

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.

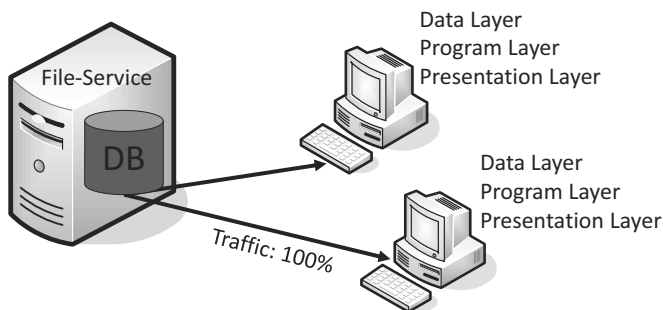


Bild 1.1
Konzept von Desktop-Datenbanken

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.

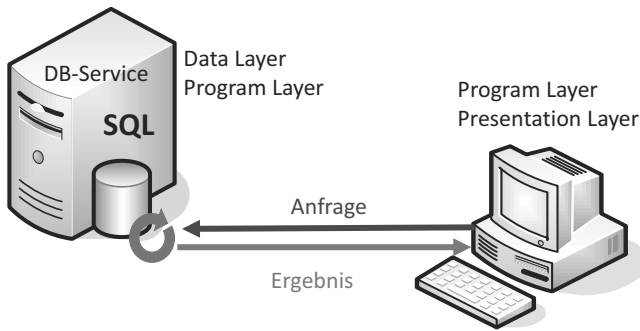


Bild 1.2
Konzept von Server-Datenbanksystemen

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.

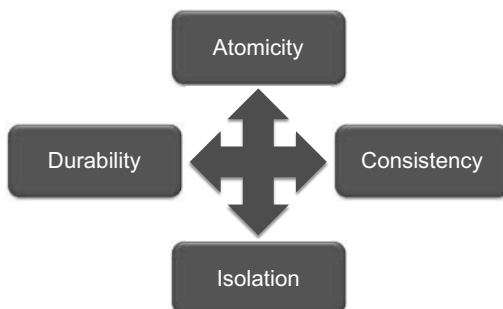


Bild 1.3
Das ACID-Konsistenzmodell

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.

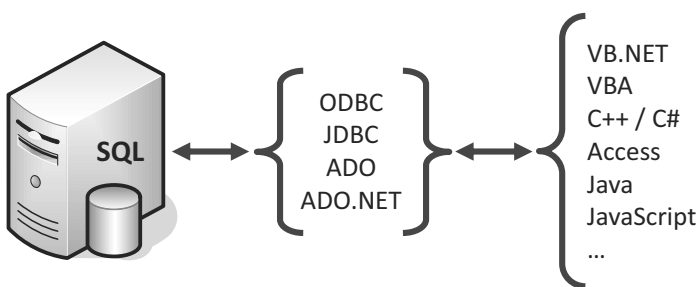


Bild 1.4
Zugriff auf eine Datenbank über Standard-schnittstellen

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.

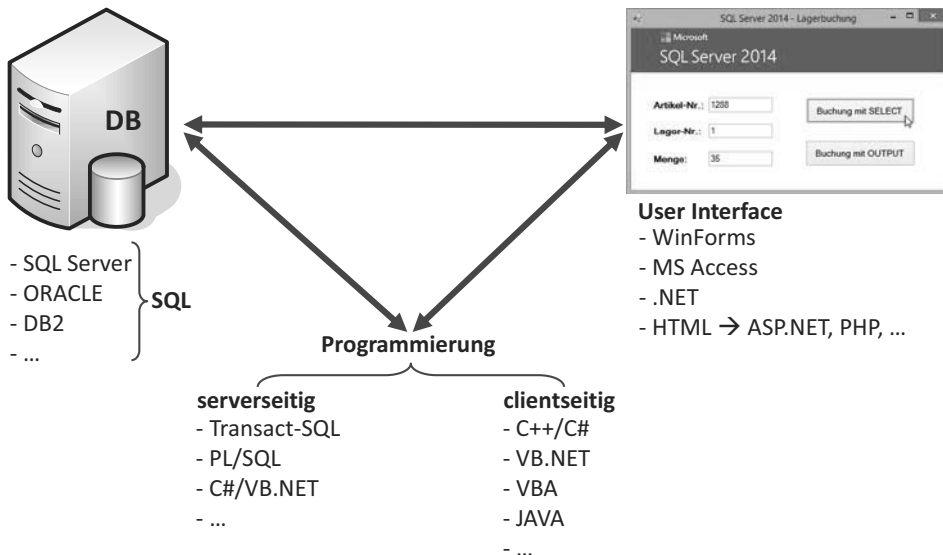


Bild 1.5 Datenbankanwendung und ihre Bestandteile

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.

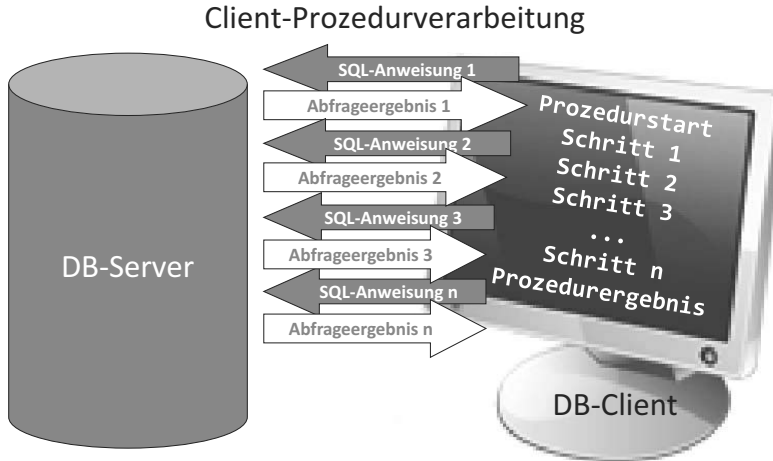


Bild 1.6 Programmlogik im Frontend

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.

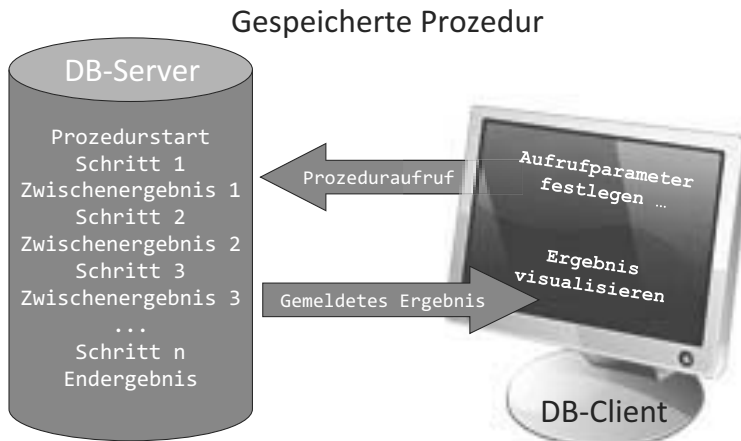


Bild 1.7 Serverseitige Programmierung

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 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?“

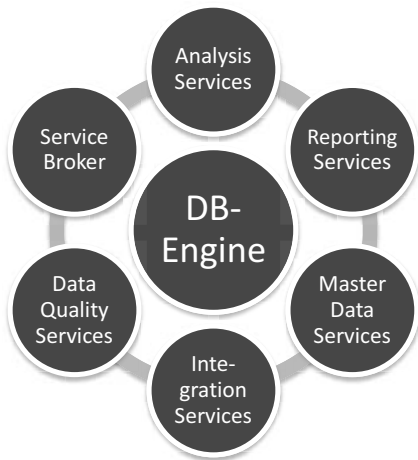


Bild 1.8

SQL Server und seine Zusatzkomponenten

Natürlich stehen diese Features nicht in jeder der verschiedenen SQL Server 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 SSDD-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/>.

■ 1.3 SQL Server 2014 installieren

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

Voraussetzungen für die Installation

Voraussetzung für die Nutzung von Microsoft SQL Server 2014 ist eine fehlerfreie Installation. Dazu 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 ca. 811 MB zu Buche. Weitere 591 MB benötigen jeweils die Integration Services und 304 MB sind für die Reporting Services erforderlich. Die Clientkomponenten beanspruchen mit 1823 MB viel Kapazität. Weitere 375 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.
- *Prozessor:* Sie benötigen für 32-Bit-Versionen des SQL Servers 2014 einen Pentium III-Prozessor mit mindestens 1,0 GHz Taktfrequenz (2 GHz oder mehr wird empfohlen). Für eine 64-Bit-Version benötigen Sie mindestens einen Pentium IV- oder Xeon-Prozessor mit 64-Bit-Unterstützung und 1,4 GHz Taktfrequenz.
- *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 <http://msdn.microsoft.com/de-de/library/ms143506.aspx>.

Tabelle 1.3 Betriebssystemanforderungen für die Installation

Edition	Betriebssystem
Enterprise, Business Intelligence, Web	Windows Server 2008 SP2, Windows Server 2008 R2 SP1, Windows Server 2012, Windows Server 2012 R2
Standard, Developer, Express	Windows Server 2008 SP2, Windows Server 2008 R2 SP1, Windows Server 2012, Windows Server 2012 R2, Windows 7 SP1, Windows 8, Windows 8.1

Für die Installation von SQL Server 2014 sind außerdem erforderlich:

- .NET Framework 3.5 SP1
- .NET Framework 4.0 (dieses wird, wenn nicht vorhanden, mit installiert)

Für die Clientkomponenten wird zusätzlich benötigt:

- Windows PowerShell 2.0

Fehlt das .NET Framework 4.0, erledigt das SQL Server Setup-Programm dies für Sie. Das .NET-Framework 3.5 SP1 und PowerShell 2.0 müssen Sie selber installieren oder je nach verwendetem Betriebssystem auch nur aktivieren.

Das SQL Server-Installationscenter



HINWEIS: Die nachfolgenden Screenshots habe ich beim Setup einer Enterprise Edition auf einem Windows Server 2012 R2 angefertigt.

Wie müssen Sie vorgehen, um den SQL Server 2014 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 2014-DVD in das Laufwerk ein. Im Normalfall wird das Installationsprogramm auf der DVD automatisch gestartet. Der angezeigte Start-Bildschirm 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*

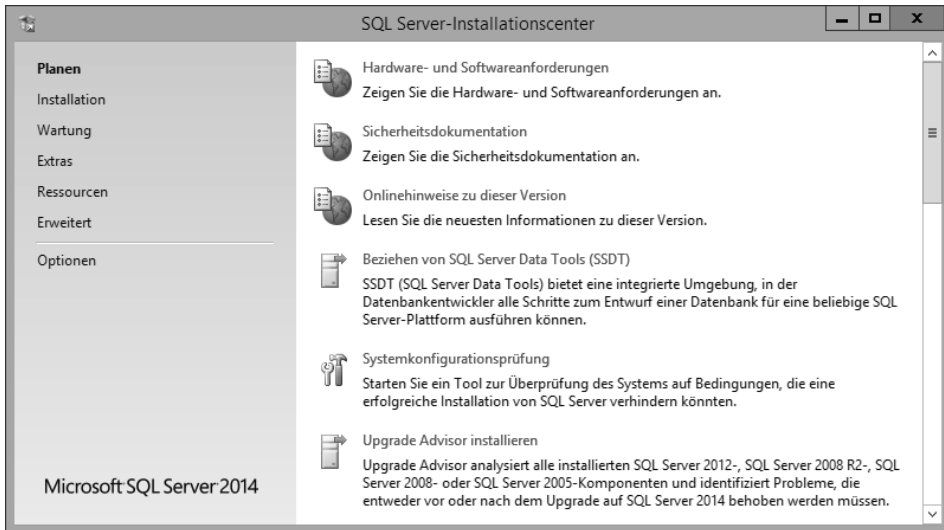


Bild 1.9 Startbildschirm von *SQL Server-Installationscenter*

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 Online-Hilfe 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.

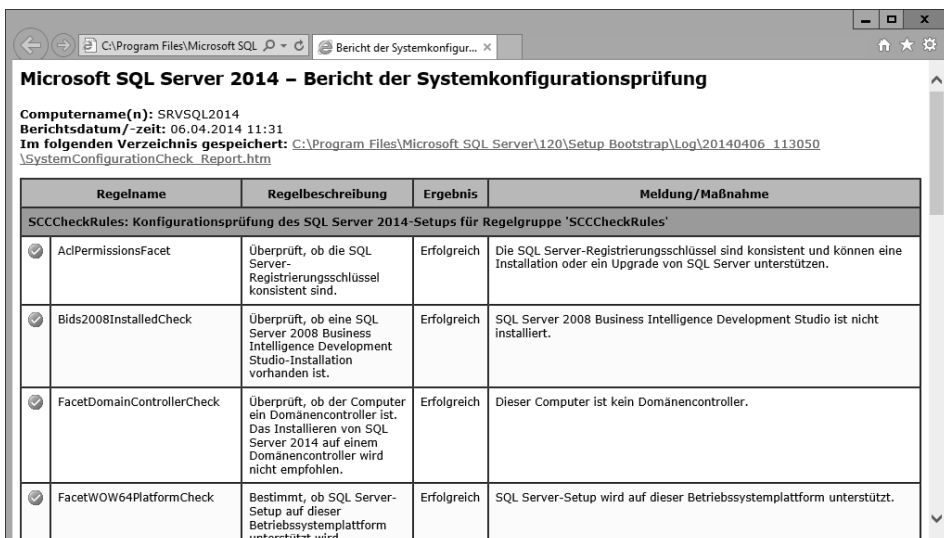


Bild 1.10 Bericht der Systemkonfigurationsprüfung



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

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 Upgrade Advisor verwendet haben – ist ebenfalls hier über *Upgrade von SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 oder SQL Server 2012* zu starten.



ACHTUNG! Da der SQL Server 2014 wie auch schon sein Vorgänger unter Windows Server 2003 nicht mehr installiert werden kann, kann ein Upgrade auf dieser Plattform nicht direkt erfolgen. Sie müssen stattdessen SQL Server 2014 auf einen neuen Server übertragen.

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. Auch das Tool, um PowerPivot in eine bestehende SharePoint-Installation zu integrieren, finden Sie hier. Neu in dieser Version ist hier die Option, die *SQL Server Data Tools - Business Intelligence* direkt zu installieren. Diese sind früher unter dem Namen *Business Intelligence Development Studio* bekannt gewesen und dienen zum Beispiel dem Erstellen von Integration Services-Paketen.

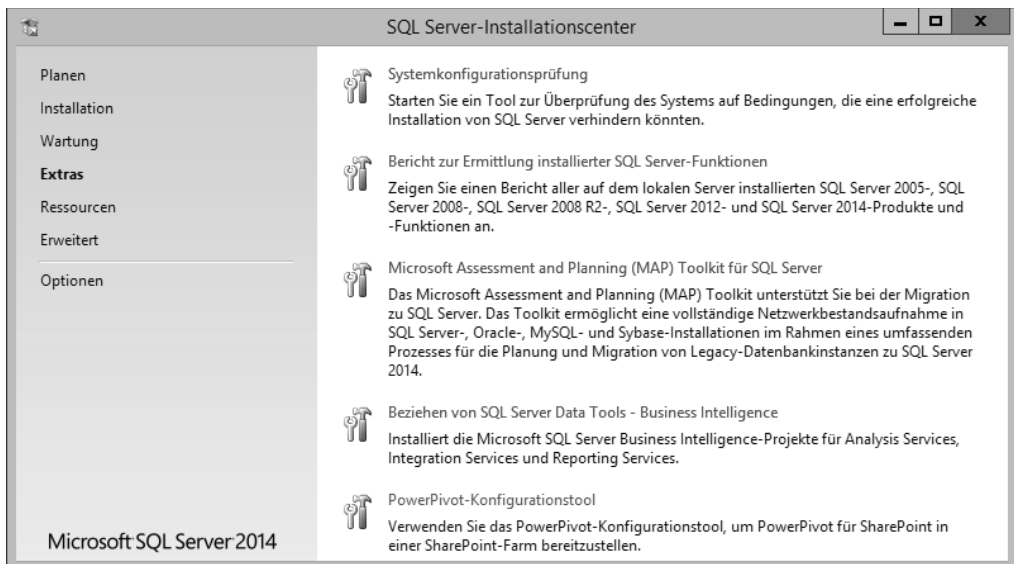


Bild 1.13 *Extras* des Installationscenters

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 eine 32- oder eine 64-Bit-Version installieren möchten. Die Wahl haben Sie in Wirklichkeit natürlich nur auf einem 64-Bit-System.

Installation ausführen

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

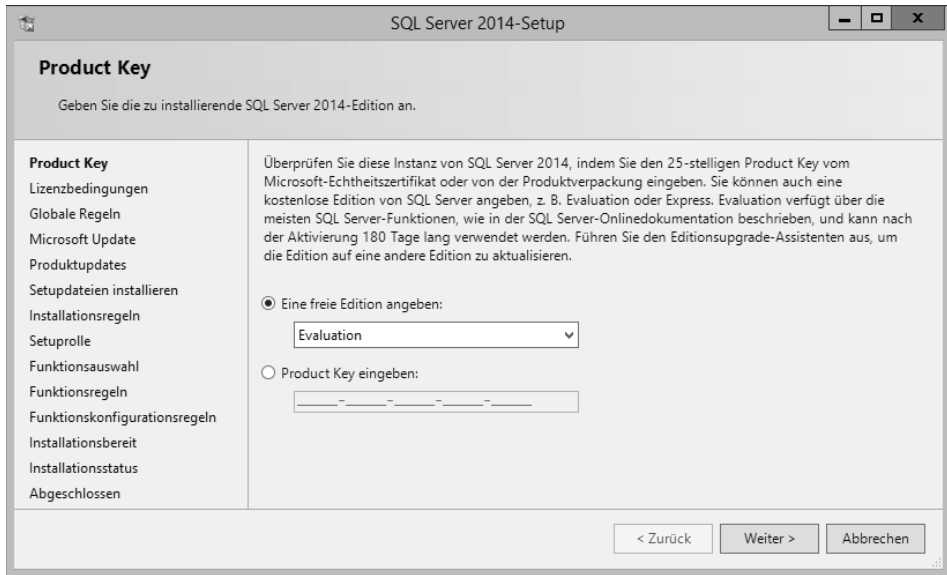


Bild 1.14 Product Key oder freie Edition

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 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. Bei der Vorversion hießen diese noch *Setupunterstützungsregeln*. 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.

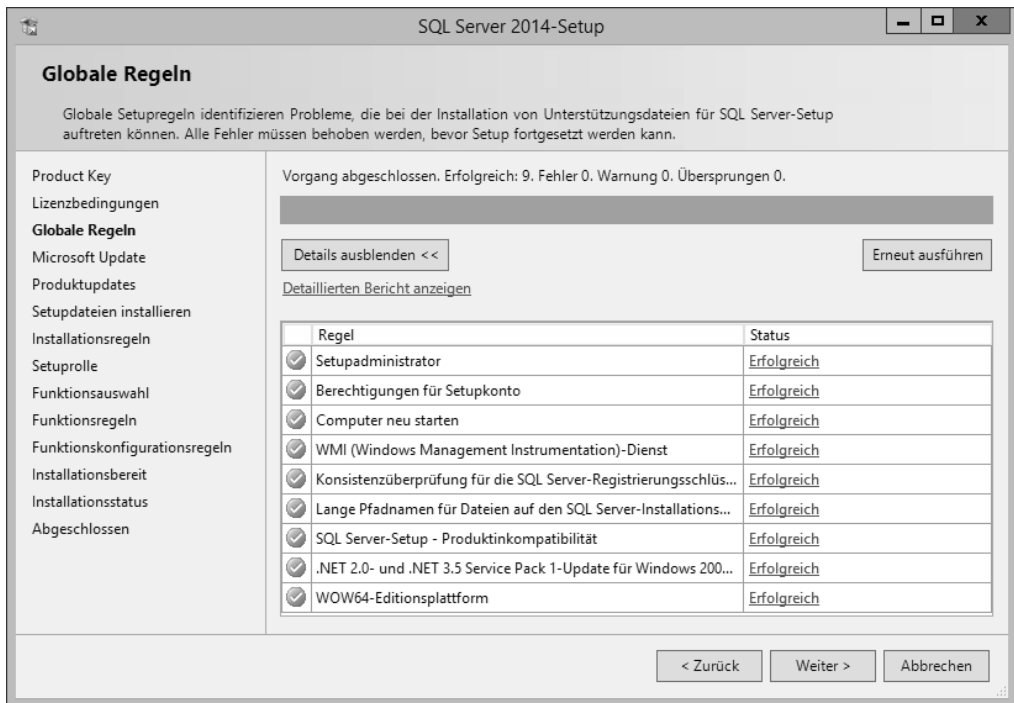


Bild 1.15 Prüfen der globalen Regeln

Im linken Dialogbereich sehen wir ja während des gesamten Vorganges den Status der Gesamtinstallation. So sehen wir, dass als Nächstes der Schritt *Microsoft Update* 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:

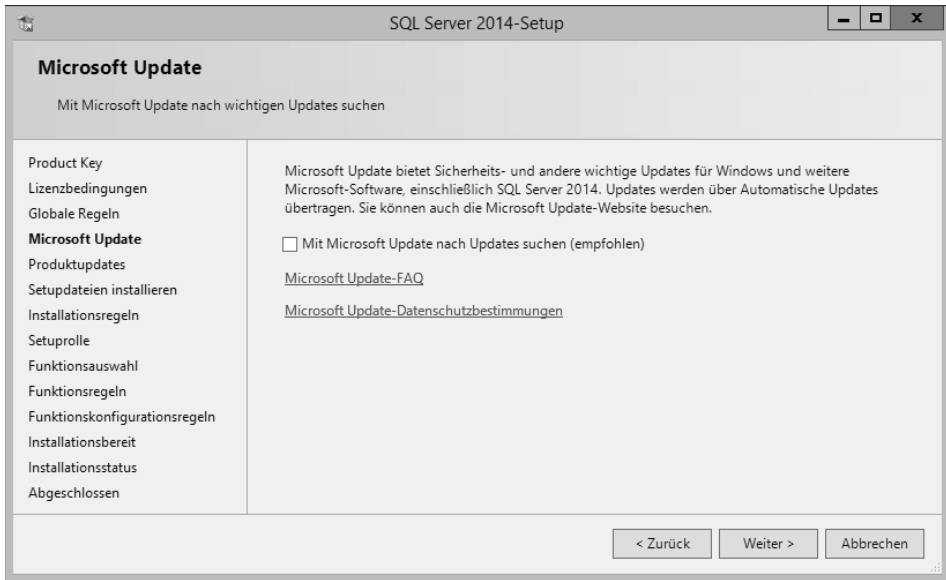


Bild 1.16 Einbeziehen vom Microsoft Update

**HINWEIS:**

- TCP Port 1433 für die Datenbank-Engine. Bei einer weiteren Instanz kann 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.

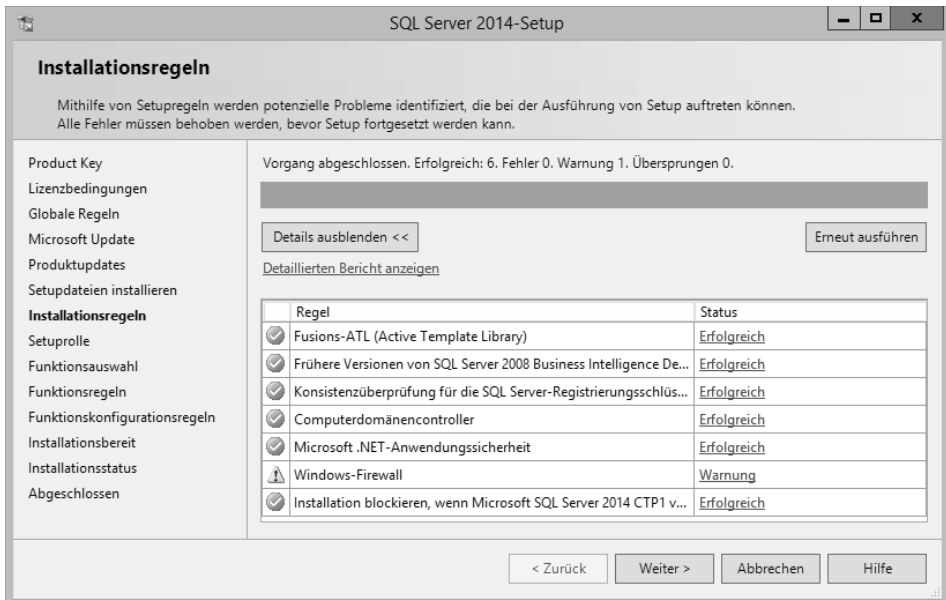


Bild 1.17 Prüfung von Installationsregeln

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.

Nachdem alle eventuellen Probleme behoben sind, erfolgt im nächsten Schritt die Auswahl der Setuprolle. Hier wählen Sie eine der drei Optionen:

- Die *SQL Server-Funktionsinstallation* wird die üblicherweise gewählte Option sein. Hier können Sie die zu installierenden Funktionen selber auswählen und Installationseinstellungen verändern.
- Mit *SQL Server PowerPivot für SharePoint* installieren Sie PowerPivot auf einen bestehenden SharePoint-Server.
- Wenn Sie eine Standardinstallation aller Features ohne „Zwischenfragen“ vornehmen wollen, wählen Sie die dritte Option *Alle Funktionen mit Standardwerten*.

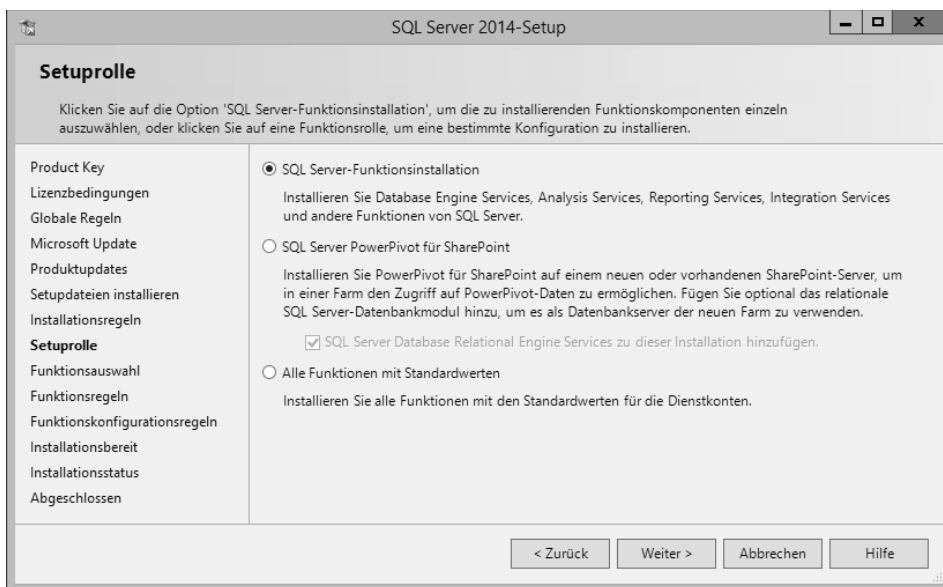


Bild 1.18 Auswahl der Setuprolle

Danach 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:
 - *Database Engine Services*: Bei der Auswahl dieser Option wird das Datenbankmodul installiert, wobei separat
 - SQL Server-Replikation,
 - Volltextsuche und
 - Data Quality Services

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 Eingabebegegnungen des jeweiligen Landes vorstellen.

- *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 SharePoint 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.
 - *Reporting Services-SharePoint, Add-In für SharePoint-Produkte*: Installiert die in SharePoint integrierte Variante des Berichtsservers. Das Add-In muss lediglich auf einem bestehenden SharePoint-Server installiert werden.
 - *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.
 - *Abwärtskompatibilität der Clienttools*: Damit können Clienttools auch für ältere SQL Server-Versionen eingesetzt werden. Hierüber bekommen Sie die auch mit dem SQL Server 2014 nicht mehr unterstützten Treiber für OLE-DB in der letzten Version für SQL Server 2012.
 - *Clienttools SDK*: Stellt das Software Development Kit mit Ressourcen für Entwickler zur Verfügung.
 - *Dokumentationskomponenten*: Installiert das Programm zur Anzeige der Online-Dokumentation. Hier wird noch nicht die Onlinedokumentation selber installiert, standardmäßig wird online darauf zugegriffen. Mit dem Hilfsbibliotheks-Manager kann diese bei Bedarf später lokal heruntergeladen werden.
 - *Verwaltungstools*: In der Variante *Einfach* werden neben dem SQL Server Management Studio noch das Befehlszeilenprogramm SQLCMD, das Distributed Replay Administration Tool und der SQL Server-PowerShell-Anbieter installiert. Das Management Studio unterstützt in dieser Auswahl nur das Datenbankmodul. Um auch Reporting Services, Analysis Services und Integration Services zu unterstützen, wählen Sie die Variante *Vollständig*. Zusätzlich erhalten Sie dann auch den SQL Server Profiler und den Datenbankoptimierungsratgeber.
 - *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.

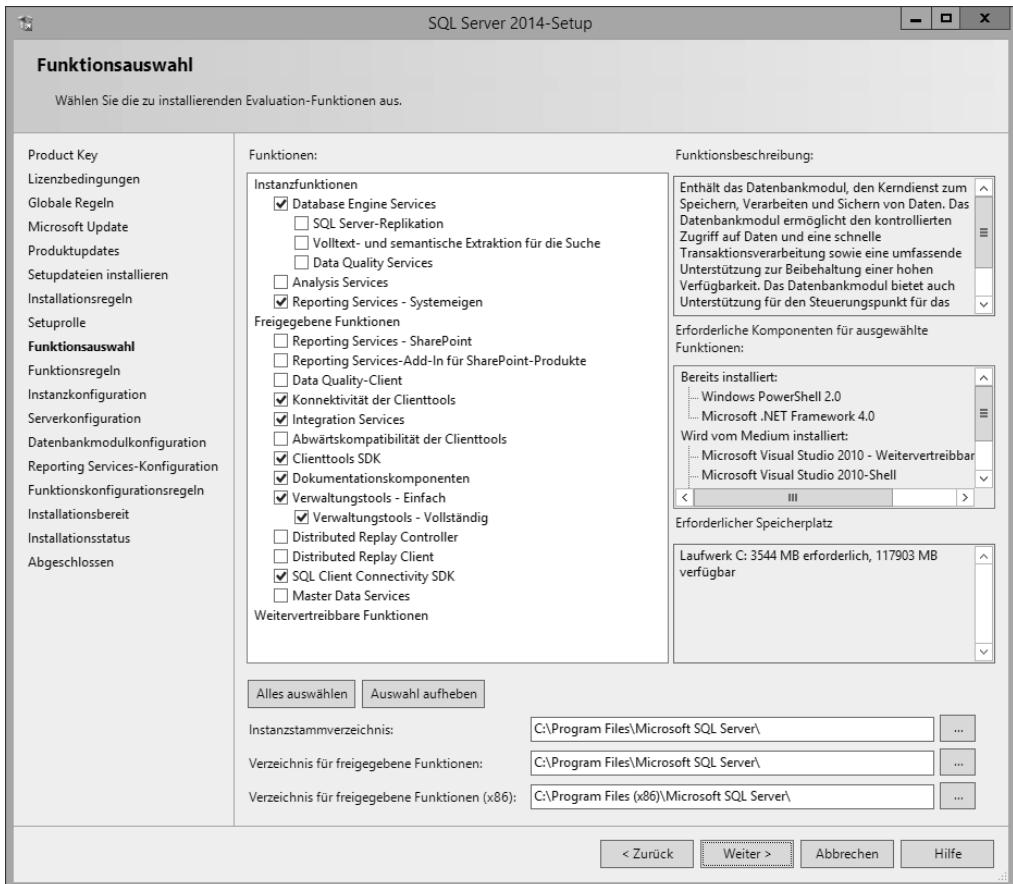


Bild 1.19 Auswahl der zu installierenden Komponenten

Für manche Features kann der Installationspfad an dieser Stelle angepasst werden. In dieser Version wird bereits 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 1.19 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. Im Beispielsfall muss das Microsoft .NET Framework 3.5 SP1 installiert werden. Bei der Installation auf Windows Server 2008 muss tatsächlich installiert werden, bei neueren Versionen ist lediglich die entsprechende Funktion zu aktivieren.

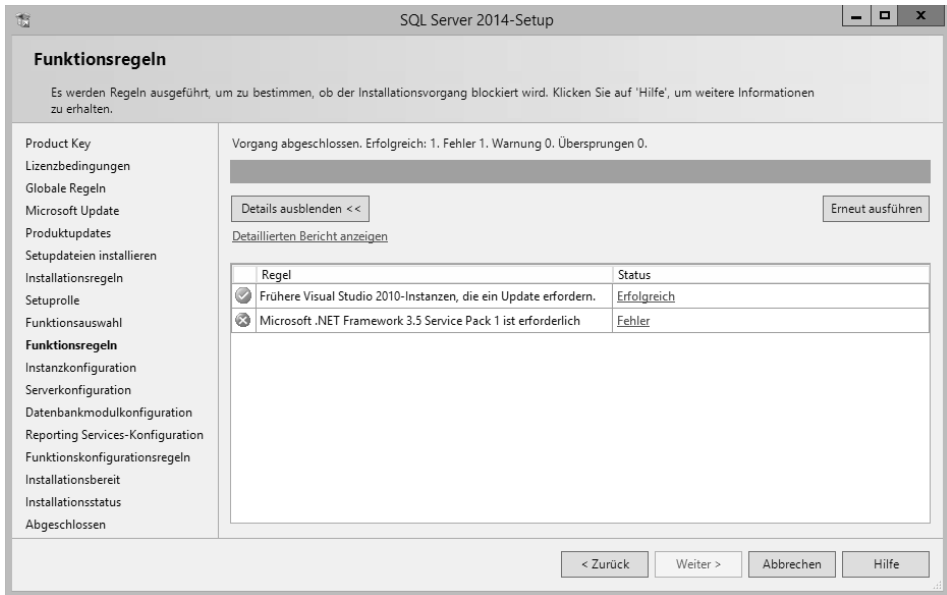


Bild 1.20 Fehler bei der Prüfung der Funktionsregeln

Beim Windows Server 2012 R2 beispielsweise verwenden Sie dazu den Server Manager und wählen unter **Verwalten** den Befehl **Rollen und Features hinzufügen** aus. Das Aktivieren dieses Features erfordert in diesem Fall zum Glück keinen Neustart. Daher kann nach dem Aktivieren des Features mit der Schaltfläche **Erneut ausführen** die Prüfung der Funktionsregeln wiederholt und das Setup, ohne es abbrechen zu müssen, an dieser Stelle fortgesetzt werden.

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.

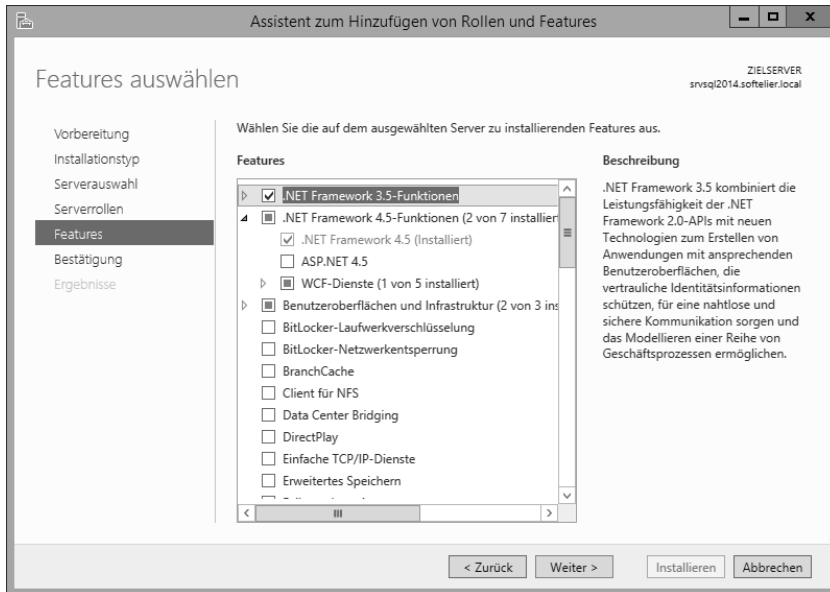


Bild 1.21 .NET Framework-Feature aktivieren

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.

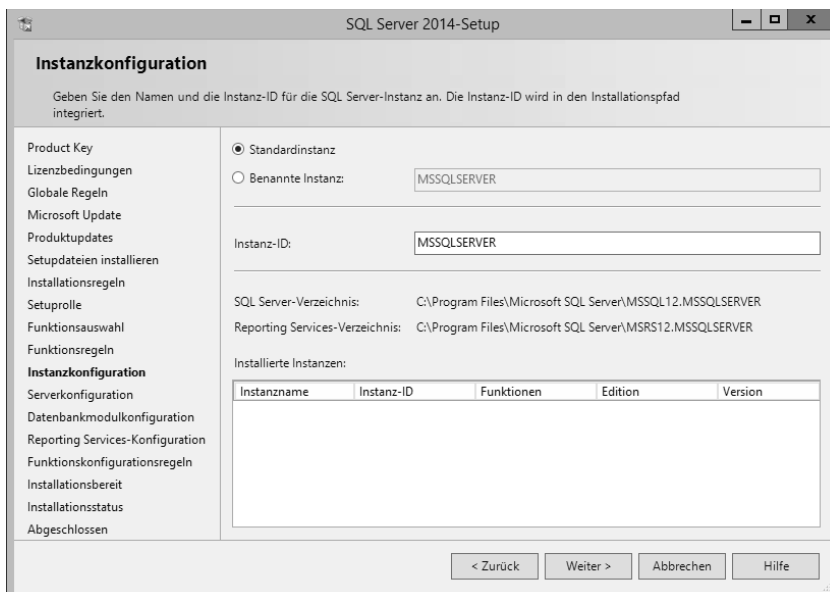


Bild 1.22 Konfiguration der Instanz

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 Dienstknoten 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.

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*.

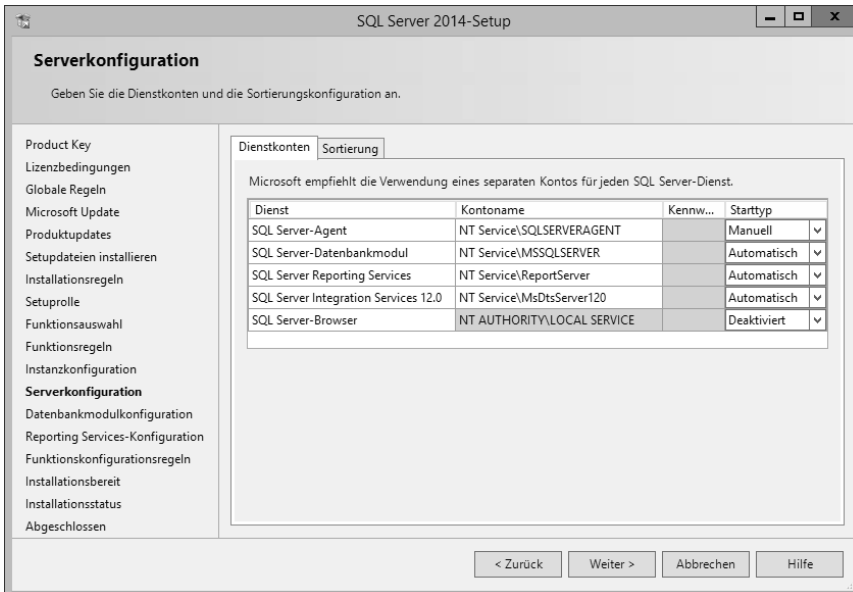


Bild 1.23 Dienstkonten auswählen

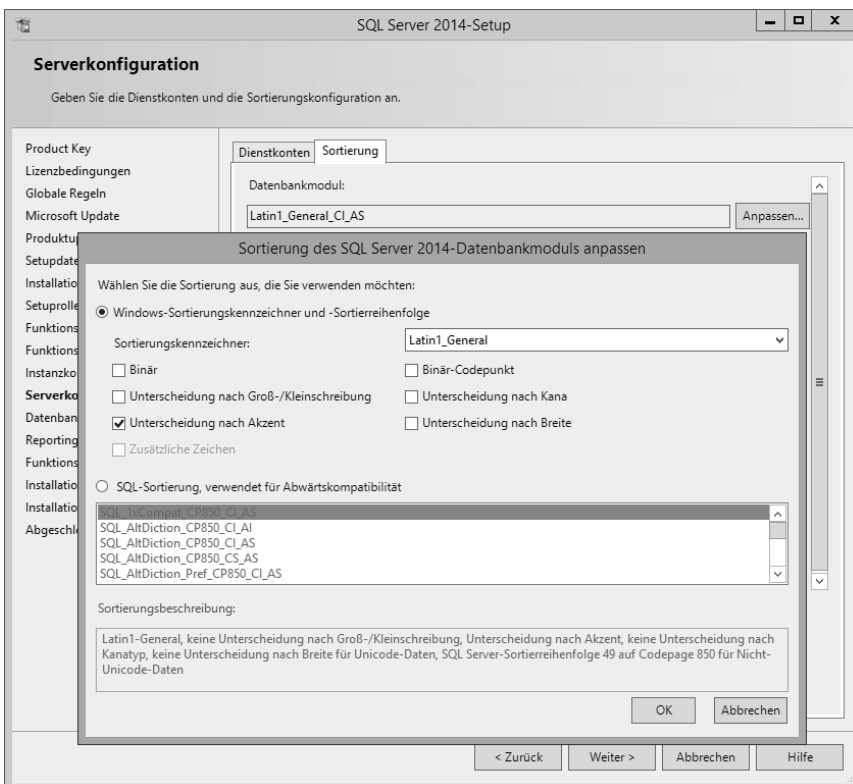


Bild 1.24 Sortierung anpassen



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 Anmeldemöglichkeiten 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 2014 ist es zum Glück an dieser Stelle nicht mehr 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.

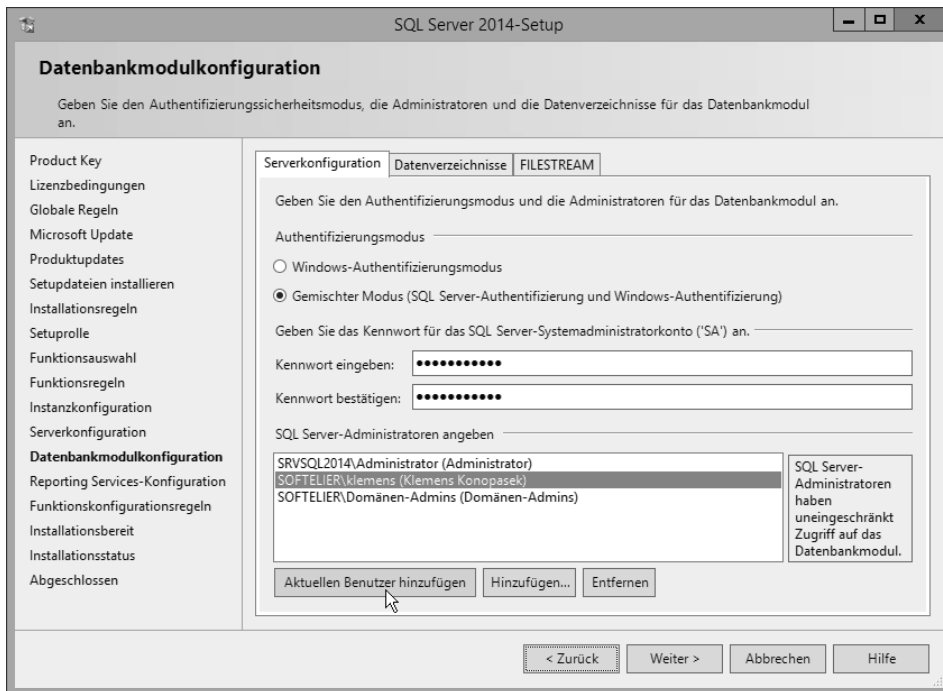


Bild 1.25 Sicherheit für SQL Server konfigurieren

Sie können die Standardverzeichnisse 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 2014 ist dies 12.
- 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 meine 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.

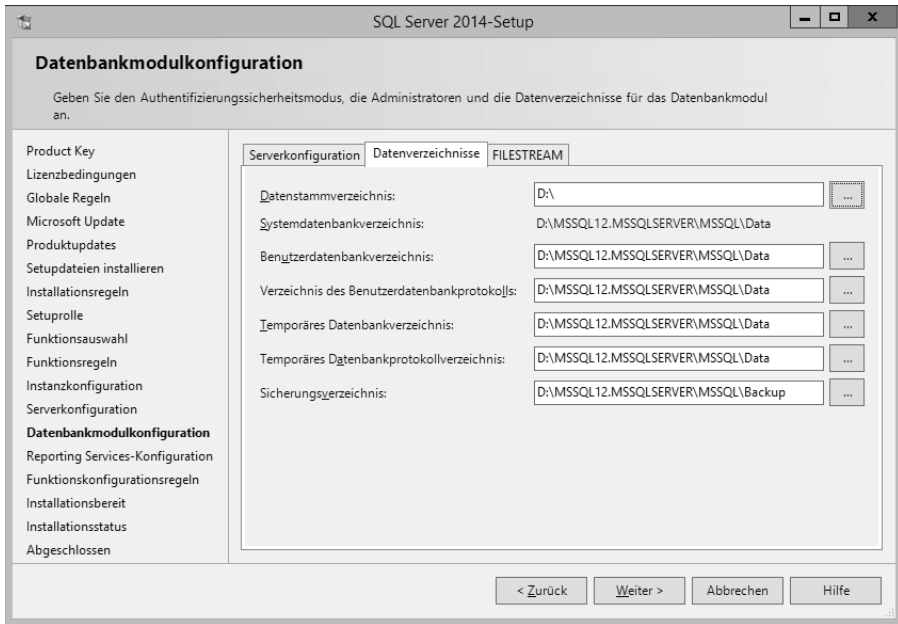


Bild 1.26 Datenverzeichnisse

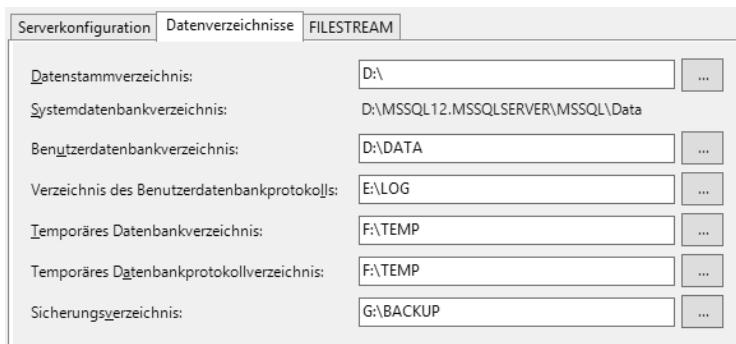


Bild 1.27
Unterschiedliche
HDD für Daten
und Protokolle

Auf dem dritten 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 Freigabe-

name 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.

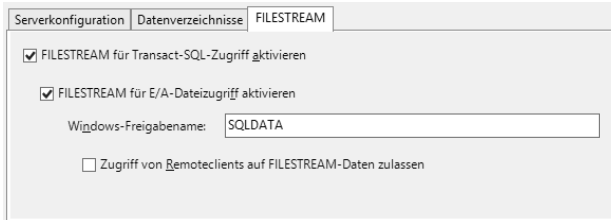


Bild 1.28
Filestream konfigurieren

Je nach Featureauswahl folgen noch weitere Dialoge; zum Beispiel jener zur Konfiguration der Reporting Services. Hier wählen Sie zum Beispiel *Einheitlicher Modus von Reporting Services* mit der Option *Installieren und konfigurieren aus*. Dies ermöglicht die sofortige Verwendung des Berichtsservers nach Abschluss der Installation.

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. Neu beim Setup des SQL Server 2014 ist, dass, wenn Prüfungen erfolgreich passiert werden, sogleich die Anzeige des Folgedialoges erfolgt.

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

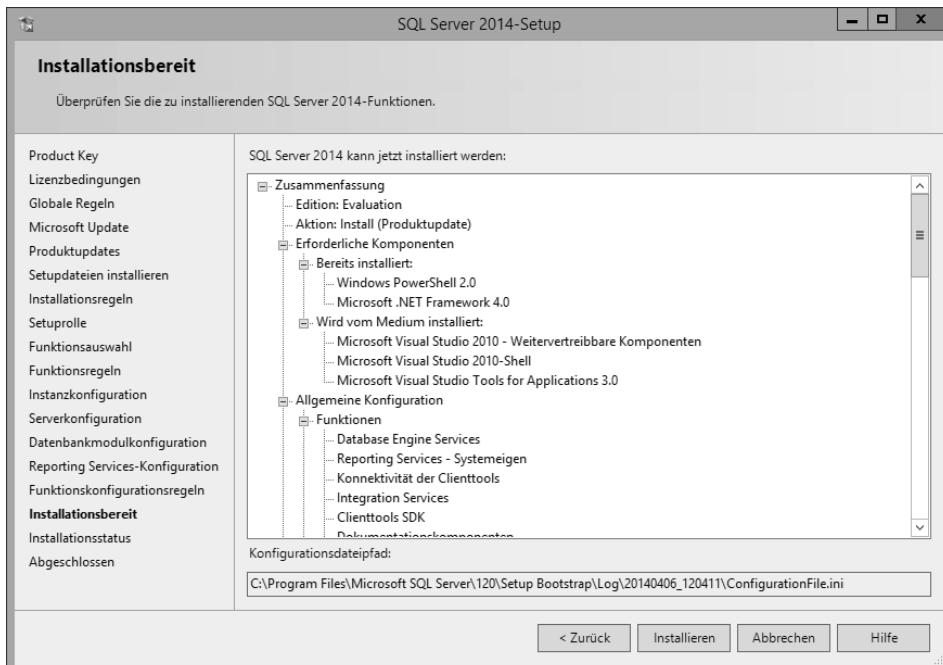


Bild 1.29 Zusammenfassung der Konfiguration



PRAXISTIPP: Starten Sie nach der Installation bitte noch Windows Update, um weitere verfügbare Updates zu installieren. Auch wenn Sie bereits beim Setup die Option zur Einbindung von Updates gewählt haben, werden zum Beispiel für das erst aktivierte .NET Framework 3.5 Updates vorhanden sein.

Ergebnis der Installation

Bei der Installation von Microsoft SQL Server 2014 erstellt das Setup-Programm die in der folgenden Tabelle gezeigten Datenbank- und Protokolldateien (= Systemdatenbanken):

Tabelle 1.4 Datenbanken nach dem Setup

Datenbank	Datenbankdatei	Protokolldatei
master	master.mdf	mastlog.ldf
model	model.mdf	modellog.ldf
msdb	msdbdata.mdf	msdblog.ldf
tempdb	tempdb.mdf	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 *MSSQL12.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*.



Die bekannte Beispieldatenbank *AdventureWorks* wird nicht mit installiert. Sie können sie aber über folgenden Link für verschiedene SQL Server-Versionen herunterladen und installieren:

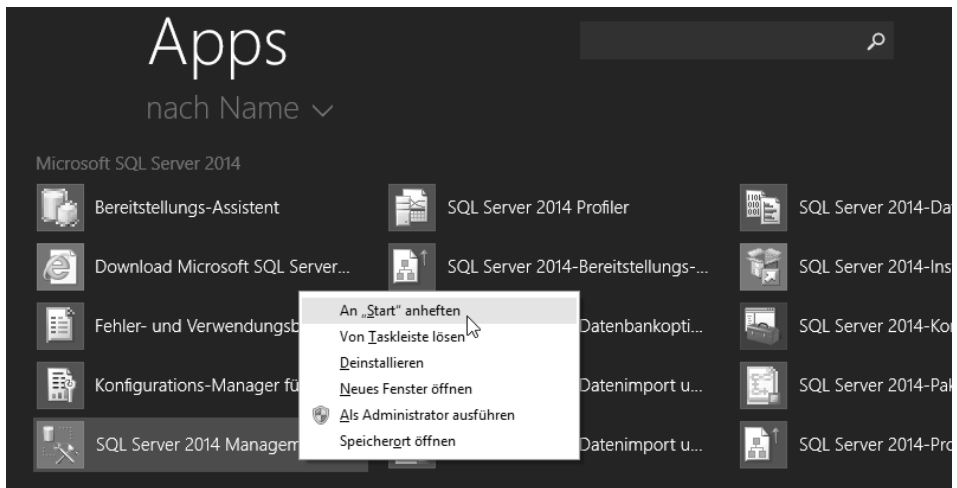
<http://msftdbprodsamples.codeplex.com/>

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 2014 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
Datenbankoptimierungsratgeber	Unterstützt beim Erstellen einer optimalen Menge von Indizes, indizierten Sichten und Partitionen.
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.

Installieren Sie den SQL Server 2014 unter Windows Server ab Version 2012 oder Windows ab Version 8, werden die Verknüpfungen zu diesen Programmen zwar im klassischen Startmenü erstellt, aber nicht automatisch auf Start angeheftet.

**Bild 1.30** Verknüpfungen im Startmenü

An Start müssen Sie die gewünschten Programme leider selber anheften, um sie von dort direkt starten zu können. Ich empfehle Ihnen, zumindest das Management Studio und den Konfigurations-Manager anzuheften.



Bild 1.31 Programme an Start angeheftet

Um nach der Installation auf diese 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.

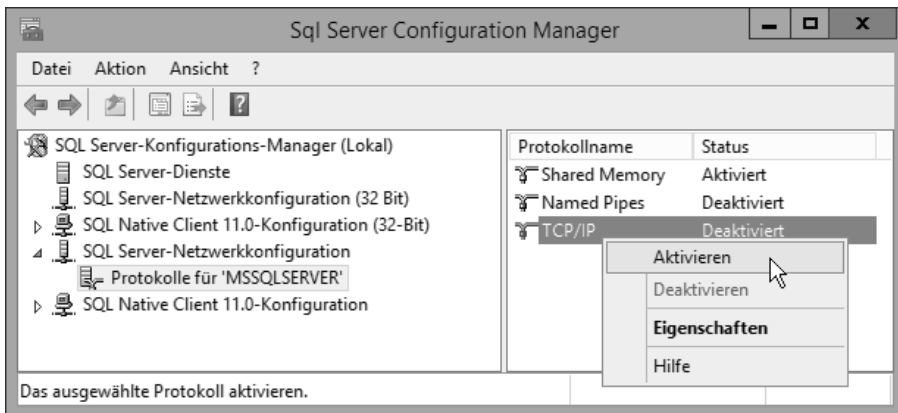


Bild 1.32 TCP/IP-Serverprotokoll aktivieren

- Da mittlerweile auf allen Host-Betriebssystemen typischerweise die Windows-Firewall aktiv ist, muss in dieser eine Ausnahme 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.

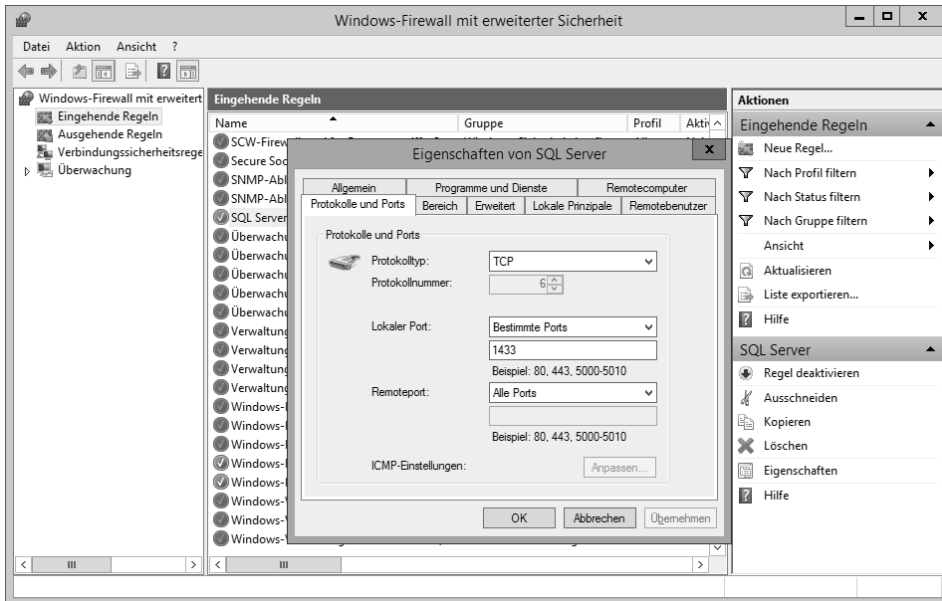


Bild 1.33 Port für SQL Server in Windows-Firewall freischalten

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 Beispielskript, die Beschreibung dazu finden Sie in Kapitel 3.
- Sie verwenden ein Backup der Datenbank, um diese damit wieder herzustellen. 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 *MSSQL12.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.

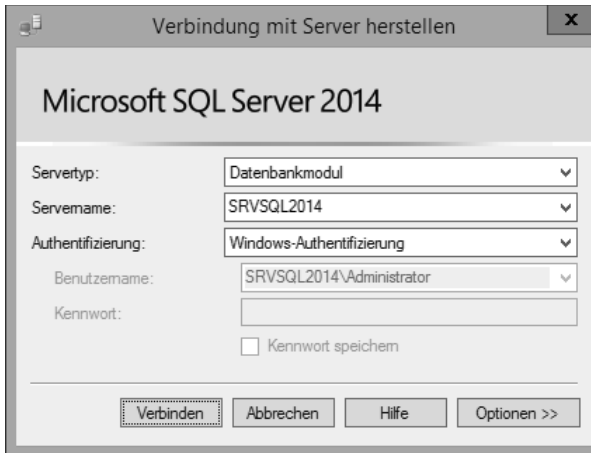


Bild 1.34
Mit Management Studio
anmelden

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.

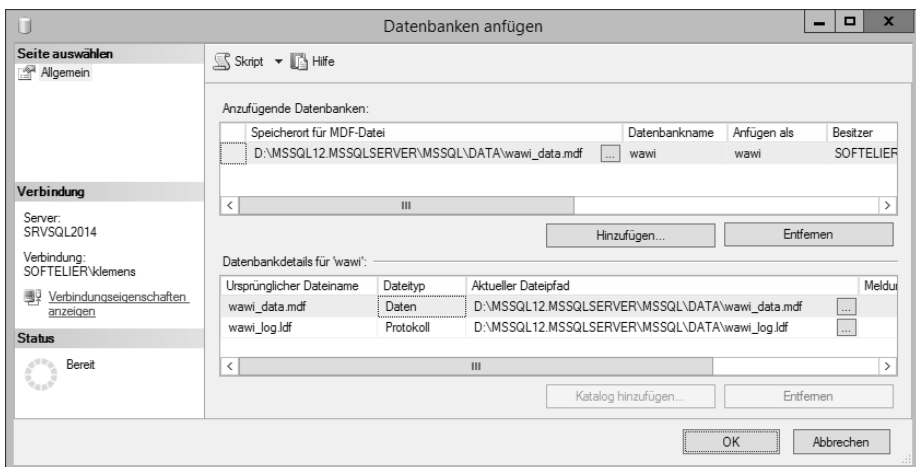


Bild 1.35 Datenbank anfügen

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

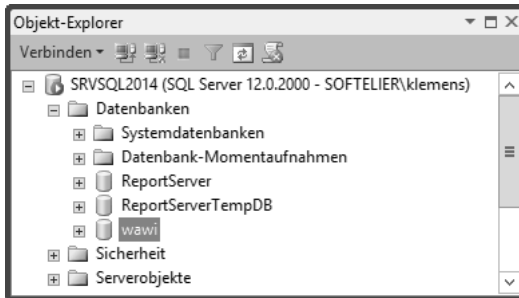


Bild 1.36
Neue Datenbank im
Objekt-Explorer

Die beiden Datenbanken *ReportServer* und *ReportServerTempDB* werden nur dann angezeigt, wenn Sie die Reporting Services bei der Installation ausgewählt haben und eine Basiskonfiguration für diese durchführen haben lassen. Diese sind bei der Express Edition nicht verfügbar.

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.

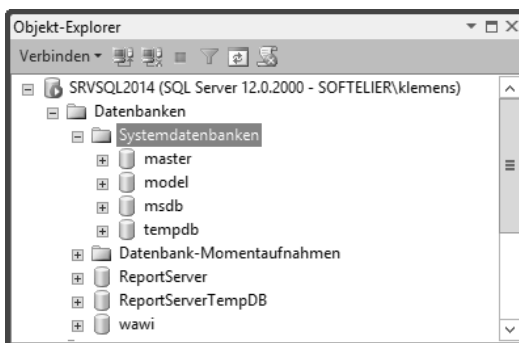


Bild 1.37
Systemdatenbanken

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 verbleibende 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.



HINWEIS: Wurde für *tempdb* nur ein kleiner Umfang definiert, muss bald nach jedem Neustart von SQL Server eventuell ein Teil der Verarbeitungslast des Systems dafür aufgewendet werden, *tempdb* automatisch in dem Umfang zu vergrößern, der zur Unterstützung der anfallenden Arbeitslast erforderlich ist. Sie können diesen zusätzlichen Aufwand vermeiden, indem Sie ALTER DATABASE oder das grafische Werkzeug des Management Studios verwenden, um die Größe von *tempdb* heraufzusetzen. Wie dies funktioniert, können Sie in Kapitel 3 lesen.

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.

Tabelle 1.6 Datenbankobjekte einer SQL Server-Datenbank (*Fortsetzung*)

Ordner	Bedeutung/Anwendung
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 Einfluss genommen 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 2014 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. Idealerweise verwendet man diese weiterhin als Frontend und verwendet dazu den SQL Server als Backend.

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 einen Prozessor und maximal 1 GB RAM.
- Mit *SQL Server Management Studio* steht das grafische Tool auch für die kleine Edition frei zur Verfügung. Es besitzt seit dem SP1 des SQL Server 2012 den vollen Funktionsumfang. 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 2014 steht unter der Adresse <http://msdn.microsoft.com/de-de/evalcenter/dn434042.aspx> in drei Varianten zur Verfügung:

- *SQL Server 2014 Express*: In dieser Download-Variante ist ausschließlich das Datenbankmodul enthalten. Sie enthält keine grafischen Client-Tools. Lediglich das Kommandozeilentool SQLCMD ist enthalten.
- *SQL Server 2014 Express mit Tools*: Das Management Studio Express ist bei dieser Variante ebenso im Download enthalten.
- *SQL Server 2014 Express mit Advanced Services*: Zusätzlich zum Management Studio Express bekommen Sie mit dieser Downloadvariante auch Unterstützung für Reporting Services und Volltextsuche.

Zusätzlich finden Sie hier auch das Management Studio Express als eigenen Download, falls Sie nur dieses benötigen, um sich mit einem bestehenden Server verbinden zu können. Auch die *LocalDB* können Sie von hier beziehen. Dabei handelt es sich um eine abgespeckte Version, die als lokale DB im Einzelbenutzermodus verwendet wird und auch direkt in eine Anwendung eingebettet werden kann.

Für den Download benötigen Sie ein Microsoft-Konto, früher als Live-ID bekannt, wie auch schon bei Windows 8, um Apps installieren zu können. Wenn Sie noch keines besitzen, können Sie eines unter <https://signup.live.com/signup.aspx?mkt=DE-DE&lc=1031&lic=1> erstellen.



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 2014 gibt es, 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 dem SQL Server 2012 hinzugekommen, aber es fehlen auch bisher bekannte wie zum Beispiel der SQL Server Native Client.



Sie finden das aktuelle Feature Pack für den SQL Server 2014 unter folgender Adresse:

<http://www.microsoft.com/de-de/download/details.aspx?id=42295>

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

- SQL Server Backup to Windows Azure Tool
- SQL Server Cloud Adapter
- SQL Server 2014 Semantic Language Statistics
- SQL Server Data-Tier Application Framework
- SQL Server 2014 Transact-SQL Language Service
- Windows PowerShell Extensions für SQL Server 2014
- SQL Server 2014 Shared Management Objects
- Microsoft Befehlszeilenprogramme 11 für Microsoft SQL Server
- ODBC Driver 11 für Microsoft SQL Server
- JDBC Driver 4.0 für Microsoft SQL Server
- SQL Server 2014 Master Data Services-Add-In für Microsoft Excel
- SQL Server StreamInsight
- SQL Server Migration Assistant für Access, MySQL, Oracle und Sybase
- SQL Server-Berichts-Generator für Microsoft SQL Server 2014

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. Hinter dem Begriff Befehlszeilenprogramme verbirgt sich das Tool SQLCMD. Für Umsteiger von Access ist vor allem der Migration Assistant eine wertvolle Hilfe. Mit diesem können Sie die ganze Struktur einer Datenbank einfach und auch die enthaltenen Daten schnell auf den SQL Server bringen.



HINWEIS: Der SQL Server Native Client, der neben den ODBC-Treibern auch jene für OLE-DB bereitgestellt hat, steht mit dem SQL Server 2014 nicht mehr zur Verfügung. Microsoft hat den OLE-DB Provider (ADO) für den SQL Server mit der Version 2012 eingestellt. Daher stehen an dieser Stelle nur mehr die ODBC-Treiber zur Verfügung. Dies hat starke Auswirkungen für Entwickler, denen Microsoft Ende der 90er-Jahre den Umstieg von ODBC auf ADO nahegelegt hat. Diese 180-Grad-Drehung hat zur Folge, dass wieder ODBC für Transaktionsanwendungen favorisiert werden. Dies ist nicht ganz verständlich, da ADO in anderen Bereichen wie zum Beispiel für Analysis Services nach wie vor unterstützt wird. In der Übergangszeit können Sie noch den Native Client des SQL Server 2012 einsetzen, aber früher oder später wird man sich von ADO in Clientapplikationen wieder in Richtung ODBC verabschieden müssen.