

Microsoft SQL Server 2008 R2 Common Language Runtime

4. SQL Server Entwicklerkonferenz

Thomas Trefz

- Praktische Demonstrationen
- Anmerkungen zu Triggern
- Sicherheitsoptionen
- Einsatzgebiete der CLR, Abgrenzung zu T-SQL
- Diskussion

- Wie immer: viel Praxis, wenig Folien!
- Sage ich „Sie“ meine ich „Du“
- Folien und Beispiele auf SEK Download Seite
- Es geht nicht um den besten .NET Code!
 - Zumindest nicht in diesem Vortrag
- Beispiele basieren auf MSDN Artikel
- Nutzen der weltweiten Kreativität / Erfahrung
 - Aber: „Kopieren“ und „Kapiieren“!

- Entwickeln von Datenbankobjekten in allen .NET kompatiblen Sprachen
- Objekte
 - Gespeicherte Prozeduren, Trigger und Funktionen
 - Benutzerdefinierte Typen
 - Benutzerdefinierte Aggregatfunktionen
- Datenzugriff über ADO.NET
 - Es wird weiterhin T-SQL verwendet
- Aufruf / Nutzung wie Standard T-SQL Objekte
 - Der Benutzer wird keinen Unterschied erkennen

Live Demo: Aggregatfunktion



- Init() Methode wird einmal pro Gruppenkriterium aufgerufen
- Accumulate() wird für jeden Wert des Gruppenkriteriums aufgerufen
 - Wert wird als Parameter übergeben (oder mehrere)
- Merge() wird aufgerufen wenn Teilmengen zusammengeführt werden müssen
- Terminate() wird einmal pro Gruppenkriterium aufgerufen
 - Rückgabe des Ergebnisses
- Datentypen berücksichtigen → Serialisierung!

- Debugging aus der IDE: Voraussetzungen
 - Berechtigungsebene „Sicher“
 - CLR-Debugging für Serververbindung
 - Debuggen über Testskript
 - Festlegen „Standardskript zum Debuggen“
 - Debuggen über Server Explorer
 - Debuggen von Prozeduren und Funktionen
- Debugging über Serverprozess
 - Anhängen an einen Serverprozess (sqlservr.exe)
 - z.B. wenn eine andere Anwendung den Kontext um das CLR Objekt zu Debuggen bereitstellt.

- Registrieren der Assembly in SQL Server über T-SQL Anweisung `CREATE ASSEMBLY`
 - Angabe des Pfades für die Assembly
 - Angabe der Sicherheitsoption
- Erstellen der Objekte über T-SQL Anweisung `CREATE . . . EXTERNAL NAME`
 - Unterschiedliche Syntax für die verschiedenen Objekte
- Aktualisierung der Assembly über T-SQL Anweisung `ALTER ASSEMBLY`
 - Aber nur bedingt, z.B. keine Signaturänderungen!

Live Demo: Datentyp



- Werden als Public Structure implementiert
- Müssen Interface INullable implementieren
- ToString() und Parse() für Konvertierung
- SqlMethod Attribut: OnNullCall definiert ob die Methode bei einem Null Wert aufgerufen wird.
- SqlMethod Attribut: IsMutator definiert ob eine Methode den Status der UDT ändert (Werte)
- SqlUserDefinedType Attribut: ValidationMethodName definiert eine Methode zur Prüfung der Werte.

Live Demo: Tabellenwertfunktion



- Werden als Methoden einer Klasse definiert
- SqlFunction Attribut: FillRowMethodName
 - Definiert die Methode die durch die CLR aufgerufen wird um die Ergebnismenge zurückzugeben.
- SqlFunction Attribut: TableDefinition
 - Definiert die Metadaten der Ergebnismenge.

- T-SQL Implementierung:
 - Materialisieren die Ergebnisse in einer Zwischentabelle
 - Können Einschränkungen und Indizes unterstützen
- CLR Implementierung
 - IEnumerable Objekt wird direkt von der aufrufenden Abfrage genutzt
 - Ergebnisse werden sofort verarbeitet sobald der erste Ergebniswert vorhanden ist (Streaming Modell)

Live Demo: Prozeduren



- Typsicherheit
- Eingeschränktes Programmiermodell
- Zugriff auf die lokale Datenbank
- Assemblyinterne Berechnungen

Zugriff auf SQL Server externe Ressourcen, z.B. Dateisystem oder Netzwerk

Alle Einschränkungen der höheren Sicherheitsstufen sind aufgehoben, es gilt also uneingeschränkter Zugriff innerhalb und außerhalb von SQL Server

Unsicher

r

Extern

Sicher

r



Empfohlene Assemblyberechtigung: Sicher / Extern

- Die Qual der Wahl
 - Beide Techniken bringen Vor- und Nachteile mit sich
 - Es gibt keine „bessere Wahl“
- Grundsätzliche Vorzüge
 - T-SQL
 - Einfache / gewohnte Anwendungspflege
 - Hohe Performance bei deklarativen Statements
 - CLR
 - Erweiterter Funktionsumfang
 - Hohe Performance z.B. bei Rechenoperationen
- Kombination ist möglich

- Die Nutzung von CLR Prozeduren scheint im Hinblick auf umfangreiche Verarbeitungsabläufe (Batch) sinnvoll zu sein.
- Die Nutzung von CLR Datentypen ist zumindest mit Vorsicht einzuplanen.
- Die Pflege wird durch mehrere Assemblies vereinfacht
- Die Einbindung von „Unmanaged Code“ ist nicht trivial, speziell im Bereich Deployment / Sicherheit

- **SQL Server 2008 – Programmierung mit der CLR und .NET**
Microsoft Press
ISBN: 3-86645-436-8
- **SQL Server 2008 R2 – Der schnelle Einstieg**
Addison-Wesley
ISBN: 3-82732-955-4
- **Und viele andere ...**



Updates zu den Skripten und Inhalten
dieser Präsentation
gibt es auf / unter
<http://www.thomastrefz.de>