# DDL

## CREATE

### Datenbank erstellen

USE master;  
DROP DATABASE IF EXISTS Datenbankname;

CREATE DATABASE Datenbankname

### Schema erstellen

USE Datenbankname;  
GO  
CREATE SCHEMA Schemenname;

### Tabelle erstellen

CREATE TABLE [schema.]Tabellenname (  
id DATENTYP NOT NULL IDENTITY(1,1),  
feld1 DATENTYP);

## ALTER

### Spalte hinzufügen

ALTER TABLE [schema.]Tabellenname ADD neuer Feldname DATENTYP;

### Spalte verändern

#### Datentyp verändern

ALTER TABLE [schema.]Tabellenname ALTER COLUMN Feldname DATENTYP;

#### Spalte entfernen

ALTER TABLE [schema.]Tabellenname DROP COLUMN Feldname;

## DROP

DROP TABLE Tabellenname;

## TRUNCATE

TRUNCATE TABLE [schema.]Tabellenname;

## Schlüsselfelder

### Primärschlüssel

Innerhalb des CREATE Statements definieren:

CREATE TABLE Tabellenname (  
id DATENTYP PRIMARY KEY);

Nachträglich als ALTER Statement:

ALTER TABLE Tabellenname   
ADD CONSTRAINT Primärschlüssel Name PRIMARY KEY (Feldname);

Über mehrere Spalten einen Primärschlüssel definieren:

CREATE TABLE Tabellenname (  
feld1 DATENTYP NOT NULL,  
feld2 DATENTYP NOT NULL,   
feld3 DATENTYP  
CONSTRAINT Primary Key Name PRIMARY KEY (feld1, feld2)  
);

Nachträglich als ALTER Statement:

ALTER TABLE tabelle3 ADD CONSTRAINT PK\_tab3 PRIMARY KEY (id1, id2);

### Fremdschlüssel

Nachträglich als ALTER Statement:

ALTER TABLE Tabellenname   
ADD CONSTRAINT FK\_Name FOREIGN KEY (Feldname)   
REFERENCES Tabellenname (Feldname);

Beim Erstellen der Tabelle:

CREATE TABLE Tabellenname (  
feld1 DATENTYP NOT NULL,,,  
feld2 DATENTYP NOT NULL,   
feld3 DATENTYP  
FOREIGN KEY (Spaltenname) REFERENCES Tabellenname (Spaltenname)  
);

### Weitere Constraints

#### NOT NULL

ALTER TABLE Tabellenname   
ALTER COLUMN Feldname DATENTYP NOT NULL;

#### DEFAULT

ALTER TABLE Tabellenname   
ADD DEFAULT 'Default Wert' FOR Feldname;

#### CHECK

ALTER TABLE Tabellenname ADD CHECK (Feldname > 100);

# Datentypen

Datentyp Beschreibung

Text

Description automatically generated

# DML

## INSERT

INSERT [INTO] tabelle (spalte1, spalte2) VALUES (wert1, wert2);

INSERT INTO tabellenname VALUES

(zeile1-wert1, zeile1-wert2),

(zeile2-wert1, zeile2-wert1);

## UPDATE

UPDATE tabelle

SET spalte1 = wert1, spalte2 = wert2,...

[WHERE kondition];

## DELETE

DELETE FROM tabelle WHERE id = 1;

# DQL

## SELECT

SELECT spalte1, spalte2 FROM tabelle ORDER BY spaltenname [DESC];

## JOIN

SELECT e.spalte 1, z.spalte1 FROM tabelleEins e INNER JOIN tabelleZwei z ON e.spalte1 = z.spalte1;

## Gruppierungen

### Aggregatsfunktionen

SUM(), MIN(), MAX(), COUNT(), AVG(), STEDV(), VAR()

### GOUP BY

Fasst alle identischen Werte innerhalb einer Spalte zusammen. Die Gruppenbildung kann auch auf mehrere Spalten angewandt werden.

SELECT spalte1, COUNT(spalte2) FROM tabelle GROUP BY spalte1;

SELECT spalte1, spalte2, SUM(spalte3) FROM tabelle GROUP BY spalte1, spalte2;