



```
SELECT *
FROM [AdventureWorks2012].
[Production].[Product]
order by Name
offset (@von -1) rows
fetch next @bis - @von + 1 rows on-
ly;
```

THROW

Verwendet man in einem Try-Catch Block für die Fehlerbehandlung. Im Gegensatz zum bisherigen RAISERROR kann THROW die Fehler auch innerhalb eines CATCH Blocks werfen.

FIRST_VALUE und LAST_VALUE

Liefert den ersten oder den letzten Wert einer Ergebnismenge.

```
SELECT FIRST_VALUE(Name) OVER
(PARTITION BY ProductNumber ORDER BY
SellStartDate ASC)
```

```
FROM [AdventureWorks2012].
[Production].[Product]
```

```
SELECT LAST_VALUE(Name) OVER
(PARTITION BY ProductNumber ORDER BY
SellStartDate ASC)
```

```
FROM [AdventureWorks2012].
[Production].[Product]
```

LAG und LEAD

Mit der neuen Funktion LAG und LEAG ist man in der Lage, auf vorherige (LAG) und folgende Werte (LEAD) zuzugreifen. Um das zu bewerkstelligen muss zwingend eine Sortierung mit ORDER BY im OVER Zweig existieren, so ist auch die Reihenfolge der Zeilen definiert. Beide Funktionen haben mehrere Überladungen.

```
SELECT Name, LEAD(Name) OVER (order by
ListPrice ASC)
```

```
FROM [AdventureWorks2012].
[Production].[Product]
```

```
SELECT Name, LAG(Name) OVER (order by
ListPrice ASC)
```

```
FROM [AdventureWorks2012].
[Production].[Product]
```

CUME_DIST

Dient zur Ermittlung der kumulierten Verteilung einer numerischen Datenmenge.

PERCENT_RANK

RANK() und DENSE_RANK() konnte schon seit dem SQL Server 2005 verwendet werden, um die Position (eng. rank) einer Zeile in der Ergebnismenge ermitteln zu können. Der Wert war aber als absolute Zahl angegeben. Mit PERCENT_RANK() wird ein relativer Prozentwert als Position geliefert.

PERCTILE_CONT und PERCTILE_DISC

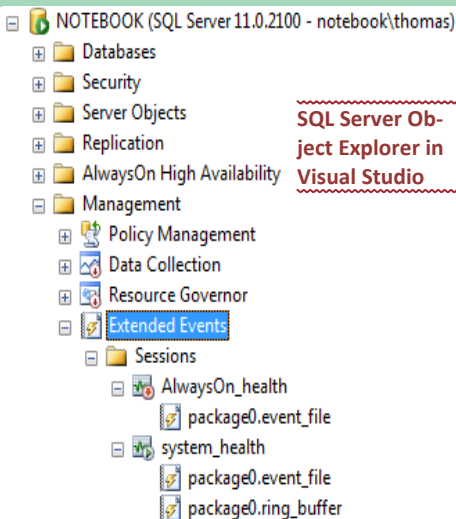
Damit kann der Quantilwert einer sortierten Zahlenreihe ermittelt werden.

Deprecated Database Engine Features in SQL Server 2012

Die detaillierte Information zu nicht mehr unterstützen Funktionen und den *breaking changes* zum SQL Server 2012 findet man auf Technet. <http://technet.microsoft.com/en-us/library/ms143729.aspx>

SQL Profiler und SQL Server Extended Events

Den SQL Profiler gibt es schon seit dem SQL Server 7.0. Ab SQL Server 2008 kamen die Extended Events hinzu. Sie dienen als Systemmonitor des SQL Servers, zum Troubleshooting und um einen generellen Einblick in die Abarbeitung der SQL Statements zu bekommen. Zudem können



SQL Server Object Explorer in Visual Studio

Kurzübersicht der wichtigsten Unterschiede

SQL Profiler	SQL Extended Events
Seit SQL Server 7.0	Ab SQL Server 2008
Ein Trace kann einen ausgelasteten (Produktions) Server „den Rest“ geben	Minimale Beeinträchtigung der Prozesse bei der Überwachung
	Unterstützung vorhandener ETW (Event Tracing for Windows) Tools
	PowerShell Unterstützung

Sie so Dead Locks und nicht performanten SQL Queries auf den Grund kommen.

Welches Tool kommt wann zum Einsatz?

Extended Events ist das neuere Tool für die Diagnose. Es führt zu wesentlich besserer Performance auf dem Server. Die Skalierbarkeit bietet mehr Funktionen und Vorteile als sie beim SQL Trace des SQL Profiles geboten werden. Mit dem SQL Server 2012 wurden die Extended Events gegenüber 2008 und 2008R2 weiter ausgebaut. So gesehen sind Extended Events die Zukunft, der SQL Profiler hat seine Daseinsberechtigung noch aus Kompatibilitätsgründen.

Fazit

Der SQL Server 2012 bietet eine Reihe neuer Funktionen und Befehle. Für den einen oder anderen ist sicherlich einen interessante Neuerung dabei. Es macht also Sinn, bei seinem bestehenden T-SQL Anwendung einen Frühjahrsputz zu machen und bestehende, selbst angefertigte Konstruktionen, durch die neu angebotenen zu ersetzen.



ClubComputer.at

... begleitet Dich in die digitale Zukunft.

*gemeinsam
innovative
Technologien
entdecken*



www.clubcomputer.at
buero@clubcomputer.at