Tabellen erreichen. Aber leider mit dem Tribut, auf den darunter liegende Table nur im *readonly* modus zugreifen zu können. Im SQL Server 2014 wurde diese Limitierung aufgehoben. Mit dem *clustered columsstore index* kann diese Performance gehalten werden, der darunter verwendete Table kann mit allen DML Operationen *(insert, update, delete)* betrieben werden.

Systemmonitor nutzen um deprecated features am SQL Server anzuzeigen

Mit dem Start von *perfmon.exe* – dem Windows Systemmonitor, werden eine Vielzahl von Überwachungsmöglichkeiten des Systems angeboten. Neben Messungen des Betriebssystems wird hier auch eine Menge für den SQL-Server angeboten, beginnend von *Alerts, Jobs, Statistics, über Broker, Locks und Memory, Transaction* lässt sich hiermit auch die Verwendung von *Deprecated Features* aufspüren.

Datentyp	Wertebereich von	bis	Info
date	0001-01-01	9999-12-31	ab 2008 Auf 1 Tag genau
time	00:00:00.0000000	23:59:59.9999999	ab 2008 auf 100 ns genau
datetime	1753-01-01 00:00:00	9999-12-31 23:59:59:997	Auf ~3⅓ Millisekunden genau (.000, .003, .007, .010) Obsolete ab 2012
datetime2	0001-01-01	9999-12-31	auf 100ns genau ab 2008
datetimeoffset	0001-01-01 00:00:00	9999-12-31 23:59:9999999	auf 100ns genau ab 2008
smalldatetime	1900-01-01 00:00	2079-06-06 23:59	auf 1 Minute genau Obsolete ab 2012

Festlegen der Sprache am SQL Server

Die Liste der möglichen Sprachen erhält man mittels

```
Select * from sys.syslanguages
```

Will man die Deutsche Auslegung des Datums, setzt man

```
SET LANGUAGE German;
```

Überblick über die Kategorien von Datentypen am SQL Server

SQL Server Datum und Zeit [2]

Warum wurde gerade 1753-01-01 als Beginn Wert des Datentyps `datetime` gewählt?

Das hat historische Gründe, es gibt den Julianischen und den Gregorianischen Kalender. Die beiden Kalender differieren 10 bis 13 Tage, abhängig vom Jahrhundert. Die Kulturen wechselten den Kalender zu unterschiedlichen Zeitpunkten. So wechselte Großbritannien am 02.09.1752, durch die Differenz der Tage folgte der nächste Tag der 14.09.1752. Sybase hatte damals beschlossen, 1753 als frühestens Datum zu verwenden, um diese unterschiedliche Berechnungslogik früherer Datumswerte nicht durchführen zu müssen. SQL Server tat es ebenso. Andere Länder haben diesen Kalenderwechsel erst später vollzogen, die Türkei etwa erst 1927.

Beim Verwenden der Daten ist man oft mit Konvertierungen konfrontiert, eine weitere Fehlerquelle, die nur bei bestimmen Tagen des Jahres auftritt, wenn datetime als Cast Typ verwendet wird.

```
SET LANGUAGE British;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 107) AS British;
SET LANGUAGE US_English;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 107) AS US;
SET LANGUAGE German
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 104) AS German;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 102) AS ANSI;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 102) AS ISO;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 126) AS ISO8601;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 127) AS ISO8601WithTimezone;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 113) AS EuropeDefault;
```

British	US	German	ANSI	ISO	ISO8601	ISO8601 WithTimezone	Europe Default-Date
Dec 09, 2013	Sep 12, 2013	09.12.2013	2013.12.09	2013.12.09	2013-12-09T00:00:00	2013-12-09T00:00:00	09 Dez 2013 00:00:00

```
SET LANGUAGE German;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime), 107) AS DateOnlyDate;
SELECT CONVERT(varchar(20), CAST('2013-09-12' AS datetime2), 107) AS DateOnlyDate2;
SELECT CONVERT(varchar(20), CAST('2013-09-12T00:00:00' AS datetime), 107) AS ISODateTimeDate;
SELECT CONVERT(varchar(20), CAST('2013-09-12T00:00:00' AS datetime), 104) AS GermanDateTimeDate;
```

Wird `datetime2` beim Cast bei einem Wert verwendet, bei dem die Zeitangabe ohne `T` formatiert ist, ist das Resultat korrekt: