

12 Datalog: Rechnen mit Relationen

Beispiel 1: Das Lehrangebot einer TU



⇒ Entity-Relationship Diagram

Diskussion:

- Viele Anwendungsbereiche lassen sich mit Hilfe von **Entity-Relationship**-Diagrammen beschreiben.
- Entitäten im Beispiel: **Dozent**, **Vorlesung**, **Student**.
- Die Menge aller **vorkommenden** Entitäten d.h. Instanzen lassen sich mit einer Tabelle beschreiben ...

Dozent :

Name	Telefon	Email
Brauer	17204	brauer@in.tum.de
Nipkow	17302	nipkow@in.tum.de
Seidl	18155	seidl@in.tum.de

Vorlesung:

Titel	Raum	Zeit
Diskrete Strukturen II	MI 1	Do 12:15-13, Fr 10-11:45
Perlen der Informatik II	MI 3	Do 8:30-10
Einführung in die Informatik II	MW 1802	Di 12-14, Fr 11:45-13:15
Compilerbau	MI 2	Mo 12-14, Mi 10-12

Student:

Matr.nr.	Name	Sem.
123456	Hans Dampf	02
007042	Fritz Schluri	10
543345	Anna Blume	02
131175	Effi Briest	04

Diskussion (Forts.):

- Die Zeilen entsprechen den Instanzen.
- Die Spalten entsprechen den **Attributen**.
- **Annahme:** das erste Attribut **identifiziert** die Instanz
 \implies **Primärschlüssel**

Folgerung: Beziehungen sind ebenfalls Tabellen ...

liest:

Name	Titel
Brauer	Diskrete Strukturen II
Nipkow	Perlen der Informatik II
Seidl	Einführung in die Informatik II
Seidl	Compilerbau

hört:

Matr.nr.	Titel
123456	Einführung in die Informatik II
123456	Compilerbau
123456	Diskrete Strukturen II
543345	Einführung in die Informatik II
543345	Diskrete Strukturen II
131175	Compilerbau

Mögliche Anfragen:

- In welchen Semestern sind die Studierenden der Vorlesung “Diskrete Strukturen II” ?
- Wer hört eine Vorlesung bei Dozent “Seidl” ?
- Wer hört sowohl “Diskrete Strukturen II” wie “Einführung in die Informatik II” ?

⇒ Datalog

Idee: **Tabelle** \iff **Relation**

Eine **Relation** R ist eine Menge von **Tupeln**, d.h.

$$R \subseteq \mathcal{U}_1 \times \dots \times \mathcal{U}_n$$

wobei \mathcal{U}_i die Menge aller möglicher Werte für die i -te Komponente ist. In unserem Beispiel kommen etwa vor:

`int`, `string`, möglicherweise Aufzählodatentypen

// Einstellige Relationen sind **Mengen** :-)

Relationen können durch **Prädikate** beschrieben werden ...

Prädikate können wir definieren durch Aufzählung von **Fakten ...**

... im Beispiel:

liest ("Brauer", "Diskrete Strukturen II").

liest ("Nipkow", "Perlen der Informatik II").

liest ("Seidl", "Einführung in die Informatik II").

liest ("Seidl", "Compilerbau").

hört (123456, "Compilerbau").

hört (123456, "Einführung in die Informatik II").

hört (123456, "Diskrete Strukturen").

hört (543345, "Einführung in die Informatik II").

hört (543345, "Diskrete Strukturen II").

hört (131175, "Compilerbau").

Wir können aber auch **Regeln** benutzen, mit denen weitere Fakten abgeleitet werden können ...

... im Beispiel:

```
hat_Hörer (X,Y) :- liest (X,Z), hört (M,Z), student (M,Y,_).  
semester (X,Y) :- hört (Z,X), student (Z,_,Y).
```

- `:-` bezeichnet die logische **Implikation** " \Leftarrow ".
- Die komma-separierte Liste sammelt die Voraussetzungen.
- Die linke Seite, der **Kopf** der Regel, ist die Schlussfolgerung.
- Die Variablen werden groß geschrieben.
- Die **anonyme Variable** `_` bezeichnet irrelevante Werte **:-)**

An die Wissensbasis aus Fakten und Regeln können wir jetzt Anfragen stellen ...

... im Beispiel:

?- hat_Hörer ("Seidl", Z).

- Datalog findet alle Werte für Z, für die die Anfrage aus den gegebenen Fakten mit Hilfe der Regeln beweisbar ist :-)
- In unserem Beispiel ist das:

Z = "Hans Dampf"

Z = "Anna Blume"

Z = "Effi Briest"

Weitere Anfragen:

?- semester ("Diskrete Strukturen II", X).

X = 2

X = 4

?- hört (X, "Einführung in die Informatik II"),

hört (X, "Diskrete Strukturen II").

X = 123456

X = 543345

Weitere Anfragen:

?- semester ("Diskrete Strukturen II", X).

X = 2

X = 4

?- hört (X, "Einführung in die Informatik II"),

hört (X, "Diskrete Strukturen II").

X = 123456

X = 543345

Achtung:

Natürlich kann die Anfrage auch gar keine oder mehr als eine Variable enthalten :-)

Ein Beispiel-Beweis:

Die Regel:

`hat_Hörer (X,Y) :- liest (X,Z), hört (M,Z), student (M,Y,_).`

gilt für alle `X, M, Y, Z`.

Ein Beispiel-Beweis:

Die Regel:

`hat_Hörer (X,Y) :- liest (X,Z), hört (M,Z), student (M,Y,_).`

gilt für alle `X, M, Y, Z`. Mit Hilfe der Substitution:

`"Seidl"/X "Einführung ..."/Z 543345/M "Anna Blume"/Y`

können wir schließen:

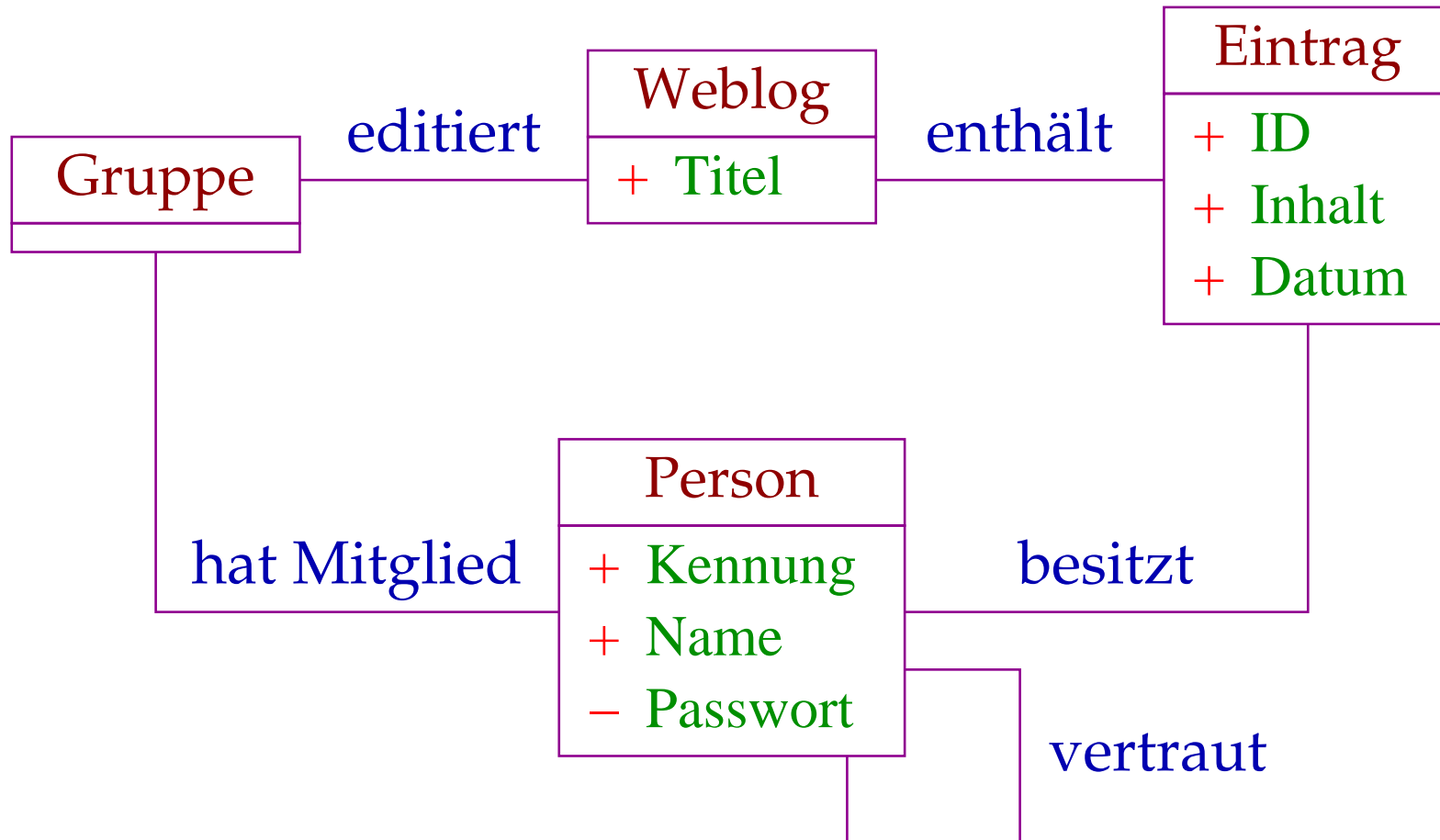
`liest ("Seidl", "Einführung ...")`

`hört (543345, "Einführung")`

`student (543345, "Anna Blume", 2)`

`hat_Hörer ("Seidl", "Anna Blume")`

Beispiel 2: Ein Weblog



Aufgabe: Festlegung der Zugriffsberechtigung

- Jedes Mitglied der editierenden Gruppe darf einen neuen Eintrag hinzufügen.
- Nur die Besitzerin eines Eintrags darf ihn löschen.
- Modifizieren darf ihn jeder, dem die Besitzerin traut.
- Lesen darf ihn jedes Mitglied der Gruppe und jeder ihrer mittelbar Vertrauten ...

Spezifikation in Datalog:

```
darf_hinzufügen (X,W) :- editiert (Z,W),  
                           hat_Mitglied (Z,X).  
darf_löschen (X,E) :- besitzt (X,E).  
darf_modifizieren (X,E) :- besitzt (X,E).  
darf_modifizieren (X,E) :- besitzt (Y,E),  
                           vertraut (Y,X).  
darf_lesen (X,E) :- enthält (W,E),  
                   darf_hinzufügen (X,W).  
darf_lesen (X,E) :- darf_lesen (Y,E),  
                   vertraut (Y,X).
```

Beachte:

- Zur Definition neuer Prädikate dürfen wir selbstverständlich alle vorhandenen benutzen oder sogar Hilfsprädikate definieren.
- Offenbar können Prädikatsdefinitionen auch **rekursiv** sein :-)
- Mit einer Person X , die einen Eintrag besitzt, dürfen auch alle Personen modifizieren, denen X traut.
- Mit einer Person Y , die einen Eintrag lesen darf, dürfen auch alle Personen lesen, denen Y traut :-))

12.1 Beantwortung von Anfragen

Gegeben: eine Menge von Fakten und Regeln.

Gesucht: die Menge aller ableitbaren Fakten.

Problem:

`equals (X,X) .`

\implies Die Menge aller ableitbaren Fakten ist nicht endlich :-)

Satz:

Sei W eine endliche Menge von Fakten und Regeln mit den folgenden Eigenschaften:

- (1) Fakten enthalten keine Variablen.
- (2) Jede Variable im Kopf kommt auch im Rumpf vor.

Dann ist die Menge der ableitbaren Fakten **endlich**.

Satz:

Sei W eine endliche Menge von Fakten und Regeln mit den folgenden Eigenschaften:

- (1) Fakten enthalten keine Variablen.
- (2) Jede Variable im Kopf kommt auch im Rumpf vor.

Dann ist die Menge der ableitbaren Fakten **endlich**.

Beweisskizze:

Man zeigt für jedes beweisbare Faktum $p(a_1, \dots, a_k)$, dass jede Konstante a_i bereits in W vorkommt (:-))