

```
-- Author: Philipp Lenz
-- Mail: philipp.lenz@flip-it.de
-- Datum: 13/03/2011
--
-- Beschreibung:
-- Erzeugt eine Datenbank (Produktion) in der eine Buchungs-Tabelle eingefügt wird.
--
-- Achtung: Verwendung auf eigene Gefahr!

-- Datenbank Produktion erstellen
CREATE DATABASE [Produktion] ON PRIMARY
( NAME = N'Produktion',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\Produktion.mdf' , SIZE =
3072KB , FILEGROWTH = 10%)
LOG ON
( NAME = N'Produktion_log',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\Produktion_log.ldf' ,
SIZE = 1024KB , FILEGROWTH = 10%)
GO

-- Datenbank Produktion verwenden
USE Produktion
GO

-- Dateigruppen hinzufügen für die Partitionierung
ALTER DATABASE [Produktion] ADD FILEGROUP [Archiv2009]
GO
ALTER DATABASE [Produktion] ADD FILEGROUP [Archiv2010]
GO
ALTER DATABASE [Produktion] ADD FILEGROUP [Archiv2011]
GO
ALTER DATABASE [Produktion] ADD FILEGROUP [Archiv]
GO

-- Dateien zu den Dateigruppen hinzufügen
ALTER DATABASE [Produktion] ADD FILE (
  NAME = N'2009',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\2009.ndf' ,
  SIZE = 3072KB ,
  FILEGROWTH =
```

```

1024KB ) TO FILEGROUP [Archiv2009]
GO
ALTER DATABASE [Produktion] ADD FILE (
    NAME = N'2010',
    FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\2010.ndf' ,
    SIZE = 3072KB ,
    FILEGROWTH = 1024KB ) TO FILEGROUP [Archiv2010]
GO
ALTER DATABASE [Produktion] ADD FILE (
    NAME = N'2011',
    FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\Aktuell.ndf' ,
    SIZE = 3072KB ,
    FILEGROWTH = 1024KB ) TO FILEGROUP [Archiv2011]
GO
ALTER DATABASE [Produktion] ADD FILE (
    NAME = N'Archiv',
    FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\Archiv.ndf' ,
    SIZE = 3072KB ,
    FILEGROWTH = 1024KB ) TO FILEGROUP [Archiv]
GO

-- Partitionierungsfunktion erstellen
CREATE PARTITION FUNCTION Archivierungsfunktion (DATETIME) AS RANGE RIGHT
    FOR VALUES (
        '20090101' -- Alles was kleiner ist als der 01.01.2009 geht in Archiv
        , '20100101' -- Alles was kleiner ist als der 01.01.2010, jedoch größer als der 01.01.2009 geht in
Archiv2009
        , '20110101' -- Alles was kleiner ist als der 01.01.2011, jedoch größer als der 01.01.2010 geht in
Archiv2010
        -- Alles was größer ist als der 01.01.2011 geht in Archiv2011
    );
GO

-- Partitionierungsschema erstellen
CREATE PARTITION SCHEME Archivierungsschema AS PARTITION Archivierungsfunktion
    TO ('Archiv', 'Archiv2009', 'Archiv2010', 'Archiv2011');
GO

-- Buchungstabelle erstellen
CREATE TABLE dbo.buchungen (
    [uid] uniqueidentifier default NEWID() NOT NULL,

```

```

        [name] VARCHAR(50),
        [datum] DATETIME DEFAULT GETDATE() NOT NULL,
        [wert] MONEY) ON ArchivierungsSchema(datum);
GO
ALTER TABLE dbo.buchungen ADD CONSTRAINT
    PK_buchungen PRIMARY KEY CLUSTERED
    ([uid], [datum]);
GO

-- Insert in Archiv
INSERT INTO dbo.buchungen (name, datum, wert) VALUES ('Januar 2008', '20080308', '100');
GO
-- Insert in Archiv2009
INSERT INTO dbo.buchungen (name, datum, wert) VALUES ('Januar 2009', '20090308', '100');
GO
-- Insert in Archiv2010
INSERT INTO dbo.buchungen (name, datum, wert) VALUES ('Februar 2010', '20100209', '100');
GO
-- Insert in Archiv2011
INSERT INTO dbo.buchungen (name, datum, wert) VALUES ('Januar 2011', '20110120', '100');
GO

-- Datenverteilung in den Partitionen abfragen
SELECT
    sys.filegroups.name 'Partitionsname'
    , sys.partitions.rows 'Zeilenanzahl'
FROM
    sys.partitions INNER JOIN sys.allocation_units
    ON sys.allocation_units.container_id = sys.partitions.hobt_id
    INNER JOIN sys.filegroups ON sys.filegroups.data_space_id = sys.allocation_units.data_space_id
    WHERE sys.partitions.object_id = OBJECT_ID('dbo.buchungen')
GO

-- Problem: Neues Geschäftsjahr:
-- Insert in Archiv2011
INSERT INTO dbo.buchungen (name, datum, wert) VALUES ('Januar 2012', '20120115', '300');
GO

-- Alle 2011er Daten in eine eigene Partitions verschieben
-- Alles was nach 2011 kommt, geht weiterhin in Aktuell

```

```
-- Neue Dateigruppe für 2012
ALTER DATABASE [Produktion] ADD FILEGROUP [Archiv2012]
GO

-- Neue Datei für die Dateigruppe 2011
ALTER DATABASE [Produktion] ADD FILE (
    NAME = N'2012',
    FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.VERTEILER\MSSQL\DATA\2012.ndf' ,
    SIZE = 3072KB ,
    FILEGROWTH =
    1024KB ) TO FILEGROUP [Archiv2012]
GO

ALTER PARTITION SCHEME ArchivierungsSchema
    NEXT USED [Archiv2012]
GO
ALTER PARTITION FUNCTION ArchivierungsFunktion ()
    SPLIT RANGE ('20120101');
GO
```

(c) Philipp Lenz, 13/02/2010