**Aktionsabfragen**

Aktionsabfragen verändern also den Zustand der Daten in der Datenbank im Gegensatz zu Abfragen wie SELECT, die lediglich Daten auslesen. Bestimmte Aktionsabfragen verändern auch die Attribute oder sogar die Existenz von Datenbankobjekten.

**INSERT**

INSERT Produkt(Name, Preis)

VALUES ('Heft', 0.80)

INSERT Produkt(Name, Preis)

SELECT 'Blumenvase', 29.90

**Wenn in alle Felder geschrieben wird kann man die Auflistung der Feldnamen auslassen.:**

INSERT Produkt

VALUES('Papierkorb', 9.90)

INSERT Produkt

SELECT 'Aktenordner', 14.00

INSERT Produkt

SELECT TOP 1 [ProductName], p2.Preis

FROM [Northwind]..[Products] p1

INNER JOIN [dbo].[Produkt] p2 ON p1.ProductID = p2.ProduktId

**Mehrere Datensätze in einem Tusch**

INSERT Produkt

VALUES

('Gutes Buch', 34.90),

('Klebeband', 5.50),

('Bleistift', 0.20)

INSERT Produkt

SELECT 'Gutes Buch', 34.90

UNION ALL

SELECT 'Klebeband', 5.50

UNION ALL

SELECT 'Bleistift', 0.20

**Einfügen aus einer anderen Datenquelle (mit Bedingung)**

DECLARE @NeuesProdukt TABLE(NeuesProduktId INT IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(50), Preis MONEY)

INSERT @NeuesProdukt

VALUES

('Filzstift blau', 1.30),

('Filzstift gelb', 1.30),

('Filzstift grün', 1.30)

INSERT Produkt

SELECT Name, Preis From @NeuesProdukt

WHERE Name <> 'Filzstift blau'

**Abwandlungen (Typ: Andere Datenquelle)**

INSERT Produkt

SELECT **TOP 1 PERCENT** Name, Preis From @NeuesProdukt

INSERT Produkt

SELECT **TOP 1** Name, Preis From @NeuesProdukt

**Wenn man Daten einer Tabelle als Sicherheitskopie in einer Tabelle ablegen will.**

1. Es erstellt eine neue Datentabelle
2. Es insertet darin Daten aus der Quelltabelle

SELECT \*

INTO ProduktKopie

FROM Produkt

**UPDATE**

UPDATE Produkt

SET

Preis = 1.40

WHERE

ProduktId = 110

**Mit anderer Datenquelle**

DECLARE @ProduktUpdate TABLE(ProduktId INT PRIMARY KEY, Name VARCHAR(50), Preis MONEY)

INSERT @ProduktUpdate

VALUES

(107, 'Filzstift blau', 0.30),

(108, 'Filzstift gelb', 0.30),

(109, 'Filzstift grün', 0.30)

(Auch Alias erklären!)

UPDATE Produkt

SET

Preis = PU.Preis

FROM

@ProduktUpdate PU

WHERE

Produkt.ProduktId = PU.ProduktId

SELECT \* FROM Produkt WHERE ProduktId IN(107,108,109)

**Mit allerlei Operatoren, wie:**

UPDATE Produkt

SET

Preis = Preis \* 0.2

**DELETE, TRUNCATE**

CREATE TABLE #ProduktTemp(ProduktTempId INT IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(50), Preis MONEY)

INSERT #ProduktTemp

VALUES

('Gutes Buch', 34.90),

('Klebeband', 5.50),

('Bleistift', 0.20)

SELECT \* FROM #ProduktTemp

DELETE #ProduktTemp

WHERE Name = 'Bleistift'

SELECT \* FROM #ProduktTemp

**Mit TRUNCATE (ist schneller, performante, wenn man alle Datensätze löschen will)**

TRUNCATE TABLE #ProduktTemp

SELECT \* FROM #ProduktTemp

**DROP TABLE**

GANZE Tabelle (samt Definitionen und Daten) wird aus der Datenbank entfernt.

CREATE TABLE ProduktTemp(ProduktTempId INT IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(50), Preis MONEY)

INSERT ProduktTemp

VALUES

('Gutes Buch', 34.90),

('Klebeband', 5.50),

('Bleistift', 0.20)

DROP TABLE ProduktTemp

**Änderungen an Datenbanktabellen**

**Hinzufügen von neuer Spalte**

ALTER TABLE Kunde

ADD Telefonnummer varchar(20)

**Hinzufügen und Löschen von Spalte**

ALTER TABLE Kunde

DROP COLUMN Telefonnummer2

**Ändern von Feldattributen** (nützliche, nachträgliche Modifikation)

ALTER TABLE Kunde

ALTER COLUMN Telefonnummer VARCHAR(15)

**Hinzufügen von Einschränkung mit Standardwert** (nützliche, nachträgliche Modifikation)

ALTER TABLE Bestellung ADD CONSTRAINT

DF\_Bestellung\_Zeitstempel DEFAULT GETDATE() FOR Zeitstempel

**Entfernen von Standardwert**

ALTER TABLE Bestellung DROP CONSTRAINT DF\_Bestellung\_Zeitstempel