

Mobile und Verteilte Datenbanken

WS 2008/2009

Blatt 6

Aufgabe 1: Abgeleitete horizontale Fragmentierung

Gegeben seien folgende Relationen:

ABT (ANR, ANAME, AORT, ABUDGET)
PERSONAL (PNR, PNAME, ANR, BERUF, GEHALT)
PROJEKT (PRONR, PRONAME, PROBUDGET)
PMITARBEIT (PNR, PRONR, DAUER).

Es handelt sich um das klassische Beispiel von Abteilungen, Projekten und ihren Mitarbeitern. Primärschlüssel sind unterstrichen; übereinstimmende Attributnamen kennzeichnen Fremdschlüssel-Primärschlüsselbeziehungen.

Abteilungen seien an drei Standorten vertreten: Kaiserslautern, Frankfurt und Leipzig.

Definieren Sie eine entsprechende horizontale Partitionierung von ABT sowie abgeleitete horizontale Partitionierungen für PERSONAL und PMITARBEIT. Welche Alternative besteht für PMITARBEIT zur Definition einer abgeleiteten horizontalen Fragmentierung? Wann wäre diese sinnvoll?

Aufgabe 2: Hybride Fragmentierung

Auf der horizontalen Fragmentierung von PERSONAL aus der vorherigen Aufgabe soll zusätzlich eine vertikale Fragmentierung durchgeführt werden, so daß das GEHALT-Attribut in einem eigenen Fragment geführt wird. Geben Sie die Spezifikation einer solchen vertikalen Fragmentierung an. Zeichnen Sie den Fragmentierungsbaum sowie den Operatorbaum zur Rekonstruktion der PERSONAL-Relation.

Aufgabe 3: Daten-Lokalisierung (abgeleitete Fragmentierung)

Gegeben seien folgende Relationen:

PERSONAL (PNR, PNAME, BERUF, GEHALT)
PROJEKT (PRONR, PRONAME, PROBUDGET)
PMITARBEIT (PNR, PRONR, DAUER).

Weiterhin sei folgende horizontale Fragmentierung von Relation PROJEKT gegeben:

PROJEKT₁ = $\sigma_{\text{PROBUDGET} < 100000}$ (PROJEKT)
PROJEKT₂ = $\sigma_{100000 \leq \text{PROBUDGET} \leq 800000}$ (PROJEKT)
PROJEKT₃ = $\sigma_{\text{PROBUDGET} > 800000}$ (PROJEKT).

Für PMITARBEIT liege eine abhängige horizontale Fragmentierung vor:

$PMITARBEIT_1 = PMITARBEIT \bowtie PROJEKT_1$
 $PMITARBEIT_2 = PMITARBEIT \bowtie PROJEKT_2$
 $PMITARBEIT_3 = PMITARBEIT \bowtie PROJEKT_3$.

Bestimmen Sie für die Anfrage

```
SELECT PNR
FROM PROJEKT PT, PMITARBEIT PM
WHERE DAUER > 10 AND PROBUDGET > 1000000
AND PT.PRONR=PM.PRONR
```

zunächst den initialen Fragment-Ausdruck. Nehmen Sie algebraische Optimierungen zur weitestgehenden Reduzierung des Ausdrucks vor.

Aufgabe 4: Ship-Whole vs. Semi-Join vs. Bitvektor-Join

Auf den Relationen PERSONAL und PMITARBEIT (siehe vorherige Aufgabe) sei folgende Join-Query zu bearbeiten:

```
SELECT P.PNR, PNAME, BERUF, PRONR, DAUER
FROM PERSONAL P, PMITARBEIT PM
WHERE P.PNR=PM.PNR AND P.GEHALT > 60000
```

Es gelte $Card(PERSONAL) = 1000$, $Card(PMITARBEIT) = 1500$; beide Relationen seien an verschiedenen Knoten gespeichert. Die Anfrage soll an einem dritten Knoten K initiiert werden; das Ergebnis ist dort auch auszugeben. Die Gehaltsbedingung soll von 20% der Angestellten erfüllt werden ($SF=0.2$); 25% der Angestellten sollen in keinem Projekt mitarbeiten.

Bestimmen Sie die Kommunikationskosten (#Nachrichten, #AW) für folgende Join-Strategien:

- Ship-Whole; Join-Berechnung an Knoten $K_{PMITARBEIT}$
- Ship-Whole; Join-Berechnung an Knoten K
- Semi-Join; Join-Bestimmung an Knoten $K_{PERSONAL}$
- Semi-Join; Join-Berechnung an Knoten K
- Bitvektor-Join; Join-Berechnung an Knoten K

Vor der Übertragung sollen alle anwendbaren Selektionen und Projektionen durchgeführt werden. Die Länge des Bitvektors soll 5 Attributwerten entsprechen; durch Anwendung des Bitvektors soll sich die zurückzuliefernde Tupelanzahl um 5% erhöhen.

Beispiellösungen aus:

Rahm, E.: Mehrrechner-Datenbanksysteme, Addison-Wesley 1994.