

# 1. Übungsblatt (WS 2018) – Musterlösung

3.0 VU Datenmodellierung / 6.0 VU Datenbanksysteme

## Informationen zum Übungsblatt

### Allgemeines

In diesem Übungsteil üben Sie den Entwurf einer kleinen Datenbank mittels EER-Diagrammen, das Überführen eines EER-Diagramms in ein Relationenschema, sowie den Umgang mit der relationalen Algebra und dem Relationenkalkül.

Lösen Sie die Beispiele **eigenständig** (auch bei der Prüfung und vermutlich auch in der Praxis sind Sie auf sich alleine gestellt)! Wir weisen Sie darauf hin, dass sämtliche abgeschriebene Lösungen mit 0 Punkten beurteilt werden (sowohl das “Original” als auch die “Kopie”).

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

Das Übungsblatt enthält 7 Aufgaben, auf welche Sie insgesamt 15 Punkte erhalten können.

### Deadlines

**bis 29.10. 12:00Uhr** Upload der Abgabe über TUWEL  
ab 12.11. 13:00Uhr Korrektur und Feedback in TUWEL verfügbar

### Tutorensprechstunden (freiwillig)

Rund eine Woche vor der Abgabedeadline bieten die TutorInnen Sprechstunden an. Falls Sie Probleme mit oder Fragen zum Stoff des Übungsblattes haben, es Verständnisprobleme mit den Beispielen oder technische Fragen gibt, kommen Sie bitte einfach vorbei. Die TutorInnen beantworten Ihnen gerne Ihre Fragen zum Stoff, oder helfen Ihnen bei Problemen weiter.

Ziel der Sprechstunden ist es, Ihnen beim **Verständnis des Stoffs** zu helfen, nicht, das Übungsblatt für Sie zu rechnen, oder die eigenen Lösungen vorab korrigiert zu bekommen.

Die Teilnahme ist vollkommen freiwillig — Termine und Orte der Tutorensprechstunden finden Sie in TUWEL.

### Durchsprache der Übungsbeispiel (freiwillig)

In den Tagen nach Rückgabe der korrigierten Abgaben gibt es die Möglichkeit die Übungsbeispiele in kleineren Gruppen (max. 25 Personen) durchzusprechen. Jede dieser Gruppen wird von einer Assistentin/einem Assistenten geleitet. Der genaue Ablauf in einer Übungsgruppe kann variieren, und hängt auch von Ihren Wünschen und Fragen ab. Die grundsätzliche Idee ist es, die Beispiele durchzurechnen, und speziell auf Ihre Fragen und mögliche Unklarheiten einzugehen. Die (relativ) kleine Gruppengröße soll eine aktive Teilnahme ermöglichen. Daher ist es auch wichtig, dass Sie sich bereits im Vorfeld mit Ihrer korrigierten Abgabe auseinandersetzen, und Unklarheiten identifizieren. Trauen Sie sich, entsprechend Fragen zu stellen – keine Frage kann irgendeinen (negativen) Einfluss auf Ihre Note haben.

Die Teilnahme an so einer Gruppe ist absolut freiwillig. Um die Gruppengröße klein zu halten ist eine Anmeldung in TUWEL erforderlich. Termine und Orte finden Sie in TUWEL.

### Weitere Fragen – TUWEL Forum

Sie können darüber hinaus das TUWEL Forum verwenden, sollten Sie inhaltliche oder organisatorische Fragen haben.

## Aufgaben: EER-Diagramme

### Aufgabe 1 (EER-Diagramm erstellen) [3 Punkte]

Eine Webseite für den Verkauf von Gebrauchtwagen und Neuwagen benötigt eine Datenbank für die Nutzerverwaltung und Speicherung der verfügbaren Wagen. Zeichnen Sie aufgrund der vorliegenden Informationen ein EER-Diagramm. Verwenden Sie dabei die (min,max)-Notation. Es sind keine NULL-Werte erlaubt, und Redundanzen sollen vermieden werden.

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 User der Webseite werden mit einer eindeutigen Identifikationsnummer (ID) identifiziert. Zusätzlich wird ein Name (NAME) gespeichert. KäuferInnen und VerkäuferInnen sind verschiedenen Arten von Usern. Autos werden von VerkäuferInnen online gestellt, wobei jede/r VerkäuferIn beliebig viele Autos online stellen kann. Autos werden eindeutig identifiziert durch eine ID (ID) und den/die Verkäufer/in der/die es online gestellt hat. Jedes Auto besitzt außerdem einen Preis und eine Beschreibung. Es gibt drei unterschiedliche Typen von Autos: Gebrauchtwagen, Unfallauto und Neuwagen. Für Gebrauchtwagen werden zusätzlich das Jahr der Erstzulassung (EZJ), die Anzahl der Voreigentümer (ANZ\_EIG) und die gefahrenen Kilometer (KILOMETER) gespeichert. Bei Unfallautos wird ein Schadensbericht (SCHADENSBERICHT) gespeichert.

Autos haben eine bestimmte Marke und ein Modell. Marken werden durch den eindeutigen Namen (NAME) identifiziert. Von jeder Marke gibt es mehrere Modelle. Modelle werden durch die Marke, zu der sie gehören und durch die Bezeichnung (BEZEICHNUNG) identifiziert. Für jedes Modell wird außerdem der Treibstoff (TREIBSTOFF), der Verbrauch (VERBRAUCH) und die Power (PS/KW) gespeichert. In der Datenbank wird außerdem erfasst, welche Modelle Nachfolger anderer Modelle sind, wobei jedes Modell beliebig viele Nachfolger haben kann und auch Nachfolger von beliebig vielen anderen Modellen sein kann. Jedes Modell kann außerdem bis zu 12 Ausstattungspakete enthalten. Ausstattungspakete werden durch den Typ (TYP) eindeutig identifiziert. Zusätzlich wird eine Beschreibung (BESCHREIBUNG) in der Datenbank gespeichert.

Zu jedem Auto wird ein (möglicherweise auch mehr) Foto gespeichert. Fotos werden durch eine Nummer (NR) und durch das Auto, zu dem sie gehören, identifiziert.

Wenn ein Käufer ein Auto von einem Verkäufer kauft werden Datum (DATUM) und Preis (PREIS) gespeichert. Jedes Auto kann nur einmal verkauft werden, aber Käufer und Verkäufer können mehrere Autos kaufen bzw. verkaufen.

**Lösung:** Siehe Abbildung 1.

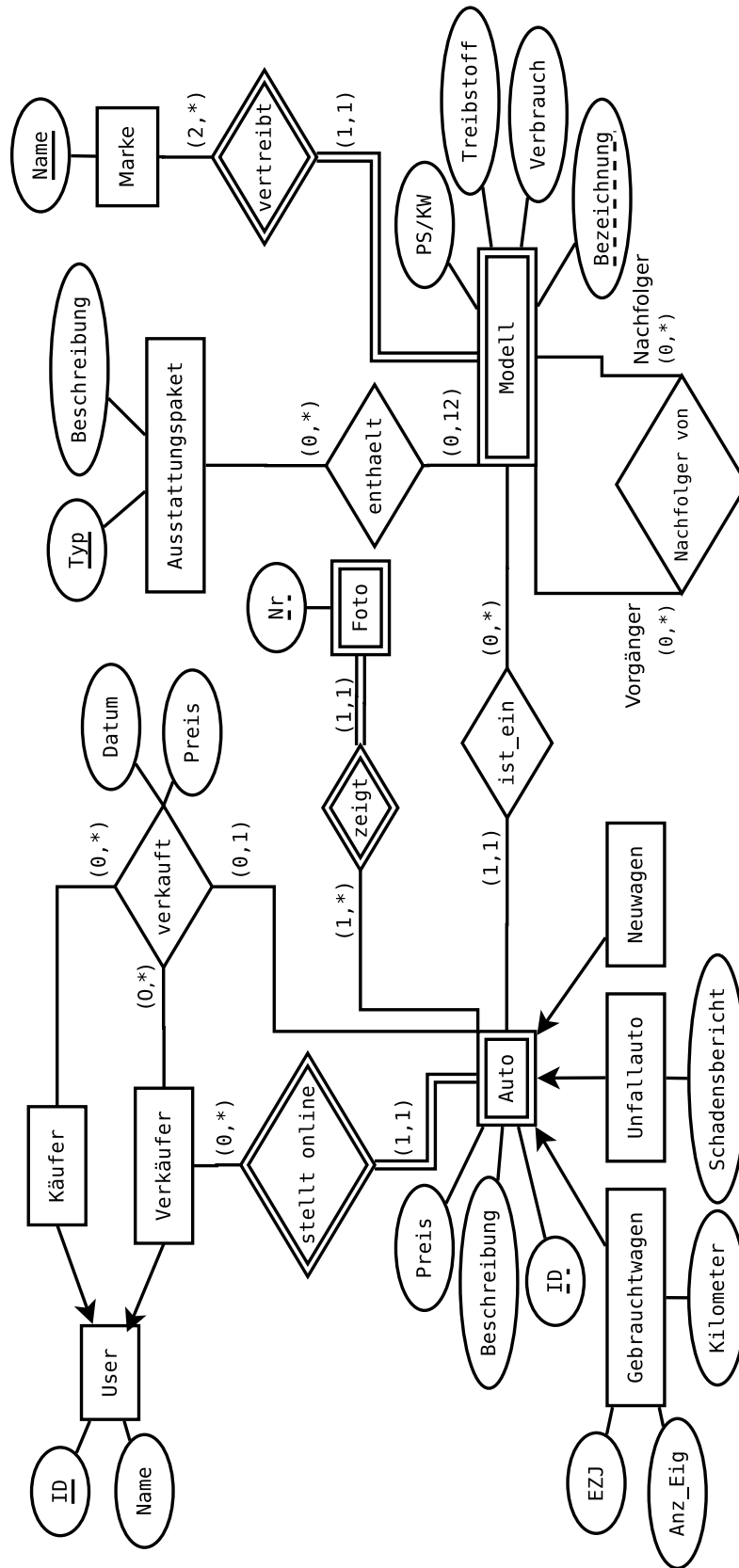


Abbildung 1: Lösung von Aufgabe 1

**Aufgabe 2 (Semantik von EER Diagrammen)** [1 Punkte]

Betrachten Sie das in Abbildung 2 dargestellte ER-Diagramm, welches (fiktive) Richtlinien zur Schichtplanung- und Einteilung beschreibt.

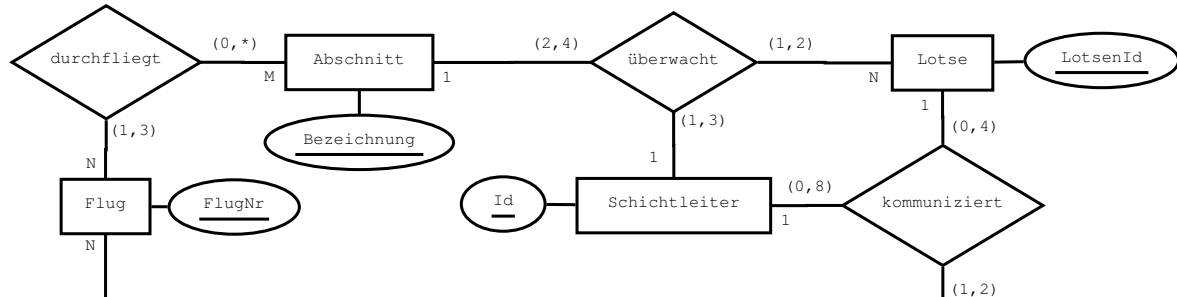


Abbildung 2: EER-Diagramm zu Aufgabe 2

1. In dem ER-Diagramm wird sowohl die Notation mittels Funktionalitäten, als auch die (min,max)-Notation verwendet (*Anmerkung*: dies geschieht hier zu Übungszwecken und ist in der Praxis nicht üblich).

Das Diagramm enthält dadurch mehr Informationen als wenn nur eine der beiden Notationen verwendet worden wäre.

- Geben Sie einen konkreten Beziehungstyp im Diagramm an, bei welchem das Weglassen einer Notation zum Verlust von Informationen führt.
- Für den gewählten Beziehungstyp, das Weglassen welcher Notation führt zum Informationsverlust?
- Erklären Sie kurz in eigenen Worten, welche Information nicht mehr dargestellt werden kann.
- Geben Sie ein konkretes Beispiel für die verlorene Information an. D.h., für den von Ihnen gewählten Beziehungstyp, geben Sie eine Ausprägung an welche (mindestens) eine durch die weggelassene Notation ausgedrückte Bedingung verletzt, aber sämtliche Beschränkungen der verbleibenden Notation erfüllt.

**Lösung:**

Ja, beide Notationen enthalten teilweise Informationen, welche durch die jeweils andere Notation nicht ausgedrückt werden können. Wir geben im folgenden jeweils nur ein Beispiel für jede Notation an - es gibt noch mehr:

- Lässt man beim Beziehungstyp **überwacht** die *Funktionalitäten* weg, so kann durch die (min,max)-Notation weder ausgedrückt werden, dass jedes Paar aus Lotse und Schichtleiter nur einen einzigen Abschnitt überwachen darf, noch dass es für jeden Lotsen und jeden von ihr/ihm überwachten Abschnitt jeweils nur einen Schichtleiter geben darf.

Ein konkretes Gegenbeispiel wären Lotsen L1, Schichtleiter S1, S2, und Abschnitte A1, A2. Dann wäre es nach der (min,max)-Notation eine Einteilung zulässig, bei der Lotse L1 gemeinsam mit S1 Abschnitt A1 überwacht, und außerdem mit S2 den Abschnitt A1 überwacht (d.h. die Ausprägung von **überwacht** enthält sowohl das Tripel (L1, S1, A1) als auch das Tripel (L1, S2, A1). Ähnlich verhält es sich mit den

Abschnitten. Die (min,max)-Notation würde erlauben dass L1 und S1 gemeinsam sowohl den Abschnitt A1 als auch den Abschnitt A2 überwachen. All dies ist durch die Funktionalitäten verboten.

- Lässt man beim Beziehungstyp **kommuniziert** die (min,max)-Notation weg, so wird durch die Angabe der Funktionalitäten weder ausgedrückt, dass jeder Flug mindestens einmal kommunizieren muss, noch dass jeder Flug mit maximal zwei verschiedenen Kombinationen aus Lotse und Schichtleiter kommunizieren darf. Ebenso wird nicht mehr ausgedrückt dass jeder Lotse mit maximal vier Flügen kommunizieren darf.

Ein konkretes Gegenbeispiel wäre Lotse L1, Schichtleiter S1, sowie Flüge F1, F2, F3, F4 und F5. Dann wäre es auf Grund der Funktionalitäten erlaubt dass F1 mit keinem Schichtleiter und Lotsen in einer **kommuniziert** Verbindung steht (die konkrete Ausprägung des Beziehungstyps könnte zum Beispiel leer sein). Genauso wäre es ok wenn Lotse L1 mit allen 5 Flügen (und pro Flug mit jeweils einem Schichtleiter) in einer **kommuniziert** Verbindung stehen würde. D.h. die Ausprägung von **kommuniziert** könnte die Einträge (L1, S1, F1), (L1, S1, F2), (L1, S1, F3), (L1, S1, F4) und (L1, S1, F5) enthalten. Beide Fälle sind durch die (min, max)-Notation verboten.

2. Nehmen Sie an, Sie erhalten den weiter unten dargestellten Schichtplan vorgelegt um den ordnungsgemäßen Ablauf der Schicht zu überprüfen.

Beschreiben Sie mindestens 8 Verletzungen der im ER-Diagramm dargestellten Vorschriften durch die Protokolle. (Hinweis: Beschränken Sie sich ausschließlich auf Verletzungen von im ER-Diagramm beschriebenen Sachverhalten. Ignorieren Sie insbesondere Sachverhalte die zwar “unlogisch” erscheinen, aber nicht im ER-Diagramm spezifiziert wurden.)

**Lotsen:** L1, L2, L3, L4, L5, L6, L7

**Schichtleiter:** S1, S2, S3, S4, S5, S6, S7

**Abschnitt:** A1, A2, A3, A4, A5

**Flug:** FLN1, FLN4, FLN16, FLN64, FLN256, FLN1048, FLN1701

A1	A2	A3	A4	A5
-----	-----	-----	-----	-----
L3 (S1)	L1 (S2)	L2 (S2)	L4 (S4)	L4 (S4)
L4 (S1)	L3 (S3)	L5 (S5)	L2 (S2)	L6 (S5)
L5 (S3)	L1 (S3)		L7 (S1)	L7 (S6)
	L6 (S7)			L2 (S3)
				L5 (S7)

Flugpläne	Kommunikation
-----	-----
FLN1: A3, A2	L1 <--> FLN 1701 (S1)
FLN4: A1, A2, A5	L1 <--> FLN 4 (S2)
FLN16: A2, A3, A4, A5	L1 <--> FLN 64 (S2)
FLN64: A1	L2 <--> FLN 64 (S3)
FLN256: A3, A2	L2 <--> FLN 1048 (S5)
FLN1048: A4, A3, A1	L4 <--> FLN 64 (S6)
FLN1701: A1, A3, A5	L4 <--> FLN 16 (S2)
	L4 <--> FLN 1 (S1)
	L4 <--> FLN 1 (S3)
	L4 <--> FLN 1701 (S5)
	L5 <--> FLN 4 (S2)
	L6 <--> FLN 1048 (S6)
	L7 <--> FLN 1048 (S6)
	L7 <--> FLN 16 (S5)

(Der Schichtplan ist wie folgt zu lesen: Zuerst werden jeweils die Menge an Entitäten der Typen **Lotse**, **Schichtleiter**, **Abschnitt** und **Flug** angegeben, wobei die Bezeichner jeweils die Werte der Schlüsselattribute darstellen.

Die "Zettel" mit  $A_i$  ( $1 \leq i \leq 5$ ) bezeichnen die Ausprägung der Relation **überwacht**:  $A_i$  gibt die Bezeichnung des jeweiligen Abschnitts an, der Wert am Beginn jeder Zeile die LotsenId, und der Wert in der Klammer die Schichtleiter-Id. Z.B. entspricht die Zeile L3 (S1) für Abschnitt A1 einem Eintrag (A1, L3, S1) in der Relation **überwacht**(**Abschnitt**, **Lotse**, **Schichtleiter**). Der "Zettel" **Flugpläne** gibt an, welche Abschnitte welcher Flug durchfliegt (stellt also die Ausprägung des Beziehungstyps **durchfliegt** dar). Am "Zettel" **Kommunikation** ist die Ausprägung des Beziehungstyps **kommuniziert** dargestellt, wobei jede Zeile  $L_i \text{ <-> FLN } x \text{ (S}_j\text{)}$  für ein Tripel ( $L_i$ ,  $FLN \ x$ ,  $S_j$ ) in der Ausprägung von **kommuniziert**(**Lotse**, **Flug**, **Schichtleiter**) steht.)

### Lösung:

Es finden sich die folgenden 16 Verletzungen:

- Lotse/Lotsin  $L2$  überwacht 3 Abschnitte, wobei maximal zwei erlaubt sind.
- Lotse/Lotsin  $L4$  überwacht 3 Abschnitte, wobei maximal zwei erlaubt sind.
- Lotse/Lotsin  $L5$  überwacht 3 Abschnitte, wobei maximal zwei erlaubt sind.
- SchichtleiterIn  $S3$  überwacht 4 Abschnitte, wobei maximal drei erlaubt sind.
- Abschnitt  $A5$  wird von 5 Lotsen (plus dazugehörigen Schichtleitern) überwacht, wobei maximal vier erlaubt sind.
- Lotse/Lotsin  $L1$  überwacht den Abschnitt  $A2$  mit zwei verschiedenen Schichtleitern. Dies ist jedoch nicht erlaubt, da Schichtleiter die Funktionalität 1 besitzt, d.h. zu jedem Paar (Abschnitt, Lotse) darf es maximal einen Schichtleiter geben.
- Das (Lotse, Schichtleiter)-Paar ( $L2, S2$ ) überwacht 2 verschiedene Abschnitte ( $A3$  und  $A4$ ). Nachdem die Funktionalität des Entitätstyps "Abschnitt" jedoch 1 ist, darf jede solche Paarung maximal einen Abschnitt überwachen.
- Das (Lotse, Schichtleiter)-Paar ( $L4, S4$ ) überwacht 2 verschiedene Abschnitte ( $A3$  und  $A4$ ). Nachdem die Funktionalität des Entitätstyps "Abschnitt" jedoch 1 ist, darf jede solche Paarung maximal einen Abschnitt überwachen.

- Flug *FLN16* durchfliegt 4 Abschnitte (es sind jedoch max. drei pro Flug erlaubt).
- Lotse/Lotsin *L4* steht mit 5 verschiedenen Paaren aus Flügen und Schichtleitern in einer **kommuniziert** Relation (es sind jedoch maximal vier pro Lotse/Lotsin erlaubt).
- Das (Flug, Lotsin)-Paar (*FLN1, L4*) hat mit 2 verschiedenen SchichtleiterInnen kommuniziert (*S1* und *S3*; auf Grund der Funktionalität 1 des Entitätstyps **Schichtleiter** darf jedes solches Paar mit maximal einer Lotsin/einem Lotsen kommunizieren).
- Das (Flug, Schichtleiter)-Paar (*FLN4, S2*) hat mit 2 verschiedenen Lotsen/Lotsinnen kommuniziert (*L1* und *L5*; auf Grund der Funktionalität 1 des Entitätstyps **Lotse** darf jedes solches Paar mit maximal einer Lotsin/einem Lotsen kommunizieren).
- Das (Flug, Schichtleiter)-Paar (*FLN1048, S6*) hat mit 2 verschiedenen Lotsen/Lotsinnen kommuniziert (*L6* und *L7*; auf Grund der Funktionalität 1 des Entitätstyps **Lotse** darf jedes solches Paar mit maximal einer Lotsin/einem Lotsen kommunizieren).
- Flug *FLN64* kommt in 3 **kommuniziert** Relationen vor (maximal zwei sind erlaubt).
- Flug *FLN256* kommt in 0 **kommuniziert** Relationen vor (jeder Flug muss aber mindestens einmal kommunizieren).
- Flug *FLN1048* kommt in 3 **kommuniziert** Relationen vor (maximal zwei sind erlaubt).

### Aufgabe 3 (Überführung ins Relationenschema) [2 Punkte]

Überführen Sie das EER-Diagramm aus Abbildung 3 in ein Relationenschema. Nullwerte sind nicht erlaubt (Sie können dabei annehmen, dass alle für einen Entitätstyp angegebenen Attribute für alle Entitäten dieses Typs existieren; d.h. die Definiertheit sämtlicher Attribute ist 100%). 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.

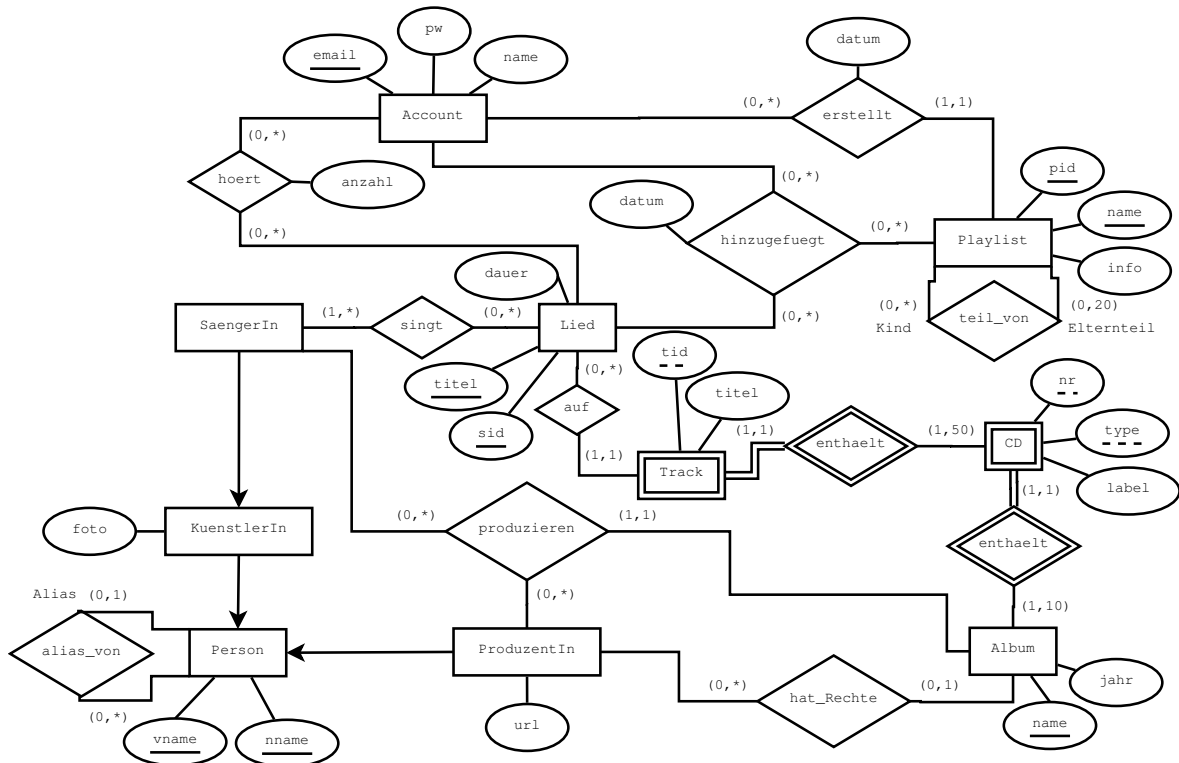


Abbildung 3: EER-Diagramm zu Aufgabe 3

**Lösung:**

Lied (titel, sid, dauer)  
 Account (email, pw, name)  
 Playlist (pid, name, info, email: *Account.email*, datum, )  
 Person (vname, nname)  
 KuenstlerIn (vname: *Person.vname*, nname: *Person.nname*, foto)  
 SaengerIn (vname: *KuenstlerIn.vname*, nname: *KuenstlerIn.nname*)  
 ProduzentIn (vname: *Person.vname*, nname: *Person.nname*, url)  
 Album (name, jahr,  
     ProdVName: *ProduzentIn.vname*, ProdNName: *ProduzentIn.nname*,  
     SaengVName: *SaengerIn.vname*, SaengNName: *SaengerIn.nname*)  
 CD (name: *Album.name*, nr, type, label)  
 Track (name: *CD.name*, nr: *CD.nr*, type: *CD.type*, tid, titel,  
     LiedTitel: *Lied.titel*, sid: *Lied.sid*)  
 hoert (sid: *Lied.sid*, titel: *Lied.titel*, email: *Account.email*, anzahl)  
 hinzugefuegt (sid: *Lied.sid*, titel: *Lied.titel*, email: *Account.email*,  
     pid: *Playlist.pid*, name: *Playlist.name*, datum)  
 teil\_von (ElternPid: *Playlist.pid*, ElternName: *Playlist.name*,  
     KindPid: *Playlist.pid*, KindName: *Playlist.name*)  
 singt (vname: *SaengerIn.vname*, nname: *SaengerIn.nname*,  
     titel: *Lied.titel*, sid: *Lied.sid*)  
 hat\_Rechte (vname: *ProduzentIn.vname*, nname: *ProduzentIn.nname*,  
     name: *Album.name*)  
 alias\_von (AliasVName: *Person.vname*, AliasNName: *Person.nname*,



vname: *Person.vname*, nname: *Person.nname*)

## Aufgaben: 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  $\LaTeX$  Befehle vermerkt.

### Aufgabe 4 (Äquivalenzen) [2 Punkte]

Gegeben sind folgende Paare an Ausdrücken der relationalen Algebra über den Relationenschemata  $R(ABC)$ ,  $S(BCD)$  und  $T(ADF)$ .

- Überprüfen Sie, ob die jeweiligen Ausdrücke äquivalent sind (also ob Sie über allen möglichen Ausprägungen der Schemata immer das gleiche Ergebnis liefern). Sie können dabei davon ausgehen dass NULL- Werte in den Ausprägungen verboten sind.
  - Begründen Sie Ihre Antwort mit einer kurzen **Erklärung**.
  - Falls die beiden Ausdrücke *nicht* äquivalent sind, geben Sie zusätzlich noch ein **Gegenbeispiel** an. (Ein Gegenbeispiel besteht aus konkreten Ausprägungen der beteiligten Relationenschemata sowie den Ergebnissen beider Ausdrücke über diesen Ausprägungen.) Das Gegenbeispiel kann entfallen wenn einer der beiden Ausdrücke kein gültiger Ausdruck der Relationalen Algebra ist. In diesem Fall reicht die Erklärung aus.
- (a)  $q_1: \pi_{BC}(R) - \pi_{BC}(S \bowtie T)$  und  
 $q_2: \pi_{BC}(R) - \pi_{BC}(S \bowtie (\pi_D(S) \cap \pi_D(T)))$
- (b)  $q_3: \pi_B(R) \cap \pi_B(\sigma_{B>3}(S))$  und  
 $q_4: \pi_B(R) \cap (\pi_B(S) - \sigma_{B\leq 3}(S))$
- (c)  $q_5: R \bowtie_{R.B=S.C \wedge R.C=S.D} (S \bowtie (\pi_D(\sigma_{B>7}(S)) \cap \pi_D(T)))$  und  
 $q_6: R \bowtie_{R.B=S.C \wedge R.C=S.D \wedge S.B>7} (S \bowtie (\pi_D(S) \bowtie T))$
- (d)  $q_7: R \div \pi_{BC}(S)$  und  
 $q_8: \pi_A((\pi_A(R) \times \pi_{BC}(S)) - R)$

### Lösung:

#### Aufgabe (a)

**Nein**,  $q_1$  und  $q_2$  sind nicht äquivalent.

Die beiden Abfragen unterscheiden sich darin, welche Tupel aus  $\pi_{BC}(R)$  entfernt werden. In  $q_1$  werden jene Tupel entfernt so dass es ein Tupel in **S** gibt welche die selben Werte für die Attribute *B* und *C* hat, und zusätzlich muss dieses Tupel in **S** einen Wert für das Attribut *D* besitzen, welcher ebenfalls in irgendeinem Tupel in **T** für das Attribut *D* vorhanden ist (auf Grund des Semijoins  $S \bowtie T$ ).

In  $q_2$  reicht es hingegen aus dass es ein Tupel in **S** mit den selben Werten auf *B* und *C* gibt, um einen Eintrag aus  $\pi_{BC}(R)$  zu entfernen. Die zusätzliche Bedingung, dass dieses Tupel in **S**

auch einen Joinpartner in  $T$  haben muss entfällt. Zwar wird im “rechten Teil” des Left-Outer-Joins  $\bowtie$  genau diese Bedingung überprüft, auf Grund der Verwendung des Left-Outer-Joins ist der Ausdruck

$$\pi_{BC}(S \bowtie (\pi_D(S) \cap \pi_D(T)))$$

äquivalent zu dem Ausdruck

$$\pi_{BC}(S),$$

unabhängig davon ob für den Wert von  $D$  einen entsprechenden Wert in  $\pi_D(S) \cap \pi_D(T)$  gibt oder nicht. Nach der Projektion  $\pi_{BC}$  bleiben dann genau diese Paare für  $BC$  erhalten.

Das heißt, in  $q_2$  werden potentiell mehr Tupel von  $\pi_{BC}(R)$  entfernt als in  $q_1$ .

### Gegenbeispiel

R		
A	B	C
1	1	1

S		
B	C	D
1	1	1

T		
A	D	F
2	2	2

q <sub>1</sub>	
B	C
1	1

q <sub>2</sub>	
B	C
1	1

#### Aufgabe (b)

**Nein**,  $q_3$  und  $q_4$  **sind nicht** äquivalent.

$q_3$  ist eine gültige Anfrage der Relationalen Algebra, während  $q_4$  dies nicht ist. Im Teilausdruck  $\pi_B(S) - \sigma_{B \leq 3}(S)$  unterscheiden sich die Schemata von  $\pi_B(S)$  (besteht nur aus dem Attribut  $B$ ) und  $\sigma_{B \leq 3}(S)$  (enthält  $BCD$ ). Dies ist bei der Mengendifferenz nicht erlaubt.

#### Aufgabe (c)

**Nein**,  $q_5$  und  $q_6$  **sind nicht** äquivalent.

Beide Anfragen berechnen jeweils einen Join zwischen  $R$  und einer Teilmenge der Attribute in  $S$ , jedoch unterscheiden sie sich dabei welche Tupel von  $S$  für den Join in Betracht gezogen werden. In beiden Fällen (bei  $q_5$  und  $q_6$ ) werden zum Einen nur Tupel genommen, für welche es ein Tupel mit dem selben Wert auf  $D$  in  $T$  gibt. Dies ist zwar in beiden Abfragen unterschiedlich umgesetzt (in  $q_5$  mittels dem Mengendurchschnitt  $\cap$ , in  $q_6$  mit Hilfe des natürlichen Verbunds  $\bowtie$ ), hat aber die selbe Auswirkung.

Die beiden Anfragen unterscheiden sich in der zweiten Eigenschaft, welche Tupel in  $S$  erfüllen müssen. Bei  $q_6$  werden auf Grund der zusätzlichen Join-Bedingung  $S.B > 7$  nur Tupel von  $S$  zugelassen welche für das Attribut  $B$  einen Wert größer als 7 haben. Bei  $q_5$  hingegen werden alle Tupel von  $S$  akzeptiert, bei welchen das Attribut  $D$  einen Wert annimmt, so dass es *irgendein* Tupel in  $S$  gibt, in dem dieser Wert für  $D$  gemeinsam mit einem Wert größer als 7 für  $B$  gemeinsam auftritt (es muss aber nicht notwendigerweise genau dieses Tupel sein).

Dies hat folgenden Grund: in  $\sigma_{B > 7}(S)$  werden zuerst einmal alle Tupel in  $S$  ausgewählt welche für  $B$  einen Wert größer als 7 haben. Mittels  $\pi_D(\sigma_{B > 7}(S))$  wird von diesen Tupeln jedoch nur der Wert von  $D$  übernommen. Nachdem dann (mittels  $\cap \pi_D(T)$ ) nur jene Werte für  $D$  behalten werden welche auch in  $T$  vorkommen, bleiben bei  $S \bowtie \dots$  alle Tupel erhalten, welche einen dieser Werte für  $D$  annehmen – unabhängig davon welchen Wert  $B$  hat.

Die Abfrage  $q_5$  enthält dadurch potentiell mehr Ergebnisse als  $q_6$

### Gegenbeispiel

R		
A	B	C
1	1	1
2	2	1

S		
B	C	D
8	1	1
2	2	1

T		
A	D	F
1	1	1

q <sub>5</sub>					
R.A	R.B	R.C	S.B	S.C	S.D
1	1	1	8	1	1
2	2	1	2	2	1

q <sub>6</sub>					
R.A	R.B	R.C	S.B	S.C	S.D
1	1	1	8	1	1

**Aufgabe (d)**

**Nein, q<sub>7</sub> und q<sub>8</sub> sind nicht äquivalent.**

Mit q<sub>7</sub> wird die relationale Division berechnet. D.h. das Ergebnis enthält alle Werte welche, welche mit *sämtlichen* Werten in π<sub>BC</sub>(S) jeweils ein gemeinsames Tupel in R bilden.

Die Abfrage q<sub>8</sub> berechnet in gewisser Weise genau das Komplement dieser Werte. Die Abfrage gibt all jene Werte in π<sub>A</sub>(R) aus (also alle Werte welche das Attribut A in irgendeinem Tupel von R annimmt), so dass es zumindest ein Tupel in π<sub>BC</sub>(S) gibt welches gemeinsam mit dem Wert für A kein Tupel in R darstellt: Das Kreuzprodukt π<sub>A</sub>(R) × π<sub>BC</sub>(S) berechnet alle möglichen Kombinationen aus Tupeln in π<sub>BC</sub>(S) und “A-Werten” in R. Von dieser Menge an Tupeln werden dann all jene Tupel entfernt welche in R vorkommen (mittels − R). D.h. übrig bleiben alle Kombinationen von “A-Werten” in R und Tupeln in π<sub>BC</sub>(S) welche gemeinsam kein Tupel in R bilden. Von diesen wird am Ende nun auf die “A-Werte projiziert”.

Um einen äquivalenten Ausdruck zu q<sub>7</sub> zu erhalten, müsste man das Ergebnis von q<sub>8</sub> noch einmal von π<sub>A</sub>(R) abziehen, da q<sub>8</sub> eben genau jene Werte von π<sub>A</sub>(R) enthält welche man *nicht* erhalten möchte.

**Gegenbeispiel**

R		
A	B	C
1	1	1
2	2	2

S		
B	C	D
1	1	1

q <sub>7</sub>	q <sub>8</sub>
A	A
1	2

**Aufgabe 5 (Primitive Operatoren) [1 Punkte]**

Gegeben ist der Ausdruck

$$\pi_{ABC}(S \bowtie (R \bowtie S)) \cap \pi_{ABC}(R)$$

der Relationalen Algebra über den Relationenschemata R(ABCDE) und S(EFGABC).

Geben Sie einen *äquivalenten* Ausdruck der Relationalen Algebra an welcher nur *primitive Operatoren* verwendet, d.h. nur die Operatoren σ, π, ∪, −, × und ρ.

**Lösung:**

Es gibt zwei Möglichkeiten diese Aufgabe anzugehen: Wenn man erkennt dass die Abfrage äquivalent ist zu π<sub>ABC</sub>(S × R) kann man diesen Ausdruck mit Hilfe der primitiven Operatoren ausdrücken. Die andere Möglichkeit ist jeden Operator in π<sub>ABC</sub>(S × (R × S)) ∩ π<sub>ABC</sub>(R) durch die äquivalente Entsprechung in primitiven Operatoren ersetzen.

Wir beginnen mit der ersten Variante, und drücken π<sub>ABC</sub>(S × R) mit Hilfe primitiver Operatoren aus. Wir beginnen mit S × R, und wir verwenden die Explizite Umbenennung zur besseren Lesbarkeit. Wir erhalten

$$S \times R \equiv \pi_{EFGABC}(\sigma_{A=RA \wedge B=RB \wedge C=RC \wedge E=RE}(S \times \rho_{RA \leftarrow A, RB \leftarrow B, RC \leftarrow C, RE \leftarrow E}(R)))$$

Darauf muss noch π<sub>ABC</sub> angewandt werden, wobei wir auf eine doppelte Projektion verzichten und gleich die bereits vorhandene Projektion entsprechend einschränken. Somit ergibt sich der Ausdruck

$$\pi_{ABC}(\sigma_{A=RA \wedge B=RB \wedge C=RC \wedge E=RE}(S \times \rho_{RA \leftarrow A, RB \leftarrow B, RC \leftarrow C, RE \leftarrow E}(R)))$$

Ohne zu erkennen, dass es eine Vereinfachung des Ausdrucks gibt ist die Vorgehensweise ident, dauert jedoch etwas länger:

Wir beginnen wieder “innen” und arbeiten uns nach “außen” vor. Für  $R \bowtie S$  erhalten wir (wieder mittels explizitier Umbenennung)

$$\pi_{ABCDEFG}(\sigma_{RA=A \wedge RB=B \wedge RC=C \wedge RE=E}(\rho_{RA \leftarrow A, RB \leftarrow B, RC \leftarrow C, RE \leftarrow E}(R) \times S))$$

Im folgenden verwenden wir  $\phi$  um diesen Ausdruck zu bezeichnen. D.h. als nächstes betrachten wir den Ausdruck  $S \bowtie \phi$ . Wir erhalten

$$S \bowtie \phi \equiv \pi_{EFGABC}(\sigma_{A=PA \wedge B=PB \wedge C=PC \wedge E=PE}(S \times \rho_{PA \leftarrow A, PB \leftarrow B, PC \leftarrow C, PE \leftarrow E}(\phi)))$$

und kürzen diesen Ausdruck im Folgenden mit  $\psi$  ab.

$\pi_{ABC}(\psi)$  ist somit ein Ausdruck bestehend nur aus primitiven Operatoren, genauso wie  $\pi_{ABC}(R)$ . Um nun  $\pi_{ABC}(\psi) \cap \pi_{ABC}(R)$  mittels primitiver Operatoren auszudrücken erhalten wir

$$\pi_{ABC}(\psi) - (\pi_{ABC}(\psi) - \pi_{ABC}(R))$$

### Aufgabe 6 (Größenabschätzung) [3 Punkte]

Gegeben sind die Relationenschemata  $R(\underline{ABC})$ ,  $S(\underline{CDE})$ , und  $T(\underline{CBF})$  sowie je eine Ausprägung für jedes dieser Schemata, wobei  $|R|$  Tupel in der Ausprägung für  $R$  vorhanden sind,  $|S|$  Tupel in jener für  $S$ , und  $|T|$  Tupel in jener für  $T$ .

- Geben Sie die minimale bzw. maximale Größe (= Anzahl der Tupel) folgender Ausdrücke unter Annahme der angegebenen Werte für  $|R|$ ,  $|S|$ ,  $|T|$  in relationaler Algebra an.
- Begründen Sie Ihre Antwort.
- Geben Sie sowohl für das Minimum als auch für das Maximum jeweils konkrete Ausprägungen mit der angegebenen Anzahl von Tupeln an ( $|R|$  für  $R$ ,  $|S|$  für  $S$ ,  $|T|$  für  $T$ ), auf welchen die Anfrage tatsächlich die von Ihnen berechnete minimale/maximal Anzahl an Tupeln enthält. Geben Sie jeweils außerdem das Ergebnis der Anfrage an.

(a)  $q_1: \sigma_{C=4}(R) \bowtie \sigma_{C=4}(S)$  (mit  $|R| = 5$  und  $|S| = 3$ )

(b)  $q_2: \pi_{ABCD}((R \bowtie S) \cup \rho_{A \leftarrow F}(S \bowtie T)) - \rho_{D \leftarrow F}(R \bowtie T)$  (mit  $|R| = 6$ ,  $|S| = 3$  und  $|T| = 5$ )

(c)  $q_3: ((R \bowtie S) \cap (R \bowtie (\pi_C(R) \cup \pi_C(S)))) \cup (R \bowtie (\pi_C(R) - \pi_C(S)))$  (mit  $|R| = 3$  und  $|S| = 4$ )

(d)  $q_4: (\pi_C(S) - \pi_C(T)) \bowtie (\rho_{U \leftarrow C}(T) \bowtie R)$  (mit  $|R| = 2$ ,  $|S| = 3$  und  $|T| = 2$ )

### Lösung:

#### Aufgabe (a)

[Minimum: 0 | Maximum: 5]

Die Teilabfrage  $\sigma_{C=4}(R)$  kann zwischen 0 und 5 Ergebnissen liefern, da das Attribut  $C$  kein Schlüssel in  $R$  ist und daher beliebige Werte annehmen kann. Anders verhält es sich mit der

Unterabfrage  $\sigma_{C=4}(S)$ . Das Attribut  $C$  ist der Schlüssel von  $S$ . Daher kann es maximal ein Tupel in  $S$  geben in welchem  $C$  den Wert 4 annimmt.

Daraus ergibt sich, dass das kleinstmögliche Ergebnis die leere Menge ist: Wenn  $\sigma_{C=4}(R)$  kein Tupel liefert, dann bleibt auch die Ergebnismenge der Left-Outer-Joins leer.

Für das größtmögliche Ergebnis spielt es nun keine Rolle ob es ein Tupel in  $S$  gibt welches für  $C$  den Wert 4 annimmt oder nicht. In beiden Fällen enthält das Ergebnis von  $q_1$  ein Tupel für jedes Tupel im Ergebnis von  $\sigma_{C=4}(R)$ . Falls das Ergebnis von  $\sigma_{C=4}(S)$  ein Tupel enthält, so entstehen die Ergebnistupel indem jeder Tupel in  $\sigma_{C=4}(R)$  mit dem Tupel in  $\sigma_{C=4}(S)$  kombiniert wird. Enthält  $\sigma_{C=4}(S)$  kein Ergebnis so nehmen die Attribute  $D$  und  $E$  in jedem Ergebnistupel den Wert NULL an (“die Tupel in  $\sigma_{C=4}(R)$  werden mit NULLs aufgefüllt”).

**Minimum: 0**

R		
A	B	C
1	1	1
2	1	1
3	3	1
4	4	3
5	5	5

S		
C	D	E
1	1	2
4	2	3
7	2	3

Ergebnis				
A	B	C	D	E

**Maximum: 5**

R		
A	B	C
1	1	4
2	1	4
3	3	4
4	4	4
5	5	4

S		
C	D	E
1	1	2
4	2	3
7	2	3

Ergebnis				
A	B	C	D	E
1	1	4	2	3
2	1	4	2	3
3	3	4	2	3
4	4	4	2	3
5	5	4	2	3

**Aufgabe (b)**

[Minimum: 0 | Maximum: 9]

Die untere Schranke der Ergebnisgröße von 0 Tupeln ergibt sich aus der Tatsache dass die Ergebnisse von  $R \bowtie S$  und  $S \bowtie T$  leer sein können. Dies geschieht wenn sowhol  $R$  und  $S$  als auch  $S$  und  $T$  in keinem Tupel den selben Wert auf den gemeinsamen Attributen besitzen.

Das größtmögliche Ergebnis erhält man, wenn die Teilanfrage  $(R \bowtie S) \cup \rho_{A \leftarrow F}(S \bowtie T)$  ein möglichst großes Ergebnis ergibt, welches so wenig Tupel wie möglich mit dem Ergebnis der Teilanfrage  $\rho_{D \leftarrow F}(R \bowtie T)$  gemeinsam hat.

Das größtmögliche Ergebnis eines Joins  $\bowtie$  erhält man, wenn in beiden Relationen sämtliche Tupel auf den gemeinsamen Attributen den selben Wert annehmen, da dann jedes Tupel der einen Relation mit jedem Tupel der anderen Relation zu einem eigenen Ergebnistupel kombiniert werden kann (d.h. man erhält das selbe Ergebnis wie beim Kreuzprodukt). Im Fall  $R \bowtie S$  ist dies jedoch nicht möglich: das gemeinsame Attribut von  $R$  und  $S$  ist  $C$ , welches der Schlüssel von  $S$  ist. D.h. keine zwei Tupel in  $S$  können den selben Wert auf  $C$  annehmen. Deshalb kann es für jedes Tupel in  $R$  maximal ein “passendes” Tupel in  $S$  geben. Da  $C$  in  $R$  kein Schlüssel ist können verschiedene Tupel den selben Wert auf  $C$  annehmen, weshalb es tatsächlich Fälle geben kann, in denen für jedes Tupel in  $R$  genau ein Ergebnistupel erzeugt wird. Das größtmögliche Ergebnis von  $R \bowtie S$  enthält also 6 Tupel.

Für  $S \bowtie T$  ist die Situation ähnlich, aber es gibt einen entscheidenden Unterschied: das gemeinsame Attribut  $C$  ist in beiden Relationen der Schlüssel. Daher kann zu jedem Tupel in

S maximal ein passendes Tupel in T existieren, und umgekehrt. Deshalb ist es zwar möglich dass jedes der drei Tupel in S ein entsprechendes Tupel in T findet, aber für maximal drei der fünf Tupel in T kann ein entsprechendes Tupel in S existieren. Daher enthält das größtmögliche Ergebnis von  $S \bowtie T$  drei Tupel.

Da es möglich ist die Inhalte von R, S und T so zu wählen dass die Ergebnisse von  $R \bowtie S$  und  $\rho_{A \leftarrow F}(S \bowtie T)$  kein gemeinsames Tupel enthalten, besteht das größtmögliche Ergebnis der Vereinigung dieser beiden Ergebnisse aus neun Tupeln. Da es wiederum möglich ist R, S und T so zu wählen dass das Ergebnis der Vereinigung kein Tupel enthält das auch in  $\rho_{D \leftarrow F}(R \bowtie T)$  vorkommt bleiben all diese neun Tupel im Ergebnis enthalten. (Ein Beispiel wie die Ausprägungen von R, S und T gewählt werden müssen ist weiter unten zu finden; wichtig ist sowohl für die Vereinigung als auch die Mengendifferenz darauf zu achten dass es für jedes Paar von Relationen jeweils ein Attribut gibt auf welchem sich die Werte unterscheiden – für die Vereinigung sind das A in R und F (welches zu A umbenannt wird) in T, und für die Mengendifferenz sind dies D in S und F (welches zu D umbenannt wird) in T.)

**Minimum: 0**

R		
A	B	C
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0

S		
C	D	E
11	11	11
12	12	12
13	13	13

T		
C	B	F
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0

Ergebnis			
A	B	C	D

**Maximum: 9**

R		
A	B	C
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1

S		
C	D	E
1	7	7
2	7	7
3	7	7

T		
C	B	F
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0

Ergebnis			
A	B	C	D
1	1	1	7
2	1	1	7
3	1	1	7
4	1	1	7
5	1	1	7
6	1	1	7
0	1	1	7
0	2	2	7
0	3	3	7

**Aufgabe (c)**

[Minimum: 3 | Maximum: 3]

Diese Abfrage gibt als Ergebnis immer den Inhalt von R zurück. Dies ist der Fall weil die Teilabfrage  $(R \bowtie S) \cap (R \bowtie (\pi_C(R) \cup \pi_C(S)))$  äquivalent ist zu  $R \bowtie S$ , und die Teilabfrage  $(R \bowtie (\pi_C(R) - \pi_C(S)))$  äquivalent ist zu  $R - (R \bowtie S)$ . Die Vereinigung der Ergebnisse der beiden Abfragen ergibt daher R.

Das Ergebnis der ersten Teilabfrage  $(R \bowtie S) \cap (R \bowtie (\pi_C(R) \cup \pi_C(S)))$  muss, auf Grund des Durchschnitts  $\cap$ , eine Teilmenge von  $R \bowtie S$  sein. Dass bei der Berechnung des Durchschnitts kein Tupel aus  $R \bowtie S$  entfernt wird ergibt sich wie folgt: Nachdem  $\pi_C(R) \cup \pi_C(S)$  alle Werte enthält welche das Attribut C in R annimmt, findet jedes Tupel in R in  $R \bowtie (\pi_C(R) \cup \pi_C(S))$

einen “passenden” Partner. Das heißt das Ergebnis von  $R \bowtie (\pi_C(R) \cup \pi_C(S))$  ist wiederum  $R$ . Dadurch ist  $(R \bowtie S) \cap (R \bowtie (\pi_C(R) \cup \pi_C(S)))$  äquivalent zu  $(R \bowtie S) \cap R$ , was wiederum das selbe Ergebnis liefert wie  $R \bowtie S$ .

Für die Teilabfrage  $(R \bowtie (\pi_C(R) - \pi_C(S)))$  ist es wichtig dass  $C$  das einzige gemeinsame Attribut von  $R$  und  $S$  ist. D.h. der natürliche Verbund (Natural-Join) geschieht über  $C$ . Die Teilabfrage  $\pi_C(R) - \pi_C(S)$  liefert nun alle Werte welche das Attribut  $C$  in  $R$  annimmt, welche *nicht* in  $S$  vorkommen (wiederum für das Attribut  $C$ ). Auf Grund dessen bleiben im Ergebnis von  $R \bowtie (\pi_C(R) - \pi_C(S))$  nur jene Tupel von  $R$  erhalten welche auf  $C$  einen Wert annehmen welcher nicht auf  $C$  in  $S$  vorkommt – d.h. genau jene Tupel welche *keinen* “Join-Partner” in  $S$  haben. Dies sind aber genau jene Tupel in  $R$ , welche im Ergebnis von  $R \bowtie S$  *nicht* enthalten sind.

Anstatt nur eines Beispiels sind unten drei Beispiele angegeben: das erste zeigt den Fall dass das gesamte Ergebnis in der linken Teilabfrage der Vereinigung erzeugt wird, das zweite den Fall in dem das gesamte Ergebnis in der rechten Teilabfrage der Vereinigung erzeugt wird, und das dritte einen Fall in dem verschiedene Teile des Ergebnisses aus beiden Teilabfragen kommen.

### Minimum = Maximum: 3

R		
A	B	C
1	1	1
2	2	2
3	3	3

S		
C	D	E
5	1	1
6	2	2
7	3	3
8	4	4

Ergebnis		
A	B	C
1	1	1
2	2	2
3	3	3

R		
A	B	C
1	1	1
2	2	2
3	3	3

S		
C	D	E
1	1	1
2	2	2
3	3	3
4	4	4

Ergebnis		
A	B	C
1	1	1
2	2	2
3	3	3

R		
A	B	C
1	1	1
2	2	2
3	3	3

S		
C	D	E
1	1	1
2	2	2
7	3	3
8	4	4

Ergebnis		
A	B	C
1	1	1
2	2	2
3	3	3

#### Aufgabe (d)

[Minimum: 1 | Maximum: 6]

Auf Grund der Verwendung des Left-Outer-Joins kann das Gesamtergebnis nie weniger Tupel enthalten als das Ergebnis der Teilabfrage  $\pi_C(S) - \pi_C(T)$ . Um also das kleinstmögliche Ergebnis zu erhalten muss das kleinstmögliche Ergebnis dieser Teilabfrage ermittelt werden. Dies erhält man dann, wenn  $\pi_C(S)$  möglichst wenig Tupel enthält und so viele wie möglich davon ebenfalls in  $\pi_C(T)$  vorkommen. Nachdem  $C$  der Schlüssel sowohl von  $S$  als auch von  $T$  ist kann es keiner der beiden Relationen zwei Tupel geben welche den selben Wert für  $C$  annehmen. D.h. egal wie die Ausprägung von  $S$  und  $T$  aussehen,  $\pi_C(S)$  enthält immer drei Tupel, und  $\pi_C(T)$  immer zwei. (Wäre  $C$  kein Schlüssel wäre der kleinstmögliche Wert jeweils ein Tupel, und drei bzw. zwei würde den größtmöglichen Wert darstellen.) Das kleinstmögliche

Ergebnis erhält man nun also dann wenn die beiden Tupel in  $\pi_C(T)$  auch in  $\pi_C(S)$  vorkommen, da das Ergebnis der Mengendifferenz dann das eine zusätzliche Tupel von  $\pi_C(S)$  enthält.

Das Ergebnis des Left-Outer-Joins enthält nun dann ebenfalls ein Tupel wenn das Ergebnis der Teilabfrage  $\rho_{U \leftarrow C}(T) \bowtie R$  entweder leer ist oder genau ein Tupel enthält welches den selben Wert für  $C$  hat wie das Ergebnis der Teilanfrage  $\pi_C(S) - \pi_C(T)$ .

Um das größtmögliche Ergebnis zu erhalten betrachten wir zuerst einmal den Left-Outer-Join. Nachdem das Ergebnis auf der linken Seite nur ein Attribut enthält und Ergebnisse in der relationalen Algebra keine Duplikte enthalten, kann jedes Tupel auf der rechten Seite des  $\bowtie$  Operators *maximal ein* Tupel auf der linken Seite mit dem selben Wert für  $C$  finden. Nachdem es sich um den Left-Outer-Join handelt, erhalten wir die meisten Ergebnistupel wenn: alle Tupel auf der rechten Seite den selben Wert für  $C$  haben, und es auf der linken Seite ein Tupel mit diesem Wert für  $C$  gibt. Dies führt dazu dass für das eine Tupel dessen Wert für  $C$  auf der rechten Seite auftaucht so viele Tupel im Ergebnis erzeugt werden wie es Tupel auf der rechten Seite gibt (das eine Tupel auf der linken Seite wird mit jedem Tupel auf der rechten Seite gejoined). Zusätzlich wird für die verbleibenden Tupel auf der linken Seite (für welche nun kein "Join-Partner" auf der rechten Seite existiert) jeweils ein Tupel im Ergebnis erzeugt in welchem die zusätzlichen Attribute den Wert NULL annehmen.

Wir müssen uns also zum Einen überlegen wie viele Tupel das Ergebnis der Teilabfrage  $\pi_C(S) - \pi_C(T)$  maximal haben kann. Wir wissen bereits dass die beiden Abfragen immer jeweils drei bzw. zwei Tupel enthalten. Das größtmögliche Ergebnis ergibt sich daher wenn die zwei Tupel im Ergebnis von  $\pi_C(T)$  nicht im Ergebnis von  $\pi_C(S)$  enthalten sind. Dann besteht das Ergebnis der Teilabfrage aus drei Tupeln.

Für die Teilabfrage  $\rho_{U \leftarrow C}(T) \bowtie R$  gilt dass (auf Grund der Umbenennung von  $C$  in  $T$ ) das einzige gemeinsame Attribut von  $T$  und  $R$  das Attribut  $B$  ist, welches weder in  $T$  noch in  $R$  ein Schlüssel ist. Daher können alle Tupel in  $R$  und in  $T$  den selben Wert für  $B$  haben (dass es in  $R$  ein Teil des Schlüssels ist hindert es nicht daran in verschiedenen Tupel den selben Wert anzunehmen). Daher kann jedes Tupel in  $R$  mit jedem Tupel in  $T$  zu einem Ergebnistupel kombiniert werden, und wir erhalten  $2 * 2 = 4$  Tupel im Ergebnis. Nachdem es möglich ist dass all diese Tupel den selben Wert für  $C$  haben (dafür ist es nur nötig dass beide Tupel in  $R$  den selben Wert für  $C$  haben, was möglich ist weil  $C$  kein Schlüssel ist), und dieser Wert für  $C$  in  $\pi_C(S) - \pi_C(T)$  vorkommt, wenden wir nun die zuvor gemachte Überlegung an: ein Tupel aus  $\pi_C(S) - \pi_C(T)$  wird mit jedem der vier Tupel in  $\rho_{U \leftarrow C}(T) \bowtie R$  zu insgesamt vier Ergebnistupeln kombiniert, und dazu kommen noch die beiden verbliebenen Tupel in  $\pi_C(S) - \pi_C(T)$  welche mit NULLs zu Ergebnistupeln kombiniert werden. In Summe ergibt dies sechs Tupel.

**Minimum: 1**

R		
A	B	C
4	4	4
5	5	5

S		
C	D	E
1	1	1
2	2	2
3	3	3

T		
C	B	F
1	1	1
2	2	2

Ergebnis				
C	A	U	B	F
3	NULL	NULL	NULL	NULL

**Maximum: 6**



<b>R</b>		
<u>A</u>	<u>B</u>	<u>C</u>
1	0	1
2	0	1

<b>S</b>		
<u>C</u>	<u>D</u>	<u>E</u>
1	1	1
2	2	2
3	3	3

<b>T</b>		
<u>C</u>	<u>B</u>	<u>F</u>
0	0	1
4	0	1

<b>Ergebnis</b>				
<b>C</b>	<b>A</b>	<b>B</b>	<b>U</b>	<b>F</b>
1	1	0	0	1
1	1	0	4	1
1	2	0	0	1
1	2	0	4	1
2	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL

**Aufgabe 7 (Formalisieren von Anfragen)** [3 Punkte]

Eine Autowerkstatt speichert in einer Datenbank alle Autos inklusive Reparaturen und Servicearbeiten, die durchgeführt wurden (Primärschlüssel sind unterstrichen, Fremdschlüssel sind kursiv geschrieben).

Auto	( <u>Bez</u> , Marke, Modell, Zulassungsjahr)
Service	( <u>AutoBez:Auto.Bez</u> , <u>Datum</u> , Kommentar)
Reparatur	( <u>AutoBez:Auto.Bez</u> , <u>SchadenNr</u> , Datum, Kommentar)
TechnikerIn	( <u>PNr</u> , Name)
Repariert	( <u>PNr:Techniker.PNr</u> , <u>AutoBez:Reparatur.AutoBez</u> , <u>SchadenNr:SchadenNr</u> )
Serviciert	( <u>AutoBez:Service.AutoBez</u> , <u>Datum:Service.Datum</u> , <u>TechnikerIn:TechnikerIn.PNr</u> )

Sie dürfen im Folgenden gerne passende (eindeutige) Abkürzungen sowohl für die Relationen- als auch die Tabellennamen verwenden. Formulieren Sie alle Abfragen in **relationaler Algebra**, **Tupelkalkül** und **Domänenkalkül**.

- (a) Geben Sie den Namen aller TechnikerInnen aus, die schon ein Auto der Marke ‘Maserati’ repariert haben.

**Lösung:**

*Relationale Algebra:*

$$\pi_{\text{Name}} \left( \text{Techniker} \bowtie (\text{Repariert} \bowtie \sigma_{\text{Marke}='Maserati'} (\rho_{\text{AutoBez} \leftarrow \text{Bez}}(\text{Auto}))) \right)$$

*Tupelkalkül:*

$$\{[t.\text{Name}] \mid t \in \text{Techniker} \wedge \exists r \in \text{Repariert} (t.\text{PNR} = r.\text{PNR} \wedge \exists a \in \text{Auto} \\ (a.\text{Marke} = \text{'Maserati'} \wedge a.\text{Bez} = r.\text{AutoBez}))\}$$

*Domänenkalkül:*

$$\{[name] \mid \exists pnr ([pnr, name] \in \text{Techniker} \wedge \\ \exists autobez, schadennr ([pnr, autobez, schadennr] \in \text{Repariert} \wedge \\ \exists modell, zj ([autobez, \text{'Maserati'}, modell, zj] \in \text{Auto})))\}$$

- (b) Geben Sie alle Automodelle gemeinsam mit der Marke aus, von denen in der Werkstatt bereits ein Service an einem Auto durchgeführt wurde, aber jedoch noch nie ein Auto repariert wurde.

**Lösung:**

*Relationale Algebra:*

$$\pi_{\text{Marke,Modell}} \left( \rho_{\text{Bez} \leftarrow \text{AutoBez}}(\text{Service}) \bowtie \text{Auto} \right) - \pi_{\text{Marke,Modell}} \left( \rho_{\text{Bez} \leftarrow \text{AutoBez}}(\text{Reparatur}) \bowtie \text{Auto} \right)$$

*Tupelkalkül:*

$$\{[a.\text{Marke}, a.\text{Modell}] \mid a \in \mathbf{Auto} \wedge \exists s \in \mathbf{Service}(a.\text{Bez} = s.\text{AutoBez} \wedge \\ \neg \exists a2 \in \mathbf{Auto}(a.\text{Marke} = a2.\text{Marke} \wedge a.\text{Modell} = a2.\text{Modell}) \wedge \\ \exists r \in \mathbf{Reparatur}(r.\text{AutoBez} = a.\text{Bez}))\}$$

*Domänenkalkül:*

$$\{[marke, modell] \mid \exists bez, zj([bez, marke, modell, zj] \in \mathbf{Auto} \wedge \\ \exists datum, kommentar([bez, datum, kommentar] \in \mathbf{Service} \wedge \\ \neg \exists bez2, zj2([bez2, marke, modell, zj2] \in \mathbf{Auto} \wedge \\ \exists schadenNr, dat, kommentar([bez2, schadenNr, dat, kommentar] \in \mathbf{Reparatur})))\}$$

- (c) Gesucht ist die SchadenNr von allen Schäden die nach dem 1.6.2018 repariert wurden und zwar nur an Autos die nach 2015 zugelassen worden sind, gemeinsam mit de Namen der Techniker welche den Schaden repariert haben.

**Lösung:**

*Relationale Algebra:*

$$\pi_{\text{Name, SchadenNr}} \left( \mathbf{Techniker} \bowtie \mathbf{Repariert} \bowtie \right. \\ \left. \sigma_{\text{Zulassungsjahr} > 2015}(\rho_{\text{AutoBez} \leftarrow \text{Bez}}(\mathbf{Auto})) \bowtie \sigma_{\text{Datum} > 1.6.2018}(\mathbf{Reparatur}) \right)$$

*Tupelkalkül:*

$$\{[t.\text{Name}, r.\text{SchadenNr}] \mid t \in \mathbf{Techniker} \wedge r \in \mathbf{Repariert} \wedge t.\text{PNr} = r.\text{PNr} \wedge \\ \exists rep \in \mathbf{Reparatur}(rep.\text{AutoBez} = r.\text{AutoBez} \wedge rep.\text{SchadenNr} = r.\text{SchadenNr} \wedge rep.\text{Datum} > 1.6.2018) \wedge \\ \exists a \in \mathbf{Auto}(a.\text{Zulassungsjahr} > 2015 \wedge a.\text{Bez} = r.\text{AutoBez})\}$$

*Domänenkalkül:*

$$\{[name, schadenNr] \mid \exists pnr, autobez([pnr, name] \in \mathbf{Techniker} \wedge \\ [pnr, autobez, schadenNr] \in \mathbf{Repariert} \wedge \\ \exists datum, kommentar([autobez, schadenNr, datum, kommentar] \in \mathbf{Reparatur} \wedge datum > 1.6.2018) \wedge \\ \exists marke, modell, zj([autobez, marke, modell, zj] \in \mathbf{Auto} \wedge zj > 2015))\}$$