



Übung zur Vorlesung *Grundlagen: Datenbanken im WS20/21*

Christoph Anneser, Josef Schmeißer, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2021/grundlagen/>

Blatt Nr. 03

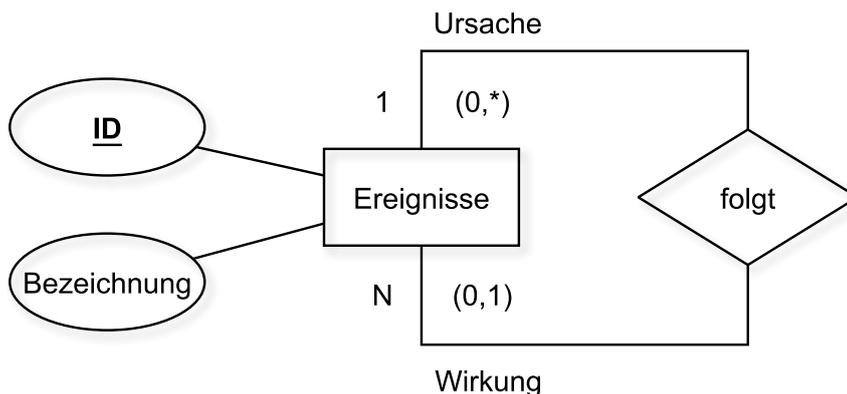
Tool zum Üben der relationalen Algebra: <https://dbis-uibk.github.io/relax/calc>.
Für das Unischema im Dropdown links „Kemper Datenbanksysteme“ auswählen.

HyPer-Webinterface zum Ausprobieren der SQL-Queries: <https://hyper-db.com/interface.html>

Hausaufgabe 1

- Erstellen Sie ein ER-Modell womit sich kausale Zusammenhänge darstellen lassen (Prinzip von Ursache und Wirkung). Nehmen Sie an, dass eine Ursache mehrere Wirkungen haben kann, und dass eine Wirkung auf maximal eine Ursache zurückzuführen ist. Geben Sie die Funktionalitäten an. Verwenden Sie die (min,max)-Notation.
- Übertragen Sie das ER-Modell in ein relationales Schema.
- Verfeinern Sie das relationale Schema durch Elimination von Relationen.
- Formulieren Sie folgende Anfrage in relationaler Algebra jeweils für die Schemas aus den Teilaufgaben b) und c): Finden Sie alle Auswirkungen des Ereignisses mit ID=10.

Lösung:



- Man kann von Ursache und Wirkung abstrahieren und beides als Ereignis betrachten. Ereignisse können als Ursache sowie als Wirkung auftreten. Für die Relation folgt gilt dann:

$folgt \subseteq Ursache : Ereignisse \times Wirkung : Ereignisse$

Eine Ausprägung könnte wie folgt aussehen:

<i>Ursache</i>		<i>Wirkung</i>	
<i>ID</i>	<i>Bezeichnung</i>	<i>ID</i>	<i>Bezeichnung</i>
0	Big Bang	1	Beginn der Zeit
	⋮		⋮
10	Fahrzeugpanne	11	Stau auf der A99
11	Stau auf der A99	12	Student verpasst Vorlesung
11	Stau auf der A99	13	Telekom-Techniker verspätet sich mit DSL-Anschluss
	⋮		⋮

(min,max)-Notation

Ursache (0,*)

Sprich: Ein Ereignis kommt in der Rolle “*Ursache*” minimal 0 mal und maximal beliebig oft vor.

Zum aktuellen Zeitpunkt ist noch nicht bekannt, welche Auswirkungen das Ereignis “*Student verpasst Vorlesung*” haben wird, es taucht also 0 mal als Ursache auf. Das Ereignis “*Stau auf der A99*” löst in der oben gezeigten Ausprägung zwei Folgeereignisse aus: “*Student verpasst Vorlesung*” und “*Telekom-Techniker verspätet sich mit DSL-Anschluss*”. Denkbar sind beliebig viele (weitere) Auswirkungen.

Wirkung (0,1)

Sprich: Ein Ereignis kommt in der Rolle “*Wirkung*” minimal 0 mal und maximal 1 mal vor.

Den Beginn einer Kausalkette bildet ein Ereignis, welches keine Wirkung einer Vorangegangenen Ursache ist. So ist der “*Big Bang*” ein Ereignis welches keine (bekannte) Ursache jedoch viele Auswirkungen hat. Er taucht also nur in der Rolle “*Ursache*” auf und somit 0 mal als Wirkung. Das Ereignis “*Stau auf der A99*” wurde durch die “*Fahrzeugpanne*” verursacht. Nach der Definition in der Aufgabenstellung gilt, dass ein Ereignis maximal eine Ursache hat. Es taucht somit auf der rechten Seite als Wirkung maximal 1 mal auf.

b)

$$\begin{aligned} \text{Ereignisse} &: \{ \underline{[ID : Integer, Bezeichnung : String]} \} \\ \text{folgt} &: \{ \underline{[UrsacheID : Integer, WirkungID : Integer]} \} \end{aligned}$$

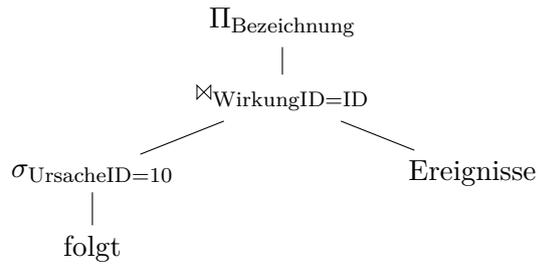
c)

$$\text{Ereignisse} : \{ \underline{[ID : Integer, Bezeichnung : String, UrsacheID : Integer]} \}$$

d) vor der Vereinfachung...

$$\prod_{\text{Bezeichnung}} (\sigma_{\text{UrsacheID}=10}(\text{folgt}) \bowtie_{\text{WirkungID}=ID} \text{Ereignisse})$$

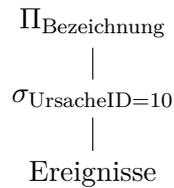
In Operatorbaumdarstellung:



nach der Vereinfachung...

$$\Pi_{\text{Bezeichnung}}(\sigma_{\text{UrsacheID}=10}(\text{Ereignisse}))$$

In Operatorbaumdarstellung:



Hausaufgabe 2

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in der relationalen Algebra, sowie im Tupel- und Domänenkalkül:

- Finden Sie die *Vorlesungen*, die keine Hörer haben.
- Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

Lösung:

- Finden Sie die *Vorlesungen*, die keine Hörer haben.

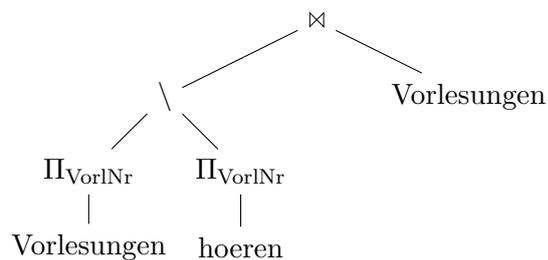
Formulierung in relationaler Algebra

$$(\Pi_{\text{VorlNr}}(\text{Vorlesungen}) \setminus \Pi_{\text{VorlNr}}(\text{hören})) \bowtie \text{Vorlesungen}$$

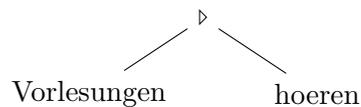
oder

$$\text{Vorlesungen} \triangleright \text{hören}$$

In Operatorbaumdarstellung:



oder



Formulierung im Tupelkalkül

$$\{v \mid v \in \text{Vorlesungen} \wedge \exists h \in \text{hören}(v.\text{VorlNr} = h.\text{VorlNr})\}$$

oder

$$\{v \mid v \in \text{Vorlesungen} \wedge \forall h \in \text{hören}(v.\text{VorlNr} \neq h.\text{VorlNr})\}$$

Formulierung im Domänenkalkül

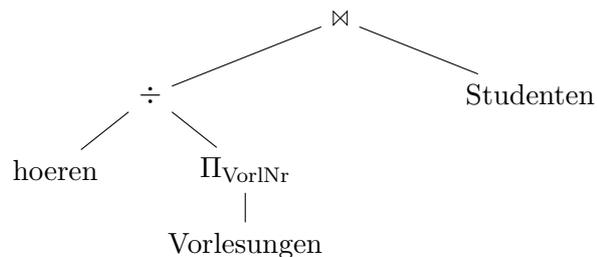
$$\{[v,t,s,g] \mid [v,t,s,g] \in \text{Vorlesungen} \wedge \exists m([m,v] \in \text{hören})\}$$

b) Finden Sie die *Studenten*, die alle *Vorlesungen* hören.

Formulierung in relationaler Algebra

$$(\text{hoeren} \div \Pi_{\text{VorlNr}}(\text{Vorlesungen})) \bowtie \text{Studenten}$$

In Operatorbaumdarstellung:



Formulierung im Tupelkalkül

$$\{s \mid s \in \text{Studenten} \wedge \forall v \in \text{Vorlesungen}(\exists h \in \text{hoeren}(v.\text{VorlNr} = h.\text{VorlNr} \wedge s.\text{MatrNr} = h.\text{MatrNr}))\}$$

Formulierung im Domänenkalkül

$$\{[m,n,s] \mid [m,n,s] \in \text{Studenten} \wedge \forall v,t,s,w,g([v,t,s,w,g] \in \text{Vorlesungen} \Rightarrow [m,v] \in \text{hören})\}$$

Hausaufgabe 3

Gegeben seien die beiden Relationen $R : \{[a_1, \dots, a_n]\}$ und $S : \{[b_1, \dots, b_m]\}$. Geben Sie die folgenden Ausdrücke im Tupel- und Domänenkalkül an:

a) $Q_1 := R \bowtie_{a_1=b_1} S$

b) $Q_2 := R \bowtie_{a_1=b_1} S$

c) $Q_3 := R \bowtie_{a_1=b_1} S$

$$d) Q_4 := R \triangleleft_{a_1=b_1} S$$

Lösung:

Bitte beachten Sie, dass in dieser Aufgabe ausschließlich allgemeine *Theta*joins ($\bowtie_{\Theta}, \bowtie_{\Theta}, \dots$) verwendet werden. Gemäß Definition werden somit alle Attribute der beiden Eingaberelationen in die Ausgabere Relation projiziert, einschließlich der Attribute, welche in der Joinbedingung enthalten sind. Lediglich bei *natürlichen* Joins, wo implizit eine Gleichheitsbedingung für alle gleichnamigen Attribute erfüllt sein muss, werden gleichnamige Attribute nicht doppelt in die Ausgabere Relation projiziert. Siehe hierzu auch Folie Kapitel 3, "Andere Join-Arten".

$$a) Q_1 := R \bowtie_{a_1=b_1} S$$

Formulierung im Tupelkalkül

$$Q_1 := \{[r.a_1, \dots, r.a_n, s.b_1, \dots, s.b_m] \mid r \in R \wedge s \in S \wedge r.a_1 = s.b_1\}$$

Da der Joinoperator Tupel aus verschiedenen Relationen verbindet, müssen für die Ergebnismenge neue Tupel mithilfe des Tupelkonstruktors konstruiert werden: $[\text{attribut}_1 : \text{wert}_1, \dots, \text{attribut}_n : \text{wert}_n]$. Die oben verwendete Tupelkonstruktion $[r.a_1, \dots]$ ist eine verkürzte Schreibweise für $[a_1 : r.a_1, \dots]$ und kann verwendet werden, wenn der Attributname im neuen Tupel unverändert bleibt.

Im Falle des Thetajoins kann auch die Tupelkonkatenation $t_1 \circ t_2$ verwendet werden:

$$Q_1 := \{r \circ s \mid r \in R \wedge s \in S \wedge r.a_1 = s.b_1\}$$

Formulierung im Domänenkalkül

$$Q_1 := \{[a_1, \dots, a_n, b_1, \dots, b_m] \mid [a_1, \dots, a_n] \in R \wedge [b_1, \dots, b_m] \in S \wedge a_1 = b_1\}$$

oder

$$Q_1 := \{[a_1, \dots, a_n, b_1 : a_1, b_2, \dots, b_m] \mid [a_1, \dots, a_n] \in R \wedge [a_1, b_2, \dots, b_m] \in S\}$$

$$b) Q_2 := R \bowtie_{a_1=b_1} S$$

Formulierung im Tupelkalkül

$$Q_2 := Q_1 \cup \{[r.a_1, \dots, r.a_n, b_1 : \text{null}, \dots, b_m : \text{null}] \mid r \in R \wedge \nexists s \in S (r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_2 := Q_1 \cup \{[a_1, \dots, a_n, b_1 : \text{null}, \dots, b_m : \text{null}] \mid [a_1, \dots, a_n] \in R \wedge \nexists c_2, \dots, c_m ([a_1, c_2, \dots, c_m] \in S)\}$$

$$c) Q_3 := R \bowtie_{a_1=b_1} S$$

Formulierung im Tupelkalkül

$$Q_3 := \{s \mid s \in S \wedge \exists r \in R(r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_3 := \{[b_1, \dots, b_m] \mid [b_1, \dots, b_m] \in S \wedge \\ \exists a_2, \dots, a_n([b_1, a_2, \dots, a_n] \in R)\}$$

d) $Q_4 := R \uparrow_{a_1=b_1} S$

Formulierung im Tupelkalkül

$$Q_4 := \{s \mid s \in S \wedge \nexists r \in R(r.a_1 = s.b_1)\}$$

Formulierung im Domänenkalkül

$$Q_4 := \{[b_1, \dots, b_m] \mid [b_1, \dots, b_m] \in S \wedge \\ \nexists a_2, \dots, a_n([b_1, a_2, \dots, a_n] \in R)\}$$

Hausaufgabe 4

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in SQL. Geben Sie alle Ergebnisse duplikatfrei aus.

- Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.
- Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.
- Finden Sie die *Assistenten* von *Professoren*, die den Studenten Carnap unterrichtet haben – z.B. als potentielle Betreuer seiner Bachelorarbeit.
- Geben Sie die Namen der *Professoren* an, die Theophrastos aus *Vorlesungen* kennt.
- Welche *Vorlesungen* werden von *Studenten* im Bachelorstudium (1. – 6. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.
- Bestimmen Sie für jede Vorlesung wie viele Studenten diese hören. Geben Sie auch Vorlesungen ohne Hörer aus. Sortieren Sie das Ergebnis absteigend nach Anzahl der Hörer.

Lösung:

- Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.

```
select distinct s.Name, s.MatrNr
from Studenten s, hoeren h, Vorlesungen v, Professoren p
where s.MatrNr = h.MatrNr
and h.VorlNr = v.VorlNr
and v.gelesenVon = p.PersNr
and p.Name = 'Sokrates';
```

- (b) Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.

```
select distinct s1.Name, s1.MatrNr
from Studenten s1, Studenten s2, hoeren h1, hoeren h2
where s1.MatrNr = h1.MatrNr
      and s1.MatrNr != s2.MatrNr
      and s2.MatrNr = h2.MatrNr
      and h1.VorlNr = h2.VorlNr
      and s2.Name = 'Fichte';
```

- (c) Finden Sie die *Assistenten* von *Professoren*, die den Studenten Carnap unterrichtet haben – z.B. als potentielle Betreuer seiner Bachelorarbeit.

```
select distinct a.Name, a.PersNr
from Assistenten a, Professoren p, Vorlesungen v, hoeren h,
  Studenten s
where a.Boss = p.PersNr
      and p.PersNr = v.gelesenVon
      and v.VorlNr = h.VorlNr
      and h.MatrNr = s.MatrNr
      and s.Name = 'Carnap';
```

- (d) Geben Sie die Namen der *Professoren* an, die Theophrastos aus *Vorlesungen* kennt.

```
select distinct p.PersNr, p.Name
from Professoren p, hoeren h, Vorlesungen v, Studenten s
where p.PersNr = v.gelesenVon
      and v.VorlNr = h.VorlNr
      and h.MatrNr = s.MatrNr
      and s.Name = 'Theophrastos';
```

- (e) Welche *Vorlesungen* werden von *Studenten* im Bachelorstudium (1. – 6. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.

```
select distinct v.Titel
from Vorlesungen v, hoeren h, Studenten s
where v.VorlNr = h.VorlNr
      and h.MatrNr = s.MatrNr
      and s.Semester between 1 and 6;
```

- (f) Bestimmen Sie für jede Vorlesung wie viele Studenten diese hören. Geben Sie auch Vorlesungen ohne Hörer aus. Sortieren Sie das Ergebnis absteigend nach Anzahl der Hörer.

```
select v.VorlNr, v.Titel, count(h.MatrNr) as hoerer
from
  Vorlesungen v left outer join
  hoeren h on (v.VorlNr = h.VorlNr)
group by v.VorlNr, v.Titel
order by hoerer desc;
```