Informationssysteme SS 2017
Prof. Dr.-Ing. Stefan Deßloch
Dipl. -Inf. Weiping Qu
TU Kaiserslautern, FB Informatik – Lehrgebiet Informationssysteme
http://wwwlgis.informatik.uni-kl.de
Übungsblatt 6: Ausgabe 23.05.2017, Abgabe 29.05.2017, 16:00 Uhr

Aufgabe 1: Integrität (1 P.)

Gegeben sei das folgende Schema:

Personal: (PNR, Name, Gehalt, Abt, Vorges)

a) Geben Sie das **CREATE TABLE** Statement an, um die Tabelle Personal zu erzeugen. Folgende Integritätsbedingungen sollen gelten:

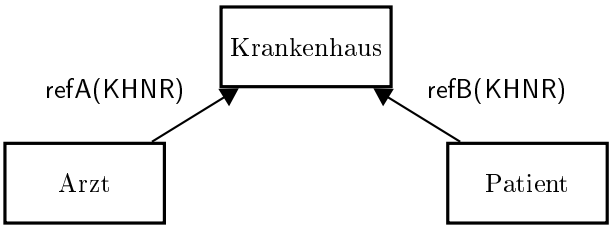
- Name, Abteilung und Gehalt müssen angegeben werden.
- Das Gehalt darf nicht kleiner als 8000 sein.
- Der Vorgesetzte ist entweder NULL oder eine PNR aus der Tabelle Personal.
- Wird eine Person gelöscht, erhalten alle anderen, die diese Person als Vorgesetzten hatten, den Vorgesetzten NULL.
- Ändert sich die PNR einer Person, soll sich diese Änderung ebenfalls auf alle Werte der Vorgess-Spalte von Personen auswirken, die diese Person als Vorgesetzten haben.

b) Formulieren Sie die folgende Bedingung mit einer SQL Assertion: *"Jede Person ist entweder in der gleichen Abteilung wie sein Vorgesetzter oder sein Vorgesetzter ist in der Abteilung 'Management'. "*

Aufgabe 2: Referentielle Aktionen (1 P.)

In diese Aufgabe sollen verschiedene Schemata in Hinblick auf die Eindeutigkeit bei Löschoperationen untersucht werden, wenn unterschiedliche referentielle Aktionen für die jeweiligen Fremdschlüssel definiert werden.

a) Gegeben sei folgendes Schema:



z.B.

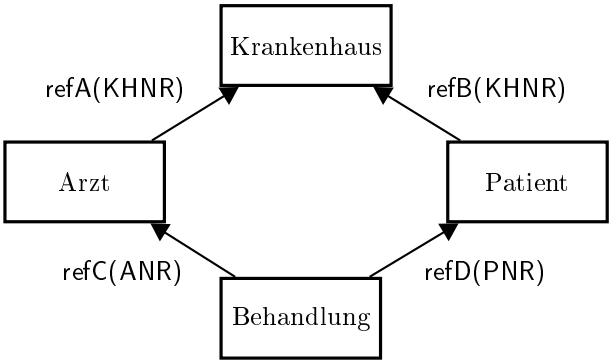
Krankenhaus	
KHNR	**Name**
K1	Pfalzklinik
K2	Waldklinik

Arzt		
ANR	**Name**	**KHNR**
A1	Dr. X	K2
A2	Dr. Y	K1
A3	Dr. Z	K2

Patient		
PNR	**Name**	**KHNR**
P1	Uwe	K1
P2	Ute	K2
P3	Ulla	K1

Diskutieren Sie die Auswirkungen auf das Löschen eines bestimmten Krankenhauses, wenn für refA bzw. refB jeweils eine der referentiellen Aktionen DC (*delete cascade*), DNA (*delete no action*), DR (*delete restrict*), DSN (*delete set null*) oder DSD (*delete set default*) spezifiziert werden.
Existieren irgendwelche Einschränkungen bzgl. DSD oder DSN?
Ist das jeweilige Ergebnis abhängig von der Reihenfolge der referentiellen Aktionen?

b) Diskutieren Sie auch im nächsten Schema die Auswirkungen der unterschiedlichen referentiellen Aktionen, die anstelle von refA, refB, refC und refD eingesetzt werden, wenn ein bestimmtes Krankenhaus (z.B. Waldklinik) gelöscht wird.



Krankenhaus	
KHNR	**Name**
K1	Pfalzklinik
K2	Waldklinik

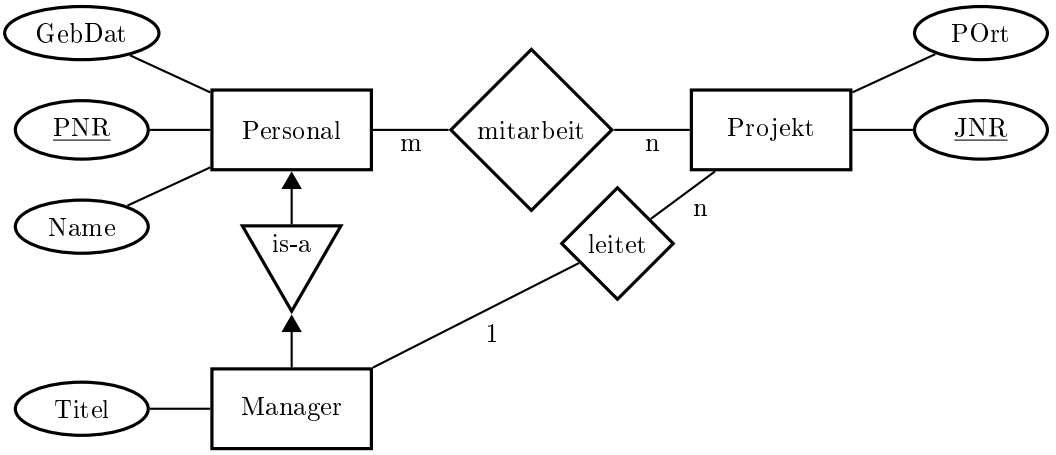
Arzt		
ANR	**Name**	**KHNR**
A1	Dr. X	K2
A2	Dr. Y	K1
A3	Dr. Z	K2

Patient		
PNR	**Name**	**KHNR**
P1	Uwe	K1
P2	Ute	K2
P3	Ulla	K1

Behandlung	
ANR	**PNR**
A2	P3
A1	P2
A3	P2

Aufgabe 3: Trigger (1 P.)

Das folgende E/R-Diagramm ist eine vereinfachte Version des Diagramms aus Übung 2 – Aufgabe 2b). Ein zugehöriges relationales Schema im **Hausklassenmodell** und Beispieldaten sind ebenfalls beigefügt.



PERSONAL: (PNR, Name, GebDat)
MANAGER: (PNR, Name, GebDat, Titel)
PROJEKT: (JNR, POrt, leitet)
MITARBEIT: (PNR, JNR)

PROJEKT.leitet ist Fremdschlüssel zu MANAGER.pnr
MITARBEIT.jnr ist Fremdschlüssel zu PROJEKT.jnr

MITARBEIT.pnr kann nicht als Fremdschlüssel auf Personal definiert werden, da in der Personal-Relation die Manager nicht auftauchen. Würde man doch einen Fremdschlüssel definieren, hätte das zur Folge, dass Manager keine Projekte leiten können, da in diesem Fall die referentielle Integrität verletzt wäre.
Bei dieser Modellierung ist die Eindeutigkeit eines Primärschlüssels innerhalb der Hierarchie nicht garantiert, da Paare von Manager- und Personal-Tupeln existieren können, deren Primärschlüssel identisch sind, die sich aber auf verschiedene Personen beziehen.

a) Das Hausklassenmodell stellt keine globale Eindeutigkeit über Instanzen von Personal und Manager sicher, d.h. Personal- und Managertupel können die gleiche PNR haben. Zum Beispiel haben der Mitarbeiter 'Andy' und der Manager 'Max' beide '001' als PNR bevor sie in die Tabellen eingefügt werden. Um die Eindeutigkeit dieses Primärschlüssels (PNR) innerhalb der Personal-Manager-Hierarchie zu garantieren definieren wir eine Regel, die besagt, dass wir vor die originalen

PNR-Werte ein 'P' oder 'M' schreiben, wenn ein Tupel in die Personal- oder Managertabelle eingefügt wird. Schreiben Sie einen Trigger, der dies implementiert. Dieser soll z.B. bei folgendem Aufruf ausgelöst werden:

```
INSERT INTO Personal VALUES('008', 'Emmy', '2000-08-00')
```

Personal

PNR	Name	GebDat
P001	Andy	2000-01-00
P002	Bob	2000-02-00
P003	Candy	2000-03-00
P004	David	2000-04-00

Manager

PNR	Name	GebDat	Titel
M001	Max	1900-01-00	Dr.
M002	Peter	1900-02-00	Sr.Mgr

Projekt

JNR	POrt	leitet
J1	FRA	M001
J2	MUC	M001
J3	ZUR	M002

Mitarbeit

PNR	JNR
M001	J1
P001	J1
M001	J2
P002	J2
P003	J2
M002	J3
M001	J3
P001	J3
P004	J3

b) Im Hausklassenmodell kann zwischen Mitarbeit(PNR) und Personal(PNR) bzw. Manager(PNR) keine Fremdschlüsselbeziehung definiert werden. Alternativ kann man Trigger verwenden, die die referentielle Integrität implementieren. Erstellen Sie solche Trigger für die Tabellen Manager und Personal, die **ON DELETE CASCADE** realisieren. Die Trigger sollen z.B. bei folgendem Aufruf ausgelöst werden:

```
DELETE FROM Personal WHERE Name='David'
```

c) Um eine vollständige Übersicht über Projektmitarbeiter zu bekommen, wird eine View **FullPersonal** benötigt. Sie muss folgende Voraussetzungen erfüllen:

- Neben PNR, Name und GebDat sollen noch mehr Informationen für jedes Tupel angezeigt werden. Dazu gehören der Name des leitenden Managers, die JNR, die Anzahl der Mitarbeiter (inkl. Manager) des Projekts und der POrt.
- Jedes Projekt wird nur von einem Manager geleitet, der auch in diesem Projekt arbeitet (z.B. ('M001','J1') in Mitarbeit). Daneben kann jeder Manager auch als ganz normaler Mitarbeiter an Projekten mitwirken, ohne es zu leiten (z.B. ('M001','J3') in Mitarbeit). Die View zeigt Informationen für jeden Mitarbeiter, aber nicht den Leiter eines Projekts.
- Mitarbeiter, die in keinem Projekt arbeiten, werden auch in der View aufgelistet.
- Die View ist zuerst nach JNR und dann nach PNR sortiert (beides aufsteigend).

Im Folgenden sehen Sie das Ergebnis der View über den Beispieldaten nach Ausführung der INSERT-/DELETE-Statements aus a) und b).

FullPersonal						
PNR	**Name**	**GebDat**	**Manager**	**Projekt**	**Umfang**	**POrt**
P001	Andy	2000-01-00	Max	J1	2	FRA
P002	Bob	2000-02-00	Max	J2	3	MUC
P003	Candy	2000-03-00	Max	J2	3	MUC
M001	Max	1900-01-00	Peter	J3	3	ZUR
P001	Andy	2000-01-00	Peter	J3	3	ZUR
P008	Emmy	2000-08-00	-	-	-	-

Implementieren Sie die View **FullPersonal** mithilfe der Vorlage.

```
CREATE VIEW FullPersonal (PNR, Name, GebDat, Manager, Projekt, Umfang, POrt) AS
```

;

d) Kann View **FullPersonal** aktualisiert werden? Implementieren Sie einen Trigger, der die folgenden Update-Statements auf die View unterstützt. Das erste Statement ordnet dem Mitarbeiter 'Emmy' das Projekt 'J2' zu und das zweite Statement versetzt den Mitarbeiter 'Candy' von 'J3' nach 'J2'

```
UPDATE FullPersonal SET Projekt = 'J2' WHERE Name = 'Emmy'
UPDATE FullPersonal SET Projekt = 'J2' WHERE Name = 'Candy' and Projekt = 'J3'
```

Mithilfe der Trigger sieht die View nach Ausführung der UPDATE-Statements wie folgt aus:

FullPersonal						
PNR	**Name**	**GebDat**	**Manager**	**Projekt**	**Umfang**	**POrt**
P001	Andy	2000-01-00	Max	J1	2	FRA
P001	Andy	2000-01-00	Max	J2	5	MUC
P002	Bob	2000-02-00	Max	J2	5	MUC
P003	Candy	2000-03-00	Max	J2	5	MUC
P008	Emmy	2000-08-00	Max	J2	5	MUC
M001	Max	1900-01-00	Peter	J3	3	ZUR