

Folien zum Lehrmodul

Abfrageverarbeitung und Optimierung

Lernziele:

- Möglichkeiten der Optimierung einschätzen können
- Unterschied zwischen algebraischer und interner Optimierung kennen
- ggf. selbst Abfragen optimieren können (Übung!)
- algebraische Ausdrücke in andere äquivalente umformen können
- grundlegende Implementierungen der relationalen Elementaroperationen kennen und einschätzen können

Inhaltsverzeichnis

1	Motivation	5
1.1	Grobablauf einer Abfrageverarbeitung	5
1.2	Optimierungsansätze	6
2	Optimierungskriterien	11
3	Algebraische Optimierung	13
3.1	Äquivalente algebraische Ausdrücke	13
3.1.1	Kommutativgesetze	15
3.1.2	Assoziativgesetze	16
3.1.3	Auflösung komplexer Selektionsbedingungen	17
3.1.4	Distributivgesetze zwischen Selektion und Mengenoperatoren	18
3.1.5	Gesetze für die Projektion	19
3.1.6	Verbund und Selektion	22
3.1.7	Kreuzprodukt und Selektion	23
3.1.8	Verbund und Projektion	24

3.2	Optimierungsheuristiken	28
3.3	Optimierungsalgorithmen	32
4	Kostenschätzung	34
4.1	Schätzung der Kosten einer Selektion	36
4.2	Schätzung der Kosten einer Projektion	37
4.3	Kosten eines Kreuzprodukts	38
4.4	Schätzung der Kosten eines Verbunds	39
4.5	Zusammenfassung zur Kostenschätzung	46

1 Motivation

1.1 Grobablauf einer Abfrageverarbeitung

hier nur relationale (incl. objektrelationale) DBMS.

1. Umformung der textuellen Darstellung der Abfrage in eine interne Darstellung (analog zum Parser in einem Compiler) und Korrektheitsüberprüfung
2. Optimierung und Vergleich möglicher Ausführungspläne
3. Ausführung des “optimalen” Plans

1.2 Optimierungsansätze

hier Beschränkung auf relationale Algebra;
komplexere Sprachen: mehr Abfrageoperationen, gleiche prinzipielle Vorgehensweise

“Optimierung”: kein Optimum, nur wahrscheinliche Verbesserung

a. Optimierung der Elementaroperationen:

Beziehung Operationen der relationalen Algebra – Implementierungen: nicht 1:1

- 1 Operation - mehrere Implementierungen, z.B. abhängig davon, ob Index vorhanden
- Implementierungen kompletter “Teilausdrücke”,
Beispiel: $\pi_{...}(\sigma_B(...))$

b. Optimierung der Abarbeitung von Ausdrücken:

Beispiel:

$$\pi_{\text{Kundenname}}(\sigma_{\text{Datum}=24.12.2000}(\text{kunden} \bowtie \text{lieferungen}))$$
$$\pi_{\text{Kundenname}}(\sigma_{\text{Datum}=24.12.2000}(\text{lieferungen}) \bowtie \text{kunden})$$

Beide Ausdrücke sind **äquivalent**,

der zweite ist aber **effizienter**

hier: Ausdruck = (vereinfachter, abstrakter) Ausführungsplan

Hauptschritte bei der Optimierung:

1. algebraische Optimierung:

Ausdruck A vorgegeben \rightarrow zu A äquivalente Ausdrücke bilden, die effizienter ausführbar erscheinen.

unabhängig vom aktuellen Inhalt der Datenbank

2. interne Optimierung: für jeden Ausdruck mehrere alternative Ausführungspläne bilden und jeweils Kosten schätzen günstigsten Plan auswählen

abhängig vom aktuellen Inhalt der Datenbank

Ausführungsplan: eine konkrete Auswahl / Festlegung von

- alternativ verfügbaren *Implementierungen von Operationen*
hierbei Entscheidung über Einsatz von Indexen
- ggf. Pufferung von *Zwischenergebnissen* auf Platte oder im Hauptspeicher
- ggf. Bestimmung *gemeinsamer Unterausdrücke*

Kostenschätzung pro Ausführungsplan auf Basis von

- Merkmalen des aktuellen Datenbestands, insb. Größe von Relationen
- vorhandenen Indexen oder Sortierungen

2 Optimierungskriterien

Aufwände / Kostenarten:

- Plattenzugriffe:
 - Lesen der Primärdaten
 - Schreiben und Lesen von Zwischenergebnissen
- Platzbedarf (Platte oder Hauptspeicher) für
 - Zwischenergebnisse
 - temporäre Indexe, sonstige Hilfsdaten
- CPU-Belastung

Kostenmaß: mehrere Meßgrößen mit geeigneten Faktoren gewichten

→ **kostenbasierte Optimierung**

wichtige Basisschätzungen:

- Laufzeiten der Implementierungen der relationalen Grundoperationen ($O(\dots)$)
- Größe von erzeugten Relationen

Ausführungskosten sind nicht exakt prognostizierbar – Kosten der Kostenschätzung müssen gering bleiben

3 Algebraische Optimierung

3.1 Äquivalente algebraische Ausdrücke

Ziel der algebraischen Optimierung:

äquivalenten Ausdruck zu finden, der effizienter ausführbar ist

Ausdrücke A1 und A2 sind **äquivalent**,

\iff bei einem beliebigen Datenbankinhalt liefern beide Ausdrücke das gleiche Ergebnis.

Trennen:

- **Äquivalenzen:** sind symmetrisch, können in beiden Richtungen gelesen werden, haben meist eine “günstigere” Seite, d.h. naheliegende Transformationsrichtung
- **Transformationsheuristik:** legt Transformationsrichtung für einzelne Äquivalenzen fest
- **Optimierungsalgorithmus:** legt fest, welche Äquivalenzen in welcher Reihenfolge ausgenutzt werden

3.1.1 Kommutativgesetze

Vereinigung, Schnitt, Verbund und Kreuzprodukt sind kommutativ:

$$\mathbf{r \cup s = s \cup r}$$

$$\mathbf{r \cap s = s \cap r}$$

$$\mathbf{r \bowtie s = s \bowtie r}$$

$$\mathbf{r \bowtie_Q s = s \bowtie_Q r}$$

$$\mathbf{r \times s = s \times r}$$

3.1.2 Assoziativgesetze

Vereinigung, Schnitt, Verbund und Kreuzprodukt sind assoziativ:

$$\mathbf{r} \cup (\mathbf{s} \cup \mathbf{t}) = (\mathbf{r} \cup \mathbf{s}) \cup \mathbf{t}$$

$$\mathbf{r} \cap (\mathbf{s} \cap \mathbf{t}) = (\mathbf{r} \cap \mathbf{s}) \cap \mathbf{t}$$

$$\mathbf{r} \bowtie (\mathbf{s} \bowtie \mathbf{t}) = (\mathbf{r} \bowtie \mathbf{s}) \bowtie \mathbf{t}$$

$$\mathbf{r} \bowtie_{Q_1} (\mathbf{s} \bowtie_{Q_2} \mathbf{t}) = (\mathbf{r} \bowtie_{Q_1} \mathbf{s}) \bowtie_{Q_2} \mathbf{t}$$

$$\mathbf{r} \times (\mathbf{s} \times \mathbf{t}) = (\mathbf{r} \times \mathbf{s}) \times \mathbf{t}$$

3.1.3 Auflösung komplexer Selektionsbedingungen

Die Booleschen Operatoren \wedge und \vee zwischen Bedingungen B_1 und B_2 können wie folgt in Schachtelungen von Selektionen oder Mengenoperationen umgeformt werden:

$$\sigma_{B_1 \wedge B_2}(\mathbf{r}) = \sigma_{B_1}(\sigma_{B_2}(\mathbf{r}))$$

$$\sigma_{B_1 \vee B_2}(\mathbf{r}) = \sigma_{B_2}(\sigma_{B_1}(\mathbf{r}))$$

$$\sigma_{B_1 \wedge B_2}(\mathbf{r}) = \sigma_{B_1}(\mathbf{r}) \cap \sigma_{B_2}(\mathbf{r})$$

$$\sigma_{B_1 \vee B_2}(\mathbf{r}) = \sigma_{B_1}(\mathbf{r}) \cup \sigma_{B_2}(\mathbf{r})$$

3.1.4 Distributivgesetze zwischen Selektion und Mengenoperatoren

$$\sigma_B(\mathbf{r} \cup \mathbf{s}) = \sigma_B(\mathbf{r}) \cup \sigma_B(\mathbf{s})$$

$$\sigma_B(\mathbf{r} \cap \mathbf{s}) = \sigma_B(\mathbf{r}) \cap \sigma_B(\mathbf{s})$$

$$\sigma_B(\mathbf{r} - \mathbf{s}) = \sigma_B(\mathbf{r}) - \sigma_B(\mathbf{s}) = \sigma_B(\mathbf{r}) - \mathbf{s}$$

3.1.5 Gesetze für die Projektion

2 Projektionen:

Seien $U \subseteq V \subseteq R$. Dann gilt

$$\pi_U(\pi_V(\mathbf{r})) = \pi_U(\mathbf{r}).$$

(wenn $U \not\subseteq V$, dann i.a. Syntaxfehler)

Vertauschung von Projektion und Selektion:

Projektion π_V kann mit Selektion σ_B vertauscht werden, also

$$\sigma_B(\pi_V(\mathbf{r})) = \pi_V(\sigma_B(\mathbf{r}))$$

wenn die Selektionsbedingung B nur Attribute aus V enthält

Projektion und Vereinigung:

Eine Projektion kann mit dem Vereinigungsoperator (= UNION in SQL, nicht UNION ALL!) vertauscht werden:

$$\pi_Y(\mathbf{r} \cup \mathbf{s}) = \pi_Y(\mathbf{r}) \cup \pi_Y(\mathbf{s})$$

gilt nicht analog für Durchschnitt und Differenz!

Aufgabe: Beweisen Sie mithilfe eines Gegenbeispiels, daß die beiden folgenden Äquivalenzen *nicht gelten*:

$$\pi_Y(\mathbf{r} \cap \mathbf{s}) = \pi_Y(\mathbf{r}) \cap \pi_Y(\mathbf{s})$$

$$\pi_Y(\mathbf{r} - \mathbf{s}) = \pi_Y(\mathbf{r}) - \pi_Y(\mathbf{s})$$

3.1.6 Verbund und Selektion

Es seien

- r_1 bzw. r_2 Relationen des Typs R_1 bzw. R_2 ,
- B eine Sel.-Bedingung, die *nur Attribute aus R_1* beinhaltet,
- Q eine Verbundbedingung.

Dann gilt:

$$\begin{aligned}\sigma_B(r_1 \bowtie r_2) &= \sigma_B(r_1) \bowtie r_2 \\ \sigma_B(r_1 \bowtie_Q r_2) &= \sigma_B(r_1) \bowtie_Q r_2\end{aligned}$$

Beispiele: sei $R_1 = \{ A, B, C, D \}$, $R_2 = \{ C, D, E, F \}$

$$\begin{aligned}\sigma_{A=B}(r_1 \bowtie r_2) &= \sigma_{A=B}(r_1) \bowtie r_2 \\ \sigma_{A=E}(r_1 \bowtie r_2) &\neq \sigma_{A=E}(r_1) \bowtie r_2\end{aligned}$$

$\sigma_{A=E}(r_1)$ ergibt einen Syntaxfehler!

3.1.7 Kreuzprodukt und Selektion

Es seien

- r_1 bzw. r_2 Relationen des Typs R_1 bzw. R_2 ,
- B eine Sel.-Bedingung, die *nur Attribute aus R_1* beinhaltet,

Dann gilt:

$$\sigma_B(r_1 \times r_2) = \sigma_B(r_1) \times r_2$$

Ist B eine Verbundbedingung (beinhaltet Attribute aus R_1 und R_2), so gilt:

$$\sigma_B(r_1 \times r_2) = r_1 \bowtie_B r_2$$

3.1.8 Verbund und Projektion

Seien wieder

$$R1 = \{ A, B, C, D \}, R2 = \{ C, D, E, F \}.$$

Gilt jetzt

$$\pi_A(r1 \bowtie r2) = \pi_A(r1) \bowtie r2$$

Nein! Fehlerursache:

Verbundattribute von r_1 und r_2 : $R_1 \cap R_2 = \{C, D\}$

... sind im rechten Teil der Formel bei r_1 *wegprojiziert* worden!

Der Verbund wird zum *Kreuzprodukt*!

korrektes Vorgehen:

gegeben ist Ausdruck der Form $\pi_U(r1 \bowtie_Q r2)$

Absicht: *Projektion am Verbund vorbei nach innen ziehen*

Bedingung: Verbundattribute müssen *innen* erhalten bleiben
(können erst nach der Verbundberechnung außen entfernt werden)

Regel: Man kann *vor der Verbundbildung* alle Attribute entfernen, die *nicht benötigt werden*:

- für die Verbundbildung oder
- für die außenstehende Projektion

korrekte Äquivalenz:

$$\pi_U(\mathbf{r1} \bowtie_Q \mathbf{r2}) = \pi_U(\pi_{U1}(\mathbf{r1}) \bowtie_Q \pi_{U2}(\mathbf{r2}))$$

darin sind:

- $\mathbf{r1}$ bzw. $\mathbf{r2}$ Relationen des Typs $\mathbf{R1}$ bzw. $\mathbf{R2}$
- $U \subseteq \mathbf{R1} \cup \mathbf{R2}$
- Q eine Verbundbedingung
- V die in Q auftretenden Verbundattribute;
(beim natürlichen Verbund: $V = \mathbf{R1} \cap \mathbf{R2}$)
- $U1 = (U \cup V) \cap \mathbf{R1}$
- $U2 = (U \cup V) \cap \mathbf{R2}$ ¹

¹im Skript sind die Mengen $U1$ und $U2$ für einen θ -Verbund ungünstig / falsch (zu groß) definiert

3.2 Optimierungsheuristiken

Problem: viele denkbare Sequenzen von Umformungsschritten

Optimierungsheuristiken: beschreiben solche Umformungsschritte, die mit sehr hoher Wahrscheinlichkeit die Effizienz verbessern

- legen "Richtung" fest, in der Äquivalenzen ausgenutzt werden
- gegeben teilweise Bedingungen an

Regel 1: Selektionen so früh wie möglich ausführen

Regel 2: eine äußere Selektion, deren Bedingung eine *Konjunktion* ist, in eine Schachtelung von Selektionen aufbrechen

Regel 3: Verbundbedingungen (= Selektionen, die nicht nach innen gezogen werden können) und Kreuzprodukte zu θ -Verbunden zusammenfassen

→ Tupel, die die Verbundbedingung nicht erfüllen, werden gar nicht erst erzeugt.

Regel 4: Bei einem Verbund von 3 und mehr Relationen:

- a) wenn möglich, 3- oder Mehrwegeverbund berechnen
- b) andernfalls Verbunde zuerst berechnen, die *voraussichtlich kleine Ergebnisse* erzeugen, z.B. wenn
 - die Verbundattribute Identifizierungsschlüssel in einer der beteiligten Relationen sind
 - mehrere Verbundattribute statt nur einem vorhanden sind
 - die beteiligten Relationen klein sind (interne Optimierung)

- Regel 5:** sofern sinnvoll zusätzliche Projektionen einfügen (z.B. bei Zwischenspeicherung von Verbundergebnissen)
Projektionen i.d.R. mit der vorhergehenden (also inneren) Operation zusammenfassen!
- Regel 6:** *Sofern an irgendeiner Stelle im Syntaxbaum Zwischenergebnisse gespeichert werden müssen, sollten alle äußeren Projektionen, soweit möglich, bis an diese Stelle verschoben werden.*

3.3 Optimierungsalgorithmen

Heuristiken \neq Algorithmus

Optimierungsalgorithmus legt fest:

- welche Äquivalenz an welcher Stelle ausgenutzt werden soll;
- wann das Verfahren abgebrochen wird;
(Aufwandsbeschränkung)

Ein einfacher Optimierungsalgorithmus:

1. Konjunktionen in Selektionen in Schachtelungen zerlegen
2. alle Selektionen so weit wie möglich nach innen verschieben
3. Selektionen und Kreuzprodukte zu Verbunden zusammenfassen
4. Reihenfolge von Verbunden gem. Größe der Relationen vertauschen (ab hier interne Optimierung!)
5. ggf. zusätzliche Projektionen bei Verbunden einfügen
6. zu puffernde Zwischenergebnisse verkleinern, indem Projektionen nach innen verschoben werden

sofern Regel an mehreren Stellen innerhalb des Ausdrucks anwendbar: jeweils an der ersten gefundenen Stelle zuerst anwenden

4 Kostenschätzung

Kostenarten:

1. Rechenaufwand (schon behandelt, s. Lehrmodul IRO)
Maßeinheit: Zahl der Schlüsselwertvergleiche
2. **Platzbedarf für Zwischen- und Endergebnisse**
Maßeinheit: Zahl der erzeugten Tupel
(relevant für die Kostenschätzung folgender Operationen)

Hauptparameter in beiden Fällen: Größe der Eingaberelationen, gemessen in Tupeln

Schwerpunkt i.f.: Platzbedarf / Größenschätzung

“**Kosten**” = i.f. Platzbedarf des (Zwischen- oder End-) Ergebnisses, gemessen in der Zahl der erzeugten Tupel

Ausgangsdaten (brauchen nicht völlig exakt zu sein):

- s_r der durchschnittliche Speicherplatzbedarf eines Tupels (incl. Hilfsdaten) der Relation r
- $|r|$ die Zahl der Tupel in der Relation r
- $V(A,r)$ die Zahl der verschiedenen Werte, die Attribut A in Relation r annimmt

4.1 Schätzung der Kosten einer Selektion

falls Selektionsbedingung: $\sigma_{A=a}$

1. A Identifizierungsschlüssel
→ $|\sigma_{A=a}(\dots)| \leq 1$.
2. es existiert ein Sekundärindex für A → Zahl der Treffer durch Auslesen eines einzigen Satzes im Sekundärindex bestimmen.
3. sonst: Gleichverteilungsannahme der Attributwerte →

$$|\sigma_{A=a}(\mathbf{r})| = \frac{|\mathbf{r}|}{V(\mathbf{A}, \mathbf{r})}$$

falls komplexe Selektionsbedingung: zerlegen

4.2 Schätzung der Kosten einer Projektion

ohne Duplikateliminierung (oder bei Projektion auf Superschlüssel):

$$|\pi_{\dots}(\mathbf{r})| = |\mathbf{r}|$$

ggf. zusätzlich Größe der Tupel heranziehen.

4.3 Kosten eines Kreuzprodukts

$$|\mathbf{r} \times \mathbf{s}| = |\mathbf{r}| * |\mathbf{s}|$$

4.4 Schätzung der Kosten eines Verbunds

Es seien

- r_1 bzw. r_2 Relationen des Typs R_1 bzw. R_2
- $V = R_1 \cap R_2$ die Verbundattribute.

1. Sonderfall: $V = \emptyset \rightarrow$ Kreuzprodukt

2. Sonderfall: V ist Superschlüssel für $R_2 \rightarrow$

$$| r_1 \bowtie r_2 | \leq | r_1 |$$

Falls zusätzlich A in r_1 **Fremdschlüssel** auf A in r_2 ist \rightarrow

$$| r_1 \bowtie r_2 | = | r_1 |$$

3. andernfalls, also V ist nicht Superschlüssel für $R1$ oder $R2$ (sehr untypisch!! wahrscheinlich ein Programmierfehler...); Annahmen:

- Gleichverteilungsannahme der Attributwerte
- Annahme i.f. zur Vereinfachung: $V = \{ A \}$

Grundidee:

- durchlaufe alle Tupel von $r1$
- schätze die Zahl der Verbundpartner

sei $t \in r1$; dessen Verbundpartner sind: $\sigma_{A=t[A]}(r2)$

Zahl der Verbundpartner *eines* Tupels:

$$|\sigma_{A=t[A]}(\mathbf{r2})| = \begin{cases} 0 & \text{falls } t[A] \text{ nicht in } \pi_A(\mathbf{r2}) \\ \frac{|\mathbf{r2}|}{V(A,\mathbf{r2})} & \text{sonst} \end{cases}$$

→ Gesamtzahl aller Verbundpartner *aller* Tupel von $\mathbf{r1}$:

$$|\mathbf{r1} \bowtie \mathbf{r2}| = |\mathbf{r1}| * \frac{|\mathbf{r2}|}{V(A,\mathbf{r2})} = |\mathbf{r1}| * |\mathbf{r2}| / V(A,\mathbf{r2})$$

vertausche Rollen von $\mathbf{r1}$ und $\mathbf{r2}$:

$$|\mathbf{r2} \bowtie \mathbf{r1}| = |\mathbf{r2}| * |\mathbf{r1}| / V(A,\mathbf{r1})$$

widersprüchliche Schätzungen, wenn $V(A,\mathbf{r1})$ und $V(A,\mathbf{r2})$ signifikant verschieden!

sei oBdA: $V(\mathbf{A}, \mathbf{r1}) < V(\mathbf{A}, \mathbf{r2})$,

d.h. in $\mathbf{r1}$ treten in Attribut \mathbf{A} viel weniger verschiedene Werte auf als in $\mathbf{r2}$

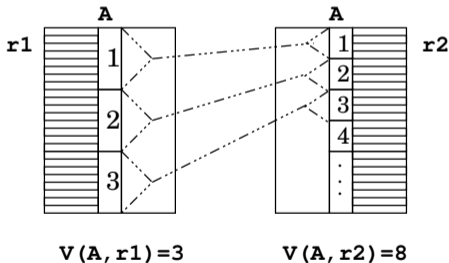
→ $\frac{|\mathbf{r1}| * |\mathbf{r2}|}{V(\mathbf{A}, \mathbf{r1})}$ ist größere Schätzung!

Annahme: $\pi_A(\mathbf{r1}) \subset \pi_A(\mathbf{r2})$

Beispiel:

$\pi_A(\mathbf{r1}) = \{ 1, 2, 3 \}$, Gleichverteilung

$\pi_A(\mathbf{r2}) = \{ 1, \dots, 8 \}$, Gleichverteilung



größere Schätzung unterstellt:

- r_2 wird durchlaufen
- pro Tupel werden jeweils $|r_1| / V(\mathbf{A}, r_1)$ Verbundpartner gefunden

ABER: Werte in $\pi_{\mathbf{A}}(r_2) - \pi_{\mathbf{A}}(r_1)$ finden keine Verbundpartner!

→ nur mit Wahrscheinlichkeit $V(\mathbf{A}, r_1) / V(\mathbf{A}, r_2)$ findet ein Tupel aus r_2 Verbundpartner (im Beispiel: 3/8)

→ Ergebnisgröße um diesen Faktor verkleinert

→ ergibt die kleinere Schätzung!

falls $\pi_A(\mathbf{r}_2) \not\subset \pi_A(\mathbf{r}_1) \rightarrow$ noch kleineres Ergebnis
u.U. Korrekturfaktor bestimmen (schwierig).

4.5 Zusammenfassung zur Kostenschätzung

in den meisten Fällen sind recht gute Schätzungen der Größe der erzeugten Relation möglich, wobei ausgenutzt werden:

- Informationen über die Schemata, insb. Identifizierungs- und Primärschlüssel
(Fremdschlüssel hingegen praktisch nicht!)
- Informationen über die Größe der Relationen
- Informationen über die Selektivität von einzelnen Attributen (Tabellenspalten), also die (ungefähre) Zahl der verschiedenen auftretenden Werte