



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken im SoSe21*

Maximilian {Bandle, Schüle}, Josef Schmeißer (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss21/impldb/>

Blatt Nr. 06

Hausaufgabe 1

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

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*

1. Zuerst erfolgt eine vertikale Fragmentierung in

$$\text{ProfVerw} := \Pi_{\text{PersNr, Name, Gehalt, Steuerklasse}}(\text{Professoren})$$
$$\text{Profs} := \Pi_{\text{PersNr, Name, Rang, Raum, Fakultät}}(\text{Professoren})$$

2. Das Fragment Profs wird weiter horizontal fragmentiert in

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

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

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

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

Hausaufgabe 2

Für die Rekonstruierbarkeit der Originalrelation R aus vertikalen Fragmenten R_1, \dots, 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

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

wobei κ ein Schlüsselkandidat aus R ist.

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

Hausaufgabe 3

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

<u>MatrNr</u>	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 *KlausurV₁* und *KlausurV₂* 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

Zeigen Sie, dass die *write-all/read-any* Methode zur Synchronisation replizierter Daten einen Spezialfall der *Quorum-Consensus*-Methode darstellt.

- Für welche Art von Workloads eignet sich dieses Verfahren besonders gut?
- Wie werden Stimmen zugeordnet um *write-all/read-any* zu simulieren?
- Wie müssen die Quoren Q_w und Q_r vergeben werden?

Hausaufgabe 5

Um Ausfallsicherheit zu garantieren ist ein Datenwert 'A' auf vier Rechnern verteilt. Jeder Rechner hält dabei eine vollständige Kopie von 'A'. Um Konsistenz zu garantieren wird das Quorum-Consensus-Verfahren eingesetzt. Dabei ist jedem Rechner ein Gewicht $w_i(A)$ wie folgt zugewiesen:

Rechner	Kopie	Gewicht
R_1	A_1	3
R_2	A_2	1
R_3	A_3	2
R_4	A_4	2

Das Lesequorum ist $Q_r(A) = 4$ und das Schreibquorum ist $Q_w(A) = 5$.

- Geben Sie **alle** Lesemöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.
- Geben Sie **alle** Schreibmöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.
- Zeigen Sie für dieses Beispiel, dass während eine Transaktion T_1 ein Schreibquorum auf A hält es für andere Transaktionen T_x nicht möglich ist ein Lesequorum für A zu bekommen.

Hausaufgabe 6

Gegeben seien die Tabellen **Studenten** und **Punkte** mit Schlüssel **MatrNr**, wobei **Punkte** auf einem separaten Rechner gespeichert ist. Es soll folgende Anfrage ausgeführt werden:

```
SELECT Name, Bonus FROM Student s, Punkte p WHERE s.MatrNr = p.MatrNr;
```

Der Datenbankadministrator entscheidet sich für einen Bloom-Filter zur Vorauswahl der Tupel. Auf **MatrNr** wird die Hash-Funktion $h(x) = x \bmod 5$ angewendet.

Studenten

<u>MatrNr</u>	Name	Hashwert
27	Magda	
4	Josef	
19	Erik	
95	Philipp	

Punkte

<u>MatrNr</u>	Bonus	Hashwert
27	ja	
16	nein	
25	nein	
95	ja	

- Berechnen Sie die Hash-Werte und tragen Sie diese in die obige Tabelle ein.
- Geben Sie den von **Studenten** zu übertragenden Bitvektor an.
- Geben Sie basierend auf dem Bitvektor an, welche Tupel aus **Punkte** übertragen werden.
- Geben Sie die Falsch-Positiv-Rate (false positive rate) an.
- Nehmen Sie an, dass jedes Tupel 8 Byte und der Bloomfilter selbst 1 Byte groß ist. Berechnen Sie zunächst die übertragenen Bytes ohne und mit Einsatz des Bloom-Filters.