TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

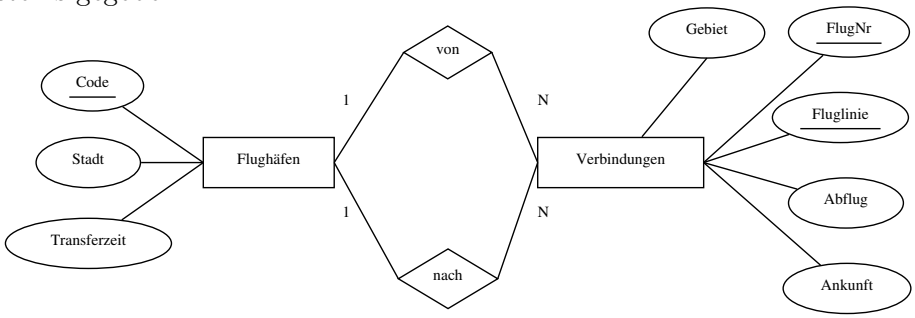
Übung zur Vorlesung *Grundlagen: Datenbanken* im WS18/19

Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
https://db.in.tum.de/teaching/ws1819/grundlagen/

Blatt Nr. 12

Hausaufgabe 1

Betrachten Sie die Anfrage *„Finde alle Flüge von New-York nach Sydney mit einmaligem Umsteigen“*. Dazu sei das nachfolgende (vereinfachte) ER-Diagramm eines Fluginformationssystems gegeben:



a) Geben Sie eine SQL-Query für die oben genannte Anfrage an.

b) Führen Sie die kanonische Übersetzung des SQL-Statements in die relationale Algebra durch.

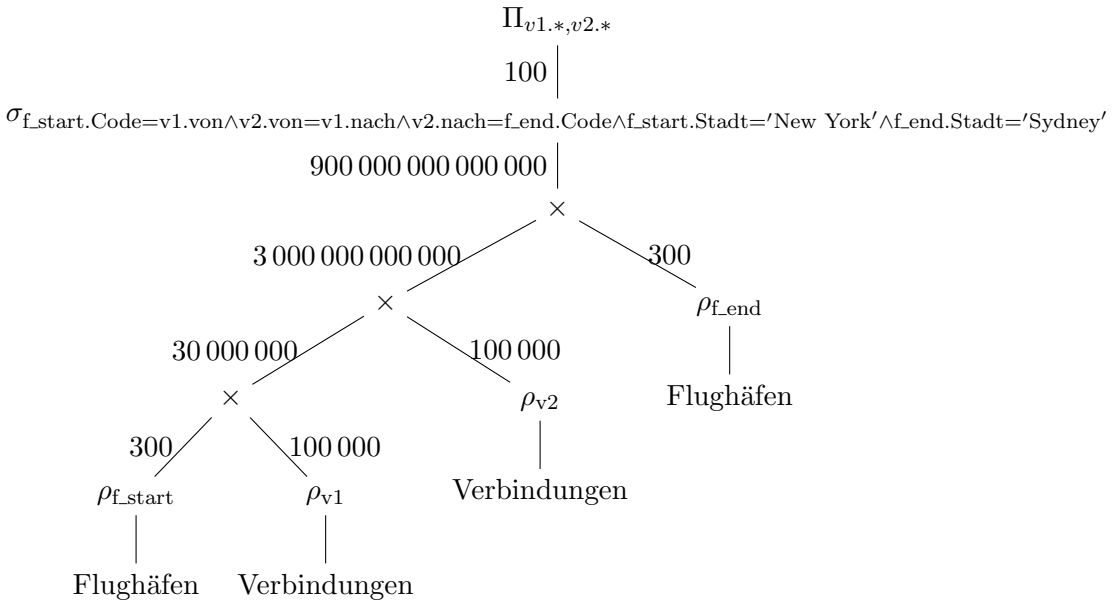
c) Schätzen Sie die Relationsgrößen sinnvoll ab (z.B. so wie in den Beispielen der Vorlesung) und transformieren Sie den kanonischen Operatorbaum aus Teilaufgabe b) zur optimalen Form. Wie haben sich die Kosten dabei geändert? (Kosten = Anzahl der Zwischenergebnistupel)

Lösung:

a) SQL-Anfrage:

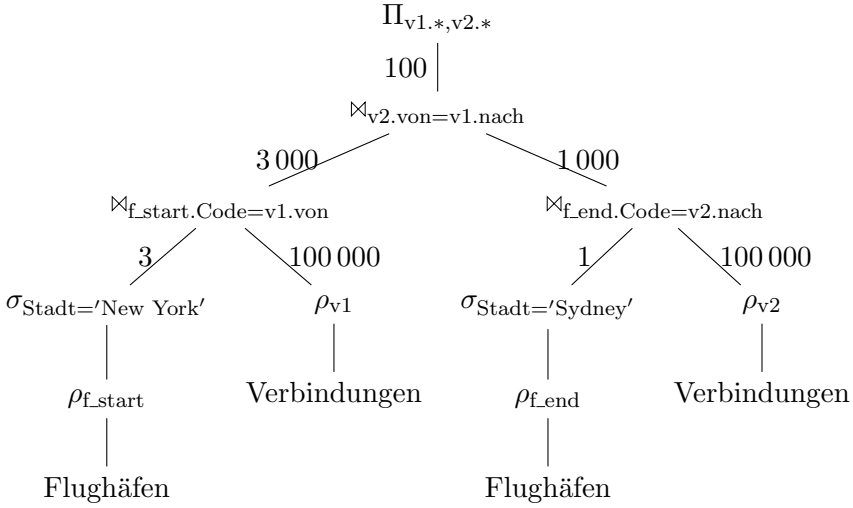
```
SELECT DISTINCT v1.*, v2.*
FROM Flughäfen f_start, Verbindungen v1, Verbindungen v2, Flughäfen f_end
WHERE f_start.Stadt = "New York"
  AND f_end.Stadt = "Sydney"
  AND v2.von = v1.nach
  AND v2.nach = f_end.Code
  AND f_start.Code = v1.von
```

b) **Kanonische Übersetzung:**



Kosten > 900 Billionen Tupel.

c) **Optimaler Ausführungsplan:**



Kosten ca. 205 000 Tupel

Hausaufgabe 2

a) Was ist ein Equi-Join?

b) Bei welchen Join-Prädikaten ($<, =, >$) kann man sinnvoll einen Hashjoin einsetzen?

c) Gegeben die Relation Profs = {<u>PersNr</u>, Name} und Raeume = {<u>PersNr</u>, RaumNr}.

1) Skizzieren Sie eine geschickte Möglichkeit, den Equi-Join Profs $\bowtie$ Raeume durchzuführen.

2) In welchem Fall wäre selbst ein Ausdruck wie

$$\text{Profs} \bowtie_{\text{Profs.Persnr} < \text{Raeume.PersNr}} \text{Raeume}$$

effizient auswertbar?

d) Der Student Maier hat einen Algorithmus gefunden, der den Ausdruck $A \times B$ in einer Laufzeit von $O(|A|)$ materialisiert. Was sagen Sie Herrn Maier?

Lösung:

a) Ein Equi-Join hat eine Äquivalenz als Joinbedingung, etwa die Gleichheit zweier Attribute.

b) Ein Hash Join bietet sich nur für Equi-Joins an, da lediglich ein Join-Partner mit gleichem Attributwert effizient auffindbar ist. Das Finden eines Partners, dessen Attributwert beispielsweise kleiner sein soll kann mittels Hashing i.A. nicht effizient bearbeitet werden.

c) 1) Offenbar ist das Joinattribut gerade der Primärschlüssel, womit von der Existenz eines Indexes ausgegangen werden kann. Somit bietet sich ein Index-basierter Join an, etwa dadurch, dass die eine Relation Element für Element abgearbeitet wird, während Joinpartner aus der anderen Relation mittels des Indexes gefunden werden.

2) Falls der Index sortiert ist, dies wäre etwa bei einem B-Baum der Fall. Dadurch liegen Joinpartner zumindest nacheinander im Index, anders als bei einer Implementierung des Indexes mittels Hash.

d) Dies ist mit Sicherheit nicht der Fall, da ein Algorithmus keine bessere Komplexitätsklasse haben kann als sein Ergebnis wächst. Mit anderen Worten, $A \times B$ hat eine Ergebnisgröße von $|A| \cdot |B|$ und dieses Ergebnis kann sicher nicht schneller als in $O(|A| \cdot |B|)$ materialisiert werden.

Hausaufgabe 3

Gegeben sei ein erweitertes Universitätsschema mit den folgenden zusätzlichen Relationen *StudentenGF* und *ProfessorenF*:

$$\begin{aligned} \text{StudentenGF} \quad &: \{[\underline{\text{MatrNr} : \text{integer}}, \text{Name} : \text{varchar}(20), \text{Semester} : \text{integer}, \\ &\quad \text{Geschlecht} : \text{char}, \text{Fakultaet} : \text{varchar}(20)]\} \\ \text{ProfessorenF} \quad &: \{[\underline{\text{PersNr} : \text{integer}}, \text{Name} : \text{varchar}(20), \text{Rang} : \text{char}(2), \\ &\quad \text{Raum} : \text{integer}, \text{Fakultaet} : \text{varchar}(20)]\} \end{aligned}$$

Die erweiterten Tabellen sind auch in der Webschnittstelle angelegt.

a) Ermitteln Sie den Männeranteil an den verschiedenen Fakultäten in SQL!

b) Ermitteln Sie in SQL die Studenten, die alle Vorlesungen ihrer Fakultät hören. Geben Sie zwei Lösungen an, höchstens eine davon darf auf Abzählen basieren.

Lösung:

a)

```
with
FakultaetTotal as (
  select Fakultaet, count(*) as Total
  from StudentenGF
  group by Fakultaet),
FakultaetMaenner as (
  select Fakultaet,count(*) as Maenner
  from StudentenGF
  where Geschlecht = 'M'
  group by Fakultaet)
select
    FakultaetTotal.Fakultaet,
    (case when Maenner is null then 0 else Maenner end)/(total*1.00)
from FakultaetTotal left outer join FakultaetMaenner
on FakultaetTotal.Fakultaet = FakultaetMaenner.Fakultaet
```

Wir müssen beachten, dass nicht jede Fakultät Männer beherbergt, weswegen diese Fakultäten (in der Standardausprägung im SQL Interface ist dies für Theologie der Fall) dann aus dem Ergebnis herausfallen würden. Aus diesem Grund verwenden wir einen `left outer join` um die Zahl der Männer und die Zahl der Studenten insgesamt zu verbinden, wodurch auch die Theologie-Fakultät im Ergebnis enthalten ist, auch wenn es keine Männer gibt.

Das `case`-Konstrukt dient in der oberen Anfrage dazu, den NULL Wert, die durch den Left Join für die Anzahl der Männer entstehen, wenn es keine Männer gibt, durch die Zahl 0 zu ersetzen. Alternativ ist dies möglich, indem man `coalesce(maenner,0) / (total*1.00)` verwendet.

Alternativ können wir das `case`-Konstrukt verwenden, um die Anzahl der Männer an den jeweiligen Fakultäten zu ermitteln. Den Männeranteil erhalten wir dann, indem wir die Anzahl der Männer durch die Gesamtanzahl der Studenten an der Fakultät teilen.

```
select Fakultaet,
 (sum(case when Geschlecht = 'M' then 1.00 else 0.00 end)) / count(*)
from StudentenGF
group by Fakultaet
```

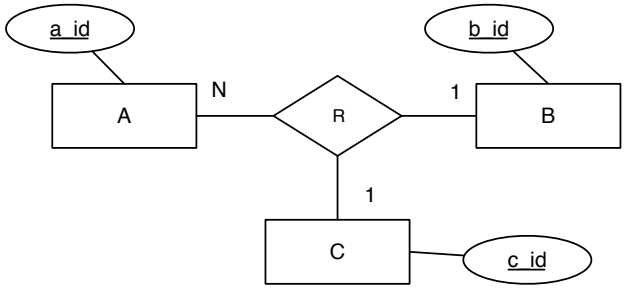
b) Wir fordern hier, dass es keine Vorlesung an der Fakultät des Studenten (d.h. von einem Professor der gleichen Fakultät gelesen) geben darf, die vom Studenten nicht gehört wird.

```
select s.*
from StudentenGF s
where not exists (select *
                  from Vorlesungen v, ProfessorenF p
                  where v.gelesenVon = p.PersNr
                    and p.Fakultaet = s.Fakultaet
                    and not exists
                       (select *
                        from hoeren h
                        where h.VorlNr = v.VorlNr
                          and h.MatrNr = s.MatrNr));
```

Alternativ:

```
select * from StudentenGF s
where
(select count(*)
from Vorlesungen v, ProfessorenF p
where v.gelesenVon = p.PersNr and p.Fakultaet = s.Fakultaet)
=
(select count(*)
from hoeren h, Vorlesungen v, ProfessorenF p
where
    h.MatrNr = s.MatrNr and
    h.VorlNr = v.VorlNr and
    p.PersNr = v.gelesenVon and
    p.Fakultaet = s.Fakultaet
)
```

Hausaufgabe 4



a) Welche partiellen Funktionen gelten?

b) Setzen Sie das ER Modell in Relationen um.

c) Bestimmen Sie einen Schlüssel für die Beziehung `R`, so dass möglichst viele Einschränkungen aus dem ER Modell auch in der Relation für die Beziehung modelliert werden.

d) Wieso ist ein Semantikverlust zunächst unvermeidbar? Welche Einschränkung müsste der Relation hinzugefügt werden, um die volle Semantik des ER Modells zu modellieren? [1]

Lösung:

Es gelten die partiellen Funktionen

$$A \times C \rightarrow B \tag{1}$$
$$A \times B \rightarrow C. \tag{2}$$

Aus dem Modell entstehen die folgenden Relationen:

$$A : \{[\underline{a\_id}]\} \tag{3}$$
$$B : \{[\underline{b\_id}]\} \tag{4}$$
$$C : \{[\underline{c\_id}]\} \tag{5}$$

sowie entweder

$$R : \{[\underline{a\_id}, b\_id, \underline{c\_id}]\} \tag{6}$$

oder

$$R : \{[\underline{a\_id}, \underline{b\_id}, c\_id]\} \tag{7}$$

Egal ob zur Modellierung der Beziehung die Variante in Gleichung 6 oder 7 verwendet wird, kann hier nur die Einschränkung einer der zwei im ER Modell geltenden partiellen Funktionen in die Modellierung mittels Relationen übertragen werden. Eine andere Wahl des Schlüssel außer den hier gezeigten ist entweder falsch, da sie zu starke Anforderungen stellt (etwa wenn lediglich ein Attribut als Schlüssel markiert wurde) oder semantisch schwächer (etwa die Markierung aller Attribute als Schlüssel).

Der Semantikverlust ist zunächst unvermeidbar, da lediglich eine Kombination von Attributen als Primärschlüssel der Beziehung markiert werden kann. Jede Kombination dieser

[1] Diese Teilaufgabe ist keine Hausaufgabe. Diskutieren Sie dies in der Übung!

Attribute darf daher nur einmal in der Ausprägung der Beziehungsrelation auftreten, was es erlaubt, die Einschränkung einer der zwei geltenden partiellen Funktionen in die Modellierung als Relation zu übertragen. Die andere geltende partielle Funktion kann nicht in die Relationenmodellierung übernommen werden. Hierfür wäre es notwendig, für eine weitere Menge von Attributen die Einschränkung modellieren zu können, dass jede Kombination dieser Attribute nur einmal in der Ausprägung der Relation auftreten kann. Dies ist beispielsweise in SQL mittels dem `unique`-Schlüsselwort möglich.