1. Übungsblatt (WS 2019) – Musterlösung

3.0 VU Datenmodellierung / 6.0 VU Datenbanksysteme

Informationen zum Übungsblatt

Allgemeines

In diesem Übungsteil entwerfen Sie eine kleine Datenbank mittels EER-Diagrammen, überführen ein EER-Diagramm in ein Relationenschema, und üben den Umgang mit der relationalen Algebra und dem Relationenkalklül.

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 8 Aufgaben, auf welche Sie insgesamt 15 Punkte erhalten können.

Deadlines

bis 30.10.	**12:00Uhr**	Upload der Abgabe über TUWEL
ab 13.11.	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 auseinander setzen, 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]

NASA und ESA planen eine gemeinsame Neuanschaffung ihrer wichtigsten Hardwarekomponenten. Um in dieser sensiblen Angelegenheit eine bestmögliche Entscheidung treffen zu können, wird eine Datenbank benötigt in welcher die wichtigsten Informationen zu den möglichen Alternativen zusammengefasst werden können. Nach einem intensiven weltweiten Auswahlverfahren hat man sich schlussendlich dazu entschieden, den Entwurf der Datenbank Ihnen zu übertragen, und übermittelt Ihnen das Ergebnis der Anforderungsanalyse.

Zeichnen Sie aufgrund der vorliegenden Informationen (siehe nächste Seite) ein EER-Diagramm. Verwenden Sie dabei die in der Vorlesung vorgestellte Schreibweise, sowie 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!). Sie können das EER-Diagramm aber natürlich mit jeder beliebigen Software erstellen.

Beschreibung des zu modellierenden Sachverhalts:

Sowohl Kaffeemaschinen als auch Kaffeekapseln sind Produkte, welche einen Preis (PREIS) und eine Modellnummer (MODELNR) haben, wobei die Modellnummer für jedes Produkt eindeutig ist. Für Kaffeemaschinen werden zur Entscheidungsfindung darüber hinaus der Name (NAME), der maximale Wasserdruck (DRUCK), sowie die erwartete Lebensdauer (ANZKAPS) gemessen in Anzahl der zubereiteten Portionen gesammelt. Um etwas Ordnung zu schaffen soll außerdem vermerkt werden falls eine Kaffeemaschine eine Weiterentwicklung (maximal) einer älteren Maschine ist (wobei jede Kaffeemaschine die Weiterentwicklung einer älteren Maschine sein kann). Für Kaffeekapseln wird das Material (MATERIAL) der Kapsel gespeichert, sowie der eindeutige Typ der Kaffeekapsel. Kapsel-Typen besitzen ein Volumen (VOLUMEN) und lassen sich eindeutig über die Kombination aus ihrer Größe (GROESSE) und einer ID (KTID) identifizieren. Ein Kapsel-Typ kann mit anderen Kapsel-Typen kompatibel sein. Jede Kaffemaschine unterstützt zumindest einen Kapsel-Typ, und keine mehr als zehn.

Produkte werden von Herstellern erzeugt. Zu jedem Hersteller wird das Land in welchem er steuerpflichtig ist (STL) vermerkt, sowie der eindeutige Markenname (MARKE). Darüber hinaus wird zu jedem Hersteller eine beliebige Anzahl an Reviews gesammelt, wobei die Reviews für jeden Hersteller durchnummeriert (ID) werden und die Beurteilung gespeichert wird (TXT). Manche Kaffeemaschinen benötigen Lizenzen um bestimmte Kapsel-Typen unterstützen zu dürfen. In solchen Fällen soll für jede betroffene Kaffeemaschine gespeichert werden für welche Kapsel-Typen Lizenzen von welchen Herstellern erworben wurden, sowie die jeweilige Lizenzgebühr (GEBUEHR).

Produkte werden durch Händler verkauft. Für jeden Händler sind der Name (NAME) und die Website (WEB) bekannt, wobei sowohl der Name als auch die Website aller Händler unterschiedlich ist. Händler bieten oftmals Mengenrabatte an. Die unterschiedlichen Mengenrabatte eines Händlers können durch die Kombination ihrer Bezeichnung (LABEL) und des gewährten Rabattes (RABATT) unterschieden werden; die Händler sprechen sich dabei jedoch nicht untereinander ab. Um einen Mengenrabatt in Anspruch zu nehmen muss von einer bestimmten Auswahl an Produkten jeweils eine Mindestbestellmenge (MENGE) gekauft werden. Dabei werden solche Rabatte erst ab dem Kauf von mindestens vier verschiedenen Produkten gewährt, und jedes Produkt kann eine andere Mindestbestellmenge aufweisen. Zusätzlich kann es zu einem Mengenrabatt noch Gutschein-Codes geben. Diese Codes (CODE) sind in Kombination mit dem Mengenrabatt eindeutig.

Zu jedem Kaffee wird das Aroma (AROMA), die Säure (SAEURE) sowie der Körper (KOERPER) vermerkt. Die verschiedenen Kaffees sollen an Hand Ihres Namens (NAME) unterschieden werden. Es wird außerdem gespeichert, falls es sich bei einem Kaffee um einen Ristrettos, Lungos oder Espressos handelt. Darüber hinaus wird bei einem Ristretto seine Rösttemperatur (ROETEMP) und bei einem Espresso der beim Brühen verwendete Druck (DRUCK) gespeichert.

Schlussendlich wird für jede Kaffekapsel der darin enthaltene Kaffee vermerkt, sowie für jede Kaffeemaschine welche Kaffees sie zubereiten kann (wobei jede Kaffeemaschine mindestens einen Kaffee zubereiten kann).

Lösung: Siehe Abbildung 1.

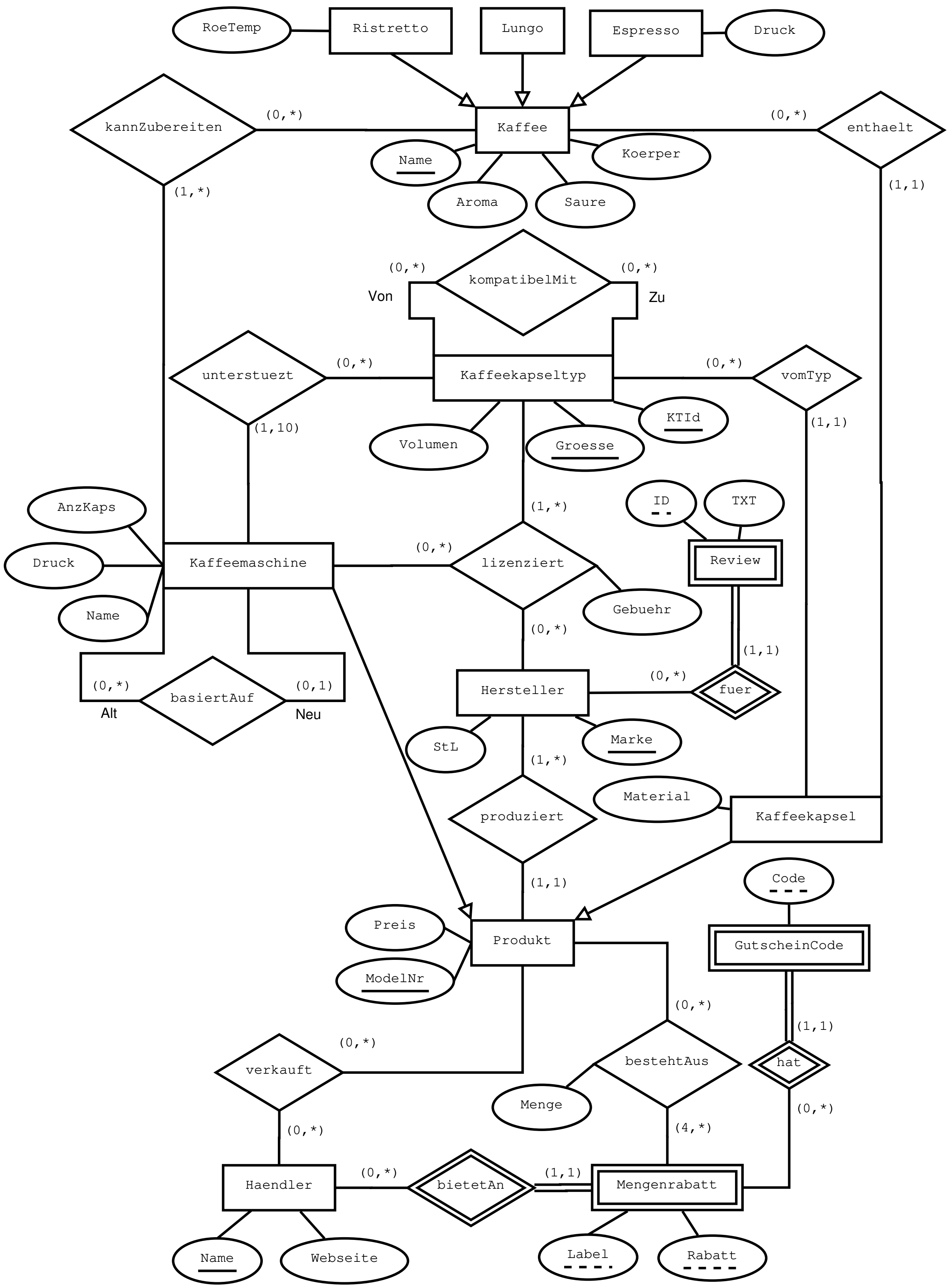


Abbildung 1: Lösung von Aufgabe 1

Aufgabe 2 (Semantik von EER Diagrammen) [1 Punkt]

Betrachten Sie das in Abbildung 2 dargestellte EER-Diagramm, welches eine (willkürlich festgelegte) Variante des sogenannten *Property Graph* Modells beschreibt. (Hintergrund: Property Graphs, wenn auch in leicht unterschiedlichen Varianten, sind ein ebenfalls recht populäres Datenmodell, welches sich besonders zur Speicherung und Verarbeitung von netzwerkartigen Daten eignet; zur Lösung dieser Aufgabe ist jedoch keinerlei Wissen über Property Graphs erforderlich, es geht nur um die im EER-Diagramm dargstellten Informationen.)

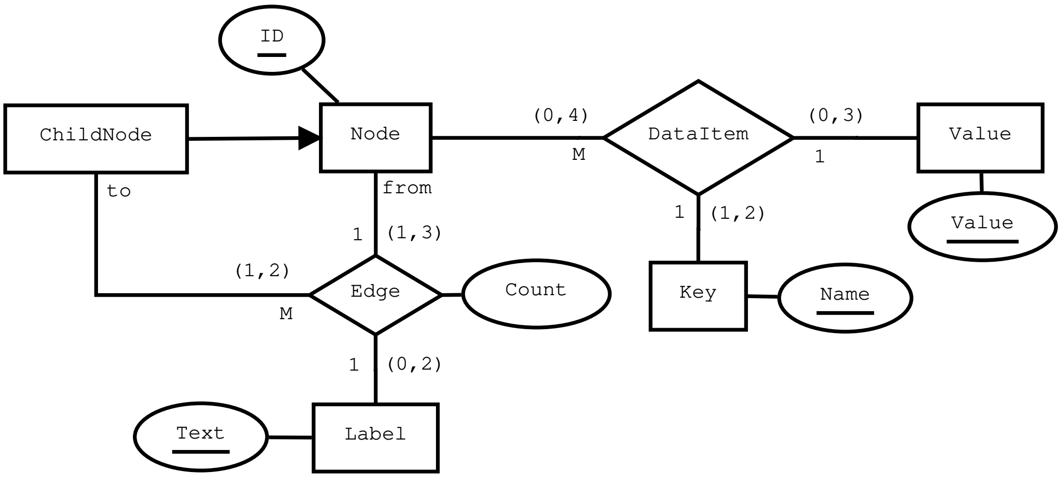


Abbildung 2: EER-Diagramm zu Aufgabe 2

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 führt das Weglassen welcher Notation 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 `Edge` die *Funktionalitäten* (1:1:M) weg, so kann durch die (min,max)-Notation weder ausgedrückt werden, dass jeder Kante (=

jedes Paar aus `from`- und `to`- Knoten) ein eindeutiges Label haben muss, noch dass kein `ChildNode` zwei oder mehr eingehende Kanten mit dem selben Label von verschiedenen `Nodes` haben darf (jede Kombination aus `ChildNode` und `Label` darf mit maximal einem `Node` in einer `Edge`-Beziehung stehen).

Ein konkretes Gegenbeispiel wären Knoten v_1 und v_2, wobei v_1 auch ein `ChildNode` ist, sowie ein `Label` "Kante". Nach der (min,max)-Notation wäre folgendes eine gültige Instanz des Beziehungstyps `Edge`: Die Instanz beinhaltet zwei (gerichtete) Kanten $e_1 = (v_1, v_1)$ und $e_2 = (v_2, v_1)$. Dabei ist beiden Kanten das `Label` "Kante" zugeordnet und der Wert von `count` ist 1. Dies wäre durch die Funktionalitäten verboten, da nun zwei Entitäten vom Typ `Node` mit dem Paar (v_1,"Kante") aus Entitäten v_1 und "Kante" in einer `Edge`-Beziehung stehen. Dies sind die Entitäten v_1 und v_2.

- Lässt man beim Beziehungstyp `Label` die *(min,max)-Notation* weg, so wird durch die Angabe der Funktionalitäten weder ausgedrückt, dass jede Entität vom Typ `ChildNode` in mindestens einer und maximal zwei Beziehungen vom Typ `Label` auftreten muss, noch dass jede Entität vom Typ `Node` in mindestens einer und maximal drei solcher Beziehungen auftreten darf, noch dass jede Entität vom Typ `Label` in maximal zwei solchen Beziehungen vorkommen darf.

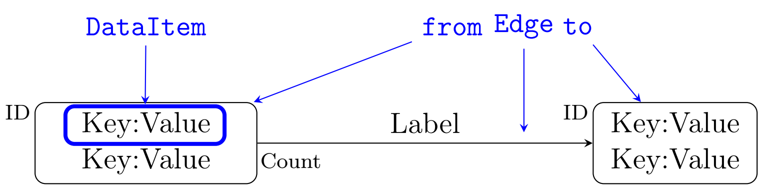
  Ein konkretes Gegenbeispiel wären daher zwei Entitäten v_1, v_2 vom Typ `Node`, wobei v_1 auch vom Typ `ChildNode` ist, sowie eine Entität "Text" vom Typ `Label`. Eine Menge aus `Edge`-Beziehungen, welche nur eine einzelne gerichtete Kante von v_2 nach v_1 mit dem `Label` "Text" und einem Wert 1 für `count` enthält (also nur ein Tupel $(v_2, v_1, \text{"Text"}, 1)$) würde alle durch die Funktionalitäten ausgedrückten Bedingungen erfüllen, nicht jedoch die durch die (min,max)-Notation beschriebene Bedingung dass jeder Knoten mindestens eine ausgehende Kante besitzen muss (mindestens ein mal die `from` Rolle in einer `Edge` Beziehung einnehmen muss). Der Knoten v_1 verletzt diese Bedingung.

- Beim Beziehungstyp `DataItem` ergibt sich ein analoger Sachverhalt.

2. Nehmen Sie an, Sie erhalten den in Abbildung 3 graphisch dargestellten Graph vorgelegt, und sollen nun überprüfen, ob es sich dabei um eine gültige Instanz eines Property Graphs entsprechend der im EER-Diagramm beschriebenen Spezifikation handelt.

   Beschreiben Sie mindestens sieben (7) Verletzungen der im EER-Diagramm dargestellten Vorschriften durch die Graphinstanz. (*Hinweis:* Beschränken Sie sich ausschließlich auf Verletzungen von im EER-Diagramm beschriebenen Sachverhalten. Ignorieren Sie insbesonders Sachverhalte die zwar "unlogisch" erscheinen, aber nicht im EER-Diagramm spezifiziert wurden.)

   Das "Mapping" zwischen den im EER-Diagramm beschriebenen Entitäts- und Beziehungstypen und den in Abbildung 3 konkreten Ausprägungen (Instanzen), ist in der folgenden Grafik dargestellt:



Dabei sind folgende Punkte zu beachten:

- Knoten ohne eingehende Kanten sind immer nur Entitäten vom Typ `Node`, und nicht vom Typ `ChildNode`.
- Knoten mit eingehenden Kanten sind immer vom Typ `ChildNode`.
- Entitäten vom Typ `Label`, `Key` und `Value` werden durch den Wert ihrer Schlüsselattribute `Text`, `Name` bzw. `Value` dargestellt.
- Strichpunkte ";" Trennen verschiedene, durch ihren Key repräsentierte, Entitäten des selben Typs.
- In der obigen Grafik blau dargestellte Elemente sind zusätzliche Erklärungen und nicht Teil des Graphen.

Lösung:
Es finden sich (mindestens) die folgenden 13 Verletzungen:

- In `Node` $n9$ sind an einer `DataItem` Beziehung zwei Entitäten vom Typ `Value` beteiligt, nämlich die Entitäten "secret" und "important". Der Beziehungstyp `DataItem` ist jedoch drei-stellig, d.h. an jeder Beziehung ist eine `Node`, eine `Key` und eine `Value` Entität beteiligt.
- An der Beziehung `Edge` zwischen $n10$ und $n6$ sind neben den beiden `Node`-Entitäten auch zwei Entitäten vom Typ `Label` beteiligt ("next" und "prev"). Der Beziehungstyp `Edge` ist jedoch 3-stellig und verlangt die Beteiligung von *genau einer* Entität vom Typ `Label`.
- Das `Label` "next" taucht in drei `Edge` Beziehungen auf, nämlich auf den Kanten $n6 \to n9$, $n9 \to n10$ sowie $n10 \to n6$. Jedes Label darf jedoch in maximal zwei solcher Beziehungen auftreten.
- Das Tripel ($n2, n6,$"docked") ist zwei mal als Instanz des Beziehungstyps `Edge` aufgeführt (jeweils mit anderen Werten für `count`). Jede Beziehung kann jedoch maximal einmal auftreten.
- Der Knoten $n6$ übernimmt in 4 Beziehungen vom Typ `Edge` die Rolle `to` (hat vier eingehende Kanten). Das erlaubte Maximum hierfür liegt jedoch bei 2.
- Der Knoten $n11$ übernimmt nie die Rolle `from` in einer Beziehung vom Typ `Edge` (hat keine ausgehende Kante). Gefordert wird mindestens 1.
- Der Knoten $n12$ übernimmt nie die Rolle `from` in einer Beziehung vom Typ `Edge` (hat keine ausgehende Kante). Gefordert wird mindestens 1.
- Die beiden Beziehungen ($n4, n8,$"selects") und ($n4, n8,$"trained") (für die Notation (`from`, `to`, `label`) verletzen die geforderte Funktion (`from`, `to`) $\to$ `Label`.
- An der Beziehung vom Typ `Edge` zwischen $n7$ und $n10$ nimmt keine Entität vom Typ `Label` teil. Da dies ein drei-stelliger Beziehungstyp ist, ist dies nicht erlaubt.
- Die beiden Kanten (= Entitäten vom Typ `Edge`) ($n7, n12,$"achieve") und ($n8, n12,$"achieve") verletzen die geforderte Funktion (`to`, `Label`) $\to$ `from`, da hier zwei Kanten mit verschiedenem Start- aber selben Endknoten das selbe Label besitzen.
- Der `Key` "position" kommt in 3 Beziehungen des Types `DataItem` vor. Erlaubt wäre maximal 2 mal.

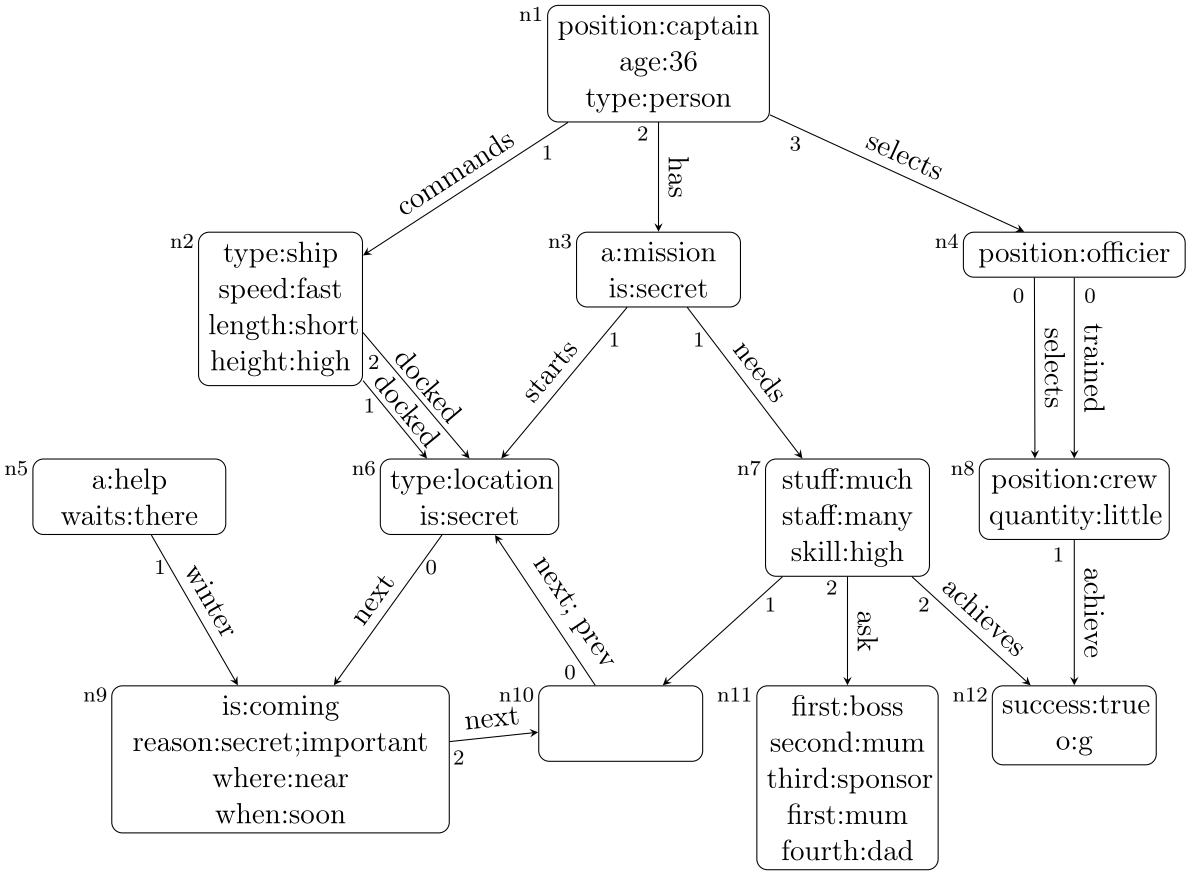


Abbildung 3: Angabe für Aufgabe 2: Beispielgraph welcher an Hand des EER-Diagramms in Abbildung 2 überprüft werden soll.

- Der `Key` "type" kommt in 3 Beziehungen des Types `DataItem` vor. Erlaubt wäre maximal 2 mal.
- Der `Key` "is" kommt in 3 Beziehungen des Types `DataItem` vor. Erlaubt wäre maximal 2 mal.
- Der `Node` $n11$ kommt in 5 Beziehungen des Typs `DataItem` vor. Erlaubt sind maximal 4.
- Die beiden Beziehungen ($n11$,"first","boss") und ($n11$,"first","mum") verletzen die geforderte Funktion $(\texttt{Node}, \texttt{Key}) \rightarrow \texttt{Value}$.
- Die beiden Beziehungen ($n11$,"second","mum") und ($n11$,"first","mum") verletzen die geforderte Funktion $(\texttt{Node}, \texttt{Value}) \rightarrow \texttt{Key}$.

Aufgabe 3 (Überführung ins Relationenschema)	[2 Punkte]

Überführen Sie das EER-Diagramm aus Abbildung 4 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.

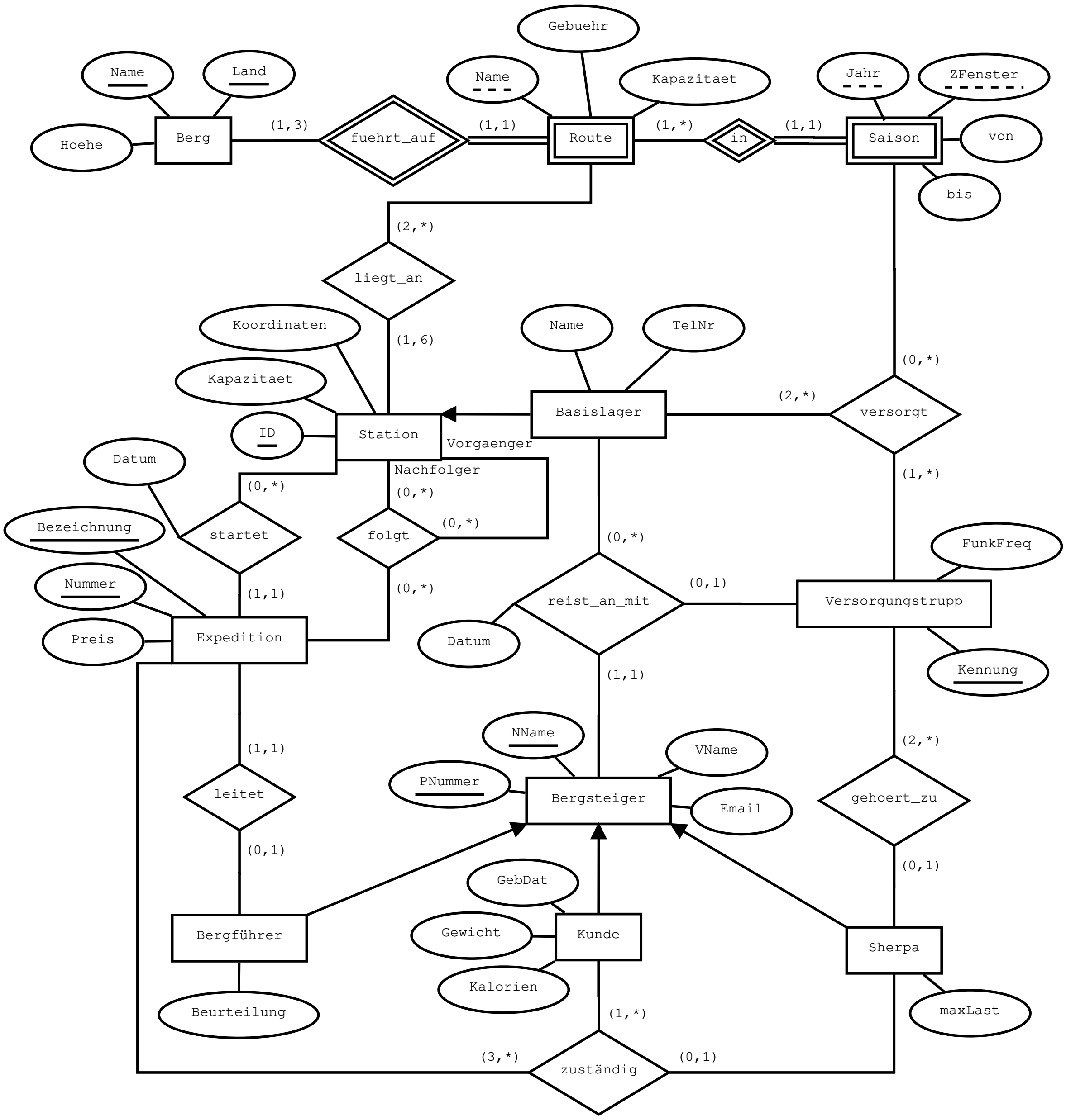


Abbildung 4: EER-Diagramm zu Aufgabe 3

Lösung:

Berg	(name, land, hoehe)
Route	(berg_name: *Berg.name*, land: *Berg.land*, route_name, gebuehr, kapazitaet)
Saison	(berg_name: *Route.berg_name*, land: *Route.land*, route_name: *Route.route_name*, jahr, zfenster, von, bis)
Station	(ID, koordinaten, kapazitaet)
Basislager	(ID:*Station.ID*, name, telnr)
Expedition	(bezeichnung, nummer, preis, ID: *Station.ID*, datum, pnummer: *Bergfuehrer.pnummer*, nname: *Bergfuehrer.nname*)
Bergsteiger	(pnummer, nname, vname, email, datum, basis_id: *Basislager.ID*, kennung: *Versorgungstrupp.kennung*)
Bergfuehrer	(pnummer: *Bergsteiger.pnummer*, nname: *Bergsteiger.nname*, beurteilung)
Kunde	(pnummer: *Bergsteiger.pnummer*, nname: *Bergsteiger.nname* gebdat, gewicht, kalorien)
Sherpa	(pnummer: *Bergsteiger.pnummer*, nname: *Bergsteiger.nname*, maxlast)
Versorgungstrupp	(kennung, funkfreq)
liegt_an	(berg_name: *Route.berg_name*, land: *Route.land*, route_name: *Route.route_name*, ID: *Station.ID*)
folgt	(vorgaenger_id: *Station.ID*, nachfolger_id: *Station.ID*, bezeichnung: *Expedition.bezeichnung*, nummer: *Expedition.nummer*)
zustaendig	(sherpa_pnummer: *Sherpa.pnummer*, sherpa_nname: *Sherpa.nname*, kunde_pnummer: *Kunde.pnummer*, kunde_nname: *Kunde.nname*, bezeichnung: *Expedition.bezeichnung*, nummer: *Expedition.nummer*)
gehoert_zu	(sherpa_pnummer: *Sherpa.pnummer*, sherpa_nname: *Sherpa.nname*, kennung: *Versorgungstrupp.kennung*)
versorgt	(lager_id: *Basislager.ID*, kennung: *Versorgungstrupp.kennung* berg_name: *Saison.berg_name*, land: *Saison.land*, route_name: *Saison.route_name*, jahr: *Saison.jahr*, zfenster: *Saison.zfenster*)

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 (Auswerten) [0.4 Punkte]

Gegeben sind die folgenden vier Relationen.

Medikament		
name	**hersteller**	**wirkstoff**
Haarausfall	TU	Prüfung
Müdigkeit	Uni	Angabe
Erfolg	TU	Wissen

Packung		
name	**pgroesse**	**preis**
Haarausfall	2	19
Haarausfall	5	20
Müdigkeit	1	5
Erfolg	42	18, 70

Filiale	
adresse	**fleiter**
FH	Z.U. Ende
GM	G.E. Schafft
Trei	R.E. Used
Fav	F. Ertig
EI	A.B. Geschrieben

lagernd			
adresse	**name**	**pgroesse**	**anzahl**
Fav	Müdigkeit	1	5
GM	Haarausfall	5	7
FH	Haarausfall	2	13
Fav	Erfolg	42	1
Trei	Müdigkeit	1	8

Bestimmen Sie das Ergebnis der folgenden Anfragen über diesen Relationen.

(a)

$$\pi_{\text{wirkstoff,fleiter}}\Big(\pi_{\text{wirkstoff}}(\sigma_{\text{hersteller}=\text{“Uni”}}(\texttt{Medikament}))\times \big((\sigma_{\text{pgroesse}<2}(\texttt{Packung}) \bowtie \sigma_{\text{anzahl}>6}(\texttt{lagernd})) \rtimes \texttt{Filiale}\big) \Big)$$

Lösung:

q	
wirkstoff	**fleiter**
Angabe	R.E. Used

(b)

$$\{f \mid f \in \texttt{Filiale} \wedge \forall l \in \texttt{lagernd}(l.\text{adresse} = f.\text{adresse} \rightarrow \neg(\exists m \in \texttt{Medikament}(m.\text{name} = l.\text{name} \wedge m.\text{hersteller} \neq \text{“TU”})))\}$$

Lösung:

q	
adresse	**fleiter**
FH	Z.U. Ende
GM	G.E. Schafft
EI	A.B. Geschrieben

Aufgabe 5 (Äquivalenzen) [2 Punkte]

Gegeben sind folgende Paare q_i, q_j an Ausdrücken q_k der relationalen Algebra über den Relationenschemata $R(\underline{A}BC)$, $S(B\underline{D}E)$ und $T(\underline{A}DF)$.

- 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) $q_1\colon \big(\sigma_{A=D}(R \bowtie S)\big) \ltimes T$ und
$q_2\colon (R \bowtie_{R.A=S.D \wedge R.B=S.B} S) \ltimes \sigma_{A=D}(T)$

(b) $q_3\colon \pi_{AD}\Big(\big((\pi_B(R) - \pi_B(S)) \bowtie R\big) \bowtie S\Big) \cap \pi_{AD}(T)$ und
$q_4\colon \pi_{AD}(T) - \rho_{D\leftarrow B}\big(\pi_{AB}(R \bowtie S) \cup \rho_{B\leftarrow D}(\pi_{AD}(T))\big)$

(c) $q_5\colon R \bowtie \big(\pi_{AF}(T) \cap \rho_{A\leftarrow E, F\leftarrow D}(\pi_{DE}(S))\big)$ und
$q_6\colon (R \bowtie \pi_{AF}(T)) \cap (R \bowtie \rho_{A\leftarrow D, F\leftarrow E}(\pi_{DE}(S)))$

(d) $q_7\colon \rho_{A\leftarrow Q.A, B\leftarrow Q.B}\Big(\pi_{Q.A,Q.B}\big(\sigma_\theta(\rho_Q(\pi_A(R) \times \pi_B(R)) \times (R \times T))\big)\Big)$
mit $\theta = (Q.A = R.A \wedge Q.B = R.B) \vee (Q.A = T.A \wedge Q.B = T.F)$ und
$q_8\colon \big(\pi_{AB}(R) \cup \rho_{B\leftarrow F}(\pi_{AF}(T))\big)$

Lösung:

Aufgabe (a) **Nein**, q_1 und q_2 **sind nicht** äquivalent.

Die Schemata der Ergebnisrelationen der beiden Abfragen unterscheiden sich! Das Schema von q_1 ist $q_1(A, B, C, D, E)$, während das Schema von q_2 ein zusätzliches Attribut enthält. Das Schema von q_2 ist $(A, R.B, C, S.B, D, E)$.

Der Grund dafür ist, dass q_1 den `natürlichen Verbund` (*natural join*) $R \bowtie S$ enthält. Bei diesem Operator scheinen gemeinsame Attribute in beiden Relationen nur einmal in der Ergebnisrelation auf. In q_2 hingegen wird der θ-join verwendet. Dessen Ergebnisschema ist ident mit dem Ergebnisschemas des Kreuzproduktes, d.h. die Anzahl der Attribute im Ergebnisschema entspricht der Summe der Attribute in den Ausgangsschemata, und es findet keine Projektion statt. Mit Ausnahme des zusätzlichen Attributes liefern die beiden Abfragen aber das selbe Ergebnis.

Gegenbeispiel

R		
<u>A</u>	**B**	**C**
1	2	1

S		
B	**<u>D</u>**	**E**
2	1	1

q_1				
A	**B**	**C**	**D**	**E**
1	2	1	1	1

q_2					
A	**R.B**	**C**	**S.B**	**D**	**E**
1	2	1	2	1	1

Aufgabe (b) **Ja**, q_3 und q_4 **sind** äquivalent.

Beide Abfragen liefern über allen möglichen Ausprägungen immer eine leere Relation als Ergebnis. Darüber hinaus stimmen die Schemata der beiden (leeren) Ergebnisrelationen überein.

Bei Abfrage q_3 liefert der Ausdruck

$$\Big(\big((\pi_B(R) - \pi_B(S)) \bowtie R\big) \bowtie S\Big)$$

immer eine leere Ergebnisrelation. Daher ist auch das Endergebnis immer leer (der Durchschnitt mit einer leeren Menge liefert immer eine leere Menge). Dass der gezeigte Teilausdruck immer ein leeres Ergebnis liefert kann wie folgt gesehen werden: In einem ersten Schritt liefert die Teilabfrage

$$\pi_B(R) - \pi_B(S)$$

nur jene B-Werte aus R zurück, welche nicht in irgendeinem Tupel in S als Wert für B auftreten. D.h. das Ergebnis von

$$\big((\pi_B(R) - \pi_B(S)) \bowtie R\big)$$

enthält dann genau jene Tupel von R, für welche es kein Tupel in S gibt welches den selben Wert auf Attribut B besitzt. Daher liefert der natürliche Verbund zwischen dieser Relation und S immer eine leere Relation zurück.

Im Fall von q_4 kommt das leere Ergebnis wie folgt zustande. Der Ausdruck

$$\rho_{D\leftarrow B}\big(\pi_{AB}(R \bowtie S) \cup \rho_{B\leftarrow D}(\pi_{AD}(T))\big)$$

auf der rechten Seite des Differenzoperators enthält die Vereinigung zweier Relationen, wobei eine dieser Relationen die Relation

$$\pi_{AD}(T)$$

ist, welche auch exakt jene Relation ist, welche auf der linken Seite des Differenzoperators steht. Nun wird zwar das Attribut D in B umbenannt, bevor diese Relationen mit

$$\pi_{AB}(R \bowtie S)$$

vereinigt wird, jedoch wird diese Umbenennung im nächsten Schritt wieder rückgängig gemacht. Daher enthält die Relation welche auf der rechten Seite des Differenzoperators steht jene Relation, welche auf der linken Seite des Operators steht. Im Ergebnis werden daher alle Tupel aus

$$\pi_{AD}(T)$$

entfernt, was in einer leeren Ergebnisrelation endet.

Aufgabe (c) **Nein**, q_5 und q_6 **sind nicht** äquivalent.

Würde man in q_5 den Teilausdruck

$$\rho_{A\leftarrow E,F\leftarrow D}(\pi_{DE}(S))$$

durch

$$\rho_{A\leftarrow D,F\leftarrow E}(\pi_{DE}(S))$$

ersetzen, dann wären q_5' (das Ergebnis dieser Ersetzung) und q_6 in der Tat äquivalent, da die Äquivalenz $U \bowtie (V \cap W) \equiv ((U \bowtie V) \cap (U \bowtie W))$ gegeben ist.

Einschub: Um dies zu sehen, betrachten wir zunächst ein Tupel $t = (t_A, t_B, t_C, t_F)$ im Ergebnis zu q_5' mit den Attributen $ABCF$. Dann muss es in R ein Tupel $\bar{t} = (t_A, t_B, t_C)$ geben, sowie in T ein Tupel $t' = (t_A, t_D, t_F)$ (für einen beliebigen Wert t_D) und in S ein Tupel $\hat{t} = (t_{B'}, t_D, t_F)$ (wiederum für einen beliebigen Wert $t_{B'}$). Für das Ergebnis von q_6 bedeutet dies, dass sowohl das Ergebnis von $R \bowtie \pi_{AF}(T)$ das Tupel $t = (t_A, t_B, t_C, t_F)$ enthält (t' projiziert auf AF wird mit $\bar{t}$ gejoint und ergibt t), als auch das Ergebnis von $R \bowtie \rho_{A\leftarrow D,F\leftarrow E}(\pi_{DE}(S))$ (als Ergebnis der joins von $\bar{t}$ mit der Projektion und Umbenennung von $\hat{t}$). D.h. jedes Tupel im Ergebnis von q_5' erscheint auch im Ergebnis von q_6. Es bleibt zu zeigen, dass auch jedes Tupel im Ergebnis von q_6 Teil des Ergebnisses von q_5' ist. Dazu betrachten wir nun ein Tupel $t = (t_A, t_B, t_C, t_F)$ im Ergebnis von q_6. Dieses Tupel muss sowohl im Ergebnis von als auch von $R \bowtie \pi_{AF}(T)$ $R \bowtie \rho_{A\leftarrow D,F\leftarrow E}(\pi_{DE}(S))$ vorkommen. Im ersten Fall folgt daraus wiederum die Existenz des Tupels $\bar{t}$ in der Ausprägung von R und t' in der Ausprägung von T. Aus dem zweiten Teilausdruck können wir wiederum die Existenz von $\bar{t}$ in R sowohl von $\hat{t}$ in S ableiten. Aus der Existenz dieser Tupel folgt dann wiederum dass t im Ergebnis zu q_5' auftaucht.
Einschub Ende.

Wir haben es jedoch mit q_5 und nicht mit q_5' zu tun. In diesem Fall werden die Attribute von S "vertauscht", d.h. während in q_6 das Attribut D von S dem Attribut A in T und R entspricht und das Attribut E in S dem Attribut F in T, wird in q_5 das Attribut D in S mit dem Attribut F in T verglichen, und das Attribut E in S mit dem Attribut A in R und T. Unterschiedliche Eregbnisse ergeben sich dadurch sobald S ein Tupel enthält welches in einem der beiden Fälle in $\pi_{AF}(T)$ auftaucht und mit R kompatibel ist, jedoch nicht wenn man die Werte von D und E vertauscht (oder umgekehrt, eine dieser beiden Eigenschaften nicht erfüllt, jedoch beide Eigenschaften erfüllt wenn die Werte von D und E vertauscht werden).

Gegenbeispiel

R		
<u>A</u>	**B**	**C**
2	1	1

S		
B	**<u>D</u>**	**E**
1	2	3

T		
<u>A</u>	**D**	**F**
2	1	3

q_5			
A	**B**	**C**	**F**

q_6			
A	**B**	**C**	**F**
2	1	1	3

Aufgabe (d) **Nein**, q_7 und q_8 **sind nicht** äquivalent.

Die Abfrage q_8 berechnet die Vereinigung der Tupel in $\pi_{AB}(R)$ und $\pi_{AF}(T)$, wobei das Attribut F von T in B umbenannt wird, damit die beiden Relationen kompatibel sind. D.h. es wird die Menge aller Paare (a, b) ausgegeben so dass es entweder ein Tupel (a, b, c) (für ein beliebiges c) in der Ausprägung von R gibt, oder ein Tupel (a, d, b) (wieder für ein beliebiges d) in T.

Ein einfacher Grund, weshalb q_7 nicht immer das selbe Ergebnis liefern kann wie q_8 ist, dass das Ergebnis von q_8 Werte enthält die entweder in R oder in T vorkommen. Eine Betrachtung von q_7 zeigt jedoch, dass das Ergebis dieser Abfrage immer nur Werte aus R enthalten kann: Das Ergebnis enthält die Attribute $Q.A$ und $Q.B$. Dabei enthält die Relation Q ausschließlich Paare (a, b) so dass es zwei (nicht notwendigerweise verschiedene) Tupel (a, b', c') und (a', b, c) in R gibt (dies ist auf Grund von $\rho_Q(\pi_A(R) \times \pi_B(R))$). Ein Gegenbeispiel lässt sich daher schon dadurch ableiten, dass T Werte enthält welche in R nicht vorkommen.

Die Abfrage q_7 bildet also in

$$\rho_Q(\pi_A(R) \times \pi_B(R))$$

sämtliche Paare (a, b), welche sich dadurch bilden lassen dass man einen Wert welchen das Attribut A in R annimmt mit einem Wert kombiniert welches das Attribut B in R annimmt. Die Teilabfrage

$$R \times T$$

wiederum erzeugt sämtliche Kombinationen aus Tupeln von R und T. D.h. für jedes Tupel t in

$$\pi_{AB}(R) \cup \rho_{B\leftarrow F}(\pi_{AF}(T))$$

gibt es nun mindestens ein Tupel t' in $R \times T$ so dass t entweder mit den Werten von $R.A$ und $R.B$ oder den Werten $T.A$ und $T.F$ in t' übereinstimmt. Würde

$$\rho_Q(\pi_A(R) \times \pi_B(R))$$

daher eine Übermenge von

$$\pi_{AB}(R) \cup \rho_{B\leftarrow F}(\pi_{AF}(T))$$

darstellen, wären q_7 und q_8 in der Tat äquivalent: die Selektion σ_θ behält nun genau jene Elemente von

$$\rho_Q(\pi_A(R) \times \pi_B(R))$$

welche entweder in $\pi_{AB}(R)$ oder in $\pi_{AF}(T)$ vorkommen, was genau die Vereinigung in q_8 beschreibt. Der Grund weshalb die beiden Abfragen dennoch nicht äquivalent sind liegt jedoch wie erwähnt darin, dass $\rho_Q(\pi_A(R) \times \pi_B(R))$ nicht alle möglichen Paare enthält, welche im Ergebnis der Vereinigung auftreten können. Würde man

$$\rho_Q(\pi_A(R) \times \pi_B(R))$$

z.B. durch

$$\rho_Q((\pi_A(R) \cup \pi_A(T)) \times (\pi_B(R) \cup \rho_{B\leftarrow F}(\pi_F(T)))$$

ersetzen, wäre die daraus resultierde Abfrage in der Tat äquivalent zu q_8.

Gegenbeispiel

R		
A	**B**	**C**
1	2	1

T		
A	**D**	**F**
a	b	c

q_7	
A	**B**
1	2

q_8	
A	**B**
1	2
a	c

Aufgabe 6 (Größenabschätzung) [2 Punkte]

Gegeben sind die Relationenschemata $R(\underline{A}B)$, $S(AB\underline{CD})$, und $T(AC\underline{E})$ 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 angegeben 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) $q_1\colon\ (\pi_A(R) \cup \pi_A(T)) \cup (\pi_A(R) \cap \pi_A(T))$ (mit $|R| = 4$ und $|T| = 7$)

(b) $q_2\colon\ \big((\sigma_{C=4 \land D=2}(S) \bowtie S) \rtimes T\big) \ltimes T$ (mit $|S| = 5$ und $|T| = 7$)

(c) $q_3\colon\ \rho_{F \leftarrow B}(R) \ltimes \sigma_{A=4 \land B=2}(S)$ (mit $|R| = 3$ und $|S| = 2$)

(d) $q_4\colon\ \pi_{AF}(\rho_{F \leftarrow A}(T) \times R) - \pi_{AF}(\sigma_{F>A}(\rho_{F \leftarrow A}(S) \bowtie R))$ (mit $|R| = 2$, $|S| = 3$ und $|T| = 3$)

Lösung:

Aufgabe (a) [Minimum: 4 | Maximum: 11]

Nachdem A der Schlüssel von R ist enthält die Teilabfrage $\pi_A(R)$ 4 verschiedene Tupel (da jedes der vier Tupel in R einen anderen Wert für A besitzen muss). Nachdem A in T kein Schlüssel ist kann $\pi_A(T)$ zwischen 1 (im Fall dass alle Tupel in T den selben Wert für A besitzen) un 7 Ergebnistupel besitzen. Für $\pi_A(R) \cup \pi_A(T)$ ergibt sich daher, dass das Ergebnis zwischen 4 (für den Fall dass das Tupel in $\pi_A(T)$ auch in $\pi_A(R)$ vorkommt) und 11 (vier von $\pi_A(R)$ und sieben davon verschiedene aus $\pi_A(T)$) Tupeln enthalten kann.

Das Ergebnis der Teilabfrage $\pi_A(R) \cap \pi_A(T)$ ist immer eine Teilmenge des Ergebnisses der Teilabfrage $\pi_A(R) \cup \pi_A(T)$, und verändert das Ergebnis daher nicht weiter.

Minimum: 4

R	
A	**B**
1	1
2	2
3	3
4	4

T		
A	**C**	**E**
1	1	1
1	1	2
1	1	3
1	1	4
1	1	5
1	1	6
1	1	7

Ergebnis
A
1
2
3
4

Maximum: 11

R	
<u>A</u>	**<u>B</u>**
1	1
2	2
3	3
4	4

T		
A	**C**	**<u>E</u>**
5	1	1
6	1	2
7	1	3
8	1	4
9	1	5
a	1	6
b	1	7

Ergebnis
A
1
2
3
4
5
6
7
8
9
a
b

Aufgabe (b) [Minimum: 7 | Maximum: 7]

Sowohl die obere als auch die untere Schranke ergeben sich aus folgenden Beobachtungen: Das Ergebnis der Unteranfrage

$$(\sigma_{C=4\wedge D=2}(S) \bowtie S) \rtimes T$$

ist entweder T oder eine Relation welche eine Teilmenge der Tupel in T enthält. Das Schema der Unteranfrage ist auf jeden Fall identisch mit dem Schema von T.

Bei sämtlichen Varianten des äußeren Joins werden, genauso wie beim Natural Join, Attribute welche in den Schemata beider Operanden auftreten nur einmal ins Ergebnisschema übernommen. Daher ist das Schema des Ergebnisses der Abfrage q_2 ident mit dem Schema von T.

Auf Grund dessen ist das Ergebnis von q_2 *immer* die Relation T: Nachdem das Ergebnis von

$$(\sigma_{C=4\wedge D=2}(S) \bowtie S) \rtimes T$$

nur Tupel von T enthalten kann findet jedes Tupel in T maximal einen Joinpartner. Gleichzeitig werden auch Tupel welche keine Joinpartner finden bei einem äßeren Join in die Ergebnisrelation übernommen. Dabei werden diese Tupel mittel `NULL`s auf das Ergebnisschema erweitert. Da das Ergebnisschema in unserem Fall aber ident ist mit dem Schema von T heißt dies, dass keine `NULL` ergänzt werden müssen, und jedes Tupel aus T ins Ergebnis übernommen wird: Entweder, weil das Tupel auch im Ergebnis der Unteranfrage auf der linken Seite das ⟖ vorkommt (und als Join Partner fungiert), oder weil es ohne Partner übernommen wird.

(Als Ergänzung eine Anmerkung wie das kleinste und größte Ergebnis der Untaranfrage

$$(\sigma_{C=4\wedge D=2}(S) \bowtie S) \rtimes T$$

zustande kommt – auch wenn dies wie gerade ausgeführt für das Ergebnis der Abfrage irrelevant ist. Das kleinste mögliche Ergebnis ist die leere Relation ist: Wenn es in S kein Tupel mit dem Wert 4 für das Attribut C und 2 für das Attribut D gibt, dann ist das Ergebnis der Selektion und dadurch auch das anschließende Ergebnis des natürlichen Verbundes mit S leer. Dadurch ist auch das Ergebnis von

$$(\sigma_{C=4\wedge D=2}(S) \bowtie S) \rtimes T$$

leer.

Für die obere Schranke beobachten wir dass das Ergebnis der Unteranfrage

$$(\sigma_{C=4 \wedge D=2}(S) \bowtie S) \rtimes T$$

entweder T oder eine Teilmenge von T ist.

Die obere Schranke von maximal sieben Tupeln ergibt sich aus der Tatsache, dass CD der Schlüssel von S ist. Daher kann es in S maximal ein Tupel mit dem Wert 4 für C und 2 für D geben. Der natürliche Verbund dieses Tupels mit S ergibt dann wiederum genau diese eine Tupel. Das größte Ergebnis des Semijoins

$$(\sigma_{C=4 \wedge D=2}(S) \bowtie S) \rtimes T$$

erhält man nun, wenn alle Tupel in T erhalten bleiben. Dies ist der Fall, wenn alle Tupel in T den selben Wert auf den gemeinsamen Attributen haben wie das Tupel im Ergebnis von

$$\sigma_{C=4 \wedge D=2}(S) \bowtie S$$

Die gemeinsamen Attribute von S und T sind A und C. Nachdem die Kombination AC in T kein Schlüssel ist, ist es möglich dass alle sieben in Tupel in T genau den Wert auf AC haben, welchen das eine Tupel aus S (mit $C = 4$ und $D = 2$) auf AC hat. Daraus ergeben sich dann die sieben Ergebnistupel, da der rechte äußere Verbund von T mit T wiederum T liefert.)

Minimum: 7

S			
A	**B**	**C**	**D**
1	1	0	0
1	1	3	2
1	1	2	2
1	1	1	2
1	1	5	2

T		
A	**C**	**E**
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6
1	4	7

Ergebnis		
A	**C**	**E**
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6
1	4	7

Maximum: 7

S			
A	**B**	**C**	**D**
1	1	4	2
1	1	3	2
1	1	2	2
1	1	1	2
1	1	5	2

T		
A	**C**	**E**
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6
1	4	7

Ergebnis		
A	**C**	**E**
1	4	1
1	4	2
1	4	3
1	4	4
1	4	5
1	4	6
1	4	7

Aufgabe (c) [Minimum: 3 | Maximum: 5]

Das Ergebnis von

$$\rho_{F \leftarrow B}(R)$$

enthält immer die drei Tupel aus R (nur unter einem anderen Schema). Dadurch enthält der volle äußere Verbund

$$\rho_{F\leftarrow B}(R) \bowtie \sigma_{A=4\wedge B=2}(S)$$

immer mindestens drei Tupel, da die drei Tupel aus $\rho_{F\leftarrow B}(R)$ auf jeden Fall im Ergebnis erhalten bleiben (evtl. mit `NULL`-Werten "aufgefüllt"). Das Gesamtergebnis enthält daher immer mindestens drei Tupel. Diese Größe kann auch tatsächlich erreicht werden, und das auf mehrere Arten. Eine Möglichkeit ist, dass

$$\sigma_{A=4\wedge B=2}(S)$$

eine leere Ergebnismenge zurückliefert. Eine andere Möglichkeit besteht darin dass diese Abfrage genau ein Tupel zurückliefert welches mit einem der drei Tupel auf der linken Seite "kompatibel" ist, also den selben Werte auf dem gemeinsamen Attribut A besitzt.

Für die obere Schranke stellen wir zuerst fest, dass das größtmögliche Ergebnis für

$$\sigma_{A=4\wedge B=2}(S)$$

zwei Tupel sind, da AB kein Schlüssel von S ist. Als nächstes betrachten wir daher, was das größtmögliche Ergebnis des äußeren Joins darstellt. Hier gilt es zwei Möglichkeiten zu überprüfen. Die erste Möglichkeit ein möglichst großes Ergebnis zu erreichen besteht darin, alle Tupel der linken Seite mit allen Tupeln auf der rechten Seite zu einem Ergebnistupel zu kombinieren. Das würde ein Ergebnis von sechs Tupeln erzeugen. Nachdem das gemeinsame Attribute über welches der Verbund ausgeführt wird jedoch ein Schlüssel in A ist, ist es nicht möglich alle Tupel der linken Seite mit allen Tupel der rechten Seite zu kombinieren, da dies voraussetzen würde dass alle Tupel den selben Wert auf A haben. Eine Alternative in diesem Fall ist es daher, ein Tupel der linken Seite mit beiden Tupeln auf der rechten Seite zu je einem Ergebnistupel zu vereinen (ergibt zwei Tupel), und dann die beiden weiteren Tupel jeweils mit `NULL`-Werten "aufzufüllen". Das ergibt in Summe vier Ergebnistupel.

Dies ist jedoch nicht das größtmögliche Ergebnis. Dieses erhält man durch die zweite Möglichkeit, welche darin besteht dass überhaupt keine Tupel der linken und rechten Seite den selben Wert für A haben. In diesem Fall enthält das Ergebnis alle Tupel der linken Seite, ergänzt um `NULL`-Werte für die Attribute BCD, sowie die beiden Tupel der rechten Seite ergänzt um `NULL`-Werte für das Attribut F. In Summe erhalten wir daher also fünf Tupel.

Minimum: 3

R

<u>A</u>	<u>B</u>
1	1
2	2
4	4

S

A	B	<u>C</u>	<u>D</u>
4	2	1	1
2	2	2	2

Ergebnis

A	F	B	C	D
1	1	NULL	NULL	NULL
2	2	NULL	NULL	NULL
4	4	2	1	1

Maximum: 5

R

<u>A</u>	<u>B</u>
1	1
2	2
3	3

S

A	B	<u>C</u>	<u>D</u>
4	2	1	1
4	2	2	2

Ergebnis

A	F	B	C	D
1	1	NULL	NULL	NULL
2	2	NULL	NULL	NULL
3	3	NULL	NULL	NULL
4	NULL	2	1	1
4	NULL	2	2	2

Aufgabe (d) [Minimum: 0 | Maximum: 6]

Für die untere Schranke von keinem Ergebnistupel reicht es, dass das Ergebnis der rechten Seite der Differenz alle Tupel des Ergebnisses der linken Seite enthält. Ein Fall in dem dies so ist wird weiter unten im Beispiel gezeigt.

Generell ergibt sich die Überlegung daraus, dass das kleinstögliche Ergebnis der linken Seite zwei Tupel enthalten muss. Nachdem A der Schlüssel von R ist, muss es in R genau zwei verschiedene Werte für A geben. D.h. selbst wenn alle Tupel den selben Wert für F hätten würden sich mindestens zwei verschiedene Tupel ergeben. (Das Ergebnis von $\rho_{F \leftarrow A}(T) \times R$) enthält auf jeden Fall immer sechs Tupel).

Nachdem das Ergebnis auf der rechten Seite der Differenz bis zu sechs Tupel enthalten kann (was der Fall ist, wenn alle Tupel in R und S den selben Wert für B besitzen, und die Werte für A in der Relation S größer sind als die Werte für A in der Relation R), und diese auch in der Tat die selben Werte auf den selben Attributen annehmen können wie die Tupel auf der linken Seite, ergibt sich eine untere Schranke einer leeren Ergebnismenge.

Die obere Schranke ergibt sich, wenn das Ergebnis des Ausdrucks auf der linken Seite der Differenz größtmöglich ist, und kein (oder möglichst wenig) Tupel davon im Ergebnis der rechten Seite aufscheint.

Wie bereits erwähnt enthält das Ergebnis von

$$\rho_{F \leftarrow A}(T) \times R$$

immer sechs Tupel. Nachdem in R keine zwei Tupel den selben Wert für A haben dürfen, und nichts dagegen spricht dass alle drei Tupel in T unterschiedliche Werte auf A haben, können sich auch all diese sechs Tupel auch auf AF unterscheiden, wodurch ein größtmögliches Ergebnis der linken Seite sechs Tupel enthält.

Gleichzeitig kann das Ergebnis des Ausdrucks auf der rechten Seite aus mehreren Gründen gar kein Tupel enthalten: entweder es gibt keine kompatiblen Tupel für den Join

$$\rho_{F \leftarrow A}(S) \bowtie R,$$

oder keines der Ergebnistupel erfüllt die Selektionsbedingung $F > A$. In diesen Fällen bleiben alle sechs Tupel im Ergebnis enthalten.

Minimum: 0

R

<u>A</u>	<u>B</u>
1	1
2	1

S

A	B	<u>C</u>	<u>D</u>
7	1	1	1
8	1	2	2
9	1	3	3

T

A	C	<u>E</u>
7	4	1
7	4	2
7	4	3

Ergebnis

A	F

Maximum: 6

R

<u>A</u>	<u>B</u>
1	1
2	2

S

A	B	<u>C</u>	<u>D</u>
0	1	0	0
7	7	7	7
8	8	8	8

T

A	C	<u>E</u>
3	3	3
4	4	4
5	5	5

Ergebnis

A	F
1	3
1	4
1	5
2	3
2	4
2	5

Aufgabe 7 (Primitive Operatoren) [1 Punkt]

(a) Drücken Sie den Operator ⟕ mit Hilfe der primitiven Operatoren der Relationalen Algebra aus.

Nehmen Sie dazu zwei Relationen R und S an, wobei das Schema von R die Attribute $(R_1, \dots, R_r, G_1, \dots, G_g)$ und das Schema von S die Attribute $(S_1, \dots, S_s, G_1, \dots, G_g)$ enthält. Drücken Sie $R ⟕ S$ mit Hilde der primitiven Operatoren aus.

Hinweis: Ihr Ausdruck darf neben den R, S und primitiven Operatoren auch noch konstante Relationen enthalten. Bedenken Sie jedoch, dass r, s und g keine Konstanten sind. Eine konstante Relation wäre z.B. $\{(1)\}$.

Lösung:
In einem ersten Schritt zerlegen wir den Operator ⟕ in zwei Komponenten: Eine Komponente (ein natrürlicher Verbund) sorgt dafür, dass alle Tupel welche einen Join-Partner finden mit diesen zu neuen Tupeln im Ergebnis verbunden werden.

Der zweite Teil wählt die verbliebenen Tupel aus (also all jene, welche keinen Join Partner finden), und verbindet sie mit der benötigten Anzahl an `NULL` Werten.

$$R ⟕ S = (R \bowtie S) \cup \big((\dots((R - \pi_{R_1,\dots,R_r,G_1,\dots G_g}(R \bowtie S)) \times \rho_{S_1 \leftarrow N}(N)) \times \dots \times \rho_{S_s \leftarrow N}(N))\big),$$

wobei N die konstante Relation mit dem Schema $N(N)$ und der Ausprägung $\{(\texttt{NULL})\}$ beschreibt.

Da es sich bei dem natürlichen Verbund $\bowtie$ nicht um eine Basisoperation handelt, muss diese nun ebenfalls mittels primitiven Operatoren ausgedrückt werden. D.h. die beiden Ausdrücke $R \bowtie S$ müssen jeweils durch den Ausdruck

$$\pi_{R_1,\dots,R_r,G_1,\dots,G_g,S_1,\dots,S_s}(\sigma_{\bigwedge_{i=1}^{g} G_i = H_i}(R \times \rho_{H_1 \leftarrow G_1,\dots,H_g \leftarrow G_g}(S)))$$

ersetzt werden.

(b) Übersetzen Sie die Abfrage $R \ltimes S$ für die Relationen R und S mit dem selben Schema wie in Aufgabe (a) sowohl in den Tupel- als auch in den Domänenkalkül.

Lösung:
Tupelkalkül:

$$\{t \mid t \in R \land \exists u \in S(\bigwedge_{i=1}^{g} t.G_i = u.G_i)\}$$

Domänenkalkül:

$$\begin{aligned}\{[u_1, \dots, u_r, v_1, \dots, v_g] \mid [u_1, \dots, u_r, v_1, \dots, v_g] \in R \land \\ \exists w_1, \dots, w_s([w_1, \dots, w_s, v_1, \dots, v_g] \in S)\}\end{aligned}$$

(c) Übersetzen Sie die folgende Abfrage über den Relationen R, S und T, wobei R und S wiederum das selbe Schema wie in den vorerhigen Aufgaben haben, und T das Schema $(T_1, \dots T_t, G_1, \dots G_g)$, zuert in eine Abfrage der Relationalen Algebra welche nur die primitiven Operatoren verwendet, sowie in den Tupel- und den Domänenkalkül.

$$\pi_{T_1,\dots,T_t,G_1,\dots G_g}(T \bowtie (\pi_{G_1,\dots,G_g}(R) - \pi_{G_1,\dots,G_g}(S)))$$

Lösung:
Primitive Operatoren:

$$\pi_{T.T_1,\ldots,T.T_t,T.G_1,\ldots T.G_g}\big(\sigma_{T.G_1=R.G_1\wedge\cdots\wedge T.G_g=R.G_g}(T \times \rho_R(\pi_{G_1,\ldots,G_g}(R) - \pi_{G_1,\ldots,G_g}(S)))\big)$$

Tupelkalkül:

$$\{w \mid w \in T \wedge \exists u \in R(\bigwedge_{i=1}^{g} u.g_i = w.g_i \wedge \neg(\exists v \in S(\bigwedge_{i=1}^{g} v.g_i = w.g_i)))\}$$

Domänenkalkül:

$$\begin{aligned}\{[w_1,\ldots,w_t,a_1,\ldots,a_g] \mid\ & [w_1,\ldots,w_t,a_1,\ldots,a_g] \in T \wedge \\ & \exists u_1,\ldots,u_r([u_1,\ldots,u_r,a_1,\ldots,a_g] \in R) \wedge \\ & \neg(\exists v_1,\ldots,v_s([v_1,\ldots,v_s,a_1,\ldots,a_g] \in S))\}\end{aligned}$$

Aufgabe 8 (Formalisieren von Anfragen) [3.6 Punkte]

Ein Kaffeehaus verwaltet Informationen über seine Verkäufe in einer Datenbank mit folgendem Schema (Primärschlüssel sind unterstrichen, Fremdschlüssel sind kursiv geschrieben).

`Barista`	(<u>`VName`</u>, <u>`NName`</u>, `GebDat`, `Gehalt`)
`Gelernt_von`	(<u>*`LehrerVName`*</u>*`: Barista.VName`*, <u>*`LehrerNName`*</u>*`: Barista.NName`*, <u>*`SchuelerVName`*</u>*`: Barista.VName`*, <u>*`SchuelerNName`*</u>*`: Barista.NName`*)
`Kunde`	(<u>`KNr`</u>, `Email`)
`Kaffee`	(<u>`Bezeichnung`</u>, `Beschreibung`, `Preis`, `Schwierigkeit`)
`Kauf`	(<u>`datum`</u>, <u>*`KNr`*</u>*`: Kunde.KNr`*, *`Kaffee: Kaffee.Bezeichnung`*, *`VName: Barista.VName`*, *`NName: Barista.NName`*, `Menge`, `Bewertung`)
`Bohnen`	(<u>`Name`</u>, <u>`Herkunftsland`</u>, `Aroma`, `Qualitaet`)
`enthaelt`	(<u>*`Kaffee`*</u>*`: Kaffee.Bezeichnung`*, <u>*`BohnenN`*</u>*`: Bohnen.Name`*, <u>*`BohnenL`*</u>*`: Bohnen.Herkunftsland`*, `Menge`)

(Sie dürfen im Folgenden gerne passende (eindeutige) Abkürzungen sowohl für die Relationen- als auch die Tabellennamen verwenden.)

Formulieren Sie die unten beschriebenen Abfragen jeweils sowohl in der **relationaler Algebra**, dem **Tupelkalkül** und dem **Domänenkalkül**.

Die mit ** gekennzeichneten **Aufgaben (a), (b), (c) und (h)** sind dabei *freiwillig* und geben **keine Punkte**. Aufgaben (a), (b) und (c) sind einfache Aufgaben zum Einstieg, falls Sie bislang wenig oder keine Erfahrung mit dem Formulieren von Datenbankabfragen haben, und lieber eine etwas flachere Schwierigkeitskurve hätten. Aufgabe (h) ist einfach eine weitere etwas komplexere Abfrage, falls Sie noch ein weiteres Beispiel zum Üben suchen. Die Musterlösungen zu allen freiwilligen Aufgaben finden Sie am Ende dieses Dokuments.

(a) ** Geben Sie alle Barista aus (VName, NName, GebDat) die mehr als 2500 verdienen.

Lösung:
Relationale Algebra:

$$\pi_{\text{VName,NName,GebDat}}(\sigma_{\text{Gehalt}>2500}(\texttt{Barista}))$$

Tupelkalkül:

$$\{[b.\text{VName}, b.\text{NName}, b.\text{GebDat}] \mid b \in \texttt{Barista} \wedge b.\text{Gehalt} > 2500\}$$

Domänenkalkül:

$$\{[vn, nn, geb] \mid \exists g([vn, nn, geb, g] \in \texttt{Barista} \wedge g > 2500)\}$$

(b) ** Geben Sie alle Barista aus (VName, NName, GebDat) die noch nie eine 5-Stern Bewertung (oder besser) erhalten haben. Gehen Sie davon aus, dass die Bewertung als natürliche Zahl gespeichert wird.

Lösung:
Relationale Algebra:

$$\pi_{\text{VName,NName,GebDat}}\big(\texttt{Barista} \ltimes \big(\pi_{\text{VName,NName}}(\texttt{Barista}) - \pi_{\text{VName,NName}}(\sigma_{\text{Bewertung}\geq 5}(\texttt{Kauf}))\big)\big)$$

Tupelkalkül:

$$\{[b.\text{VName}, b.\text{NName}, b.\text{GebDat}] \mid b \in \texttt{Barista} \wedge \neg(\exists k \in \texttt{Kauf}(k.\text{VName} = b.\text{VName} \wedge k.\text{NName} = b.\text{NName} \wedge k.\text{Bewertung} \geq 5))\}$$

Domänenkalkül:

$$\{[vn, nn, geb] \mid \exists g([vn, nn, geb, g] \in \texttt{Barista} \wedge \neg(\exists d, knr, kaf, m, b([d, knr, kaf, vn, nn, geb, m, b] \in \texttt{Kauf} \wedge b \geq 5)))\}$$

(c) ** Geben Sie für die Kundin mit der KNr 74656 die Liste aller Länder aus, aus denen die in ihren Käufen zwischen dem 01.10.2371 und dem 20.12.2371 enthaltenen Bohnen stammen. Sie dürfen dabei annehmen, dass die Relation `enthält` nur Einträge mit einem Wert größer als 0 für das Attribut `Menge` enthält; d.h. dass nur Einträge für Bohnen gespeichert werden, welche tatsächlich im Kaffee enthalten sind.

Lösung:
Relationale Algebra:

$$\pi_{\text{BohnenL}}(\sigma_{\text{Datum}\geq 01.10.2371 \wedge \text{Datum}\leq 20.12.2371 \wedge \text{KNr}=74656}(\rho_{\text{Anzahl}\leftarrow\text{Menge}}(\texttt{Kauf})) \bowtie \texttt{enthaelt})$$

Tupelkalkül:

$$\begin{aligned}\{[e.\text{BohnenL}] \mid e \in \texttt{enthaelt} \wedge \exists k \in \texttt{Kauf}(&k.\text{Kaffee} = e.\text{Kaffee} \wedge \\ &k.\text{KNr} = 74656 \wedge \\ &k.\text{datum} \geq 01.10.2371 \wedge \\ &k.\text{datum} \leq 20.12.2371)\}\end{aligned}$$

Domänenkalkül:

$$\begin{aligned}\{[l] \mid \exists k, b, m, d, vn, nn, m2, w(&[k, b, l, m] \in \texttt{enthaelt} \wedge \\ &[d, 74656, k, vn, nn, m2, w] \in \texttt{Kauf} \wedge \\ &d \geq 01.10.2371 \wedge \\ &d \leq 20.12.2371)\}\end{aligned}$$

(d) Geben Sie die Nachnamen alle Barista mit einem Gehalt von weniger als 1800 aus welche schon einmal eine 3-Stern Bewertung (oder besser) auf einen Kaffe der Schwierigkeit 5 erhalten haben. Gehen Sie davon aus, dass sowohl die Bewertung als auch die Schwierigkeit als natürliche Zahl gespeichert werden.

Lösung:
Relationale Algebra:

$$\begin{aligned}\pi_{\text{NName}}\big(&\sigma_{\text{Gehalt}<1800}(\texttt{Barista}) \bowtie \\ &\sigma_{\text{Schwierigkeit}=5 \wedge \text{Bewertung} \geq 3}\big(\rho_{\text{Kaffee} \leftarrow \text{Bezeichnung}}(\texttt{Kaffee}) \bowtie \texttt{Kauf}\big)\big)\end{aligned}$$

Tupelkalkül:

$$\begin{aligned}\{[\mathit{kauf}.\text{NName}] \mid \mathit{kauf} \in \texttt{Kauf} \wedge \mathit{kauf}.\text{Bewertung} \geq 3 \wedge \\ \exists k \in \texttt{Kaffee}(k.\text{Bezeichung} = \mathit{kauf}.\text{Kaffee} \wedge k.\text{Schwierigkeit} = 5 \wedge \\ \exists b \in \texttt{Barista}(&b.\text{VName} = \mathit{kauf}.\text{VName} \wedge \\ &b.\text{NName} = \mathit{kauf}.\text{NName} \wedge \\ &b.\text{Gehalt} < 1800))\}\end{aligned}$$

Domänenkalkül:

$$\begin{aligned}\{[nn] \mid \exists k, d, knr, vn, m, b, t, p, geb, g(& \\ &[k, t, p, 5] \in \texttt{Kaffee} \wedge \\ &[vn, nn, geb, g] \in \texttt{Barista} \wedge \\ &[d, knr, k, vn, nn, m, b] \in \texttt{Kauf} \wedge \\ &b \geq 3 \wedge g < 1800)\}\end{aligned}$$

(e) Geben Sie die Bezeichnung aller Kaffees aus, welche Bohnen aus allen in der Datenbank erwähnten Ländern enthalten.

Lösung:

Relationale Algebra:

$$\pi_{\text{Kaffee,BohnenL}}(\texttt{enthaelt}) \div \rho_{\text{BohnenL}\leftarrow\text{Herkunftsland}}(\pi_{\text{Herkunftsland}}(\texttt{Bohnen}))$$

Tupelkalkül:

$$\{[k.\text{Bezeichnung}] \mid k \in \texttt{Kaffee} \wedge \forall b \in \texttt{Bohnen}(\exists e \in \texttt{enthaelt}(e.\text{Kaffee} = k.\text{Bezeichnung} \wedge e.\text{BohnenL} = b.\text{Herkunftsland}))\}$$

Domänenkalkül:

$$\{[k] \mid \exists txt, pr, lvl\big([k, txt, pr, lvl] \in \texttt{Kaffee} \wedge \forall b, land, ar, qu([b, land, ar, qu] \in \texttt{Bohne} \rightarrow \exists bn, m([k, bn, land, m] \in \texttt{enthaelt}))\big)\}$$

(f) Geben Sie für alle Barista jene Länder (Sie können sich dabei auf die in der Datenbank vorkommenden Länder beschränken) aus, aus denen noch nie Bohnen in einem von der Barista verkauften Kaffee enthalten waren. (Geben Sie eine Tabelle (NName, Land) aus, wobei "NName" der Nachname der Barista und "Land" der Name des Landes ist.)

Lösung:
Relationale Algebra:

$$\pi_{\text{NName, BohnenL}}(\rho_{\text{BohnenL}\leftarrow\text{Herkunftsland}}(\pi_{\text{VName,NName,Herkunftsland}}(\texttt{Barista} \times \texttt{Bohnen})) - \pi_{\text{VName,NName,BohnenL}}(\rho_{\text{Anzahl}\leftarrow\text{Menge}}(\texttt{Kauf}) \bowtie \texttt{enthaelt}))$$

Tupelkalkül:

$$\{[ba.\text{NName}, bo.\texttt{Herkunftsland}] \mid ba \in \texttt{Barista} \wedge bo \in \texttt{Bohnen} \wedge \neg\big(\exists e \in \texttt{enthaelt}(e.\text{BohnenL} = bo.\text{Herkunftsland} \wedge \exists k \in \texttt{Kauf}(k.\text{VName} = ba.\text{VName} \wedge k.\text{NName} = ba.\text{NName} \wedge k.\text{Kaffee} = e.\text{Kaffee}))\big)\}$$

Domänenkalkül:

$$\{[nn, land] \mid \exists vn, geb, geh, bo, ar, qual([vn, nn, geb, geh] \in \texttt{Barista} \wedge [bo, land, ar, qual] \in \texttt{Bohnen} \wedge \neg(\exists dat, knr, kaff, me, bew, bme, lbo([dat, knr, kaff, vn, nn, me, bew] \in \texttt{Kauf} \wedge [kaff, lbo, land, bme] \in \texttt{enthaelt})))\}$$

(g) Geben Sie die Namen aller Bohnen aus, welche in irgendeinem Kaffee waren der am 1.10.2019 verkauft wurde und die in einem Kaffee der Schwierigkeit 3 oder höher enthalten sind.

Lösung:
Relationale Algebra:

$$\pi_{\text{BohnenN}}\big(\pi_{\text{BohnenN,BohnenL}}\big(\rho_{\text{Anzahl}\leftarrow\text{Menge}}\big(\sigma_{\texttt{Datum}=1.10.2019}(\texttt{Kauf})\big) \bowtie \texttt{enthaelt}\big)\cap \\ \pi_{\text{BohnenN,BohnenL}}\big(\sigma_{\text{Schwierigkeit}\geq 3}(\texttt{Kaffee}) \bowtie \texttt{enthaelt}\big)\big)$$

Tupelkalkül:

$$\begin{aligned}\{[b.\text{Name}] \mid b \in \texttt{Bohnen} \wedge \exists \mathit{kauf} \in \texttt{Kauf}(&\mathit{kauf}.\text{Datum} = 1.10.2019 \wedge \\ &\exists e \in \texttt{enthaelt}(e.\text{Kaffee} = \mathit{kauf}.\text{Kaffee} \wedge e.\text{BohnenN} = b.\text{BohnenN} \wedge \\ &\quad e.\text{BohnenL} = b.\text{BohnenL} \wedge \\ &\quad \exists \mathit{kaff} \in \texttt{Kaffee}(\mathit{kaff}.\text{Schwierigkeit} \geq 3 \wedge \\ &\quad\quad \exists \mathit{ent} \in \texttt{enthaelt}(\mathit{ent}.\text{Kaffee} = \mathit{kaff}.\text{Bezeichnung} \wedge \\ &\quad\quad\quad \mathit{ent}.\text{BohnenN} = b.\text{BohnenN} \wedge \\ &\quad\quad\quad \mathit{ent}.\text{BohnenL} = b.\text{BohnenL}))))\}\end{aligned}$$

Domänenkalkül:

$$\begin{aligned}\{[\mathit{bohne}] \mid \exists &\mathit{land}, \mathit{ar}, \mathit{qual}, \mathit{knr}, \mathit{kaffk}, \mathit{vn}, \mathit{nn}, \mathit{anz}, \mathit{bew}, \mathit{men1}, \mathit{men2}, \mathit{kaffs}, \mathit{txt}, \mathit{pr}, \mathit{sc}(\\ &[\mathit{bohne}, \mathit{land}, \mathit{ar}, \mathit{qual}] \in \texttt{Bohne} \wedge \\ &[\mathit{kaffk}, \mathit{bohne}, \mathit{land}, \mathit{men1}] \in \texttt{enthaelt} \wedge \\ &[1.10.2019, \mathit{knr}, \mathit{kaffk}, \mathit{vn}, \mathit{nn}, \mathit{anz}, \mathit{bew}] \in \texttt{Kauf} \wedge \\ &[\mathit{kaffs}, \mathit{bohne}, \mathit{land}, \mathit{men2}] \in \texttt{enthaelt} \wedge \\ &[\mathit{kaffs}, \mathit{txt}, \mathit{pr}, \mathit{sc}] \in \texttt{Kaffee} \wedge \\ &\mathit{sc} \geq 3)\}\end{aligned}$$

(h) ** Betrachten Sie jene Lehrenden von denen zumindest einE SchülerIn ein Gehalt von mindestens 5000 verdient. Geben Sie nun die Liste von Bewertungen für alle SchülerInnen dieser Lehrenden aus.

Lösung:
Relationale Algebra:

$$\pi_{\texttt{Kauf}.\text{VName,Bewertung}}\big(\sigma_{\phi}(\sigma_{\text{Gehalt}\geq 5000}(\texttt{Barista})\times\rho_{g1}(\texttt{Gelernt\_von})\times\rho_{g2}(\texttt{Gelernt\_von})\times\texttt{Kauf})\big)$$

wobei ϕ die folgende Bedingung darstellt:

$$\begin{aligned}\phi =&\texttt{g1}.\text{SchuelerVName} = \text{VName} \wedge \texttt{g1}.\text{SchuelerNName} = \text{NName} \wedge \\ &\texttt{g1}.\text{LehrerVName} = \texttt{g2}.\text{LehrerVName} \wedge \texttt{g1}.\text{LehrerNName} = \texttt{g2}.\text{LehrerNName} \wedge \\ &\texttt{g2}.\text{SchuelerVName} = \texttt{Kauf}.\text{VName} \wedge \texttt{g2}.\text{SchuelerNName} = \texttt{Kauf}.\text{NName}\end{aligned}$$

Tupelkalkül:

$$
\begin{aligned}
\{[s.\text{VName}, k.\text{Bewertung}] \mid\ & s \in \texttt{Gelernt\_von} \land k \in \texttt{Kauf} \land \\
& s.\text{SchuelerVN} = k.\text{VName} \land s.\text{SchuelerNN} = k.\text{NName} \land \\
& \exists l \in \texttt{Gelernt\_von}(\\
& \quad l.\text{LehrerVN} = s.\text{LehrerVN} \land l.\text{LehrerNN} = s.\text{LehrerNN} \land \\
& \quad \exists b \in \texttt{Barista}(\\
& \quad\quad b.\text{VN} = l.\text{SchuelerVN} \land \\
& \quad\quad b.\text{NN} = l.\text{SchuelerNN} \land \\
& \quad\quad b.\text{Gehalt} \geq 5000))\}
\end{aligned}
$$

Domänenkalkül:

$$
\begin{aligned}
\{[svn, bew] \mid \exists bbn, bnn, geb, geh([bvn, bnn, geb, geh] \in \texttt{Barista} \land geh \geq 5000 \land \\
\exists lvn, lnn([lvn, lnn, bvn, bnn] \in \texttt{Gelernt\_von} \land \\
\exists snn([lvn, lnn, svn, snn] \in \texttt{Gelernt\_von} \land \\
\exists dat, knr, kaff, men(\\
[dat, knr, kaff, svn, snn, men, bew] \in \texttt{Kauf})))\}
\end{aligned}
$$