

## Aufgabenblatt 2

### 2.1 Aufgabe

#### a)

Die ANSI-SPARC Architektur (Drei-Schema-Architektur) separiert die Benutzeranwendungen und Views von der physikalischen Datenbank. Die Präsentationsschicht, die den Endnutzern und Anwendungen individuelle Benutzersichten bereitstellt. Die logische Schicht (auch konzeptuelle Schicht genannt), die beschreibt welche Daten in der Datenbank gespeichert werden, in welcher Relation sie zueinander stehen. Ziel ist hierbei auch eine redundanzfreie Darstellung aller Informationen. Zuletzt die physische Schicht, wie und wo Daten in der Datenbank gespeichert werden. Ziel ist hier einen effizienten Zugriff auf die Daten zu ermöglichen.

#### 1. Darstellungsebene

Diese Schicht ist extern und nicht direkt ein Teil des DBMS. Hier werden aus dem komplexen Datenmodell bestimmte Teile für den Endnutzer zur Verfügung gestellt. Es kommt darauf an, welcher Nutzer welche Daten für seine Aufgaben benötigt (z. B. benötigt die Personalabteilung die Daten der Mitarbeiter, nicht aber die Verkaufsdaten). Hier spielen auch Sicherheitsaspekte eine wichtige Rolle, denn es soll nicht jeder auf sensible Daten (Personaldaten) zugreifen können. Den Produktionsarbeiter gehen die Personaldaten der Angestellten nichts an. Die Datenbank wird aus Sicht des Endnutzers gesehen.

#### 2. logische Ebene

In dieser Ebene wird der logische Zusammenhang der gespeicherten Daten definiert. Das logische Schema basiert dabei auf dem konzeptuellen Modell des Datenbankdesigners. Das konzeptuelle Modell wird auf der Grundlage der Aussagen der Benutzer erstellt, es bestimmt welche Rahmenbedingungen vorherrschen sollen. Das logische Schema hat dabei die Aufgabe, zwischen der „Welt“ (Ansicht der Benutzer) und der Ansicht der Datenbank zu vermitteln.

#### 3. physikalische Ebene

Bestimmt wo welche und in welcher Form Daten gespeichert werden. Zum Beispiel es ob die Daten auf Festplatte N oder Festplatte M liegen, oder wie

die interne Representation, beispielsweise einem Integer auszusehen hat. Die Struktur der logischen Ebene wird hier auf die Technik abgebildet.

b)

**Physische Datenunabhängigkeit** Dies bedeutet, dass es einer Applikation, die bestimmte Daten benötigt, egal ist, wo diese Daten gespeichert sind. Es wird der Datenbank überlassen, wo sie bestimmte Daten, z.B. bei einem Ausfall einer Festplatte oder häufigem Zugriff auf gleiche Daten, hinschiebt. Eine Änderung des physikalischen Datenschemas soll sich nicht auf die des logischen Datenschemas auswirken.

**Logische Datenunabhängigkeit** (zwischen externe Schicht und Logischer Schicht)

Darunter versteht man, dass sich die Sichtweise auf die Daten ändern kann, dies aber keinen Einfluss auf die Applikation hat. Es ist also egal, ob ich das Alter direkt gespeichert habe oder ob das Alter aus dem das Geburtsdatum errechnet wird. Zudem kann ich Datensätze ergänzen und Daten hinzufügen, ohne dass es die Applikation interessiert. (Beispiel: Fuhrpark wird den Personendaten zugefügt, hat aber keinen Einfluss auf die Applikation die das Alter ausgeben soll) Problematisch ist es, wenn Datensätze verändert werden. Dies hat einen Einfluss auf die Applikation, denn es macht einen Unterschied ob ich nach dem Alter einer Person suche oder den Lebensabschnitt. Zudem können auch Datensätze gelöscht werden, was wiederum dazu führt, dass Applikationen nicht mehr funktionieren.

## 2.2 Aufgabe

a)

Ein Entitätstyp ist eine »Struktur« welche allen der Entitäten diesen Typs gemein ist. So ist ein Entitätstyp durch seinen Namen und die Menge seiner Attribute beschrieben.

**Beispiel** Ein Entitätstyp beschreibt wie Katzen generell aussehen. Allerdings nicht die einzelne Katze.

Eine Entität repräsentiert ein reales Objekt mit einer unabhängigen Existenz. Die einzelne Entität ist immer eine Instanz eines Entitätstyps.

**Bespiel** Ein Stift, ein Auto, der Nachbar mit Namen »Dieter«, u.v.m.

b)

Ein »abgeleitetes« Attribut wird nicht in der Datenbank gespeichert sondern bei einer Abfrage dynamisch von DBMS aus vorhandenen Daten generiert.

**Beispiel** Hierbei kann es sich zum Beispiel um ein Attribut »age« handeln welches von dem Attribut »birthday« abgeleitet wird.

c)

**Beziehungstyp** Benannte Menge von ähnlichen Beziehungen die mit Entitätstypen in Beziehung stehen, die dieselben Attribute besitzen.

**Beziehung** Eine Beziehung beschreibt, dass Entitäten zueinander eine Verbindung haben. Zum Beispiel, dass der Schüler einen Lehrer hat.

**Beziehungsmenge** Menge aller Beziehungsinstanzen eines bestimmten Beziehungstyps.

d)

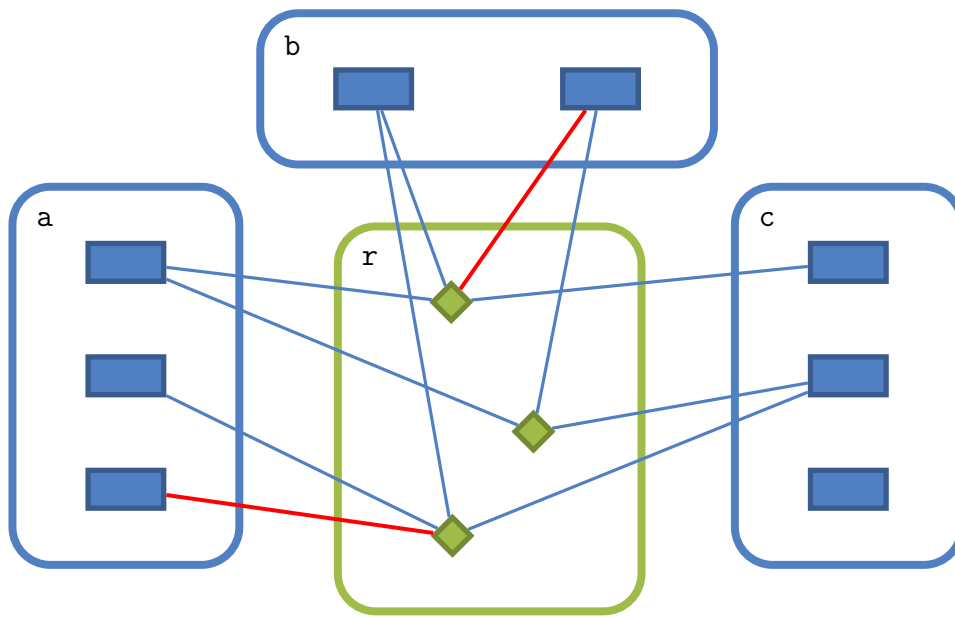
Schlüsselattribute dienen zur eindeutigen Identifikation von Datensätzen. Da es in einer Tabelle keine doppelten Datensätze geben darf, muss, für den Fall das dies vorkommen kann, ein Schlüssel für diese Datensätze definiert werden.

**Beispiel** Man stelle sich eine Tabelle vor in der die Namen von Studenten der TU-Braunschweig gespeichert werden. Nun kann es aber mehr als eine Person mit dem selben Namen geben es würde also ein weiteres Schlüsselattribut benötigt werden. Denkbar ist hierbei die Matrikelnummer des jeweiligen Studenten.

## 2.3 Aufgabe

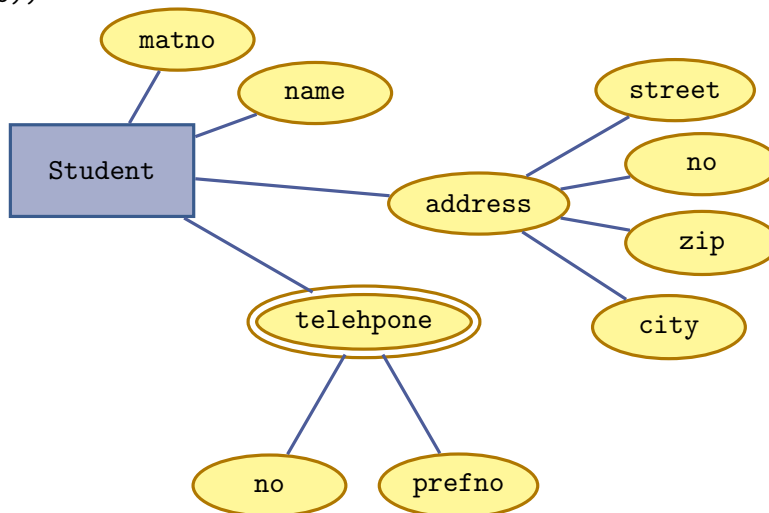
Nach dem gegebenen ER Diagramm, aus der Aufgabenstellung, muss von den Beziehungslosen Element aus der Menge a eine Verbindung in die Menge r geben. In der Menge b müssen alle Elemente mindestens zwei Verbindungen in die Menge r haben, dies wurde hier ergänzt.

Die ergänzten Verbindungen werden hier durch rote Linien gekennzeichnet.



## 2.4 Aufgabe

Student = (matno, name, (address(street, no, zip, city)), telephone(prefno, no))



## 2.5 Aufgabe

Halle = (Id, Name)

Stand = (Nummer)

Firma = (Name)

Raum = (Nummer)

