

Datenbanksysteme II

SS 2017 – Übungsblatt 5

1. Aufgabe (SQL:2003: Tabellendefinitionen, Anfragen)

Für eine Universitätsanwendung seien u.a. folgende SQL:2003-Typen definiert:

```
CREATE TYPE PersonT
(Name      VARCHAR (40),
 Fak      REF (FakultaetT));
```

```
CREATE TYPE StudentT UNDER PersonT
(MatNr     INT,
 Hauptfach VARCHAR (40),
 Nebenfach VARCHAR (40),
 Bachelor  BOOLEAN);
```

```
CREATE TYPE ProfT UNDER PersonT
(Buero     BueroT,
 Besoldung CHAR(2),
 DRTitel   VARCHAR(20),
 ForschGebiete VARCHAR(20) ARRAY[5]);
```

```
CREATE TYPE BueroT
(Gebaeude VARCHAR(40),
 Stockwerk INTEGER,
 Nummer   INTEGER,
 Telefon   INTEGER);
```

```
CREATE TYPE FakultaeT
(FName     VARCHAR (40),
 Dekan     REF (ProfT),
 Professoren REF(ProfT) ARRAY[50],
 Studenten  REF(StudentT) ARRAY[3000]);
```

- a) Definieren Sie die zur Speicherung von Fakultäten, Studenten und Professoren benötigten Tabellen in SQL:2003 (ohne Tabelle zur Abspeicherung von *BueroT*-Instanzen).
- b) Definieren Sie zu dem Typ *FakultaetT* eine Methode *countStudents(b BOOLEAN) RETURNS INTEGER* die zu einer gegebenen Fakultät (d.h. zu dem Fakultätsobjekt, auf dem sie aufgerufen wird), die Anzahl aller Bachelor-Studenten (*Bachelor = TRUE*) liefert.
- c) Schreiben Sie einen Konstruktor für *StudentT*. Dieser soll neben der Attributinitialisierung das neu generierte Objekt in das ARRAY *Studenten* des entsprechenden Fakultätsobjekts eintragen.
- d) Geben Sie SQL:2003-Anfragen (ohne Join) zur Beantwortung folgender Fragen an:
 1. Wie heißt der Dekan der Fakultät, in der der Student mit Matrikelnummer 234567 immatrikuliert ist?
 2. Finden Sie alle Professoren mit Name und Telefonnummer, die Dekan einer Fakultät sind und die Besoldungsstufe W3 haben.
- e) Definieren Sie in SQL eine Tabelle Dekane vom Typ *ProfT*. Schreiben Sie in SQL einen Ausdruck, der den Dekan der Fakultät 'Informatik' in die Tabelle Dekane einträgt.

2. Aufgabe (Temporale Datenbanken)

Gegeben sei eine **anwendungsversionierte** Tabelle *Schauspieler* und *Serie*, die wie folgt definiert sind:

```
CREATE TABLE Schauspieler (  
    Name VARCHAR(200),  
    SID INT,  
    Startzeit DATE,  
    Endzeit DATE,  
    PERIOD FOR Teilnahme (Startzeit, Endzeit),  
    PRIMARY KEY(Name, Teilnahme),  
    FOREIGN KEY(SID, Teilnahme) REFERENCES Serie (SID, Ausstrahlung)  
)
```

```
CREATE TABLE Serie (  
    SID INT,  
    Name VARCHAR(100),  
    Genre VARCHAR(100),  
    Start DATE,  
    Ende DATE,  
    PERIOD FOR Ausstrahlung (Start, Ende),  
    PRIMARY KEY (SID, Ausstrahlung WITHOUT OVERLAPS)  
)
```

Des Weiteren sei folgende Ausprägung gegeben:

Schauspieler

Name	SID	Startzeit	Endzeit
Kyle MacLachlan	0	1990-04-08	2017-09-04
Walter Koenig	1	1967-09-10	1969-06-04

Serie

SID	Name	Genre	Start	Ende
0	Twin Peaks	Mystery	1990-04-08	2017-09-04
1	Star Trek	SF	1966-09-08	1969-06-04
2	Akte X	Krimi	1993-09-10	2016-02-23
3	How I Met Your Mother	Sitcom	2005-09-19	2014-04-01

- a) Führen Sie folgende Änderungsoperation durch, indem Sie das Statement definieren und die resultierende Tabelle skizzieren:
1. Kyle MacLachlan spielt vom 2010-11-09 bis zum 2011-02-22 sowie vom 2013-02-18 bis zum 2013-05-07 bei How I Met Your Mother mit (Änderung).
 2. Leonard Nimoy hat vom 1966-09-08 bis 1970-06-04 bei Star Trek mitgespielt (Einfügen)
 3. Akte X wurde zwischen 2002-05-20 und 2016-01-24 nicht ausgestrahlt (Löschen).
 4. Das Genre von Akte X soll für den Zeitraum 1993-09-10 bis 1994-05-14 auf SF geändert werden.

b) Formulieren Sie folgende Anfragen:

1. Welche Schauspieler haben in der ersten Staffel von Star Trek (1966-09-08 bis 1967-04-13) mitgespielt. Dabei sollen auch die Schauspieler berücksichtigt werden, die erst im Laufe der Staffel aufgetreten sind.
2. Welche Serien wurden vor 2000-01-01 ausgestrahlt.
3. Welche Schauspieler haben in der Zeit von 2011-02-01 bis 2011-03-01 eine Serie verlassen. (Angabe des Schauspielers und der vorherigen Serie)

3. Aufgabe (XML: DTD, XSD)

a) Geben Sie zu nachfolgender XML-Instanz eine konforme DTD an. Annahme: Element *biblioentry* soll stets mindestens die Elemente *authorgroup*, *title* und *pubdate* enthalten.

```
<?xml version="1.0" ?>
<bib>
  <biblioentry id="Rade97">
    <authorgroup>
      <author>
        <surname>Rade</surname>
        <firstname>Lennart</firstname>
      </author>
      <author>
        <surname>Westergren</surname>
        <firstname>Bertil</firstname>
      </author>
      <othercredit>
        <surname>Vachenauer</surname>
        <firstname>Peter</firstname>
        <address>
          <email>pv@springer.de</email>
        </address>
      </othercredit>
    </authorgroup>
    <title>Springers Mathematische Formeln</title>
    <subtitle>Taschenbuch für Ingenieure, Naturwissenschaftler,
      Informatiker, Wirtschaftswissenschaftler</subtitle>
    <publisher>
      <publishername>Springer-Verlag</publishername>
      <address>
        <city>Berlin</city>
      </address>
    </publisher>
    <pubdate>1997</pubdate>
    <pagenums start="33" end="36" />
  </biblioentry>
</bib>
```

b) Erstellen Sie eine zusätzliche DTD-konforme XML Instanz zu dem Buch: "Zohra Bellahsene, Angela Bonifati, Erhard Rahm: Schema Matching and Mapping, Springer-Verlag Heidelberg, 2011".