```
-- Author: Philipp Lenz
-- Mail: philipp.lenz@flip-it.de
-- Datum: 20/02/2011
--
-- Beschreibung:
-- Erzeugt eine Datenbank in der eine Personal-Tabelle eingefügt wird.
-- Die schützenswerten Daten sollen verschlüsselt werden, dies wird mit einem symetrischen Schlüssel
-- über ein Zertifikat und einem Masster Schlüssel realisiert.

-- Achtung: Verwendung auf eigene Gefahr!

-- Datenbank erzeugen
CREATE DATABASE PersonalDaten;
GO
-- Auf Datenbank verbinden
USE PersonalDaten;
GO
-- Master Schlüssel erzeugen auf welcher dann der Schlüssel verweist
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Kennwort1!';
GO
-- Personal Schema erzeugen in der sich die schützenswerten Daten befinden werden
CREATE SCHEMA personal;
GO

-- Personal Tabelle erzeugen, schützenswerte Daten
CREATE TABLE personal.PersonalDaten (
      [uid] uniqueidentifier DEFAULT NEWID() NOT NULL,
      [personalnr] int IDENTITY (1,1) NOT NULL,
      [name] varchar(255) NOT NULL,
      [gehalt] varbinary(256) NOT NULL,
      [kreditkartennr] varbinary(256) NULL,
      CONSTRAINT PK_PersonalDaten PRIMARY KEY CLUSTERED (
            [uid] ASC
      ) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY])
      ON [PRIMARY]
GO

-- Zertifkat: Symmetrischer Schlüssel wird dadurch geschützt
CREATE CERTIFICATE PersonalDatenZertifikat
      WITH SUBJECT = 'Firmen Zertifikat',
```

```
      START_DATE = '1/1/2011',
      EXPIRY_DATE = '01/12/2011';
GO

-- Symmetrischen Schlüssel erzeugen
CREATE SYMMETRIC KEY PersonalDatenSchluessel
      WITH ALGORITHM = TRIPLE_DES
      ENCRYPTION BY CERTIFICATE PersonalDatenZertifikat;
GO

-- VIEW erzeugen um auf die Daten zuzugreifen
CREATE VIEW dbo.[PersonalDaten]
      AS
      SELECT
            [personalnr],
            [name],
            CONVERT(MONEY, DecryptByKeyAutoCert(CERT_ID('PersonalDatenZertifikat'), NULL, gehalt)) AS gehalt,
            CONVERT(VARCHAR, DecryptByKeyAutoCert(CERT_ID('PersonalDatenZertifikat'), NULL, kreditkartennr)) AS
kreditkartennr
      FROM personal.PersonalDaten
GO

-- Schlüssel öffnen um Daten lesen oder schreiben zu können
OPEN SYMMETRIC KEY PersonalDatenSchluessel DECRYPTION BY CERTIFICATE PersonalDatenZertifikat;

-- 2 Datensätze einfügen mit der Verwendung des Schlüssels
INSERT INTO personal.PersonalDaten (name, gehalt, kreditkartennr)
      VALUES (
      'Philipp',
      ENCRYPTBYKEY(KEY_GUID('PersonalDatenSchluessel'), CONVERT(VARBINARY(256), €2000)),
      ENCRYPTBYKEY(KEY_GUID('PersonalDatenSchluessel'), '1234-5678-91277')
      ),
      (
      'Walter',
      ENCRYPTBYKEY(KEY_GUID('PersonalDatenSchluessel'), CONVERT(VARBINARY(256), €3000.52)),
      ENCRYPTBYKEY(KEY_GUID('PersonalDatenSchluessel'), '9876-6539-0815')
      );

-- Schlüssel schliessen
CLOSE SYMMETRIC KEY PersonalDatenSchluessel;
```

```
GO

-- Daten auslesen
-- Schlüssel wieder öffnen
OPEN SYMMETRIC KEY PersonalDatenSchluessel DECRYPTION BY CERTIFICATE PersonalDatenZertifikat;
-- Daten aus der Sicht auslesen
SELECT * FROM dbo.PersonalDaten ORDER BY personalnr DESC
-- Schlüssel schliessen
CLOSE SYMMETRIC KEY PersonalDatenSchluessel;
GO
```