

Online on-demand Project Support



SQL Vertiefung

CTEs , Unterabfragen, Window Functions & Co



Thorsten Kansy (tkansy@dotnetconsulting.eu)

Thorsten Kansy



Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor



Motivation

- Übersicht
- Grundlagen
- Wichtiges

- Spaß

Zeitliche Planung

9:00 Uhr bis 17:00 Uhr

Mittagessen

12:00 Uhr bis 13:00 Uhr

Agenda

- Common Table Expressions (CTEs)
- Unterabfragen
- Window Functions
- Analytische Funktionen
- Row Ranges
- Outer/ Cross Apply
- Sequenzen



A photograph of a building with a cat statue on its roof, overlaid with a dark blue banner containing the text 'Common Table Expressions (CTEs)'. The background shows a clear blue sky, green trees, and a power line.

Common Table Expressions (CTEs)

Common Table Expressions (CTEs)

- Temporär benannte Ergebnismengen/ Datenmengen
 - SELECT
 - Rekursion
 - Update-fähig
- Sind aber keine temporären Tabellen
- Mehrere CTEs können verkettet werden
- Anforderunge
 - Alle Spalten müssen eindeutig benannt sein
- Semikolon nicht vergessen!

Common Table Expressions (CTEs)

```
;WITH [cte] AS  
(  
    SELECT [pv].[id],  
           [cte1].[Fullname],  
           CAST([cte2].[NameLength] AS DECIMAL) AS [NameLength],  
           (SELECT [MaxLength] FROM [cte3]) AS [MaxLength]  
    FROM [dbo].[Personenverzeichnis] [pv]  
    LEFT JOIN [cte1] ON [pv].[ID] = [cte1].[ID]  
    LEFT JOIN [cte2] ON [pv].[ID] = [cte2].[ID]  
)  
SELECT * FROM [cte]  

```



Demo

Rekursion

```
WITH [cte]
AS
(
    -- Anker der Rekursion
    SELECT *, 0 AS [Level] FROM [dbo].[Personenverzeichnis] WHERE [VorgesetzterID] IS NULL

    UNION ALL

    -- Rekursion
    SELECT [pv].*, [cte].[Level] + 1 FROM [dbo].[Personenverzeichnis] [pv] INNER JOIN [cte]
    ON [cte].[ID] = [pv].[VorgesetzterID]
)
SELECT [ID], [Level] INTO [#temp] FROM [cte]
OPTION (MAXRECURSION 50); -- 100 ist der Standard
```



Demo



Unterabfragen



Unterabfragen

- Abfrage in Abfrage
- für Spalten
- für Datenquelle (FROM)
 - aka „abgeleitete Tabelle“ (derived Tables)
 - aka Tabellenausdrücke

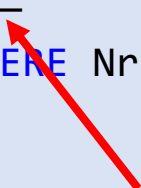
Spaltenquelle

```
SELECT *, -- Spalten wie gewöhnlich auswählen
        -- In Klammern kann die skalare Rückgabe einer Unterabfrage
        -- als Spalte verwendet werden
        (SELECT MAX(LEN([name]))
         FROM [dbo].[Personenverzeichnis])      AS 'Maximale Länge'
FROM [dbo].[Personenverzeichnis];
```

Demo

Datenquelle (FROM)

```
SELECT * FROM
(
    SELECT
    ROW_NUMBER() OVER (PARTITION BY [name] ORDER BY [id] ) AS 'Nr',
    *
    FROM [dbo].[Personenverzeichnis] [pv]
)
WHERE Nr = 1;
```



Demo



Window Functions



Window Functions

ROW_NUMBER ()	Nummerierung
RANK ()	Ranking
PERCENT_RANK ()	Prozentuales Ranking
DENSE_RANK ()	Ranking ohne Lücken bei mehrfach Belegung
NTILE (n)	Aufteilen in n (gleichgroße) Chargen (Bucketing)

Ab SQL Server 2022+

WINDOW Klausel	Benannte Abfragefenster für Vereinfachung von Abfragen

Window Functions

```
-- Absteigend nummerieren
```

```
SELECT ROW_NUMBER() OVER  
(ORDER BY [ID] DESC), * FROM [dbo].[Vertriebszahlen];
```

```
-- Nummerieren, pro Land separat
```

```
SELECT ROW_NUMBER() OVER  
(PARTITION BY [Land] ORDER BY [ID]), * FROM [dbo].[Vertriebszahlen];
```

Demo



Analytische Funktionen

Analytische Funktionen

<code>FIRST () / LAST ()</code>	Wert aus erster/ letzten Zeile eines Bereiches
<code>LAG () / LEAD ()</code>	Wert auf vorherigen/ nächsten Zeile

Analytische Funktionen

-- Sortiert nach Umsatz

```
SELECT LAST_VALUE([Umsatz]) OVER  
(ORDER BY Umsatz), *  
FROM [dbo].[Vertriebszahlen];
```

-- Sortiert nach Umsatz, pro Land

```
SELECT LAST_VALUE([Umsatz]) OVER  
(PARTITION BY Land ORDER BY Umsatz), *  
FROM [dbo].[Vertriebszahlen];
```

Demo



Row Ranges

Row Ranges

- Bereichsvorgabe für
 - Window Function
 - Analytische Funktionen
 - Aggregate

Row Ranges

```
-- Gleitender Durchschnitt 5 Zeilen vor und nach
SELECT
  [Name],
  [Umsatz],
  AVG([Umsatz])
  OVER (ORDER BY [Umsatz] DESC
        ROWS BETWEEN 5 PRECEDING AND 5 FOLLOWING)
FROM [dbo].[Vertriebszahlen]
ORDER BY [Umsatz] DESC;
```


Demo



Outer/ Cross Apply

OUTER/ CROSS APPLY

- „Join“ zwischen
 - Tabellenausdrücke
 - Tabellenwertfunktionen
- OUTER APPLY => LEFT JOIN
- CROSS APPLY => INNER JOIN

OUTER/ CROSS APPLY

```
SELECT TOP 10 [pv].*, [inner].[Name], [inner].[Vorname]  
FROM [dbo].[Personenverzeichnis] [pv]
```

OUTER APPLY

```
(  
    -- Hier kann ein kompletter Abfrage-Ausdruck stehen  
    SELECT * FROM [dbo].[Personenverzeichnis]  
        WHERE [Name] = [pv].[Vorname]  
    UNION ALL  
    SELECT * FROM [dbo].[Personenverzeichnis]  
        WHERE [Name] = [pv].[Vorname]  
) [inner]  
WHERE NOT [inner].[Vorname] IS NULL;
```


Demo



Sequenzen

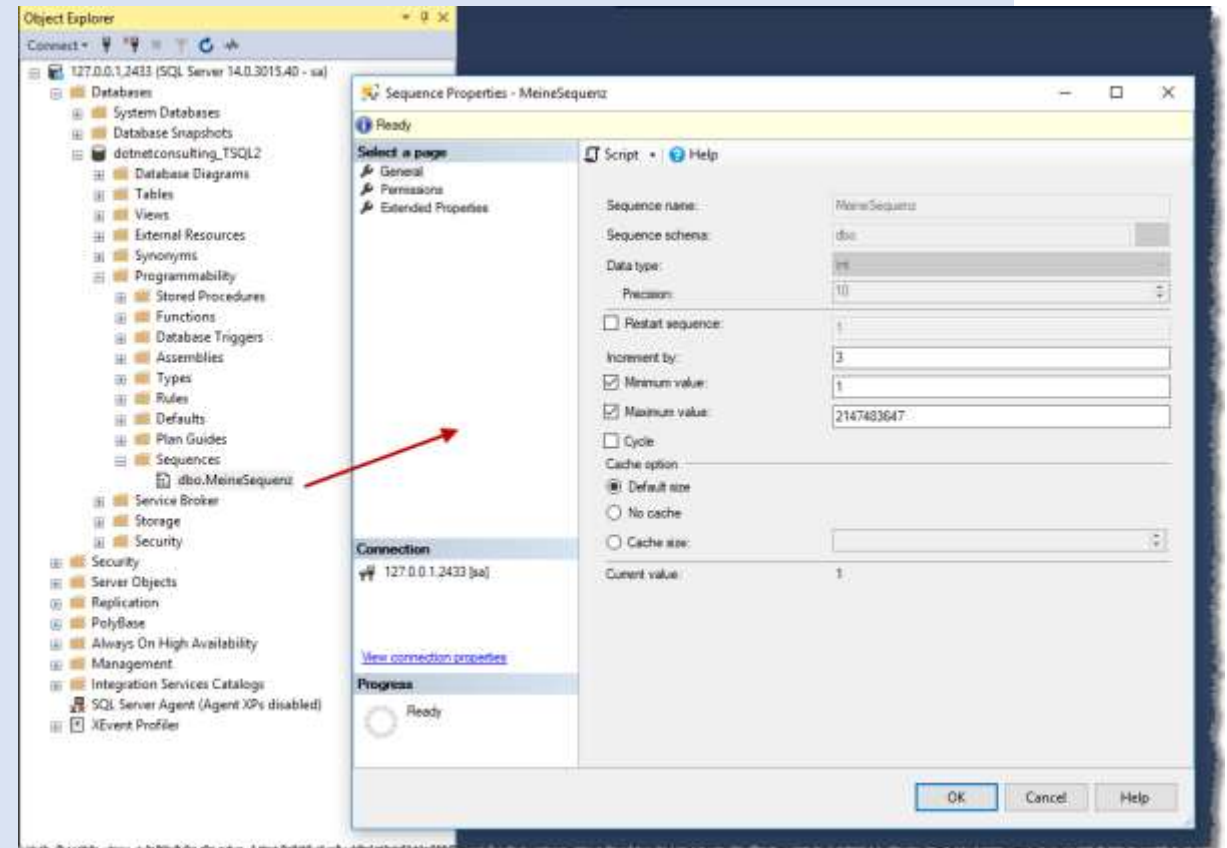


Sequenzen

- Numerische Zahlenkreis
 - INT, BIGINT, SMALLINT, etc.
 - Tabellen unabhängig
 - Transaktion unabhängig
- Entnahme eines Wertes
 - `FETCH NEXT FOR <SeqName>`
- Entnahme mehrerer (aufeinander folgedender) Werte
 - `EXEC sys.sp_sequence_get_range ...`

Sequenzen

```
CREATE SEQUENCE MeineSequenz  
AS INT  
MINVALUE 1  
NO MAXVALUE  
INCREMENT BY 3  
START WITH 1;
```



Demo

Fragen?

Links



<http://dotnetconsulting.eu/blog/>



[@Tkansy](https://twitter.com/Tkansy)



tkansy@dotnetconsulting.eu



www.dotnetconsulting.eu