Darstellungsweisen

ER-Diagramme

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

(A1)

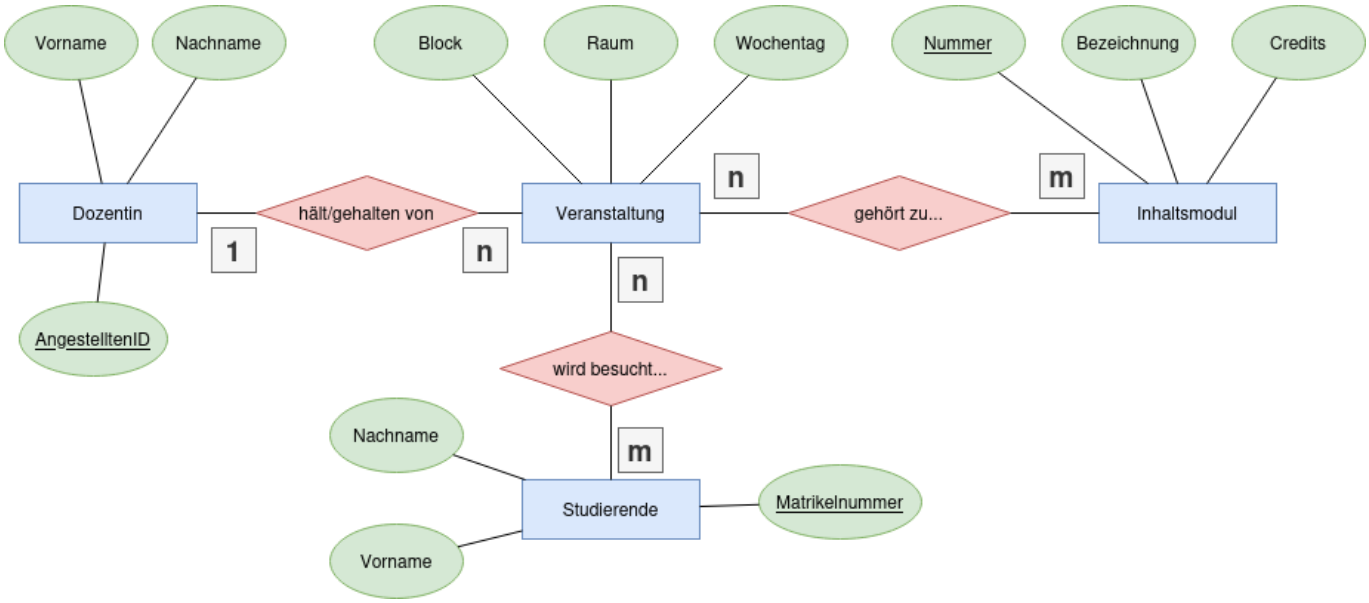
Eine Universität bietet Veranstaltungen an, die an einem bestimmten Wochentag in einem bestimmten Zeitblock in einem bestimmten Raum stattfinden. Jede Veranstaltung wird von genau einer Dozentin angeboten, von dem Nachname, Vorname und eine eindeutige AngestelltenId bekannt sind.

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 den 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?

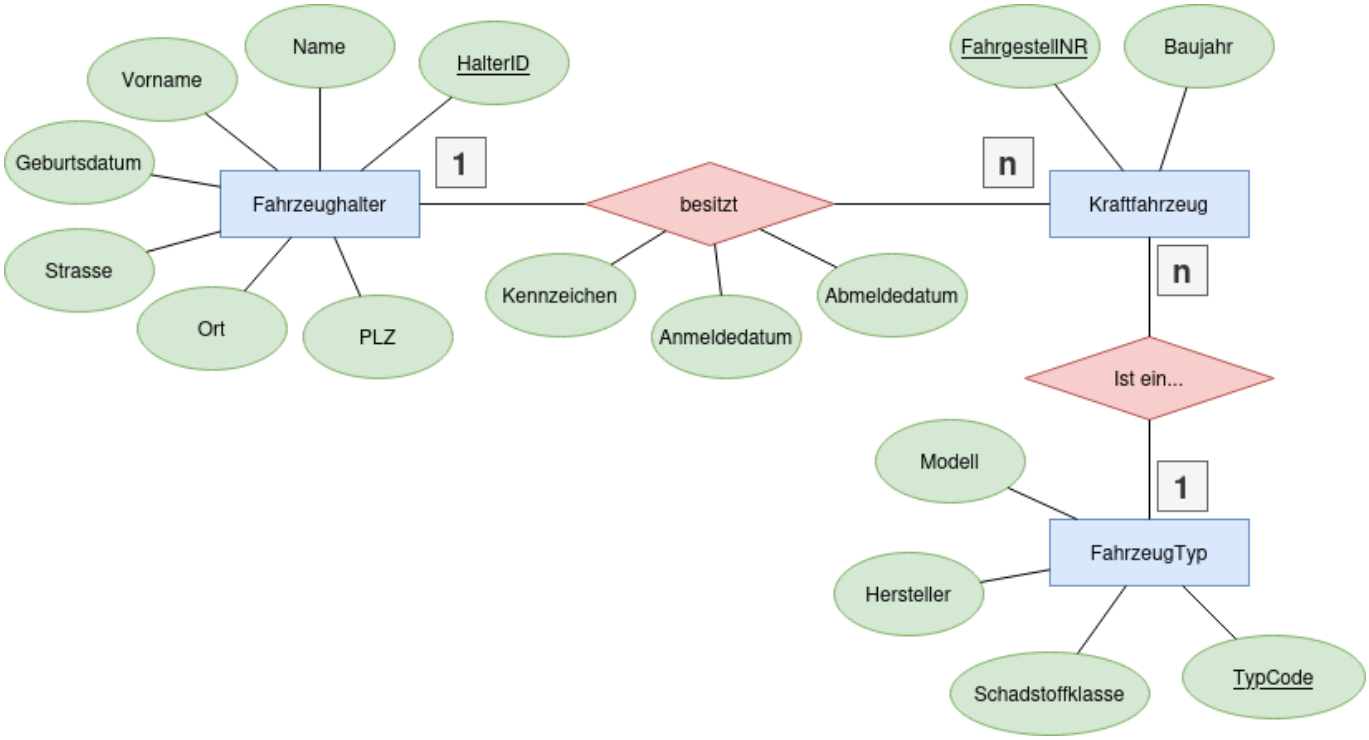
Lösungsvorschlag



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:



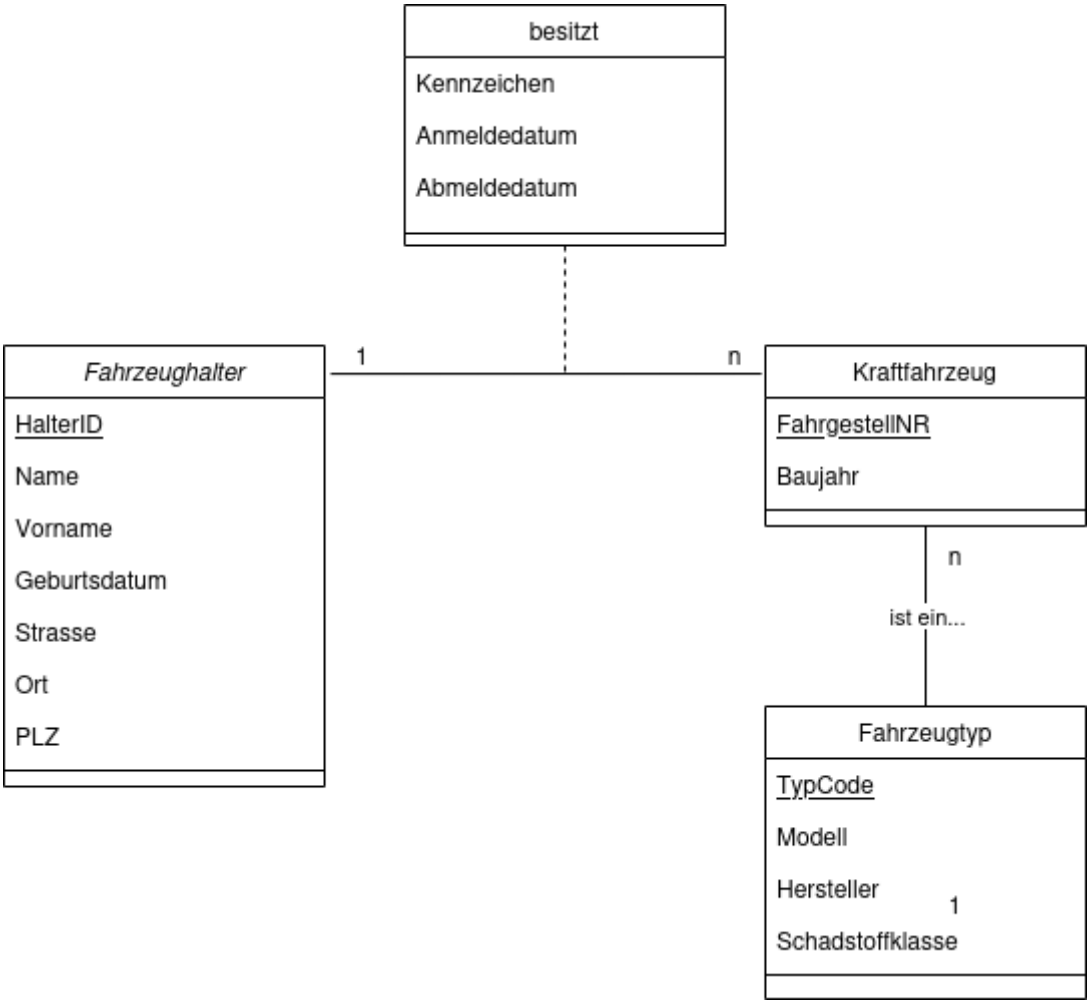
(A2)

- 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.

UML Diagramme

Eine weiter Möglichkeit, Datenbankentwürfe grafisch darzustellen, ist mit Hilfe von UML Diagrammen. Die Darstellung verwendet dabei die aus der objekorientierten Programmierung üblichen Konventionen für Objekte und deren Beziehungen zueinander. Die Beispieldatenbank der KFZ Verwaltung aus dem vorigen Abschnitt in der Darstellung als UML-Diagramm folgendermaßen aussehen:



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[1)]

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-Zulassungstelle

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↑,TypCode↑)

Wenn man die Lösung genauer betrachtet[2)], 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 einen direkte Beziehung der beiden Tabellen modellieren und benötigt keine eigene "Beziehungstabelle".

Man benötigt **keine eigene Beziehungs-Tabelle**:

- 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 das Tabellen hergestellt werden.

Für n:m Beziehungen benötigt man immer eine eigene Beziehungstabelle

→ n-m-Beziehungen sind also ein wichtiger "Spezialfall", denn diese kann man nicht ohne eine eigene Beziehungstabelle abbilden. Lies also auf der Wiki Seite zu den n-m-Beziehungen weiter!

(A5)

Erstelle ein relationales Datenbankschema zum

ER-Modell der Universität

.

(A6)

3)

Eine Fluggesellschaft speichert alle Buchungen der Passagiere. Für das Vielfliegerprogramm "Sky-High" werden diese Daten ausgewertet. Ein Auschnitt aus der verwendeten Datenbank soll modelliert werden, und zwar die Miniwelt mit den folgenden Entitäten und Eigenschaften:

- Ein Flugzeug hat einen bestimmten Typ, z.B. "B747-A", ein Kennzeichen, das es identifiziert, z.B. "D-LH54, ein Baujahr sowie eine Anzahl von Sitzplätzen.
- Flugzeuge legen Flugstrecken zurück. Jede Flugstrecke hat eine Flugnnummer, wie z.B. "LH-3186". Die Flugnummer bezeichnet den Flug der Flugstrecke mitsamt seiner Abflugzeit. Der Flug der Strecke wird somit durch die Flugnummer und der Abflugszeit eindeutig festgelegt.
- Flugstrecken führen von einem Startflughafen zu einem Zielflughafen, haben eine bestimmte Dauer und eine bestimmte Länge in Meilen.
- Passagiere haben einen Namen und Vornamen. Sie buchen Flugstrecken.
- Vielflieger sind Passagiere, die häufig fliegen und an "Sky-High" teilnehmen. Sie haben ein Meilenkonto mit einer eindeutigen Kontonummer. Über jeden Vielflieger ist bekannt, wieviele Sky-High-Meilen er bisher angesammelt hat.

Aufgaben

- Erstelle zunächst ein ER-Diagramm der Miniwelt
- Überführe das ER-Diagramm in ein relationales Datenbankschema

- Überlege dir, ob der "Umweg" über das ER Diagramm Vorteile bei der Modellierung bietet gegenüber dem Versuch, das relationale DB-Schema direkt niederzuschreiben? (Welche)

1)

manchmal werden Fremdschlüssel auch gestrichelt unterstrichen

2)

Bitte aufklappen!

3)

Image by Lars_Nissen

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:datenbanken:darstellungsweise:start?rev=1706000557

Last update: **23.01.2024 09:02**