

# 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

<b>bis 27.11.</b>	<b>06:55 Uhr</b>	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.

**Lösung:** Siehe Abbildung 1.

Anmerkungen:

- Bei “Person” hätte auch PNR als Schlüssel ausgewählt werden können.

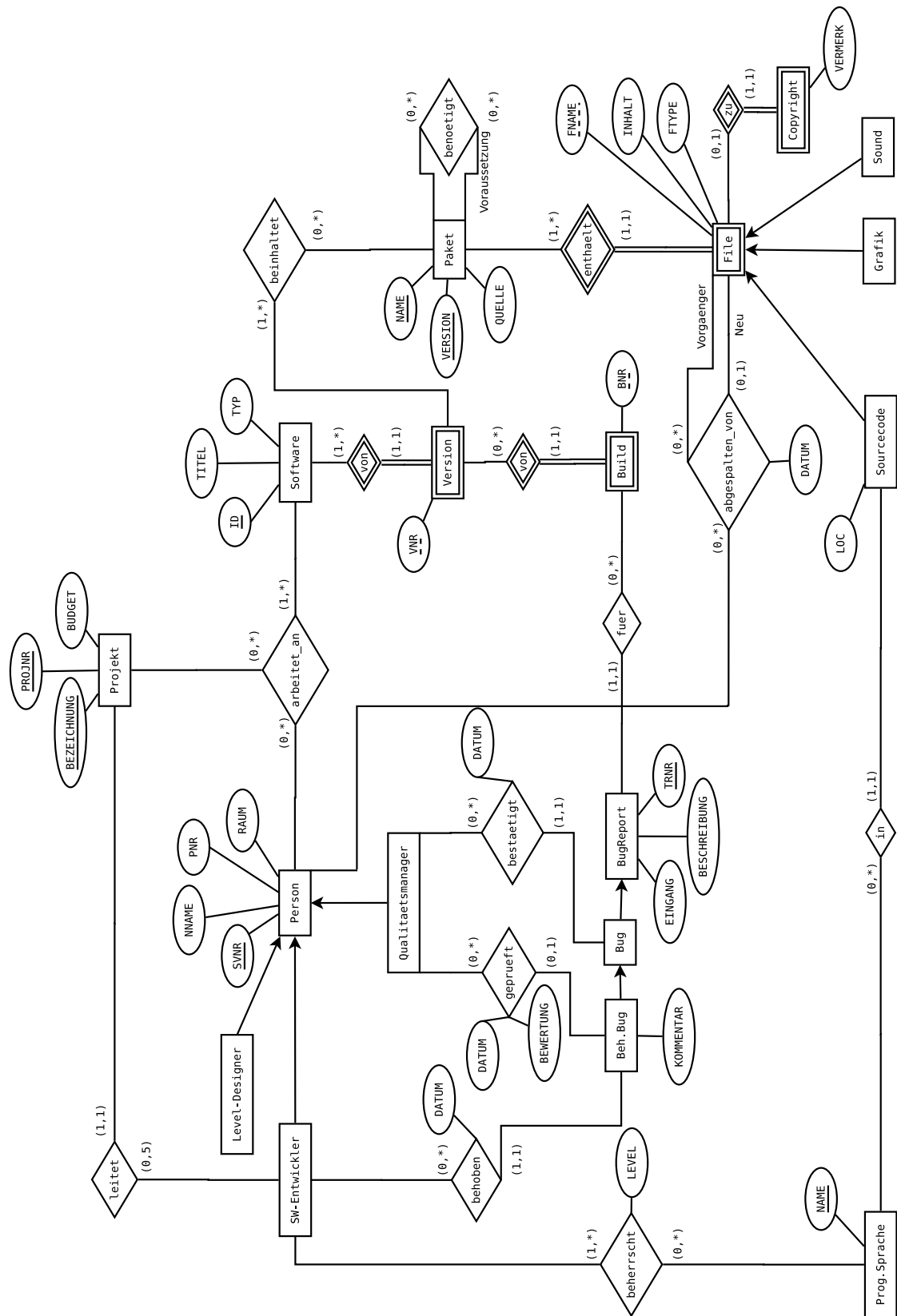


Abbildung 1: Lösung von Aufgabe 1

## 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 2 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.

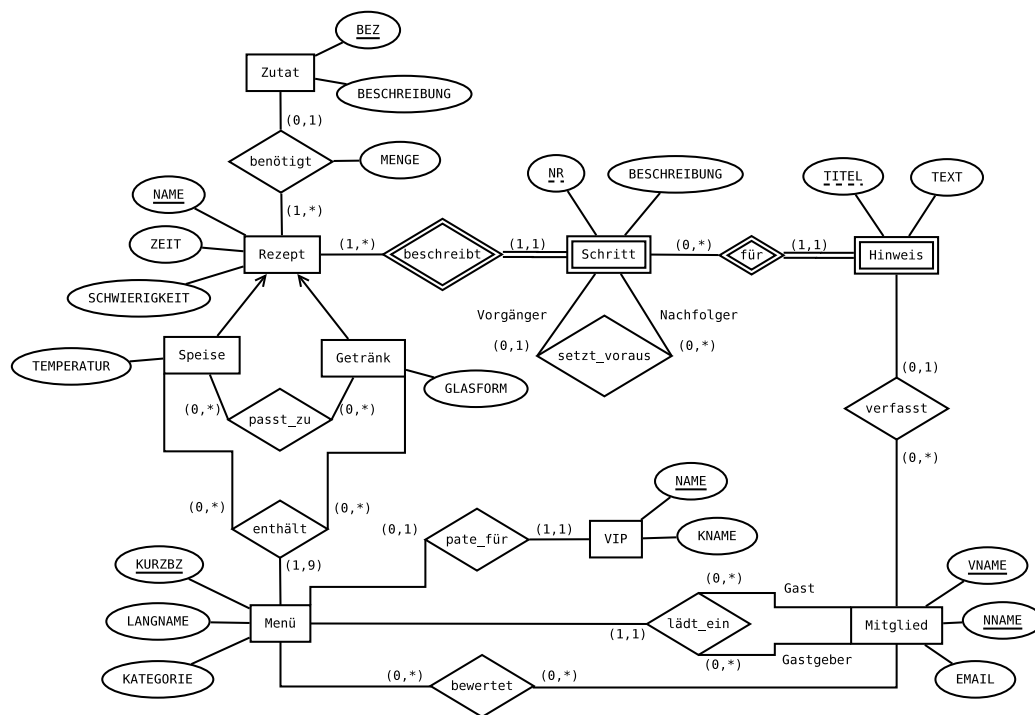


Abbildung 2: EER-Diagramm zu Aufgabe 3

### Lösung:

Rezept (NAME, ZEIT, SCHWIERIGKEIT)  
Zutat (BEZ, BESCHREIBUNG)  
benötigt (Rezept.*NAME*, Zutat.*BEZ*, MENGE)

Schritt(Rezept.NAME, NR, BESCHREIBUNG)  
 Hinweis(Schritt.NAME, Schritt.NR, TITEL, TEXT)  
 setzt\_voraus(Vorgänger: Schritt.NAME, Vorgänger: Schritt.NR,  
                   Nachfolger: Schritt.NAME, Nachfolger: Schritt.NR)  
 Mitglied(VNAME, NNAME, EMAIL)  
 verfasst(Hinweis.NAME, Hinweis.NR, Hinweis.TITEL,  
           Mitglied.VNAME, Mitglied.NNAME)  
 Menü(KURZBZ, LANGNAME, KATEGORIE,  
       Gast: Mitglied.VNAME, Gast: Mitglied.NNAME,  
       Gastgeber: Mitglied.VNAME, Gastgeber: Mitglied.NNAME)  
 bewertet(Mitglied.VNAME, Mitglied.NNAME, Menü.KURZBZ)  
 VIP(NAME, KNAME, Menü.KURZBZ)  
 Speise(Rezept.NAME, TEMPERATUR)  
 Getränk(Rezept.NAME, Glasform)  
 passt\_zu(Speise.NAME, Getränk.NAME)  
 enthält(Speise.NAME, Getränk.NAME, Menü.KURZBZ)

#### Aufgabe 4 (EER und Relationenschema) [0.6 Punkte]

Gegeben ist das EER-Diagramm aus Abbildung 3 (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.

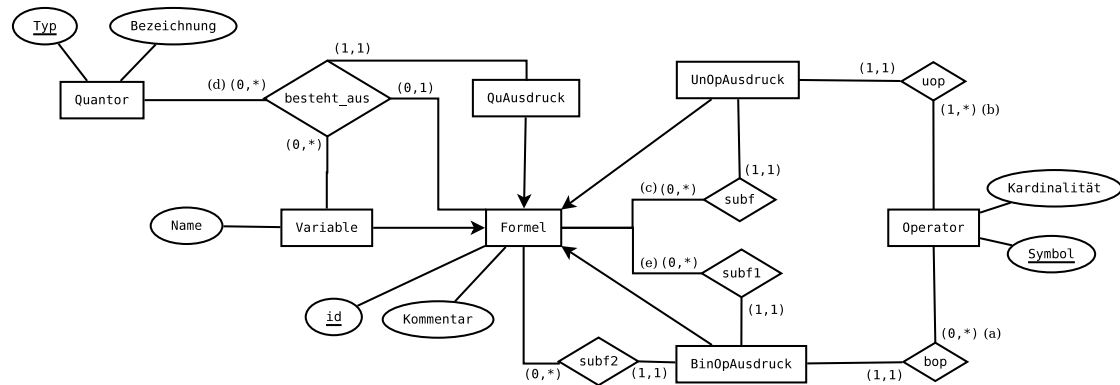


Abbildung 3: EER-Diagramm zu Aufgabe 4

- 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 "subf1" mit einem binären Operatorausdruck steht (= Minimum), 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	
SYMBOL	Kardinalität
$\neg$	1
$\wedge$	2
$\vee$	2
$\subset$	2
$\supset$	2
$\equiv$	2
$\neq$	2
$\uparrow$	2
$\downarrow$	2

Quantor	
TYP	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

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
$f_7$	$\vee$	$f_6$	$f_2$
$f_8$	$\supset$	$f_1$	$f_2$
$f_9$	$\equiv$	$f_7$	$f_8$
$f_{11}$	$\vee$	$f_1$	$f_{10}$
$f_{15}$	$\equiv$	$f_{14}$	$f_{13}$
$f_{17}$	$\supset$	$f_4$	$f_5$
$f_{17}$	$\supset$	$f_5$	$f_6$
$f_{18}$	$\wedge$	$f_{17}$	$f_{16}$
$f_{20}$	$\supset$	$f_{18}$	$f_{19}$

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

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

Lösung:

1. Nein. Die Relation **besteht\_aus** hätte mit der Relation **QuAusdruck** vereinigt werden müssen: Auf Grund der Tatsache, dass laut EER-Diagramm jeder “Quantor-Ausdruck” an nur genau in einer “besteht\_aus” Beziehung teilhaben kann, identifiziert bereits das Feld **Formel.ID** jeden Eintrag in der Relation eindeutig. Daher ist **Formel.ID** der Schlüssel. Da es weiters zu jedem Eintrag in **QuAusdruck** einen Eintrag in **besteht\_aus** geben muss, können die beiden Relationen zusammengefasst werden.
2. Die Ausprägung verletzt die im EER-Diagramm angegebenen Eigenschaften in folgenden Punkten:
  - Die Variable  $f_5$  und der “UnOpAusdruck”  $f_{10}$  verletzen die Generalisierungsbeziehung zum Entitätstyp “Formel”, da sie nicht in der Ausprägung der Relation **Formel** enthalten sind.
  - Der Beziehungstyp **uop** verlangt, dass jeder Operator mit mindestens einem “unären Operatorausdruck” in Beziehung steht. (D.h., dass es für jeden Operator  $op$  mindestens ein Paar  $(op, a)$  gibt, wobei  $a$  ein “unärer Operatorausdruck” ist). Diese Bedingung wird jedoch nur von den Operatoroeren  $\neg$  und  $\subset$  erfüllt. Beachten Sie, dass der Eintrag von  $\subset$  in der Tabelle **UnOpAusdruck** (d.h. die Verwendung des binären Operators als unären Operator) keine im EER-Diagramm spezifizierte Bedingung verletzt.
  - Die Relation **besteht\_aus** verletzt die Kardinalität des “besteht\_aus” Beziehungstyps: Laut EER-Diagramm muss jede Entität vom Typ “QuAusdruck” an genau einer “besteht\_aus” Beziehung teilnehmen. Der “QuAusdruck”  $f_{21}$  ist allerdings nie Teil einer solchen Beziehung.
  - Die Relation **besteht\_aus** verletzt die Kardinalität des “besteht\_aus” Beziehungstyps ein weiteres mal: Laut EER-Diagramm kann jede Entität vom Typ “QuAusdruck” nur an genau einer “besteht\_aus” Beziehung teilnehmen. Der “QuAusdruck”  $f_{14}$  ist allerdings Teil zwei solcher Beziehungen.
  - $f_{17}$  verletzt die Schlüsseleigenschaft des Entitätstyps “BinOpAusdruck”, und somit auch die Schlüsseleigenschaft der Relation **BinOpAusdruck**: Es gibt zwei verschiedene Einträge in **BinOpAusdruck** welche beide den Wert  $f_{17}$  für das Attribut **ID** annehmen.
3. Die Ausprägung erfüllt die folgenden (min,max) Werte:
  - (a): (0, 4)
  - (b): (0, 5)
  - (c): (0, 2)
  - (d): (1, 2)

## 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<sup>A</sup>T<sub>E</sub>X 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 ( $\pi$ ,  $\times$ ,  $\rho$ ,  $-$ ,  $\cup$ ) aus. Drücken Sie den Join/Operator ( $\bowtie$ ) mittels  $\sigma'$  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.

### Lösung:

- (a)  $\sigma_F(S) \equiv S - \sigma'_F(S)$
- (b)  $S \bowtie T \equiv \pi_{S.G_1, \dots, S.G_g, S_1, \dots, S_s, T_1, \dots, T_t}((S \times T) - \sigma'_{S.G_1=T.G_1 \wedge \dots \wedge S.G_g=T.G_g}(S \times T))$
- (c)  $R \cap S \equiv R - (R \circ S)$

### 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  $r_{10} : \pi_D(\sigma_{D < 4 \wedge D > 7}(S \bowtie T))$

### Lösung:

- (a) **Nein:** Durch die Umbenennung haben  $R$  und  $S$  ein gemeinsames Attribut, wodurch der Join im Allgemeinen weniger Tupel liefert.
- (b) **Ja:** Die gemeinsame Attribute von  $R$  und  $U$  sind  $A$  und  $B$ , beide Seiten liefern daher alle Tupel aus  $U$ , welche einen Partner in  $R$  finden.
- (c) **Nein:** Die Selektionsbedingung in  $r_5$  enthält ein ODER, welches auch als Vereinigung aber nicht wie in  $r_6$  als Durchschnitt ausgedrückt werden kann.
- (d) **Ja:** Hier enthält die Selektion ein ODER, welches mittels Vereinigung ausgedrückt werden kann. Darüber hinaus kann eine geschachtelte Selektion wie in  $r_8$  durch ein entsprechendes UND in der Selektionsbedingung wie in  $r_7$  ausgedrückt werden.
- (e) **Ja:** Beide Seiten beschreiben eine Relation mit Schema  $D$  welche keine Tupel enthält. In  $r_9$  wird dies mittels Mengendifferenz erreicht, in  $r_{10}$  mittels einer Selektionsbedingung, die von keinem Tupel erfüllt werden kann.

**Aufgabe 7 (Größenabschätzung) [0.5 Punkte]**

Gegeben sind die Relationen  $R(\underline{A}BCD)$  mit  $x$  Tupeln,  $S(\underline{B}DE)$  mit  $y$  Tupeln und  $T(\underline{A}CD)$  mit  $z$  Tupeln. Weiters sind folgende Ausdrücke in relationaler Algebra gegeben:

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

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  $z = 5$ . **Lösung:** (min 0/max 7)
- (b) Betrachten Sie  $r_2$  unter der Annahme  $x = 7, y = 5, z = 8$ . **Lösung:** (min 8/max 15)
- (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. **Lösung:** (min 2/max 36)
- (d) Betrachten Sie  $r_4$  unter der Annahme  $1 \leq x \leq y \leq z$ . **Lösung:** (min 0/max  $x$ )

**Lösung:**

Aufgabe (a):  $\rho_{F \leftarrow D}(T)$  und  $S$  haben keine Attribute gemeinsam und der Join wird zum Kreuzprodukt mit 35 Tupeln. Von diesen 35 Tupel erfüllen maximal alle und minimal 0 Tupel die Selektionsbedingung  $A < 2$ . An der unteren Schranke 0 ändert sich durch die Projektion nichts. Maximal erhält man allerdings nur 7 Tupel, da es nur 7 unterschiedliche Werte für  $B$  und  $D$  aus  $S$  geben kann.

Aufgabe (b):  $R \bowtie S$  hat minimal 0 Tupel (es gibt keine Joinpartner) und maximal 7 Tupel (jedes Tupel aus  $R$  kann maximal einen Partner aus  $S$  finden, da die Joinattribute Schlüssel in  $S$  sind). Durch die Projektion auf  $A$  ändert sich an den Schranken nichts.

$\pi_A(T)$  hat allerdings exakt 8 Tupel, da  $A$  Schlüssel in  $T$  ist. Durch die Vereinigung von  $\pi_A(R \bowtie S)$  und  $\pi_A(T)$  ergeben sich damit minimal 8 und maximal 15 Tupel.

Aufgabe (c): Per Annahme hat  $\pi_{B,D}(R) \cap \pi_{B,D}(S)$  mindestens 2 Tupel, höchstens sind es 4 Tupel aus  $\pi_{B,D}(R)$ . Der linke äußere Join liefert dann minimal 2 Tupel (beide Tupel der linken Seite finden keine Joinpartner, bleiben aber erhalten) und maximal 36 Tupel (alle vier Tupel der linken Seite können jeweils mit allen Tupel der rechten Seite gejoint werden).

Aufgabe (d):  $R \bowtie T$  hat minimal 0 Tupel (es gibt keine Joinpartner) und maximal findet jedes Tupel aus  $R$  einen Partner und es ergeben sich  $x$  Tupel. Für  $(R \bowtie T) \bowtie S$  bleibt 0 als untere Schranke erhalten. Maximal können es nur  $x$  Tupel werden, da die Joinattribute  $B$  und  $D$  Schlüssel in  $S$  sind, und jedes der  $x$  Tupel der linken Seite maximal einen Partner in  $S$  finden kann.

### 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.

- Gesucht sind die Nachnamen der Vorsitzenden von allen Ausschüssen, in denen ein Politiker, der nach 1990 geboren wurde, teilnimmt.
- 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.
- Betrachten Sie nun den folgenden Ausdruck in relationaler Algebra. Beschreiben Sie in eigenen Worten, welches Ergebnis dieser Ausdruck liefert.

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

### Lösung:

- $\pi_{\text{Nachname}} \left( P \bowtie (A \bowtie \pi_{\text{AID}}(N \bowtie \sigma_{\text{Geburtsjahr} > 1990}(P))) \right)$
- Es gibt viele mögliche Lösungen, zwei davon sind:

$$\pi_{\text{Vorname, Nachname}} \left( \sigma_{\text{Geburtsjahr} > 1990 \vee \text{Geburtsjahr} < 1960} (P) \bowtie \right. \\ \left. \left( \pi_{\text{PNR}} (P) - \left( \pi_{\text{PNR}} (A) \cup \pi_{\text{PNR}} (N \bowtie \pi_{\text{AID}} (\sigma_{\text{Bezeichnung} = \text{'Tabaksteuer'}} (A))) \right) \right) \right) \\ \pi_{\text{Vorname, Nachname}} \left( \sigma_{\text{Geburtsjahr} > 1990 \vee \text{Geburtsjahr} < 1960} (P) \bowtie \right. \\ \left. \left( \left( \pi_{\text{PNR}} (P) - \pi_{\text{PNR}} (A) \right) - \pi_{\text{PNR}} (N \bowtie \pi_{\text{AID}} (\sigma_{\text{Bezeichnung} = \text{'Tabaksteuer'}} (A))) \right) \right)$$

- (c) Gesucht sind die Nachnamen aller Politiker, die an allen Ausschüssen teilnehmen, deren Vorsitzender 1981 geboren wurde.

**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.

**Lösung:**

(a) Tupelkalkül:  $\left\{ [t.\text{Nachname}] \mid t \in P \wedge \exists a \in A (t.\text{PNR} = a.\text{PNR} \wedge \right.$   
 $\left. \exists n \in N (a.\text{AID} = n.\text{AID} \wedge \right.$   
 $\left. \exists p \in P (n.\text{PNR} = p.\text{PNR} \wedge p.\text{Geburtsjahr} > 1990) \right\}$

Domänenkalkül:  $\left\{ [n] \mid \exists p, g, v ([p, v, n, g] \in P \wedge \right.$   
 $\left. \exists a, b ([a, b, p] \in A \wedge \right.$   
 $\left. \exists p', s ([p', a, s] \in N \wedge \right.$   
 $\left. \exists v', n', g' ([p', v', n', g'] \in P \wedge g' > 1990) \right\}$

(b) Tupelkalkül:  $\left\{ [t.\text{Vorname}, t.\text{Nachname}] \mid t \in P \wedge t.\text{Geburtsjahr} > 1990 \vee \right.$   
 $\left. t.\text{Geburtsjahr} < 1960 \wedge \right.$   
 $\left. \neg \exists a \in A (a.\text{PNR} = t.\text{PNR}) \wedge \right.$   
 $\left. \neg \exists n \in N (n.\text{PNR} = t.\text{PNR} \wedge \right.$   
 $\left. \exists a \in A (a.\text{AID} = n.\text{AID} \wedge a.\text{Bezeichnung} = \text{'Tabaksteuer'}) \right\}$

Domänenkalkül:  $\left\{ [v, n] \mid \exists p, g ([p, v, n, g] \in P \wedge g > 1990 \vee g < 1960 \wedge \right.$   
 $\left. \neg \exists a, b ([a, b, p] \in A) \wedge \right.$   
 $\left. \neg \exists a, s ([p, a, s] \in N \wedge \right.$   
 $\left. \exists b, p' ([a, b, p'] \in A \wedge b = \text{'Tabaksteuer'}) \right\}$

## 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)  $\{[t.B] \mid t \in R \wedge \exists s \in S (s.D > 4)\}$

### Lösung:

- (a) **Falsch:** Schlüssel müssen Teilmengenminimal sein, nicht Kardinalitätsminimal. Das heißt, die Minimalität bezieht sich nicht auf die absolute Anzahl an Attributen, sondern nur darauf, dass keine Teilmenge der Attribute ebenfalls noch die Tupel (bzw. Entitäten im Fall von EER Diagrammen) eindeutig identifiziert. Daher kann  $AC$  genauso als Schlüssel gewählt werden wie  $B$ .
- (b) **Falsch:** Anders als die  $(min, max)$  Notation bieten Funktionalitäten nur Informationen über die maximale Häufigkeit, aber nicht über die minimale Häufigkeit. D.h., es kann bei der Notation  $1 : N$  (bzw.  $N : 1$ ,  $1 : 1$ ) nicht unterschieden werden ob jede Entität “genau einmal” oder “maximal einmal” in der Beziehung vorkommt. In der  $(min, max)$  Notation ist dies durch die Angabe von entweder  $(0, 1)$  oder  $(1, 1)$  möglich. Da es im Fall dass jede Entität maximal einmal die Beziehung eingeht (also  $(0, 1)$  in der  $(min, max)$ -Notation) beim Zusammenziehen NULL-Werte entstehen können bieten die beiden Notationen in dieser Hinsicht nicht die gleiche Ausdruckskraft.

(c) Es gilt folgendes:

- (1) **Nein**, es handelt sich hier nicht um eine gültige Anfrage, weder im Tupel- noch im Domänenkalkül:  $t \mid t \in R$  ist die Syntax des Tupelkalküls:  $t$  wird an die Tupel der Relation  $R$  gebunden. Da  $t$  Tupel als Werte annimmt definiert die Abfrage auch richtigerweise eine Menge von Tupeln. Der Teil  $\exists s, u([s, u] \in S$  hingegen ist in der Syntax des Domänenkalküls verfasst. Hier werden Variablen an einzelne Domänenelemente gebunden. Da die Variablen nun einzelne Werte und keine Tupel mehr darstellen müssen sie mittels Tupelkonstruktor  $[ ]$  zu Tupeln gruppiert werden damit ausgedrückt werden kann, dass es ein entsprechendes Tupel in  $S$  geben soll. Der letzte Teil  $t.A = s$  ist wiederum eine Mischung aus Tupel- und Domänenkalkül:  $t.A$  ist die Schreibweise des Tupelkalküls um auf den Wert des Attributs  $A$  in dem Tupel auf welches  $t$  gerade gebunden ist zuzugreifen, während man im Domänenkalkül auf einzelne Werte sofort Zugriff hat, da die Variablen ja direkt daran gebunden werden.
- (2) **Ja**, es handelt sich um eine gültige Abfrage im Tupelkalkül: Die Variablen werden an die Tupel der Relationen gebunden.