

1. Übungsblatt

3.0 VU Datenmodellierung

10. November 2015

Allgemeines

In diesem Übungsteil werden Sie den Entwurf einer kleinen Datenbank üben, deren Überführung in das Relationenschema, sowie die relationale Algebra und den Relationenkalkül kennenlernen.

Lösen Sie die Beispiele **eigenständig**, denn bei der Prüfung und in der Praxis sind Sie auch auf sich alleine gestellt. Wir weisen Sie darauf hin, dass abgeschriebene Lösungen mit 0 Punkten beurteilt werden.

Geben Sie ein einziges PDF Dokument ab (max. 5MB). Erstellen Sie Ihr Abgabedokument computerunterstützt. Wir akzeptieren keine gescannten handschriftlichen PDF-Dateien.

Deadlines

bis 27.11.	06:55 Uhr	Upload über TUWEL
ab 04.12.	13:00 Uhr	Feedback in TUWEL verfügbar
bis 06.12.	23:59 Uhr	Reservierung eines Termins für das Tutorengespräch
bis 06.12.	23:59 Uhr	Reservierung eines Termins für das Abgabegespräch

(Terminreservierung für Tutoren- und Abgabegespräch über TUWEL)

Tutorengespräch

1. Sie müssen sich über TUWEL zu einem Tutorengespräch anmelden. Bitte machen Sie das rechtzeitig, je später Sie sich anmelden, umso eingeschränkter ist das Terminangebot.
2. Wenn Sie kein Blatt abgegeben haben, sind Sie nicht zum Tutorengespräch zugelassen.
3. Sie kommen mit Ihrem Studierendenausweis zu der von Ihnen reservierten Zeit vorbei, und absolvieren das Tutorengespräch. Sie erhalten dabei wertvolles Feedback zu Ihrem Übungsblatt. Das Gespräch ist verpflichtend und dauert ca. 30 Minuten.

4. Nützen Sie das Tutorengespräch um wertvolles Feedback zu Ihrem Übungsblatt zu erhalten: Gehen Sie vor dem Gespräch Ihre korrigierte Lösungen durch, identifizieren Sie Unklarheiten und stellen Sie den TutorInnen entsprechende **Fragen**. Trauen Sie sich zuzugeben, wenn Sie etwas nicht verstanden haben. Ihre Fragen haben keinerlei (negativen) Einfluss auf Ihre Note!

Abgabegespräch

1. Sie müssen sich über TUWEL zu einem Abgabegespräch anmelden. Bitte machen Sie das rechtzeitig, je später Sie sich anmelden, umso eingeschränkter ist das Terminangebot.
2. Sie kommen mit Ihrem Studierendenausweis zu der von Ihnen reservierten Zeit vorbei, und absolvieren das Abgabegespräch. Stoffgebiet des Abgabegesprächs sind die mit dem Übungsblatt abgedeckten Themengebiete. Wir setzen voraus, dass Sie sich mit Ihrer korrigierten Abgabe auseinandergesetzt haben. Das Abgabegespräch hat Prüfungscharakter.
3. Sie absolvieren Ihr Abgabegespräch gemeinsam mit anderen KollegInnen. Das Gespräch dauert ca. 60 Minuten.
4. Sie können auf die Abgabe maximal 15 Punkte erreichen. Diese setzen sich wie folgt zusammen:
5 Punkte auf das Übungsblatt
10 Punkte auf das Abgabegespräch
5. Die Assistenten tragen die Punkte des Abgabegesprächs in TUWEL ein und Sie sehen dort, wieviele Punkte Sie bekommen haben.
6. Falls Sie nicht zu Ihrem Gesprächstermin erscheinen, bekommen Sie automatisch 0 Punkte auf das Gespräch.

EER

Aufgabe 1 (EER-Diagramm erstellen) [0.5 Punkte]

Zeichnen Sie folgenden Sachverhalt in einem EER-Diagramm. Verwenden Sie dazu die (min, max)-Notation. Nullwerte sind nicht erlaubt. Manchmal kann es notwendig sein, zusätzliche künstliche Schlüssel einzuführen. Eine Unterstützung bei der Erstellung von EER-Diagrammen bietet das Tool dia (<http://wiki.gnome.org/Apps/Dia>, binaries unter <http://dia-installer.de>). Achtung: im Diagramm Editor ER auswählen!

Die Softwarefirma “Beta-Release” möchte eine Datenbank anlegen um nicht den Überblick über ihre Produkte und deren Komponenten zu verlieren.

In der Datenbank sollen die an der Softwareentwicklung beteiligten Personen gespeichert werden. Jede Person hat eine Sozialversicherungsnummer (SVNR), einen Nachnamen (NNAME), und eine Personalnummer (PNR). Außerdem soll der Raum (RAUM) gespeichert werden, in welchem die Person derzeit arbeitet. Sowohl die Personalnummer, als auch die Sozialversicherungsnummer reichen aus um eine Person eindeutig zu identifizieren.

Von den verschiedenen Arten an Personen sollen drei in der Datenbank abgebildet werden: Software Entwickler, Qualitätsmanager, und Level-Designer.

Weiters sollen die Projekte von “Beta-Release” gespeichert werden. Jedes Projekt wird eindeutig identifiziert durch seine Bezeichnung (BEZEICHNUNG) gemeinsam mit der Projektnummer (PROJNR). Weiters besitzt jedes Projekt noch ein Budget (BUDGET). Jedes Projekt wird von genau einem Software Entwickler geleitet, wobei jeder Software Entwickler maximal fünf verschiedene Projekte leiten darf.

Jede Software ist eindeutig durch eine id (ID) identifiziert. Weiters wird ein Titel (TITEL) festgehalten, und um was für eine Art von Software es sich handelt (TYP). Darüber hinaus soll gespeichert werden welche Personen im Rahmen welcher Projekte an welcher Software gearbeitet haben. Dabei soll nur Software in die Datenbank aufgenommen werden die von mindestens einer Person in mindestens einem Projekt bearbeitet wurde.

Mit Hilfe der Datenbank sollen außerdem die verschiedenen Versionen der erstellten Software verwaltet werden. Eine Version ist eindeutig identifiziert durch die entsprechende Software sowie eine Versionsnummer (VNR) (Hinweis: Das bedeutet, dass die Versionsnummer alleine im Allgemeinen nicht ausreicht um eine Softwareversion eindeutig zu identifizieren.) Es muss zu jeder Software jeweils mindestens eine Version existieren. Zu jeder Softwareversion können darüber hinaus eine beliebige Anzahl an Builds existieren. Diese können eindeutig identifiziert werden durch die Softwareversion und eine Buildnummer (BNR).

Jede Softwareversion beinhaltet ein oder mehrere Pakete. Zu jedem Paket wird sein Name (NAME), seine Version (VERSION), und die Quelle (QUELLE) woher es stammt gespeichert, wobei der Name und die Version gemeinsam ein Paket eindeutig identifizieren. Darüber hinaus werden die Abhängigkeiten zwischen Paketen vermerkt, also welches Paket welche anderen Pakete als Voraussetzungen benötigt.

Ein Paket enthält ein oder mehrere Files, wobei ein File eindeutig identifiziert ist durch das Paket in dem es sich befindet, und seinen Namen (FNAME). Zusätzlich sollen sein Inhalt (INHALT) und der Dateityp (FTYPE) gespeichert werden. Zu manchen Files (jedoch nicht allen) muss außerdem ein Copyright-Vermerk (VERMERK) gespeichert werden.

Ein File kann durch Abspaltung von (maximal) einem anderen File entstanden sein. In so einem Fall soll vermerkt werden, welche Person das File aus

welchem anderem File abgespalten hat, und wann dies passiert ist (DATUM).

Es werden drei Arten von Files unterschieden: Source-Files – für diese soll ihre Länge (LOC) gespeichert werden – Grafik-Files, und Sound-Files.

Für Source-Files soll vermerkt werden, in welcher Programmiersprache sie verfasst sind, wobei in jedem Source-File nur eine Programmiersprache verwendet werden darf. Eine Programmiersprache wird eindeutig identifiziert durch ihren Namen (NAMEN). Es soll außerdem für jeden Software Entwickler gespeichert werden, welche Programmiersprache diese Person beherrscht, und wie gut (LEVEL). Dabei muss jeder Software Entwickler mindestens eine Programmiersprache beherrschen.

Zu einem Bugreport soll seine Beschreibung (BESCHREIBUNG) gespeichert werden sowie die Information, wann der Report gemeldet wurde (EINGANG). Jeder Bugreport besitzt außerdem eine eindeutige Tracking Nummer (TRNR), und ist genau einem Build zugeordnet. Bevor ein Bug bearbeitet wird, muss er von einem Qualitätsmanager bestätigt werden. Es soll nicht nur vermerkt werden, wer einen Bug bestätigt hat, sondern auch wann das passiert ist (DATUM). Es sollen jedoch auch Reports gespeichert werden können, die noch nicht bestätigt wurden. Für behobene Bugreports wird zusätzlich ein Kommentar (KOMMENTAR) vermerkt, sowie von welchem Softwareentwickler der Fehler behoben wurde, und wann (DATUM). Dabei wird jeder Bug von genau einem Entwickler behoben. Zusätzlich kann ein behobener Bug durch maximal einen Qualitätsmanager geprüft werden. Zu jeder Prüfung soll eine (BEWERTUNG) und das Datum (DATUM) der Prüfung gespeichert werden.

Aufgabe 2 (EER-Beispiel selbst erstellen) *[0.5 Punkte]*

Ihr Kollege Herr Mustermann hat Sie gebeten, ihm ein wenig Nachhilfe zu geben. Erstellen Sie für ihn ein kleines Übungsbeispiel zum Thema EER-Modellierung. Das Beispiel sollte mindestens 5 Entitytypen, 5 Beziehungstypen, eine Generalisierung und einen schwachen Entitytyp enthalten. Vielleicht bringen Sie in dem Beispiel auch noch eine rekursive oder eine ternäre Beziehung unter. Sie erhalten die Punkte für die textuell ausformulierte Angabe sowie für die Musterlösung. (Trauen Sie sich durchaus kreativ bei der Wahl Ihres Beispiels zu sein.)

Aufgabe 3 (Überführung ins Relationenschema) *[0.5 Punkte]*

Überführen Sie das EER-Diagramm aus Abbildung 1 ins Relationenschema. Nullwerte sind nicht erlaubt. Verwenden Sie möglichst wenig Relationen. Unterstreichen Sie sämtliche Primärschlüssel, schreiben Sie die Fremdschlüssel kursiv und stellen Sie sicher, dass ein Fremdschlüssel eindeutig der passenden Relation zugeordnet werden kann.

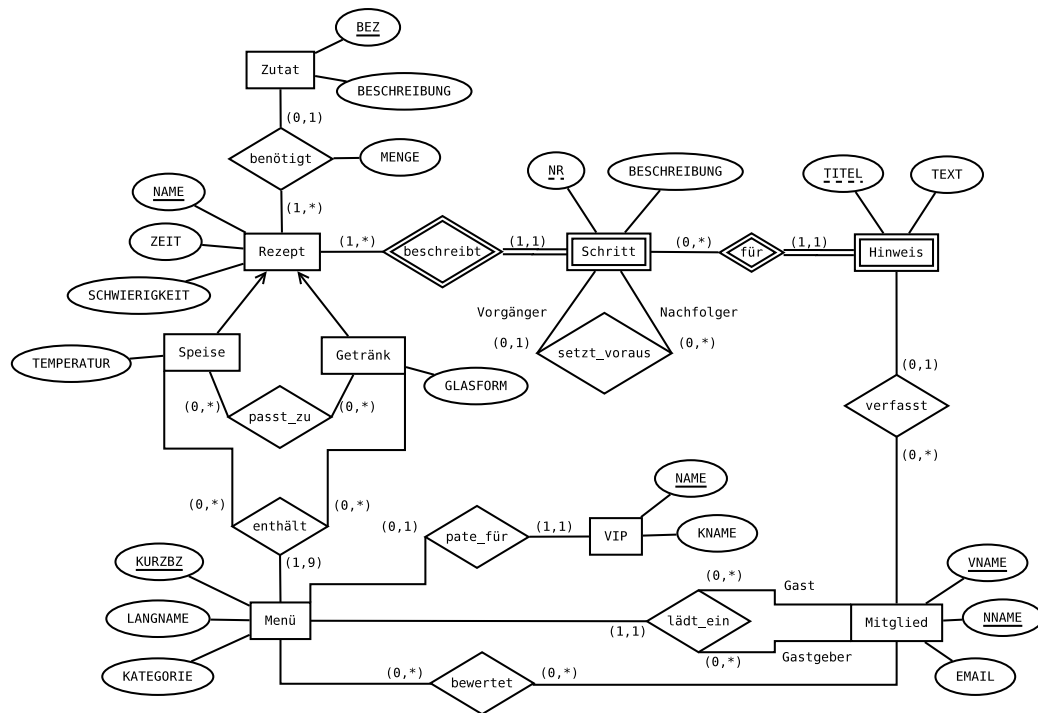


Abbildung 1: EER-Diagramm zu Aufgabe 3

Aufgabe 4 (EER und Relationenschema) [0.6 Punkte]

Gegeben ist das EER-Diagramm aus Abbildung 2 (welches eine grobe Beschreibung von quantifizierten aussagenlogischen Formeln darstellt), sowie die unten angeführte Übersetzung in ein Relationenschema und die in den darunterliegenden Tabellen dargestellte Ausprägung für dieses Schema.

- Bestimmen Sie, ob die Übersetzung in das Relationenschema korrekt ist. Falls nicht, erklären Sie bitte weshalb, und wie die richtige Übersetzung aussehen sollte (es ist nicht nötig eine vollständig verbesserte Version anzugeben. Eine Beschreibung der Fehler und Korrekturen ist ausreichend).
- Bestimmen Sie, ob die in den Tabellen angegebenen Ausprägungen alle im EER-Diagramm spezifizierten Bedingungen erfüllen. D.h., werden z.B. die im EER-Diagramm angegebenen Schlüssel und Kardinalitäten eingehalten, etc.? Finden Sie mindestens vier Verletzungen.
- Bestimmen Sie für die im EER-Diagramm mit (a), (b), (c), (d) bezeichneten Kardinalitäten die genauesten Werte in der (min,max) Notation, welche von den gegebenen Ausprägungen erfüllt werden.

(Beispiel: für (e) wäre die Antwort (0,2) da zum Beispiel die Formel f_{21} nie in der Beziehung "subfl" mit einem binären Operatorausdruck steht (= Minimum),

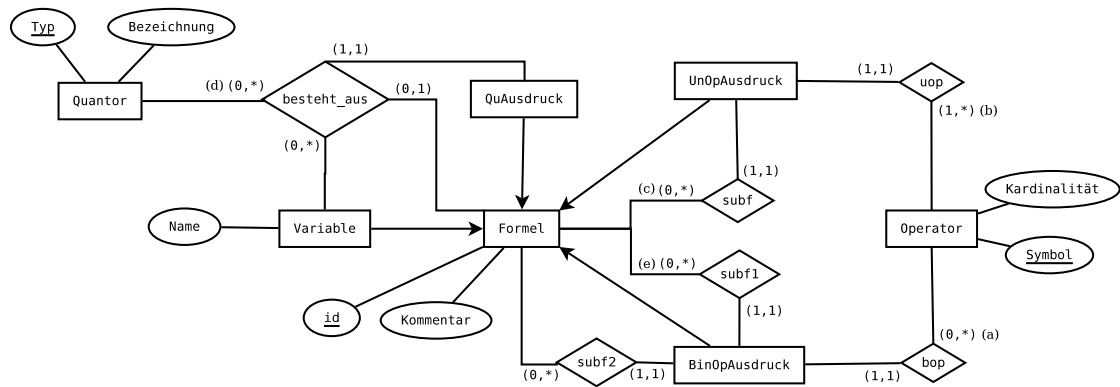


Abbildung 2: EER-Diagramm zu Aufgabe 4

während die Formel f_1 zweimal in dieser Beziehung vorkommt (= Maximum).)

Quantor(TYP, BEZEICHNUNG)

Operator(SYMBOL, KARDINALITÄT)

Formel(ID, KOMMENTAR)

Variable(*Formel.ID*, NAME)

UnOpAusdruck(*Formel.ID*, *Operator.SYMBOL*, *Subformel: Formel.ID*)

BinOpAusdruck(*Formel.ID*, *Operator.SYMBOL*, *Subformel1: Formel.ID*,
Subformel2: Formel.ID)

QuAusdruck(*Formel.ID*)

besteht_aus(*Formel.ID*, *Variable.ID*, *Quantor.TYP*, *Subformel.ID*)

Operator	
<u>SYMBOL</u>	Kardinalität
\neg	1
\wedge	2
\vee	2
\subset	2
\supset	2
\equiv	2
\neq	2
\uparrow	2
\downarrow	2

Quantor	
<u>TYP</u>	BEZEICHNUNG
\forall	Allquantor
\exists	Existenzquantor

Formel	
ID	KOMMENTAR
f_1	foo
f_2	bar
f_3	foo
f_4	var
f_6	a
f_7	b
f_8	c
f_9	true
f_{11}	d
f_{12}	e
f_{13}	var
f_{14}	bar
f_{15}	wahr
f_{16}	f
f_{17}	g
f_{18}	h
f_{19}	i
f_{20}	gilt
f_{21}	h

QuAusdruck	
ID	
f_{14}	
f_{13}	
f_{21}	

Variable	
ID	NAME
f_1	A_1
f_2	A_2
f_3	A_3
f_4	A_4
f_5	A_5

UnOpAusdruck		
ID	SYMBOL	Subf.ID
f_6	\neg	f_1
f_{10}	\neg	f_1
f_{12}	\neg	f_3
f_{16}	\neg	f_5
f_{19}	\neg	f_4
f_{21}	\subset	f_{12}

BinOpAusdruck			
ID	SYMBOL	Subf1.ID	Subf2.ID
f_7	\vee	f_6	f_2
f_8	\supset	f_1	f_2
f_9	\equiv	f_7	f_8
f_{11}	\vee	f_1	f_{10}
f_{15}	\equiv	f_{14}	f_{13}
f_{17}	\supset	f_4	f_5
f_{17}	\supset	f_5	f_6
f_{18}	\wedge	f_{17}	f_{16}
f_{20}	\supset	f_{18}	f_{19}

besteht_aus			
ID	Var.ID	Quantor.TYP	Subformel.ID
f_{14}	f_1	\forall	f_{11}
f_{13}	f_3	\exists	f_{12}
f_{14}	f_2	\exists	f_{15}

Relationale Algebra - Relationenkalkül

Um Ihnen die Erstellung Ihrer Abgabe zu den folgenden Aufgaben zu erleichtern, haben wir unter <http://dbai.tuwien.ac.at/education/dm/resources/symbols.html> eine Liste mit den wichtigsten Symbolen der relationalen Algebra zusammengestellt. Sie können diese per copy/paste in Ihr Word/LibreOffice/OpenOffice/... Dokument einfügen. Zusätzlich sind die entsprechenden L^AT_EX Befehle vermerkt.

Aufgabe 5 (Primitive Operatoren) [0.5 Punkte]

Die Anti-Selektion $\sigma'_F(R)$ liefert alle Tupel aus R , welche F nicht erfüllen. Nehmen Sie an, dass zwei Relationen S und T gegeben sind. Hierbei umfasst das Schema von S $g + s$ viele Attribute $(S_1, \dots, S_s, G_1, \dots, G_g)$ und das Schema von T $g + t$ viele Attribute $(T_1, \dots, T_t, G_1, \dots, G_g)$. Drücken Sie $\sigma_F(S)$ mittels $\sigma'_F(S)$ und der restlichen primitiven Operatoren (π , \times , ρ , $-$, \cup) aus. Drücken Sie den Join/Operator (\bowtie) mittels σ' und der restlichen primitiven Operatoren aus (d.h., drücken Sie $S \bowtie T$ aus).

Betrachten Sie die Mengenoperation $R \circ S$, welche voraussetzt, dass R und S das gleiche Schema haben. $R \circ S$ ist definiert als die Menge aller Tupel aus $R \cup S$, welche

nicht in R oder nicht in S enthalten sind. Drücken Sie $R \cap S$ mittels \circ und der primitiven Operatoren ohne \cup aus.

Beschreiben Sie weiters einen neuen, zusammengesetzten relationalen Operator in natürlicher Sprache und veranschaulichen Sie dessen Anwendung an einem Beispiel. Drücken Sie diesen Operator anschließend mit Hilfe der primitiven Operatoren aus. Der neue Operator soll aus zumindest drei primitiven Operatoren zusammengesetzt werden.

Aufgabe 6 (Äquivalenzen) [0.5 Punkte]

Gegeben sind folgende relationale Ausdrücke über $R(\underline{ABC})$, $S(\underline{DE})$, $T(\underline{BD})$ und $U(\underline{AB})$. Überprüfen Sie, ob die folgenden Ausdrücke in relationaler Algebra äquivalent sind. Falls ja, so erklären Sie warum. Falls nein, geben Sie eine Begründung an.

- (a) $r_1 : \pi_A(R \bowtie \rho_{A \leftarrow D}(S))$ und $r_2 : \pi_A(R \times S)$
- (b) $r_3 : \pi_{A,B}(R) \bowtie U$ und $r_4 : \pi_{A,B}(R \bowtie U)$
- (c) $r_5 : \sigma_{B=1 \vee A > 7}(R \bowtie U)$ und $r_6 : \sigma_{A > 7}(R \bowtie U) \cap \sigma_{B=1}(R \bowtie U)$
- (d) $r_7 : \sigma_{(A \neq 3 \vee B < 5) \wedge C=1}(R)$ und $r_8 : \sigma_{C=1}(\sigma_{A \neq 3}(R) \cup \sigma_{B < 5}(R))$
- (e) $r_9 : (\pi_D(S) - \pi_D(T)) - \pi_D(S)$ und $r_{10} : \pi_D(\sigma_{D < 4 \wedge D > 7}(S \bowtie T))$

Aufgabe 7 (Größenabschätzung) [0.5 Punkte]

Gegeben sind die Relationen $R(\underline{ABCD})$ mit x Tupeln, $S(\underline{BDE})$ mit y Tupeln und $T(\underline{ACD})$ mit z Tupeln. Weiters sind folgende Ausdrücke in relationaler Algebra gegeben:

$$\begin{array}{ll} r_1 : \pi_{B,D}(\sigma_{A < 2}(\rho_{F \leftarrow D}(T) \bowtie S)) & r_2 : \pi_A(R \bowtie S) \cup \pi_A(T) \\ r_3 : (\pi_{B,D}(R) \cap \pi_{B,D}(S)) \bowtie T & r_4 : (R \bowtie T) \bowtie S \end{array}$$

Geben Sie die minimale bzw. maximale Größe (= Anzahl der Tupel) der durch die folgenden Annahmen entstehenden Relationen an und begründen Sie Ihre Antwort.

- (a) Betrachten Sie r_1 unter der Annahme $y = 7$ und $z = 5$.
- (b) Betrachten Sie r_2 unter der Annahme $x = 7, y = 5, z = 8$.
- (c) Betrachten Sie r_3 unter der Annahme $x = 4, y = 6, z = 9$. Außerdem gilt dass $\pi_{B,D}(R)$ und $\pi_{B,D}(S)$ mindestens zwei Tupel gemeinsam haben.
- (d) Betrachten Sie r_4 unter der Annahme $1 \leq x \leq y \leq z$.

Aufgabe 8 (Formalisieren von Anfragen) [0.5 Punkte]

Gegeben ist eine Datenbank über parlamentarische Ausschüsse in Form des folgenden Relationenschemas:

P = Politiker(PNR, Vorname, Nachname, Geburtsjahr)
A = Ausschuss(AID, Bezeichnung, *Politiker.PNR*)
N = nimmt_teil(*Politiker.PNR*, *Ausschuss.AID*, Sitzungen)

Die in der Relation **A** gespeicherte PNR bezeichnet den Ausschussvorsitzenden.
Drücken Sie folgende Anfragen in relationaler Algebra aus.

- (a) Gesucht sind die Nachnamen der Vorsitzenden von allen Ausschüssen, in denen ein Politiker, der nach 1990 geboren wurde, teilnimmt.
- (b) Gesucht sind Vorname und Nachname aller Politiker, die nach 1990 oder vor 1960 geboren wurden, selbst keinen Ausschüssen vorsitzen, und nicht an dem Ausschuss „Tabaksteuer“ teilnehmen.
- (c) Betrachten Sie nun den folgenden Ausdruck in relationaler Algebra. Beschreiben Sie in eigenen Worten, welches Ergebnis dieser Ausdruck liefert.

$$\pi_{\text{Nachname}} \left(P \bowtie \left(\pi_{\text{PNR}, \text{AID}}(N) \div \pi_{\text{AID}} \left(A \bowtie \sigma_{\text{Geburtsjahr}='1981'}(P) \right) \right) \right)$$

Aufgabe 9 (Relationenkalkül) [0.5 Punkte]

Übersetzen Sie die Anfragen, die Sie in Aufgabe 8 (a) und Aufgabe 8 (b) in relationaler Algebra formuliert haben, in sichere Ausdrücke des Tupel- und Domänenkalküls. Geben Sie bei jeder der Anfragen an, in welchem Kalkül sie formuliert ist.

Gesamter Stoff

Aufgabe 10 (Vermischtes) [0.4 Punkte]

Ihr Kollege Karl Konfus erklärt einige wichtige Zusammenhänge, die er sich für die Prüfung merken will. Leider hat er manche Dinge falsch verstanden, und ist sich bei anderen Dingen selber nicht ganz sicher. Finden Sie die Fehler in seinen Aussagen, begründen Sie warum die Erklärungen richtig oder falsch sind, und stellen Sie die falschen Aussagen richtig. Beantworten Sie außerdem seine Fragen und erklären beziehungsweise begründen Sie Ihre Antworten kurz.

- (a) Wenn in einer Relation **R(ABC)** sowohl die Werte von **AC** als auch von **B** ausreichen um jedes Tupel eindeutig zu identifizieren (aber **A** und **C** alleine jeweils nicht), dann muss **B** als Schlüssel gewählt werden, da **B** weniger Attribute enthält als **AC**.
- (b) Für das Zusammenfassen von Relationen während der Übersetzung von EER Diagrammen in ein relationales Schema bietet die Angabe von Funktionalitäten in EER Diagrammen mittels $M : N$, $1 : N$, $N : 1$, und $1 : 1$ die selbe Information wie die (\min, \max) Notation. D.h. es können jeweils die selben Relationen zusammengefasst werden so dass die Notwendigkeit NULL-Werte benutzen zu müssen ausgeschlossen ist.
- (c) Handelt es sich bei den folgendem Ausdrücken um gültige Anfragen – im Tupel- oder Domänenkalkül – über dem Schema **R(ABC)**, **S(AD)**, und falls ja, in was für einer Anfragesprache wurde sie formuliert?

- (1) $\{t \mid t \in R \wedge \exists s, u([s, u] \in S \wedge t.A = s)\}$
- (2) $\{[t.B] \mid t \in R \wedge \exists s \in S(s.D > 4)\}$