

STEPHAN MENNICKE
CARL SCHILLER

Relationale Datenbanken I

*Ein Übungsbuch zur Vorlesung im
Wintersemester 2018/2019*

Stand vom
28. März 2019

Technische Universität Braunschweig
Institut für Informationssysteme

Vorwort und Hinweise

Um die *Studienleistung* für die Vorlesung zu bestehen, benötigen Sie **mindestens 50% der Punkte** in den Übungsaufgaben. Um das Modul RDB 1 zu bestehen, müssen Sie die Studienleistung erbringen **und** die Klausur bestehen. Die Übungsaufgaben müssen in **Gruppen zu je zwei Studierenden** gelöst und **wöchentlich bis Donnerstag vor Beginn der Vorlesung** entweder im Vorlesungssaal oder in unserem roten Briefkasten (2. Etage im Informatikzentrum) abgegeben werden. Vergessen Sie nicht, Ihre **Matrikel- und Übungsgruppennummer** auf die Lösung zu schreiben. Wird eine der beiden Informationen vergessen, wird die Lösung automatisch **mit 0 Punkten** bewertet. Lösungen werden in deutscher und englischer Sprache akzeptiert. **Verwenden Sie stets Ihre eigenen Worte!**

Inhaltsverzeichnis

Vorwort und Hinweise	i
1 Eine Einführung	1
2 Datenmodellierung 1	6
3 Datenmodellierung 2	8
4 Sichtenintegration	11
5 Relationales Modell	13

<i>INHALTSVERZEICHNIS</i>	ii
6 Relationale Algebra	18
7 Relationale Kalküle	21
8 SQL 1	25
9 SQL 2	29
10 Weihnachtsübung	31
11 Anwendungsprogrammierung	36
12 Das Letzte	39

Übungsblatt 1
Abgabe bis 25. Okt. 2018

RELATIONALE
DATENBANKEN I
Wintersemester 2018/2019

Eine Einführung

Herzlich Willkommen zu den Übungsaufgaben zur Vorlesung Relationale Datenbanken I im Wintersemester 2018/2019. Die Übungsaufgaben wurden von **Stephan Mennicke** (EMAIL: mennicke@ifis.cs.tu-bs.de) und **Carl Schiller** ausgearbeitet. Jedes Aufgabenblatt wird Ihnen bis zu 25 Punkte einbringen.

Um die *Studienleistung* für die Vorlesung zu bestehen, benötigen Sie **mindestens 50% der Punkte** in den Übungsaufgaben. Um das Modul RDB 1 zu bestehen, müssen Sie die Studienleistung erbringen **und** die Klausur bestehen. Die Übungsaufgaben müssen in **Gruppen zu je zwei Studierenden** gelöst und **wöchentlich bis Donnerstag vor Beginn der Vorlesung** entweder im Vorlesungssaal oder in unserem roten Briefkasten (2. Etage im Informatikzentrum) abgegeben werden. Vergessen Sie nicht, Ihre **Matrikel-** und **Übungsgruppennummer** auf die Lösung zu schreiben. Wird eine der beiden Informationen vergessen, wird die Lösung automatisch **mit 0 Punkten** bewertet. Lösungen werden in deutscher und englischer Sprache akzeptiert. **Verwenden Sie stets Ihre eigenen Worte!**

Aufgabe 1.1 (7 Punkte): Beantworten Sie die folgenden Fragen bzw. lösen Sie die folgenden Teilaufgaben.

- (a) Was ist eine Datenbank (DB)? Was macht ein Datenbankmanagementsystem (DBMS) aus? Nennen und erläutern Sie einen Unterschied. **(3 P)**
- (b) Nennen Sie die drei Benutzertypen von Datenbanksystemen. Grenzen Sie die Typen mit je einem Satz voneinander ab. **(4 P)**

In der Vorlesung haben wir erfahren, dass die Speicherung von Daten im Dateisystem keine Datenbank im Sinne der Vorlesung darstellt. Jedoch wur-

den genau diese vor der Entwicklung des relationalen Modells für Datenbanken im Jahr 1970 als solche benutzt. Im folgenden Abschnitt wollen wir uns anhand eines kleinen Fallbeispiels vergewissern, dass die Nachteile, die sich aus der Verwendung von klassischen Dateien ergeben, überwiegen.

Wir wollen ein kleines Informationssystem für Dozenten und ihre Vorlesungen aufsetzen. Dozenten haben eine Dozentennummer (Dozenten#), einen Namen sowie ein Department, dem sie zugeordnet sind. Jede Vorlesung hat eine Vorlesungsnummer (Vorlesungs#), einen Titel, eine Anzahl an Credits und einen Studiengang, für den die Veranstaltung angeboten wird. Grundsätzlich kann man sich die Speicherung der angegebenen Informationen wie in einem Spreadsheet (vgl. Microsoft's Excel, Google Sheets/Tabellen, usw.) vorstellen:

Vorlesungen				Dozenten		
Vorlesungs#	Titel	Credits	Studiengang	Dozenten#	Name	Department

Wollen wir alle Dozenten der 'Informatik' ausgeben, könnten wir das folgende Programm ausführen:

```
for d in Dozenten do  
  if d.Department='Informatik' then  
    print d.Name  
  end if  
end for
```

Damit wir Vorlesungen mit Dozenten verbinden, werden wir zusätzlich das Semester abspeichern, in dem ein Dozent eine Vorlesung gehalten hat. Es gibt nun mehr als eine Möglichkeit, wie sich diese Beziehung in ein Dateisystem abbilden lässt. Drei hierarchische Möglichkeiten werden in Abbildung 1.1 dargestellt.

Unsere Dateisysteme bestehen aus ein oder mehreren Dateien. Jede Datei ist in Segmente aufgeteilt, die hierarchisch angeordnet sein können. Die Hierarchie wird durch entsprechende Einrückungen kenntlich gemacht. So werden beispielsweise in Abbildung 1.1(a) für jeden Eintrag einer Vorlesung die Dozenten, die die Vorlesung im jeweiligen Semester gehalten haben, dem Vorlesungseintrag untergeordnet. Um die Namen der Dozenten auszugeben, die eine Vorlesung mit dem Titel 'Relationale Datenbanken 1' gehalten haben, müssten wir für dieses Dateisystem folgendes Programm ausführen:

<i>Datei</i>	<i>Segment</i>	<i>Feld</i>
F	Vorlesungen	Vorlesungs# Titel Credits Studiengang Semester
	Dozenten	Dozenten# Name Department

(a)

<i>Datei</i>	<i>Segment</i>	<i>Feld</i>
F	Vorlesungen	Vorlesungs# Titel Credits Studiengang
	Dozenten	Dozenten# Semester
G	Dozenten	Dozenten# Name Department

(b)

<i>Datei</i>	<i>Segment</i>	<i>Feld</i>
F	Dozenten	Dozenten# Name Department
G	Vorlesungen	Vorlesungs# Titel Credits Studiengang
H	gehalten	Vorlesungs# Dozenten# Semester

(c)

Abbildung 1.1: Drei Möglichkeiten, ein Dateisystem über Vorlesungen und ihre Dozenten aufzubauen.

```
for v in F.Vorlesungen do  
  if v.Titel ≠ 'Relationale Datenbanken 1' then  
    continue  
  end if  
  for d in v.Dozenten do  
    print d.Name  
  end for  
end for
```

Der Zugriff auf die Dozenteneinträge über die Variable v ist hier gestattet, da in diesem Dateisystemvorschlag die Dozenten den Vorlesungen untergeordnet sind.

Aufgabe 1.2 (3 Punkte): Ist das angegebene Programm zur Ausgabe der Dozentennamen der Vorlesungen mit Titel 'Relationale Datenbanken 1' auch auf die anderen beiden Dateisysteme anwendbar (siehe Abbildungen 1.1(b) und 1.1(c))? Begründen Sie jeweils Ihre Aussage.

Betrachten wir nun den Vorschlag in Abbildung 1.1(b). Wenn wir die Namen, Departments und die jeweiligen Semester von den Dozenten ausgeben wollen, die die Vorlesungen mit Titel 'Algorithmen und Datenstrukturen' gehalten haben, so könnten wir folgendes Programm nehmen, um die Suchanfrage zu beantworten:

```
for v in F.Vorlesungen do  
  if v.Titel = 'Algorithmen und Datenstrukturen' then  
    for d1 in v.Dozenten do  
      for d2 in G.Dozenten do  
        if d1.Dozenten# = d2.Dozenten# then  
          print d2.Name d2.Department d1.Semester  
        end if  
      end for  
    end for  
  end if  
end for
```

Zunächst wird in Datei F und im Vorlesungssegment nach Vorlesungen mit dem Titel 'Algorithmen und Datenstrukturen' gesucht. Nur in dem Fall, dass diese erfolgreich ist, wird weiterhin nach den aufgeführten Dozenten ($d1$) gesucht. Auf Namen und Departments der jeweiligen Dozenten haben wir nur in Datei G Zugriff, weswegen wir für jedes $d1$ auch über G .Dozenten

iterieren müssen (d2), um die korrekte Ausgabe zu erzeugen. Neben den Vergleichen von Feldern mit Werten, können also auch Felder untereinander verglichen werden (z. B. $d1.Dozenten\# = d2.Dozenten\#$). Dieses Programm hat für das angegebene Dateisystem eine optimale Ausführungszeit, da die inneren Schleifen über die Dozenten nur dann genommen werden, wenn die Variable v tatsächlich einen Eintrag der passenden Vorlesung gefunden hat.

Aufgabe 1.3 (11 Punkte):

- (a) Eine wirklich optimale Ausführung würde nicht für jeden Dozenten der Veranstaltung über alle Dozenten in Datei G iterieren müssen. Welche Technologie wurde Ihnen in der Vorlesung genannt, die hier helfen könnte, um von $d1$ direkt(er) zu den passenden Dozenten in Datei G zu springen? **(1 P)**
- (b) Schreiben Sie zwei verschiedene Programme in Pseudo-Code, die die Namen, Departments und Semester der Dozenten ausgibt, die eine Vorlesung 'Algorithmen und Datenstrukturen' gehalten haben, dieses Mal allerdings für das Dateisystem in Abbildung 1.1(c). Die Unterscheidung muss sich in der Schleifenschachtelung bemerkbar machen. Beschreiben Sie kurz, was Ihr Code tut um das angegebene Ziel zu erreichen. **(6 P)**
- (c) Welches Ihrer beiden Programme hat die bessere Ausführungszeit? Gibt es ein schnelleres? Was müsste dieses anders machen? **oder** Warum ist einer Ihrer Vorschläge bereits optimal? **(4 P)**

Aufgabe 1.4 (4 Punkte): Wir wollen nun auch Daten einfügen, ändern oder löschen. Welche der drei Dateisysteme eignet sich für welche der Operationen am besten. Begründen Sie Ihre Aussagen.

Datenmodellierung 1

Aufgabe 2.1 (3 Punkte): Beantworten Sie die nachfolgenden Fragen in jeweils (mindestens) zwei Sätzen und eigenen Worten.

1. Was steckt hinter dem Konzept 'Universe of Discourse'?
2. Warum benötigen wir Schlüsselattribute in der Datenmodellierung?
3. Wodurch können schwache Entitäten identifiziert werden?

Aufgabe 2.2 (8 Punkte): Erstellen Sie ein Datenmodell in ER-Chen-Notation, basierend auf folgendem Szenario. Falls benötigt, notieren Sie im Diagramm nicht ausdrückbare Integritätsbedingungen an geeigneter Stelle in Textform. Führen Sie gegebenenfalls geeignete Schlüssel ein.

Wir wollen für unseren Übungsbetrieb eine Datenbank nutzen und müssen hierfür das Datenmodell definieren. Wir speichern Studierende, Übungen und Übungsblätter. Übungen werden durch ihr zugehöriges Modul und das Semester, in dem sie stattfinden, identifiziert. Außerdem soll der Name des Betreuers der Übung gespeichert werden. Eine Übung enthält mindestens ein Übungsblatt. Die Übungsblätter einer Übung sind einer Übung zugeordnet und durchnummeriert. Jedes Übungsblatt hat eine maximal zu erreichende Punktzahl und mehrere Aufgabentexte. Über Studierende wird die eindeutige Matrikelnummer, der Name, bestehend aus Vor- und Nachnamen, sowie ein Studiengang gespeichert. Der Studiengang ist entweder 'Informatik', 'Wirtschaftsinformatik', 'Informationssystemtechnik' oder 'Anderer'. Studierende bearbeiten Übungsblätter immer zu zweit. Es soll gespeichert werden, wieviele Punkte Studierende im jeweiligen Übungsblatt erhalten haben und wieviel Prozent der Maximalpunktzahl des Blatts dies entspricht.

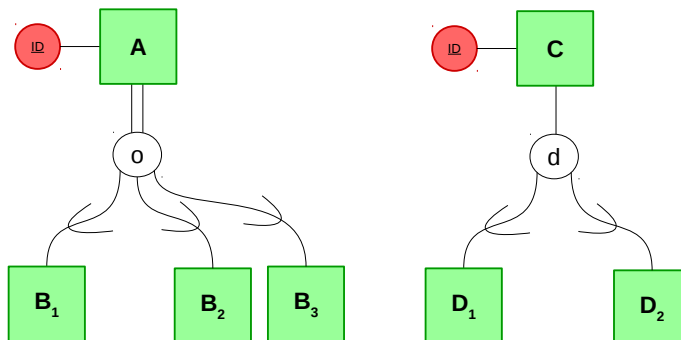
Aufgabe 2.3 (14 Punkte): Erstellen Sie ein Datenmodell in ER-Chen-Notation, basierend auf folgendem Szenario. Falls benötigt, notieren Sie im Diagramm nicht ausdrückbare Integritätsbedingungen an geeigneter Stelle in Textform. Führen Sie gegebenenfalls geeignete Schlüssel ein.

In diesen Fallbeispiel soll es um die Verteilung der Räume an unserer Universität gehen: Unsere Universität ist über mehrere Campus verteilt, die über ihren Namen identifiziert werden. Auf jedem Campus befinden sich verschiedene Gebäude der Universität. Jedes Gebäude hat eine Adresse (Straße, Hausnummer, Postleitzahl). Die einzelnen Räume im Gebäude sind durchnummeriert. In den Räumen befinden sich verschiedene Gegenstände (Tafeln, Beamer, Tische, etc.), die über eindeutig identifizierende Inventar-nummer verfügen.

Die Räume werden auf unterschiedliche Weisen verwendet. Aus der Vorlesung kennen Sie bereits den Aufbau unseres Vorlesungsbetriebs: Ein Modul hat einen eindeutigen Titel und eine Anzahl an Leistungspunkten (Credits). Einige Module bauen auf andere Module auf, z. B. darf das Modul 'Softwareentwicklungspraktikum' (SEP) in Braunschweig nur dann absolviert werden, wenn das Modul 'Software Engineering 1' (SE1) bereits bestanden wurde. Jedes Jahr wird ein Modul von einem Professor unterrichtet. Für einen Professor speichern wir uns seine ID, Vor- sowie Nachnamen und das Institut, dem er angehört. Die Veranstaltungen einer Moduls finden an einem festen Wochentag zu einer festen Uhrzeit in einem bestimmten Raum statt. Jeder Durchgang eines Moduls hat bis zu zwei Klausuren (maximal eine im Sommer- und eine im Wintersemester). Für Klausuren müssen Räume an einem bestimmten Tag zu einer bestimmten Uhrzeit bereitgestellt werden. Einige Räume sind einem Institut zugeordnet. Die Institute an dieser Universität haben einen eindeutigen Namen, von welchem häufig das Kürzel des Instituts abgeleitet ist.

Datenmodellierung 2

Aufgabe 3.1 (3 Punkte): Sie sehen nachfolgend zwei EER-Diagramme. Seien $A, B_1, B_2, B_3, C, D_1, D_2$ jeweils Mengen von Entitäten des jeweiligen Entitätstyps (A von A, B_1 von B_1 , usw.), die den Restriktionen der gegebenen Diagramme genügen.

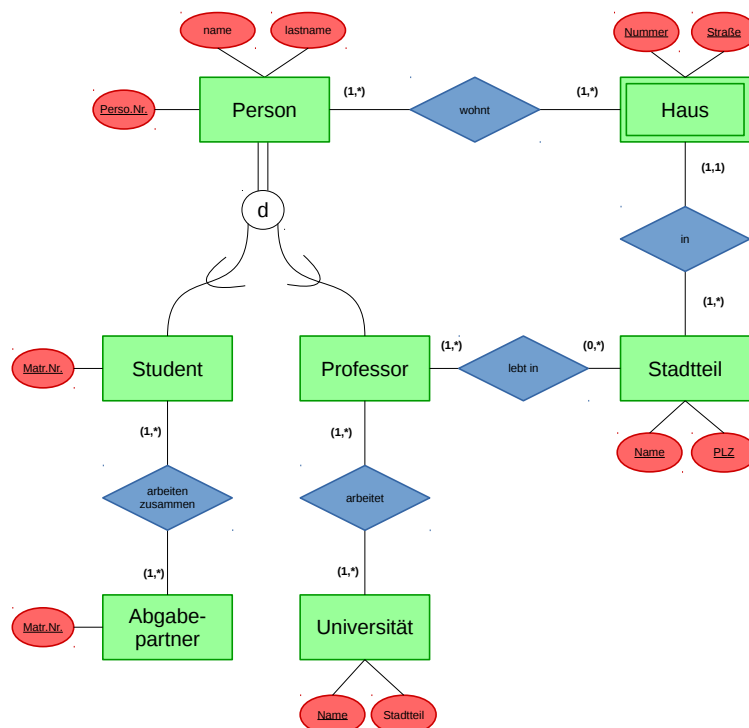


Entscheiden Sie für jede der nachfolgend genannten Eigenschaften ob diese im Allgemeinen gelten.

1. $B_1 \cup B_2 \cup B_3 = A$
2. $B_1 \cap B_2 \neq \emptyset$
3. $|B_1| + |B_2| + |B_3| \leq |A|$
4. $D_1 \cup D_2 = C$
5. $D_1 \cap D_2 = \emptyset$
6. $|D_1| + |D_2| \leq |C|$

Beachten Sie, dass die angegebenen Mengen beliebig gewählt sind, die Diagramme aber erfüllt werden.

Aufgabe 3.2 (6 Punkte): Im folgenden EER-Diagramm sind 6 Fehler enthalten. Diese sind von sowohl syntaktischer als auch inhaltlicher Natur. Benennen und diskutieren Sie die Fehler. Zur Diskussion gehört auch mindestens ein Lösungsvorschlag.



Aufgabe 3.3 (16 Punkte): Erstellen Sie ein Datenmodell basierend auf folgendem Szenario in EER-Chen-Notation. Falls benötigt, notieren Sie im Diagramm nicht ausdrückbare Integritätsbedingungen an geeigneter Stelle in Textform. Führen Sie gegebenenfalls geeignete Schlüssel ein.

Nachdem wir bereits die TU Braunschweig als Datenbankmodell dargestellt haben, ist dieses Mal eine etwas andere Bildungseinrichtung unser Kunde: das *Xaviers Institut für begabte Jugendliche (X-Men)*.

Das Institut speichert für jeden bekannten Mutanten den Vor- und Nachnamen, sowie den Geburtstag und eine Liste von Kräften, die diese besitzen. Mutanten, die jünger als 21 Jahre sind, können sich als Schüler einschreiben.

Schüler schließen die Schule mit 21 Jahren ab und erhalten eine Abschlussnote. Jeder Schüler gehört in genau eine Klasse, von der es auch immer nur eine pro Jahrgang gibt. Für Mutanten, die keine Schüler mehr sind, wollen wir einen Alias speichern, falls es einen solchen gibt. Außerdem haben ältere Mutanten die Möglichkeit als Lehrer und/oder X-Men am Institut zu arbeiten. Lehrer unterrichten Klassen in einem der Räume des Instituts. Räume haben einen Namen und werden durch ihre Raumnummern unterschieden. X-Men sind Teil von Teams, welche über einen eindeutigen Namen und eine Anzahl von Teammitgliedern¹ verfügen. Teams beteiligen sich an Missionen, die über einen geheimen Decknamen und ein Einsatzgebiet verfügen. Das Institut stellt den Teams Hilfsmittel für ihre Mission zur Verfügung. Hilfsmittel sind Ausrüstungsgegenstände, jeweils mit Lager-ID und Beschreibung versehen, oder Fahrzeuge, jeweils mit Typ, Beschreibung und Baujahr. Fahrzeuge sind abgezählt nach Typ (Pkw1, Pkw2, etc.). Es ist unabdinglich, dass vermerkt wird, in welchem Zeitraum ein Team ein Hilfsmittel für eine Mission ausleiht.

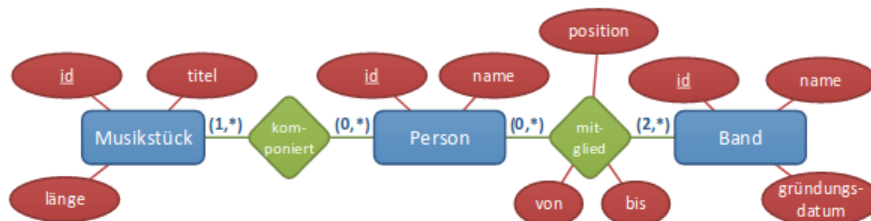
¹mind. drei Mitglieder pro Team

Sichtenintegration

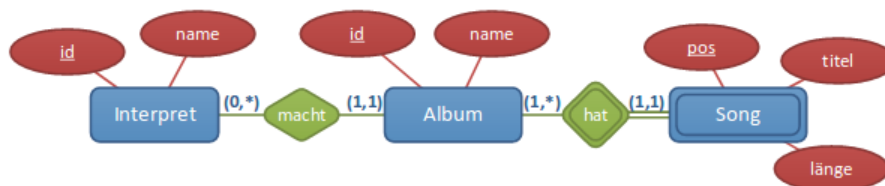
Aufgabe 4.1 (3 Punkte): Beschreiben Sie in wenigstens fünf Sätzen, was es für zwei Schemata bedeutet, zueinander *kompatibel* zu sein. Nennen Sie mindestens drei Eigenschaften.

Aufgabe 4.2 (5 Punkte): In dieser Aufgabe sollen Sie die Modelle (1) und (2) zu einem Modell zusammenführen.

(1)



(2)



Aufgabe 4.3 (17 Punkte): Nachfolgend werden Ihnen zwei Miniwelten präsentiert.

- (a) Erstellen Sie für beide Miniwelten jeweils ein Datenmodell in EER-Chen-Notation. Notieren Sie im Diagramm nicht ausdrückbare Inte-

gritätsbedingungen an geeigneter Stelle in Textform. Führen Sie gegebenenfalls geeignete Schlüssel ein. **(12 P)**

Miniworld A: Wir beschäftigen uns mit der Erfassung aller Information von Menschen mit besonderen Kräften. Für alle diese Person speichern wir uns ihren eindeutigen Alias sowie das Level ihrer Kräfte (0 ist der Normalwert) und ihre Gesinnung (0 ist normal, +1 ist gut, -1 ist böse). Superschurken haben eine böse Gesinnung. Für Superschurken speichern wir uns eine böse Firma, welche sie kontrollieren, sowie die Adressen ihrer geheimen Verstecke und die Anzahl an Waffen, die sich in diesen befinden. Mehrere Handlanger arbeiten für einen Schurken. Handlanger haben eindeutige Namen. Die übrigen Personen mit Kräften sind Helden. Helden besitzen Helfer, Helfer haben keine Kräfte, aber besitzen auch einen Alias. Bis zu zwei Helfer unterstützen einen Helden. Wir speichern auch die Adressen der verschiedenen Hauptquartiere, in welchen die Helden leben. Helden verteidigen ihre Heimat. Eine Heimat hat einen eindeutigen Name (z. B. Gotham) und eine Anzahl an Einwohnern.

Miniworld B: Das landesweite Einwohnermeldeamt führt ein Register über alle Einwohner des Landes mit Namen und einem Wohnort, bestehend aus Postleitzahl, Straße und Hausnummer. Um die Personen auch nach einem Umzug zu identifizieren, gibt man jedem Einwohner eine datenbankweit eindeutige Kennnummer. Alle aufgeführten Wohnhäuser werden darüber hinaus von jemandem besessen, sei es eine einzelne Person oder eine registrierte Firma. Dieser Besitzer vermietet also sein Haus an alle Personen, die dort wohnen. Für eine Firma wird es reichen, nur den eindeutigen vollständigen Namen und die Postleitzahl des Stadtteils zu speichern, in welchem die Firma stationiert ist. Zusätzlich zur Postleitzahl hat jeder Stadtteil auch einen Namen, der allerdings über mehrere Postleitzahlen gültig sein kann.

(b) Führen Sie Ihre beiden Modelle zu einem Modell zusammen. **(5 P)**

Relationales Modell

Nachfolgend ein paar Bemerkungen zur notationellen Konvention in den Übungsaufgaben.

- Wir werden *Relationenschemata* immer mit serifenlosen Großbuchstaben wie R, S, T oder mit Wörtern zusammengesetzt aus Großbuchstaben wie BUCH notieren.
- Attributnamen werden ebenfalls mit serifenlosen Großbuchstaben benannt, allerdings generisch mit Buchstaben vom Anfang des Alphabets, also A, B, C oder auch indiziert, also A_1, A_2 , usw. Alternativ verwenden wir auch hier reale Attributnamen wie Titel oder ISBN.
- Eine Relation R , die dem Schema $R(A_1, A_2, \dots, A_m)$ genügt, wird durch $R(R)$ notiert. Ist aus dem Kontext klar, welchem Schema eine Relation folgt, so kann auch nur der Relationsbezeichner verwendet werden.
- Die Tupel einer Relation $R(R)$ werden durch t, t_1, t_2 , usw. identifiziert. Natürlich folgen die Tupel dem Typ von R , der durch das Schema R definiert ist.
- Solange nicht explizit vorgeschrieben, verfolgen wir in Vorlesung und Übung die *benannte Perspektive* des relationalen Modells. Dies bedeutet zunächst, dass wir Relationenschemata mit benannten Attributen wie A oder A_1 notieren. Außerdem können wir in dieser Notation Tupel $t \in R(R)$ als Funktionen $t : \{A_1, A_2, \dots, A_m\} \rightarrow \{NULL\} \cup \bigcup_{0 < i \leq m} dom(A_i)$ auffassen.
- Gemäß der Funktionsdarstellung von Tupeln wollen wir auf die einzelnen Elemente eines Tupels $t = (a_1, a_2, \dots, a_m) \in R(R)$ über die Attributnamen zugreifen, also $t[A_1] = a_1, t[A_2] = a_2$, usw.

- Anstelle der üblichen runden Klammern $t(\mathbf{A}_1)$ verwenden wir eckige Klammern, da wir auch mehr als ein einzelnes Attribut nennen wollen, also $t[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_m] = (a_1, a_2, a_m)$ oder $t[\mathbf{A}_{42}, \mathbf{A}_m] = (a_{42}, a_m)$.
- Als Kurzschreibweise können wir so auch einzelne Tupel über Teilmengen der Attribute anfragen, also $t[\{\mathbf{A}_1\}] = t[\mathbf{A}_1]$ oder $t[\{\mathbf{A}_m, \mathbf{A}_1\}] = t[\mathbf{A}_1, \mathbf{A}_m]$.

Dass die Tupelelemente nach dem Index des zugehörigen Attributnamens geordnet sind, hat keinerlei Auswirkungen auf die Allgemeinheit. Es sei hier lediglich als Konvention zu verstehen.

Aufgabe 5.1 (3 Punkte): Geben Sie für die folgenden Mengenbeschreibungen Definitionen an. Falls möglich, sollte die Definition extensional formuliert sein. Falls nicht, geben Sie einen Grund an, weswegen nur die Intension erfasst werden kann.

- (a) Die Menge aller positiven ganzen Zahlen.
- (b) Die Menge aller positiven ganzen Zahlen, die kleiner als 5 und durch 3 teilbar sind.
- (c) Die Menge aller Postleitzahlen, die in Deutschland vergeben sind.
- (d) Die Menge aller Primzahlen p , sodass $p + 2$ und $p + 4$ Primzahlen sind.
- (e) Die Menge aller Teilmengen einer gegebenen Menge A .
- (f) Die Menge Ihrer Namen, Matrikel- und Übungsgruppennummer.

Aufgabe 5.2 (1 Punkt): Was ist der Unterschied zwischen einem Relationenschema (engl. relation schema) und einer Relation(-sinstanz) (engl. relation instance)?

Aufgabe 5.3 (3 Punkte): Sei R ein Relationenschema mit $R(A_1, A_2, A_3)$ und $R_1(R), R_2(R)$ zwei Instanzen. Bewerten Sie die folgenden Aussagen bzgl. ihrer Allgemeingültigkeit.

- (a) $R_1(R) \cup R_2(R)$ ist eine Instanz von R .
- (b) $R_1(R) \cap R_2(R)$ ist eine Instanz von R .
- (c) $R_1(R) \setminus R_2(R)$ ist eine Instanz von R .
- (d) $R_1(R) \times R_2(R)$ ist eine Instanz von R .
- (e) $R_1(R) \cup \{()\}$ ist eine Instanz von R . *Beachten Sie:* $()$ ist das leere Tupel. Mit den Mitteln des relationalen Modells, die wir in der Vorlesung und hier kennengelernt haben: Wie kann das leere Tupel definiert werden?

Für die nachfolgenden Aufgaben beschäftigen wir uns näher mit dem Konzept des Schlüssels. In der Vorlesung haben wir bereits zwei Arten von Schlüsseln kennengelernt, die Primär- und die Fremdschlüssel (primary keys, foreign keys). Der Begriff des Schlüssels, ohne weitere Qualifikation, ist einer der grundlegendsten Begriffe der relationalen Datenbanken. Er basiert auf den so genannten *Superschlüsseln*, welche wir nachfolgend definieren.

Definition 1 Seien $R(A_1, A_2, \dots, A_m)$ ein Relationenschema und $R(R)$ eine Relation (oder Relationsinstanz). Eine nicht leere Teilmenge der Attribute von R , $SK \subseteq \{A_1, A_2, \dots, A_m\}$, heißt *Superschlüssel von R* (oder *super key*), falls für alle Tupel $t_1, t_2 \in R$ gilt, dass $t_1[SK] = t_2[SK]$ genau dann gilt, wenn $t_1 = t_2$.

Ein Superschlüssel SK von R heißt *Schlüssel von R* , falls SK minimal ist, d. h. es gibt keine echte Teilmenge von SK , die auch Superschlüssel ist. \square

Beachten Sie, dass sowohl Superschlüssel als auch Schlüssel selbst nur indirekt über Relationenschemata definiert sind. Teilmengen der Attributmenge eines Relationenschemas können für eine konkrete Instanz Schlüssel sein, für eine andere nicht.

Aufgabe 5.4 (7 Punkte): Sei $R(A_1, \dots, A_m)$ ein Relationenschema mit Instanz $R(R)$.

- (a) Wieviele Superschlüssel kann R höchstens haben?

- (b) Wieviele Schlüssel kann R höchstens haben?
- (c) Muss es immer einen Schlüssel für R geben? Begründen Sie Ihre Aussage.
- (d) Aus welchem Grund sollte ein Primärschlüssel immer aus der Menge der Schlüssel gewählt werden, nicht aber aus der Menge aller Super-schlüssel?
- (e) Geben Sie ein Relationenschema und eine Relation an, die mindestens zwei verschiedene Schlüssel verschiedener Kardinalitäten besitzt.

Aufgabe 5.5 (4 Punkte): Sei $\mathfrak{R} = \{R, S\}$ ein relationales Schema mit $R(\underline{A}, \underline{B} \rightarrow S)$ und $S(\underline{C}, D)$. Wir betrachten die relationale Datenbank $D = \{R(R), Z(S)\}$ mit der folgenden tabellarischen Darstellung:

(a) Relation $R(R)$

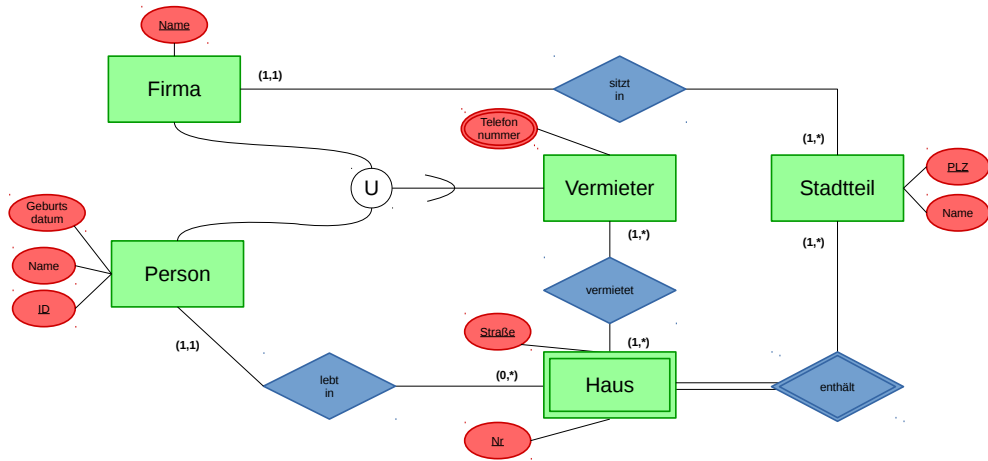
<u>A</u>	<u>B</u>
1	3
2	3
2	6
3	9
4	12

(b) Relation $Z(S)$

<u>C</u>	<u>D</u>
2	4
3	6
6	12
9	18
12	24

- (a) Geben Sie die in den Tabellen dargestellten Relationen in Relation-schreibweise an.
- (b) Darf in den Relationen $R(R)$ und $Z(S)$ beliebig hinzugefügt, gelöscht und geändert werden?
- (c) Nehmen wir an, dass auch D ein Fremdschlüssel zu R darstellt, also S definiert ist als $S(\underline{C}, D \rightarrow R)$. Welche Probleme bezüglich des Hinzufügens oder Löschens können in einer beliebigen Datenbankinstanz D auftreten? Sollten Sie bereits bei (b) Probleme diskutiert haben, klären Sie darüber hinausgehende Probleme. *Denken Sie daran, dass Sie eventuell zwei oder mehr Einträge hintereinander hinzufügen oder entfernen wollen.*

Aufgabe 5.6 (7 Punkte): Konvertieren Sie das folgende EER-Diagramm in ein relationales Schema. Beachten Sie Schlüssel und Fremdschlüssel.

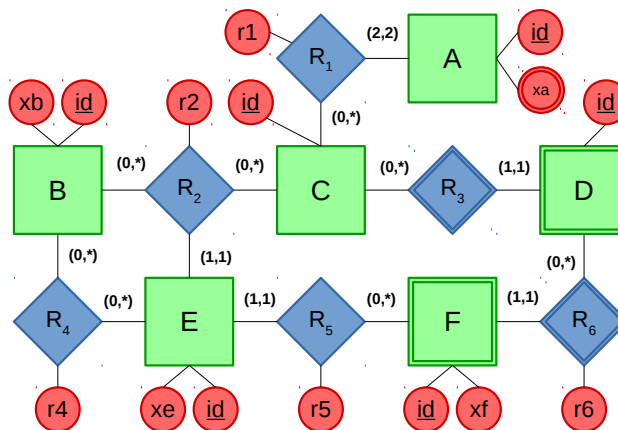


Relationale Algebra

Aufgabe 6.1 (3 Punkt): Sie haben in der Vorlesung zwei alternative Algebren zur Formulierung von sogenannten konjunktiven Anfragen (engl. conjunctive queries oder CQs); die SPC- und die SPRJ-Algebra.

- (a) Während SPC nur drei Operatoren verwendet, hat SPRJ einen vierten, den unären Operator *Renaming* (ρ). Wozu wird dieser benötigt? *Hinweis:* Überlegen Sie, welche Ausdrücke in SPC nur unter Zuhilfenahme der Umbenennung in SPRJ ausgedrückt werden können.
- (b) Aus welchem Grund lässt sich das kartesische Produkt (C) nicht durch eine Kombination der anderen Operatoren, also Selektion und Projektion, ausdrücken?

Aufgabe 6.2 (7 Punkte): Überführen Sie das EER-Diagramm in ein relationales Schema. Beachten Sie Schlüssel sowie Fremdschlüssel und beschreiben Sie nichttriviale Designentscheidungen.



Aufgabe 6.3 (5 Punkte): Formulieren Sie die nun folgenden Anfragen in relationaler Algebra über dem von Ihnen in Aufgabe 6.2 erstellten Schema.

- (a) Finden Sie die *ids* aller Entitäten vom Typ *E*.
- (b) Finden Sie die *ids* aller Entitäten vom Typ *E*, die mit dem *C* in Relation R_2 mit $r_2 > 5$ stehen, das die *id* 42 hat.
- (c) Finden Sie alle *F* mit $xf = \text{'Foobar'}$, die mit wenigstens einem *D* in R_6 stehen.
- (d) Finden Sie alle *D*, die nicht mit allen *C* in Relation R_3 stehen.

Aufgabe 6.4 (8 Punkte): Gegeben sei folgendes relationales Schema einer Datenbank über wissenschaftliche Artikel und ihre Autoren.

ARTICLE (DOI, Title, NoA, Date)
AUTHOR (ORCID, Name)
wrote (A → AUTHOR, P → ARTICLE)
cites (A₁ → ARTICLE, A₂ → ARTICLE)

Eine DOI ist der sogenannte *Digital Object Identifier* und kann als eindeutiges Attribut für wissenschaftliche Artikel verstanden werden. Das Attribut *NoA* steht für *Number of Authors* und beschreibt die Anzahl der Autoren eines Artikels. Bei der *ORCID*-Initiative kann man sich als Autor wissenschaftlicher Papiere eine global eindeutige ID zuordnen lassen.

Formulieren Sie, falls möglich, die nun folgenden Anfragen in relationaler Algebra über dem gegebenen Datenbankschema. Falls es keinen Ausdruck gibt, der mit den in der Vorlesung vorgestellten Mitteln gebildet werden kann, beschreiben Sie, was Ihnen fehlt, um einen Ausdruck zu bilden.

- (a) Finden Sie alle Paare von Autorennamen, die mindestens einen Artikel gemeinsam verfasst haben.
- (b) Welche Artikel wurden am häufigsten zitiert? Finden Sie die Titel.
- (c) Die Erdős-Zahl ist der Abstand der Koautorenschaft zum Mathematiker Paul Erdős. Paul Erdős hat die Erdős-Zahl 0. Ein Autor, der mit Paul Erdős einen Artikel verfasst hat, hat die Erdős-Zahl 1. Ein Autor hat die Erdős-Zahl $k > 1$, falls unter allen Koautoren (Autoren, die einen Artikel mit verfasst haben) die kleinste Erdős-Zahl $k - 1$ ist.

Finden Sie den Autoren in unserer Datenbank mit der größten Erdős-Zahl bzgl. der Autoren in unserer Datenbank.

- (d) Finden Sie die Titel aller Artikel A_1 , die einen Artikel A_2 zitieren, der nach A_1 erschienen ist. Beachten Sie, dass das Datumsformat JJJMMTT ist, also bspw. 20181206 für das Abgabedatum des aktuellen Blatts.
- (e) Finden Sie Titel und Erscheinungsdatum (**Title** und **Date**) der Artikel, die weniger gelistete Autoren haben, als in der Datenbank unter **NoA** angegeben ist.

Relationale Kalküle

Gegeben sei folgendes relationales Schema einer Datenbank über wissenschaftliche Artikel und ihre Autoren.

ARTICLE (DOI, Title, NoA, Date)
AUTHOR (ORCID, Name)
wrote (A → AUTHOR, P → ARTICLE)
cites (A₁ → ARTICLE, A₂ → ARTICLE)

Eine DOI ist der sogenannte *Digital Object Identifier* und kann als eindeutiges Attribut für wissenschaftliche Artikel verstanden werden. Das Attribut NoA steht für *Number of Authors* und beschreibt die Anzahl der Autoren eines Artikels. Bei der ORCID-Initiative kann man sich als Autor wissenschaftlicher Papiere eine global eindeutige ID zuordnen lassen.

Aufgabe 7.1 (6 Punkte): Formulieren Sie, falls möglich, die nun folgenden Anfragen in TRC oder DRC über dem gegebenen Datenbankschema. Falls es keinen Ausdruck gibt, der mit den in der Vorlesung vorgestellten Mitteln gebildet werden kann, beschreiben Sie, was Ihnen fehlt, um einen Ausdruck zu bilden.

- (a) Finden Sie alle Paare von Autorennamen, die mindestens einen Artikel gemeinsam verfasst haben.
- (b) Welche Artikel wurden am häufigsten zitiert? Finden Sie die Titel.
- (c) Die Erdős-Zahl ist der Abstand der Koautorenschaft zum Mathematiker Paul Erdős. Paul Erdős hat die Erdős-Zahl 0. Ein Autor, der mit Paul Erdős einen Artikel verfasst hat, hat die Erdős-Zahl 1. Ein Autor hat die Erdős-Zahl $k > 1$, falls unter allen Koautoren (Autoren, die einen Artikel mit verfasst haben) die kleinste Erdős-Zahl $k - 1$ ist.

Finden Sie alle Autoren mit einer Erdős-Zahl von 2. Gehen Sie davon aus, dass Paul Erdős höchstens einmal in der Datenbank vorkommt.

- (d) Finden Sie die Titel aller Artikel A_1 , die einen Artikel A_2 zitieren, der nach A_1 erschienen ist. Beachten Sie, dass das Datumsformat JJJMMTT ist, also bspw. 20181206 für das Abgabedatum des aktuellen Blatts.
- (e) Finden Sie Titel und Erscheinungsdatum (Title und Date) der Artikel, die weniger gelistete Autoren haben, als in der Datenbank unter NoA angegeben ist.

Aufgabe 7.2 (8 Punkte): Formulieren Sie die folgenden Anfragen, gegeben in relationaler Algebra, in TRC.

- (a) $(\pi_{\text{Name}}\sigma_{\text{ORCID}<9}\text{AUTHOR}) \cup (\pi_{\text{Name}}(\text{AUTHOR} \bowtie_{\text{ORCID}=\text{A}} \sigma_{\text{P}=5}\text{wrote}))$
- (b) $(\text{AUTHOR} \times_{\text{ORCID}=\text{A}} \text{write}) \setminus (\text{AUTHOR} \times_{\text{ORCID}=\text{A}} \text{write} \times_{\text{P}=\text{A}_2} \text{cites})$
- (c) $\mathfrak{F}_{\max(\text{NoA})}\text{ARTICLE}$
- (d) Annahme es gibt nur einen ältesten Artikel:
 $\pi_{\text{Titel}}\text{ARTICLE} \times_{\text{A}_1=\text{DOI}} (\sigma_{\text{A}_2=(\pi_{\text{DOI}}\sigma_{\text{Date}=[\mathfrak{F}_{\min(\text{Date})}\text{ARTICLE}]\text{ARTICLE})}\text{cites})$
- (e) $\pi_{\text{Titel}}\text{ARTICLE} \times_{\text{A}_1=\text{DOI}} (\text{cites} \div (\pi_{\text{A}_1}\sigma_{\text{A}_2=1337}\text{cites}))$

Aufgabe 7.3 (2 Punkte): Sei $R(A, B)$ ein Relationenschema. Betrachten wir folgende Anfrage in DRC über R :

$$q = \{x, y, z \mid R(x, y) \vee R(y, z)\}.$$

Sei $R(R) = \{(1, 2), (2, 3)\}$ eine Instanz. Welche Antwort liefert die obige Anfrage q über $R(R)$ und warum?

Aufgabe 7.4 (9 Punkte): Das Schema entspringt dem Modell aus dem Modellierung-1 Aufgabenblatt. Beachten Sie, dass das Datumsformat JJJJMMTT ist. Die Semesteranzahl sind Ganzzahlen welche am 01.07. und 01.01. für jeden Studenten um einen erhöht werden.

MODUL	(<u>ID</u> , Name, Credits)
VORAUSSETZUNG	(<u>M</u> → MODUL, <u>braucht</u> → MODUL)
KURS	(Jahr, <u>M</u> → MODUL, P → PROF)
PERSON	(<u>ID</u> , Vorname, Nachname, Geburtstag)
STUDENT	(<u>pid</u> → Person, Semesterzahl)
PROFESSOR	(<u>pid</u> → Person, Fachrichtung)
BESTEHT	(<u>S</u> → STUDENT, <u>J</u> , <u>M</u> (J, M) → KURS, Note)

Formulieren Sie die folgenden Anfragen in DRC.

- (a) Was sind die Namen der Professoren, welche einen Kurs des Moduls 'RDB1' gehalten haben und nicht die Fachrichtung „Informationssysteme“ haben.
- (b) Was für Noten haben sich Personen selbst gegeben?
- (c) Welche zwei Module haben nicht dieselben Voraussetzungen, aber ihre Voraussetzungen haben eine gemeinsame Voraussetzung?
- (d) Welche Kurse wurden von einem Professor gehalten, die jünger waren als 30 Jahre.
- (e) Zähle die Namen von Studenten auf, welche am 02.07.2018 ihr halbes Leben lang studiert haben.

Gegeben sei das Datenbankschema $\mathfrak{D} = \{R, S\}$ mit $R(A, B)$ und $S(B, A)$

- (a) $\{t.A \mid \exists u(S(u) \vee R(u))\}$
- (b) $\{t, u \mid R(u) \wedge S(t)\}$
- (c) $\{t \mid \forall u(\neg R(u))\}$
- (d) $\{u \mid \neg R(u)\}$

Gegeben seien das Relationenschema $R(\underline{A}, B, C)$ und folgende Eigenschaften, notiert als DRC-Bedingung:

- (a) $\forall a \exists b, c(R(a, b, c))$
- (b) $\forall a(\exists b, c(R(a, b, c) \rightarrow \neg \exists b', c'(R(a, b', c'))))$
- (c) $\forall a(\exists b, c(R(a, b, c) \rightarrow \neg \exists b', c'(R(a, b', c') \wedge b \neq b')))$

Gegeben seien das Relationenschema $R(\underline{A}, B, C)$ und folgende DRC-Anfragen mit einem beliebigen einstelligen Prädikat F :

- (1) $\{a \mid \exists b, c(R(a, b, c) \wedge F(a))\}$
- (2) $\{a \mid \forall b, c(R(a, b, c) \rightarrow F(a))\}$

Gegeben seien das Relationenschema $R(\underline{A}, B, C)$ und folgende DRC-Anfragen mit einem einstelligen Prädikat F , dass nur dann zu wahr ausgewertet, wenn der übergebene Attributwert tatsächliches A-Attribut der Relation R ist:

- (1) $\{a \mid \exists b, c(R(a, b, c) \wedge F(a))\}$
- (2) $\{a \mid \forall b, c(R(a, b, c) \rightarrow F(a))\}$

Übungsblatt 8
Abgabe bis 13. Dez. 2018

RELATIONALE
DATENBANKEN I
Wintersemester 2018/2019

SQL 1

Bevor wir mit dem eigentlichen Blatt beginnen, möchten wir darauf hinweisen, dass wir Ihnen eine *Probedatenbank* aufgesetzt haben, mit denen Sie die SQL-Anfragen dieses Blatts prüfen können. Die entsprechende Datei, die Sie benötigen, ein so genannter *SQL-Dump*, befindet sich auf der Website zur Vorlesung. Wir verwenden das Open-Source-System PostgreSQL, das Sie sich auf der Website kostenfrei herunterladen können:

<https://www.postgresql.org/download/>

Das ist lediglich das Datenbanksystem. Sie benötigen noch einen Client, mit dem Sie die Datenbank bearbeiten können. Hierfür stehen Ihnen entweder `psql` als Kommandozeilenwerkzeug oder `pgadmin` mit graphischer Oberfläche zur Verfügung. Eine Beschreibung wie Sie den SQL-Dump per `pgadmin` einlesen, finden Sie in der Dokumentation:

https://www.pgadmin.org/docs/pgadmin4/dev/restore_dialog.html

Auch Youtube bietet mehr oder weniger ausführliche Tutorials zu dem System.

Das nachfolgende Schema entspringt dem Modell aus dem Aufgabenblatt Modellierung 1, welches auch im vergangenen Blatt verwendet wurde. Wir haben lediglich den Relationsnamen `BESTEHEN` durch `ABSCHLIESSEN` ersetzt.

MODUL (ID, Name, Credits)
VORAUSSETZUNG (M → MODUL, braucht → MODUL)
KURS (Jahr, M → MODUL, P → PROF)
PERSON (ID, Vorname, Nachname, Geburtstag)
STUDENT (pid → Person, Semesterzahl)
PROFESSOR (pid → Person, Fachrichtung)
ABSCHLIESSEN (S → STUDENT, J, M(J, M) → KURS, Note)

Aufgabe 8.1 (5 Punkte): Formulieren Sie die nachfolgenden Anfragen in SQL über dem gegebenen Schema.

- Finden Sie die Namen aller Studierenden, die bereits ein Modul bestanden haben. Bestanden ist ein Modul, das in einem Kurs zum Modul mit einer Note $\leq 4,0$ abgeschlossen wurde.
- Finden Sie die Namen der Module, für welche die Durchschnittsnote unter 2,7 liegt.
- Finden Sie die Module, welche sich selbst als direkte Voraussetzung haben.
- Wieviele eindeutige Namen haben die Personen in der Datenbank.
- Finden Sie die Namen aller Professoren, die einen Kurs gehalten haben, deren Modul sie schon einmal bestanden haben.

Aufgabe 8.2 (6 Punkte): Gegeben sind die nachfolgenden Ausdrücke in DRC. Formulieren Sie entsprechende SQL-Anfragen über dem gegebenen Schema.

- $\forall = \text{Vorraussetzung}, b = \text{braucht}$
 $\{M_1, M_2 | \exists b_1, b_2 (\forall (M_1, b_1) \wedge \forall (M_2, b_2) \wedge$
 $\exists M_3, b_3, M_4, b_4 (\forall (M_3, b_3) \wedge \forall (M_4, b_4) \wedge b_1 = M_3 \wedge b_2 = M_4 \wedge b_3 = b_4) \wedge$
 $\neg \exists M_5, b_5, M_6, b_6 (\forall (M_5, b_5) \wedge \forall (M_6, b_6) \wedge M_5 = M_1 \wedge M_6 = M_2 \wedge b_5 = b_6))\}$
- $\{Jahr, M | \exists p \text{Kurs}(Jahr, M, p) \wedge \exists v, n, gb (\text{Person}(p, v, n, gb)$
 $\wedge gb > 10000 * (Jahr - 30))\}$
- $\{Vorname, Nachname | \exists p, gb \text{Person}(p, Vorname, Nachname, gb) \wedge$
 $\exists \#Sem (\text{Student}(p, \#Sem) \wedge 20180702 - gb > \#Sem * 10000)\}$

Aufgabe 8.3 (14 Punkte): Formulieren Sie die folgenden Anfragen in SQL über dem gegebenen Schema.

- (a) Finden Sie die Namen der Studierenden, die 30 Credits pro Semester durch Bestehen von Prüfungen gesammelt haben.
- (b) Finden Sie die Durchschnittsnote aller Studierenden, die Kursen von *Diana Jackson* bestanden haben.
- (c) Welche Kurse (Jahr,Modul) haben ein Voraussetzungsmodul, welches innerhalb der letzten 2 Jahre von einem Professor mit gleicher Fachrichtung gehalten worden ist.
- (d) Geben Sie Vorname, Nachname und Fachrichtung der Professoren aus, welche die längste Vorlesungsserie (in Jahren) gehalten haben. Sie können davon ausgehen, dass wenn ein Professor mehrere Kurse gehalten hat, hat er dies auch am Stück getan.
- (e) Welcher Student (Vorname, Nachname) hat innerhalb eines Jahres die meisten Credits erhalten?
- (f) Finden Sie die durchschnittliche Anzahl an Kursen, die von Professoren unserer Uni gehalten worden sind.
- (g) Geben sie die Namen der Studenten, des Moduls und des Voraussetzungsmoduls an, für welche gelten, die Studenten haben das Modul bestanden, aber nicht das Voraussetzungsmodul.

Gegeben seien das Relationenschema PERSON (ID,Name,Vorname) und folgende Anfragen:

RA π_{Name} PERSON

SQL SELECT Name FROM PERSON

```
CREATE TABLE student (  
    matrikelNo INTEGER NOT NULL PRIMARY KEY,  
    name VARCHAR(20),  
    age INTEGER CONSTRAINT adult  
        CHECK (age >= 18),  
    CONSTRAINT matNoFormat  
        CHECK (matrikelNo BETWEEN 999999 AND 100000000)  
)
```

SQL 2

Das nachfolgende Schema entspringt dem Modell aus dem Aufgabenblatt Modellierung 1. Beachten Sie, dass das Datumsformat JJJJMMTT genutzt wird. Die Semesteranzahl sind Ganzzahlen welche am 01.07. und 01.01. für jeden Studierenden um einen erhöht werden.

MODUL (ID, Name, Credits)
VORAUSSETZUNG (M → MODUL, braucht → MODUL)
KURS (Jahr, M → MODUL, P → PROF)
PERSON (ID, Vorname, Nachname, Geburtstag)
STUDENT (pid → Person, Semesterzahl)
PROFESSOR (pid → Person, Fachrichtung)
ABSCHLIESSEN (S → STUDENT, J, M, (J, M) → KURS, Note)

Aufgabe 9.1 (5 Punkte): Überführen Sie das obige relationale Modell in DDL.

Aufgabe 9.2 (8 Punkte): Überführen Sie die nachfolgenden in TRC gegebenen Anfragen in SQL-Anfragen.

(a)
$$\left\{ \begin{array}{l} \text{vor, nach} \mid \exists \text{stud, bd, prof, fach, jahr, modul, note} \forall \text{stud2, note2} (\\ \text{STUDENT}(\text{stud, vor, nach, bd}) \wedge \\ \text{PROFESSOR}(\text{prof, fach}) \wedge \text{PROFESSOR}(\text{stud, fach}) \wedge \\ \text{KURS}(\text{jahr, modul, prof}) \wedge \\ \text{ABSCHLIESSEN}(\text{stud, jahr, modul, note}) \wedge \\ (\text{ABSCHLIESSEN}(\text{stud2, jahr, modul, note2}) \rightarrow \\ \text{note2} \geq \text{note}) \end{array} \right\}$$

$$\begin{array}{l}
 \text{(b)} \left\{ \begin{array}{l}
 prof, stud, note \mid \exists modul, jahr \forall modul2, jahr2, note2 (\\
 \quad KURS(jahr, modul, prof) \wedge \\
 \quad ABSCHLIESSEN(stud, jahr, modul, note) \\
 \quad (KURS(jahr2, modul2, prof) \wedge \\
 \quad ABSCHLIESSEN(stud, jahr2, modul2, note2) \rightarrow \\
 \quad note = note2)
 \end{array} \right. \\
 \\
 \text{(c)} \left\{ \begin{array}{l}
 prof, modul, jahr \mid \forall stud, note (\\
 \quad KURS(jahr, modul, prof) \wedge \\
 \quad (ABSCHLIESSEN(stud, jahr, modul, note) \rightarrow \\
 \quad \neg(note > 4.0))
 \end{array} \right.
 \end{array}$$

Aufgabe 9.3 (12 Punkte): In den letzten Übungsblättern haben Sie versucht, die Erdős-Zahl von Autoren mittels relationaler Algebra und der Kalküle zu bestimmen. In dieser Aufgabe sollen Sie die RDB-Zahl von Modulen bestimmen, wobei das Modul mit der kleinsten ID und dem Namen (Name) 'RDBI' die RDB-Zahl 0 hat. Die direkten Voraussetzungen von 'RDBI' haben die RDB-Zahl 1, deren Voraussetzungen die RDB-Zahl 2, usw. Ein Modul, welches keine Voraussetzungskette besitzt, die in 'RDBI' mündet, hat die RDB-Zahl ∞ .

- (a) Erstellen Sie eine neue Tabelle, durch Angabe eines CREATE TABLE-Ausdrucks, in welcher die Module und ihre RDB-Zahl gespeichert werden können.
- (b) Schreiben Sie einen einzelnen INSERT-Ausdruck mit einer rekursiven Anfrage, welches die neue Tabelle mit den korrekten Daten befüllt.
- (c) Finden Sie alle Module, welche die RDB-Zahl ∞ besitzen.
- (d) Finden Sie alle Namen von Studierenden (STUDENT), die für jede RDB-Zahl $n \neq \infty$ mindestens einen Kurs abgeschlossen haben, die zu einem Modul mit RDB-Zahl n gehören.

Übungsblatt 10
Abgabe bis 10. Jan. 2019

RELATIONALE
DATENBANKEN I
Wintersemester 2018/2019

Weihnachtsübung

Bitte beachten Sie: Dieses Übungsblatt enthält Aufgaben im Wert von 50 Punkten (in Worten: fünfzig Punkten). Das ist die doppelte Anzahl an Punkten im Vergleich zu den übrigen Blättern. Jedes Übungsblatt, und das aktuelle stellt keine Ausnahme dar, geht mit 25 Maximalpunkten in die Bewertung der Studienleistung ein. Da es insgesamt 12 Übungsblätter in diesem Semester geben wird, sind also maximal 300 Punkte zu erreichen. Um die Studienleistung zu bestehen, benötigen Sie die Hälfte der Maximalpunktzahl. Die Extrapunkte, die Sie hier über die 25 Punkte hinaus verdienen, werden Ihrem Punktekonto voll angerechnet.

Das vorliegende Übungsblatt ist an die Struktur der Klausur angelehnt.

Modellierung (10 Punkte)

Aufgabe 10.1 (10 Punkte): Erstellen Sie ein Datenmodell, basierend auf folgendem Szenario, in EER-Chen-Notation. Falls benötigt, notieren Sie im Diagramm nicht ausdrückbare Integritätsbedingungen an geeigneter Stelle in Textform. Führen Sie gegebenenfalls geeignete Schlüssel ein.

Weihnachten steht vor der Tür und in der Werkstatt des Weihnachtsmanns geht mal wieder alles drunter und drüber, sodass, wie jedes Jahr, einige Kinder ihre Geschenke leider nicht pünktlich zu Weihnachten bekommen könnten. Damit es in der Werkstatt am Nordpol in den nächsten Jahren entspannter zugehen kann, plant der Weihnachtsmann eine Digitalisierung seiner Produktionsstätten. An dieser Stelle sind Sie gefragt, das Datenbankdesign der Weihnachtswerkstätten am Nordpol zu übernehmen, damit in Zukunft jedes Kind pünktlich sein Geschenk bekommen kann.

Die Weihnachtswerkstatt umfasst mehrere Gebäude. Für jedes Gebäude erfassen wir das Baujahr. Zur besseren Planung wird zusätzlich die Größe der Gebäude in Quadratmetern benötigt. Gebäude sind entweder Produktionsstellen, in denen Geschenke hergestellt werden, oder Quartiere, in denen die Mitarbeiter wohnen und schlafen. Produktionsstellen werden durch ihre Namen unterschieden, Quartiere sind durchnummeriert.

Zusätzlich werden in der Datenbank Mitarbeiter und Kunden (die Kinder) verwaltet. Beides sind Wesen mit Namen und Geburtsdaten. Alle Wesen mit einem Alter unter 15 Jahren sind Kinder, alle anderen sind Mitarbeiter. Mitarbeiter sind Elfen, die in der Produktion arbeiten, und logistisches Fachpersonal, die Rentiere.

Elfen leben in Häusern und Rentiere schlafen in Ställen auf dem Betriebsgelände am Nordpol. Elfen können in mehreren Häusern leben, Rentiere wohnen in genau einem Stall. Um die maximale Kapazität der Häuser auszunutzen, umfasst die Datenbank die Anzahl der aktuellen Bewohner pro Haus. Damit die Rentiere stets motiviert bei der Arbeit sind, wird dafür gesorgt, dass alle ihr Lieblingsessen bekommen und sie immer dieselbe Position am Schlitten haben. Hierfür wird für jeden Stall das vorhandene Futter gespeichert und für jedes Rentier seine Stammposition am Schlitten.

Elfen stellen an einer Produktionsstätte Produkte her. Produkte haben einen Namen sowie einen Preis und sind einer Weihnachtslieferung zugeordnet, welche eindeutig durch ein Kind und das entsprechende Jahr identifiziert werden können, denn kein Kind wird an Weihnachten mehrfach besucht.

Um jedes Jahr aufs Neue angemessene Geschenke für die Kinder zu finden, wird für jedes Kind ein sogenannter *niceness*-Score berechnet und gespeichert. Wenn dieser Score einen minimalen Schwellwert unterschreitet, bekommt das Kind in diesem Jahr nur die Rute.

Relationales Modell (17 Punkte)

Aufgabe 10.2 (5 Punkte): Überführen sie das von Ihnen erstellte EER-Diagramm in ein relationales Schema. Für diese Aufgabe können Sie nicht überführbare Modellierungsentscheidungen ignorieren.

Aufgabe 10.3 (12 Punkte): Formulieren Sie die folgenden Anfragen über dem von Ihnen erstellten relationalen Schema entsprechend in relationaler

Algebra ((a) bis (e)) und einem relationalen Kalkül ((f) bis (h)).

- (a) Geben Sie das durchschnittliche Alter der Elfen an, die seit höchstens fünf Jahren beim Weihnachtsmann arbeiten.
- (b) Berechnen Sie den durchschnittlichen Platz (in Quadratmetern), den ein Rentier zur Verfügung hat.
- (c) Ein neuer Elf soll ein Haus beziehen, das noch Kapazitäten frei hat. Allerdings wissen Sie, dass die Zahl der Bewohner (`#bewohner`) in jedem Haus ihr Maximum erreicht hat, leider aber nicht aktuell gehalten wurde, d. h. es wohnen tatsächlich weniger Elfen in einigen Häusern als in der Datenbank gelistet sind. Finden Sie alle Häuser, die also als Wohnort für unseren neuen Elfen in Frage kommen.
- (d) Geben Sie für jedes Jahr die durchschnittliche Anzahl an Geschenken pro Kind sowie die Namen der Kinder mit den meisten und wenigsten Geschenken aus.
- (e) Geben Sie Namen, Alter und Nettigkeit der Kinder aus, die bislang Geschenke mit dem maximalen Warenwert (akkumulierte Preise) erhalten hat.
- (f) Geben Sie die Namen aller Mitarbeiter aus, die ausschließlich Kinder mit einem Nettigkeitswert größer 10 beliefern.
- (g) *Legolas* ist unser Elfenprototyp. Neue Elfen bekommen exakt dieselben Aufträge (Produkte mit gleichen Namen) wie *Legolas*, um ihm die Arbeit nachahmen zu können. Finden Sie alle Elfenamen, die in den Jahren 2010 bis 2015 Elfenneulinge waren.
- (h) Geben Sie Namen und Alter der Kinder an, die im Jahr 2017 das teuerste Geschenk erhalten haben.

SQL (10 Punkte)

Aufgabe 10.4 (10 Punkte): Überführen Sie das von Ihnen erzeugte relationale Schema in DDL. Berücksichtigen Sie hier auch sämtliche Bedingungen vom Beschreibungstext, die bislang nicht dargestellt werden konnten.

Normalisierung (13 Punkte)

Aufgabe 10.5 (7 Punkte): Jedes Jahr an Weihnachten wird am IFIS die RDBäckerei in Betrieb genommen und natürlich gibt es auch eine Tabelle, in welcher abgespeichert ist, wer welche Aufgaben hat. Begründen Sie ihre Antworten.

RDBäckerei

ID	Name	Stadt	Haus	Dauer	Aufsicht	Aufgabe
4	René	B	II	4h	Tilo	kneten
1	René	A	I	2h	Tilo	rühren
1	Eike	B	II	4h	Janus	kneten
2	Torben	B	I	3h	Tilo	verpacken
2	Ezz	B	II	4h	Herrmann	kneten
1	Carl	A	II	3h	Tilo	verpacken
1	Markus	D	III	7h	Tilo	sortieren
3	Alexander	C	I	4h	Stephan	kneten
1	Johannes	C	III	6h	Herrmann	bestreuen
1	Eike	C	II	5h	Janus	ausstechen
3	Carl	A	II	3h	Stephan	verpacken

- Finden Sie alle nicht-trivialen funktionalen Abhängigkeiten in der Instanz von RDBäckerei(ID, ...) mit FD: $X \rightarrow Y$ und $|X| \leq 2$.
- Was sind geeignete *Candidate Keys* von der Instanz RDBäckerei?
- In welcher Normalform befindet sich die Instanz von RDBäckerei?

Aufgabe 10.6 (6 Punkte): Gegeben das Relationenschema R (A,B,C,D,E,F) mit den funktionalen Abhängigkeiten:

- $A \rightarrow C, F$
- $C \rightarrow D, F$
- $C, F \rightarrow A, B, E$
- $D \rightarrow B, C$
- $E \rightarrow B, C, D$

- (a) Minimieren Sie die funktionalen Abhängigkeiten von R .
- (b) Erzeugen Sie anhand Ihres Ergebnisses von (a) eine verlustfreie Dekomposition mit dem Ziel, eine möglichst hohe Normalform zu erreichen. Geben Sie die neuen Relationen an.
- (c) Sind die neuen Relationen in BCNF? Begründen Sie ihre Antwort.

Anwendungsprogrammierung

Aufgabe 11.1 (7 Punkte):

- (a) Gegeben sei das Schema $R(A, B, C, D, E, F, G)$ mit folgenden funktionalen Abhängigkeiten:

$$\mathcal{F} = \{B \rightarrow A, AC \rightarrow D, C \rightarrow EF, G \rightarrow F, D \rightarrow B\}$$

Analysieren Sie das gegebene Schema und transformieren Sie es in die höchstmögliche Normalform (maximal Boyce-Codd), ohne dass funktionale Abhängigkeiten verloren gehen. Begründen Sie Ihre Antwort. Sollte es mehr als eine mögliche Lösung geben, geben Sie alle Transformationen an.

- (b) Sei $S(X, Y, Z)$ ein Relationenschema mit der funktionalen Abhängigkeit $X \rightarrow Y$. Betrachten wir die Dekomposition von S : $S_1(X, Y)$ und $S_2(Z)$. Diskutieren Sie die angegebene Dekomposition von S . Bewahrt die Dekomposition Abhängigkeiten? Kann eine gegebene Instanz über S auf das neue Datenbankschema $\mathfrak{S}_\nu = \{S_1, S_2\}$ verlustfrei projiziert werden? Geben Sie zwei verschiedene Instanzen über S an, die auf \mathfrak{S}_ν projiziert ein und dasselbe Ergebnis liefern.

Aufgabe 11.2 (5 Punkte): In dieser Aufgabe verwenden wir das bereits bekannte Schema:

MODUL	(<u>ID</u> , Name, Credits)
VORAUSSETZUNG	(<u>M</u> \rightarrow MODUL, <u>braucht</u> \rightarrow MODUL)
KURS	(<u>Jahr</u> , <u>M</u> \rightarrow MODUL, P \rightarrow PROF)
PERSON	(<u>ID</u> , Vorname, Nachname, Geburtstag)
STUDENT	(<u>pid</u> \rightarrow Person, Semesterzahl)
PROFESSOR	(<u>pid</u> \rightarrow Person, Fachrichtung)
ABSCHLIESSEN	(<u>S</u> \rightarrow STUDENT, <u>J</u> , <u>M</u> , (J, M) \rightarrow KURS, Note)

Gegeben ist folgende SQL Anfrage:

```
WITH A AS(
    SELECT k.m, k.p, a.s,
    FROM kurs AS k
    JOIN abschliessen AS a ON k.jahr = a.j AND k.m = a.m),
B AS(
    SELECT p.id,vorname, nachname
    FROM person AS p
    JOIN A ON A.s = p.id
    --WHERE NOT note <= 4
    GROUP BY p.id
    ORDER BY count( distinct A.m) DESC
    FETCH FIRST 10 ROW ONLY)

SELECT p.pid
FROM professor AS p
WHERE NOT EXISTS(
    SELECT *
    FROM A
    WHERE A.s = ANY(SELECT id FROM B)
    AND A.p = p.pid)
OR p.pid <> ALL(SELECT p FROM A)
```

- (a) Beschreiben Sie in 2-3 Sätzen die Ausgabe der Anfrage.
- (b) Mit welchen Indexen kann die Anfrage beschleunigt werden.

Aufgabe 11.3 (10 Punkte): Wir nutzen in dieser Aufgabe das Schema aus Aufgabe 11.2. In dieser Aufgabe sollen Sie Views erstellen. Begründen Sie Ihre Designentscheidungen(z. B. materialized oder nicht) und geben Sie an, ob Ihre Views updateable sind. Wenn nicht, beschreiben Sie kurz warum.

- (a) Erstellen Sie eine View *credits_count*, in dem sich die Studenten (mit Vor- und Nachnamen) und ihre zur Zeit erworbenen 'credits' befinden.
- (b) Erstellen Sie eine View *info-sys*, in dem sich die Professoren befinden, die sich mit 'information' oder 'system' beschäftigen. Es soll auch markiert werden, wenn beides der Fall ist.

- (c) Erstellen Sie eine View, in dem sich alle Studenten (mit Vor- und Nachnamen) befinden, die jemals eine Klausur bei einem Professor aus (b) geschrieben haben und ihre Durchschnittsnote in deren Klausuren. Hierbei sollen die Noten der Professoren von (b) (bzw. *info-sys*) eine höhere Gewichtung haben. Die Noten der Professoren, die sich sowohl mit 'information' und 'system' beschäftigen, zählen dreifach. Mit nur einem von beiden, doppelt. Mit keinem von beiden, einfach.

Aufgabe 11.4 (3 Punkte): Wir nutzen in dieser Aufgabe das Schema aus Aufgabe 11.2. Gesucht sind alle Namen und Semesterzahlen der Studierenden, die in einem Jahr ein Modul erfolgreich abgeschlossen haben ($\text{Note} \leq 4,0$) und im selben Jahr auch alle (direkten) Voraussetzungen des Moduls abgeschlossen haben (ebenfalls erfolgreich). Schreiben Sie einen entsprechenden TRC-Ausdruck.

Das Letzte

Aufgabe 12.1 (12 Punkte): In dieser Aufgabe erhalten Sie Anforderungen bezüglich des nachfolgenden Schemas, welches Sie in Aufgabe 9.1 bereits in DDL überführt hatten. Implementieren Sie die Anforderungen (a)–(d) in SQL. **Hinweis:** Bei ähnliche Tiggern mit unterschiedlichen Typ (Delete/Update/Insert), reicht es aus sich für einen zu entscheiden.

MODUL (ID, Name, Credits)
VORAUSSETZUNG (M → MODUL, braucht → MODUL)
KURS (Jahr, M → MODUL, P → PROF)
PERSON (ID, Vorname, Nachname, Geburtstag)
STUDENT (pid → Person, Semesterzahl)
PROFESSOR (pid → Person, Fachrichtung)
ABSCHLIESSEN (S → STUDENT, J, M, (J, M) → KURS, Note)

- (a) Jede Person muss Student oder Professor sein.
- (b) Kurse sollen eine Bestehensquote erhalten.
- (c) Module sollen sich nicht selbst als Vorraussetzung haben.
- (d) Semesterzahl soll nicht kleiner sein als die Jahre in denn der Student Kurse abgeschlossen hat.

Aufgabe 12.2 (4 Punkte): Gegeben folgendes DDL-Statement.

```
CREATE TABLE people(  
  id INT PRIMARY KEY,  
  age INT CHECK( 0<age AND age<150),  
  code_name VARCHAR(100) UNIQUE,
```

```
        favorite_subject VARCHAR(4) NOT NULL DEFAULT 'RDB'
    );
CREATE TRIGGER bias
BEFORE INSERT ON people
REFERENCES NEW AS n
FOR EACH ROW
WHEN(n.favorite_subject = 'SE')
BEGIN
INSERT INTO people Values(
    n.age, 22, n.code_name||'2000', 'TBA');
END
```

Die erzeugte Relation ist bereits, wie in Tabelle 12.1 angegeben, gefüllt. Die nachfolgenden INSERT-Statements (a)–(i) werden der Reihe nach aus-

id	age	code_name	favorite_subject
1	144	Genie	TI
2	32	Karla	GDB
3	1	Sheherazade	???
4	30	Gandalf	TI2

Tabelle 12.1: People

geführt und erweitern die Relation, solange sie keinen Fehler ausgeben.

- (a) INSERT INTO people VALUES(5,NULL,'Lotta','BB');
- (b) INSERT INTO people VALUES(6,170,'Miraculix','ALC');
- (c) INSERT INTO people VALUES(7,12,'Urmel','SE');
- (d) INSERT INTO people VALUES(8,100,Findus,'OOO');
- (e) INSERT INTO people VALUES(9,21,'SchlauWie','RDBII');
- (f) INSERT INTO people VALUES(10,'70','Miraculix2000');
- (g) INSERT INTO people VALUES(11,33,'Tiffi',NULL);
- (h) INSERT INTO people VALUES(12,35,'Kim','EN');
- (i) INSERT INTO people VALUES(13,0,'Lotta','SE');

Welche der Statements fügen etwas in die `people`-Tabelle ein und geben keine Fehler aus? Begründen Sie Ihre Entscheidungen.

Aufgabe 12.3 (9 Punkte): Kreuzen Sie bei den entsprechenden Aufgaben die korrekten Antworten an. Es können mehr als eine Option richtig sein. Sie erhalten die Punkte einer Aufgabe nur dann, wenn alle korrekten Antworten angekreuzt und alle inkorrekten Antworten nicht angekreuzt sind.

1. Welche Aussagen über Indexe sind wahr?
 - Ein Index kann über kombinierte Attribute liegen.
 - Indexe beschleunigen Datenbanken.
 - Zusätzliche Indexe können Anfragen nur beschleunigen.
 - Indexe ordnen Tabellen, nach ihren Primary Keys.
2. Welche der folgenden TRC Anfragen sind äquivalent zu $\{a \mid R(a) \wedge \forall b((S(b) \wedge a.i = b.m) \rightarrow \exists c(T(c) \wedge b.j < c.j))\}$?
 - $\{a \mid R(a) \wedge \forall b((S(b) \wedge a.i = b.m) \rightarrow \forall c(\neg T(c) \vee b.j \geq c.j))\}$
 - $\{a \mid R(a) \wedge \forall b(\neg S(b) \vee \exists c(T(c) \wedge b.j < c.j) \vee a.i \neq b.m)\}$
 - $\{a \mid R(a) \wedge \neg \exists b(S(b) \wedge a.i = b.m \wedge \forall c(T(c) \rightarrow b.j \geq c.j))\}$
 - $\{a \mid R(a) \wedge \forall c(S(c) \wedge a.i = c.m \rightarrow \exists b(T(b) \wedge b.j < c.j))\}$
3. Welche Aussage über den Tuple Relational Calculus (TRC) ist wahr?
 - Sind alle Tupelvariablen durch jeweils ein Range-Atom gebunden, so ist die Anfrage sicher.
 - Kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - Sind alle Tupelvariablen durch jeweils ein Range-Atom gebunden und kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - Jede der übrigen Antwortmöglichkeiten ist falsch.
4. Sei $R(\underline{A}, B, C)$ ein Relationenschema. Welche der folgenden Anfragen sind äquivalent zu $\{c \mid \exists a, b(R(a, b, c) \wedge b < c)\}$?
 - $\{c \mid \exists a, b(R(a, b, c) \wedge b \geq c)\}$
 - $\{c \mid \exists a \forall b(R(a, b, c) \rightarrow b < c)\}$

- $\{c \mid \neg \exists a, b (R(a, b, c) \rightarrow b < c)\}$
 - $\{a \mid \forall b, c (R(c, b, a) \rightarrow b < c)\}$
5. Welche Aussage über den Domain Relational Calculus (DRC) ist wahr?
- Sind alle Domänenvariablen durch mindestens ein Range-Atom gebunden, so ist die Anfrage sicher.
 - Kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - Sind alle Domänenvariablen durch jeweils ein Range-Atom gebunden und kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - Jede der übrigen Antwortmöglichkeiten ist falsch.
6. Seien A, B, C, D Mengen mit $A \subseteq B \cup C \cup D$. Welche der folgenden Aussagen sind wahr?
- $A \cap B = A \cap C$
 - $A \cap B \subseteq A$
 - $A \cap B \subseteq B$
 - $A \cup B \subseteq B$
7. Ein Relationenschema $R(A, B, C)$ ist in dritter Normalform, falls
- alle Attribute Schlüsselattribute sind.
 - es in 1NF ist und für jede nicht triviale funktionale Abhängigkeit $X \rightarrow Y$ gilt, Y besteht aus höchstens einem Attribut.
 - es in BCNF ist.
 - es in 1NF ist und für jede nicht triviale funktionale Abhängigkeit $X \rightarrow Y$ gilt, Y ist Schlüsselattribut oder $X \rightarrow \{C, A, B\}$.
8. Welche Aussagen stimmen bezüglich des Uniontypes in der EER-Modellierung?
- Es gilt total participation von Subklasse zu den Superklassen.
 - Subklassen können verschiedene Attribute haben.
 - Die Primary Keys der Superklassen müssen unterschiedlich benannt sein oder unterschiedliche Datentypen haben.

- Es gilt total participation von den Superklasse zur Subklasse.
- 9. Gegeben die Relation $R(A, B, C, D, E, F)$. Die Attributmenge $\{A, F\}$ haben die Superschlüsseleigenschaft über der Instanz $R(R)$, falls gilt:
 - Für alle Tupel $t_1, t_2 \in R$, gilt sind die Attribute $\{A, F\}$ von t_1, t_2 gleich so müssen auch t_1, t_2 gleich sein.
 - $\{A, F\}^+ = R$.
 - Die Werte der Attribute B, C, D, E sind konstant.
 - $\{B, C, D, E\}$ sind funktional abhängig von $\{A, F\}$.

Aufgabe 12.4 (9 Punkte): Kreuzen Sie bei den entsprechenden Aufgaben die korrekten Antworten an. Es können mehr als eine Option richtig sein. Sie erhalten die Punkte einer Aufgabe nur dann, wenn alle korrekten Antworten angekreuzt und alle inkorrekten Antworten nicht angekreuzt sind.

1. Welche Aussagen über Indexe sind wahr?
 - a) Ein Index kann über kombinierte Attribute liegen.
 - b) Indexe beschleunigen Datenbanken.
 - c) Zusätzliche Indexe können Anfragen nur beschleunigen.
 - d) Indexe ordnen Tabellen, nach ihren Primary Keys.
2. Welche der folgenden TRC Anfragen sind äquivalent zu $\{a \mid R(a) \wedge \forall b((S(b) \wedge a.i = b.m) \rightarrow \exists c(T(c) \wedge b.j < c.j))\}$.
 - a) $\{a \mid R(a) \wedge \forall b((S(b) \wedge a.i = b.m) \rightarrow \forall c(\neg T(c) \vee b.j \geq c.j))\}$
 - b) $\{a \mid R(a) \wedge \forall b(\neg S(b) \vee \exists c(T(c) \wedge b.j < c.j) \vee a.i \neq b.m)\}$
 - c) $\{a \mid R(a) \wedge \neg \exists b(S(b) \wedge a.i = b.m \wedge \forall c(T(c) \rightarrow b.j \geq c.j))\}$
 - d) $\{a \mid R(a) \wedge \forall c(S(c) \wedge a.i = c.m \rightarrow \exists b(T(b) \wedge b.j < c.j))\}$
3. Welche Aussage über den Tuple Relational Calculus (TRC) ist wahr?
 - a) Sind alle Tupelvariablen durch jeweils ein Range-Atom gebunden, so ist die Anfrage sicher.
 - b) Kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - c) Sind alle Tupelvariablen durch jeweils ein Range-Atom gebunden und kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - d) Jede der übrigen Antwortmöglichkeiten ist falsch.
4. Sei $R(\underline{A}, B, C)$ ein Relationenschema. Welche der folgenden Anfragen sind äquivalent zu $\{c \mid \exists a, b(R(a, b, c) \wedge b < c)\}$?
 - a) $\{c \mid \exists a, b(R(a, b, c) \wedge b \geq c)\}$
 - b) $\{c \mid \exists a \forall b(R(a, b, c) \rightarrow b < c)\}$
 - c) $\{c \mid \neg \exists a, b(R(a, b, c) \rightarrow b < c)\}$
 - d) $\{a \mid \forall b, c(R(c, b, a) \rightarrow b < c)\}$

5. Welche Aussage über den Domain Relational Calculus (DRC) ist wahr?
- a) Sind alle Domänenvariablen durch mindestens ein Range-Atom gebunden, so ist die Anfrage sicher.
 - b) Kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - c) Sind alle Domänenvariablen durch jeweils ein Range-Atom gebunden und kommt in der Anfrage keine Negation vor, so ist die Anfrage sicher.
 - d) Jede der übrigen Antwortmöglichkeiten ist falsch.
6. Seien A, B, C, D Mengen mit $A \subseteq B \cup C \cup D$. Welche der folgenden Aussagen sind wahr?
- a) $A \cap B = A \cap C$
 - b) $A \cap B \subseteq A$
 - c) $A \cap B \subseteq B$
 - d) $A \cup B \subseteq B$
7. Ein Relationenschema $R(A, B, C)$ ist in dritter Normalform, falls
- a) alle Attribute Schlüsselattribute sind.
 - b) es in 1NF ist und für jede nicht triviale funktionale Abhängigkeit $X \rightarrow Y$ gilt, Y besteht aus höchstens einem Attribut.
 - c) es in BCNF ist.
 - d) es in 1NF ist und für jede nicht triviale funktionale Abhängigkeit $X \rightarrow Y$ gilt, Y ist Schlüsselattribut oder $X \rightarrow \{C, A, B\}$.
8. Welche Aussagen stimmen bezüglich des Uniontypes in der EER-Modellierung?
- a) Es gilt total participation von Subklasse zu den Superklassen.
 - b) Subklassen können verschiedene Attribute haben.
 - c) Die Primary Keys der Superklassen müssen unterschiedlich benannt sein oder unterschiedliche Datentypen haben.
 - d) Es gilt total participation von den Superklasse zur Subklasse.

9. Gegeben die Relation $R(A, B, C, D, E, F)$. Die Attributmengemenge $\{A, F\}$ haben die Superschlüsseleigenschaft über der Instanz $R(R)$, falls gilt:
- a) Für alle Tupel $t_1, t_2 \in R$, gilt sind die Attribute $\{A, F\}$ von t_1, t_2 gleich so müssen auch t_1, t_2 gleich sein.
 - b) $\{A, F\}^+ = R$.
 - c) Die Werte der Attribute B, C, D, E sind konstant.
 - d) $\{B, C, D, E\}$ sind funktional abhängig von $\{A, F\}$.