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

# Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe24

Alice Rey, Maximilian Bandle, Michael Jungmair (i3erdb@in.tum.de)
http://db.in.tum.de/teaching/ss24/impldb/

**Blatt Nr. 05**

**Hinweise Beachten Sie, dass dieses Blatt in KW21 und KW22 behandelt wird. Die Übungen werden zwischen 22.05. und 29.05. abgehalten. Übung 30 & 31 finden am 22.05 statt. Übung 32 findet am 29.05. statt.**

Die Datalogaufgaben können auf `https://souffle.db.in.tum.de/` getestet werden. Auf der Seite kann unter *examples* ein entsprechender Datensatz geladen werden. Die neuen IDB Regeln sollten am Ende der EDB definiert und dann im Query-Eingabefeld abgefragt werden.

Zusätzlich zu der in der Vorlesung vorgestellten Syntax hier noch eine Kurzübersicht der Vergleichsoperatoren: $X < Y, Y > X$ (kleiner, größer), $X <= Y, X >= Y$ (kleiner oder gleich, größer oder gleich), $X = Y, X! = Y$ (gleich, ungleich), $!pred(X, Y)$ (existiert nicht pred(X,Y)).

## Gruppenaufgabe 1

Ist folgendes Datalog-Programm stratifiziert?

$$\begin{aligned} p(X,Y) &:- q_1(Y,Z), \neg q_2(Z,X), q_3(X,P). \\ q_2(Z,X) &:- q_4(Z,Y), q_3(Y,X). \\ q_4(Z,Y) &:- p(Z,X), q_3(X,Y). \end{aligned}$$

Ist das Programm sicher – unter der Annahme, dass $p, q_1, q_2, q_3, q_4$ IDB- oder EDB-Prädikate sind?

## Hausaufgabe 2

Gehen Sie von folgender kombinierter Fragmentierung der in Abbildung 1 dargestellten Relation *Professoren* aus:

1. Zuerst erfolgt eine vertikale Fragmentierung in

$$\begin{aligned} \text{ProfVerw} &:= \Pi_{PersNr,Name,Gehalt,Steuerklasse}(\text{Professoren}) \\ \text{Profs} &:= \Pi_{PersNr,Name,Rang,Raum,Fakultt}(\text{Professoren}) \end{aligned}$$

2. Das Fragment Profs wird weiter horizontal fragmentiert in

$$\begin{aligned} \text{TheolProfs} &:= \sigma_{\text{Fakultät = 'Theologie'}}(\text{Profs}) \\ \text{PhysikProfs} &:= \sigma_{\text{Fakultät = 'Physik'}}(\text{Profs}) \\ \text{PhiloProfs} &:= \sigma_{\text{Fakultät = 'Philosophie'}}(\text{Profs}) \end{aligned}$$

Übersetzen Sie aufbauend auf dieser Fragmentierung die folgende SQL-Anfrage in die kanonische Form.

| Professoren | | | | | | |
|---|---|---|---|---|---|---|
| PersNr | Name | Rang | Raum | Fakultät | Gehalt | Steuerklasse |
| 2125 | Sokrates | C4 | 226 | Philosophie | 85000 | 1 |
| 2126 | Russel | C4 | 232 | Philosophie | 80000 | 3 |
| 2127 | Kopernikus | C3 | 310 | Physik | 65000 | 5 |
| 2133 | Popper | C3 | 52 | Philosophie | 68000 | 1 |
| 2134 | Augustinus | C3 | 309 | Theologie | 55000 | 5 |
| 2136 | Curie | C4 | 36 | Physik | 95000 | 3 |
| 2137 | Kant | C4 | 7 | Philosophie | 98000 | 1 |

Abbildung 1: Beispielausprägung der um drei Attribute erweiterten Relation *Professoren*

```
select Name, Gehalt Rang
from Professoren
where Gehalt > 80000;
```

Optimieren Sie diesen kanonischen Auswertungsplan durch Anwendung algebraischer Transformationsregeln (Äquivalenzen).

## Hausaufgabe 3

Gegeben sei folgende Relation `Klausur` mit Schlüssel `MatrNr`:

| MatrNr | Name | Note | Standort |
|---|---|---|---|
| 10101 | Philipp | 1,0 | München |
| 10102 | Magdalena | 1,0 | Garching |
| 10103 | Erik | 1,0 | Garching |
| 10104 | Josef | 1,0 | Garching |
| 10105 | Alex | 1,0 | Garching |
| 10106 | Maxmilian | 1,0 | München |

Für eine verteilte Datenbank soll die Tabelle geeignet fragmentiert werden. Ziel ist, Namen mit Standort der Studenten lokal und die Noten getrennt abzuspeichern.

1) Fragmentieren Sie die Relation geeignet *vertikal*.
   a) Geben Sie das Schema für die zwei resultierenden Relationen $KlausurV_1$ und $KlausurV_2$ an. Unterstreichen Sie jeweils den Primärschlüssel.
   b) Geben Sie in SQL-92 die zwei resultierenden Relationen `KlausurV1` und `KlausurV2` als Hilfstabellen (mittels `with`) an.
2) Die geeignetere der beiden resultierenden Relationen soll *horizontal* fragmentiert werden.
   a) Geben Sie das Prädikat der Selektion an, mit dem fragmentiert wird.
   b) Geben Sie in SQL-92 die zwei resultierenden Relationen `KlausurH1` und `KlausurH2` als Hilfstabellen (mittels `with`) an.
3) Schreiben Sie eine SQL-Abfrage, die die Ursprungsrelation aus den Teilrelationen zusammensetzt.

### Hausaufgabe 4

Für die Rekonstruierbarkeit der Originalrelation $R$ aus vertikalen Fragmenten $R_1, \ldots, R_n$ reicht es eigentlich, wenn Fragmente paarweise einen Schlüsselkandidaten enthalten. Illustrieren Sie, warum es also nicht notwendig ist, dass der Durchschnitt aller Fragmentschemata einen Schlüsselkandidaten enthält. Es muss also nicht unbedingt gelten

$$R_1 \cap \cdots \cap R_n \supseteq \kappa,$$

wobei $\kappa$ ein Schlüsselkandidat aus $R$ ist.

Geben Sie ein anschauliches Beispiel hierfür – am besten bezogen auf unsere Beispiel-Relation *Professoren*.

### Gruppenaufgabe 5

Gegeben sei folgende Faktenbasis, die einen direkten azyklischen Graphen (DAG) darstellt.

```
.decl kante(a: number, b: number)
kante(1,2).
kante(2,3).
kante(3,4).
kante(2,5).
kante(5,3).
```

1. Geben Sie in Datalog ein Prädikat `pfad(V,N,L)` an, dass alle möglichen Pfade von `V` nach `N` mit Länge `L` ausgibt.

   ```
   .decl pfad(von: number, nach: number, laenge: number)
   ```

2. Geben Sie nun das Prädikat `kuerzestePfade(V,N,L)` an, das pro Beginn `V` und Ziel `N` nur den kürzesten Pfad ausgibt.
3. Bestimmen Sie nun den längsten kürzesten Pfad `.decl laengsterkuerzesterPfad(L: number)`.
4. Erstellen Sie in SQL eine rekursive CTE `pfad(V,N,L)`, die die Länge aller Pfade im DAG ausgibt.
5. Basierend auf `pfad(V,N,L)`, geben Sie die Länge des längsten kürzesten Pfades aus.

### Hausaufgabe (wird nicht in der Übung besprochen)

Schreiben Sie zu dem U-Bahn-Netz-Beispiel auf der Datalog Seite (unter Examples) folgende Anfragen in Datalog:

1. Erstellen Sie den Stationsplan für den U-Bahnhof Fröttmanning, der alle Stationen, die ohne Umstieg erreichbar sind, auflistet.
2. Erstellen Sie für Garching-Forschungszentrum einen Plan, der alle erreichbaren Stationen, die minimale Anzahl an Umstiegen und Stops auflistet. Beschreiben Sie Ihren Ansatz ausführlich.
3. Gibt es zwischen allen Paaren von Stationen Verbidungen mit maximal einmaligem Umsteigen? Versuchen Sie Gegenbeispiele mittels Datalog zu finden.