

# LU04a - Vom Modell zur Tabelle

In dieser Lerneinheit bauen wir Schritt für Schritt die Brücke von der **Analyse** bis zur **Umsetzung einer Datenbank**. Wir haben bereits das ERD (Entity Relationship Diagram) in Form der **Chen-Notation** (Entitäten als Rechtecke, Attribute als Ovale etc.) kennengelernt, mit der man auf fachlicher Ebene Entitäten, Attribute und Beziehungen beschreiben kann. Nun gehen wir weiter in Richtung **logisches Schema** und **SQL-Umsetzung**.

- Wir vergleichen das konzeptionelle und logische Schema – ähnlich wie beim Bau eines Hauses: zuerst die Analyse und der Bauplan, dann die technische Umsetzung.
- Wir lernen die **Crow's-Foot-Notation** kennen, die sich für das logische Schema eignet: Tabellen mit Schlüsseln, Attributen und Kardinalitäten.
- Wir schauen uns an, wie man **Datentypen** passend wählt, damit die Datenbank robust, effizient und praxistauglich ist.
- Am Ende setzen wir ein kleines Beispiel **in SQL (DDL)** um: Wir erstellen Datenbanken, Tabellen mit Primär- und Fremdschlüsseln und führen erste Abfragen aus.

Damit verstehen wir, wie ein **Entity-Relationship-Modell (ERM)** Schritt für Schritt zu einer funktionierenden **relationalen Datenbank** wird – und legen die Grundlage für alle weiteren Arbeiten mit SQL.

## Konzeptionelles vs. Logisches Schema



*Eine Datenbank zu designen ist wie ein Haus zu bauen – ohne durchdachten Bauplan geht es nicht.*

*Erst die Planung, dann die Umsetzung (in SQL).*

Stellen wir uns vor, wir sollen ein **Haus bauen**. Bevor die Handwerker loslegen, passiert einiges:

- Zuerst sprechen wir mit der Bauherrschaft: **Was soll das Haus können?** Wie viele Zimmer? Soll eine Garage dabei sein? Welche Vorstellungen gibt es vom Budget oder vom Stil?
- Danach macht die Architektin oder der Architekt einen **Bauplan**: Räume, Türen, Fenster, Treppen, vielleicht auch schon, wo Strom- oder Wasserleitungen verlaufen.
- Erst im letzten Schritt kommen die Handwerker:innen und legen genau fest, **welche Materialien** wo eingesetzt werden (z. B. Beton für tragende Wände, Holz für Türen, Kabel für Strom).

Genau so funktioniert auch das **Design einer Datenbank**:

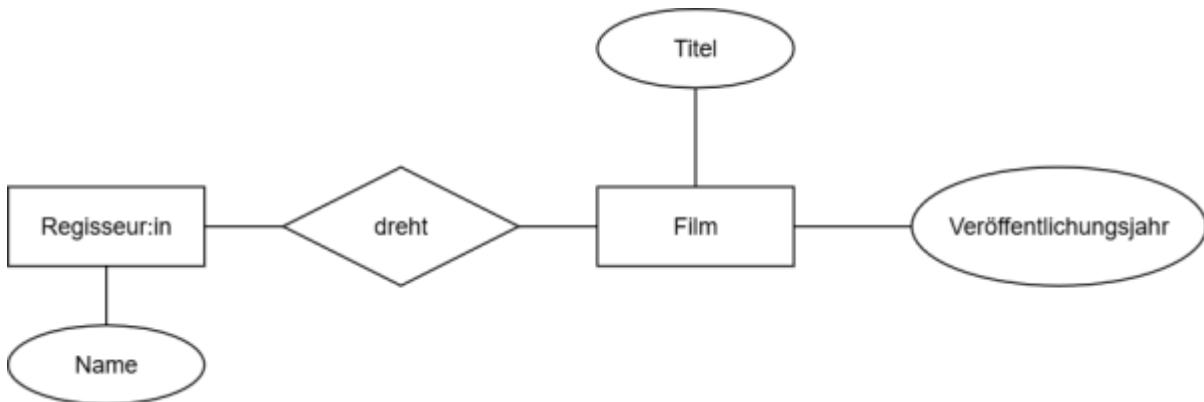
- Zuerst klären wir: **Welche Informationen sollen gespeichert werden?** (fachliche Sicht)
- Dann machen wir einen **Plan**: Welche Objekte gibt es und wie hängen sie zusammen?
- Am Ende folgt die **technische Umsetzung**: Tabellen, Primär- und Fremdschlüsse, Datentypen.

## 1. Konzeptionelles Schema

Das **konzeptionelle Schema** ist wie der Bauplan: Es zeigt die Objekte (Entitäten), ihre Eigenschaften (Attribute) und die Beziehungen – aber noch ohne technische Details.

Beispiel:

- Entität **Film** (Attribute: Titel, Jahr)
- Entität **Regisseur** (Attribute: Name)
- Beziehung: dreht zwischen Film und Regisseur



Charakteristik:

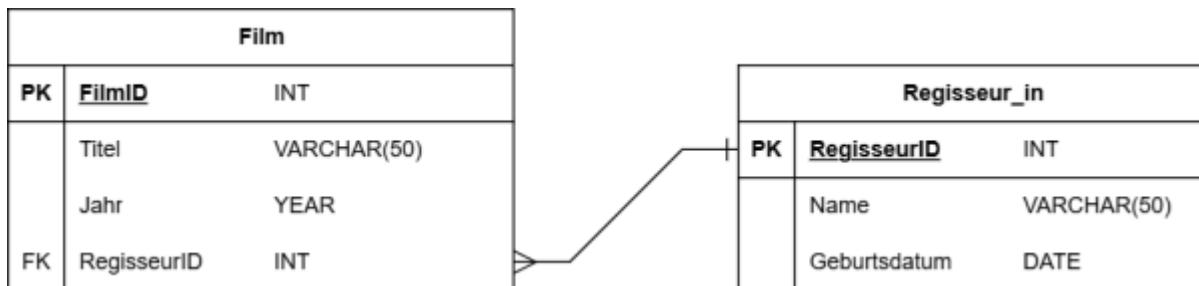
- Fokus: **Welche Objekte gibt es und wie hängen sie zusammen?**
- Nur grobe Attribute sichtbar
- Ideal, um mit Anwender:innen oder Auftraggeber:innen zu sprechen
- Hierfür eignet sich ein ERD in Chen-Notation<sup>1)</sup>

## 2. Logisches Schema

Im nächsten Schritt übersetzen wir den Bauplan in eine **technische Skizze**: Das **logische Schema**. Hier sehen wir schon, wie die Datenbank als Zusammenspiel mehrerer Tabellen aussieht.

Beispiel:

- **Film** (FilmID **PK<sup>2)</sup>**, Titel VARCHAR(50), Jahr YEAR, RegisseurID **FK<sup>3)</sup>**)
- **Regisseur** (RegisseurID **PK**, Name VARCHAR(50), Geburtstag DATE)
- Beziehung: **1 Regisseur ↔ n Filme**



Charakteristik:

- Fokus: **Technische Umsetzung**
- Entitäten werden zu **Tabellen**
- Attribute werden zu **Spalten mit Datentypen**
- **Primärschlüssel (PK)<sup>4)</sup>** und **Fremdschlüssel (FK)<sup>5)</sup>** klar markiert
- Kardinalitäten (1:1, 1:n, n:m) sichtbar
- Direkte Vorbereitung für die Umsetzung der Datenbank mit SQL
- Hierfür eignet sich ein ERD in Crow's-Foot-Notation → Erklärung auf der nächsten Seite

### 3. Datentypen



So wie

man beim Hausbau das passende Baumaterial (Holz, Metall, Ziegel) auswählt, wählt man in einer Datenbank die passenden Datentypen für jede Spalte.

Bei einem Haus legen wir fest, aus welchem **Material** die einzelnen Teile bestehen sollen (z. B. Beton für tragende Wände, Glas für Fenster, Holz für Türen). Ähnlich ist es bei den **Spalten einer Tabelle** (bzw. den Attributen einer Entität): Damit eine Datenbank stabil, effizient und sinnvoll funktioniert, müssen wir den passenden **Datentyp** für jede Spalte wählen.

- **INT** = ganze Zahlen (z. B. 1, 2, 3, ... 99554, 99555). → Eignet sich für IDs oder Stückzahlen, die **immer ganze Werte** haben.
- **FLOAT** oder **DOUBLE** = Kommazahlen mit **ungefährer Genauigkeit**. → Gut für Messwerte, die eine gewisse Toleranz erlauben (z. B. Temperaturen, Sensorwerte).
- **DECIMAL(n,m)** = Kommazahlen mit **exakter Genauigkeit**. → Ideal für Geldbeträge, Preise oder Rechnungen (z. B. *DECIMAL(10,2)* für Werte bis 99999999.99).
- **VARCHAR(n)** = Texte mit variabler Länge (z. B. Namen, Beschreibungen). → *VARCHAR(50)* für kurze Titel, *VARCHAR(255)* für längere Beschreibungen.
- **DATE** = speichert nur ein Datum (YYYY-MM-DD).
- **DATETIME** = speichert Datum **und Uhrzeit** (YYYY-MM-DD hh:mm:ss). → Gut für Bestelldatum, Lieferzeitpunkt etc.

Die Wahl des „Materials“ (der richtigen Datentypen) ist entscheidend:

- Mit dem richtigen Datentyp bleibt die Datenbank **robust** (keine widersprüchlichen Werte),
- **effizient** (keine unnötige Speicherplatzverschwendung)
- und **praxisnah** (z. B. exakte Preise statt ungenauer Rundungen).

1)

Die **Chen-Notation** ist eine Darstellungsform für Entity-Relationship-Modelle (ERM). Entitäten werden als Rechtecke, Attribute als Ovale und Beziehungen als Rauten dargestellt. Schlüsselattribute sind unterstrichen.

2) 4)

Primary key - der Primärschlüssel identifiziert eine Zeile (einen Datensatz) eindeutig in einer Tabelle

3) 5)

Foreign key - der Fremdschlüssel verweist auf einen Primärschlüssel in einer anderen Tabelle und stellt damit eine Verknüpfung zu dieser Tabelle her

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

[https://wiki.bzz.ch/modul/m290\\_guko/learningunits/lu04/theorie/a\\_konzeptionelles\\_logisches\\_schema](https://wiki.bzz.ch/modul/m290_guko/learningunits/lu04/theorie/a_konzeptionelles_logisches_schema)

Last update: 2025/09/08 07:19

