

Übung zur Vorlesung *Grundlagen: Datenbanken* im WS08/09

Martina Albutiu (albutiu@in.tum.de)

<http://www-db.in.tum.de/teaching/ws0809/dbsys/exercises/>

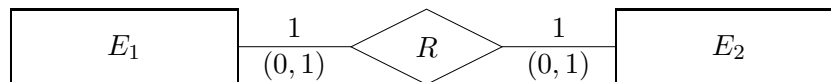
Blatt Nr. 14

Dieses Übungsblatt wird in der 6. Kalenderwoche in den Übungen besprochen.

Hinweis:

Am 30.01.2009, 16-18 Uhr findet eine Zentralübung im HS 1 statt. Hierbei werden Fragen zum Vorlesungs- und Übungsstoff beantwortet.

Aufgabe 1 Eine 1:1-Beziehung der Art



kann man sowohl durch Übernahme des Primärschlüssels von E_2 (als Fremdschlüssel) in E_1 als auch umgekehrt modellieren. Wenn die Beziehung aber nur für wenige Elemente von E_1 definiert ist, enthält die Relation viele Tupel mit Null-Werten für diesen Fremdschlüssel.

Geben Sie Beispiele aus der realen Welt, wo dies der Fall ist und man die Beziehungen deshalb besser in E_2 repräsentiert.

Geben Sie Beispiele, wo es sowohl für E_1 als auch für E_2 viele Elemente gibt, die die Beziehung R nicht „eingehen“. Diskutieren Sie für diesen Fall die Vor- und Nachteile einer separaten Repräsentation der Beziehung als eigenständige Relation.

Aufgabe 2

- Ermitteln Sie den Bekanntheitsgrad der Professoren unter den Studenten, wobei wir annehmen, dass Studenten die Professoren nur durch Vorlesungen oder Prüfungen kennenlernen.
- Ermitteln Sie für die einzelnen Vorlesungen die Durchfallquote als die Anzahl der durchgefallenen Prüflinge relativ zur Anzahl der für diese Vorlesung angetretenen Prüflinge.
Als Variation der obigen Anfrage ermitteln Sie die Durchfallquote bei den einzelnen Professoren.

Aufgabe 3

Beschreiben Sie die Auswirkungen der folgenden Operationen wenn die Beispielausprägung aus Abbildung 1 mit dem Schema aus Abbildung 2 vorgegeben sind:

- **delete from** Vorlesungen **where** Titel = 'Ethik';
- **insert into** prüfen **values** (24002, 5001, 2138, 2.0);
- **insert into** prüfen **values** (28106, 5001, 2127, 4.3);
- **drop table** Studenten;

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen			
VorlNr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Assistenten			
PersNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2134

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Abbildung 1: Beispielausprägung unserer Universitäts-Datenbank

```

create table Studenten
  ( MatrNr      integer primary key,
    Name        varchar(30) not null,
    Semester    integer check (Semester between 1 and 13));

create table Professoren
  ( PersNr      integer primary key,
    Name        varchar(30) not null,
    Rang        character(2) check (Rang in ('C2', 'C3', 'C4')),
    Raum        integer unique);

create table Assistenten
  ( PersNr      integer primary key,
    Name        varchar(30) not null,
    Fachgebiet  varchar(30),
    Boss        integer,
    foreign key (Boss) references Professoren on delete set null);

create table Vorlesungen
  ( VorlNr      integer primary key,
    Titel       varchar(30),
    SWS         integer,
    gelesenVon  integer references Professoren on delete set null);

create table hören
  ( MatrNr      integer references Studenten on delete cascade,
    VorlNr      integer references Vorlesungen on delete cascade,
    primary key (MatrNr, VorlNr));

create table voraussetzen
  ( Vorgänger   integer references Vorlesungen on delete cascade,
    Nachfolger  integer references Vorlesungen on delete cascade,
    primary key (Vorgänger, Nachfolger));

create table prüfen
  ( MatrNr      integer references Studenten on delete cascade,
    VorlNr      integer references Vorlesungen,
    PersNr      integer references Professoren on delete set null,
    Note        numeric(2,1) check (Note between 0.7 and 5.0),
    primary key (MatrNr, VorlNr));

```

Abbildung 2: Das vollständige Universitätsschema mit Integritätsbedingungen

Aufgabe 4

Gegeben sei das folgende Schema:

- Familie: {[Opa, Oma, Vater, Mutter, Kind]}

Hierbei sei vereinfachend vorausgesetzt, dass Personen eindeutig durch ihren Vornamen identifiziert werden. Für ein Tupel [Theo, Martha, Herbert, Maria, Else] soll gelten, dass Theo und Martha entweder die Eltern von Herbert oder von Maria sind – die Großeltern werden also immer als Paar gespeichert, ohne dass ersichtlich ist, ob es die Großeltern väterlicher- oder mütterlicherseits sind. Wir gehen weiterhin davon aus, dass zu einem Kind immer beide Elternteile und beide Großeltern-Paare (also sowohl mütterlicherseits als auch väterlicherseits) bekannt sind.

- Bestimmen Sie alle FDs.
- Bestimmen Sie den/die Kandidatenschlüssel der Relation *Familie*.
- Überführen Sie das Schema unter Verwendung des Synthesealgorithmus in die dritte Normalform.
- Überführen Sie das Schema unter Verwendung des Dekompositionsalgorithmus in die BCNF.