

Dirk Mertins
Jörg Neumann
Andreas Kühnel

Inklusive Migration von
SQL Server 2014



```
SELECT cust.CustomerID  
cont.FirstName + ' ' +  
func.ProductLine,  
Total  
FROM Sales.Customer cu  
Person.Contact cont ON  
APPLY  
GetOrderTotalsOnProduc
```



SQL Server 2016

Das Handbuch für Entwickler

- ▶ Installation, Migration, Datenbankmodellierung
- ▶ T-SQL, .NET-Programmierung, XML und Webservices
- ▶ Einsatz als Programmierplattform und Datenbankmanagement-Server

7., aktualisierte Auflage

 Rheinwerk
Computing

- 34.4.4 Skalare Funktionen
- 34.4.5 Eine Skalarfunktion erstellen
- 34.4.6 Tabellenwertige Funktionen
- 34.4.7 Unterschiede zwischen T-SQL- und CLR-Funktionen
- 34.4.8 Eine tabellenwertige Funktion erstellen

34.5 Serverseitiger Datenzugriff mit ADO.NET

- 34.5.1 Der Kontext
- 34.5.2 Verbindung zur Datenbank herstellen

34.6 Pipes

- 34.6.1 Die »Send«-Methode
- 34.6.2 Die »ExecuteAndSend()«-Methode
- 34.6.3 Ausgabestrom steuern

34.7 Impersonalisierung

34.8 Benutzerdefinierte Typen

- 34.8.1 »Das SqlUserDefinedType«-Attribut
- 34.8.2 Einen benutzerdefinierten Datentyp erstellen
- 34.8.3 Das »INullable«-Interface
- 34.8.4 Die »ToString()«-Methode
- 34.8.5 »Die Parse()«-Methode
- 34.8.6 Erstellung des »CalendarWeek«-Datentyps
- 34.8.7 Den »CalendarWeek«-Typ installieren
- 34.8.8 Den »CalendarWeek-Typ« testen
- 34.8.9 Validierungen
- 34.8.10 Methoden implementieren
- 34.8.11 Serialisierung
- 34.8.12 Typen als Klassen abbilden
- 34.8.13 Zugriff vom Client

34.9 Benutzerdefinierte Aggregate

- 34.9.1 Das »SqlUserDefinedAggregate«-Attribut
- 34.9.2 Die »Init()«-Methode
- 34.9.3 Die »Accumulate()«-Methode
- 34.9.4 Die »Merge()«-Methode
- 34.9.5 Die »Terminate()«-Methode
- 34.9.6 Ein Beispielaggregat
- 34.9.7 Das »MaxRange«-Aggregat implementieren

34.9.8 Hinweise zur Implementierung

34.9.9 Aggregate installieren

34.10 Benutzerdefinierte Trigger

34.10.1 Das »SqlTrigger«-Attribut

34.10.2 Die »TriggerContext«-Klasse

34.10.3 Trigger implementieren

34.10.4 Hinweise zur Implementierung

34.10.5 Trigger installieren

34.11 Administration und Monitoring von CLR-Objekten

34.11.1 Informationen über die installierten Assemblies ermitteln

34.11.2 Laufzeitinformationen von CLR-Objekten ermitteln

35 T-SQL: erweiterte Themen

35.1 Kompatibilität festlegen

35.2 Änderungen der Kompatibilität ab SQL Server 2008

35.3 Änderungen bei nicht ANSI-konformen Abfragen

35.4 »CROSS APPLY« und »OUTER APPLY«

35.5 Kreuztabellen mit »PIVOT« erstellen

35.5.1 Entscheidungstabellen mit »PIVOT« erstellen

35.5.2 Kreuztabellen in flache Listen transformieren

35.6 Common Table Expressions

35.6.1 Verarbeitung von hierarchischen Daten

35.6.2 Maximale Rekursionsstufe einstellen

35.7 Fehlerbehandlung

35.7.1 Informationen über den Fehler ermitteln

35.7.2 Eigene Fehler auslösen

35.7.3 Verschachtelte Fehlerbehandlung

35.8 Ranking- und Windowing-Funktionen

35.8.1 Ergebnisse mit »RANK()« gewichten

35.8.2 Ranking ohne Lücken mit »DENSE_RANK()«

35.8.3 Gruppierte Ranglisten mit Windowing

- 35.8.4 Zeilen mit »ROW_NUMBER()« nummerieren
- 35.8.5 Paging mit »ROW_NUMBER()«
- 35.8.6 Daten mit »NTILE()« partitionieren

35.9 Die »OUTPUT«-Klausel

35.10 Die »TABLESAMPLE«-Klausel

35.11 »EXCEPT«- und »INTERSECT«-Statement

35.12 Tabellenwertparameter

- 35.12.1 Tabellenwertparameter definieren
- 35.12.2 Tabellenwertparameter verwenden
- 35.12.3 Prozeduren mit Tabellenwertparametern aufrufen
- 35.12.4 Aufruf vom Client mit ADO.NET
- 35.12.5 Limitationen

36 Einsatz von XML und JSON in der Datenbank

36.1 Warum XML?

36.2 HTML und XML

- 36.2.1 Auszeichnungssprachen
- 36.2.2 Auszeichnungen

36.3 XML-Fähigkeiten von SQL Server

36.4 Der XML-Datentyp

- 36.4.1 Vor- und Nachteile der XML-Datenspeicherung
- 36.4.2 XML-Felder anlegen
- 36.4.3 XML-Felder mit einem Schema verknüpfen

36.5 Auf gespeicherte XML-Daten zugreifen

- 36.5.1 Eine kurze Einführung in XPath
- 36.5.2 XQuery als Abfragesprache
- 36.5.3 Die Methoden des XML-Datentyps
- 36.5.4 Indizierung von XML-Feldern

36.6 Darstellung von Abfrageergebnissen im XML-Format

- 36.6.1 Der »RAW«-Modus
- 36.6.2 Allgemeine Optionen der »FOR XML«-Klausel

- 36.6.3 Der »AUTO«-Modus
- 36.6.4 Der »PATH«-Modus
- 36.6.5 Der »EXPLICIT«-Modus

36.7 XML-Schema ermitteln

36.8 XML-Daten in einem relationalen Modell speichern

- 36.8.1 Ein XML-Dokument in eine Tabelle importieren

36.9 Bereitstellung von Daten im JSON-Format

- 36.9.1 Eine kurze Beschreibung des JSON-Formats
- 36.9.2 Die »FOR JSON«-Direktive
- 36.9.3 Der »AUTO«-Modus
- 36.9.4 Der »PATH«-Modus
- 36.9.5 JSON-Daten mit »OPENJSON« konvertieren
- 36.9.6 Weitere JSON-Funktionen in SQL Server 2016

37 Datenbankzugriff mit ADO.NET

37.1 Einleitung

- 37.1.1 Projekte anlegen und speichern
- 37.1.2 Die Beispieldatenbank

37.2 Die Datenprovider

37.3 Die Verbindung zu einer Datenbank herstellen

- 37.3.1 Die Verbindungszeichenfolge
- 37.3.2 Die Verbindung mit einer bestimmten SQL-Server-Instanz aufbauen
- 37.3.3 Öffnen und Schließen einer Verbindung
- 37.3.4 Das Verbindungspooling
- 37.3.5 Die Ereignisse eines »Connection«-Objekts
- 37.3.6 Verbindungszeichenfolgen aus einer Konfigurationsdatei abrufen
- 37.3.7 Die Klasse »SqlConnection« im Überblick
- 37.3.8 Verbindungen mit dem OleDb-Datenprovider

37.4 Die Datenbankabfrage

- 37.4.1 Das »SqlCommand«-Objekt
- 37.4.2 Abfragen, die genau ein Ergebnis liefern

37.5 Das »SqlDataReader«-Objekt

- 37.5.1 Datensätze einlesen
- 37.5.2 Schließen des »SqlDataReader«-Objekts
- 37.5.3 MARS (Multiple Active Resultsets)
- 37.5.4 Batchabfragen mit »NextResult« durchlaufen
- 37.5.5 Das Schema eines »SqlDataReader«-Objekts untersuchen

37.6 Parametrisierte Abfragen

- 37.6.1 Parametrisierte Abfragen mit dem SqlClient-Datenprovider
- 37.6.2 Die Klasse »SqlParameter«
- 37.6.3 Asynchrone Abfragen
- 37.6.4 Gespeicherte Prozeduren (Stored Procedures)

37.7 Der »SqlDataAdapter«

- 37.7.1 Was ist ein »DataAdapter«?
- 37.7.2 Die Konstruktoren der Klasse »DataAdapter«
- 37.7.3 Die Eigenschaft »SelectCommand«
- 37.7.4 Den lokalen Datenspeicher mit »Fill« füllen
- 37.7.5 Tabellenzuordnung mit »TableMappings«
- 37.7.6 Das Ereignis »FillError« des »SqlDataAdapters«

37.8 Daten im lokalen Speicher – das »DataSet«

- 37.8.1 Verwenden des »DataSet«-Objekts
- 37.8.2 Dateninformationen in eine XML-Datei schreiben
- 37.8.3 Dem »DataSet« Schemainformationen übergeben
- 37.8.4 Schemainformationen bereitstellen
- 37.8.5 Eigenschaften einer »DataColumn«, die der Gültigkeitsprüfung dienen
- 37.8.6 Die Constraints-Klassen einer «DataTable«
- 37.8.7 Das Schema mit Programmcode erzeugen
- 37.8.8 Schemainformationen mit »SqlDataAdapter« abrufen
- 37.8.9 Änderungen in einer »DataTable« vornehmen
- 37.8.10 Was bei einer Änderung einer Datenzeile passiert
- 37.8.11 Manuelles Steuern der Eigenschaft »DataRowState«

37.9 Mit mehreren Tabellen arbeiten

- 37.9.1 Der Weg über »JOIN«-Abfragen
- 37.9.2 Mehrere Tabellen in einem »DataSet«
- 37.9.3 Eine »DataRelation« erzeugen
- 37.9.4 »DataRelations« und Einschränkungen
- 37.9.5 In Beziehung stehende Daten suchen
- 37.9.6 Ergänzung zum Speichern von Schemainformationen in einer XML-