

# LU03c – ERM und ERD

## Einleitung

Bevor wir eine Datenbank mit Tabellen in MySQL umsetzen, machen wir uns Gedanken über die **Struktur** der Daten. Dazu benutzen wir ein **konzeptionelles Modell**, das sogenannte **Entity-Relationship Model (ERM)**. Das Ergebnis wird dann in einem **Entity-Relationship Diagramm (ERD)** dargestellt.

→ **Merksatz:** Das ERM/ERD ist der **Bauplan einer Datenbank**.

## Entity-Relationship Model (ERM)

Ein **ERM** beschreibt die Daten und ihre Beziehungen in der realen Welt.

- Eine **Entität** (engl. entity) ist ein Objekt oder ein Begriff, über den wir Informationen speichern möchten. → Beispiele: *Film, Schauspieler, Kunde, Bestellung*.
- Ein **Attribut** ist eine Eigenschaft einer Entität. → Beispiel: Ein Film hat einen *Titel*, ein *Erscheinungsjahr* und ein *Genre*.
- Eine **Beziehung** (engl. relationship) beschreibt, wie Entitäten zueinander in Verbindung stehen. → Beispiel: Eine Bestellung wird von einem Kunden ausgelöst.

Das ERM dient also dazu, die **Realität in Datenbank-Form zu modellieren**, bevor wir Tabellen anlegen.

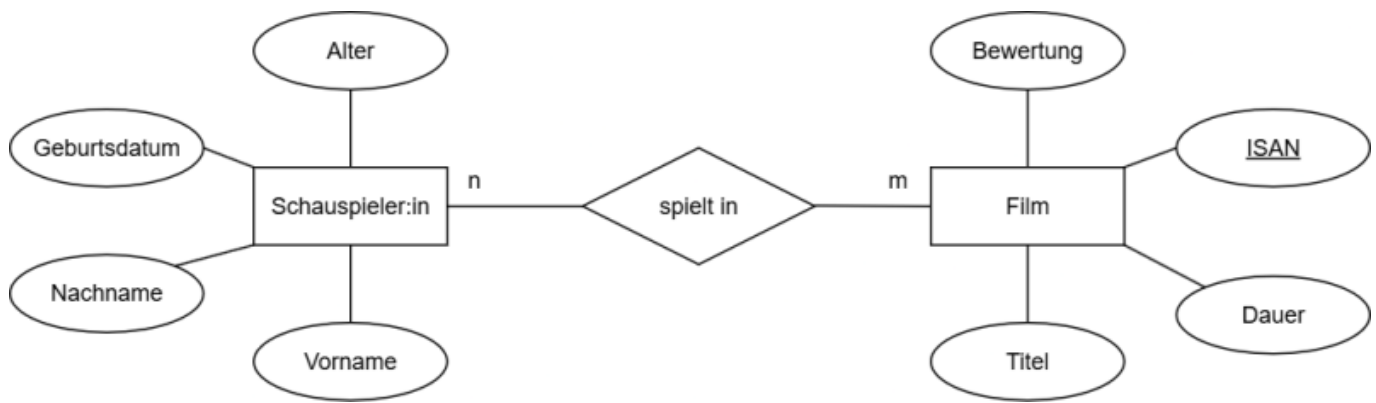
[Erklärungsvideo ERM - 6:35min](#) <sup>1)</sup>

## Entity-Relationship Diagram (ERD)

Ein **ERD** ist die **grafische Darstellung** des ERM. Es zeigt **Entitäten als Rechtecke**, **Attribute als Ovale** **Felder** um die Entität, und **Beziehungen als Linien bzw. Rauten** zwischen den Entitäten.

Beispiel (Filmdatenbank):

- Entität **Film** (Titel, Jahr, Regisseur, Genre)
- Entität **Schauspieler** (Name, Geburtsjahr)
- Beziehung **spielt mit in**: Ein Schauspieler/eine Schauspielerin spielt mit in mehreren Filmen und in einem Film spielen viele Schauspieler:innen (n:m Beziehung)



Das Attribut **ISAN** wird im ER-Diagramm unterstrichen dargestellt, da es ein guter Kandidat für den Primärschlüssel wäre. Die **ISAN** (International Standard Audiovisual Number) ist für Filme vergleichbar mit der **ISBN** bei Büchern – sie identifiziert einen Film eindeutig.

- Rechtecke = Entitäten
- Ovale = Attribute
- Rauten = Beziehungen zwischen den Entitäten

## Entität und Attribut unterscheiden

Viele Lernende stolpern anfangs darüber, **was eine Entität, ein Attribut oder eine Beziehung ist**.

Folgendes Vorgehen, kann Ihnen dabei helfen zu bestimmen, was was ist:

### 1. Starten Sie mit der Geschichte bzw. dem Business Case

Fragen Sie sich: **Welche „Dinge“ sollen in der Datenbank gespeichert werden?**

→ Wenn es ein **„Ding“** ist, das für sich allein existieren kann und von dem es viele geben kann → **Entität**

- Beispiele:
  - Film
  - Kunde
  - Produkt
  - Bestellung
  - Schauspieler:in

### 2. Schauen Sie IN dieses Ding

Fragen Sie sich: **Welche Eigenschaften wollen wir über dieses Ding speichern?**

→ Wenn es eine **Eigenschaft** ist, die das Ding beschreibt, aber nicht für sich alleine stehen kann → **Attribut**

- Beispiele:
  - Filmtitel, Erscheinungsjahr
  - Produktpreis, Lagerbestand

## Entity

vs.

## Attribute

eigenständiges “Ding”, Objekt das gespeichert wird:



ich kann daraus eine Liste erstellen:

### Kundenliste

- Kundin Hanna Musterfrau
- Kunde Martin Mustermann

### Bestellungen

- Bestellung von Kaffee kapseln am 19.08.25
- Bestellung von Fairphone am 25.08.25

beschreibt eine Eigenschaft einer Entity



hat immer einen Bezug zu einem Objekt



### 3. Schauen Sie ZWISCHEN die Dinge

Fragen Sie sich: **Wie sind diese Dinge miteinander verbunden oder interagieren sie?**

Wenn es um die Verknüpfung zwischen zwei Entitäten geht → **Beziehung**

- Beispiele:
  - Film ↔ Regisseur → „dreht“
  - Kunde ↔ Bestellung → „gibt auf“
  - Bestellung ↔ Produkt → „enthält“

### 4. Der 3-Fragen-Schnelltest

- **Entität?** Können Sie sich vorstellen, eine Liste davon zu speichern (Film 1, Film 2, Film 3 ...)?
- **Attribut?** Beschreibt es nur eine einzelne Eigenschaft einer Entität (Titel eines Films)?
- **Beziehung?** Verknüpft es zwei Entitäten (Kunde gibt Bestellung auf)?

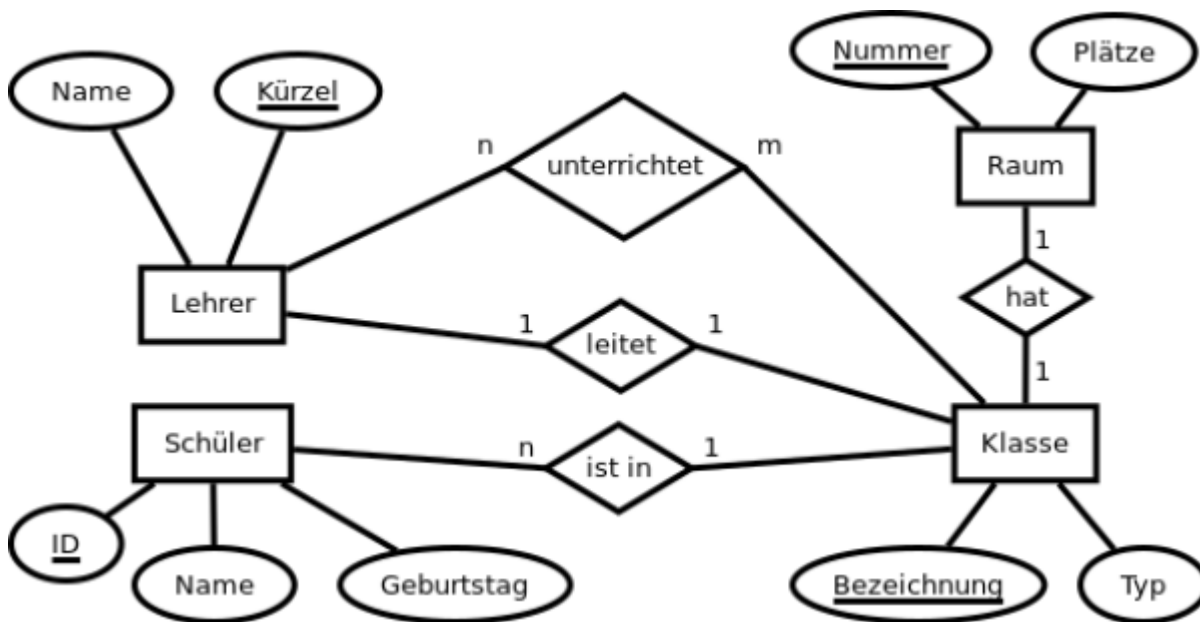
### 5. Übungsbeispiel

**Geschichte:** \*Ein Kunde bestellt Produkte in einem Webshop.\*

- **Entitäten:** Kunde, Produkt, Bestellung

- **Attribute:** Kundenname, E-Mail; Produkttitel, Preis; Bestelldatum, Gesamtbetrag
- **Beziehung:**
  - Kunde ↔ Bestellung → „gibt auf“
  - Bestellung ↔ Produkt → „enthält“

Im folgenden ER-Diagramm wird dargestellt, wie am BZZ die Entitäten **Lehrer**, **Schüler/Lernende**, **Klasse** und **Räume** miteinander in Beziehung stehen.



## Kardinalitäten

Die **Kardinalität** beschreibt die Art der Beziehung zwischen zwei Entitäten (z. B. Lehrer und Klasse). Sie zeigt, **wie viele Objekte einer Entität mit wie vielen Objekten einer anderen Entität verbunden sind**.

Es gibt drei Hauptarten von Kardinalitäten:

**1. One-to-One (1:1)** Eine Instanz <sup>2)</sup> von Entität A gehört genau zu einer Instanz von Entität B.

Beispiel:

- **Ein:e** Mitarbeitende hat genau **ein** Arbeitsplatz
- Jeder Reisepass gehört genau einer Person

**2. One-to-Many (1:N)** Eine Instanz <sup>3)</sup> von Entität A kann mit vielen Instanzen von Entität B verbunden sein. Umgekehrt gehört jede Instanz von B zu genau einer Instanz von A.

Beispiel:

- **Ein** Regisseur kann **viele** Filme drehen, aber **ein Film** hat in der Regel nur **ein Regisseur, eine Regisseurin**.
  - **Ein** Spotify-Artist produziert **viele** Songs. **Ein Song** ist (meistens) nur von **einem Artist** produziert.
  - **Eine** Klasse hat **viele** Lernende. **Eine Lernende, ein Lernender** ist Teil (in der Regel) **einer Klasse**.
- 

**3. Many-to-Many (M:N)** Mehrere Instanzen von Entität A können mit mehreren Instanzen von Entität B verknüpft sein.

Beispiel:

- **Mehrere** Schauspieler:innen ↔ spielen in **mehreren** Filmen (ein Schauspieler/eine Schauspielerin kann in mehreren Filmen mitspielen und ein Film kann mehrere Schauspieler:innen haben)
  - **Viele** Lernende ↔ besuchen **mehrere** Module
  - **Viele** Songs ↔ sind in **mehreren** Playlists enthalten
- 

## Tools: ERM/ERD zeichnen

Zum Zeichnen benutzen wir **Draw.io**, ein kostenloses Online-Tool.

- [Draw.io öffnen](#)
- Links in der Leiste gibt es die Kategorie **Entity Relation** → Symbole für Tabellen, Attribute, Beziehungen.
- Modelle können gespeichert oder exportiert werden (Datei → Exportieren → PNG).

## Warum lernen wir das?

Wenn wir **Tabellen direkt in MySQL** erstellen würden, ohne vorher ein ERD, hätten wir schnell Probleme:

- doppelte oder widersprüchliche Daten,
- fehlende Schlüssel → Daten lassen sich nicht eindeutig verknüpfen,
- unklare Strukturen → niemand versteht die Datenbank ausser dem Ersteller.

Ein **ERD** sorgt dafür, dass:

- die Struktur **klar und nachvollziehbar** ist,
- wir **richtige Beziehungen** zwischen Tabellen setzen,
- wir später in SQL effizient arbeiten können.

## Zusatzmaterial: ERM & ERD

Folgende Videos helfen Ihnen, das Thema noch besser zu verstehen:

- [Entity Relationship Diagrams<sup>4\)</sup>](#) → (20:23, en) Ein leicht verständliches Tutorial zu ERDs – mit Beispielen zu Chen-Notation, Crow's Foot, Kardinalitäten und Weak Entities.
- [ENTITY RELATIONSHIP MODELL einfach erklärt \(ER-Modell\)<sup>5\)</sup>](#) → (5:03, de) Einführung ins ER-Modell: Grundelemente, Beziehungen (1:1, 1:n, m:n) und deren Bedeutung.
- [ER-Modell - Teil 1 \(ABITUR 2018\)<sup>6\)</sup>](#) → (4:53, de) Aufbau von Entities, Attributen und Beziehungen. Mit Beispiel „Buch – Autor“.
- [Kardinalitäten und Primärschlüssel | ERM bei Datenbanken<sup>7\)</sup>](#) → (13:05, de) Anschauliche Erklärung von Primärschlüsseln und Kardinalitäten mit Beispielen.



<sup>1)</sup>

Quelle: Timo Bäuerle/YouTube

<sup>2)</sup> <sup>3)</sup>

Eine Instanz ist ein konkretes Beispiel oder eine einzelne Ausprägung einer Entität → Entität „Film“ → Instanz: „The Godfather“

<sup>4)</sup>

Decomplexify / YouTube

<sup>5)</sup>

IT & Medien einfach erklärt / YouTube

<sup>6)</sup>

Informatik - simpleclub / YouTube

<sup>7)</sup>

Sebastian Philippi / YouTube

From:

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

Permanent link:

[https://wiki.bzz.ch/de/modul/m290\\_guko/learningunits/lu04/theorie/c\\_erm\\_erd](https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu04/theorie/c_erm_erd)

Last update: **2026/08/31 11:40**

