



SQL Server 2012 – Neues für Entwickler

SNEK 1 – März/April 2012
Bernd Jungbluth

www.berndjungbluth.de



■ Vorstellung

■ Bernd Jungbluth

■ IT-Erfahrung

- ☐ SQL und Datenbanken seit 1991
- ☐ SQL Server seit Version 7.0

■ Freiberuflicher Berater und Entwickler

- ☐ Administration SQL Server
- ☐ Entwicklung und Optimierung von SQL Server-Datenbanken
- ☐ SQL Server Reporting Services
- ☐ SQL Server Integration Services
- ☐ Migration Access nach SQL Server



■ Agenda

■ Datenbankcontainer

- ☐ Datenbank und Datenbankcontainer im Vergleich
- ☐ Konfiguration von Datenbankcontainern

■ Nicht relationale Daten

- ☐ FILESTREAM und FileTable
- ☐ Volltextsuche in nicht relationalen Daten

■ T-SQL

- ☐ Fensterfunktionen
- ☐ Neue Befehle und Funktionen

■ SQL Server Data Tools

- ☐ Die neue Entwicklungsumgebung
- ☐ Business Intelligence- und Datenbankprojekte

■ Datenbankcontainer

■ Datenbanken

- ☐ Integriert in einer SQL Server-Instanz
- ☐ Administration über eine SQL Server-Instanz
- ☐ Datenbankzugriff über die als Benutzer zugeordneten Anmeldungen
- ☐ Zentrale Verwaltung der Anmeldungen in der SQL Server-Instanz
- ☐ Zentrales Speichern temporärer Objekte in der tempdb
- ☐ Von der SQL Server-Instanz abhängige Datenbanken

■ Datenbankcontainer

- ☐ Nur teilweise integriert in einer SQL Server-Instanz
- ☐ Administration über eine SQL Server-Instanz
- ☐ Datenbankzugriff über die als Benutzer zugeordneten Anmeldungen
- ☐ Verwalten der Anmeldungen im Datenbankcontainer
- ☐ Speichern temporärer Objekte im Datenbankcontainer
- ☐ Von der SQL Server-Instanz unabhängige Datenbanken

■ Datenbankcontainer / Vorteile

■ Vorteile bei der Administration

- ☐ Einfaches Verschieben eines Datenbankcontainers von Instanz zu Instanz
- ☐ Einfaches Verschieben eines Datenbankcontainers zu SQL Azure
- ☐ Einfaches Sichern und Wiederherstellen von Datenbankcontainern
- ☐ Keine »verwaisten Logins« nach Wiederherstellen eines Datenbankcontainers
- ☐ Keine doppelte Pflege von Anmeldungen bei Hochverfügbarkeit-Szenarien
- ☐ Keine zentrale Verwaltung von Anmeldungen erforderlich
- ☐ Keine Zuordnung zentraler Anmeldungen zum Datenbankcontainer notwendig

■ Vorteile bei der Entwicklung

- ☐ Möglichkeit zur Entwicklung unabhängiger Datenbanken
- ☐ Liste der unterstützten Funktionen unter <http://msdn.microsoft.com/en-us/library/ff929188%28v=sql.110%29.aspx>
- ☐ Unabhängig von der Sortierung der tempdb einer SQL Server-Instanz
- ☐ Vereinfachtes Bereitstellen von Datenbankencontainern

■ Datenbankcontainer / Konfiguration

■ Konfiguration Datenbankcontainer

- ☐ Basiskonfiguration in den Eigenschaften der SQL Server-Instanz
- ☐ Aktivieren der Option *Eigenständige Datenbank aktivieren* in der Seite *Erweitert*
- ☐ Konfiguration zum Datenbankcontainer in den Datenbank-Eigenschaften
- ☐ Auswahl des Einschlusstyps *Teilweise* in der Seite *Optionen*
- ☐ Konfiguration fünf weiterer Optionen, bspw. Auswahl der Standardsprache

■ Konfiguration Integrierter Datenbankbenutzer

- ☐ »Contained Database User«
- ☐ Auswahl des Benutzertyps *SQL-Benutzer mit Kennwort*
- ☐ Eingabe von Benutzername und Kennwort
- ☐ Natürlich nur möglich bei Instanzen mit gemischter Authentifizierung

■ Konfiguration Datenbankbenutzer mit Windows-Authentifizierung

- ☐ Auswahl des Benutzertyps *Windows-Benutzer*
- ☐ Eingabe des Benutzerkontos ohne Zuordnung zu einer Anmeldung

- Datenbankcontainer

- Demo



■ Agenda

■ Datenbankcontainer

- ☐ Datenbank und Datenbankcontainer im Vergleich
- ☐ Konfiguration von Datenbankcontainern

■ Nicht relationale Daten

- ☐ FILESTREAM und FileTable
- ☐ Volltextsuche in nicht relationalen Daten

■ T-SQL

- ☐ Fensterfunktionen
- ☐ Neue Befehle und Funktionen

■ SQL Server Data Tools

- ☐ Die neue Entwicklungsumgebung
- ☐ Business Intelligence- und Datenbankprojekte

■ Nicht relationale Daten / FILESTREAM

■ Nicht relationale Daten

- ☐ Speichern von Dateien und Dokumenten im SQL Server
- ☐ Verwalten und Suchen von Dateien und Dokumenten
- ☐ Skalierbar bis zu 100 Millionen Dateien und Dokumente

■ FILESTREAM

- ☐ Verfügbar seit SQL Server 2008
- ☐ Zum Speichern von Dateien und Dokumenten per SQL Server
- ☐ Tatsächlicher Speicherort der Dateien und Dokumente im Dateisystem
- ☐ Bearbeiten der so gespeicherten Dateien nur über SQL Server möglich
- ☐ Kein direkter Zugriff auf die Dateien über das Dateisystem möglich
- ☐ Transaktionale und konsistente Verwaltung der Dateien durch SQL Server
- ☐ Konsistentes Backup der Dateien und Dokumente per Datenbanksicherung
- ☐ Kein Datentyp, lediglich eine Erweiterung einer Spalte vom Typ *varbinary(max)*
- ☐ Mit SQL Server 2012 mehrere FILESTREAM-Dateigruppen pro Datenbank möglich
- ☐ Mit SQL Server 2012 Hochverfügbarkeit mittels AlwaysOn möglich

■ Nicht relationale Daten / FileTable

■ FileTable

- ☐ Spezielle Art von Tabellen einer SQL Server-Datenbank
- ☐ Zum Verwalten von Verzeichnissen und deren Dateien in einer Tabelle
- ☐ Speichert ein Verzeichnis mitsamt den Unterverzeichnissen und Dateien
- ☐ Enthält FILESTREAM-Daten, die Verzeichnisstruktur und zehn Dateiattribute

■ Möglichkeiten

- ☐ Verwalten von Verzeichnissen und Dateien
- ☐ Mit SQL Server sowie dem Betriebssystem und Windows-Applikationen
- ☐ Volltextsuche in Dateien mit SQL Server

■ Dateizugriff

- ☐ Direktes Bearbeiten der Dateien über Windows-Applikationen möglich
- ☐ Verfügbar über eine Windowsfreigabe vom Stammverzeichnis der FileTable
- ☐ Zugriffssteuerung über das Berechtigungssystem der Datenbank
- ☐ Unterstützt keine Applikationen, die Speicherabbilddateien verwenden

■ Nicht relationale Daten / Konfiguration

■ Konfiguration SQL Server

- ☐ Aktivieren von FILESTREAM an der SQL Server-Instanz
- ☐ Angeben des FILESTREAM-Verzeichnisses in den Datenbankeigenschaften
- ☐ Option *FILESTREAM-Verzeichnisname* in der Seite *Optionen*
- ☐ Aktivieren des nicht transaktionalen Zugriffs in den Datenbankeigenschaften
- ☐ Option *nicht transaktionsgebundener FILESTREAM-Zugriff* in der Seite *Optionen*
- ☐ Anlegen einer FileTable inklusive Zuordnung zum Stammverzeichnis

■ Nicht transaktionaler Zugriff

- ☐ Aktivierung für den direkten Dateizugriff mit Windows-Applikationen erforderlich
- ☐ Transaktionen bei Dateizugriff durch Windows-Applikationen nicht möglich
- ☐ Transaktionaler Zugriff per T-SQL unabhängig dieser Konfiguration gegeben

■ Konfiguration Virens Scanner

- ☐ Verschieben als »infiziert« erkannte Dateien in Quarantäne-Ordner
- ☐ Als infiziert erkannte Dateien nicht löschen



■ Nicht relationale Daten

■ Demo

... unterstützt durch Produktempfehlungen ☺

■ Nicht relationale Daten / Volltextsuche

■ Volltextsuche

- ☐ In Spalten mit Zeichenfolgen
- ☐ In FILESTREAM-Daten – und somit in Dateien und Dokumenten
- ☐ Unterstützt mit Microsoft Filter Pack 2.0 bis zu 150 Dokumenttypen

■ Neue Suchmöglichkeiten

- ☐ Konfigurierbare NEAR-Suche
- ☐ Semantische Suche
- ☐ Suche nach Datei-Eigenschaften

■ Konfigurierbare NEAR-Suche

- ☐ Suche nach Suchbegriffen in der Nähe eines anderen Suchbegriffs
- ☐ Nach mehreren miteinander verbundenen Suchbegriffen
- ☐ Nach zwei Suchbegriffen, die durch eine maximale Anzahl von Nicht-Suchbegriffen getrennt sind
- ☐ Nach mehreren Suchbegriffen in einer festgelegten Reihenfolge

■ Nicht relationale Daten / Volltextsuche

■ Semantische Suche

- ☐ Ähnlichkeitssuche in Zeichenfolgen, Dateien und Dokumenten
- ☐ Ermitteln von verwandten Inhalten
- ☐ Suche nach der Bedeutung des Inhalts
- ☐ Erstellen eines Volltextkatalogs zur jeweiligen Spalte erforderlich
- ☐ Beinhaltet sprachenabhängige Analyse vom Inhalt der Spalte
- ☐ Extrahiert Informationen und deren Beziehungen zum Inhalt der Spalte
- ☐ Verwendet zur Extraktion die »Language Statistical Database«
- ☐ »Language Statistical Database« als separater Download verfügbar
- ☐ Speichert die extrahierten Informationen in drei Systemtabellen

■ Suche nach Datei-Eigenschaften

- ☐ Suche nach Eigenschaften von Dateien, wie Autor, Titel, Erstellt am ...
- ☐ Erstellen einer Sucheigenschaftenliste notwendig
- ☐ Selbständiges Füllen des Volltextkatalogs durch automatische Tagextraktion



■ Nicht relationale Daten / Volltextsuche

■ Demo

... unterstützt durch Produktempfehlungen ☺



■ Agenda

■ Datenbankcontainer

- ☐ Datenbank und Datenbankcontainer im Vergleich
- ☐ Konfiguration von Datenbankcontainern

■ Nicht relationale Daten

- ☐ FILESTREAM und FileTable
- ☐ Volltextsuche in nicht relationalen Daten

■ T-SQL

- ☐ Fensterfunktionen
- ☐ Neue Befehle und Funktionen

■ SQL Server Data Tools

- ☐ Die neue Entwicklungsumgebung
- ☐ Business Intelligence- und Datenbankprojekte

■ T-SQL / Fensterfunktionen

■ Fensterfunktionen

- ☐ Ermitteln von Werten aus einem definierten Bereich von Datensätzen einer Ergebnismenge im Bezug zum jeweils aktuellen Datensatz
- ☐ Betrachten der Daten einer Ergebnismenge anhand eines definierten Rahmens
- ☐ Auch »Windowing«, »Window-Function« und »Fensterrahmen« genannt

■ Verwendung

- ☐ Bei Rangfolge- und Aggregatfunktionen sowie bei analytischen Funktionen
- ☐ Bei der NEXT VALUE FOR-Funktion zur Erstellung von Sequenzen

■ Syntax

- ☐ Erweiterung der OVER-Klausel: OVER (<partition> <order> <frame>)
- ☐ <partition> zum Unterteilen der Ergebnismenge in mehrere Partitionen
- ☐ <order> zum Sortieren der Datensätze innerhalb der Partitionen
- ☐ <frame> zum Eingrenzen der Datensätze innerhalb der Partitionen
- ☐ Definition des Rahmens per ROWS oder RANGE

■ T-SQL / Fensterfunktionen

■ ROWS

- Begrenzt den Bereich ausgehend von der aktuellen Zeile
- Beispiel zur Ausgabe der aktuellen Zeile und der fünf folgenden Zeilen:
ROWS BETWEEN CURRENT ROW AND 5 FOLLOWING

■ RANGE

- Begrenzt den Bereich ausgehend von Werten der aktuellen Zeile
- Beispiel zur Ausgabe aller Zeilen der Partition bis zur aktuellen Zeile:
RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

■ Ergänzende Parameter

- `<x> PRECEDING` und `<x> FOLLOWING`
zur Ausgabe einer definierten Anzahl vorheriger bzw. nachfolgender Datensätze
- `UNBOUNDED PRECEDING` und `UNBOUNDED FOLLOWING`
zur Ausgabe vom Anfang bis zum aktuellen Datensatz bzw. von dort bis zum Ende
- `CURRENT ROW` zur Definition des aktuellen Datensatzes

■ T-SQL / Fensterfunktionen

■ Kombinationen mit ROWS

- ☐ ROWS UNBOUNDED PRECEDING
- ☐ ROWS <unsigned integer literal> PRECEDING
- ☐ ROWS CURRENT ROW
- ☐ ROWS BETWEEN UNBOUNDED PRECEDING AND <unsigned integer literal> PRECEDING
- ☐ ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
- ☐ ROWS BETWEEN UNBOUNDED PRECEDING AND <unsigned integer literal> FOLLOWING
- ☐ ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
- ☐ ROWS BETWEEN <unsigned integer literal> PRECEDING AND <unsigned integer literal> PRECEDING
- ☐ ROWS BETWEEN <unsigned integer literal> PRECEDING AND CURRENT ROW
- ☐ ROWS BETWEEN <unsigned integer literal> PRECEDING AND <unsigned integer literal> FOLLOWING
- ☐ ROWS BETWEEN <unsigned integer literal> PRECEDING AND UNBOUNDED FOLLOWING
- ☐ ROWS BETWEEN CURRENT ROW AND CURRENT ROW
- ☐ ROWS BETWEEN CURRENT ROW AND <unsigned integer> FOLLOWING
- ☐ ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
- ☐ ROWS BETWEEN <unsigned integer literal> FOLLOWING AND <unsigned integer> FOLLOWING
- ☐ ROWS BETWEEN <unsigned integer literal> FOLLOWING AND UNBOUNDED FOLLOWING

■ Kombinationen mit RANGE

- ☐ RANGE CURRENT ROW
- ☐ RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
- ☐ RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
- ☐ RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
- ☐ RANGE BETWEEN CURRENT ROW AND CURRENT ROW



- T-SQL / Fensterfunktionen

- Demo

■ T-SQL / Neue Befehle und Funktionen

■ Sequenzen

- ☐ Transaktional konsistenter Zähler einer Datenbank
- ☐ Für einheitliche Primärschlüssel über mehrere Tabellen
- ☐ Für Zeilennummern in Ergebnismengen

■ Ad-hoc-Abfrage-Auslagerung – »Paging«

- ☐ Selektive Ausgabe von Datensätzen aus dem Abfrageergebnis
- ☐ Positionieren im Abfrageergebnis

■ Metadatenermittlung

- ☐ Metadaten der Spalten einer Ergebnismenge
- ☐ Metadaten der Parameter einer SELECT-Anweisung oder Gespeicherten Prozedur
- ☐ Per Systemprozeduren und Dynamic Management Views
- ☐ Mit Erweiterung WITH RESULT SETS bei EXECUTE

■ Beispiele im Download der SEK 4

■ T-SQL / Neue Befehle und Funktionen

■ Analytische Funktionen

- ☐ LAG – Ermitteln eines Werts vom vorherigen Datensatz
- ☐ LEAD – Ermitteln eines Werts vom nachfolgenden Datensatz
- ☐ FIRST_VALUE – Ermitteln eines Werts vom ersten Datensatz der Ergebnismenge
- ☐ LAST_VALUE – Ermitteln eines Werts vom letzten Datensatz der Ergebnismenge
- ☐ PERCENTILE_DISC und PERCENTILE_CONT – Berechnen eines Quantils
- ☐ CUME_DIST – Berechnen der relativen Position eines Werts
- ☐ PERCENT_RANK – Berechnen des relativen Rangs eines Werts

■ Logische Funktionen

- ☐ IIF – Auswählen eines Werts in Abhängigkeit einer booleschen Prüfung
- ☐ CHOOSE – Auswählen eines Werts einer Werteliste

■ Datumsfunktionen

- ☐ EOMONTH – Ermitteln des Datums vom letzten Tag eines Monats
- ☐ 10 Funktionen zum Erstellen eines Datums aus dessen Bestandteilen

■ T-SQL / Neue Befehle und Funktionen

■ Konvertierungsfunktionen

- PARSE – Konvertieren von Datumswerten und numerischen Werten zu Zeichenfolgen unter Berücksichtigung der Ländereinstellung
- TRY_CONVERT und TRY_CAST sowie TRY_PARSE – Konvertieren von Werten mit Ausgabe von NULL bei fehlerhafter Konvertierung

■ Zeichenfolgenfunktionen

- CONCAT – Verkettung von mehreren Zeichenfolgen
- FORMAT – Formatieren von Datumswerten und numerischen Werten unter Berücksichtigung der Ländereinstellung

■ Funktionen zur Fehlerbehandlung

- THROW – Auslösen eines Fehlers mit Schweregrad 16
- Übergabe des Fehlers an den CATCH-Part eines TRY-CATCH-Blocks
- Wiederholtes Auslösen des Fehlers im CATCH-Block möglich
- Keine Fehlernummer in sys.messages erforderlich



- T-SQL / Neue Befehle und Funktionen

- Demo



■ Agenda

■ Datenbankcontainer

- ☐ Datenbank und Datenbankcontainer im Vergleich
- ☐ Konfiguration von Datenbankcontainern

■ Nicht relationale Daten

- ☐ FILESTREAM und FileTable
- ☐ Volltextsuche in nicht relationalen Daten

■ T-SQL

- ☐ Fensterfunktionen
- ☐ Neue Befehle und Funktionen

■ SQL Server Data Tools

- ☐ Die neue Entwicklungsumgebung
- ☐ Business Intelligence- und Datenbankprojekte

■ SQL Server Data Tools

■ SQL Server Data Tools – »SSDT«

- ☐ »Application Lifecycle Management« für Datenbankprojekte
- ☐ Neue Entwicklungsumgebung für Business Intelligence- und Datenbank-Entwickler
- ☐ Ersetzt Business Intelligence Development Studio
- ☐ Nicht ganz so neue Entwicklungsumgebung für .NET-Entwickler
- ☐ Bekannt als Visual Studio Team Edition for Database Professionals – »Data Dude«
- ☐ Integriert in Visual Studio 2010 und Visual Studio vNext

■ Installation

- ☐ Visual Studio 2010 im Lieferumfang von SQL Server 2012 enthalten
- ☐ Separate Installation von SQL Server Data Tools erforderlich
- ☐ Download und Installation bei der ersten Verwendung eines Datenbankprojekts

■ Projektvorlagen

- ☐ Business Intelligence – SSIS, SSAS, SSRS
- ☐ SQL Server Datenbankprojekte

■ SQL Server Data Tools / Datenbankprojekte

■ SQL Server Datenbankprojekte

- ☐ Projektbasierte Datenbankentwicklung in Visual Studio
- ☐ Unterstützt Datenbankprojekte für SQL Server ab Version 2005
- ☐ Unterstützt Datenbankprojekte für SQL Azure
- ☐ Funktionsumfang abhängig von der im Projekt verwendeten SQL Server-Edition

■ Entwicklungsumgebung

- ☐ Online- und Offline-Entwicklung
- ☐ Skriptbasierte und graphische Entwicklung
- ☐ Definition von Codierungsregeln
- ☐ Erstellen von Projekt-Snapshots zum Sichern einzelner Projektstufen
- ☐ Vergleichen und Synchronisieren von Datenbanken und Datenbankprojekten
- ☐ Bereitstellen über Veröffentlichen, per Skripte oder mittels dacpac-Dateien
- ☐ Bietet bekannte Möglichkeiten und Funktionalität von Visual Studio
- ☐ IntelliSense, GoTo Definition, Find All References, Refactoring, Debuggen
- ☐ Mit Visual Studio vNext noch Unit Testing, Data Generation, Data Compare

■ SQL Server Data Tools / Datenbankprojekte

■ Entwickeln von Datenbanken

- ☐ Projektbasierte Datenbankentwicklung unabhängig von SQL Server
- ☐ Skriptbasierte und graphische Entwicklung von SQL Server-Objekten
- ☐ Syntaktische Prüfung von T-SQL per »SQL Server T-SQL Compiler Service«
- ☐ Importieren der Objekte bestehender Datenbanken als separate SQL-Skripte
- ☐ Speichern der SQL Server-Objekte im Projekt primär als SQL-Skripte
- ☐ Speichern der Entwicklungsdatenbank in »SQL Server 2012 Express LocalDB«
- ☐ Debuggen von Gespeicherten Prozeduren und Funktionen
- ☐ Vorbereiten und prüfen der Veröffentlichung in einer SQL Server-Instanz

■ Veröffentlichen von Datenbanken

- ☐ Direktes Ausführen der Änderungen in der Ziel-Datenbank
- ☐ Bietet Auswirkungsanalyse vor Ausführung mit Darstellung möglicher Probleme
- ☐ Erstellen von SQL-Skripten zum Installieren oder Ändern einer Datenbank
- ☐ Vergleichen und Synchronisieren des Datenbankprojekts mit der Ziel-Datenbank
- ☐ Erstellen von dacpac-Dateien zum Installieren oder Ändern einer Datenbank

■ SQL Server Data Tools / SQL Server 2012 Express LocalDB

■ SQL Server 2012 Express LocalDB

- ☐ SQL Server Express Edition für Entwickler
- ☐ Kein SQL Server-Dienst, sondern applikationsbezogene Datenbank
- ☐ Ausführung als Prozess unter der Anmeldung des ausführenden Benutzers
- ☐ Eigenständige SQL Server-Instanz im Benutzerverzeichnis

■ LocalDB-Instanzen

- ☐ Name der Standardinstanz (*localdb*)\<projektname>
- ☐ Erstellen eigener weiterer Instanzen möglich
- ☐ Administration per SQL Server Data Tools und SQL Server Management Studio
- ☐ Direkte Verbindung zur Datenbankdatei möglich – »Database as a File«

■ Unterschied zur SQL Server Compact Edition

- ☐ Bietet vollen Funktionsumfang einer SQL Server Express Edition
- ☐ Ausführung als eigener Prozess
- ☐ 140 MB benötigter Speicherplatz für die Programmdateien



- SQL Server Data Tools

- Demo

■ Zusammenfassung und Fazit

■ Datenbankcontainer

- Unabhängige Datenbanken in einer SQL Server-Instanz
- Keine Abhängigkeit zu Anmeldungen und tempdb der SQL Server-Instanz

■ Nicht relationale Daten

- Verwalten von Verzeichnissen und Dateien mit SQL Server
- Direkter Zugriff auf Dateien einer FileTable mit Windows-Applikationen
- Einfache und semantische Volltextsuche in Dateien und Dokumenten

■ T-SQL

- Fensterfunktionen als neue und effiziente Möglichkeit in der Datenermittlung
- Neue Analytische Funktionen und weitere hilfreiche T-SQL-Befehle

■ SQL Server Data Tools

- In SQL Server integrierte Entwicklungsumgebung für Datenbankprojekte
- »Application Lifecycle Management« in der Datenbankentwicklung

■ Hinweise und Links

■ Editionen

- »Enterprise Edition«, »Business Intelligence Edition«, »Standard Edition«
- Übersicht: <http://www.microsoft.com/de-de/server/sql/editionen.aspx>
- Verfügbar ab 01. April 2012 – RTM seit 06. März 2012

■ Links

- Website: <http://www.microsoft.com/de-de/server/sql/default.aspx>
- Alle Downloads: <http://blog.sqlauthority.com/2012/03/11/sql-server-2012-all-download-links-in-single-page-sql-server-2012/>
- Blog: <http://www.sqlserverblogs.de>
- Community: <http://www.sqlpass.de>

■ Links zu SQL Server Data Tools

- Website: <http://msdn.microsoft.com/de-de/data/tools.aspx>
- Team Blog: <http://blogs.msdn.com/b/ssdt/>
- MSDN-Artikel: <http://msdn.microsoft.com/de-de/magazine/hh394146.aspx>



■ Danke

Vielen Dank für die Aufmerksamkeit ☺