

.net goes SQL

CLR-Integration im SQL-Server

Alexander Karl

.net goes SQL

seit der Version SQL-Server 2005 können Programmierungen zusätzlich zum T-SQL auch mit .net-Sprachen erfolgen.

- Data Types
- Stored Procedures
- Triggers
- Functions
 - Scalar
 - Aggregate

.net goes SQL

seit der Version SQL-Server 2005 können Programmierungen zusätzlich zum T-SQL auch mit .net-Sprachen erfolgen.

Programmierung in C#,
integrieren in eine SQL-DB
und überwachen mittels DMVs

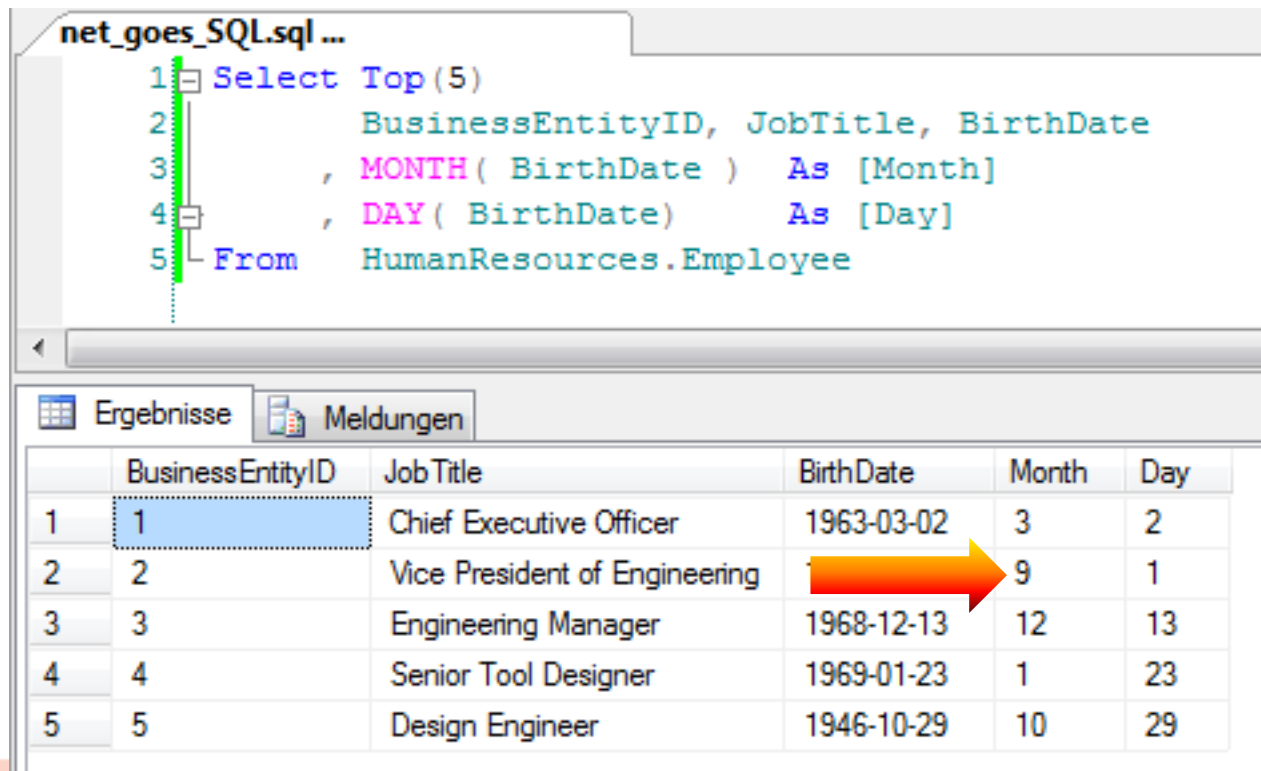
...heute am Beispiel der **Aggregat**-Funktion

.net goes SQL



aus der „Wirkungsrichtung“ der Funktionen ergeben sich die Besonderheiten.

1. Skalar-Funktion



```
net_goes_SQL.sql ...
1 Select Top (5)
2     BusinessEntityID, JobTitle, BirthDate
3     , MONTH( BirthDate ) As [Month]
4     , DAY( BirthDate)      As [Day]
5 From   HumanResources.Employee
```

	BusinessEntityID	Job Title	BirthDate	Month	Day
1	1	Chief Executive Officer	1963-03-02	3	2
2	2	Vice President of Engineering	1968-12-13	9	1
3	3	Engineering Manager	1969-01-23	1	23
4	4	Senior Tool Designer	1946-10-29	10	29
5	5	Design Engineer			

.net goes SQL

2. Aggregat-Funktion (a)

net_goes_SQL.sql ...

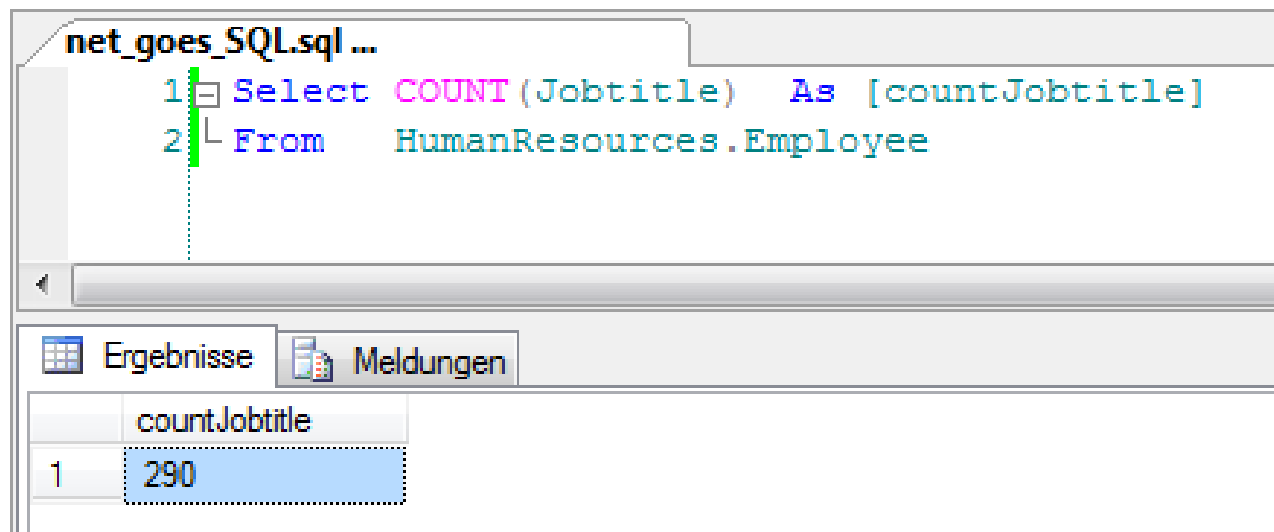
```
1 Select Jobtitle As [Jobtitle]
2 From HumanResources.Employee
```

Ergebnisse Meldungen

	count	Jobtitle
1	1	Chief Executive Officer
2	1	Vice President of Engineering
3	1	Engineering Manager
4	1	Senior Tool Designer
5	1	Design Engineer
6	1	Design Engineer
7	1	Research and Development Manager

.net goes SQL

2. Aggregat-Funktion (a)



The screenshot shows a SQL query editor window titled "net_goes_SQL.sql ...". The query is as follows:

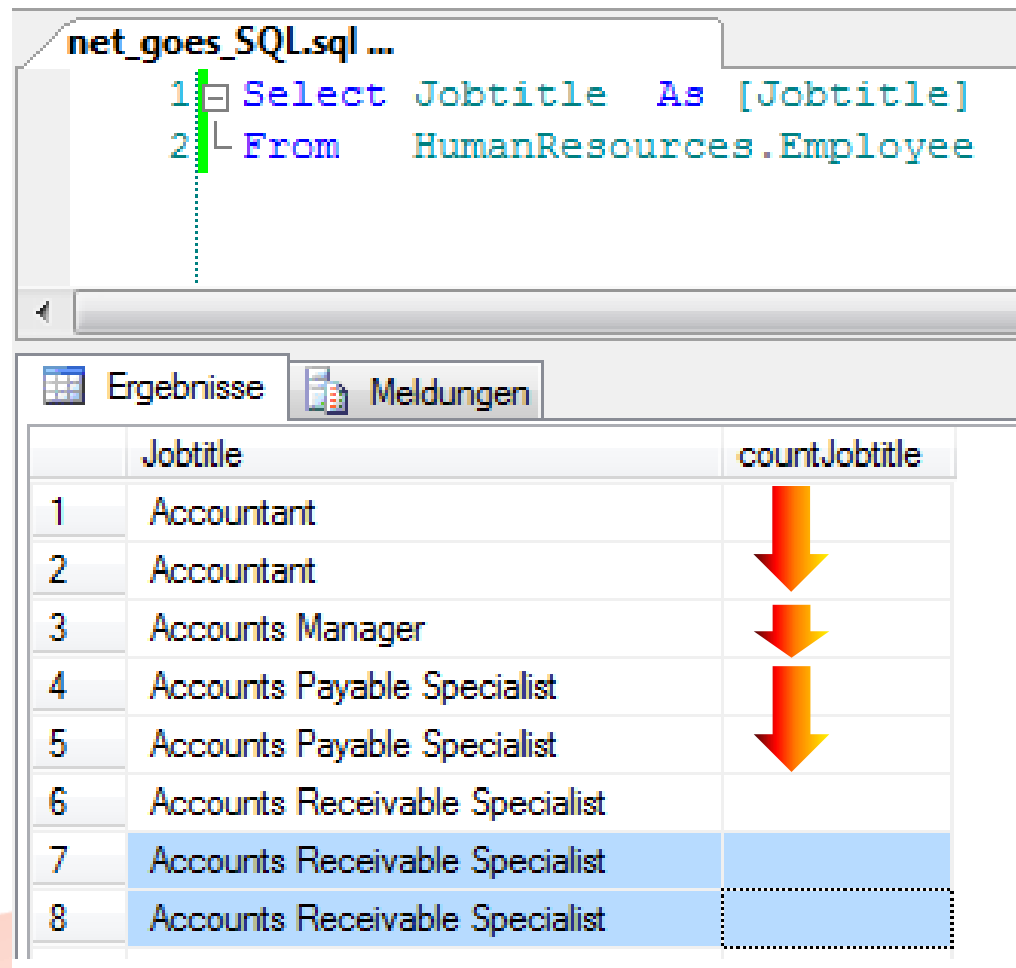
```
1 Select COUNT(Jobtitle) As [countJobtitle]
2 From HumanResources.Employee
```

Below the query editor, there are two tabs: "Ergebnisse" (Results) and "Meldungen" (Messages). The "Ergebnisse" tab is active, displaying a table with the following data:

	count.Jobtitle
1	290

.net goes SQL

2. Aggregat-Funktion (b)



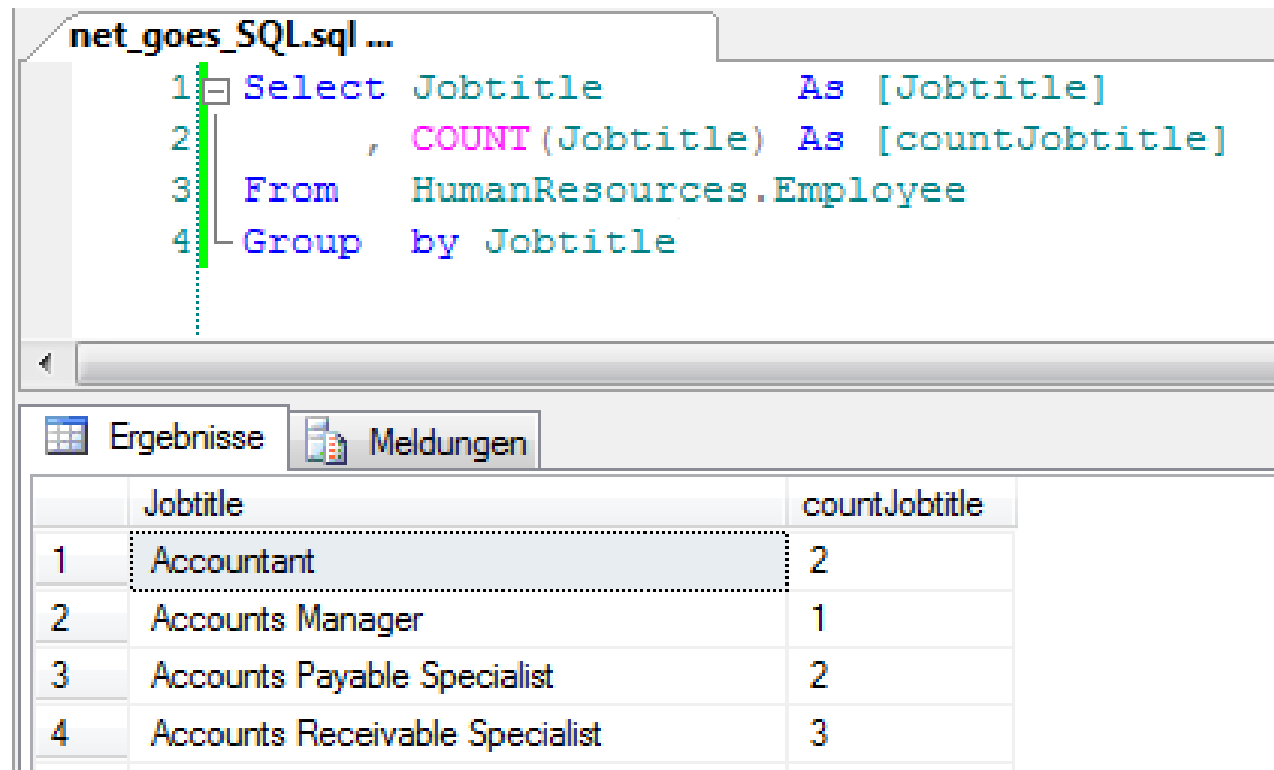
```
net_goes_SQL.sql ...
1 Select Jobtitle As [Jobtitle]
2 From HumanResources.Employee
```

Ergebnisse | Meldungen

	Jobtitle	countJobtitle
1	Accountant	
2	Accountant	
3	Accounts Manager	
4	Accounts Payable Specialist	
5	Accounts Payable Specialist	
6	Accounts Receivable Specialist	
7	Accounts Receivable Specialist	
8	Accounts Receivable Specialist	

.net goes SQL

2. Aggregat-Funktion (b)



The screenshot shows a SQL query editor window titled "net_goes_SQL.sql ...". The query is as follows:

```
1 Select Jobtitle As [Jobtitle]
2     , COUNT(Jobtitle) As [countJobtitle]
3 From   HumanResources.Employee
4 Group by Jobtitle
```

Below the query editor, there are two tabs: "Ergebnisse" (Results) and "Meldungen" (Messages). The "Ergebnisse" tab is active, displaying a table with the following data:

	Jobtitle	countJobtitle
1	Accountant	2
2	Accounts Manager	1
3	Accounts Payable Specialist	2
4	Accounts Receivable Specialist	3

.net goes SQL

VS Template Skalar-Funktion

```
Function1.cs X
UserDefinedFunctions
1 using System;
2   using System.Data;
3   using System.Data.SqlClient;
4   using System.Data.SqlTypes;
5   using Microsoft.SqlServer.Server;
6
7 public partial class UserDefinedFunctions
8 {
9     [Microsoft.SqlServer.Server.SqlFunction]
10    public static SqlString Function1()
11    {
12        // Fügen Sie hier Ihren Code ein.
13        return new SqlString("Hello");
14    }
15 };
16
```

.net

VS

```
Aggregat1.cs X
Aggregat1
1 using System;
2 using System.Data;
3 using System.Data.SqlClient;
4 using System.Data.SqlTypes;
5 using Microsoft.SqlServer.Server;
6
7
8 [Serializable]
9 [Microsoft.SqlServer.Server.SqlUserDefinedAggregate(Format.Native)]
10 public struct Aggregat1
11 {
12     public void Init()
13     {
14         // Fügen Sie hier Ihren Code ein.
15     }
16
17     public void Accumulate(SqlString Value)
18     {
19         // Fügen Sie hier Ihren Code ein.
20     }
21
22     public void Merge(Aggregat1 Group)
23     {
24         // Fügen Sie hier Ihren Code ein.
25     }
26
27     public SqlString Terminate()
28     {
29         // Fügen Sie hier Ihren Code ein.
30         return new SqlString("");
31     }
32
33     // Dies ist ein Platzhalter-Memberfeld.
34     private int var1;
35
36 }
```



.net

```
Aggregate1.cs X
Aggregate1
1 using System;
2 using System.Data;
3 using System.Data.SqlClient;
4 using System.Data.SqlTypes;
5 using Microsoft.SqlServer.Server;
6
7
8 [Serializable]
9 [Microsoft.SqlServer.Server.SqlUserDefinedAggregate(Format.Native)]
10 public struct Aggregate1
11 {
12     public void Init()
13     {
14         this.theCount = 0; // <<
15     }
16
17     public void Accumulate(SqlString Value)
18     {
19         if (Value.IsNull) // <<
20         {
21             this.theCount++;
22         }
23     }
24
25     public void Merge(Aggregate1 Group)
26     {
27         this.theCount += Group.theCount; // <<
28     }
29
30     public SqlInt32 Terminate()
31     {
32         return this.theCount; // <<
33     }
34
35     private int theCount;
36 }
```



.net goes SQL

nach der C#-Programmierung erfolgt die Einbindung in die SQL-Server Datenbank.

erst ist zu prüfen, ob in der SQL-Instance die CLR-Integration aktiviert ist.

dann erfolgt das laden der .dll

.net goes SQL

```
-- check ob 'clr enabled' = 1
```

```
Select *  
From sys.configurations  
Where name = 'clr enabled'
```

```
-----
```

```
-- sonst  
exec sp_configure 'clr enabled' , 1  
GO  
RECONFIGURE  
GO
```

.net goes SQL

Abhängig vom „Funktionsumfang“ der Programmierung folgt die Konfiguration des Permissionsets.

- **Safe**
 - **External_Access**
 - **Unsafe**
- 

.net goes SQL

Safe

Berechtigung	Beschreibung
SecurityPermission	Erlaubt die Ausführung von Managed Code.
SqlPermission	Zugriff auf die eig. DB mit der Verbindungszeichenfolge Context connection=true

.net goes SQL

External_Access (1/2)

Berechtigung	Beschreibung
Dist.TransactionPermission	Erlaubt verteilte Transactionen
DNS Permission	Namensauflösung u.Ä. Abfragen
Environment Permission	Vollständiger Zugriff auf die Umgebungsvariablen des Sicherheitskontextes und des Systems
EventLog Permission	OS Eventlog: Anlegen/Löschen einer Ereignisquelle und Lesen/Löschen von Protokollen
File IO Permission	Vollständiger Zugriff auf Dateien u. Ordner
KeyContainer Permission	Vollständiger Zugriff auf Schlüsselcontainer
Network Info Permission	z.B. Pings

.net goes SQL

External_Access (2/2)

Berechtigung	Beschreibung
Registry Permission	Auslesen der Registry
Smtp Permission	Verbindung zu SMT-Host über Port 25
Socket Permission	Verbindungsaufbau (alle Ports und alle Protokolle) zu anderen Computern via TCP/IP
SqlClient Permission	Zugriff auf alle DB's über Verbindungszeichenfolge. Local und remote
Store Permission	Vollständiger Zugriff auf X.509 Zertifikate
Web Permission	Zugriff auf Webressourcen. .net 2.0 Webservice und WCF

.net goes SQL

External Access oder Unsafe
→ Signierung der .dll

oder

```
ALTER DATABASE
```

```
N'<database_name, sysname, sample_database>'
```

```
SET Trustworthy ON
```

```
-- und
```

```
-- der Datenbankbesitzer (DBO) hat die
```

```
-- EXTERNAL_ACCESS ASSEMBLY-Berechtigung,
```

```
-- = Voraussetzung für Permission_Set = External_Access
```

.net goes SQL

```
-- laden der .dll
```

```
CREATE ASSEMBLY sql_fx_Aggregate  
FROM 'C:\folder\SqlServerProject1.dll'  
WITH Permission_Set = Safe
```

```
-----
```

```
-- überprüfen
```














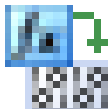
```
SELECT * FROM sys.assemblies ;  
SELECT * FROM sys.assembly_files ;  
SELECT * FROM sys.assembly_modules ;
```

.net goes SQL

-- erstellen der Aggregate-fx aus der Assembly

```
CREATE Aggregate countNull ( @Value nvarchar(4000) )  
RETURNS int  
EXTERNAL Name sql_fx_Aggregate.Aggregate1
```

.net goes SQL

-   Programmierbarkeit
 -   Gespeicherte Prozeduren
 -   Funktionen
 -   Tabellenwertfunktionen
 -   Skalarwertfunktionen
 -   Aggregatfunktionen
 -   `dbo.countNull`

.net goes SQL

T-SQL Anwendungsbeispiel:

```
1 Select vorname, nachname  
2 From   dbo.test_NullValues  
3
```

Ergebnisse

	vorname	nachname
1	Anton	Müller
2	Berti	Meier
3	Clausi	NULL
4	Daniel	NULL
5	Emil	NULL
6	NULL	NULL
7	NULL	Mustermann
8	NULL	Musterfrau
9	NULL	Musterkind

.net goes SQL

T-SQL Anwendungsbeispiel:


```
1 Select count (*)           As [count_All]  
2     , count (vorname)     As [countVorname]  
3     , count (nachname)    As [countNachname]  
4 From   dbo.test_NullValues
```



Ergebnisse



Meldungen

	count_All	countVorname	countNachname
1	9 	5	5

.net goes SQL

T-SQL Anwendungsbeispiel:

```
1 Select count (*)           As [count_All]
2       , count (vorname)    As [countVorname]
3       , dbo.countNull (vorname) As [countNull]
4 From   dbo.test_NullValues
```

Ergebnisse Meldungen

	count_All	countVorname	countNull
1	9 =	5 +	4

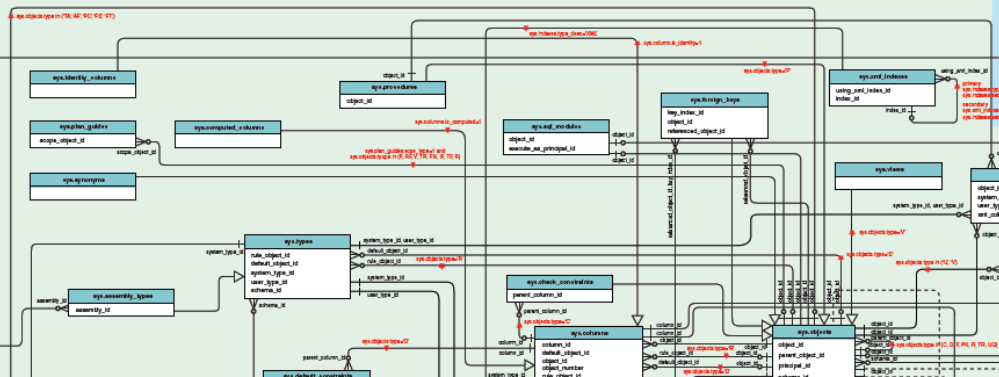
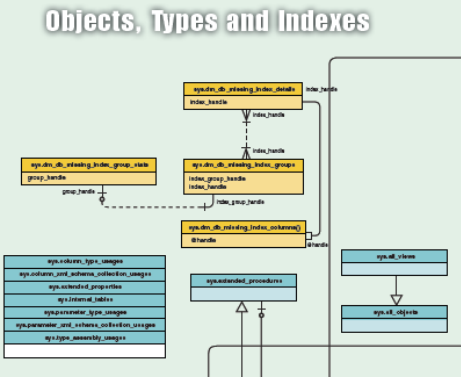
.net goes SQL

Überwachung mittels ...

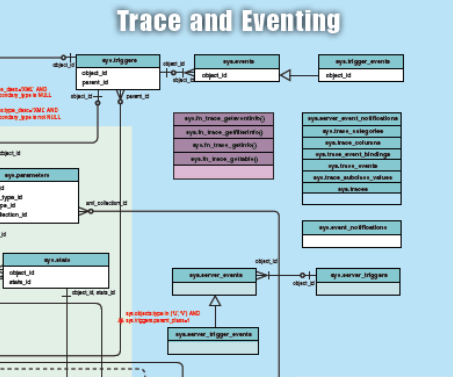


Microsoft® SQL Server® 2008 System Views

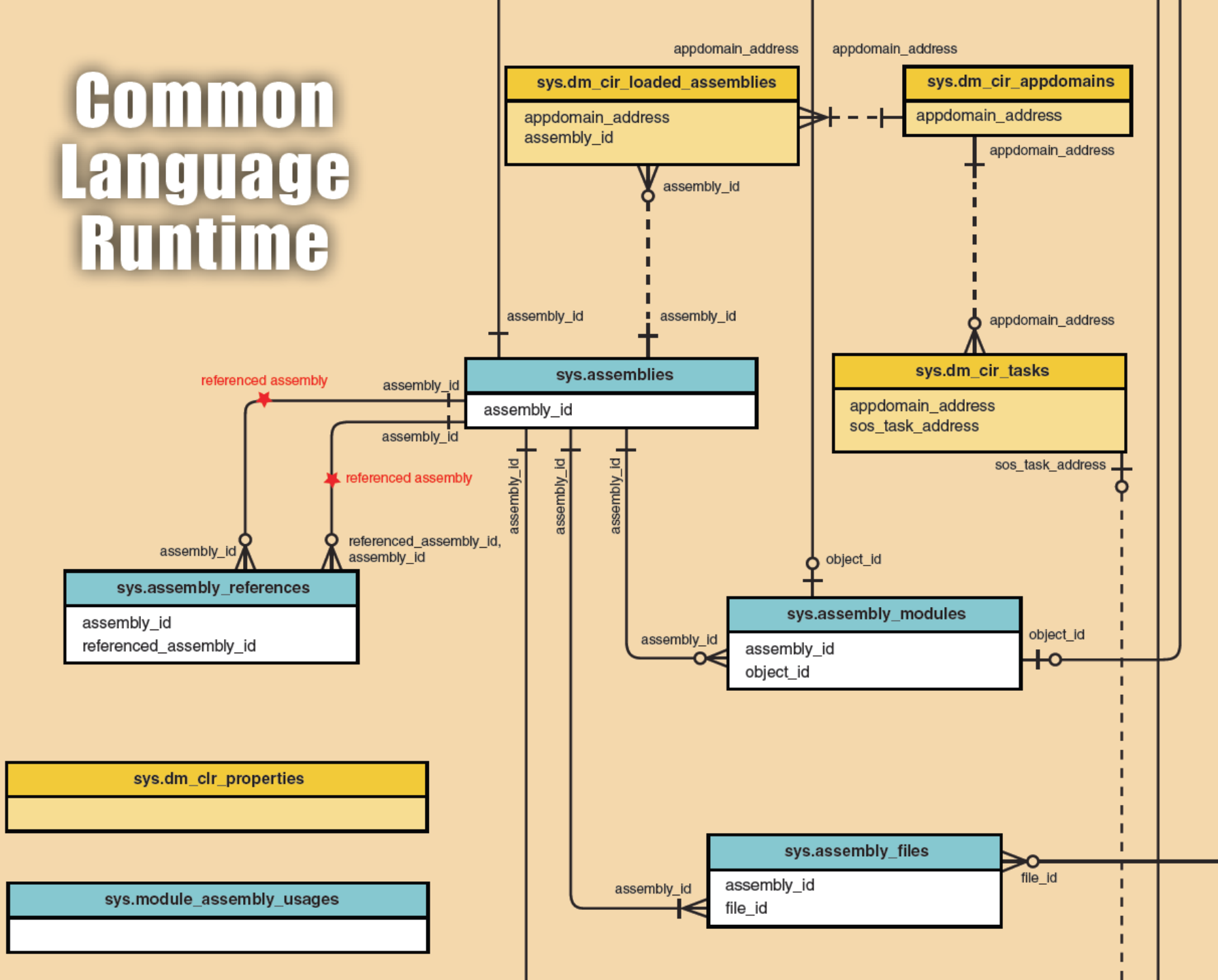
Objects, Types and Indexes



Trace and Eventing



Common Language Runtime



.net goes SQL

Demo



.net goes SQL

mittels CLR-Integration werden die Vorzüge aus beiden Welten (.net und T-SQL) insbesondere bei Funktionen und Stored Procs in der Datenbank verfügbar.

Für Aggregate gibt es in T-SQL keine Möglichkeit der Programmierung. Wie das vorliegende Beispiel zeigte, bietet dafür .net die Lösung.

.net goes SQL

vielen Dank.



.net goes SQL

links:

<http://www.net-CDE.de/netug>

[http://msdn.microsoft.com/de-de/library/91e6taax\(v=VS.90\).aspx#Y780](http://msdn.microsoft.com/de-de/library/91e6taax(v=VS.90).aspx#Y780)

[http://msdn.microsoft.com/de-de/library/ms187959\(v=SQL.100\).aspx](http://msdn.microsoft.com/de-de/library/ms187959(v=SQL.100).aspx)

<http://msdn.microsoft.com/de-de/library/ms179982.aspx>

<http://www.microsoft.com/download/en/details.aspx?id=9301>