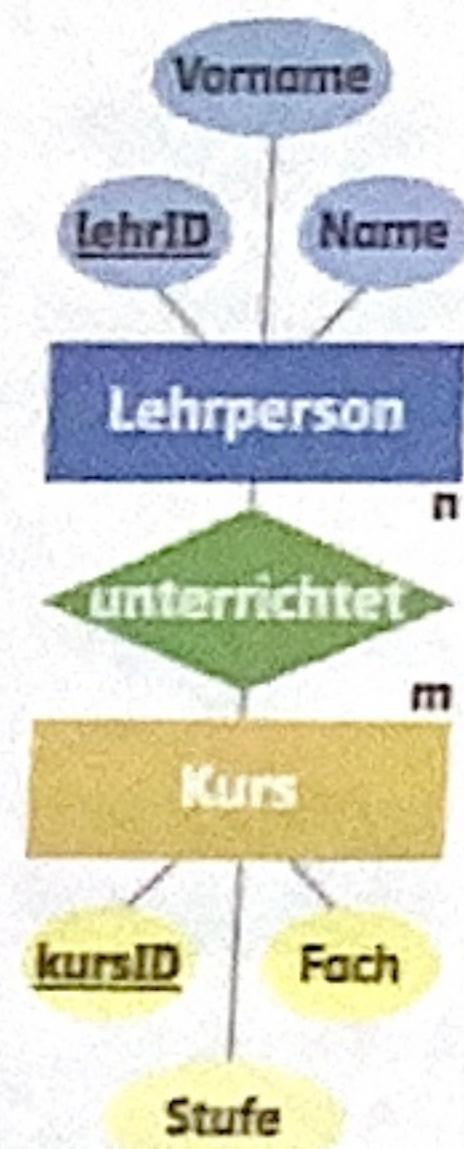


# Eine Datenbank anlegen – gute Relationen

## Relationenschemata

Beim Modellieren von Entitäten und Attributen hast du vielleicht überlegt, alle Entitäten eines Typs in Tabellenzeilen zu schreiben. Spaltenüberschriften wären die Bezeichner der Attribute, und die Tabelle hieße genau wie der Entitätstyp. Notiert man nur den Tabellennamen zusammen mit den Spaltenüberschriften, nennt man dies ein **Relationenschema**. Alle Relationenschemata einer DB nennt man **Datenbankschema** (→ Abb. 4).



1 ER-Modell

Datenbank wird häufig mit DB abgekürzt.

Lehrperson			Kurs		
lehrID	Name	Vorname	kursID	Stufe	Fach
1	Muster	Max	1	10	M
2	Trimm	Shina	2	11	IF
...	...	...	...	...	...

2 Relationen zu Entitätstypen

## Entitätstypen entsprechen Tabellen

Um ein ERM als Datenbank zu implementieren, kann man zunächst für jeden Entitätstyp eine eigene Tabelle mit je einer Spalte pro Attribut planen. Findest du für ein Attribut keinen geeigneten Datentyp, z.B. weil dein Attribut eine Liste von Werten enthalten soll, ist es gut, das ERM noch mal zu überarbeiten und z.B. aus listenwertigen Attributen eigene Entitätstypen zu machen.

unterrichtet	
lehrID	kursID
1	1
1	3
2	2
...	...

3 Beziehungstabelle

## Beziehungstypen umsetzen

Beziehungen lassen sich am leichtesten als eigene Tabellen verstehen. Jeder Datensatz einer Beziehungstabelle steht für eine konkrete Beziehung: Die Tabelle „unterrichtet“ (→ Abb. 3) enthält die Kombinationen der Form (lehrID, kursID), die anzeigen, welche Lehrperson welchen Kurs unterrichtet. Alle weiteren Daten dieser Entitäten finden sich

redundare (lat.: im Überfluss vorhanden sein)  
Daten sind redundant, wenn die gleiche Information auch gewonnen werden könnte, wenn man Daten weglassen würde.

in den Tabellen „Lehrperson“ und „Kurs“. Daher genügt es, in Beziehungstabellen die **Primärschlüssel** der in Beziehung stehenden Entitäten („Fremdschlüssel“) zu speichern: Welcher Kurs z.B. zur kursID 1 des Tupels (1,1) in der ersten Zeile gehört, findet man in der Tabelle „Kurs“, indem man dort die kursID 1 sucht. Hat eine Beziehung Attribute, tauchen diese als Spalten in der Beziehungstabelle auf. Der Primärschlüssel der Beziehungstabelle ist aus den Fremdschlüsseln zusammengesetzt. In manchen Fällen ist es sinnvoll, auch Attribute der Beziehung zum Primärschlüssel hinzuzufügen.

**Nicht für jeden Beziehungstyp eine Tabelle**  
Nicht für jeden Beziehungstyp benötigt man eine eigene Tabelle: Zum Beispiel genügt es, bei einer 1:n-Beziehung den Primärschlüssel der linken Seite (der „1-Seite“) als Fremdschlüssel in die rechte Seite (die „n-Seite“) aufzunehmen. Falls die Beziehung Attribute hat, werden auch diese in die rechte Seite übernommen.

Lehrperson (lehrID, name, vorname)  
unterrichtet (lehrID, kursID)  
Kurs (kursID, Stufe, Fach)

4 Datenbankschema (Fremdschlüssel durch Pfeile gekennzeichnet)

## Redundanz und Konsistenz

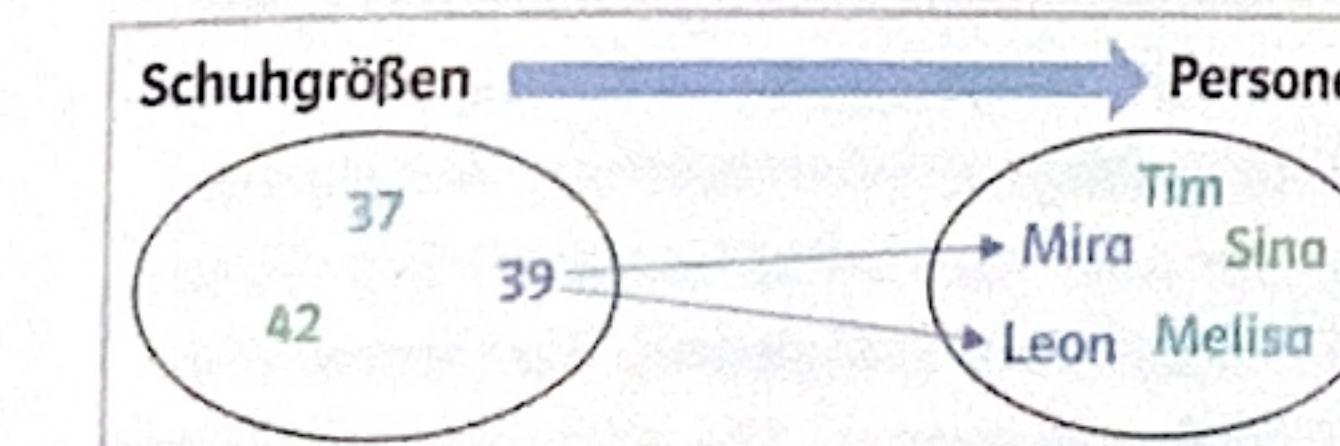
Wenn dein ERM gut war und du wie beschrieben vorgehst, erhältst du ein gutes Datenbankschema. Doch was sind eigentlich Qualitätskriterien für Datenbanken? Datenbankentwickler versuchen, **Redundanz** zu vermeiden, um Speicherplatz zu sparen, und damit die Datenbank keine **widersprüchlichen (inkonsistenten)** Daten enthält: Speicherst du neben dem Geburtsdatum einer Person auch ihr Alter, könnte das aus dem Geburtsdatum und dem heutigen Datum berechnete Alter von dem gespeicherten Alter abweichen.

## Normalformen

Durch **Normalformen** kann man die Gefahr von Redundanz und Dateninkonsistenz verringern. In nicht normalisierten Datenbanken können **Einfüge-, Änderungs- und Lösch-Anomalien** auftreten, z.B. wenn redundante Daten an mehreren Stellen gespeichert sind, doch nur an einer Stelle geändert werden, oder wenn beim Löschen mehr Daten als gewünscht verloren gehen.

### 1. Normalform (1 NF)

Eine Tabelle befindet sich in 1 NF, wenn sie eine feste Breite hat, keine **Wiederholungsgruppen** und keine zusammengesetzten oder listenwertigen Attribute enthält. Ein Beispiel wäre eine Tabelle, in der Musikalben mit den Titeln gespeichert werden: Lösungen, die der 1. Normalform widersprechen, sind z.B. **eine Spalte pro Titel (Wiederholungsgruppe)** oder **alle Titel eines Albums als ein Attributwert (Liste)**. Können Attributwerte nicht mehr sinnvoll aufgespalten werden, bezeichnet man diese als **atomar**.



5 Keine eindeutige Zuordnung

## Merke

- Entitätstypen und n:m-Beziehungstypen werden in eigenen Tabellen umgesetzt.
- Normalisierung hat das Ziel, die Gefahr von Datenkonsistenz zu vermeiden.
- In der ersten Normalform müssen alle Attribute atomare Wertebereiche haben.

## Aufgaben

- Erläutere, wie das relationale Datenbankschema in Abb. 4 aus dem ER-Modell (→ Abb. 1) entsteht.
  - a) Begründe, dass im Anwendungskontext die Beziehung *unterrichtet* (→ Abb. 1) in der Regel eine 1:n-Beziehung ist.
  - Erkläre, wo das Schema *lehrperson*(lehrID, name, vorname,
- b) Modifiziere das Relationenschema „Kurs“ so, dass die 1:n-Beziehung dort durch einen Fremdschlüssel realisiert wird.
- c) Schreibe die Tabelle „Kurs“ entsprechend auf.
- kurIDs) die 1. Normalform verletzt und wodurch diese Verletzung entsteht.
- 4 Recherchiert Beispiele für 2NF und 3NF und stellt sie im Kurs vor.

der Schlüsselkandidat möglicher (kombinierter) Primärschlüssel