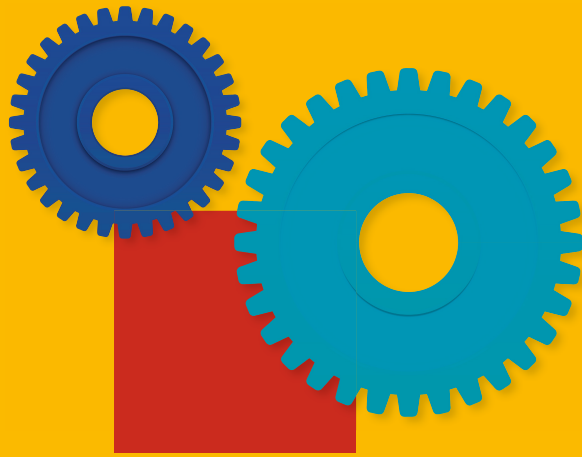


klemens KONOPASEK



# SQL SERVER 2014

DER  
SCHNELLE  
EINSTIEG

HANSER

## 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](http://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](http://www.hanser-fachbuch.de/update)





Klemens Konopasek

# SQL Server 2014

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.

© 2014 Carl Hanser Verlag München, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sieglinde Schär

Copy editing: Sandra Gottmann, Münster-Nienberge

Herstellung: Irene Weillhart

Umschlagdesign: Marc Müller-Bremer, [www.rebranding.de](http://www.rebranding.de), München

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Printed in Germany

Print-ISBN: 978-3-446-43938-2

E-Book-ISBN: 978-3-446-44108-8

**»Der Weltuntergang steht bevor,  
aber nicht so, wie Sie denken.  
Dieser Krieg jagt nicht alles in die Luft,  
sondern schaltet alles ab.«**



Tom DeMarco  
**Als auf der Welt das Licht ausging**

ca. 560 Seiten. Hardcover  
ca. € 19,99 [D] / € 20,60 [A] / sFr 28,90  
ISBN 978-3-446-43960-3  
Erscheint im November 2014

Hier klicken zur  
**Leseprobe**

Sie möchten mehr über Tom DeMarco und seine Bücher erfahren.  
Einfach reinklicken unter [www.hanser-fachbuch.de/special/demarco](http://www.hanser-fachbuch.de/special/demarco)

# Inhalt

<b>Vorwort</b> .....	<b>XI</b>
<b>1 Der SQL Server 2014 stellt sich vor</b> .....	<b>1</b>
1.1 SQL Server – wer ist das? .....	1
1.1.1 Der SQL Server im Konzert der Datenbanksysteme .....	1
1.1.2 Entscheidungsszenarien für Datenbanksysteme .....	4
1.1.3 Komponenten einer Datenbankanwendung .....	6
1.1.4 SQL Server – das Gesamtkonzept .....	9
1.2 Versionen und Editionen des SQL Servers .....	11
1.3 SQL Server 2014 installieren .....	15
1.4 Datenbanken installieren und nutzen .....	37
1.5 Gratis: die Express Edition .....	42
1.6 SQL Server Feature Pack .....	43
<b>2 Die grafischen Tools des SQL Server 2014</b> .....	<b>45</b>
2.1 Die Tools im Überblick .....	45
2.2 Das Management Studio .....	47
2.3 Das Kommandozeilentool: SQLCMD .....	63
2.4 Der Konfigurations-Manager .....	65
2.5 Das SQL Server-Installationscenter .....	67
2.6 Der Profiler .....	67
2.7 Der Datenbankoptimierungsratgeber .....	68
2.8 Die SQL Server Data Tools .....	70
2.9 Der Import/Export-Assistent .....	73
2.10 Der SQL Server Migration Assistant .....	81
<b>3 Eine neue Datenbank erstellen</b> .....	<b>85</b>
3.1 Erstellen einer neuen Datenbank .....	85
3.1.1 Bestandteile einer Datenbank .....	85
3.1.2 Datenbank mit dem grafischen Tool anlegen .....	89

3.1.3	Datenbank über eine SQL-Anweisung erstellen	96
3.1.4	Datenbank mit Filestream ausstatten	97
3.2	Tabellen in der Datenbank erstellen	102
3.2.1	Tabellenfelder definieren	103
3.2.2	Spalteneigenschaften	107
3.2.3	Constraints	110
3.2.4	Indizierung	119
3.2.5	Erste Daten erfassen	127
3.3	Datenbankdiagramme einsetzen	129
3.4	Richtlinien für Benennungsregeln einsetzen	133
3.5	Was Sie noch wissen sollten	138
3.5.1	Tabellen in anderen Dateigruppen speichern	138
3.5.2	Tabellen direkt mit DDL-Anweisungen erstellen	139
3.5.3	Gefahren der grafischen Oberfläche	141
3.5.4	Berechnete Spalten integrieren	144
3.5.5	Objekte und Datenbanken skripten	146
3.6	Tabelle mit Filestream und FileTable	150
3.6.1	Tabelle mit Filestream erstellen	150
3.6.2	Objekte in einer FileTable speichern	156
3.7	Beispieldatenbank generieren	169
3.8	Speicheroptimierte Tabellen	171
3.8.1	Datenbank mit In-Memory-Filegroup erstellen	171
3.8.2	Speicheroptimierte Tabelle anlegen	172
3.8.3	Speichernutzung beschränken	176
<b>4</b>	<b>SQL – Zugriff auf Daten</b>	<b>179</b>
4.1	Einsatz des Abfrage-Designers	180
4.1.1	Die Bereiche des Abfrage-Designers	181
4.1.2	Erstellen einer Abfrage	185
4.2	Sichten für den Datenzugriff gestalten	198
4.2.1	Gründe für den Einsatz von Sichten	198
4.2.2	Erstellen einer Sicht	199
4.2.3	Daten aus einer Sicht abrufen	201
4.3	SQL-Anweisungen verwenden	203
4.3.1	Data Query Language (DQL)	204
4.3.2	Data Manipulation Language (DML)	212
4.3.3	Die MERGE-Anweisung	213
4.3.4	Den Abfrage-Designer im Abfrageeditor einsetzen	217
4.4	Abfragen mit Geodaten	219
4.4.1	Typen im Geodatenmodell	220
4.4.2	Geodaten in Tabellen speichern und verwenden	226
4.4.3	Index für räumliche Daten	241



<b>5</b>	<b>Transact-SQL – die Sprache zur Serverprogrammierung</b>	<b>245</b>
5.1	Bestandteile und Funktionalität von Transact-SQL	247
5.1.1	Variablen und Datentypen	247
5.1.2	Benutzerdefinierte Tabellentypen	254
5.1.3	Funktionen	256
5.1.4	Kontrollstrukturen	276
5.1.5	Cursor für Datenzugriffe einsetzen	293
5.2	Transaktionen gezielt steuern	299
5.2.1	Automatische Transaktionen	299
5.2.2	Explizite und implizite Transaktionen	300
5.2.3	Benannte Transaktionen	306
5.3	SET-Optionen verwenden	307
5.4	Fehlerbehandlung in den Code einbauen	315
5.5	Sequenzen	321
5.6	Paging mit OFFSET und FETCH	323
5.7	Window-Funktionen	325
<b>6</b>	<b>Gespeicherte Prozeduren, Funktionen und Trigger</b>	<b>327</b>
6.1	Gespeicherte Prozeduren programmieren	328
6.1.1	Aufbau einer gespeicherten Prozedur	329
6.1.2	Erzeugen einer gespeicherten Prozedur	331
6.1.3	Einfache gespeicherte Prozeduren	339
6.1.4	Gespeicherte Prozeduren mit Eingabeparametern	341
6.1.5	Ergebnisrückgabe von Prozeduren	344
6.1.6	Cursor in gespeicherten Prozeduren nutzen	354
6.1.7	Transaktionen in Prozeduren	361
6.1.8	Table-Valued Parameter einsetzen	364
6.1.9	Systemintern kompilierte gespeicherte Prozeduren	368
6.1.10	Gespeicherte Prozeduren aus Client-Anwendungen heraus aufrufen	373
6.2	Mit Triggern automatisieren	387
6.2.1	DML-Trigger: Insert, Update, Delete	388
6.2.2	Trigger löschen	415
6.2.3	DDL-Trigger	415
6.3	Benutzerdefinierte Funktionen implementieren	422
6.3.1	Skalarwertfunktionen	422
6.3.2	Inline-Funktionen	428
6.3.3	Tabellenwertfunktionen	429
6.4	Debuggen	435
6.4.1	Voraussetzungen für das Debuggen	435
6.4.2	Debuggen einer gespeicherten Prozedur	437
6.4.3	Debuggen von Triggern	441
6.4.4	Debuggen von Funktionen	443

<b>7</b>	<b>SQL Server CLR-Integration</b>	<b>445</b>
7.1	Mit im Boot: .NET Framework	446
7.1.1	Integration mit dem Visual Studio	448
7.2	CLR-Aktivierung	451
7.2.1	Code auf den Server bringen: Assembly	452
7.3	.NET User-Defined Functions	455
7.4	.NET Stored Procedures	461
7.4.1	Datenzugriff aus der CLR heraus	461
7.4.2	Prozeduren mit Werterückgabe	462
7.4.3	Zugriff auf externe Daten	466
7.4.4	.NET-Trigger	470
7.5	User-Defined Aggregates (UDA)	476
7.6	Externe Assemblies verwenden	481
7.7	Verwalten des Servers mit SMO	486
7.8	Übrigens: Debuggen	490
7.8.1	Debuggen einer T-SQL Stored Procedure	491
7.8.2	Debuggen einer .NET Stored Procedure	492
<b>8</b>	<b>Data Tier Applications und SQL Server Data Tools</b>	<b>495</b>
8.1	Datenebenenanwendungen	495
8.1.1	DAC über Management Studio erstellen	496
8.1.2	Eine DAC auf dem SQL Server bereitstellen	498
8.1.3	Aktualisieren einer DAC	500
8.1.4	Entfernen einer DAC	502
8.1.5	Von DACPAC zu BACPAC	503
8.1.6	Erstellen einer DAC mit dem Visual Studio	508
8.2	Die SQL Server Data Tools	508
8.2.1	Ein neues Datenbankprojekt erstellen	509
8.2.2	Datenbankobjekte erstellen	511
8.2.3	Datenbankprojekt bereitstellen	514
8.2.4	Schemavergleich	518
8.2.5	Datenbank in ein Datenbankprojekt importieren	522
8.2.6	Ersatz für das Management Studio?	524
<b>9</b>	<b>Client-Server-Datenbank verwalten</b>	<b>527</b>
9.1	Anfügen und Trennen von Datenbanken	527
9.1.1	Trennen einer Datenbank	527
9.1.2	Anfügen einer Datenbank	530
9.1.3	Option Automatisch schließen	535
9.2	Datenbank sichern	536
9.2.1	Sicherungsvarianten	536
9.2.2	Sicherungsziele	537
9.2.3	Sicherung mit dem Management Studio	540

9.2.4	Sicherung über TRANSACT-SQL .....	545
9.2.5	Zeitgesteuerte Sicherung mit dem SQL Server-Agent .....	547
9.2.6	Zeitgesteuerte Sicherung mit der Express Edition .....	552
9.2.7	Datenbank wiederherstellen .....	555
9.2.8	Einsatz der Zeitachse beim Wiederherstellen .....	558
9.2.9	Wiederherstellung über Transact-SQL .....	563
9.2.10	Desaster Recovery .....	564
9.2.11	Recovery mit FILESTREAM .....	571
9.3	Datenänderungen protokollieren .....	573
9.3.1	Change Data Capture .....	573
9.4	Mit mehreren Instanzen arbeiten .....	579
9.4.1	Standardinstanzen und benannte Instanzen .....	579
9.4.2	Zugriff auf Instanzen steuern .....	581
<b>10</b>	<b>Sicherheit und Zugriffsberechtigungen .....</b>	<b>585</b>
10.1	Authentifizierungsmodi - Anmeldungen und Benutzer .....	585
10.1.1	Windows-Authentifizierung .....	587
10.1.2	Gemischter Modus .....	587
10.1.3	Anmeldung und Benutzer .....	588
10.2	Berechtigungen .....	589
10.3	Rollen .....	590
10.3.1	Serverrollen .....	590
10.3.2	Datenbankrollen .....	593
10.3.3	Anwendungsrollen .....	594
10.4	Anmeldeinformationen (Credentials) .....	594
10.5	Schema .....	596
10.6	Verwaltung im Management Studio .....	599
10.6.1	Serveranmeldung hinzufügen .....	599
10.6.2	Schema anlegen .....	604
10.6.3	Datenbankbenutzer hinzufügen .....	605
10.6.4	Rollen in einer Datenbank anlegen .....	608
10.7	Berechtigungen vergeben .....	610
10.7.1	Berechtigungen auf Datenbankebene .....	610
10.7.2	Berechtigungen auf Serverebene .....	616
10.8	Lösungen mit T-SQL .....	618
10.8.1	Sicherheitsobjekte anlegen .....	618
10.8.2	Generische Skripte .....	623
10.9	Contained Databases .....	624
10.10	Administratorzugriff wiederherstellen .....	630
10.11	Indirekte Zugriffe verwalten .....	633
10.11.1	Datenzugriffe über Sichten .....	633
10.11.2	Sicherheit mit Prozeduren erhöhen .....	635

10.12	Zugriff auf andere Server .....	641
10.12.1	SQL Server als Verbindungsserver .....	642
10.12.2	Verbindungsserver mit Fremdprodukten .....	649
<b>11</b>	<b>Erweiterte Funktionalitäten .....</b>	<b>653</b>
11.1	Datenbank-E-Mail .....	653
11.1.1	Einrichten von Datenbank-E-Mail .....	654
11.1.2	E-Mails aus der Anwendung heraus versenden .....	660
11.1.3	Varianten des E-Mail-Versands .....	662
11.1.4	Mailbenachrichtigung für Agent-Aufträge .....	668
11.2	Integration Services .....	673
11.2.1	Datenabgleich mit IS .....	674
11.2.2	Pakete ausführen und auf den Server bringen .....	693
11.2.3	SSIS-Projekte auf den Server bringen .....	695
11.3	PowerPivot .....	701
11.3.1	Was ist PowerPivot? .....	701
11.3.2	PowerPivot für Excel .....	702
<b>Anhang</b>	<b>.....</b>	<b>727</b>
A.1	Die Tabellen der Datenbank WAWI .....	727
<b>Index</b>	<b>.....</b>	<b>735</b>

# 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!

Der SQL Server 2014 kommt zu einer Zeit, da Servervirtualisierung salonfähig geworden ist und der Weg in die Cloud auch für Datenbanken an der Schwelle zur breiteren Anerkennung steht. Die Virtualisierung und die Cloud sind nun also endgültig auch bei der Datenbank angekommen. Dies hängt auch damit zusammen, dass sich Hyper-V von Microsoft derart weiterentwickelt, dass ursprüngliche Vorbehalte speziell für Datenbankserver nicht mehr bestehen. Durch technische Neuerungen gibt es keine Nachteile mehr gegenüber einem physischen Server. Damit ist eine der letzten Virtualisierungslücken bereits geschlossen. Wenn ich an die beiden letzten Jahre denke, sind ausnahmslos alle neuen SQL Server bei meinen Kunden virtualisierte Server. Anwendungen in die Cloud auszulagern verliert langsam an Schrecken und Vorbehalte verschwinden.

Mit Windows Azure SQL-Datenbank steht eine einfach zu verwendende und leistungsstarke Cloud-Plattform für den SQL Server zur Verfügung, die 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 früher mit dem Tausch der Server-Hardware einhergegangen, kann aufgrund der beschriebenen Entwicklungen ein Umstieg wesentlich zügiger vorstattgehen. Sie müssen nicht so lange auf den Einsatz der tollen neuen Features warten.

Die Neuerungen des SQL Server 2014 gegenüber seiner Vorversion sind auf die drei Schwerpunkte fokussiert, Daten unter dem Schlagwort „In-Memory OLTP“ zur Gänze im Arbeitsspeicher zu halten, die Hochverfügbarkeit mit AlwaysOn weiter stark zu verbessern und Tabellen mit Columnstore-Indizes nun auch aktualisieren zu können.

Verbesserte Werkzeuge für die Entwicklung unterstützen die Arbeit in einheitlicher Form für alle Plattformen. Die einheitliche Entwicklungsoberfläche stellt eines der Schwerpunktthemen dar. Die Bereiche Datenbank- und Anwendungsentwicklung wachsen immer näher zusammen. Sehen Sie sich das an, Sie werden sicher auch begeistert sein.

Dieses Buch richtet sich an all diejenigen, die sich in SQL Server 2014 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 zum Beispiel den SQL Server 2012 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 7/8 bis hin zu großen Multiprozessor-Servern unter Microsoft Windows Server 2012 R2 Datacenter Edition.

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 2014. Neben der Vorstellung der Editionen sowie der Erläuterung der Vorgehensweise zur Installation erfahren Sie, welche Voraussetzungen Ihr System für den Einsatz von SQL Server 2014 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.

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. 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 Datenebenenapplikationen ein eigenes Kapitel, das mit der Nummer 8, zu bekommen. Datenebenenapplikationen, oder Data Tier Applications, wie sie im Original genannt werden, sind nun schon in der dritten Version verfügbar und mittlerweile nun 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 wieder herstellen können. Datenbanksicherungen haben ihre Bedeutung aber nicht nur in einem Störfall, sondern sind auch in der täglichen Arbeit mit der Daten-

bank wichtig, weil sie zum Beispiel auch dafür verwendet werden, eine Datenbank von einem Server auf einen anderen zu übertragen.

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.

In Kapitel 11 erläutere ich Ihnen zwei erweiterte Funktionalitäten, die Ihnen ergänzend zur Verfügung stehen, falls Sie nicht die Gratis-Edition des SQL Servers 2014 verwenden. Ich stelle Ihnen hierbei Datenbank-E-Mail sowie die Integration Services etwas genauer vor. Als Spezialthema lesen Sie zum Abschluss, was Microsoft unter Self Service BI versteht, und ich zeige Ihnen, wie Sie mit PowerPivot für Excel arbeiten und einfach und schnell aus großen und heterogenen Datenbeständen eine vielsagende Auswertung zaubern.

Mit diesem Buch lernen Sie anhand von problembezogenen Aufgabenstellungen in anschaulicher und systematischer Form die zahlreichen Möglichkeiten des SQL Server 2014 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 2014.

*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 2014 stellt sich vor

Der SQL Server 2014 ist da – wieder eine tolle neue Version. Sie geht den Weg konsequent weiter, der mit den beiden Vorgängerversionen eingeschlagen worden ist. Ich bin von den Neuerungen der Version 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 in dieser Version ist die Möglichkeit, Tabellen einer Datenbank im Arbeitsspeicher zu halten, auch wenn dieses Feature leider auf die Enterprise-Edition beschränkt ist.

Ich hoffe, auch Sie gehen mit Freude an den SQL Server 2014 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 2014 verfügbar ist, und zeige Ihnen, wie Sie bei der Installation vorgehen. Darüber hinaus werden Sie erfahren, wie Sie mit dem SQL Server 2014 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*.

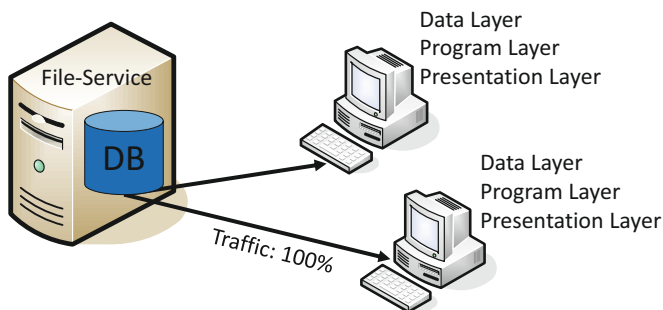
Somit gehen wir von relationalen Datenbanksystemen aus und unterteilen diese in

- *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.

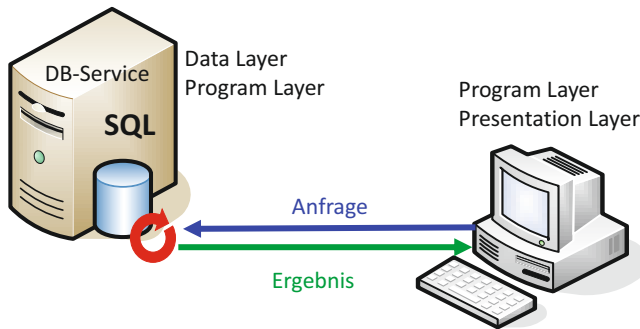


**Bild 1.1**  
Konzept von Desktop-Datenbanken

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

- Oracle
- DB2 von IBM
- Adaptive Server Enterprise von Sybase

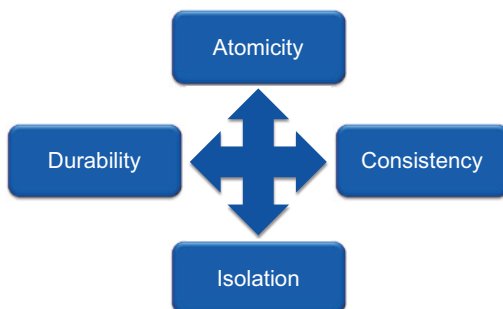
Als Open-Source-Datenbanksysteme sind zusätzlich von Bedeutung:

- PostgreSQL
- MySQL/MariaDB

Der SQL Server ist mittlerweile das führende serverbasierte Datenbanksystem auf Windows-Plattformen. Allerdings sind die anderen genannten Systeme auch für diverse Plattformen verfügbar.

### 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 Konsistenzerhaltung 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 Vorganges. Auf 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 neuen speicheroptimierten Tabellen, die im RAM des Servers gehalten werden. Auf den ersten Blick würde man vermuten, dass diese vor allem in 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.
- *Netzlast*: Aufgrund der Topologie, dass nur das Ergebnis einer Anfrage vom Server an den Client übertragen wird, der diese Daten dann anzeigt und verarbeitet, können serverbasierte 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.

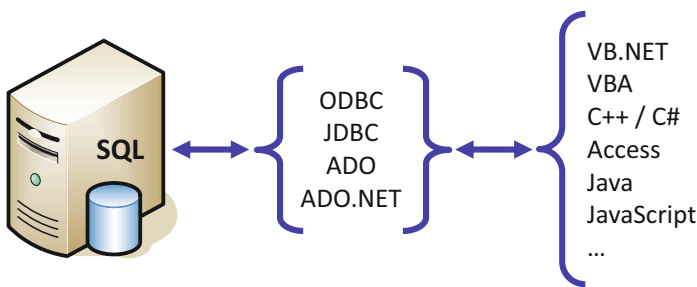
Analysieren Sie Ihre Anforderungen an ein Datenbanksystem anhand dieser Anforderungen und treffen Sie dann Ihre Entscheidung.



**HINWEIS:** Der Microsoft SQL Server bietet ein professionelles Server-Datenbanksystem zu einem vergleichsweise niedrigen 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.

### 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 Standard-schnittstellen

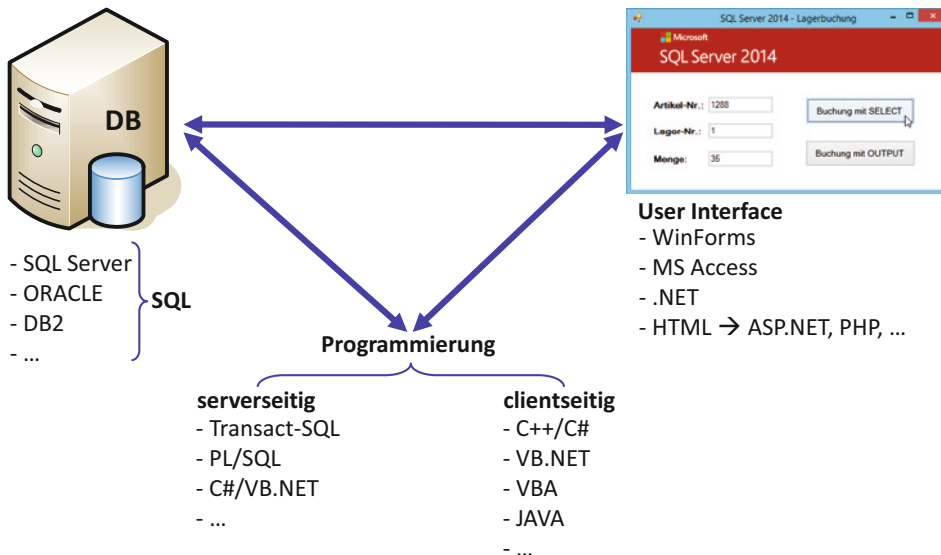
Eine Datenbankanwendung besteht in der Regel aus folgenden Komponenten:

- 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:

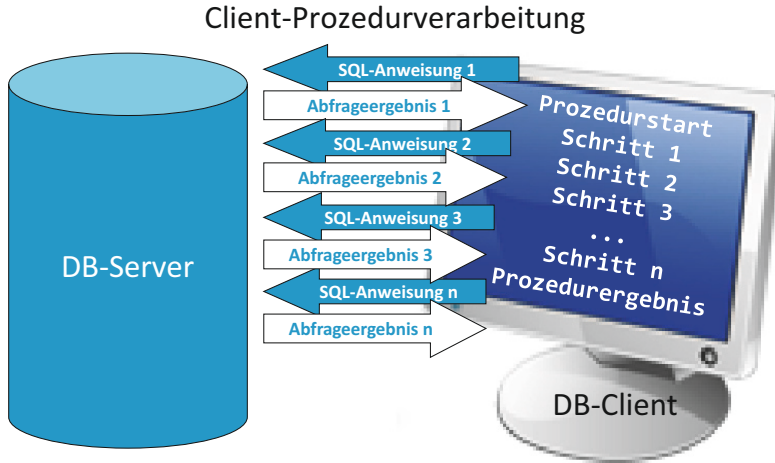
- Den SQL Server installieren und konfigurieren.
- Datenbanken und Datenbankobjekte mit dem SQL Server erstellen.
- Den Zugriff auf Daten mit der Structured Query Language (SQL) vollziehen.
- Serverseitig mit Transact-SQL und .NET programmieren.
- Die Benutzerverwaltung zur Vergabe von Berechtigungen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

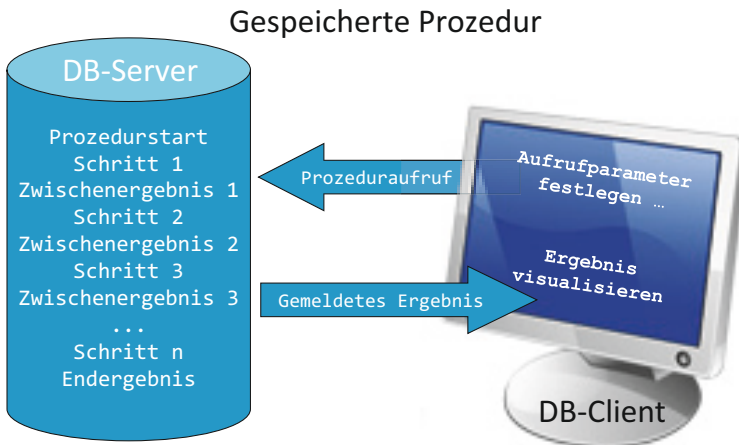
Die beiden nachfolgenden Abbildungen zeigen die Unterschiede beim Programmablauf von Programmcode, der auf dem Client oder auf dem Server läuft.

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 Datenbankumfeld.

Zur Datenbank-Engine selber zählen folgende Features:

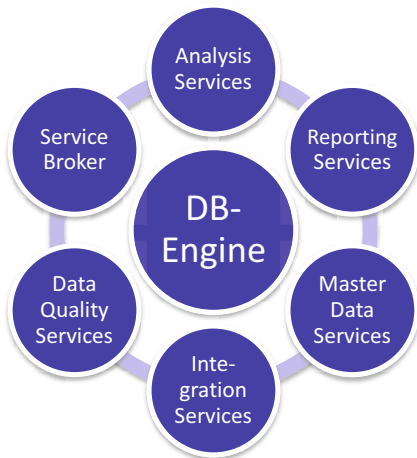
- Volltextsuche
- Datenbankreplikation

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 Berichterstellung 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 2014-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. Dies ist eine Verbesserung gegenüber älteren Versionen, bei denen dieser in der Express Edition gefehlt hat.
- *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 2014 zugrunde. Diese Version weist gegenüber dem Vorgängersystem SQL Server 2012 wesentliche Neuerungen auf. Dies betrifft nicht nur die eigentliche relationale Datenbank-Engine, die den Kern des Produktes 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. Die drei großen Schwerpunkte, die Microsoft für diese Version gesetzt hat, sind:

- *Speicheroptimierte Tabellen*: Ein wahrer Performanceboost sind die neuen speicheroptimierten Tabellen. Hinter diesem Begriff verbirgt sich die Möglichkeit, ganze Tabellen vollständig im RAM zu halten. Alles in diesem Zusammenhang ist auf Performance ausgelegt, weshalb jegliche Features, welche die Zugriffsgeschwindigkeit bremsen würden, bei diesen Tabellen fehlen. Da der Zugriff auf Festplatten in der Regel das ist, was eine Datenbank am meisten bremst, bietet diese Möglichkeit ungeahnte Performancesteigerungen gegenüber herkömmlichen Tabellen. Einziges Manko aus meiner Sicht: Sie sind nur in der Enterprise Edition verfügbar.
- *Columnstore-Indizes*: Die mit dem SQL Server 2012 neu eingeführte Indexart, die sehr schnelle Ergebnisse bei sogenannten Star-Joins liefert, belässt die zugrunde liegende Tabelle nun aktualisierbar. In der ersten Version war eine Tabelle schreibgeschützt, sobald ein solcher Index erzeugt worden ist. Dies ist nun nicht mehr der Fall, was von der Bedeutung her einer Revolution gleichkommt. Star-Joins sind Auswertungen mit einer Faktentabelle quasi in der Mitte und sternförmigen Joins zu Stammdatentabellen. Der Einsatzbereich sind große Tabellen in Data Warehouses. Dieses Feature kommt also bei OLAP- und weniger bei OLTP-Anwendungen zum Einsatz.
- *Hochverfügbarkeit mit AlwaysOn*: Der SQL Server 2014 enthält einige Erweiterungen und Verbesserungen für AlwaysOn. Darunter versteht man eine Anzahl an Mechanismen – allen voran die Failo-Verclusterung –, um Hochverfügbarkeit der Datenbanken zu gewährleisten. Damit gewinnt der SQL Server zusätzliches Terrain, wenn es um das Hosten kritischer Anwendungen geht.

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

**Tabelle 1.1** Einige Neuerungen in SQL Server 2014

Thema	Beschreibung
Windows Azure	<p>Daten können, anstelle lokal auf dem Server gehalten zu werden, in Windows Azure-BLOBs abgelegt werden.</p> <p>Das Hosten von SQL Server-Datenbanken auf einem virtuellen Computer in Windows Azur wird über eigene Bereitstellungsassistenten unterstützt.</p> <p>Die Sicherung einer SQL Server-Datenbank kann über eine URL in Windows Azure-BLOBs erfolgen.</p>

**Tabelle 1.1** Einige Neuerungen in SQL Server 2014 (Fortsetzung)

Thema	Beschreibung
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.)
Pufferpool-erweiterung	Durch diese Erweiterungen können Solid State Drives (SSD) als Datenträger für Datenbanken optimal zur Verbesserung des Datendurchsatzes eingesetzt werden.
Transact-SQL-Erweiterungen	Zahlreiche Erweiterungen für die neuen im Speicher gehaltenen Tabellen (In-Memory OLTP) beinhalten unter anderem eine eigene Art von Stored Procedures für diese Tabellen.
Sicherheits-erweiterungen	Einige neue Berechtigungen auf Serverebene erweitern die Granularität der Berechtigungsverwaltung auf dieser Ebene. Zum Beispiel ist dies die Berechtigung CONNECT ANY DATABASE, mit der man sich mit jeder Datenbank auf diesem Server verbinden kann, auch wenn man keinen Datenbankbenutzer in dieser besitzt.
SQL Server Data Tools for Business Intelligence	Die SSDT-BI lösen die bisherigen Client-Werkzeuge für Reporting, Integration und Analysis Services ab. Diese sind nicht nur in einer Version für Visual Studio 2012, sondern mittlerweile auch für Visual Studio 2013 verfügbar. Damit ist ein derartiges Tool erstmals auch in der aktuellen Version des Visual Studio verfügbar und nicht mehr immer eine Version zurück wie bisher.

## Editionen des SQL Server 2014

Microsoft liefert den SQL Server 2014 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.

Einen schnellen Überblick über die Varianten gibt die folgende Tabelle:

**Tabelle 1.2** SQL Server 2014-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.
Business Intelligence Edition	Diese Edition steht zwischen der Standard und Enterprise Edition. Sie bietet den Großteil der Features im Bereich von Business Intelligence, aber verfügt im Gegensatz zur Enterprise Edition nicht über die vollen Möglichkeiten im Bereich der Hochverfügbarkeit und fortgeschrittenen Sicherheit.

Edition	Bedeutung/Anwendung
Standard Edition	Dieses Angebot richtet sich an kleine und mittelständische Unternehmen, welche die erweiterten Leistungsoptionen der SQL Server 2014 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.
Express Edition	Die Express Edition ist eine frei erhältliche Datenbank, die einfach zu verwenden und zu verwalten ist. Sie ist auch in Microsoft Visual Studio integriert und 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 Tools und mit Advanced Services. Express Edition mit Tools enthält zusätzlich bereits mit dem Management Studio eine grafische Entwicklungs- und Verwaltungsumgebung. Die Express Edition mit Advanced Services 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 sehr kostengünstige Edition für Entwickler. Diese Edition darf keine Produktivsysteme hosten; sie entspricht aber ihrem Umfang nach der Enterprise Edition.

Der SQL Server 2014 bietet bis hinunter zur Express Edition sowohl 32- als auch 64-Bit-Versionen an. Dies trägt dem Zustand Rechnung, dass auch im Desktop-Bereich inzwischen immer mehr 64-Bit-Systeme im Einsatz sind. Überraschend für mich ist, dass auch die Enterprise Edition noch immer als 32-Bit-Version angeboten wird.

Die **Enterprise Edition** ist das Flaggschiff unter den Versionen. Sie bietet eine skalierbare Datenbankserverumgebung für Unternehmen jeglicher Größe. Die Enterprise Edition besitzt keine Beschränkungen in 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 16 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 **Business Intelligence Edition** ist zwar, was die Datenbank betrifft, ebenso wie die Standard Edition auf 16 Prozessorkerne limitiert, kann aber für BI alles nutzen, was das Betriebssystem hergibt.

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 2014 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. Sie finden den Download unter:

<http://technet.microsoft.com/de-de/evalcenter/dn205290>

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 <http://azure.microsoft.com/de-de/services/sql-database/>.