

# Darstellungsweisen

## ER-Diagramme

Eine Möglichkeit, wie man eine Datenbank strukturiert darstellen kann, hast du bereits kennengelernt - als ER-Diagramm.



### (A1)

Eine Universität bietet Veranstaltungen an, die an einem bestimmten Wochentag in einem bestimmten Zeitblock in einem bestimmten Raum stattfinden. Jede Veranstaltung wird von genau einer Dozentin angeboten, von dem Nachname, Vorname und eine eindeutige AngestelltenId bekannt sind.

Ein Inhaltsmodul umfasst alle Veranstaltungen, die in diesem Modul angeboten werden. Für jedes Inhaltsmodul gibt es eine eindeutige Modulnummer, eine Bezeichnung und die Zahl der CreditPoints, die die Studierenden angerechnet werden.

Studierende besuchen Veranstaltungen. Studierende haben einen Namen und Vornamen und eine eindeutige Matrikelnummer.

- Erstelle ein Entity-Relationship-Diagramm, das die beschriebene Struktur darstellt. Welche Entscheidung musst du bezüglich der Informationen *Zeitblock*, *Raum* und *Wochentag* treffen? Mache das ER-Diagramm möglichst einfach.
- Kennzeichne die Kardinalitäten und die Primärschlüssel.
- Erkennst du ein Problem? Wie könntest du es lösen?

### Lösungsvorschlag



#### Anmerkungen:

- Attribute, die einen Primärschlüssel darstellen werden im ER Diagramm unterstrichen, Fremdschlüssel werden in ER Diagrammen als Attribut eingetragen und auch nicht gekennzeichnet.
- Man hätte Wochentag, Block und Raum auch als weitere Entitäten auffassen können.
- Die Veranstaltungen haben keinen offensichtlichen Primärschlüssel. Hier könnte ein Surrogatschlüssel VeranstaltungsID Abhilfe schaffen.

Wir betrachten ein weiteres Beispiel:



(A2)

- Beschreibe stichwortartig, welche Situation hier modelliert wird.
- Erläutere, welche Bedeutung die Attribute an der Beziehungsraute besitzt haben.

Da jeder Fahrzeugbesitz mit einer An- oder Abmeldung des KFZ einhergeht, macht es Sinn, die Informationen Kennzeichen, An- und Abmeldedaten der Beziehung „besitzt“ zuzuschreiben. Wenn man das ER Diagramm jetzt in Datenbank-Tabellen überführen möchte, benötigt man also für die Beziehungsraute - wie auch schon für die Attribute - eine eigene Tabelle. Außerdem werden Fremdschlüssel im ER-Diagramm nicht dargestellt, es ist in dieser Hinsicht also unvollständig, obwohl es natürlich einen ausgezeichneten Überblick über die Beziehungen zwischen den Entitäten bietet.

## Relationales Modell

Wenn man das ER-Modell in ein relationales Modell überführen möchte, überlegt man sich, welche Tabellen mit welchen Spalten man erstellen würde, um die Datenbank zu modellieren.



**Grundregel:** Jede Entität und jede Beziehung wird in Form einer eigenen Tabelle (Relationenschema) abgebildet. Eine Beziehungstabelle hat als Attribute die Primärschlüssel der beteiligten Tabellen (als Fremdschlüssel), sowie alle Attribute, die zur Beziehung gehören.

### Schreibweise

Das **relationale Datenbankschema** einer Datenbank (Darstellung als Tabellen) besteht für jede Tabelle aus einer Zeile:

**TabellenName**(Attribut1, Attribut2, Attribut2 ↑, ..., AttributN)

Dabei

- stehen in den Klammer *alle* Attribute, die in der Tabelle dargestellt werden.
- Primärschlüssel werden unterstrichen
- **Fremdschlüssel** sind mit einem **Pfeil nach oben** (↑) markiert<sup>1)</sup>

Das **relationale Datenbankschema** einer Datenbank hat also genau so viele solche Zeilen, wie die

Datenbank Tabellen hat. Diese Darstellung eignet sich hervorragend, um die Datenbank tatsächlich in einem DBMS anzulegen, die Beziehungen zwischen den Tabellen sind jedoch sehr viel schlechter erkennbar, als im ER-Modell.

---



**(A3)**

Überführe das ER-Modell der

KFZ-Zulassungsstelle

oben in ein relationales Datenbankschema.

Lösungshinweis 1

Starte mit der Überlegung, welche Tabellen du benötigst - eine für jede Entität und Beziehung:

Fahrzeughalter(...

Kraftfahrzeug(...

FahrzeugTyp(...

besitzt(...

istEin(...

Lösungsvorschlag

**Fahrzeughalter**(HalterID, Name, Vorname, Geburtsdatum, Strasse, Ort, PLZ)

**Kraftfahrzeug**(FahrgestellNR, Baujahr)

**FahrzeugTyp**(TypCode, Modell, Hersteller, Schadstoffklasse

**besitzt**(HalterID ↑, FahrgestellNR ↑, Anmeldedatum, Abmeldedatum, Kennzeichen)

**istEin**(FahrgestellNR ↑, HalterID ↑)

Wenn man die Lösung genauer betrachtet<sup>2)</sup>, fällt auf, dass man sich die Tabelle für die 1:n Beziehung **istEin** auch sparen könnte: Hier könnte ein Fremdschlüssel FahrgestellNR in der Tabelle *FahrzeugTyp* auch direkt auf den Primärschlüssel FahrgestellNR der Tabelle *Kraftfahrzeug* verweisen.

---



**(A4)**

Kannst du eine Regel formulieren, wann man eine 1:n Beziehung ohne eigene Tabelle im relationalen

Datenbankschema abbilden kann? Kann man die Beziehung besitzt auch ohne eigene Tabelle modellieren?

## Lösung

Wenn die 1:n Beziehung keine eigenen Attribute hat, kann man sie durch eine direkte Beziehung der beiden Tabellen modellieren und benötigt keine eigene „Beziehungstabelle“.

Man benötigt **keine eigene Beziehungstabelle**:



- Für 1:1 Beziehungen - diese können sogar in eine Tabelle zusammengefasst werden.
- Für 1:n Beziehungen, wenn die Beziehung keine eigenen Attribute besitzt. Dann kann durch Primär- und Fremdschlüssel eine direkte Beziehung zwischen den Tabellen hergestellt werden.

**Für n:m Beziehungen benötigt man immer eine eigene Beziehungstabelle**

**n-m-Beziehungen** stellen sind also besonders wichtig, denn diese kann man nicht ohne eine eigene Beziehungstabelle abbilden. Lies also auf der [Wiki Seite zu den n-m-Beziehungen](#) weiter!

1)

manchmal werden Fremdschlüssel auch gestrichelt unterstrichen

2)

Bitet aufklappen!

From:  
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:  
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:darstellungsweise:start?rev=1643825532>

Last update: **02.02.2022 19:12**

