

Das SQL Server 2005- Entwicklerbuch

Microsoft®
.net
Development
Series

Bob Beauchemin
Dan Sullivan





Inhaltsverzeichnis

Vorwort	19
1 Einführung	35
1.1 Das .NET Framework und die Microsoft-Plattform	35
1.2 Der Einfluss des .NET Framework auf SQL Server	37
1.3 Der Standard SQL:1999 – Erweiterung des relationalen Modells	42
1.4 Benutzerdefinierte Typen und SQL Server	45
1.5 XML – Daten- und Dokumentspeicherung	48
1.6 Web Services – XML als Marshalling-Format	53
1.7 Clientzugriff – ... und dann gibt es noch Clients	55
1.7.1 Clientseitige Datenbank-APIs und SQL Server 2005	55
1.7.2 Clientseitige XML-basierte APIs und SQL Server 2005-Integration	56
1.8 SQL Server erweitern – Service Broker und Benachrichtigungsdienste	57
1.9 Zusammenfassung	59
2 Die Laufzeit hosten – SQL Server als Laufzeithost	61
2.1 Wozu muss man wissen, wie Hosting funktioniert?	61
2.2 Was ist ein .NET Framework-Laufzeithost?	63
2.3 SQL Server als Laufzeithost	65
2.3.1 SQL Server-Ressourcenverwaltung	66
2.3.2 Behandlung von Ausnahmebedingungen	68
2.3.3 Laden von Code	71
2.3.4 Sicherheit	73
2.4 Die Laufzeit laden – Prozesse und Anwendungsdomänen	74

2.5	Sicherer Code – wie die Laufzeit »fremden« Code sicherer ausführt	77
2.6	Wo der Code verwaltet wird – .NET Framework-Assemblys speichern (CREATE ASSEMBLY)	79
2.7	Assembly-Abhängigkeiten – wenn Assemblys andere Assemblys verwenden	84
2.8	Assemblys und SQL-Schemas – wer Assemblys besitzt (Informationsschema)	85
2.8.1	Metadaten-Systemtabellen und INFORMATION_SCHEMA	85
2.8.2	Assembly-Metadaten	86
2.9	Benutzer-Assemblys verwalten (ALTER ASSEMBLY, DROP ASSEMBLY)	88
2.10	Kompatibilität mit der Spezifikation	91
2.11	Schlussfolgerungen	92
2.12	Zusammenfassung	93
3	Prozeduren und Funktionen in .NET-CLR-Sprachen	95
3.1	SQL Server erweitern	95
3.2	Grundlagen der CLR-Erweiterung	97
3.3	Details zu den CLR-Erweiterungen	108
3.4	System.Data.SqlTypes	110
3.4.1	Ein paar Worte zu Nullwerten	111
3.4.2	SqlTypes verwenden	113
3.5	Parameter und Rückgabewert	126
3.6	Benutzerdefinierte Funktionen	126
3.7	Tabellenwertfunktionen	133
3.8	Gespeicherte Prozeduren	140
3.9	Trigger	142
3.10	Zusammenfassung	143
4	Prozessinterner Datenzugriff	145
4.1	Mit dem SqlConnection programmieren	145
4.2	Kontext – die Klasse SqlConnection	149
4.3	Verbindungen	152
4.4	Befehle – Dinge in Gang bringen	154
4.5	Ergebnisse erhalten	158

4.6	Transaktionen	161
4.6.1	Interessantes bei TransactionScope	165
4.6.2	Empfehlungen	166
4.7	Pipe	168
4.8	Neue Resultsets erzeugen und senden	170
4.9	WindowsIdentity verwenden	172
4.10	Einen Web Service von SQLCLR aufrufen	173
4.11	Ausnahmebehandlung	176
4.12	SqlTriggerContext	181
4.13	SqlClient-Klassen, die Sie nicht auf dem Server verwenden können	182
4.14	Zusammenfassung	183
5	Benutzerdefinierte Typen und Aggregate	185
5.1	Warum brauchen wir benutzerdefinierte Typen?	185
5.2	Benutzerdefinierte Typen im Überblick	186
5.3	Einen benutzerdefinierten Typ erzeugen	188
5.3.1	Nullwertimplementierung	192
5.3.2	Zeichenfolgenimplementierung	193
5.3.3	Binärimplementierung	197
5.3.4	IBinarySerialize.Read/Write	199
5.3.5	Benutzerdefinierte Typen erzeugen	201
5.3.6	Öffentliche Eigenschaften, Felder und Methoden	204
5.3.7	Hilfsfunktionen	213
5.3.8	Gültigkeitsprüfung von benutzerdefinierten Typen	215
5.3.9	Definitionen von benutzerdefinierten Typen verwalten	217
5.3.10	Benutzerdefinierte Typen und XML	218
5.3.11	Sollten Objekte als benutzerdefinierte Typen dargestellt werden?	225
5.3.12	Benutzerdefinierte Aggregate	237
5.3.13	Ein benutzerdefiniertes Aggregat implementieren	240
5.3.14	Eine benutzerdefinierte Aggregatfunktion erstellen	245
5.3.15	Format.Native und Format.UserDefined	245
5.4	Zusammenfassung	247
6	Sicherheit	249
6.1	Neue Sicherheitsmerkmale in SQL Server 2005	249
6.2	Optionale Features sind standardmäßig deaktiviert	251
6.2.1	Sicherheitseinstellungen konfigurieren	252
6.2.2	Sicherheit und Metadaten	254

6.3	SQL Server-Sicherheitskonzepte mit Erweiterungen – kurz und knapp	255
6.3.1	Authentifizierung und Autorisierung – Prinzipale und Berechtigungen	255
6.3.2	Ausführungskontext und Besitzverkettung	262
6.4	Kennwortrichtlinien und Anmeldeinformationen von SQL Server	265
6.5	Verschlüsselungsschlüssel und integrierte Verschlüsselungsfunktionen	268
6.6	Verschlüsselungsfunktionen	272
6.7	Trennung von Benutzern und Schemas	278
6.8	Synonyme	282
6.9	Den Ausführungskontext für prozeduralen Code spezifizieren	283
6.10	Codesignierung	288
6.11	SQL Server-Berechtigungen und SQLCLR-Objekte	290
6.12	Assembly-Berechtigungen: Wer kann eine Assembly katalogisieren und verwenden?	291
6.13	Was .NET Framework-Code aus SQL Server heraus tun kann: Sicherheitsstufen	297
6.13.1	Einführung in die Codezugriffssicherheit	297
6.13.2	Codezugriffssicherheit und .NET Framework-Assemblies	298
6.14	Zusammenfassung	302
7	Erweiterungen des SQL-Moduls	305
7.1	Verbesserungen am SQL-Modul	305
7.2	Snapshot-Isolation	306
7.2.1	Nachteile der Versionsverwaltung	313
7.2.2	Versionsverwaltung überwachen	315
7.3	DDL-Trigger	316
7.4	Ereignisbenachrichtigungen	321
7.5	Datentypen für große Werte	324
7.6	Daten mit dem neuen BULK-Provider laden	327
7.7	Neukompilierung auf Anweisungsebene	328
7.8	Abfragehinweise, Planhinweislisten und die Erzwingung eines Plans	330
7.9	Zusammenfassung	335

8	Erweiterungen der Sprache T-SQL	337
8.1	Verbesserungen an Transact-SQL	337
8.2	Fehlerbehandlung	338
8.3	INTERSECT und EXCEPT	351
8.4	TOP	355
8.5	ON DELETE und ON UPDATE	358
8.6	OUTPUT	361
8.7	APPLY-Operatoren	365
8.8	Allgemeine Tabellenausdrücke	368
8.9	Rekursive Abfragen	375
8.10	Die Operatoren PIVOT und UNPIVOT	383
8.10.1	PIVOT	383
8.10.2	UNPIVOT	389
8.11	Rangfolgefunktionen und Partitionierung	391
8.11.1	Die Funktion ROW_NUMBER	392
8.11.2	RANK	394
8.11.3	Die Funktion DENSE_RANK	394
8.11.4	Die Funktion NTILE	395
8.11.5	PARTITION BY	397
8.11.6	Partitionen aggregieren	398
8.12	TABLESAMPLE	401
8.13	Zusammenfassung	405
9	XML in der Datenbank – der XML-Datentyp	407
9.1	Der Datentyp XML	407
9.2	Den XML-Datentyp in Tabellen verwenden	408
9.3	Variablen und Parameter für XML-Daten verwenden	414
9.4	Typisiertes und nicht typisiertes XML – Katalogisieren mit XML-Schemaauflistungen	415
9.4.1	Die XML-Schemaauflistungen von SQL Server	417
9.4.2	Typisiertes XML	418
9.4.3	XML-Schemas und -Schemaauflistungen verwalten	420
9.4.4	Sicherheit für XML-Schemaauflistungen und stark typisierte Instanzen	423
9.5	Einen Index auf einer XML-Spalte erzeugen	423
9.6	Funktionen des XML-Typs	425

9.7	SELECT . . . FOR XML-Erweiterungen	426
9.7.1	FOR XML PATH-Modus	428
9.7.2	Einen XML-Datentyp erzeugen	430
9.7.3	Ein Inline-XSD-Formatschema erzeugen	431
9.7.4	NULL-Datenbankwerte	433
9.7.5	Ein Stammelement erzeugen	434
9.7.6	Andere Features	434
9.8	SQL- und XML-Datentypen zuordnen	436
9.8.1	Zuordnung von SQL-Typen zu XML-Typen	436
9.8.2	Zeichenfolgen-, binäre und dezimale Typen	437
9.8.3	Andere allgemeine Datentypen	438
9.8.4	Datumsdatentypen	438
9.8.5	Pattern-basierte Datentypen	439
9.8.6	Platzhalterdatentypen	439
9.8.7	Nullfähigkeit	439
9.9	OpenXML-Erweiterungen	440
9.10	XML aus Dateien in die Datenbank laden	441
9.11	Kompatibilität zum ANSI SQL-Standard	443
9.11.1	Der XML-Datentyp	443
9.11.2	Kataloge und Tabellen von SQL auf XML abbilden	444
9.11.3	SQL-Datentypen auf XML-Datentypen abbilden	446
9.12	Zusammenfassung	447
10	XML-Abfragesprachen – XQuery und XPath	449
10.1	Was ist XQuery?	449
10.2	Eine Einführung in XQuery	451
10.2.1	Der XQuery-Prolog	455
10.2.2	Der XQuery-Körper	456
10.3	XQuery und SQL – Gemeinsamkeiten und Unterschiede	465
10.4	XQuery mit dem XML-Datentyp verwenden	468
10.4.1	Die Funktion xml.exist(string xquery-text)	469
10.4.2	Die Funktion xml.value(string xquery-text, string SQLType)	471
10.4.3	Die Funktion xml.query(string xquery-text)	472
10.4.4	Die Funktion xml.nodes(string xquery-text)	476
10.4.5	Standardfunktionen und -operatoren von XQuery	479
10.4.6	XQuery-Funktionen und -operatoren von SQL Server	481
10.4.7	Mehrfachdokumentabfragen in SQL Server XQuery	486
10.5	XML DML – XML-Spalten aktualisieren	486
10.5.1	Die Funktion xml.modify('insert . . .')	488
10.5.2	Die Funktion xml.modify('delete . . .')	491
10.5.3	xml.modify('replace value of . . .')	491
10.5.4	Allgemeine Schlussfolgerungen und Empfehlungen	492

10.6	Besonderheiten bei Verwendung von XQuery innerhalb von SQL Server	493
10.6.1	XML-Schemas und XQuery von SQL Server 2005	493
10.6.2	Verwendung von XML-Indizes in XQuery	494
10.7	Zusammenfassung	497
11	SQL Server Service Broker	499
11.1	Messaging-Anwendungen	499
11.2	Nachrichten verarbeiten	511
11.3	Geschäftstransaktionen	517
11.3.1	Dienstprogramme	521
11.3.2	Konversationen	526
11.3.3	Konversationsgruppen	533
11.3.4	Nachrichtentypen	541
11.3.5	Verträge	545
11.3.6	Senden und Nachrichtentyp	548
11.3.7	Schleifenverarbeitung	551
11.3.8	Beschädigte Nachrichten	556
11.3.9	Kompensation	558
11.3.10	Verteilung	561
11.4	Zusammenfassung	576
12	SQL Server als Plattform für Web Services	579
12.1	Datenbanken und Web Services zusammenbringen	579
12.1.1	HTTP-Endpunktdeklaration	582
12.1.2	Die Anweisung CREATE ENDPOINT	587
12.1.3	Endpunkt-URL	587
12.1.4	Endpunktzustand	589
12.1.5	Endpunkt-Metadaten	590
12.1.6	Webmethoden	590
12.1.7	WSDL – Web Service Definition Language	595
12.1.8	XML-Namespaces	619
12.1.9	WSDL-Generierung	620
12.1.10	Gespeicherte Prozedur in Web Service	629
12.1.11	SQL-Batches	636
12.1.12	Weitere Features	640
12.1.13	SQLXML 4.0-Funktionalität und SQL Server 2005	641
12.2	Zusammenfassung	653
13	SQL Server 2005 und Clients	655
13.1	SQL Native Client	655
13.2	Neue Datentypen und der Datentypkompatibilitätsmodus	659

13.3	Benutzerdefinierte Typen und APIs für den Zugriff auf relationale Daten	660
13.4	Benutzerdefinierte .NET Framework-Typen in ADO.NET	661
13.5	UDT-Daten von einem DataReader abrufen	662
13.6	.NET Framework-UDTs in ODBC, OLE DB und ADO Clients verwenden	668
13.7	Den XML-Datentyp auf dem Client unterstützen	670
13.7.1	Den XML-Datentyp in ADO.NET verwenden	671
13.7.2	XML als XML-Daten oder als Zeichenfolge abrufen	673
13.7.3	Dokumente, Fragmente und FOR XML-Unterstützung	674
13.7.4	Den XML-Datentyp in klassischem ADO verwenden	677
13.8	Die Typen mit umfangreichen Werten auf dem Client unterstützen	679
13.9	Unterstützung für Abfragebenachrichtigung	683
13.9.1	Abfragebenachrichtigungen in SQL Server 2005	686
13.9.2	Abfragebenachrichtigungen in OLE DB und ODBC verwenden	688
13.9.3	Verteilen einer Benachrichtigung an einen Endbenutzer oder den Cache	689
13.9.4	Abfragebenachrichtigungen von einem Datenbankclient verwenden	691
13.9.5	SqlDependency verwenden	692
13.9.6	SqlNotificationRequest verwenden	694
13.9.7	SqlCacheDependency in ASP.NET verwenden	697
13.9.8	Mit Benachrichtigungen überfluten	698
13.9.9	Wann Sie Benachrichtigungen nicht verwenden sollten	700
13.10	Mehrere aktive Resultsets	701
13.10.1	MARS in ADO.NET 2.0 verwenden	703
13.10.2	MARS in SQL Native Client	706
13.11	Neue Transaktions- und Isolations-Features in ADO.NET	706
13.11.1	Die neuen Isolationsstufen verwenden	706
13.11.2	Heraufstufbare und deklarative Transaktionen	708
13.12	Änderungen in Bezug auf die SQL Server 2005-Anmeldung	710
13.12.1	Unterstützung der Kennwortänderung	710
13.12.2	Failover-Unterstützung	712
13.12.3	Unterstützung der Verschlüsselung	713
13.13	Das Client- und Servermodell für gespeicherte Prozeduren vergleichen	714
13.14	Zusammenfassung	715

14 ADO.NET 2.0 und SqlClient	717
14.1 Generisches Kodieren mit den ADO.NET 2.0-Basisklassen und Factorys	717
14.2 Provider-Factorys	721
14.3 Konfigurationsinformationen spezifizieren	723
14.4 Datenquellen aufzählen und Verbindungszeichenfolgen erstellen	725
14.5 Andere Betrachtungen zur generischen Kodierung	729
14.6 Schemas in ADO.NET 2.0	730
14.6.1 Wer braucht überhaupt Metadaten?	731
14.6.2 Welche Metadaten sind verfügbar?	733
14.6.3 Metadaten anpassen und erweitern	737
14.6.4 Anpassungen durch den Benutzer	739
14.7 Den Datenzugriff verfolgen	741
14.7.1 Eine Datenablaufverfolgung einrichten	742
14.7.2 Die Ablaufverfolgung ausführen	744
14.7.3 Die Ergebnisse als CSV-Datei erfassen	745
14.7.4 Die Ausgabe der Ablaufverfolgung lesen	745
14.7.5 Benutzerdaten und ADO.NET-Ablaufverfolgung	746
14.7.6 Konfigurieren, welche Anwendungen verfolgt werden	748
14.7.7 Per Ablaufverfolgung ein Parameterbindungsproblem debuggen	750
14.7.8 Datenablaufverfolgung intern	752
14.7.9 Was ist ETW?	752
14.8 Unterstützung für asynchrone Ausführung	753
14.9 Massenimport in SqlClient	758
14.10 Clientstatistiken	761
14.11 Erweiterungen von .NET Framework DataSet und SqlDataAdapter	763
14.12 Zusammenfassung	767
15 SQL Server Management Objects	769
15.1 Einführung	769
15.1.1 Warum SMO verwenden?	772
15.1.2 ADO vs. SMO	775
15.2 Objektmodell	777
15.3 SMO-Projekte	780

15.4	Verbindungen	785
15.4.1	Standardverbindung mit Windows-Anmeldeinformationen herstellen	786
15.4.2	Runas verwenden	788
15.4.3	Automatische Verbindung	789
15.4.4	Windows-Identitätswechsel	790
15.4.5	SQL Server-Anmeldung	791
15.4.6	Das Kennwort einer SQL Server-Anmeldung ändern	792
15.4.7	Verbindungen verwalten	793
15.5	Server	795
15.6	SMO-Objekte	797
15.6.1	Objektidentität und URNs	797
15.6.2	Die Methode GetSmoObject	800
15.6.3	Einschränkung beim Abrufen per URN	801
15.6.4	SMO-Objektmodell	803
15.6.5	SMO-Eigenschaften	803
15.6.6	SMO-Parent-Navigation	806
15.6.7	SMO-Zustand	808
15.7	Die Methoden Create, Alter und Drop	809
15.8	Skripts	814
15.9	Konfigurationsverwaltung	822
15.10	Zusammenfassung	824
16	Notification Services	827
16.1	Was sind die SQL Server Notification Services?	827
16.2	Benachrichtigungsanwendungen	829
16.3	Komponenten der SQL Server Notification Services	832
16.4	Entwurfsmuster von Benachrichtigungsanwendungen	836
16.5	Übermittlungsfeatures von Notification Services	837
16.6	Notification Services – Terminologie	837
16.7	Eine Notification Services-Anwendung entwerfen, kodieren und generieren	839
16.7.1	Planungsschritte	839
16.7.2	Kodierungsschritte	840
16.7.3	Schritte der Anwendungsgenerierung	841
16.8	Beispiel für eine Benachrichtigungsanwendung	841
16.9	Instanz- und Anwendungskonfigurationsdateien	842
16.9.1	Instanzkonfigurationsdatei	843
16.9.2	Anwendungsdefinitionsdatei	847
16.9.3	NSControl	850



16.10	Ereignisse	852
16.10.1	Ereignisklassen	853
16.10.2	Gespeicherte Prozeduren für Ereignisse	856
16.10.3	Ereignisprovider	858
16.10.4	Verläufe	862
16.11	Abonnenten und Abonnements	864
16.11.1	Abonnementklassen	866
16.11.2	Abonnenten	868
16.11.3	Abonnements	870
16.12	Benachrichtigungen	872
16.13	Formatierer und Verteiler	877
16.14	Übermittlung	879
16.14.1	Übermittlungsprotokolle	879
16.14.2	Anpassung	882
16.15	Zusammenfassung	883
17	Zusammenfassung: Dienstorientierte Datenbankanwendungen	885
17.1	Jede Menge Features und wie man Nutzen aus ihnen zieht	885
17.2	Datenmodelle, Programmierung und SQL Server	886
17.3	Jede Funktionalität in jeder Schicht	887
17.4	Welche Empfehlungen können gegeben werden?	889
17.5	Auf dem Weg zu einer dienstorientierten Datenbank-architektur	899
17.6	Die Datenbank als Teil der Plattform	902
A	.NET Framework 101	905
A.1	Die Common Language Runtime	905
A.1.1	Entwicklungssprachen	907
A.2	Assemblys und Module	908
A.2.1	Module	908
A.2.2	Assemblys	910
A.2.3	Assembly-Namen und Versionen	912
A.3	Das CLR-Typsystem	913
A.3.1	Klassen und Typen	915
A.3.2	Sichtbarkeit	916
A.3.3	Instanzen	917
A.3.4	Namespaces	919
A.4	Typmember	921
A.4.1	Felder und Eigenschaften	922
A.4.2	Parameter und Methoden	924

A.5	Speicherverwaltung	925
A.5.1	Finalisierung und Destruktoren	925
A.5.2	Löschen (Disposing)	926
B	SQL Server Management Studio	929
B.1	Einführung	929
B.2	Projektmappen und Projekte	930
B.2.1	Konfiguration	931
B.2.2	Eine Projektmappe/ein Projekt erstellen	933
B.3	Vorlagen	943
B.4	Objekt-Explorer	948
C	Visual Studio 2005-Integration: SQL Server-Projekte	955
C.1	Visual Studio 2005	955
C.1.1	Benutzerdefinierte Attribute und automatische Bereitstellung	962
C.1.2	Automatische Bereitstellung	962
C.1.3	Datentypen und SqlFacet	964
C.1.4	Den Code ausführen und debuggen	966
	Quellen	969
	Stichwortverzeichnis	971