sind diese nicht in der Lage, auf Ressourcen im Netzwerk zuzugreifen. Möchten Sie zum Beispiel automatisiert in der Nacht Daten exportieren, kann die Exportdatei nur auf dem Server selber, nicht aber auf einem Netzlaufwerk abgelegt werden. Verwenden Sie bitte ein Domänenkonto, um einerseits gezielt Berechtigungen im Netzwerk zu vergeben und um andererseits Berechtigungen auf dem lokalen

Rechner einzuschränken. Dies ist vor allem aus Sicherheitsgründen ratsam, falls der SQL Server auf einem Internetserver installiert wird.

Legen Sie daher alternativ ein Domänenkonto an und fügen Sie es der Gruppe der lokalen Administratoren hinzu. Möchten Sie die Rechte einschränken, so geben Sie dem Konto nur die vollen Zugriffsrechte auf die Datenverzeichnisse und das Recht zum Anmelden als Dienst. Für unsere Installation behalten wir die vorgeschlagenen Konten bei.



HINWEIS: Die den einzelnen Diensten zugeordneten Konten können später jederzeit über den SQL Server-Konfigurations-Manager geändert werden. Allerdings werden die speziellen Konten wie zum Beispiel *NT Service\MSSQLSERVER* beim Setup eigens eingerichtet und die Kennwörter sind für eine spätere Eingabe nicht bekannt.

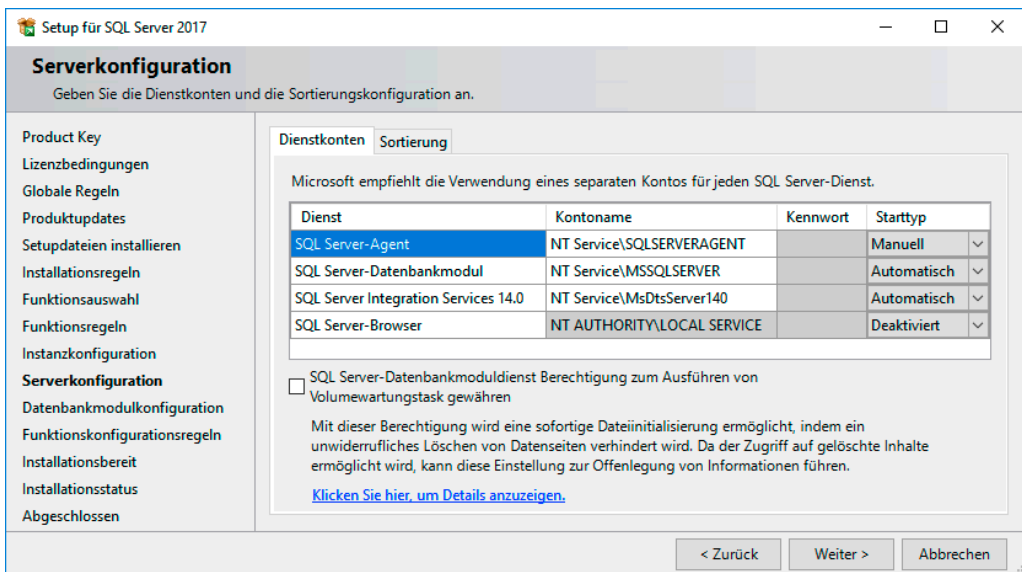


Bild 1.20 Dienstkonten auswählen

Auf dem zweiten Register mit der Beschriftung *Sortierung* wird die Standardsortierung für die SQL Server-Instanz eingestellt.

Hier sollten Sie die Standardeinstellung *LATIN1_GENERAL_CI_AS* übernehmen. Über die Sortierung wird die sogenannte *Collation* (Sortierungskennzeichner) eingestellt. Diese legt für Sortierungen und Vergleiche fest, welche Zeichen aufgrund der gewählten Ländereinstellung als gleich angesehen werden. So legen Sie je nach Auswahl beispielsweise fest, ob bei Suchvorgängen ein „ß“ einem „ss“ gleichgesetzt wird. Sie legen hier nur die Standardeinstellungen für den Server fest. Sie können für jede Datenbank später eine andere Standardeinstellung und sogar für jedes Feld einer Tabelle mit einem Character-Datentyp eine eigene Einstellung wählen. Außerdem können nur Spalten in Ausdrücken kombiniert oder in Bedingungen gemeinsam verwendet werden, falls diese dieselbe Einstellung aufweisen.

Die Einstellung innerhalb eines Ausdrucks lässt sich jederzeit mit der Anweisung `COLLATE` dynamisch anpassen.

Wenn Sie auf **ANPASSEN...** klicken, können Sie die Einstellung ändern. Das Kürzel `CI` in der Voreinstellung steht für *Case Insensitive*, da die Option *Unterscheidung nach Groß-/Kleinschreibung* (sinnvollerweise!) nicht ausgewählt ist. Das Kürzel `AS` steht für die gewählte *Unterscheidung nach Akzent*.

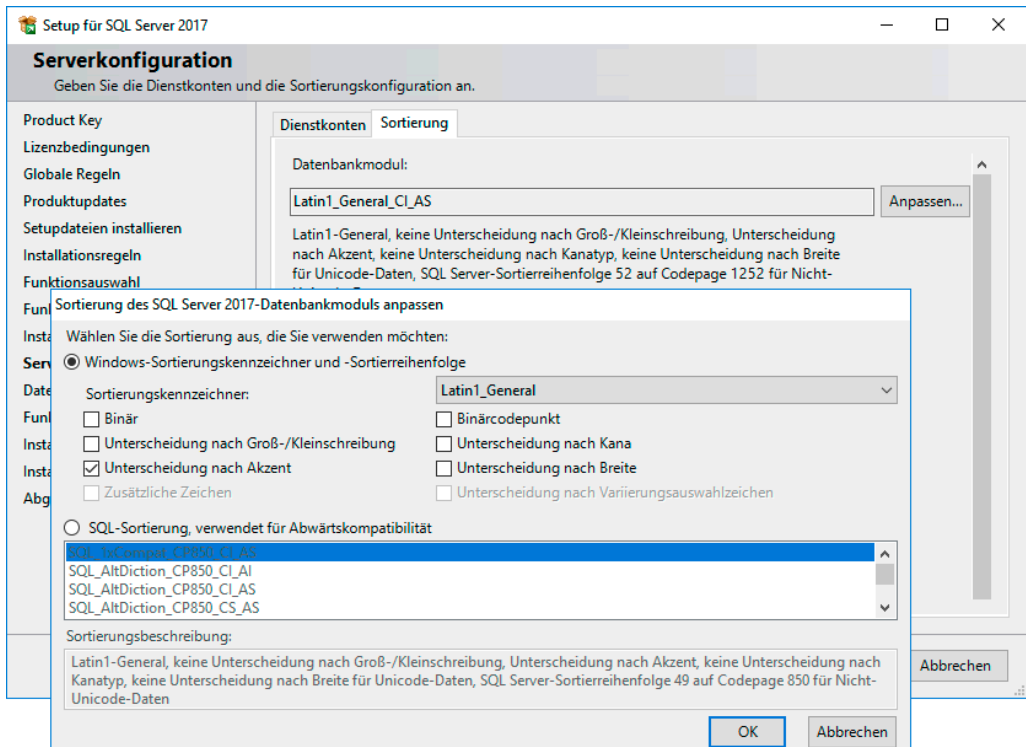


Bild 1.21 Sortierung anpassen



HINWEIS: Falls Sie sich fragen, was die Unterscheidung nach Kana und Breite bedeutet:

- *Kana* steht für die Unterscheidung der japanischen Kana-Zeichen nach Hiragana und Katakana.
- *Breite* steht dafür, ob zwischen ein und demselben Zeichen als Standard- oder Unicode-Zeichen unterschieden werden soll.

Auf die Serverkonfiguration folgt die *Datenbankmodulkonfiguration*. Unter dem Begriff *Serverkonfiguration* findet sich hier die Wahl des *Authentifizierungsmodus*. Es stehen zwei Modi zur Auswahl:

- *Windows-Authentifizierungsmodus*: Der Windows-Authentifizierungsmodus ermöglicht es Benutzern, mithilfe eines Windows-Domänenkontos eine Verbindung herzustellen. Es kann sich sowohl um ein Benutzer- als auch um ein Gruppenkonto handeln.
- *Gemischter Modus (SQL Server-Authentifizierung und Windows-Authentifizierung)*: Der gemischte Modus ermöglicht es Benutzern, mithilfe der Windows-Authentifizierung oder der SQL Server-Authentifizierung eine Verbindung zu einer Instanz von SQL Server herzustellen.

Benutzer, die eine Verbindung über ein Windows-Benutzerkonto herstellen, können entweder im Windows-Authentifizierungsmodus oder im gemischten Modus eine sogenannte vertraute Verbindung verwenden. Bei dieser werden die Informationen des Domänenkontos zur Authentifizierung verwendet. Die SQL Server-Authentifizierung wird für Benutzer benötigt, die über kein Domänenkonto verfügen oder von außen über VPN zugreifen und in dieser Konstellation kein Domänenkonto verwenden können.



HINWEIS: Ausführliche Informationen über die beiden Authentifizierungsmodi und die Anmeldeöglichkeiten finden Sie in Kapitel 10.

Übernehmen Sie hier bitte nicht die Standardeinstellungen, sondern wählen Sie den gemischten Modus und vergeben Sie ein Kennwort. Dieses wird dem Benutzer *sa* zugewiesen, der volle Systemadministratorberechtigungen am SQL Server besitzt. Mit dieser Auswahl sind Sie später etwas flexibler.

Schon seit der Version 2008 haben nicht mehr alle Mitglieder der lokalen Administratorengruppe automatisch Zugriff auf den SQL Server. Daher können – oder besser gesagt *müssen* – Sie schon beim Setup ein oder mehrere Konten, die Administratorzugriff bekommen sollen, bestimmen. Das Konto, mit dem Sie gerade arbeiten, sollten Sie über die Schaltfläche **AKTUELLEN BENUTZER HINZUFÜGEN** ergänzen. Damit stellen Sie sicher, nach der Installation selber vollen Zugriff zu haben. Beim Setup des SQL Server 2016 ist es zum Glück an dieser Stelle nicht möglich, im Setup fortzufahren, ohne einen Benutzer mit Administratorzugriff festzulegen. Sie müssen entweder den gemischten Modus aktivieren und ein Kennwort vergeben oder zumindest ein Domänenkonto angeben. Dies ist nicht unbedeutend, denn ich musste noch beim SQL Server 2008 so manche Schulungsraum-Installation aufwendig wieder reparieren, die mit unwissender „Weiter, weiter“-Installationsmethode ohne einen zukünftigen Administrator installiert worden ist.

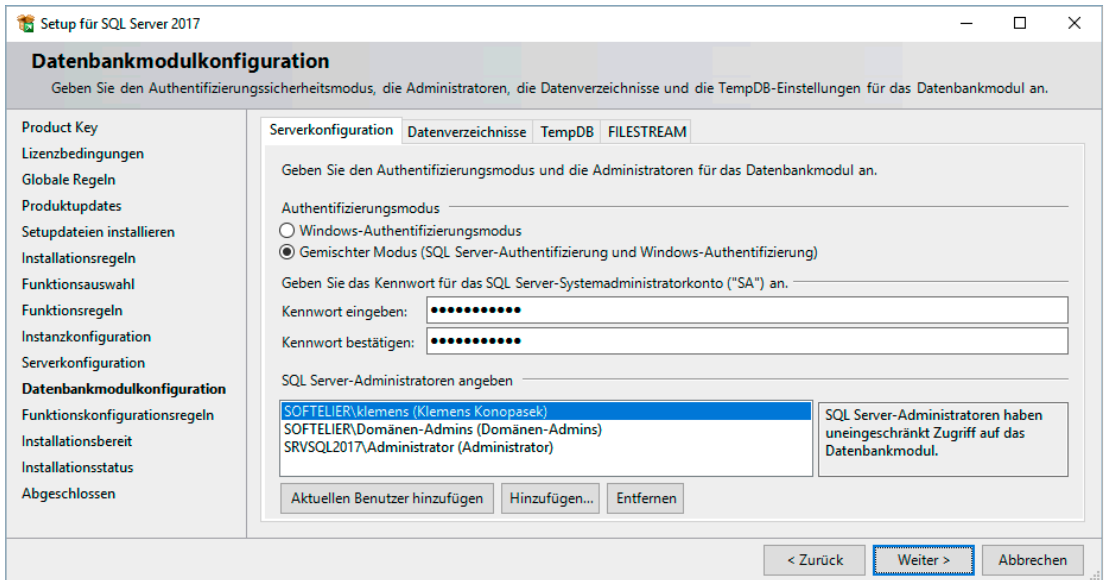


Bild 1.22 Sicherheit für SQL Server konfigurieren

Ein sehr wichtiger Schritt ist es, auf dem Register *Datenverzeichnisse* dieselben festzulegen. Sie können die Standardverzeichnisse an dieser Stelle für verschiedene Daten getrennt angeben. So lässt es sich zum Beispiel schon beim Setup festlegen, dass Datendateien und Protokolldateien auf unterschiedlichen Laufwerken abgelegt werden. Wenn Sie auf dem Register *Datenverzeichnisse* das Datenstammverzeichnis ändern, werden vorerst alle anderen Verzeichnisse nach einem fixen Schema vergeben, können dann aber noch einzeln angepasst werden. Ausgehend vom Datenstammverzeichnis wird folgendes Schema verwendet:

- MSSQL gefolgt von der Version, für den SQL Server 2017 ist dies 14.
- Mit einem Punkt getrennt kommt nun der Name der Instanz, sofern einer vergeben worden ist. Bei einer Standardinstanz wird MSSQLSERVER verwendet.
- Danach folgt der Unterordner MSSQL.
- Zum Schluss noch ein Unterordner mit dem Namen DATA für die Datendateien und BACKUP für die Sicherungen.



ACHTUNG! Ändern Sie hier bitte unbedingt das vorgegebene Datenstammverzeichnis ab. Daten haben schon per Definition unter *C:\PROGRAMME\...* nichts verloren! Selbst wenn Sie später Ihre Datenbanken beim Erstellen an anderen Orten speichern, liegen zumindest die Systemdatenbanken an dieser Stelle.

In unserem Beispiel habe ich direkt die Datenplatte *D:* als Datenstammverzeichnis angegeben, denn diese ist im Server auch nur dafür da.



PRAXISTIPP: Wenn Sie bei einem Produktivserver – bei Ihrer Entwicklungsmaschine spielt dies keine Rolle – die Möglichkeit haben, sollten Sie die Verzeichnisse für Datenbanken und Datenbankprotokolle auf unterschiedliche physische Datenträger legen. Dies bringt einerseits Performancevorteile und liefert andererseits besseren Schutz vor Datenverlust bei einem Plattencrash. Details zu diesem Thema lesen Sie in Kapitel 9.

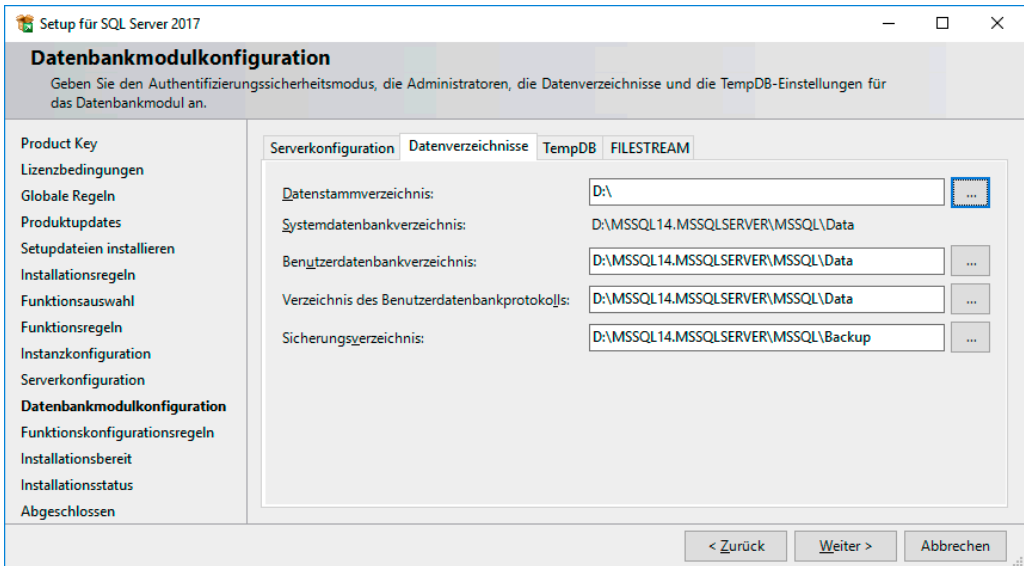


Bild 1.23 Datenverzeichnisse

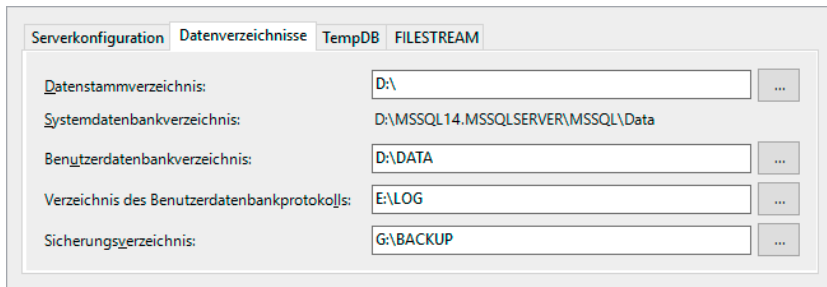


Bild 1.24 Unterschiedliche HDD für Daten und Protokolle



HINWEIS: Neu beim SQL Server 2016 und 2017 ist, dass für die temporäre Datenbank *TempDB* gleich bei der Installation mehrere Datendateien festgelegt werden können. Dadurch können temporäre Daten besser verteilt, und, wenn möglich, sogar auf unterschiedlichen Laufwerken abgelegt werden. Diese Parallelisierung bringt Vorteile bei der Performance des Gesamtsystems.

Da für die Verwaltung dieses neuen Features mehr Platz vonnöten ist, ist dafür im Dialog ein eigenes Register *TempDB* ergänzt worden. Legen Sie hier die Anzahl der Dateien, deren Anfangsgröße und deren Wert für die automatische Vergrößerung fest. Da wir uns in Kapitel 3 ausführlicher mit der Bedeutung dieser Werte auseinandersetzen werden, übernehmen wir an dieser Stelle die von Microsoft vorgeschlagenen Werte von zwei Dateien mit einer Anfangsgröße von 8 MB und einem automatischen Wachstum von 64 MB. Da die *TempDB* bei jedem Neustart des Server-Dienstes gelöscht und neu erstellt wird, besteht bei ihr nicht so sehr die Gefahr der Fragmentierung, wie dies bei Benutzerdatenbanken gegeben ist. Auch bisher konnte man beim SQL Server manuell weitere Datendateien zur *TempDB* hinzufügen. Neu in dieser Version ist, dass dies schon systematisch und automatisiert bei der Installation des Servers möglich ist.

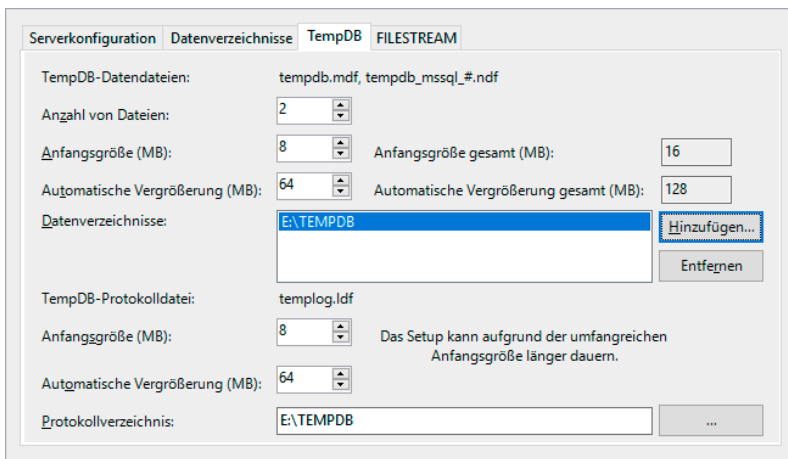


Bild 1.25 Anzahl und Speicherort für temporäre Datenbanken

Auf dem vierten Register können Sie *Filestream* für diese Instanz aktivieren. Dieses Feature, das ich Ihnen ausführlich in Kapitel 3 vorstelle, ermöglicht es, dass der SQL Server Binärdaten im Dateisystem und nicht in der Datenbank speichert. Aus Benutzersicht sieht es aber nach wie vor so aus, als wären die Daten in der Datenbank gespeichert. Als Freigabename wird der Instanzname vorgeschlagen. Dies ist der Name für den Share, mit dem später über das Netzwerk zugegriffen werden kann. Da mir persönlich hier die Vorgabe MSSQLSERVER nicht so gefällt, ändere ich den Namen auf SQLDATA ab. Damit kann der Zugriff später über den \\SERVERNAME\SQLDATA erfolgen.

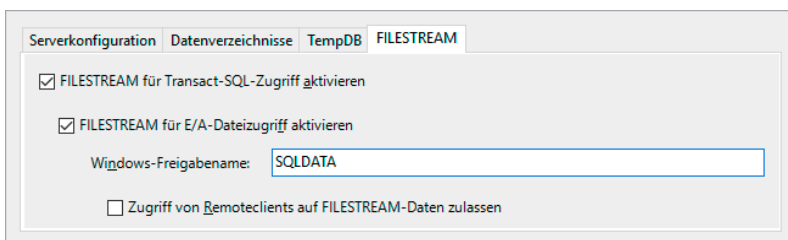


Bild 1.26 Filestream konfigurieren

Je nach Featureauswahl können noch weitere Konfigurationsdialoge folgen.

Nachdem nun alle Einstellungen für das Setup getroffen sind, werden die *Funktionskonfigurationsregeln* geprüft, um etwaige Probleme, die sich aus der getätigten Auswahl ergeben könnten, vorweg zu erkennen. Nach dieser Prüfung werden alle getroffenen Einstellungen noch einmal im Dialog *Installationsbereit* angezeigt, bevor der eigentliche Installationsvorgang startet.

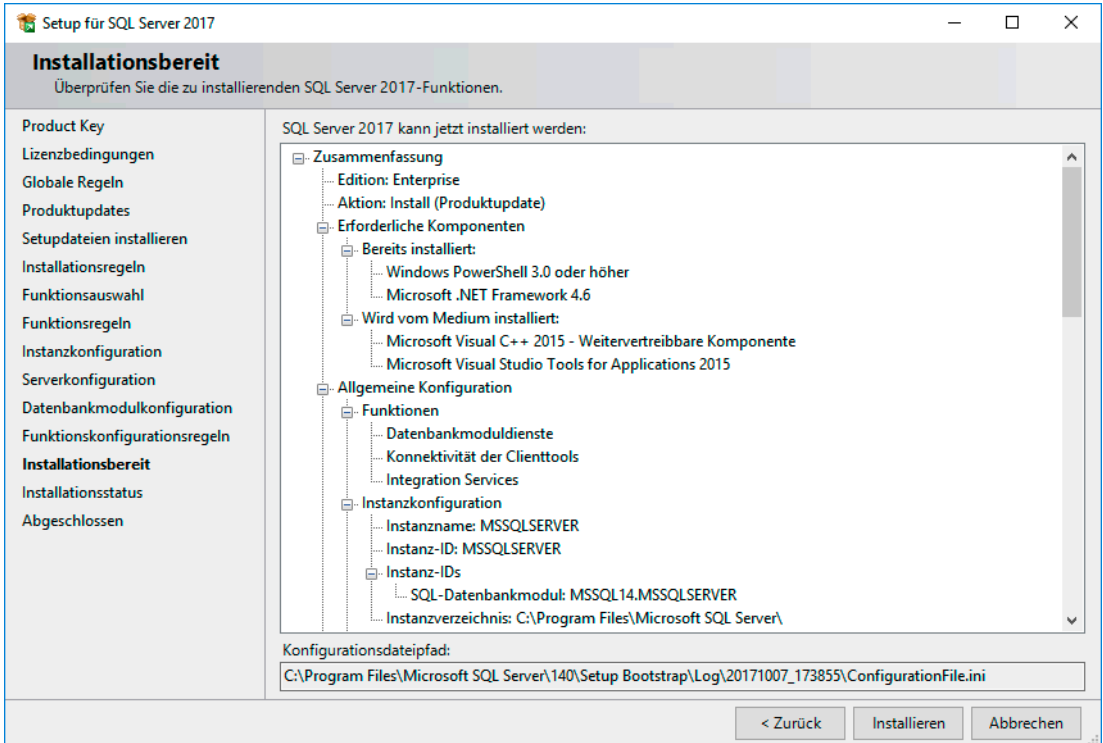


Bild 1.27 Zusammenfassung der Konfiguration

Nach Abschluss der Installation erhalten Sie eine Zusammenfassung mit einem Link zur Zusammenfassungsprotokolldatei.



PRAXISTIPP: Starten Sie nach der Installation bitte noch Windows Update, um eventuell weitere verfügbare Updates zu installieren. Auch wenn Sie bereits beim Setup die Option zur Einbindung von Updates gewählt haben, können Updates für andere Komponenten vorhanden sein.

Ergebnis der Installation

Bei der Installation von Microsoft SQL Server 2017 erstellt das Setup-Programm die in der folgenden Tabelle gezeigten Datenbank- und Protokolldateien (= Systemdatenbanken):

Tabelle 1.4 Datenbanken nach dem Setup

Datenbank	Datenbankdatei(en)	Protokolldatei
master	master.mdf	mastlog.ldf
model	model.mdf	modellog.ldf
msdb	msdbdata.mdf	msdblog.ldf
tempdb	tempdb.mdf, tempdb_mssql_#.ndf	templog.ldf

Die Datenbanken *master*, *model*, *msdb* und *tempdb* sind Systemdatenbanken, die nachfolgend noch beschrieben werden. Die Systemdatenbanken werden vom SQL Server benötigt und sollten möglichst unangetastet bleiben. Die Integrität dieser Datenbanken ist sehr wichtig, damit der SQL Server richtig funktioniert.

Die Systemdatenbanken sind in den beim Setup festgelegten Standardverzeichnissen zu finden. Zur Erinnerung: Der Standardspeicherort lautet bei einer Standardinstanz *MSSQL14.MSSQLSERVER\MSSQL\DATA* unter dem beim Setup angegebenen Datenstammverzeichnis, wenn Sie nicht manuell einen anderen Ordner angegeben haben. Bei einer benannten Instanz tritt der Instanzname anstelle von *MSSQLSERVER*.



HINWEIS: Die bekannte Beispieldatenbank *AdventureWorks* wurde mit dem SQL Server 2016 durch eine neue ersetzt. Diese neue Beispieldatenbank trägt den Namen *Wide World Importers (WWI)* und ist über GitHub mit dem Link <https://github.com/microsoft/sql-server-samples> verfügbar. Informationen über diese Beispieldatenbank finden Sie bei Microsoft auf dieser Seite: <https://docs.microsoft.com/de-de/sql/sample/microsoft-sql-server-samples>

Nachdem die Installation von SQL Server abgeschlossen ist, können mithilfe der grafischen Programme und der Eingabeaufforderungs-Dienstprogramme weitere Konfigurationen erfolgen. In der folgenden Tabelle wird die Unterstützung der Tools beschrieben, die zum Verwalten einer Instanz von SQL Server 2017 verwendet werden.

Tabelle 1.5 Übersicht über die Verwaltungstools

Tool bzw. Dienstprogramm	Erläuterung/Anwendung
Management Studio	Wird verwendet, um Abfragen zu bearbeiten und auszuführen sowie um die Standardtasks von Assistenten zu starten.
Konfigurations-Manager	Dient dem Verwalten der SQL Server-Dienste und Verbindungsprotokolle.
Profiler	Grafische Benutzeroberfläche zum Überwachen einer Instanz des SQL Server-Datenbankmoduls oder einer Instanz von Analysis Services
Datenbankoptimierungs-ratgeber	Unterstützt beim Erstellen einer optimalen Menge von Indizes, indizierten Sichten und Partitionen.

(Fortsetzung nächste Seite)

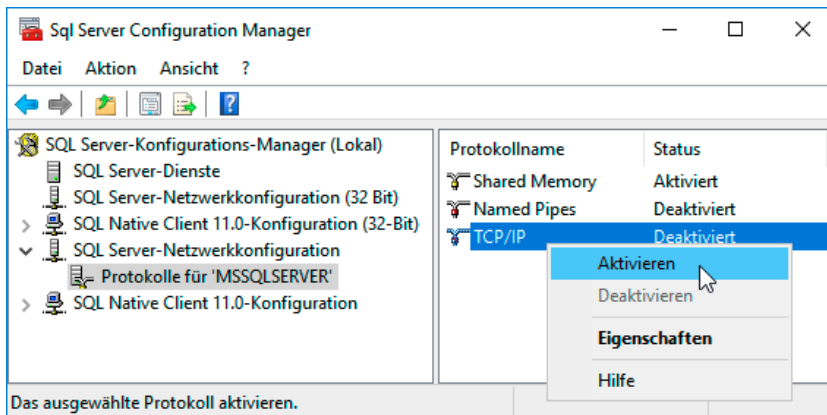
Tabelle 1.5 Übersicht über die Verwaltungstools (*Fortsetzung*)

Tool bzw. Dienstprogramm	Erläuterung/Anwendung
SQL Server Data Tools	Entwicklungsumgebung für Analysis Services, Integration Services und Datenbank-Engine
SQLCMD	Verwaltet SQL Server-Objekte von der Kommandozeile aus.
Datenimport und -export	Stellt einen Satz grafischer Tools und programmierbarer Objekte zum Verschieben, Kopieren und Transformieren von Daten bereit.

Direkt nach der Installation des SQL Server 2017 sind von den angeführten Tools vorerst nur der SQL Server 2016-Konfigurations-Manager und SQLCMD verfügbar. Die übrigen Tools stehen nach der separaten Installation des SQL Server Management Studios zur Verfügung. Die SQL Server Data Tools sind ein Bestandteil des Visual Studios. Sie können separat heruntergeladen und installiert werden. Befindet sich noch kein Visual Studio 2015 oder 2017 auf dem Rechner, wird eine abgespeckte Visual Studio Shell installiert und verwendet. Der Download kann von dieser Adresse erfolgen: <https://docs.microsoft.com/de-de/sql/ssdt/download-sql-server-data-tools-ssdt>

Um nach der Installation auf den SQL Server auch über ein Netzwerk zugreifen zu können, müssen eventuell noch folgende Einstellungen vorgenommen werden:

- Bei den Editionen Developer, Evaluation und Express ist der Zugriff über das Netzwerk standardmäßig deaktiviert. Um diesen zu erlauben, aktivieren Sie bitte das Serverprotokoll TCP/IP über den SQL Server-Konfigurations-Manager. Anschließend muss der Serverdienst neu gestartet werden, damit die Änderung aktiv wird.

**Bild 1.28** TCP/IP-Serverprotokoll aktivieren

- Da auf allen Host-Betriebssystemen typischerweise die Windows-Firewall aktiv ist, muss eine eingehende Regel erstellt werden, damit Zugriffe nicht vom Betriebssystem geblockt werden. Erstellen Sie bitte eine Ausnahme für den Port 1433, der standardmäßig vom SQL Server verwendet wird. Fügen Sie zusätzlich das SQL Server Management Studio als Programm hinzu, das durch die Firewall kommunizieren darf.

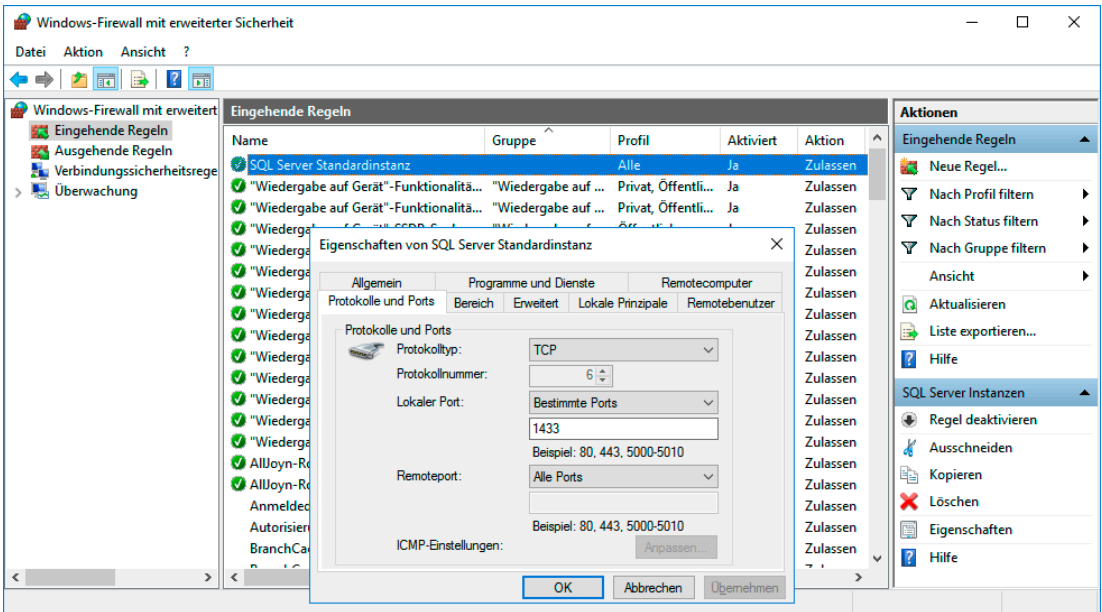


Bild 1.29 Port für SQL Server in Windows-Firewall freischalten

1.4 Datenbanken installieren und nutzen

Um eine vorhandene Datenbank nutzen zu können, müssen Sie diese vorab auf dem Server anfügen.



HINWEIS: Die Beispieldatenbank zu diesem Buch können Sie auf verschiedene Arten auf Ihr System bekommen. Die wichtigsten drei Arten sind:

- Erstellen Sie eine neue Datenbank mit dem Beispieldatenbank, die Beschreibung dazu finden Sie in Kapitel 3.
- Sie verwenden ein Backup der Datenbank, um diese damit wiederherzustellen. Informationen zum Wiederherstellen von Backups finden Sie in Kapitel 9.
- Sie verwenden die Datenbankdateien, um diese am Server zu registrieren. Die detaillierte Beschreibung der Vorgangsweise dazu finden Sie auch in Kapitel 9, die Schnellvariante zeige ich Ihnen im Anschluss.

Die für diese drei Varianten benötigten Dateien finden Sie bei den Beispieldateien zu diesem Buch.

Um die in diesem Buch verwendete Beispieldatenbank *wawi* auf Ihrem Server zu verwenden, gehen Sie bitte wie folgt vor:

1. Kopieren Sie die Dateien *wawi_data.mdf* und *wawi_log.ldf* in ein lokales Verzeichnis auf Ihrem Rechner, zum Beispiel in den Standard-Datenbankordner *MSSQL14.MSSQLSERVER\MSSQL\DATA*.
2. Starten Sie das SQL Server Management Studio (die Bedienung wird im folgenden Kapitel noch ausführlich erklärt) und melden Sie sich am zuvor installierten SQL Server mit der Windows-Authentifizierung an.

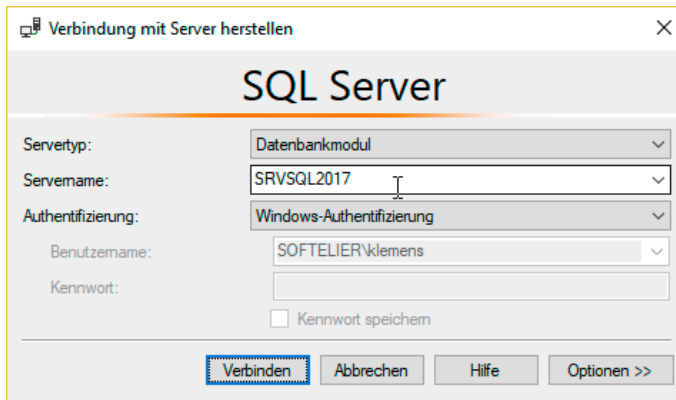


Bild 1.30 Mit Management Studio anmelden

3. Klicken Sie auf den Ordner *Datenbanken* mit der rechten Maustaste und wählen Sie den Befehl **ANFÜGEN...** aus.
4. Klicken Sie auf **HINZUFÜGEN...** und wählen Sie die gewünschte Datenbank, beispielsweise die primäre Datendatei *wawi_data.mdf* der Datenbank *wawi*.
5. Nach einem Klicken auf **OK** wird die gewählte Datenbank im Dialog angezeigt.
6. Bestätigen Sie abermals mit **OK**, um den Vorgang erfolgreich abzuschließen.

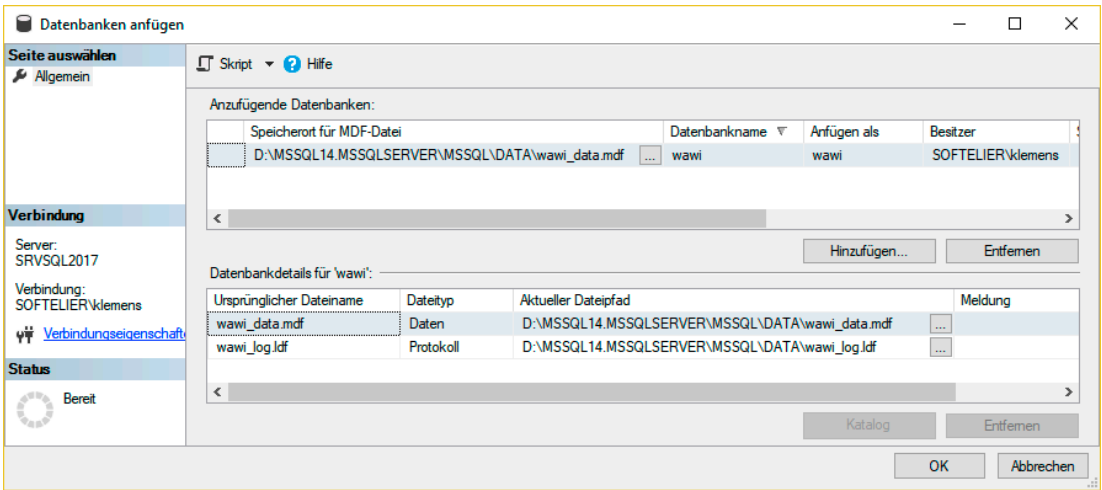


Bild 1.31 Datenbank anfügen

Als Ergebnis wird die neue Datenbank *wawi* im Objekt-Explorer angezeigt.

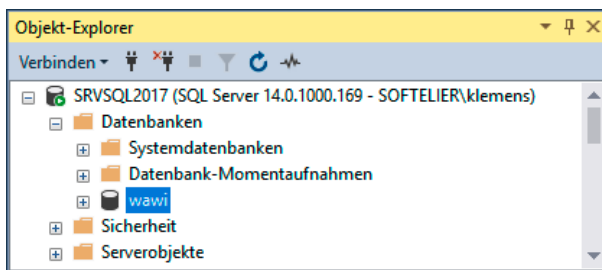


Bild 1.32 Neue Datenbank im Objekt-Explorer

Die beiden Datenbanken *ReportServer* und *ReportServerTempDB* sind deshalb vorhanden, weil wir die Reporting Services bei der Installation ausgewählt haben und eine Basiskonfiguration für diese durchführen haben lassen. Diese sind bei der Express Edition nicht verfügbar.



HINWEIS: In Bild 1.32 sehen Sie im Objekt-Explorer den Namen des Servers, mit dem sie verbunden sind. Hinter dem Servernamen werden in Klammern die Versionsnummer und der Name des verwendeten Logins angezeigt. Die Versionsnummer 14 steht für den SQL Server 2017. Die hinteren Nummern geben Aufschluss darüber, welche Patch-Variante bzw. welches Service Pack (Build Number) installiert ist. Die Versionsnummer des SQL Server 2017 RTM lautet 14.0.1000.169. Im Moment ist der aktuellste Patch das Sicherheitsupdate CU3 (Cumulative Update 3) mit der Versionsnummer 14.0.3015.40



PRAXISTIPP: Um immer auf dem aktuellsten Informationsstand über Updates zu allen aktuellen SQL Server-Versionen zu sein, legen Sie sich folgenden Link als Favorit in Ihrem Browser an:

<https://buildnumbers.wordpress.com/sqlserver>

Die Systemdatenbanken

Microsoft SQL Server-Systeme verfügen über vier Systemdatenbanken:

- master
- model
- msdb
- tempdb

Im Management Studio werden diese zur besseren Unterscheidung von den benutzererstellten Datenbanken durch den eigenen Ordner *Systemdatenbanken* getrennt.

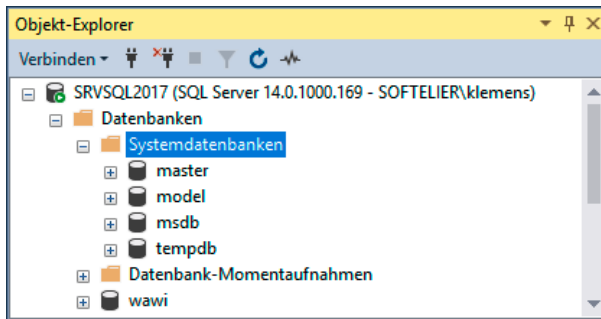


Bild 1.33 Systemdatenbanken

Die *master*-Datenbank enthält eine Aufzeichnung aller Informationen, die auf Systemebene für ein SQL Server-System wichtig sind. Dazu zählen

- alle Anmeldekonto und alle Systemkonfigurationseinstellungen,
- das Vorhandensein aller anderen Datenbanken, einschließlich der Speicherorte der Datenbankdateien,
- die Initialisierungsinformationen für den SQL Server.



PRAXISTIPP: Achten Sie bitte darauf, dass Sie stets über eine aktuelle Sicherung der *master*-Datenbank verfügen. Damit können Sie nicht nur die Benutzerdaten, sondern auch das System bei Bedarf wiederherstellen.

Die *model*-Datenbank wird als Vorlage für alle Datenbanken verwendet, die in einem System erstellt werden. Bei der Neuanlage einer Datenbank wird der erste Teil der Datenbank derart erstellt, dass der Inhalt der *model*-Datenbank kopiert wird. Anschließend wird der ver-

bleibende Teil der neuen Datenbank mit leeren Seiten aufgefüllt. Wichtig ist: Da die Datenbank *tempdb* bei jedem Start von SQL Server neu erstellt wird, muss die *model*-Datenbank immer in einem SQL Server-System vorhanden sein.

Die *msdb*-Datenbank wird vom SQL Server-Agent verwendet, um Termine für Warnungen und Aufträge zu planen und Operationen aufzuzeichnen. Auch Integration Services-Pakete können in dieser Datenbank gespeichert werden, damit sie direkt am Server ausgeführt werden können.

Jede Datenbank, einschließlich der Systemdatenbanken, verfügt über einen eigenen Satz von Dateien. Diese Dateien werden nicht mit anderen Datenbanken gemeinsam verwendet.

Die Datenbank *tempdb* nimmt alle temporären Tabellen und andere temporäre Objekte auf. Diese Datenbank wird darüber hinaus in allen anderen Situationen verwendet, in denen temporärer Speicherplatz benötigt wird, beispielsweise für Arbeitstabellen, die von SQL Server erzeugt werden. *tempdb* ist eine globale Ressource. Die temporären Tabellen und temporär gespeicherten Prozeduren für alle Benutzer, die eine Verbindung zum System hergestellt haben, werden in dieser Datenbank gespeichert. *tempdb* wird bei jedem Start von SQL Server neu erstellt; das System startet somit mit einer leeren Kopie dieser Datenbank. Da temporäre Tabellen und temporär gespeicherte Prozeduren beim Trennen von Verbindungen automatisch gelöscht werden und keine Verbindung aktiv ist, wenn das System heruntergefahren wird, enthält *tempdb* niemals Inhalte, die von einer SQL Server-Sitzung bis zur nächsten gespeichert werden. Standardmäßig wird *tempdb* während der Ausführung von SQL Server nach Bedarf vergrößert. Anders als andere Datenbanken wird *tempdb* jedoch bei jedem Start des Datenbankmoduls durch das Neuerstellen auf die Anfangsgröße zurückgesetzt.

Der erfolgreiche Betrieb von SQL Server hängt von der Integrität der Informationen in den Systemdatenbanken und Systemtabellen ab. Aus diesem Grund wird das direkte Aktualisieren der Informationen in einer Systemtabelle durch einen Benutzer nicht unterstützt.

Stattdessen stehen administrative Tools zur Verfügung, die Benutzern das umfassende Verwalten des Systems sowie aller Benutzer und Objekte in einer Datenbank ermöglichen. Benutzer können die administrativen Dienstprogramme (z. B. SQL Server Management Studio) verwenden, um das System direkt zu verwalten. Programmierer können die SMO-API (siehe dazu Kapitel 7) verwenden, um die vollständige Funktionalität für das Verwalten von SQL Server in ihre Anwendungen einzubinden. Programmierer, die Transact-SQL-Skripte und gespeicherte Prozeduren erstellen, können die gespeicherten Systemprozeduren sowie DDL-Anweisungen von Transact-SQL verwenden, um alle administrativen Funktionen in ihren Systemen zu unterstützen.

Datenbankobjekte in der SQL Server-Datenbank

Die folgende Aufstellung zeigt die wichtigsten Datenbankobjekte. Genauere Informationen finden Sie unter anderem in Kapitel 3.

Tabelle 1.6 Datenbankobjekte einer SQL Server-Datenbank

Ordner	Bedeutung/Anwendung
Datenbankdiagramme	Ermöglicht wird eine grafische Darstellung der Beziehungen zwischen den Tabellen einer SQL Server-Datenbank.
Tabellen	Tabellen sind Datenbankobjekte, die sämtliche in einer Datenbank enthaltenen Daten umfassen. Die Daten in den Tabellen sind, ähnlich wie in einer Kalkulationstabelle, in Zeilen und Spalten angeordnet.
Sichten	Eine Sicht ist eine virtuelle Tabelle, deren Inhalt durch eine Abfrage definiert wird. Wie eine echte Tabelle besteht auch eine Sicht aus einem Satz benannter Spalten und Zeilen mit Daten.
Synonyme	Synonyme können für andere Datenbankobjekte erstellt werden, um darauf mit diesen anstelle der Originalnamen zugreifen zu können.
Gespeicherte Prozeduren	Eine vorkompilierte Auflistung von Transact-SQL-Anweisungen, die unter einem Namen gespeichert und als Einheit verarbeitet wird. SQL Server stellt gespeicherte Prozeduren zum Verwalten von SQL Server und zum Anzeigen von Informationen über Datenbanken und Benutzer bereit.
Funktionen	Funktionen sind Unterroutinen, bestehend aus einer oder mehreren Transact-SQL-Anweisungen, die Code für die Wiederverwendung kapseln. SQL Server beschränkt Benutzer nicht auf die integrierten Funktionen, die im Rahmen der Transact-SQL-Sprache definiert sind. Benutzer können vielmehr ihre eigenen benutzerdefinierten Funktionen erstellen.
Datenbanktrigger	Datenbanktrigger sind ein Feature, mit dem Änderungen an anderen Datenbankobjekten protokolliert und gegebenenfalls auch verhindert werden können.
Assemblies	Stellen .NET-Code für die Verwendung über die Common Language Runtime zur Verfügung, um beispielsweise gespeicherte Prozeduren, die in .NET entwickelt worden sind, auszuführen.
Typen	Typen werden hier vom Benutzer selbst definiert. Dies ermöglicht es beispielsweise, spezielle von der Anwendung benötigte Daten zu speichern.
Regeln	Regeln werden als separate Objekte erstellt, die anschließend an die Spalte gebunden werden.
Standardwerte	Mit einer Standardeinschränkung können Sie einen Wert für eine Spalte definieren, der immer dann eingefügt wird, wenn ein Benutzer keinen Wert einträgt.
Planhinweislisten	Mit diesen kann im Rahmen der Optimierung von Abfrageleistung Einfluss auf das Erstellen von Ausführungsplänen für Anweisungen genommen werden.
Sequenzen	Diese Nummernspender können zum Generieren von eindeutigen Nummern verwendet werden, die beim Einfügen von Datensätzen gezogen und mit eingefügt werden. Diese sind an keine Tabelle oder Spalte fix gebunden und die gelieferten Werte können vielmehr beliebig in SQL-Anweisungen abgerufen und eingefügt werden.
Benutzer	Einzelpersonen und Gruppen, die Zugriff auf das System erhalten haben.
Rollen	Berechtigungsgruppen, die eingesetzt werden, um die Sicherheit zu gewährleisten.
Schemas	Ein Schema ist ein abgeschlossener Bereich, in dem Datenbankobjekte gespeichert werden.

■ 1.5 Gratis: die Express Edition

Auch von SQL Server 2017 ist wieder eine Express Edition erhältlich. Sie ist kostenlos und kann legal mit der eigenen Software vertrieben werden. Damit ist sie auch eine ernsthafte Konkurrenz für die eigene Desktop-Datenbank MS Access. Wenn man mit Access viel Erfahrung hat, verwendet man es idealerweise weiterhin als Frontend und setzt dazu den SQL Server als Backend ein.

Die wichtigsten Merkmale der Express Edition sind folgende:

- Die maximale Datenbankgröße beträgt 10 GB. Dieser Maximalwert betrifft jedoch nur eine einzelne Datenbank. Sie können also mehrere Datenbanken mit bis zu 10 GB Größe mit dieser Edition nutzen.
- Sie nutzt nur maximal 4 Prozessorkerne und maximal 1 GB RAM.
- Auch Speicheroptimierte Tabellen werden nun von der Express Edition unterstützt. Das sind Tabellen, die zur Gänze im RAM gehalten werden. Der Speicher für diese Tabellen ist auf 352 MB begrenzt.
- Reporting Services, die mit SQL Server Express with Advanced Services eingesetzt werden, sind auf die Nutzung von 4 GB RAM begrenzt.
- Mit *SQL Server Management Studio* steht das grafische Tool auch für die kleine Edition frei zur Verfügung. Welche Features verfügbar sind, hängt nur mehr von der Edition des verbundenen Servers ab.
- Leider gibt es in der Express Edition den *SQL Server-Agent* nicht, mit dem zeitgesteuerte Aufträge definiert werden. Dieses Feature fehlt in der Praxis tatsächlich, da es bei vielen Kleinstinstallationen zumindest für die tägliche automatische Sicherung Verwendung fände. Wenn Sie den Server-Agent benötigen, müssen Sie jetzt mindestens die Standard Edition einsetzen. (Lesen Sie über einen möglichen Workaround in Kapitel 9.)

Im Großen und Ganzen ist die Express Edition eine für viele Anwendungsfälle sehr gut einsetzbare Datenbank, die zudem noch kostenlos ist. Sie ist eine echte Alternative zu einer Desktop-Datenbank wie MS Access, da sie die klassischen Merkmale einer Server-Datenbank aufweist:

- Erhöhte Stabilität im Betrieb
- Reduktion der Netzlast durch Server-Datenbank
- Datensicherheit durch ein effizientes Sicherheitssystem
- Sicherung im Online-Betrieb
- Möglichkeit eines Disaster Recovery
- Serverseitige Programmierung durch Transact-SQL und CLR-Integration
- Protokollierung von Transaktionen

Die Express Edition des SQL Server 2017 steht unter den Adressen <https://www.microsoft.com/de-DE/download/details.aspx?id=55994> und <https://www.microsoft.com/de-de/sql-server/sql-server-downloads> in drei Varianten über den herunterzuladenden Installer zur Verfügung:

- *LocalDB*: Dies ist eine kleine Basisversion, die im Benutzermode gestartet wird und damit quasi wie eine Desktop-Datenbank lokal verwendet wird. Sie lässt sich direkt in mit dem Visual Studio erstellte Programme einbetten.
- *SQL Server 2017 Express*: In dieser Download-Variante ist ausschließlich das Datenbankmodul enthalten. Sie enthält keine grafischen Clienttools. Lediglich das Kommandozeilentool SQLCMD ist enthalten.
- *SQL Server 2017 Express mit Advanced Services*: In dieser Variante sind Volltextdienste sowie Reporting Services enthalten. Das Management Studio ist in dieser Version nicht mehr direkt enthalten und muss wie bei den anderen Editionen separat installiert werden.

Die Auswahl erfolgt hierbei erst, nachdem der Installer geladen und gestartet worden ist. Entweder installieren Sie direkt mit diesem Installer in einer der Varianten *Standard* oder *Benutzerdefiniert* oder Sie laden sich vorerst einmal die Installationsmedien herunter. Letzteres macht Sinn, wenn Sie die Installation auf mehreren Rechnern durchführen möchten.



Bild 1.34 Auswahl des Installationstyps

Praktisch finde ich, dass man beim Herunterladen nicht nur eine der drei zuvor beschriebenen Paketversionen auswählen kann, sondern auch noch die Sprachversion separat unabhängig von der Sprachversion des Installers wählen kann.

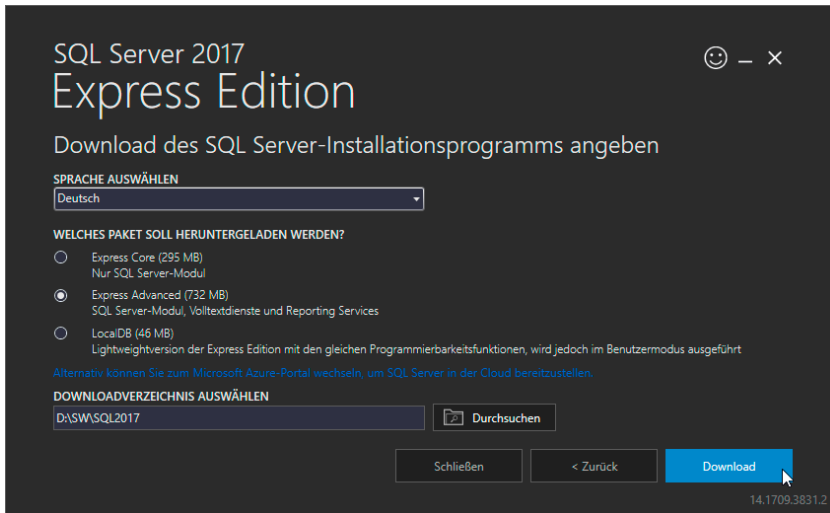


Bild 1.35 Download der Installationsdateien



HINWEIS: Die meisten der in diesem Buch behandelten Themen können in dieser Form auch mit der Express Edition ein- und umgesetzt werden. Wird ein Feature von der Express Edition nicht unterstützt, wird dies an der entsprechenden Stelle erwähnt.

■ 1.6 SQL Server Feature Pack

Auch für den SQL Server 2017 gibt es wieder, wie bereits bei den Vorversionen auch, das sogenannte Feature Pack. In diesem sind mehrere Downloads von Zusatztools und Treibern zusammengefasst. Diese sind entweder Teile des SQL Servers, die frei weitergegeben werden können, oder Add-ons. Alle diese Komponenten können separat heruntergeladen werden. Einige Komponenten sind gegenüber den Vorversionen hinzugekommen, aber es fehlen auch bisher bekannte wie zum Beispiel der SQL Server Native Client. Der SQL Server Migration Assistant ist hier auch nicht mehr zu finden, jedoch ist der Link zum Download ja nun in die Hauptmaske des Setup-Programms integriert.



HINWEIS: Sie finden das aktuelle Feature Pack für den SQL Server 2017 unter folgender Adresse:

<https://www.microsoft.com/de-DE/download/details.aspx?id=55992>

Hier ist ein Auszug an Elementen des Feature Packs, eine umfassende Auflistung finden Sie an der zuvor angegebenen Adresse:

- SQL Server Master Data Service-Add-in für Microsoft Excel
- Change Data Capture Designer und Service für Oracle von Attunity für Microsoft SQL Server
- Connector für SAP BW für Microsoft SQL Server
- Analysis Services-Clientbibliotheken
- SQL Server®-Berichts-Generator
- JDBC-Treiber für Microsoft SQL Server
- ODBC-Treiber für Microsoft SQL Server
- Treiber für PHP für Microsoft® SQL Server
- SQL Server® Semantic Language Statistics
- SQL Server® Data-Tier Application Framework
- System-CLR-Typen für SQL Server
- SQL Server® Transact-SQL Language Service
- SQL Server® Remote Blob Store
- SQL Server® Service Broker External Activator für Microsoft SQL Server
- Integration Services Feature Pack für Azure
- OLE DB-Anbieter für DB2 v6.0 für Microsoft SQL Server
- Report Viewer-Steuerelement für ASP.NET Web Forms-Apps
- Report Viewer-Steuerelement Windows Forms-Apps

Damit können Sie beispielsweise auf einem Rechner, auf dem keine SQL Server-Installation vorgenommen wird, aktuelle Treiber für den Serverzugriff (ODBC-Treiber) installieren. Für Entwickler, die andere Entwicklungsumgebungen nutzen, werden hier SQL Server-Treiber wie zum Beispiel für JDBC oder PHP bereitgestellt. Aber auch Dokumentationen und Ratgeber sind hier zu finden. Ich persönlich vermisse an dieser Stelle die Befehlszeilenprogramme. Hinter diesen verbergen sich das Tool SQLCMD, mit dem Sie über die Kommandozeile auf einen SQL Server zugreifen können, sowie das BCP-Hilfsprogramm, mit dem Sie Daten von einem SQL Server im angegebenen Format in eine Datei exportieren können. Aber diese Seite ist laufend in Veränderung, so können diese bereits verfügbar sein, wenn Sie diese Zeilen lesen. Bei Bedarf können Sie auf die Dateien des SQL Server 2016 zurückgreifen, die unter dieser Adresse zu finden sind: <https://www.microsoft.com/de-DE/download/details.aspx?id=52676>



HINWEIS: Der SQL Server Native Client, der neben den ODBC-Treibern auch jene für OLE DB bereitgestellt hat, steht ja bereits mit dem SQL Server 2014 nicht mehr zur Verfügung. Microsoft hat den OLE DB Provider (ADO) für den SQL Server mit der Version 2012 letztmalig bereitgestellt und danach abgekündigt (deprecated). Daher stehen an dieser Stelle nur mehr die ODBC-Treiber zur Verfügung. Dies hat starke Auswirkungen für Entwickler gehabt, denen Microsoft Ende der 90er-Jahre den Umstieg von ODBC auf ADO nahegelegt hat. Diese 180-Grad-Drehung hatte zur Folge, dass wieder ODBC für Transaktionsanwendungen favorisiert wurde. Dies war nicht ganz verständlich, da ADO in anderen Bereichen wie zum Beispiel für Analysis Services nach wie vor unterstützt wird, auch bei der aktuellen Version 2017. Sie können den Native Client des SQL Server 2012 noch einsetzen, er wird auch nach wie vor mit dem Setup des SQL Server 2017 installiert. Ende 2017 hat Microsoft sich entschlossen, die Abkündigung von OLE DB für den SQL Server wieder zurückzunehmen und hat für 2018 einen neuen Treiber, der auch alle Versionen nach 2012 – und vor allem viele deren neue Features – unterstützen wird, angekündigt. Suchen Sie im Web nach diesem, wenn Sie diese Zeilen lesen – vielleicht gibt es ihn inzwischen schon. Idealerweise wird er direkt beim Feature Pack verfügbar sein.

Wird beim Setup des SQL Servers die Option *Konnektivität der Clienttools* gewählt, findet sich neben dem ODBC-Treiber 13 auch der ODBC-Treiber des SQL Server Native Client 11 auf dem System. 11 ist ja die Versionsnummer des SQL Server 2012, 13 jene des SQL Server 2016. Aber auch der OLE DB-Provider des Native Client wird nach wie vor installiert.

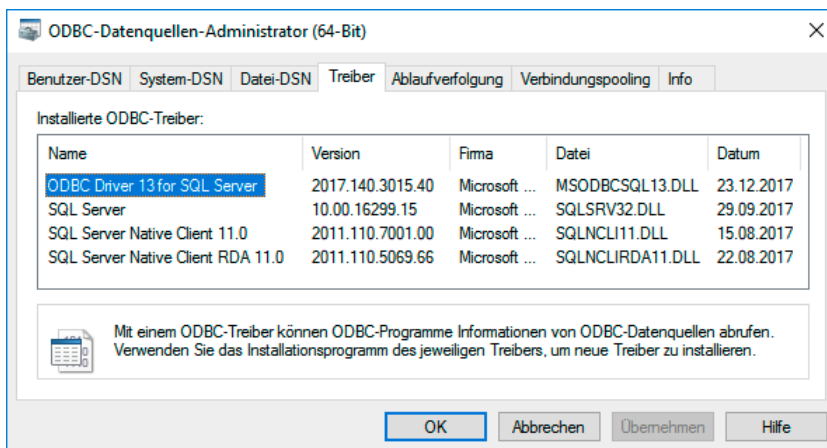


Bild 1.36 Installierte ODBC-Treiber

Wie in Bild 1.36 zu sehen, verwendet auch der mit dem SQL Server 2017 installierte ODBC-Treiber die Nummer 13 und nicht 14. Die neue Version ist aber über die Versionsnummer ersichtlich, die mit 2017.140.3015.14 der CU3-Versionsnummer des SQL Server 2017 entspricht.

2

Die grafischen Tools des SQL Server 2017

Eine der größten Stärken von Microsoft ist es, auch komplexe und komplizierte Sachverhalte und Produkte durch perfekte grafische Oberflächen vielen Benutzergruppen nahezubringen. Somit wird auch Einsteigern in ein Thema das Arbeiten mit Produkten möglich gemacht. Dem Profi wird durch einen einfachen Zugriff auf die Funktionen geholfen, die Arbeit möglichst effizient und effektiv zu erledigen. Manchmal kommt da schon der Gedanke: „Wenn das so einfach ist, kann das wirklich jeder!“

In diesem Kapitel möchte ich Ihnen die grafischen Tools für das Datenbankmodul des SQL Server 2017 vorstellen. Die Einfachheit der grafischen Tools ist mit dafür verantwortlich, dass trotz der Komplexität des Produkts im Hintergrund der SQL Server bei der Anwendung oft als sehr simpel empfunden wird. In der Praxis kann es manchmal aber auch nachteilig sein, wenn die Komplexität eines Produkts zu sehr verborgen wird. Daher verfolge ich mit diesem Buch auch das Ziel, Ihnen die Hintergründe möglichst verständlich aufzubereiten. Denn oft ist es kein Nachteil, zu wissen, was man eigentlich mit einem kleinen Mausklick in Bewegung setzt und was man damit bewirkt.



HINWEIS: In diesem Kapitel lernen Sie den grundlegenden Umgang mit den grafischen Tools. Über die Inhalte, die sich dahinter verbergen, erfahren Sie mehr in den nachfolgenden Kapiteln.

■ 2.1 Die Tools im Überblick

Zum Einstieg möchte ich Ihnen einen Überblick über die in diesem Kapitel beschriebenen Tools geben. In der Regel installieren Sie die grafischen Tools bei jeder Serverinstallation automatisch auf dem Server mit. Gerade bei Entwicklern wird der SQL Server oft auf der lokalen Maschine installiert, wodurch die Tools automatisch zur Verfügung stehen.



NEU: Wie schon im vorangegangenen Kapitel erwähnt, werden die grafischen Tools, die unter dem Überbegriff *Verwaltungstools* zusammengefasst werden, mit dem SQL Server 2016 und 2017 nicht mehr direkt über das SQL Server-Setup bereitgestellt. Sie müssen beziehungsweise können über folgenden Link bezogen werden: <https://docs.microsoft.com/de-de/sql/ssms/download-sql-server-management-studio-ssms>

Achten Sie darauf, dass sich der Downloadlink auf dieser Seite auch bei der deutschen Ausgabe – zumindest im Moment (das kann sich ja immer wieder ändern) – auf die englische Version bezieht, was Sie an der Endung *ENU* im Dateinamen erkennen. Um die deutsche Sprachversion zu bekommen, scrollen Sie auf dieser Seite etwas nach unten. Dort befinden sich die Links zu anderen Sprachversionen.

Ob Sie die Verwaltungstools auf einem dedizierten Server installieren, bleibt Ihnen überlassen. Microsoft empfiehlt es aus Sicherheitsgründen nicht. Jedenfalls werden Sie die Verwaltungstools auf Ihrem Arbeits-PC installieren. Ich persönlich habe das Management Studio immer gerne auch lokal auf dem Server, das ist manchmal ganz praktisch, wenn es Probleme beim Remotezugriff gibt. So ist zum Beispiel das Debuggen von Stored Procedures (Kapitel 7) lokal viel einfacher zu bewerkstelligen.

Starten Sie die Installation, haben Sie hierbei keine Auswahl, welche Tools Sie installieren möchten und welche nicht. Sie bekommen immer das gesamte Paket. Auch wenn bei der Installation lediglich das Management Studio genannt wird, werden sämtliche Verwaltungstools installiert, die sich zu diesem Zeitpunkt noch nicht auf dem Rechner befinden.

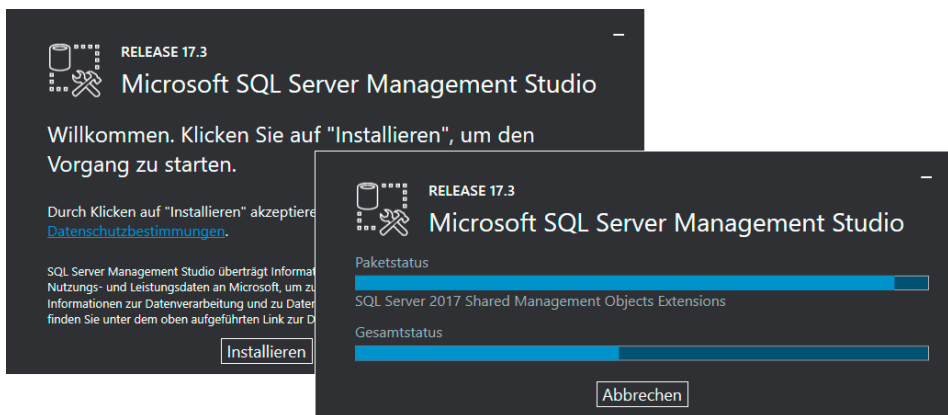


Bild 2.1 Verwaltungstools installieren

- **SQL Server Management Studio:** Das Management Studio ist quasi das Hauptwerkzeug des SQL Server 2017. Es basiert auf der Shell von Visual Studio 2015 und kommt daher Entwicklern von der Optik und der Handhabung her oft bekannt vor. Es ist sowohl das Haupttool zur Verwaltung als auch dasjenige zur Entwicklung. Der integrierte Abfrageditor dient zum Erstellen und Generieren von Abfragen und kann somit auch von

Nichtspezialisten verwendet werden. Es unterstützt nicht nur das Datenbankmodul, sondern auch andere Dienste wie zum Beispiel die Integration oder Analysis Services. Es gibt nicht mehr wie früher unterschiedliche Varianten, sondern nur mehr eine, die jene Features unterstützt, die auf der verbundenen Instanz des SQL Servers verfügbar sind.

- **SQL Server-Konfigurations-Manager:** Der Konfigurations-Manager dient zur Einstellung der verwendeten Netzwerkprotokolle und zur Konfiguration der SQL Server-Dienste.
- **SQLCMD:** Das Tool SQLCMD (SQL Command) ist das Kommandozeilentool für den Einsatz von SQL und Transact-SQL.
- **SQL Server-Installationscenter:** Das Installationscenter wird für das nachträgliche Ergänzen von Features zu einer existierenden Installation verwendet.
- **SQL Server Profiler:** Der Profiler dient zur Überwachung von Anweisungen und Prozessen auf dem Server. In erster Linie übernimmt er die Auswertung der Leistung und die Analyse der ausgeführten Anweisungen.
- **Datenbankoptimierungsratgeber:** Mithilfe des Optimierungsratgebers können Sie Ihre Datenbank analysieren und Tipps für Verbesserungen der Leistung erhalten.
- **SQL Server Data Tools:** Dieses Tool bietet Entwicklern die Möglichkeit, mit einem Tool alle Bereiche der Datenbankentwicklung abzudecken. Außerdem ermöglicht es das Entwickeln von .NET-Prozeduren ohne eine Vollversion des Visual Studios.



HINWEIS: Im Bereich der SQL Server Data Tools ist es zu einer sinnvollen Vereinheitlichung gekommen. Vielleicht fragen Sie sich, was aus den **SQL Server Data Tools BI** geworden ist. Diese hat es ja zuvor schon unter verschiedenen Namen gegeben (*Visual Studio Shell* und *Business Intelligence Development Studio*). Über diese spezielle Form des Visual Studios werden Projekte für Analysis Services, Integration Services und Reporting Services erstellt. Aufgrund des ähnlichen Namens konnten die SQL Server Data Tools und die SQL Server Data Tools BI leicht verwechselt werden, auch wenn es sich dabei um zwei eigenständige Werkzeuge gehandelt hat. Jetzt sind alle bisherigen Funktionalitäten der beiden unter den SQL Server Data Tools zusammengefasst und es gibt nur mehr dieses eine Werkzeug. Entweder werden die entsprechenden Projektvorlagen je nach Wahl mit einer Shell des Visual Studios 2015 oder 2017 installiert oder in die jeweilige Version des Visual Studios integriert, wenn dieses auf dem Rechner bereits installiert ist.

- **Import/Export-Assistent:** Ein komfortabler Assistent für den einfachen Import und Export von Daten in oder aus einer Datenbank.
- **Bereitstellungsassistent für Integration Services:** Mit diesem Assistenten können Integration Services-Projekte einfach in einen Integration Services-Katalog auf einem Server übertragen werden.
- **Paketausführungsprogramm:** Mit diesem können Integration Services-Pakete lokal auf einem Client ausgeführt werden. Über das Erstellen von solchen Paketen lesen Sie in Kapitel 11.

- **Projektkonvertierungs-Assistent:** Mit diesem Assistenten können Integration Services-Pakete von Vorversionen für die aktuelle Version konvertiert werden.
- **Integration Services Datenfeedveröffentlichungs-Assistent:** Die Ausgabe eines Integration Services-Pakets kann mithilfe dieses Assistenten als Sicht (View) in einer Datenbank zugänglich gemacht werden. Auf diese Sicht kann konventionell über SQL-Abfragen zugegriffen werden.



HINWEIS: Ein ganz neues Tool, das im Moment nur als Vorschauversion verfügbar ist, ist das *SQL Operations Studio*. Dieses Werkzeug, das dem SQL Server Management Studio ähnlich ist, ist sowohl für Windows, Linux und MacOS verfügbar. Das SQL Operations Studio ist nicht Bestandteil der Verwaltungstools und über einen eigenen Download verfügbar. Damit wird die Ausweitung auf andere Betriebssysteme neben dem SQL Server 2017 für Linux auf Clientebene ebenso fortgesetzt.

■ 2.2 Das Management Studio

Wie bereits erwähnt, ist das Management Studio dasjenige Tool des SQL Server 2017, mit dem Sie wohl am meisten arbeiten werden, da es sowohl zur Verwaltung, zur Entwicklung als auch zum Erstellen von Abfragen eingesetzt wird. Dieses Tool verwendet, wie erwähnt, das grafische Interface von Visual Studio.

Anmelden

Beim Starten des Management Studios müssen Sie sich erstmalig anmelden. In der Regel melden Sie sich am Datenbankmodul (Database Engine) an. Da das Management Studio nicht nur zur Verwaltung der Datenbanken, sondern auch der weiteren SQL Server-Services dient, steht im Anmeldedialog als Servertyp Folgendes zur Wahl:

- **Datenbankmodul:** Dies ist die Standardauswahl für das Arbeiten mit Datenbanken.
- **Analysis Services:** Ermöglicht den Zugriff auf die Verwaltung der Analysis Services zur Erstellung von Data Warehouse-Anwendungen.
- **Reporting Services:** Die Reporting Services erlauben es Ihnen, Berichte, die auf Daten des SQL Servers basieren, in verschiedenen Formaten auszugeben und Benutzern beispielsweise über einen Browser zur Verfügung zu stellen.
- **Integration Services:** Die Integration Services werden für das Abbilden von Workflows verwendet, wodurch sich externe Daten und Funktionen integrieren lassen.

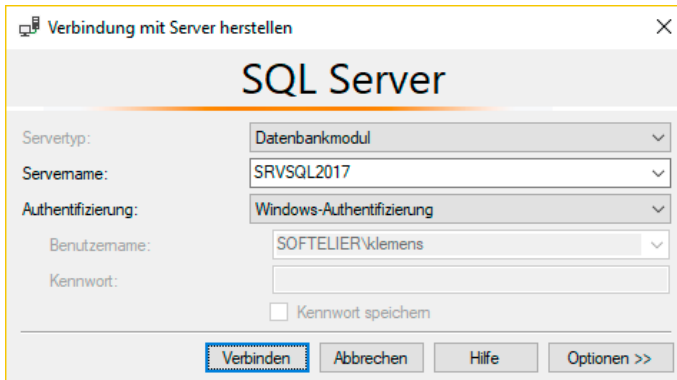


Bild 2.2 Anmeldedialog

Melden Sie sich am Datenbankmodul über Windows-Authentifizierung oder über SQL Server-Authentifizierung mit Eingabe Ihres Benutzernamens und Ihres Kennworts an.



HINWEIS: Detaillierte Informationen über Authentifizierungsmodi und Anmeldungen finden Sie in Kapitel 10.

Warum können Sie sich bei einem frisch installierten SQL Server lokal anmelden?

- Sie haben beim Setup bei der Option zum Hinzufügen von Benutzern mit Administratorzugriff Ihren Windows-Benutzer oder eine Gruppe, deren Mitglied Sie sind, hinzugefügt. Nun können Sie sich mit dem Authentifizierungstyp *Windows-Authentifizierung* anmelden.
- Wenn Sie beim Setup bereits den gemischten Modus gewählt haben, können Sie sich auch mit dem Anmeldenamen *sa* und dem im Setup vergebenen Kennwort über die Verwendung der SQL Server-Authentifizierung anmelden.



ACHTUNG! Um sich von einem Remote-Rechner aus anzumelden, müssen Sie sich entweder über eine der eben genannten Methoden anmelden oder sich zuvor lokal anmelden und ein neues Konto für den Remotezugriff erstellen. Dies ist zum Beispiel notwendig, wenn Sie bisher nur ein lokales Konto für den Zugriff zur Verfügung haben. Wie das funktioniert, lesen Sie in Kapitel 10. Achten Sie auch darauf, dass die für eine Remoteanmeldung notwendigen Serverprotokolle aktiviert und Firewall-Ausnahmen gesetzt sind. Die diesbezügliche Vorgangsweise finden Sie in Kapitel 1 beschrieben.

Wählen Sie den Server, an dem Sie sich anmelden möchten, aus. Wird er in der Auswahlliste noch nicht angezeigt, so wählen Sie *<Suche fortsetzen...>*, um einen Server zu suchen.

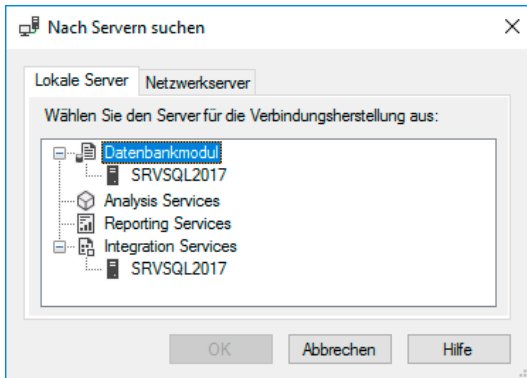


Bild 2.3 Lokalen Server auswählen

Auf dem ersten Register werden die lokal installierten Server – es können ja mehrere Instanzen auf einem Rechner installiert werden – angezeigt. In der Abbildung findet sich eine lokale Standardinstanz auf meinem Rechner. Falls Sie hier Instanzen vorfinden, deren Vorhandensein Sie sich nicht erklären können, bedenken Sie, dass häufig durch andere Programme Express-Instanzen installiert werden. Dies könnte zum Beispiel bei der Installation des Visual Studios erfolgt sein.

Auf dem Register *Netzwerkserver* werden Installationen im Netzwerk gesucht und angezeigt.

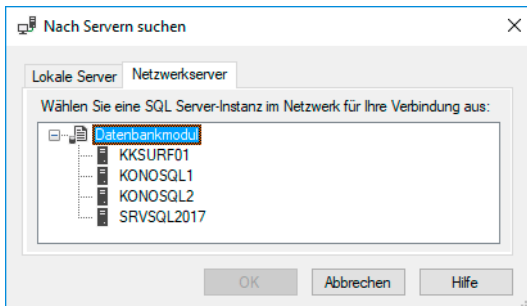


Bild 2.4 Netzwerkserver verbinden



HINWEIS: Sie können sich mit dem Management Studio Version 17 sowohl an einem SQL Server 2017 als auch an einer der Vorgängerversionen anmelden. Sie können sich an einem SQL Server 2017 auch mit einem Management Studio der Version 2014 oder der Vorgängerversion 16 anmelden. Allerdings können Sie manche Aktionen hiermit nicht vornehmen. Aber wenn Sie in erster Linie Abfragen ausführen möchten, stellt dies kein Problem dar.



PRAXISTIPP: Sie können sich an einer lokalen Standardinstanz auch durch manuelle Eingabe eines Punktes oder mittels (*local*) anmelden. Ist der Server über das Netzwerk nicht anders erreichbar, können Sie auch die IP-Adresse des Servers verwenden. Wenn Sie auf eine benannte Instanz zugreifen, müssen Sie den Instanznamen mit einem Schrägstrich ergänzen. Die Express Edition legt beim Setup standardmäßig eine benannte Instanz an, auf die Sie dann mit *Rechnername/SQLEXPRESS* zugreifen. Die Groß-/Kleinschreibung der Serverbezeichnung spielt hierbei keine Rolle.

Nach erfolgter Anmeldung werden Ihnen im Management Studio folgende Fenster angezeigt:

- **Objekt-Explorer:** Hier sehen Sie alle Objekte des Servers, an dem Sie sich angemeldet haben.
- **Details zum Objekt-Explorer:** Hier werden Details zur jeweiligen Auswahl im Objekt-Explorer angezeigt. Manchmal wird dieser nicht angezeigt, dann können Sie die Details über das Menü **ANSICHT** oder die Taste (**F7**) einblenden. Der Vorteil dieser Details ist nicht nur, dass man mehr an Informationen auf einen Blick sieht als im Objekt-Explorer, sondern dass es in manchen Situationen praktisch ist, mehrere Objekte gleichzeitig markieren und bearbeiten zu können.

Funktionen für angezeigte Server, Datenbanken und Datenbankobjekte sind über das Kontextmenü zugänglich. Daher nimmt das Kontextmenü bei der Verwendung des Management Studios, vor allem im Objekt-Explorer, eine zentrale Rolle ein.

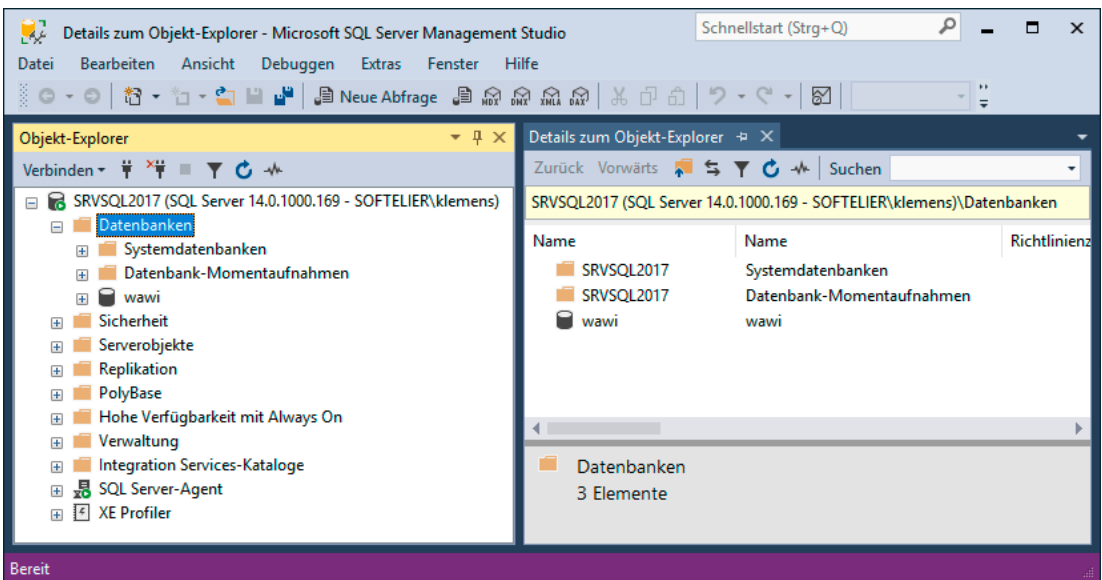


Bild 2.5 Management Studio

Neben den beiden erwähnten Fenstern können sinnvollerweise noch folgende Fenster angezeigt werden:

- Registrierte Server
- Vorlagen-Explorer
- Projektmappen-Explorer
- Eigenschaftsfenster

Weitere Fenster sind verfügbar, werden aber teilweise nur für Spezialaufgaben und daher sehr selten oder gar nicht benötigt. So zum Beispiel das Fenster *Aufrufhierarchie*, das Sie lediglich beim Debuggen von Transact-SQL-Code benötigen.

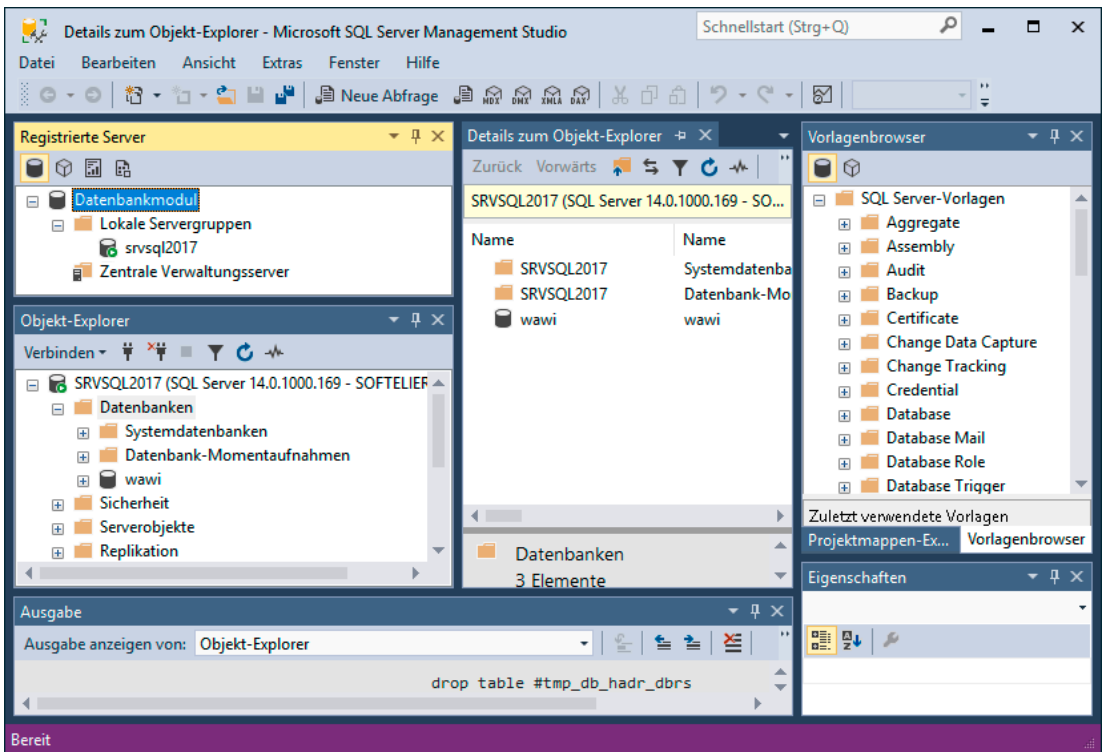


Bild 2.6 Fenster des Management Studios

Standardmäßig sind oftmals mehrere Fenster zu einer Gruppe mit Registern zusammengefasst, wie zum Beispiel der Projektmappen- und der Vorlagen-Explorer.

Die einzelnen Fenster können angedockt, verschoben und auf Register gelegt werden. Blenden Sie die gewünschten Fenster vorerst über das Menü **ANSICHT** ein, sie erscheinen an ihrer Standardposition. Über das Kontextmenü im Titelbereich der Fenster können Sie die Verankerung aufgeben, das Fenster wieder andocken oder als Dokument im Registerkartenformat andocken. Deutlich komfortabler funktioniert dies aber mit der Maus. Ziehen Sie das Fenster ein wenig, indem Sie es an der Titelleiste anfassen, und schon erscheinen die Marken zur Platzierung. Damit können Sie das Fenster über die vier äußeren Marken ent-

weder am Rand oder über die kreuzförmig über dem aktuell darunter liegenden Fenster angeordneten Marken relativ zu diesem Fenster platzieren. Der Schatten bietet eine Vorschau, so wird das wunschgemäße Platzieren zum Kinderspiel. Bild 2.7 zeigt, wie Sie zum Beispiel das Fenster *Registrierte Server* an einer neuen Position als eigenständiges Fenster rechts neben dem Objekt-Explorer andocken. Programmierern wird dieses Vorgehen von Visual Studio her bestens bekannt sein. Sonst spielen Sie ein wenig mit den Möglichkeiten, Sie werden die Funktionsweise im Nu beherrschen.

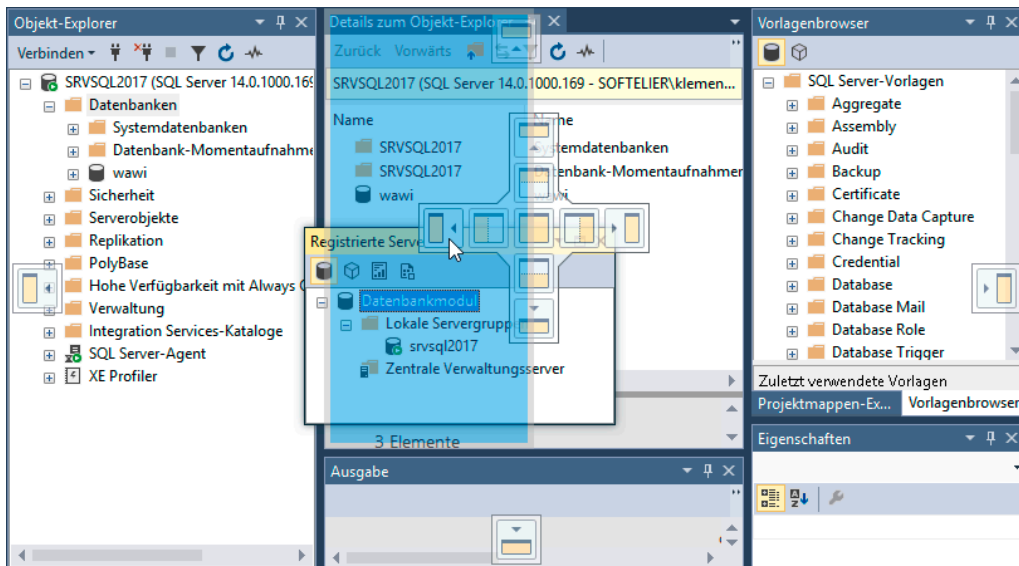


Bild 2.7 Fenster an neuer Position andocken



PRAXISTIPP: Blenden Sie nur die Fenster ein, die Sie wirklich benötigen.

Vorteilhaft sind Bildschirme mit einer hohen Auflösung, um bequem arbeiten zu können.

Wenn Sie sich den Luxus leisten können, dann ist das Arbeiten mit zwei Bildschirmen etwas sehr Praktisches. Gerade als Datenbankentwickler benötigt man oft gleichzeitig Zugriff auf das Datenbanksystem und die Entwicklungsumgebung. So sind zum Beispiel das SQL Server Management Studio einerseits und das Visual Studio andererseits geöffnet. Auch ich verwende für meine Arbeit zwei Monitore und kann diese Arbeitsweise nur empfehlen. Man könnte sich zwar in manchen Situationen das Management Studio durch den Einsatz der SQL Server Data Tools ersparen, aber ich persönlich erledige Datenbankaufgaben viel lieber mit dem Management Studio. Seit ich mein Notebook endgültig durch ein leistungsstarkes Windows-Tablet ersetzt habe, verwende ich de facto sogar drei Bildschirme. Damit hat der Mail-Client auch noch seinen eigenen Platz.

Sehr praktisch beim Einsatz von mehreren Bildschirmen ist die Möglichkeit, ein Fenster aus dem Management Studio auch ganz herauszuziehen und als eigenständiges Fenster auf

den anderen Monitor zu verschieben. Ich nutze diese Möglichkeit häufig für einzelne Abfrageeditor-Fenster. Sie sehen ein solches in Bild 2.8 herausgezogen.

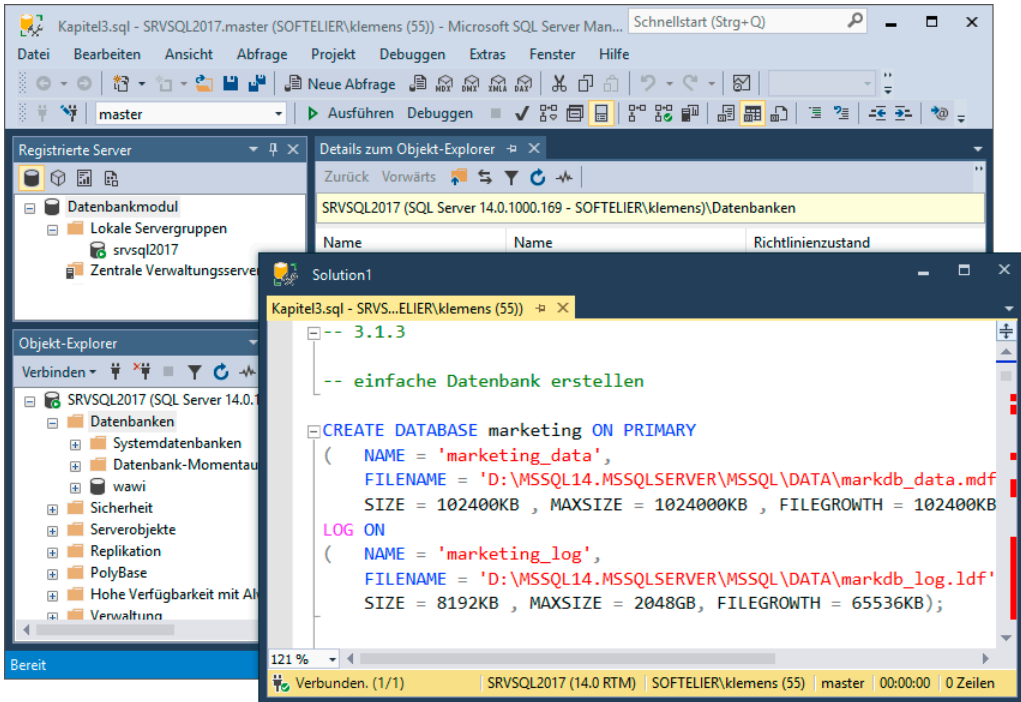


Bild 2.8 Einzelnes Fenster herausziehen

Server registrieren

Wenn Sie regelmäßig mit mehreren Servern arbeiten, empfiehlt sich die Verwendung des Fensters *Registrierte Server*. Dies erleichtert den Zugriff auf verschiedene Server im Netzwerk. Blenden Sie es über den Menübefehl **ANSICHT/REGISTRIERTE SERVER** ein. Alternativ können Sie zum Einblenden auch die Tastenkombination **STRG+ALT+G** verwenden.

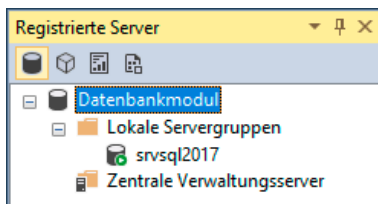


Bild 2.9 Registrierte Server

Sie können hier über das Kontextmenü neue Server registrieren. Um eine bessere Übersicht und Ordnung zu erzielen, können Sie nach Belieben Servergruppen (Ordner) erstellen, denen Sie dann die registrierten Server zuweisen. Um eine Servergruppe zu erstellen, wäh-

len Sie den Ordner *Lokale Servergruppen* und im Kontextmenü den Befehl **NEUE SERVERGRUPPE...** aus.

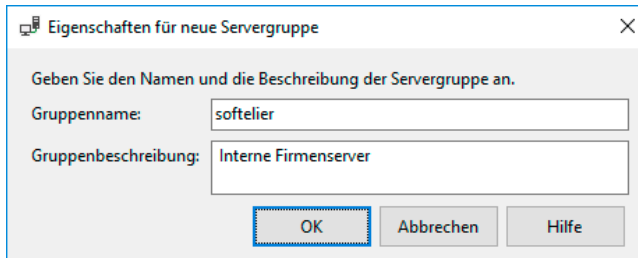


Bild 2.10 Neue Servergruppe anlegen

Im Dialog geben Sie den Namen für die Servergruppe an. Den Ort, unter dem die neue Gruppe eingefügt werden soll, können Sie in dieser Version in diesem Dialog nicht angeben. In einer früheren Version ist dies noch möglich gewesen. Legen Sie also einen Unterordner an, so müssen Sie von vornherein den korrekten übergeordneten Ordner auswählen. Allerdings können Sie einen angelegten Ordner dann nachträglich verschieben und damit in einen anderen integrieren.

Registrieren Sie neue Server über den Befehl **NEUEN SERVER REGISTRIEREN...**, den Sie ebenfalls im Kontextmenü finden. Wie bei der Anmeldung am Datenbankmodul zuvor geben Sie den Servernamen und die Anmeldeinformationen ein.

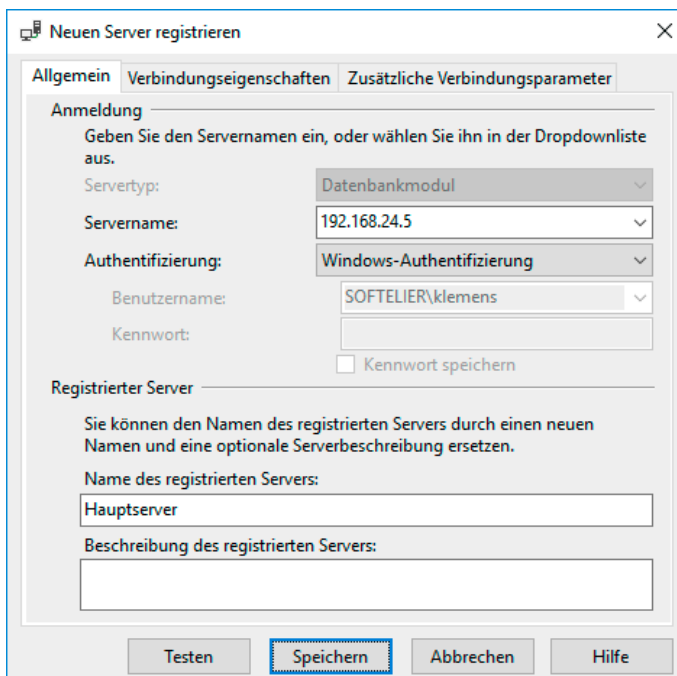


Bild 2.11 Neue Serverregistrierung



PRAXISTIPP: Sie können bei der Registrierung einen Namen für die Registrierung frei vergeben. Dies ist besonders dann hilfreich, wenn Sie einen Server über die IP-Adresse registrieren müssen, falls er über das Netzwerk nicht anders erreichbar ist. Auf diese Weise erhalten Sie eine übersichtliche Anzeige.

Testen Sie die Verbindung über die Schaltfläche **TESTEN**, bevor Sie sie endgültig speichern. Der Server wird unterhalb der Servergruppe angezeigt, die beim Ausführen des Befehls markiert gewesen ist. Sie können die Zuordnung zu einer Gruppe jederzeit über den Befehl **TASKS/VERSCHIEBEN NACH...** im Kontextmenü ändern. Sie können auch Servergruppen ineinander schachteln. Dies könnte zum Schluss ungefähr ein Bild wie das nachfolgende ergeben.

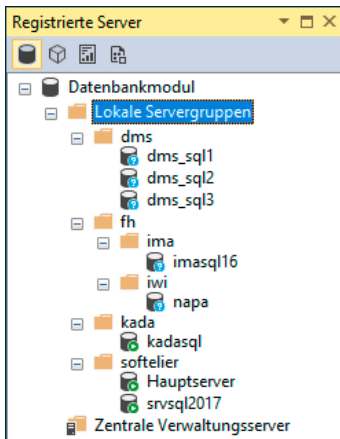


Bild 2.12 Registrierte Server in mehreren Servergruppen

Um auf die Objekte eines registrierten Servers zuzugreifen, verwenden Sie aus dem Kontextmenü einen der beiden folgenden Befehle:

- **Neue Abfrage:** Ein neues Abfrageeditor-Fenster wird geöffnet. Für die Anmeldung werden die bei der Registrierung des Servers verwendeten Informationen herangezogen.
- **Objekt-Explorer:** Im Objekt-Explorer wird der Server mit seinen Objekten dargestellt.



PRAXISTIPP: Sie können diese Einstellungen auf einen anderen Rechner übernehmen, indem Sie registrierte Server oder Gruppen exportieren. Dies geschieht über den Befehl **TASKS/EXPORTIEREN...** Der Export erfolgt im XML-Format in eine Datei, die die Erweiterung *regsrvr* trägt. Standardmäßig ist beim Export die Option *Benutzernamen und Kennwörter nicht in die Exportdatei einschließen* aktiviert. Deaktivieren Sie diese, werden die Kennwörter, wenn Sie eine Registrierung mittels SQL Server-Authentifizierung vorgenommen haben, in verschlüsselter Form in der Datei abgelegt. Die Kennwörter sind dann zwar nicht lesbar, können aber nach einem Import der Datei missbräuchlich verwendet werden. Geben Sie also auf diese Datei in diesem Fall besonders acht.

Der *Zentrale Verwaltungsserver* speichert mehrere Serververbindungen und kann zum Beispiel dazu verwendet werden, Abfragen auf mehreren Servern auszuführen. Die ist vor allem für Verwaltungsaufgaben, die auf mehreren Servern synchron ablaufen sollen, von Vorteil. Dazu müssen die betroffenen Server hier registriert werden.

Sie können natürlich nicht nur Datenbankmodule registrieren und damit schnell zugreifbar machen, sondern auch die drei weiteren Servertypen Analysis Services, Reporting Services und Integration Services wählen. Klicken Sie dazu jeweils eines der vier Symbole unter der Titelleiste des Fensters an, um die Anzeige entsprechend zu ändern.

Der Objekt-Explorer

Der Objekt-Explorer ist neben dem Abfrageeditor der Bereich, in dem Sie mit Sicherheit am häufigsten arbeiten werden.

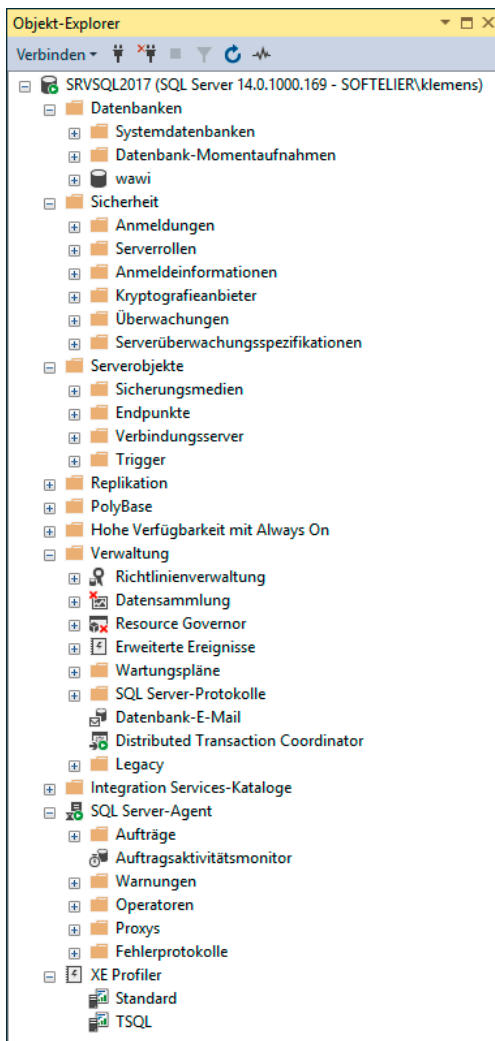


Bild 2.13 Objekt-Explorer

Von hier aus können Sie auf alle Objekte des gewählten Servers zugreifen. Um einen Server im Objekt-Explorer anzuzeigen, stehen unter anderem die folgenden Möglichkeiten zur Verfügung:

- Der Server, an dem Sie sich beim Öffnen des Management Studios angemeldet haben, wird immer automatisch im Objekt-Explorer angezeigt.
- Klicken Sie in der Symbolleiste des Objekt-Explorers auf **VERBINDEN/DATENBANKMODUL...**, um sich über einen neuen Anmeldedialog an dem gewünschten Server anzumelden. Es können so mehrere Server untereinander im Objekt-Explorer angezeigt werden. Auch über die Symbole *Objekt-Explorer verbinden* und *Trennen* können Sie jederzeit mit unterschiedlichen Servern arbeiten.

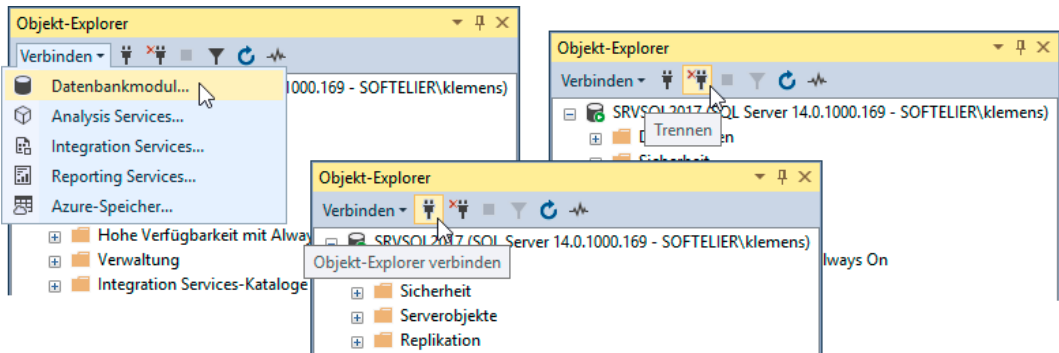


Bild 2.14 Datenbanken im Objekt-Explorer verbinden und trennen

- Über das Kontextmenü im Fenster *Registrierte Server* über den Befehl **OBJEKT-EXPLORER**.
- Im Menü **DATEI** über den Befehl **OBJEKT-EXPLORER VERBINDEN**.

Im Objekt-Explorer haben Sie vor allem folgende wichtige Objekte im Zugriff:

- **Datenbanken:** Neben den Benutzerdatenbanken werden die *Systemdatenbanken* sowie *Datenbank-Momentaufnahmen* (Snapshots) in eigenen Ordnern angezeigt. Datenbank-Snapshots speichern Momentauszüge aus der Datenbank, die nur gelesen und beispielsweise für Auswertungen verwendet werden können. Über die einzelnen Datenbanken gelangen Sie in weiterer Folge zu allen Datenbankobjekten. Lesen Sie hierzu mehr in den folgenden Kapiteln.
- **Sicherheit:** Hier werden Objekte wie Anmeldungen und Serverrollen verwaltet, die für die Organisation der Sicherheit und der Zugriffsberechtigungen auf dem Datenbankserver benötigt werden (lesen Sie dazu Kapitel 10). Außerdem finden Sie an dieser Stelle die Features *Überwachungen* und die *Kryptografie* vor (nicht Express Edition).
- **Serverobjekte:** Hier finden Sie *Sicherungsmedien* (Genaueres dazu in Kapitel 9), *Endpunkte* (nicht Express Edition) und *Verbindungsserver* (Linked Server). Letztere benötigen Sie für die Verbindung mehrerer Server miteinander. Sind Server auf diese Art miteinander verlinkt, können Sie in einer Abfrage auf mehrere Datenbankserver gleichzeitig zugreifen (Informationen dazu in Kapitel 10). Außerdem finden Sie in der Rubrik *Serverobjekte* noch *Trigger*, womit hier im Speziellen die DDL-Trigger gemeint sind. Mehr dazu lesen Sie in Kapitel 6.

- **Replikation:** Hier können Sie Objekte erstellen und verwalten, die Sie für verteilte Datenbanken benötigen. Die Bedeutung der Replikation hat allerdings in den letzten Jahren stark abgenommen. Durch die Verfügbarkeit von schnellen Datenleitungen, sowohl in terrestrischer als auch mobiler Form, wird nämlich häufig der einfachere Onlinezugriff auf eine zentrale Datenbank dem Verteilen und Abgleichen der Datenbestände mit all seinen Synchronisationsproblemen vorgezogen. Die Express Edition kann hierbei nur der Abonnent (Empfänger), nicht der Verleger sein.
- **Hohe Verfügbarkeit mit AlwaysOn:** Bei AlwaysOn handelt es sich um ein Feature, hinter dem sich eine bedeutende Weiterentwicklung für Hochverfügbarkeitsszenarien verbirgt. Daher steht dieses Feature in vollem Umfang auch nur bei der Enterprise Edition zur Verfügung, in eingeschränktem Maße auch bei der Standard Edition.
- **Verwaltung:** In dieser Rubrik sind verschiedene Punkte zusammengefasst:
 - **Richtlinienverwaltung:** Sie können zum Beispiel Richtlinien für die Benennung von Objekten erstellen. Diese werden dann bei der Erstellung von Datenbankobjekten erzwungen.
 - **Datensammlung:** Dieses Feature – früher als Datenauflistung bezeichnet – erlaubt es, Konfigurationen für ein Verwaltungs-Data-Warehouse vorzunehmen (nicht Express Edition).
 - **Ressourcenkontrolle (Resource Governor):** Um zu verhindern, dass einzelne Prozesse den ganzen Server blockieren, gibt es dieses Feature. Hierbei können die beanspruchten Ressourcen für einzelne Prozesse limitiert und optimiert werden. Auch dieses Feature steht bei der Express Edition nicht zur Verfügung.
 - **Erweiterte Ereignisse:** Hier bekommen Sie Informationen darüber, ob Ihr System in Ordnung ist, vor allem in Zusammenhang mit AlwaysOn.
 - **Wartungspläne:** Wartungspläne führen Wartungsarbeiten wie zum Beispiel Sicherungen, Index-Reorganisationen oder Prüfungen der Datenbankintegrität gemeinsam oder getrennt zu festgelegten Zeiten aus. Wartungspläne gibt es bei der Express Edition nicht, da sie den hier nicht enthaltenen Server-Agent benötigen.
 - **SQL Server-Protokolle:** Hier werden unterschiedliche Vorgänge protokolliert – wie zum Beispiel Anmeldungen oder das Anfügen und das Sichern von Datenbanken. Vor allem Fehler unterschiedlichster Art werden hier festgehalten.
 - **Datenbank-E-Mail:** Über den SQL Server können SMTP-Mails versendet werden. Diese lassen sich zum Beispiel für Benachrichtigungen aus Datenbanken heraus verwenden oder um Administratoren in bestimmten Situationen zu informieren. In der Express Edition müssen Sie leider auf dieses Feature verzichten. Wir behandeln dieses Thema in Kapitel 11.
 - **Distributed Transaction Coordinator:** Dieser Dienst wird benötigt, wenn Sie Transaktionen verwenden, bei denen mehr als ein Server beteiligt ist.
 - **Legacy:** Hier finden Sie Komponenten, die zur Gewährleistung der Abwärtskompatibilität benötigt werden. Hierzu gehören beim SQL Server 2017 nur mehr die alten Datenbankwartungspläne.



HINWEIS: Falls Sie den Aktivitätsmonitor an dieser Stelle vermissen, der ist schon mit der Version 2008 leider nicht mehr hier zu finden. Um ihn zu öffnen, verwenden Sie das Kontextmenü direkt auf dem Server-Ordner. Über den Aktivitätsmonitor kann man ablesen, welche Benutzer gerade in einer Session mit einer Datenbank verbunden sind. Dazu existieren verschiedene Statusinformationen.

- **Integration Services-Kataloge:** Dahinter verbirgt sich die Möglichkeit, Anwendungen für Integration Services auf den Server zu bringen und von dort auszuführen oder ausführen zu lassen. Diese Variante gefällt mir besonders, da sie wesentlich komfortabler als die früheren Möglichkeiten ausfällt. Sie lesen über diese Variante in Kapitel 11. Integration Services und damit auch dieses Feature sind in der Express Edition nicht enthalten.
- **SQL Server-Agent:** An dieser Stelle werden die Aufträge für den SQL Server-Agent erstellt und verwaltet. Dieser Dienst, der bei der Express Edition leider nicht verfügbar ist, ist für das Abarbeiten zeitgesteuerter Aufgaben zuständig. So ist er zum Beispiel dafür zuständig, dass das allnächtliche Backup durchgeführt wird. Beispiele zur Erstellung von Agent-Aufträgen finden Sie in den Kapiteln 9 und 11.
- **XEvent Profiler:** Dies ist ein neues Feature, das mit der Version 17.3 des Management Studios erstmals – vorerst unter dem Namen XE Profiler – verfügbar ist. Es nutzt die erweiterten Ereignisse (Extended Events/XEvents), um in vereinfachter Form Funktionalitäten des SQL Server Profilers bereitzustellen. Sie ermöglichen es, live zu beobachten, welche Anweisungen auf der aktuellen Serverinstanz gerade ausgeführt werden. Die Möglichkeiten dabei sind nicht so umfassend wie beim SQL Server Profiler, aber das „Mitschauen“ beansprucht hier auch weniger Ressourcen.



HINWEIS: Im Objekt-Explorer werden nur die Features angezeigt, die in der jeweiligen Edition des verbundenen Servers verfügbar sind. Bild 2.13 zeigt die volle Ausprägung einer Enterprise, Evaluation oder Developer Edition. Im Vergleich dazu zeigt Bild 2.15 den Objekt-Explorer mit einer Express Edition verbunden. Das Management Studio selbst unterscheidet nicht mehr zwischen der Voll- und Express-Version des Werkzeugs. Es gibt nur mehr eine einheitliche Version und der Funktionsumfang ist nur vom verbundenen Server abhängig und kann mit allen Editionen verwendet werden.

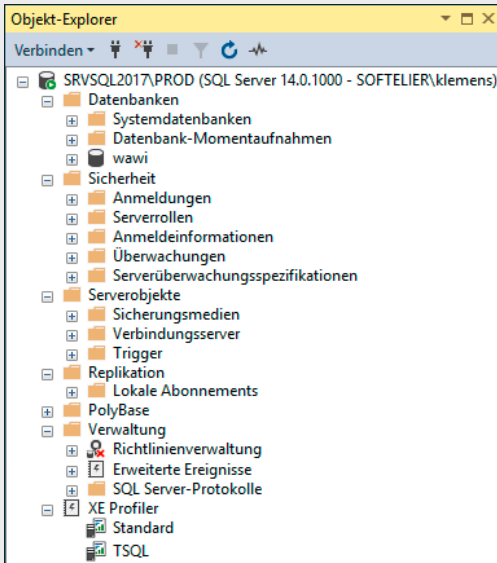


Bild 2.15 Objekt-Explorer mit einer Express Edition verbunden

Der Abfrageeditor

Eine Kernkomponente des Management Studios ist der Abfrageeditor. Dieser dient zum Erstellen und Absetzen von SQL- und Transact-SQL-Anweisungen. Er enthält auch einen grafischen Editor, der beim Schreiben von Statements hilfreich ist.

Um ein Abfrageeditor-Fenster zu öffnen, wählen Sie eine der folgenden Varianten:

- Markieren Sie eine Datenbank im Objekt-Explorer und wählen Sie im Kontextmenü den Befehl **NEUE ABFRAGE**. Die bereits bestehenden Anmeldeinformationen des im Objekt-Explorer markierten Servers werden dann für die neue Verbindung übernommen. Auch die markierte Datenbank wird im Abfrageeditor-Fenster als Vorgabe übernommen.
- Wählen Sie den Befehl **DATEI/NEU/ABFRAGE MIT AKTUELLER VERBINDUNG**. Die bereits bestehende Anmeldung des im Objekt-Explorer markierten Servers wird dann ebenso übernommen. Es muss auch gar nicht das Server-Objekt selber markiert sein, es kann ein beliebiges Objekt im Baum darunter ausgewählt sein.
- Wählen Sie im Menü **DATEI** den Befehl **NEU/DATENBANKMODUL-ABFRAGE**. Sie werden daraufhin zur Eingabe neuer Anmeldeinformationen aufgefordert.

In Bild 2.16 sehen Sie die entsprechenden Symbole markiert, die Sie alternativ zu den zwei Menübefehlen des Menüs **DATEI** verwenden können. Das linke, breitere Symbol der beiden führt den Befehl **NEUE ABFRAGE** aus und übernimmt damit Anmeldung und Datenbank von der aktuellen Markierung im Objekt-Explorer. Mit dem zweiten Symbol wird eine neue Datenbankmodul-Abfrage mit der Eingabe neuer Anmeldeinformationen erstellt.



Bild 2.16 Symbole zum Öffnen eines neuen Abfrageeditor-Fensters

Im neuen Abfrageeditor-Fenster können Sie nun SQL- und Transact-SQL-Anweisungen eingeben und ausführen. Anweisungen werden hierbei über das Ausführen-Symbol mit dem Ausrufezeichen oder über die Taste (F5) ausgeführt. Das Ergebnis wird im unteren Bildschirmbereich angezeigt. Bild 2.17 zeigt zwei einfache SQL-Anweisungen, die im Abfrageeditor gemeinsam ausgeführt worden sind, mit ihren Ergebnissen.

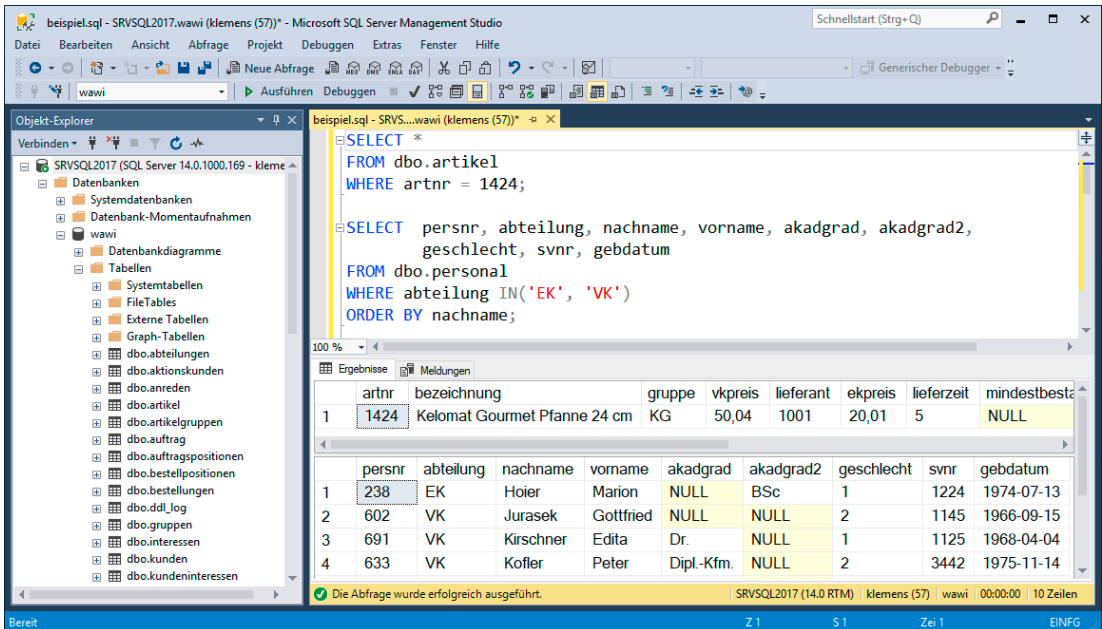


Bild 2.17 Abfrageeditor-Fenster mit SQL-Anweisungen

Die Symbole der Symbolleiste des Abfrageeditors haben von links beginnend folgende Bedeutung. Aufgrund des neuen Designs der Symbole habe ich in Bild 2.18 die der Vorversion zum Vergleich mit dargestellt. Die obere Symbolleiste in der Abbildung stammt aus der Version 16 des Management Studios, die untere Symbolleiste aus der aktuellen Version 17.

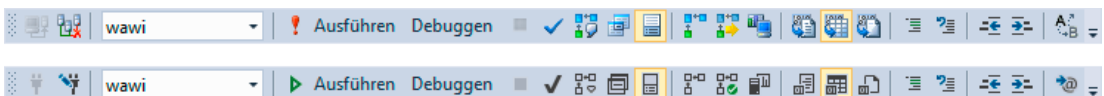


Bild 2.18 Symbolleiste SQL-Editor

- **Verbinden:** Aufbauen einer neuen Verbindung mit einer Datenbank-Engine. Dieses Symbol ist deaktiviert, wenn eine aktive Verbindung besteht. Ist die Verbindung unterbrochen worden, kann sie über dieses Symbol mit denselben Anmeldeinformationen wiederhergestellt werden.

- *Verbindung ändern*: Anmelden mit einer neuen Verbindung.
- *Verfügbare Datenbanken*: Hier wählen Sie die Datenbank aus, mit der Sie aktuell verbunden sind.
- *Ausführen*: Führt die Anweisungen aus. Ist ein Bereich markiert, wird nur der markierte Bereich ausgeführt. Sie können alternativ die Schaltfläche (**F5**) verwenden.



ACHTUNG! Das Symbol für *Ausführen* hat sich in der aktuellen Version geändert. Das bisherige rote Ausrufezeichen ist durch den grünen Rechtspfeil ersetzt worden, der in der vorletzten Version, dem Management Studio des SQL Server 2014 noch für das Debuggen in Verwendung gewesen ist. Haben Sie auch diese Version schon verwendet, müssen Sie sich hier umstellen.

- *Debuggen*: Mit diesem Werkzeug können Sie Transact-SQL-Code debuggen. Dies ist eine Methode, um logische Fehler im Programmcode leichter zu finden. Details zum Debuggen finden Sie in Kapitel 6.
- *Ausführung der Abfrage abbrechen*: Bricht die Ausführung einer Anweisung ab, wenn diese zum Beispiel zu lange dauert. Das graue Rechteck, das dieses Symbol ausmacht, wird rot, während eine Abfrage ausgeführt wird, und ist auch nur dann aktiv.
- *Analysieren*: Prüft die Syntax der Anweisung(en).
- *Geschätzten Ausführungsplan anzeigen*: Zeigt den Ausführungsplan der Anweisung an, aus dem unter anderem die Indexverwendung hervorgeht.
- *Abfrageoptionen*: Hier können Parameter eingestellt werden, die beim Ausführen einer Anweisung mit an den Server gesendet werden. So kann zum Beispiel unter den allgemeinen Einstellungen die Anzahl der vom Server zurückgelieferten Datensätze beschränkt und ein Timeout für das Abarbeiten der Abfrage festgelegt werden. Hier wird auch das Batchtrennzeichen mit **G0** festgelegt, dessen Bedeutung Sie noch in den weiteren Kapiteln kennenlernen werden. Aber auch die Art der Ergebnisausgabe kann weitergehend beeinflusst werden.

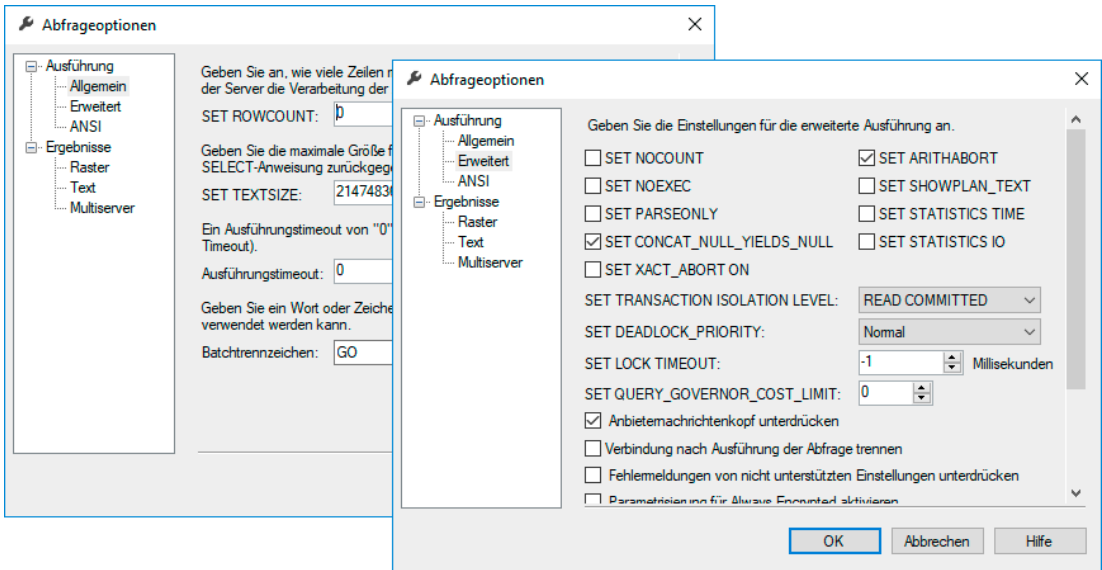


Bild 2.19 Abfrageoptionen festlegen

- IntelliSense aktiviert:** IntelliSense ist ein Feature, das das Schreiben von SQL-Anweisungen deutlich vereinfacht. Vor allem Programmierer, die dieses Feature von ihrer Entwicklungsumgebung her kennen, werden sich freuen, dies auch in diesem Kontext verwenden zu können. Das IntelliSense-Fenster klappt bei der Eingabe automatisch auf und kann so zur schnellen Auswahl von Spalten- und Tabellennamen verwendet werden. Eine Einschränkung dabei ist allerdings, dass zum Beispiel bei einer SELECT-Anweisung die Spalten erst dann über IntelliSense ausgewählt werden können, wenn zuvor die FROM-Klausel mit dem Tabellennamen eingetippt wurde. Sobald der gewünschte Wert in der Liste markiert ist, kann er mit der **(TAB)**-Taste übernommen werden. Sind noch mehrere Werte in der Liste enthalten, kann ein Wert zuvor über **(PFEIL-NACH-UNTEN)** markiert werden. Dies ist in der Regel viel effizienter, als zur Maus zu greifen. Dies funktioniert zwar auch, ist aber äußerst umständlich.

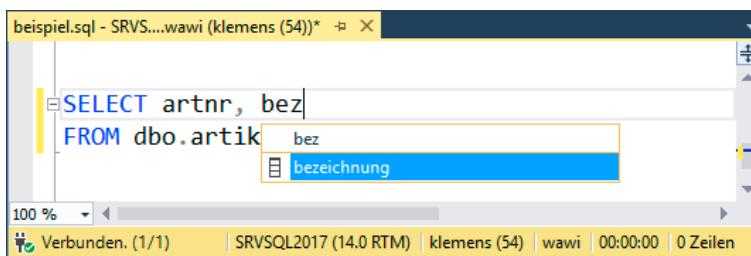


Bild 2.20 IntelliSense in Aktion

- Tatsächlichen Ausführungsplan einschließen:** Ist diese Option aktiviert, wird beim Ausführen einer Anweisung immer der verwendete Ausführungsplan in einem separaten Register des Ergebnisses mit ausgegeben.

- *Live-Abfragestatistik einschließen:* Während der Ausführungsplan einen Plan anzeigt, zeigt die Live-Abfragestatistik die tatsächlich erreichten Werte der ausgeführten Anweisung an. Sie wird, wenn sie aktiviert ist, ebenso in einem separaten Register des Ergebnisses mit ausgegeben.
- *Clientstatistiken einschließen:* Auch hierbei handelt es sich um eine zusätzliche Option zur Ausgabe. Es werden verschiedene statistische Werte aus Clientsicht, wie zum Beispiel die Anzahl der Anweisungen und Ausführungszeiten, angezeigt. Werden dieselben Anweisungen wiederholt ausgeführt, werden auch die Veränderungen und Mittelwerte angezeigt.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane contains the following SQL query:

```

SELECT *
FROM dbo.artikel
WHERE artnr = 1424;

SELECT persnr, abteilung, nachname, vorname, akadgrad, akadgrad2,
       geschlecht, svnr, gebdatum
FROM dbo.personal
WHERE abteilung IN('EK', 'VK')
ORDER BY nachname;

```

The bottom pane displays the Client Statistics for the query. The table below represents the data shown in the screenshot:

	Versuch 4	Versuch 3	Versuch 2	Versuch 1	Mittelwert
Clientausführungszeit	15:20:10	15:19:37	15:19:12	15:18:50	
Abfrageprofilstatistik					
Anzahl von INSERT-, DELETE- und UPDATE-An...	0	→ 0	→ 0	→ 0	→ 0.0000
Von INSERT-, DELETE- oder UPDATE-Anweisu...	0	→ 0	→ 0	→ 0	→ 0.0000
Anzahl von SELECT-Anweisungen	8	→ 8	→ 8	↑ 2	→ 6.5000
Von SELECT-Anweisungen zurückgegebene Zei...	22	→ 22	→ 22	↑ 10	→ 19.0000
Anzahl von Transaktionen	0	→ 0	→ 0	→ 0	→ 0.0000
Netzwerkstatistik					
Anzahl von Serverroundtrips	5	→ 5	→ 5	↑ 1	→ 4.0000
Vom Client gesendete TDS-Pakete	5	→ 5	→ 5	↑ 1	→ 4.0000
Vom Server empfangene TDS-Pakete	18	→ 18	→ 18	↑ 1	→ 13.7500

At the bottom of the screenshot, a status bar indicates: "Die Abfrage wurde erfolgreich ausgeführt." and "SRVSQL2017 (14.0 RTM) | klemens (54) | wawi | 00:00:01 | 17 Zeilen".

Bild 2.21 Clientstatistiken mit eingeschlossen

- *Ausgabe in Text:* Das Ergebnis wird in einer einfachen Textform ausgegeben, wie in Bild 2.22 zu sehen.
- *Ausgabe in Raster:* Die Ausgabe des Ergebnisses erfolgt in Rasterform, so wie zuletzt in Bild 2.17 zu sehen. Dies ist die Standardausgabeeinstellung.
- *Ausgabe in Datei:* Das Ergebnis wird direkt in eine ASCII-Datei geschrieben und nicht am Bildschirm dargestellt.
- *Kommentiert die ausgewählten Zeilen aus:* Mit diesem Symbol werden alle markierten Zeilen auskommentiert, indem am Beginn jeder Zeile ein doppelter Bindestrich eingefügt wird. Mit dem nachfolgenden Symbol kann der Vorgang rückgängig gemacht und eine entsprechende Auskommentierung aufgehoben werden.

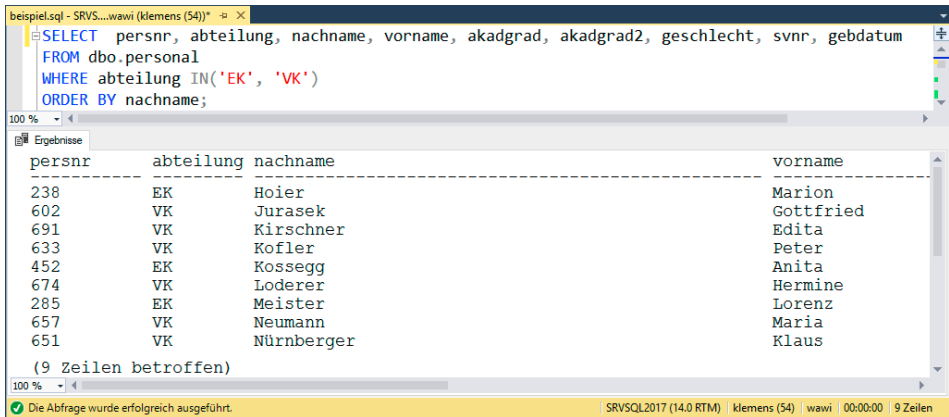


Bild 2.22 Ausgabe in Text

- *Einzug verkleinern, Einzug vergrößern:* Damit können auch mehrere Zeilen gemeinsam weiter eingerückt werden oder der Einzug verkleinert werden. Ich persönlich bevorzuge dazu allerdings die Tastenkürzel (**TAB**) sowie (**SHIFT**) + (**TAB**). Denn ich empfinde den Einsatz von Tastenkürzeln beim Tippen von Anweisungen als schneller und komfortabler als den Griff zur Maus.
- *Werte für Vorlagenwerte angeben:* Über den Vorlagenbrowser stehen viele Anweisungsvorlagen für die unterschiedlichsten Aufgaben bereit. Die in diesen Vorlagen enthaltenen Parameter lassen sich über den mit diesem Symbol öffnenden Dialog komfortabel ersetzen. In Bild 2.23 sehen Sie, wie in die Vorlagen zum Anfügen einer Datenbank die Parameter eingetragen werden, um unsere Beispieldatenbank auf dem Server anzufügen. Es werden automatisch die Parameter aus dem aktiven Abfrageeditor-Fenster in den Dialog übernommen.

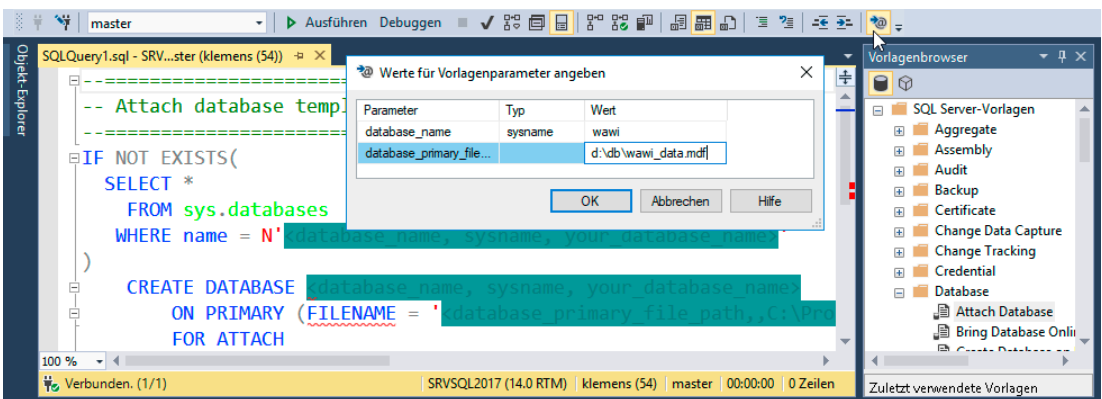


Bild 2.23 Werte für Vorlagenparameter angeben



PRAXISTIPP: Sie können – wie in jedem anderen Programm auch – die Symbolleisten anpassen und ergänzen. Somit können Sie beispielsweise aus einer Vorgängerversion bekannte und nun standardmäßig nicht mehr angezeigte Symbole wie *Trennen* und *Abfrage in Editor entwerfen* wieder aus der Versenkung holen.

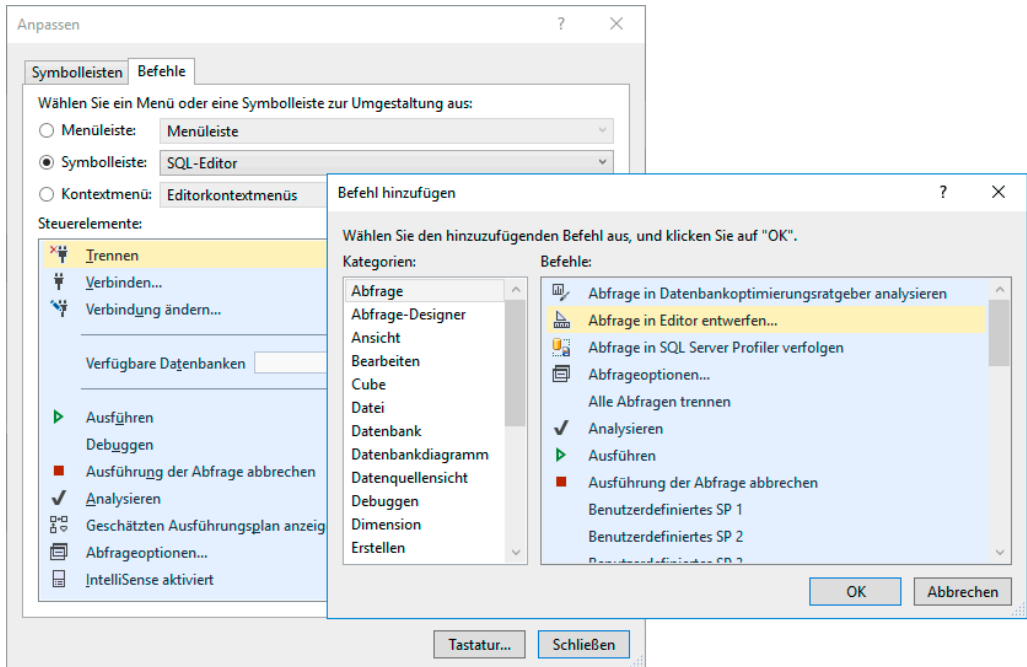


Bild 2.24 Weitere Symbole ergänzen



HINWEIS: Mehr über das Schreiben von SQL-Anweisungen lesen Sie in Kapitel 4.

Dieser kurze Überblick soll Ihnen das Arbeiten mit dem Management Studio erleichtern. Viele weitere Details zu den verschiedenen Verwendungsmöglichkeiten dieses Tools werden bei den jeweiligen Themen in den folgenden Kapiteln behandelt.

■ 2.3 Das Kommandozeilentool: SQLCMD

Ab und zu ist es hilfreich, auch von der Kommandozeile aus auf den SQL Server zuzugreifen, zum Beispiel, wenn SQL-Skripte als Bestandteile anderer Verwaltungsaufgaben – zum Beispiel auf Betriebssystemebene – ausgeführt werden sollen. Auch wenn es kein grafisches Tool im eigentlichen Sinne ist, möchte ich es an dieser Stelle dennoch erwähnen.



PRAXISTIPP: Wenn Sie die Verwaltungstools auf dem Server nicht installieren, haben Sie dieses Werkzeug dennoch lokal zur Verfügung und es kann in manchen Situationen zum willkommenen Rettungsanker werden.

Wie angedeutet, ist in der Praxis die Verwendung von SQLCMD in Kombination mit einem SQL-Skript, dessen Pfad beim Start mit angegeben wird, weit verbreitet. Dieses Skript wird dann automatisch ausgeführt. Gestartet wird das Tool von der Kommandozeile aus mit dem Befehl SQLCMD.

Die wichtigsten Parameter beim Start zeigt die nachfolgende Übersicht.

Tabelle 2.1 Aufrufparameter von SQLCMD

Parameter	Bedeutung
-S server	Name oder IP-Adresse des Servers. Wird dieser Parameter weggelassen, wird die lokale Standardinstanz verwendet.
-U anmeldeame	Anmeldeame bei einer Anmeldung mit SQL Server-Authentifizierung
-P kennwort	Kennwort bei einer Anmeldung mit SQL Server-Authentifizierung
-E	Anmeldung mittels Windows-Authentifizierung
-d datenbankname	Name der Datenbank, zu der nach der Anmeldung gewechselt werden soll
-i eingabedatei	Pfad zur SQL-Skript-Datei, die nach dem Anmelden ausgeführt werden soll. Wird diese Option verwendet, wird SQLCMD nach dem Ausführen des Skripts wieder beendet.
-o ausgabedatei	Name einer Datei, in die die Ausgaben geschrieben werden. Dies ist mit der Option AUSGABE IN DATEI im Management Studio vergleichbar.


```

C:\>sqlcmd /?
Microsoft (R) SQL Server-Befehlszeilentool
Version 14.0.1000.169 NT
Copyright (C) 2017 Microsoft Corporation. Alle Rechte vorbehalten.

Syntax: Sqlcmd          [-U Anmelde-ID]          [-P Kennwort]
[-S Server]            [-H Hostname]            [-E vertrauenswürdige Verbindung]
[-N Verbindung verschlüsseln] [-C Serverzertifikat vertrauen]
[-d Datenbanknamen verwenden] [-l Anmeldezeitlimit] [-t Abfragetimeout]
[-h Header]            [-s Spaltentrennzeichen]          [-w Bildschirmbreite]
[-a Paketgröße]        [-e Eingabe auf dem Bildschirm zeigen]      [-I Bezeichner in Anführungszeichen aktivieren]
[-c Befehlsende]       [-l[c] Server auflisten [einfache Ausgabe]]
[-q 'Befehlszeilenabfrage'] [-Q 'Befehlszeilenabfrage' und beenden]
[-m Fehlerebene]       [-V Schweregrad]          [-W nachfolgende Leerzeichen entfernen]
[-u Unicode-Ausgabe]   [-r[0|1] Meldungen nach STDERR]
[-i Eingabedatei]     [-o Ausgabedatei]        [-z neues Kennwort]
[-f <codepage> | i:<codepage>[,o:<codepage>]] [-Z neues Kennwort und beenden]
[-k[1|2] Steuerzeichen entfernen [ersetzen]]
[-y variable Anzeigebreite]
[-Y feste Anzeigebreite]
[-p[1] Statistik ausgeben [Spaltenformat]]
[-R Clienteneinstellung für Land/Region verwenden]
[-K Anwendungsabsicht]
[-M Multisubnetz-Failover]
[-b Batchabbruch bei Fehler]
[-v var = "Wert"...] [-A dedizierte Administratorverbindung]
[-X[1] Befehle, Startskript, Umgebungsvariablen deaktivieren [und beenden]]
[-x Variablenersetzung deaktivieren]
[-j Unformatierte Fehlermeldungen drucken]
[-g Spaltenverschlüsselung aktivieren]
[-G Azure Active Directory für Authentifizierung verwenden]
[-? Syntaxzusammenfassung anzeigen]

```

Bild 2.25 Die Aufrufparameter von SQLCMD

Der Aufruf des Tools könnte beispielsweise so aussehen:

```
sqlcmd -U klemens -P konosql -S srvsq12017
```

Um Windows-Authentifizierung zu verwenden, ersetzen Sie die Parameter `-U` und `-P` durch den Parameter `-E`.



ACHTUNG! Sie müssen auf die Groß- und Kleinschreibung bei den Aufrufparametern achten, da es mehrere Buchstaben gibt, die als Groß- bzw. Kleinbuchstabe einem anderen Parameter zugewiesen sind.

Nach der Anmeldung wird der Command-Prompt mit `1>` dargestellt. Geben Sie nun beliebige SQL-Anweisungen ein. Bei der Eingabe werden die Zeilen mit dem Command-Prompt fortlaufend nummeriert. Damit die Anweisung ausgeführt wird, wechseln Sie in eine neue Zeile und geben `GO` ein. Anschließend wird die Anweisung mit **ENTER** ausgeführt.

```

C:\>sqlcmd -S srvsq12017 -E
1> use wawi
2> go
Der Datenbankkontext wurde in "wawi" geändert.
1> SELECT *
2> FROM dbo.abteilungen;
3> go
abtnr  bezeichnung
-----
CO     Controlling
EK     Einkauf
FB     Finanzbuchhaltung
FE     Forschung & Entwicklung
GL     Geschäftsleitung
LA     Lager
MA     Marketing
VK     Verkauf

(8 Zeilen betroffen)
1> exit
C:\>

```

Bild 2.26 Verwenden von SQLCMD

Beenden Sie das Tool wahlweise mit der Anweisung `quit` oder `exit`.



HINWEIS: Ein Beispiel für den Aufruf einer Skript-Datei mit SQLCMD finden Sie in Kapitel 9.

Das Kommandozeilentool kann unter Umständen der Retter in der Not sein. Sie können mit diesem Tool all das realisieren, was Sie im Abfrageeditor des Management Studios erledigen können – vorausgesetzt, Sie kennen die entsprechenden Anweisungen.

■ 2.4 Der Konfigurations-Manager

Der Konfigurations-Manager ist ein Snap-in der Microsoft Management-Konsole. Als solches kann er entweder direkt über das Windows-Startmenü oder über das Programm MMC aufgerufen werden.

Der Konfigurations-Manager hat drei Aufgaben:

- *Verwaltung der SQL Server-Dienste:* Die einzelnen Dienste können gestartet und beendet, ihre Startart und ihr Dienstkonto festgelegt werden. Am Symbol ist zu erkennen, ob ein Dienst gestartet ist oder nicht. Bei gestarteten Diensten ist in einem weißen Kreis ein grünes Dreieck abgebildet. Ist ein Dienst nicht gestartet, ist an dieser Stelle ein rotes Viereck zu sehen.

- *Netzwerkconfiguration für den Server:* Hier können Sie Netzwerkprotokolle für den Server aktivieren und deaktivieren. In der Regel verwenden Sie dies nach dem Setup, um TCP/IP für den Remotezugriff zu aktivieren. Für dieses Protokoll lassen sich hier weitere Einstellungen vornehmen; beispielsweise zum Port, der standardmäßig mit 1433 belegt ist.
- *Protokolle für den Client:* Für diesen Rechner als Client können Sie hier die verwendbaren Protokolle einstellen. Sind mehrere aktiv, wird eine Reihenfolge festgelegt, nach der der Client versucht, mit dem jeweiligen Protokoll eine Verbindung zu einem Server aufzubauen.

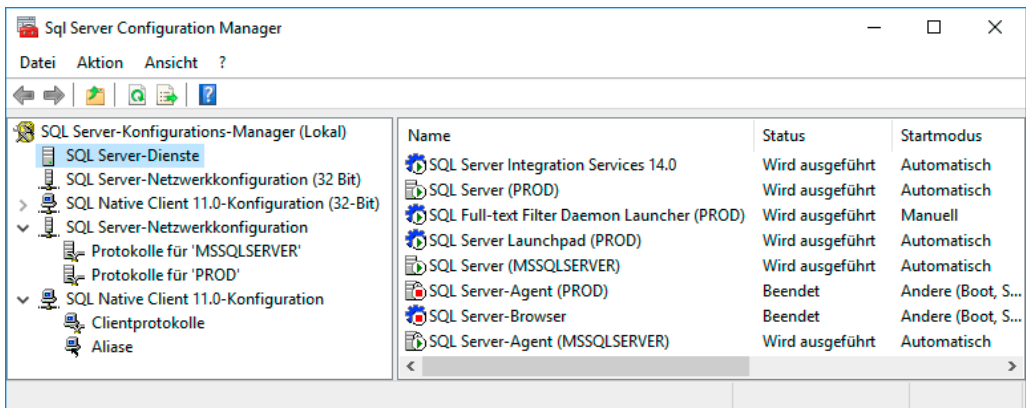


Bild 2.27 SQL Server-Konfigurations-Manager



ACHTUNG: Der SQL Server-Konfigurations-Manager unterscheidet zwischen Einstellungen, die auf dem Server einerseits und Einstellungen, die auf Clients andererseits relevant sind. Dementsprechend müssen diese Einstellungen auch auf dem Server oder auf allen betroffenen Clients vorgenommen werden. Da der Konfigurations-Manager nur lokal und nicht remote verwendet werden kann, muss er auch direkt am Server gestartet werden, um serverseitige Einstellungen vorzunehmen.

Lokal können Sie nach der Installation von Client-Tools die Konfiguration der Client-Protokolle nutzen.

Zusätzlich können Sie für den Client Aliasse für den Zugriff auf den Server erstellen. Dies ist vor allem dann hilfreich, wenn im Netzwerk nur mit der Angabe der IP-Adresse eine Verbindung zum Server aufgebaut werden kann. Ein Alias kann dann im Anmeldedialog anstelle des echten Servernamens – ähnlich einem Hosts-Eintrag – verwendet werden.



PRAXISTIPP: Hilfreich kann der Eintrag auf den Clients auch sein, wenn ein anderer Port als der Standardport 1433 für den Server verwendet wird.

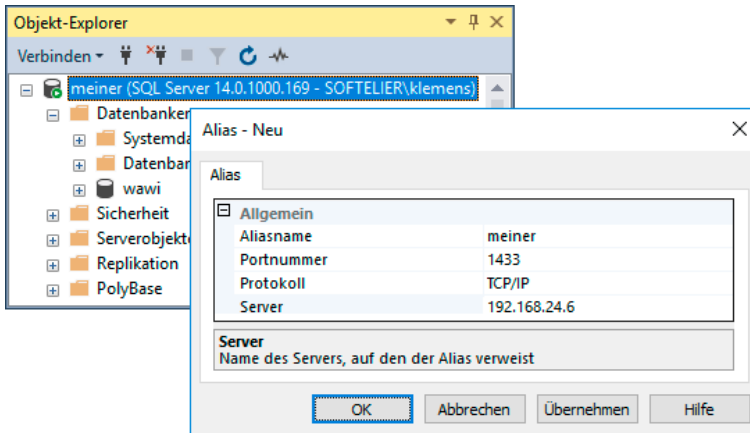


Bild 2.28 Client-Alias für Datenbankverbindung

In diesem Beispiel kann nun bei der Anmeldung als Servername *meiner* verwendet werden. Bereits in Kapitel 1 haben Sie gelesen, wie Sie einzelne Protokolle für eine Serverinstanz aktivieren und deaktivieren können. Hier können Sie auch den Port einstellen, wenn Sie mehrere Instanzen auf einem Server betreiben. Dynamische Ports, wie automatisch beim Setup für weitere Instanzen vergeben, können nämlich in Firewalls nicht konfiguriert werden und sind daher im Netzwerk wertlos. In Bild 2.29 sehen Sie, wir für die Instanz PROD, die als zweite installierte Instanz standardmäßig mit einem dynamischen Port versehen war, auf den Port 1522 umgestellt wird. Dazu werden die Nullen in den Spalten *Dynamische TCP-Ports* entfernt und der neue Port unter *TCP-Port* bei allen Adressen eingetragen.

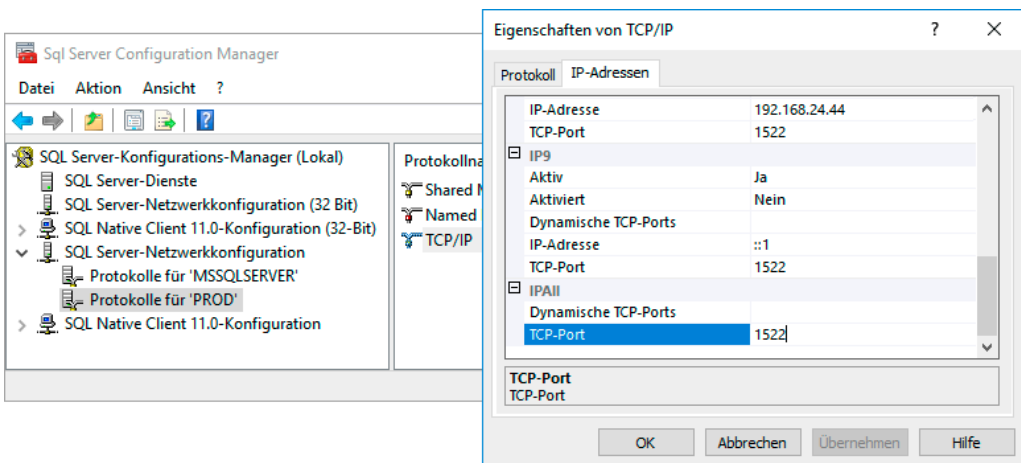


Bild 2.29 Port für TCP/IP festlegen

Je nachdem, wie viele Netzwerkadapter in Ihrem Server vorhanden sind, kann hier eine Vielzahl an IP-Adressen angezeigt werden. Zusätzlich wird noch zwischen IPv4 und IPv6 unterschieden. Bei Änderungen sollten daher immer alle Einträge synchron angepasst werden. Außerdem muss es zu einem Neustart des SQL Server-Dienstes kommen, damit Änderungen wirksam werden.

■ 2.5 Das SQL Server-Installationscenter

Das Installationscenter, das Sie schon bei der Installation Ihres SQL Servers kennengelernt haben, können Sie aus der Liste der Konfigurationstools später erneut starten, um wieder ins Setup zu gelangen und so beispielsweise einzelne Komponenten nachträglich zu installieren.

Dazu wählen Sie im Installationscenter die Option *Neue eigenständige SQL Server-Installation oder Hinzufügen von Funktionen zu einer vorhandenen Installation* und im Weiteren dann die Instanz aus, deren Features ergänzt werden sollen. Die bereits installierten Features werden bei der Funktionsauswahl ausgegraut angezeigt, weitere können analog zur Erstinstallation gewählt und damit zusätzlich installiert werden.

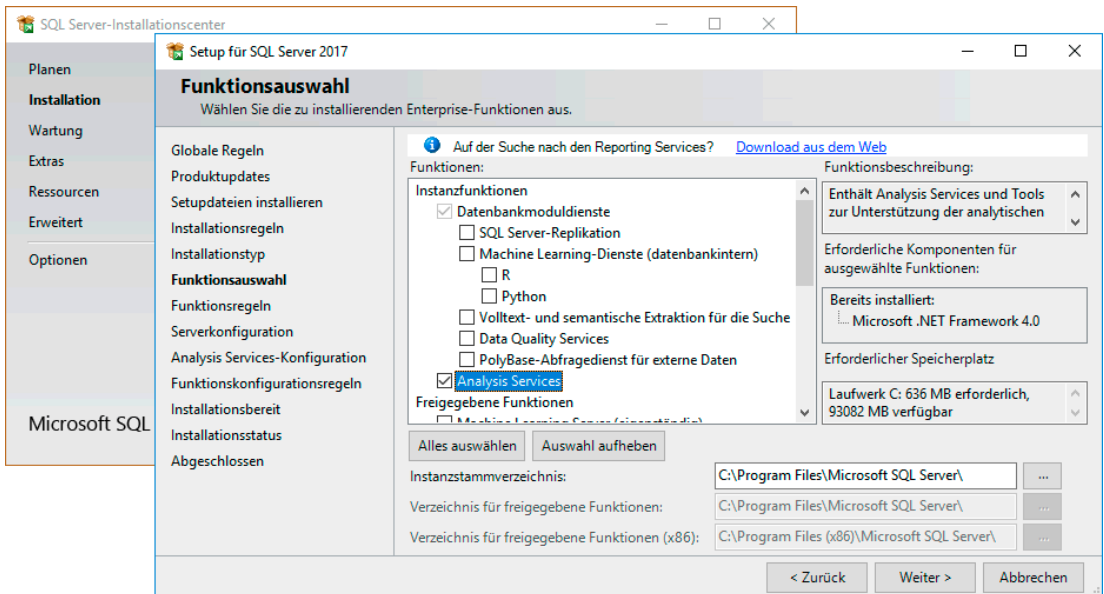


Bild 2.30 Das Installationscenter beim nachträglichen Installieren einzelner Komponenten



HINWEIS: Sind die Installationsmedien nicht mehr am ursprünglichen Platz zu finden, kann das Installationsmedien-Stammverzeichnis unter *Optionen* geändert werden.

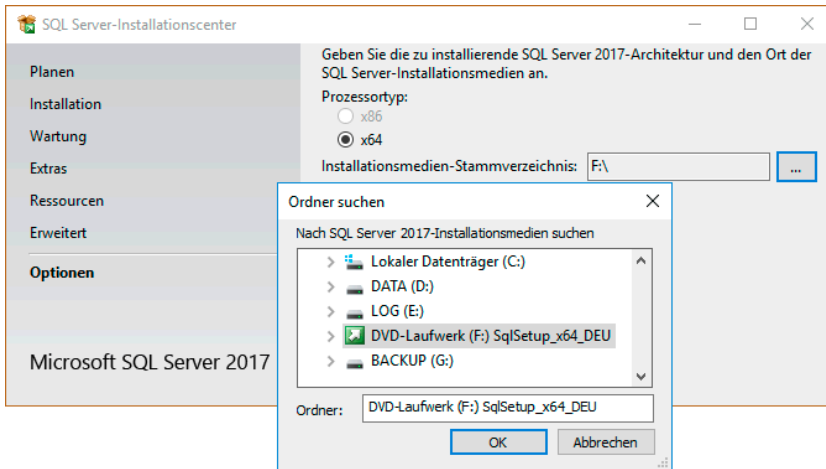


Bild 2.31 Installationsmedien (erneut) auswählen

■ 2.6 Der Profiler

Der Profiler gehört zu den *Leistungstools* und zeichnet Anweisungen auf, die auf dem Server ausgeführt werden. Diese können zur Analyse herangezogen werden und lassen sich nach unterschiedlichen Kriterien filtern. Das Ergebnis kann nicht nur am Bildschirm ausgegeben, sondern auch als sogenannte Trace-Datei gespeichert werden. Eine solche Trace-Datei kann wiederum vom Optimierungsratgeber als Grundlage verwendet werden. Da der Profiler alle Anweisungen der definierten Kategorien aufzeichnet, die auf dem Server ausgeführt werden, wird das erzeugte Protokoll auch bei der Analyse von Fehlern eingesetzt. Sie haben etwas früher vom neuen XEvent Profiler gelesen, dies ist nun, wenn man so will, der große Bruder. Er bietet eine Vielzahl an Einstellungs- und Filtermöglichkeiten.

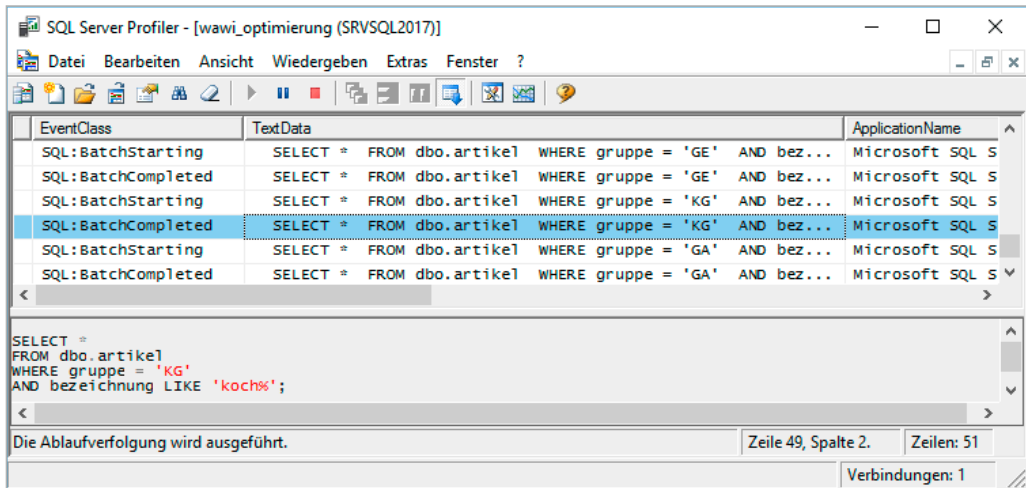


Bild 2.32 Aufzeichnung des SQL Server Profilers

■ 2.7 Der Datenbankoptimierungsratgeber

Der Datenbankoptimierungsratgeber (Database Tuning Advisor) gehört der Kategorie der *Leistungstools* an. Er liefert Änderungsvorschläge zur Indizierung. Dafür muss ihm eine Datei mit Abfragen, die typischerweise auf dieser Datenbank ausgeführt werden, bereitgestellt werden. Um sinnvolle Ergebnisse zu erzielen, muss die Datenbank allerdings realistische Datenmengen enthalten, die dem Echtbetrieb entsprechen. Dieses Eingabefile wird als Trace mit dem Profiler generiert.

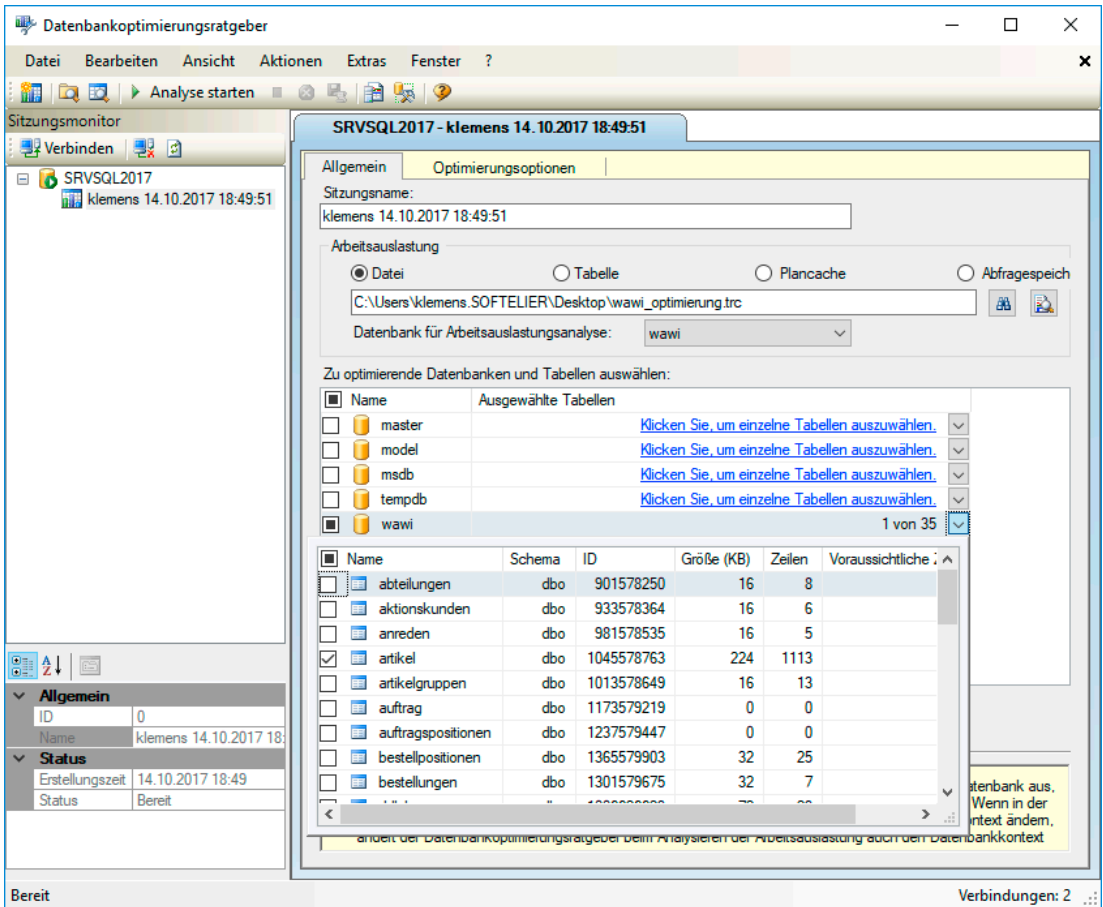


Bild 2.33 Datenbankoptimierungsratgeber

Nach der Auswahl der Trace-Datei und der zu analysierenden Datenbanken sowie Datenbankobjekte wird die Analyse gestartet. Als Ergebnis können Empfehlungen herauskommen. Im Beispiel wird empfohlen, einen gemeinsamen Index für die beiden Spalten mit der Artikelgruppe und der Bezeichnung in der Artikeltabelle erzeugen. Zusätzlich enthält die Empfehlung weitere eingeschlossene Spalten über der Option INCLUDE.

Das Skript zum Erzeugen des entsprechenden Index kann sogleich generiert und in die Zwischenablage kopiert werden. Danach kann die Anweisung im Management Studio eingefügt und ausgeführt werden.

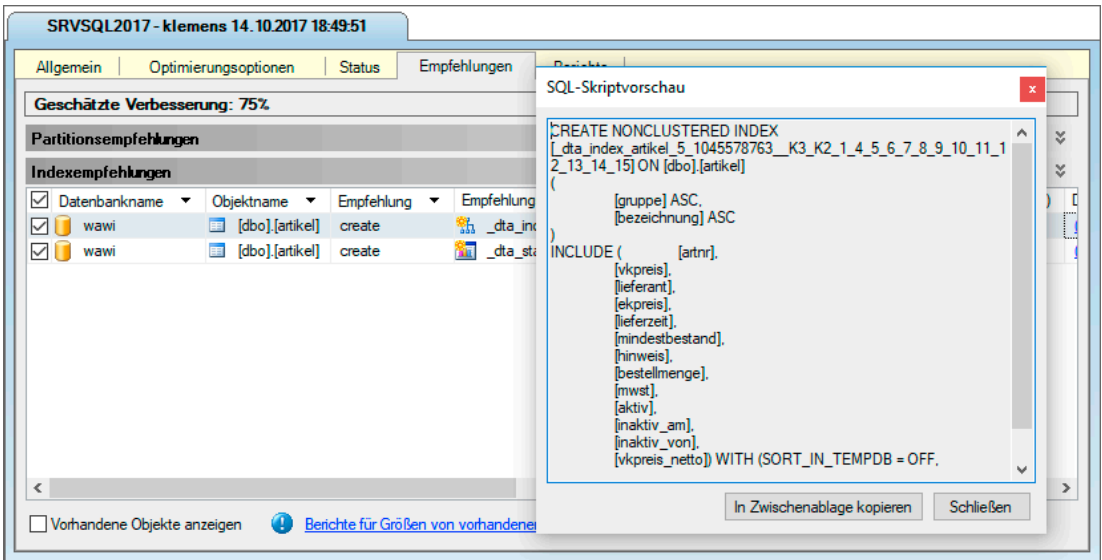


Bild 2.34 Indexempfehlung

■ 2.8 Die SQL Server Data Tools

Als Entwicklungstool über SQL und Transact-SQL hinaus fungiert das Visual Studio. Die darin inkludierten Werkzeuge werden als SQL Server Data Tools bezeichnet, sie stehen aber auch ohne eine Vollversion des Visual Studios zur Verfügung.

Mit den letzten Versionen des SQL Servers ist es regelmäßig zu Änderungen in der Art der Implementierung und zu Umbenennung, Trennung und Zusammenlegung von Werkzeugen gekommen. Um den Überblick nicht zu verlieren, finden Sie hier einen kurzen Ablauf der Entwicklung. Wenn Sie die Vorversionen nicht verwendet haben, können Sie den folgenden zeitlichen Abriss aber getrost überspringen.

Vor dem SQL Server 2012 ist unter dem Namen *Business Intelligence Development Studio* eine spezielle Form des Visual Studios installiert worden. Dieses enthielt lediglich die Projektvorlagen für *Analysis Services*, *Integration Services* und *Reporting Services*. Andere Projekte konnten mit dieser Visual Studio-Version nicht umgesetzt werden. Erst mit der Installation einer Professional Edition wurden auch andere Projekttypen verfügbar. In der Regel war dies immer eine Version hinter der jeweils aktuellen Version des Visual Studios.

Der Name *Business Intelligence Development Studio* ist beim SQL Server 2012 verschwunden. Das gleiche Werkzeug hörte nun auf den Namen *Visual Studio 2010 Shell*. Nach der SQL Server-Installation fanden Sie es im Startmenü einerseits mit dem eigenen Namen unter dem eigenen Ordner *Microsoft Visual Studio 2010* als auch im Ordner *SQL Server 2012* mit der Bezeichnung *SQL Server Data Tools* vor. Dieses konnte in zwei Versionen für das Visual