

Anfrageverarbeitung und Optimierung

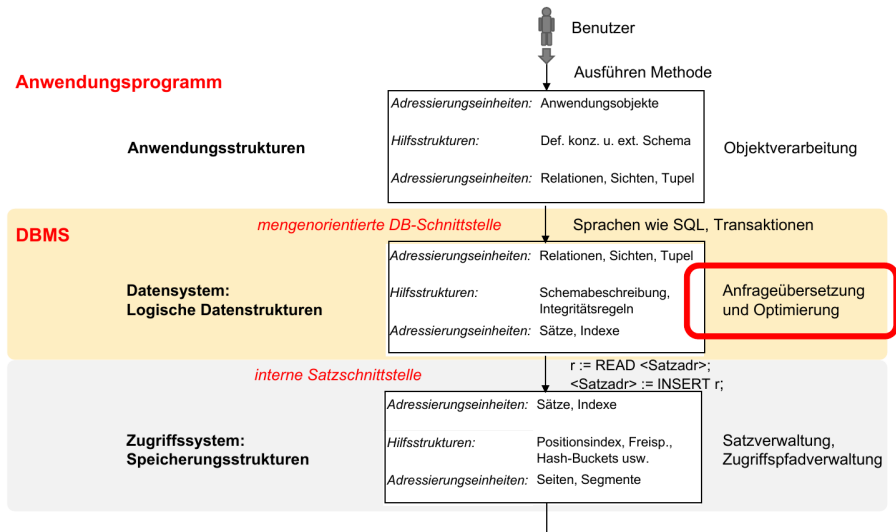
Tobias Bittner

2025-01-22

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Wiederholung - Worum gehts I



Wiederholung - Anfrageverarbeitung Ablauf I

Deklarative Anfrage

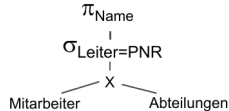
```
SELECT M.Name  
FROM Mitarbeiter M, Abteilungen A  
WHERE A.Leiter = M.PNR
```

Kanonische Übersetzung

Syntaxanalyse
semantische Analyse

Kanonische Normalform

Algebraischer Ausdruck (Auch: **Operatorbaum**)

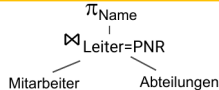


Anfrageoptimierung:

- **logische Optimierung** (Reihenfolge)
- **physische Optimierung** (Implementierung)

Optimierter Operatorbaum

↓
Ausführungsplan



QEP (Query Execution Plan)

Codeerzeugung / Ausführung

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung**
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 1: Fragen zur Vorlesung I

- a) Geben Sie die logische Abarbeitungsreihenfolge der Teile eines SQL-Statements an.
- b) Was ist die Aufgabe der Anfrageverarbeitung?
- c) Aus welchen Schritten besteht sie?
- d) Was macht die Optimierung so kompliziert?
- e) Wie kann man unter diesen Bedingungen das Ziel der Plangenerierung beschreiben?
- f) Erläutern Sie den Unterschied zwischen logischen Operatoren und Planoperatoren.
- g) Warum betrachtet man verschiedene Implementierungen von Operatoren?
- h) Welche Teilprobleme müssen bei der Übersetzung von logischen Operatoren in Planoperatoren gelöst werden?

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne**
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 2: Ausführungspläne I

Gegeben seien folgende Relationen:

Employee (num, fname, lname, addr, mgr[employee],
dep[department])

Department (num, name, mgr[employee])

Project (num, dep[department], mgr[employee])

Works_on (proj[project], empl[employee])

Setzen Sie folgende SQL-Anweisungen in nicht-optimierte Operatorgraphen um. Verwenden Sie dafür die Baum-Notation der Vorlesung.

Hausaufgabe 2: Ausführungspläne II

Logische Operatoren der Relationalenalgebra

- Selektion (σ): **SEL** (**R**, (*Bedingung*))
- Projektion (π): **PROJ** (**R**, **L**) mit $L = (A_1, \dots, A_k)$
- Kreuzprodukt ($\times$): **CROSS** (**R**, **S**)
- Verbund ($\bowtie$): **JOIN** (**R**, **S**, (*Bedingung*))
- Vereinigung ($\cup$): **UNION** (**R**, **S**)
- Schnittmenge ($\cap$): **INTERSECT** (**R**, **S**)
- Differenz ($-$): **EXCEPT** (**R**, **S**)
- Umbenennung (ρ):
RENAME (**R**, [**Rnew**,] ((**A1**, **A1new**), (**A2**, **A2new**), ...))
- Duplikat-Eliminierung (δ):
DUP-ELIM (**R**)
- Gruppierung (γ):
GROUP (**R**, **L**, **agg**) mit Gruppierungsattributen
 $L = (A_1, \dots, A_k)$ und
Aggregationen $\text{agg} =$
((**AGG**₁ (**attr**), **name**₁), (**AGG**₂ (**attr**), **name**₂), ...)

a) Alle MitarbeiterInnen und ihre Vorgesetzten.

```
SELECT e.fname, e.lname, s.fname, s.lname
FROM   employee e, employee s
WHERE  e.mgr = s.num
```

b) Alle MitarbeiterInnen der Forschungsabteilung.

Hausaufgabe 2: Ausführungspläne III

```
SELECT e.fname, e.lname, e.addr
FROM   employee e
JOIN    department d
ON      d.num = e.dep
WHERE   d.name = 'Research';
```

- e) Alle Abteilungen mit mehr als 10 Mitarbeitern.

```
SELECT    d.name
FROM      employee e
JOIN      department d
ON        d.num = e.dep
GROUP BY  d.num, d.name
HAVING    count(d.num) > 10
```

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung**
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 3: Optimierung I

Optimieren Sie die Ausführungspläne aus der vorherigen Aufgabe. Nutzen Sie dazu die Heuristiken aus der Vorlesung.

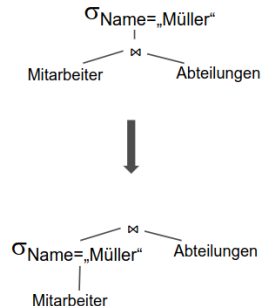
Hausaufgabe 3: Optimierung II

Regel 1: Predicate Pushdown (bei JOIN und Kreuzprodukt)

Seien R_1 und R_2 Relationen
 p ein Prädikat, das nur Bedingungen über den
Attributen von R_1 enthält, und $OP = \{x, \bowtie\}$

Dann gilt $\forall \oplus \in OP$:

$$\sigma_p(R_1 \oplus R_2) = \sigma_p(R_1) \oplus R_2$$



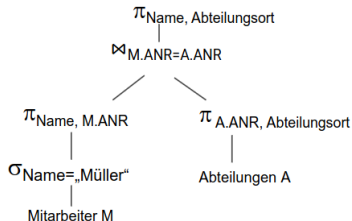
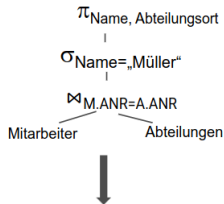
Hausaufgabe 3: Optimierung III

Regel 2: Predicate/Projection Pushdown (generell)

Selektionen und Projektionen sollten möglichst früh ausgeführt werden, damit *nicht verwendete Daten* nicht unnötig transportiert oder betrachtet werden müssen.

Das bedeutet: Verschiebe Selektionen und Projektionen so weit es geht nach unten, d.h. möglichst nah an die **Blätter** des Operatorbaumes

Im Beispiel werden zusätzliche Projektionen eingeführt, um nicht benötigte Attribute zu entfernen. Es ist darauf zu achten, daß notwendige Attribute nicht entfernt werden!
(Hier z.B.: ANR wird für den JOIN gebraucht)



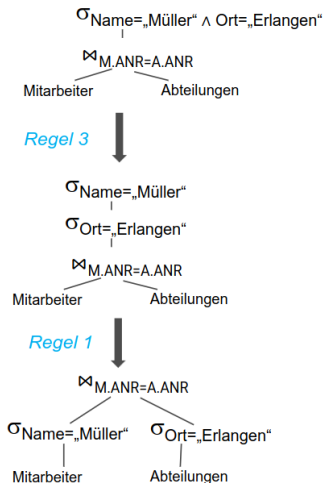
Hausaufgabe 3: Optimierung IV

Regel 3: Aufbrechen von Konjunktionen in Prädikaten

Wenn p und q Prädikate auf den Attributen einer Relation R sind, dann gilt:

$$\sigma_{p \wedge q}(R) = \sigma_p(\sigma_q(R))$$

Das Aufbrechen in hintereinander auszuführende separate Selektionen ermöglicht den unabhängigen Pushdown für p und q



Hausaufgabe 3: Optimierung V

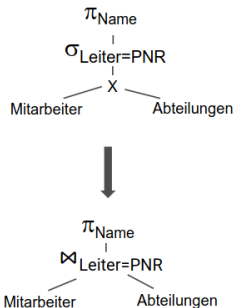
Regel 4: Fasse Selektion und Kreuzprodukt zusammen

R und S seien Relationen und p sei ein Verbund-Prädikat über den Attributen von R und S

dann gilt:

$$\sigma_p (R \times S) = R \bowtie_p S$$

Könnte man auch als Variante des Predicate Pushdown auffassen:
Filtere so früh wie möglich !

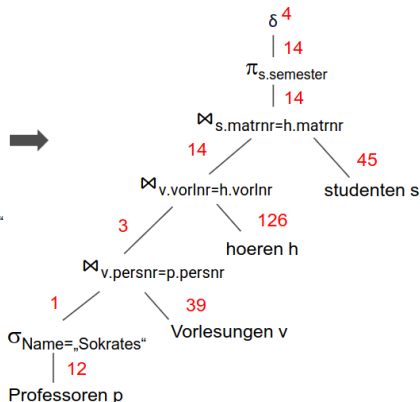
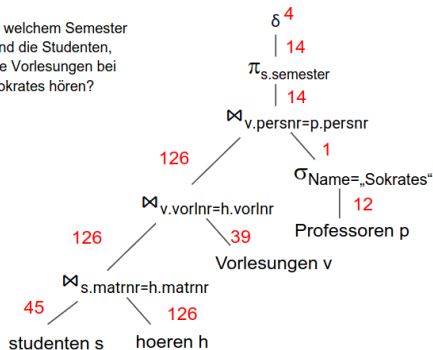


Hausaufgabe 3: Optimierung VI

Kommutative und assoziative Äquivalenzumformungen

Beispiel: Optimierung der Join-Reihenfolge

In welchem Semester sind die Studenten, die Vorlesungen bei Sokrates hören?



Regel 5: Kleine Relationen zuerst joinen

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra**
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 4: Relationale Algebra I

Gegeben seien die Relationen R und S. Gilt die folgende Umformung?
Begründen Sie Ihre Antwort!

$$\text{PROJ}(\text{INTERSECT}(R, S), E) = \text{INTERSECT}(\text{PROJ}(R, E), \text{PROJ}(S, E))$$

Umgangssprachlich: Dürfen Projektion und Schnittmenge vertauscht werden?

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung**
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 5: Kosten der Operatorausführung I

- a) Bei einer kostenbasierten Optimierung stellen die Kosten die Größe dar, die minimiert werden soll. Welche Werte könnte man bei der Optimierung von Datenbankankfragen in die Kosten einfließen lassen?
- b) Wir nehmen ein einfaches Kostenmodell an, das nur die Lese- und Schreibvorgänge auf dem Hintergrundspeicher berücksichtigt. Stellen Sie dafür Kostenformeln für die Planoperatoren Selektion, Projektion, Nested-Loop Join und Hash Join auf. Gehen Sie davon aus, dass die Eingaben vom Hintergrundspeicher gelesen werden müssen und die Ergebnisse direkt ausgegeben werden. $B(S)$ gibt die Anzahl der Blöcke an, aus denen die Relation S besteht.
- c) **Zusatzaufgabe:** Die Kosten des Sort-Merge Joins betragen $3(B(S) + B(T))$. Überlegen Sie sich, wie man diese Kosten erhält.

Hausaufgabe 5: Kosten der Operatorausführung II

- d) Pipelining bezeichnet das direkte Hintereinanderausführen mehrerer Operatoren, ohne die Zwischenergebnisse erst komplett zu erzeugen und auf den Hintergrundspeicher zu schreiben. Welche der aus der Vorlesung und Übung bekannten Planoperatoren eignen sich für Pipelining gut, welche nicht?
- e) Wie ändern sich die Kostenformeln aus Teilaufgabe b), wenn Pipelining eingesetzt wird? Verwenden Sie $C(S)$ für die Kosten zur Erzeugung der Zwischenergebnisrelation S .

- 1 Wiederholung
- 2 Hausaufgabe 1: Fragen zur Vorlesung
- 3 Hausaufgabe 2: Ausführungspläne
- 4 Hausaufgabe 3: Optimierung
- 5 Hausaufgabe 4: Relationale Algebra
- 6 Hausaufgabe 5: Kosten der Operatorausführung
- 7 Hausaufgabe 6: Optimierung

Hausaufgabe 6: Optimierung I

Gegeben sind folgende Tabellen:

$L(\underline{l}, m[M]), m[M] \text{ NOT NULL}$

$M(\underline{m}, d, e)$

Folgende Attribute sind indiziert: $L.l$, $L.m$, $M.m$

Folgende Operatorkosten sind zusätzlich zu den Kosten der vorherigen Aufgabe gegeben:

Implementierung	Kosten
TableScan (S)	$B(S)$
IndexScan(S)	$\alpha \cdot \lceil B(S) \cdot \text{Selektivitätsfaktor} \rceil$
SortMergeJoin(S, T)	$C(S) + C(T) + 2(B(S) + B(T))$

Hausaufgabe 6: Optimierung II

Beim Index Scan wurden hier die Kosten eines wahlfreien Blockzugriffs als $\alpha (= 50)$ -mal so hoch wie die eines sequentiellen Blockzugriffs angesetzt. Folgende Informationen sind jeweils im Systemkatalog abgelegt:

Bezeichnung	Beschreibung
$B(S)$	Anzahl der Blöcke, aus denen die Relation S besteht
$T(S)$	Anzahl Tupel, die in der Relation S enthalten sind
$bfr(S)$	Mittlerer Blockungsfaktor der Relation S ; ergibt sich aus $\frac{T(S)}{B(S)}$
$V(S.a)$	Anzahl der unterschiedlichen Werte, die für das Attribut a in S vorkommen

Hausaufgabe 6: Optimierung III

Folgende Werte sind bekannt:

$$B(L) = 5.000$$

$$B(M) = 1.000$$

$$T(L) = 100.000$$

$$T(M) = 10.000$$

Vereinfachende Annahmen:

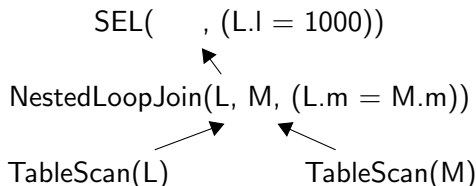
- Wir betrachten nur die Kosten für Ein-/Ausgabe vom Hintergrundspeicher.
- Alle Indizes werden komplett im Arbeitsspeicher gehalten.

Hausaufgabe 6: Optimierung IV

Gegeben seien Ausführungspläne für das folgende SQL-Statement. Bestimmen Sie jeweils die Ausführungskosten.

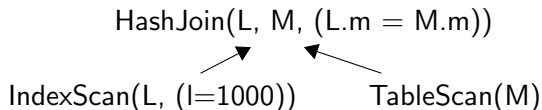
```
SELECT *  
FROM   L JOIN M ON (L.m = M.m)  
WHERE  L.l = 1000;
```

❶ Ausführungsplan 1:



Hausaufgabe 6: Optimierung V

ii) Ausführungsplan 2:



iii) Ginge es noch besser?