



Übungsblatt 10

Gruppenaufgabe 1

In den CREATE TABLE-Statements zur Erstellung der Universitätsdatenbank werden zahlreiche Integritätsbedingungen umgesetzt.

- a) Benennen und erläutern Sie die Integritätsbedingungen der Tabellen *Professoren* und *Assistenten*.
- b) Erläutern Sie den Unterschied zwischen *set null* und *cascade* in den Foreign Key Constraints der Tabellen *Assistenten* und *hören*.

```
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 1: Das vollständige Universitätsschema mit Integritätsbedingungen



Gruppenaufgabe 2

Vollziehen Sie konkret am Universitätsbeispiel nach, welche Integritätsbedingungen bereits in der ER-Modellierung (Abbildung 2) vorhanden sind und welche erst später mit dem Schema, Abbildung 1 (Gruppenaufgabe 1), festgelegt werden.

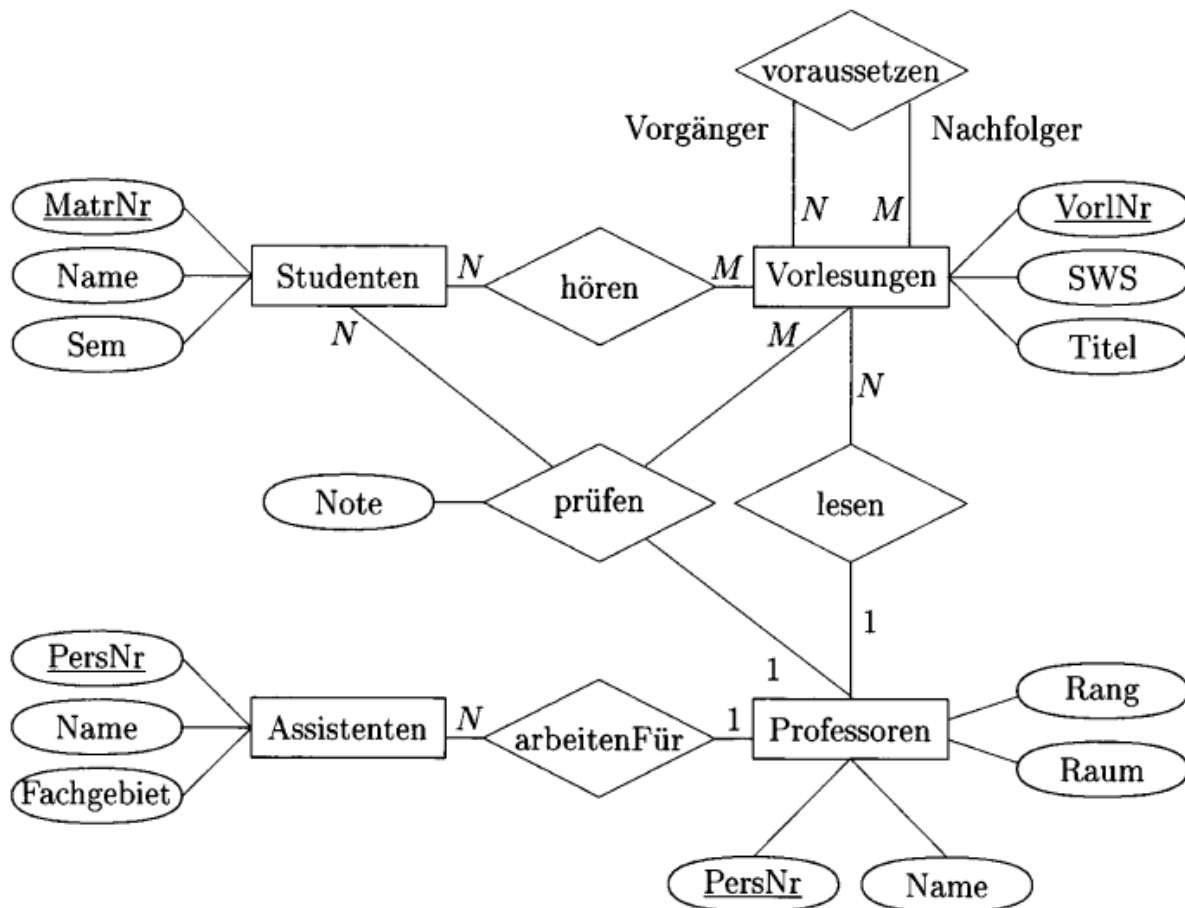


Abbildung 2: ER-Modell des Universitätsbeispiels



Gruppenaufgabe 3

Nachfolgende Aufgaben beziehen sich auf die **Universitätsdatenbank** mit geltendem **Schema aus Übung 7 (schema_rn.sql)**, nicht das Schema aus Gruppenaufgabe 1.

Führen Sie nachfolgende Anweisungen aus. Lesen Sie sorgfältig eventuell auftretende Fehlermeldungen.

1. Versuchen Sie der Tabelle `Assistenten` einen neuen Assistenten hinzuzufügen.
Der Boss soll ein Professor mit `PersNr = 1111` sein, die restlichen Daten können Sie frei wählen.
Was passiert und wieso?
2. Erstellen Sie nun zunächst einen neuen Professor mit `PersNr = 1111` in der Tabelle `Professoren`. Die restlichen benötigten Datenwerte können frei festgelegt werden. Versuchen Sie dann erneut den Assistenten aus 1. hinzuzufügen.
3. Löschen Sie den neuen Professor und Assistenten wieder aus der Datenbank.
4. Versuchen Sie die Tabelle `Vorlesungen` aus der Datenbank zu löschen.
Was passiert und wieso?
5. Versuchen Sie die Vorlesung mit der `VorlNr = 5041` aus der Tabelle `Vorlesungen` zu löschen.
Was passiert und wieso?
6. Welche Constraints hat die Tabelle `pruefen`?
Finden Sie heraus, wie das angezeigt werden kann.
7. Versuchen Sie die Tabelle `pruefen` zu löschen.
Was passiert und wieso?
8. Bauen Sie die Tabelle `pruefen` erneut auf. Ändern Sie aber den Constraint auf die `Vorlesungen` Tabelle in ein `on delete cascade`.
Wie ist das Verhalten nun im Vergleich zum Ausgangszustand?
Befüllen Sie die Tabelle auch wieder mit den Daten. Diese finden Sie im Skript *daten.sql* in Moodle (Übung 7).
9. Versuchen Sie nun nochmals die Vorlesung mit der `VorlNr = 5041` aus der Tabelle `Vorlesungen` zu löschen.
Was passiert und wieso?

Um den ursprünglichen Zustand der Datenbank herzustellen, führen Sie die entsprechenden Skripte *schema_rn.sql* und *daten.sql* aus dem **Moodle-Abschnitt Übung 7** aus.



Hausaufgabe 1

Beschreiben Sie kurz, ob die folgenden Operationen ausgeführt werden können oder wenn nicht, warum diese scheitern.

Vorgegeben sind die **Beispielausprägung aus Abbildung 3** mit dem **Schema aus Abbildung 1** (**Gruppenaufgabe 1**):

- (1) **DELETE FROM** Vorlesungen **WHERE** Titel = 'Ethik';
- (2) **INSERT INTO** pruefen **VALUES** (24002, 5001, 2138, 2.0);
- (3) **INSERT INTO** pruefen **VALUES** (28106, 5001, 2127, 4.3);
- (4) **DROP TABLE** Studenten;

Die nachfolgenden Anfragen beziehen sich auf die folgende Beispielausprägung:

Professoren				Studenten		
PersNr	Name	Rang	Raum	MatrNr	Name	Semester
2125	Sokrates	C4	226	24002	Xenokrates	18
2126	Russel	C4	232	25403	Jonas	12
2127	Kopernikus	C3	310	26120	Fichte	10
2133	Popper	C3	52	26830	Aristoxenos	8
2134	Augustinus	C3	309	27550	Schopenhauer	6
2136	Curie	C4	36	28106	Carnap	3
2137	Kant	C4	7	29120	Theophrastos	2
				29555	Feuerbach	2

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



hören		Assistenten			
MatrNr	VorlNr	PersNr	Name	Fachgebiet	Boss
26120	5001	3002	Platon	Ideenlehre	2125
27550	5001	3003	Aristoteles	Syllogistik	2125
27550	4052	3004	Wittgenstein	Sprachtheorie	2126
28106	5041	3005	Rhetikus	Planetenbewegung	2127
28106	5052	3006	Newton	Keplersche Gesetze	2127
28106	5216	3007	Spinoza	Gott und Natur	2134
28106	5259				
29120	5001				
29120	5041				
29120	5049				
29555	5022				
25403	5022				
29555	5001				

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

Abbildung 1: Beispielausprägungen der Universitäts-Datenbank

Hausaufgabe 2

R	A	B	C	D	S	A	B	C	D	E
	1	blau	2	16		3	4	1	13	128
	3	rot	4	32		3	4	3	15	64
	5	gelb	8	64		9	12	2	14	32
	9	orange	12	256		12	16	NULL	14	16
	12	gelb	16	128		9	12	5	14	16

Die Tabellen R und S seien (mit Integritätsbedingungen) in SQL-Syntax wie folgt definiert:

```
CREATE TABLE R (A INT, B VARCHAR(64), C INT, D INT,
  CONSTRAINT CR1 PRIMARY KEY (A, C),
  CONSTRAINT CR2 CHECK (LENGTH(B) IN (3,4,6,7)),
  CONSTRAINT CR3 CHECK (D IN (1,2,4,8,16,32,64,128,256)),
  CONSTRAINT CR4 UNIQUE (D)
);
```

```
CREATE TABLE S (A INT, B INT, C INT, D INT, E INT,
  CONSTRAINT CS1 PRIMARY KEY (A,E),
  CONSTRAINT CS2 FOREIGN KEY (E) REFERENCES R(D),
  CONSTRAINT CS3 CHECK (D BETWEEN 12 AND 16)
);
```

Hinweis: Die Funktion LENGTH() gibt die Anzahl der Zeichen einer Zeichenkette zurück.



Nachfolgend finden Sie INSERT-Anweisungen, die jeweils eventuell gegen eine Integritätsbedingung verstoßen. Tragen Sie zu jeder INSERT-Anweisung entweder „Verstoß“ sowie den Namen der Integritätsbedingung (z.B. CR1) ein, gegen die beim Einfügen verstoßen wird oder „OK“, wenn gegen keine Bedingung verstoßen wird (und somit das Tupel in die Datenbank eingefügt werden kann).

Jede INSERT-Anweisung soll für sich betrachtet werden, d.h. eventuell erfolgreiche Einfügungen vorangehender Anweisungen werden als zurückgesetzt angenommen.

Anweisung	Liegt ein Verstoß vor, oder kann die Zeile eingefügt werden? <ul style="list-style-type: none">• Verstoß (Name der Bedingung)• OK
a. INSERT INTO R VALUES (1, 'magenta', 45, 512)	
b. INSERT INTO R VALUES (4, 'orange', 32, 256)	
c. INSERT INTO R VALUES (11, 'magenta', 14, 8)	
d. INSERT INTO R VALUES (10, 'gruen', 15, 4)	
e. INSERT INTO S VALUES (1, 2, 3, 12, 256)	
f. INSERT INTO S VALUES (3, 4, 3, 1, 256)	
g. INSERT INTO S VALUES (9, 12, 3, 14, 32)	