

Technische Universität Kaiserslautern

Fachbereich Informatik

Lehrgebiet Datenverwaltungssysteme



Integriertes Seminar

Datenbanken und Informationssysteme

Wintersemester 2005/2006

Thema: Dependable Adaptive Information Systems

Adaptive Anfrageoptimierung

Andreas M. Weiner

a_weiner@informatik.uni-kl.de

17. Februar 2006

Motivation

„Imagine yourself standing in front of an exquisite buffet filled with numerous delicacies. Your goal is to try them all out, but you need to decide in what order. What exchange of tastes will maximize the overall pleasure of your palate? Although much less pleasurable and subjective, that is the type of problem that query optimizers are called to solve.“

Ioannidis, Y. E.: „Query Optimization“ In Tucker, A. B., ed: Computer Science Handbook, Second Edition, CRC Press (2004)

Überblick

1. Grundlagen der Anfrageoptimierung
2. Adaptive Anfrageoptimierung
3. Selbstwartende Histogramme
4. Adaptive Selektivitätsabschätzung
5. Reoptimierung während der Anfrageverarbeitung
6. LEO – Der lernende Optimierer
7. Fazit und Ausblick

1. Grundlagen der Anfrageoptimierung

Optimierung von Datenbankabfragen

SQL ist eine **deklarative** Anfragesprache

⇒ großer Spielraum für die Optimierung

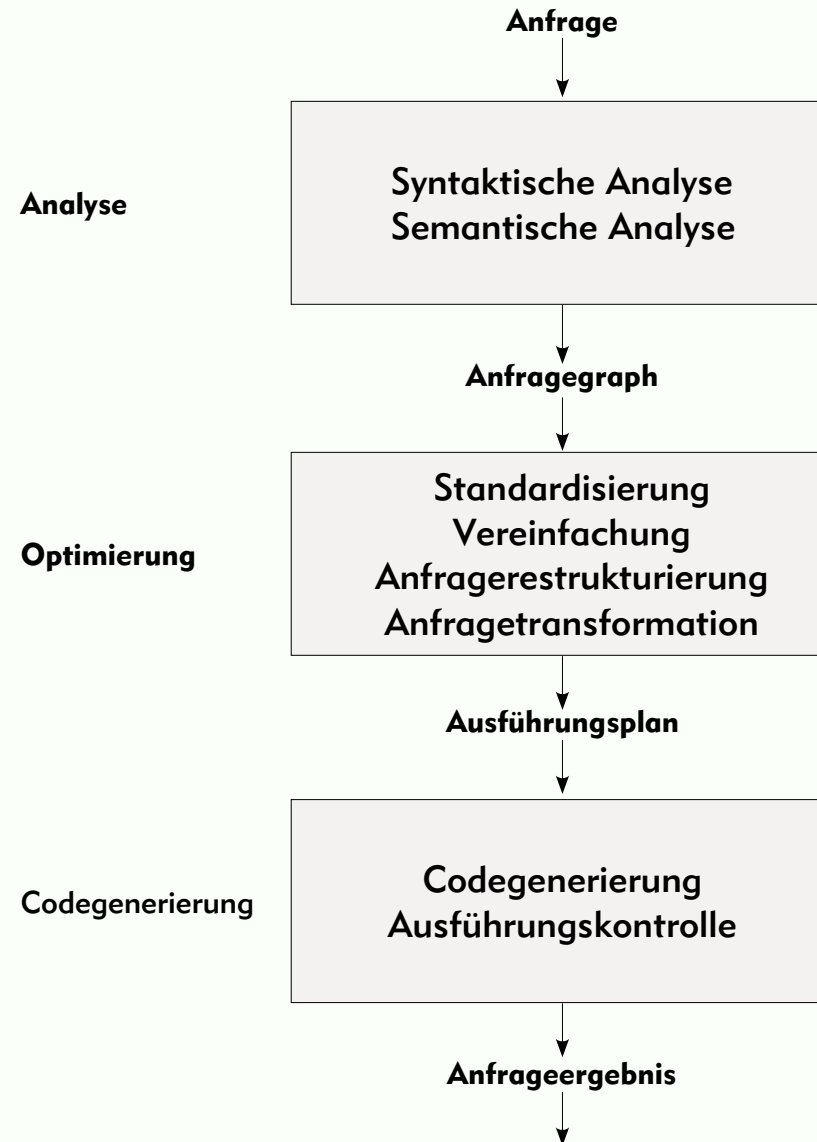
Spannungsverhältnis bei der Optimierung

Maximierung des Outputs bei gegebenen Ressourcen

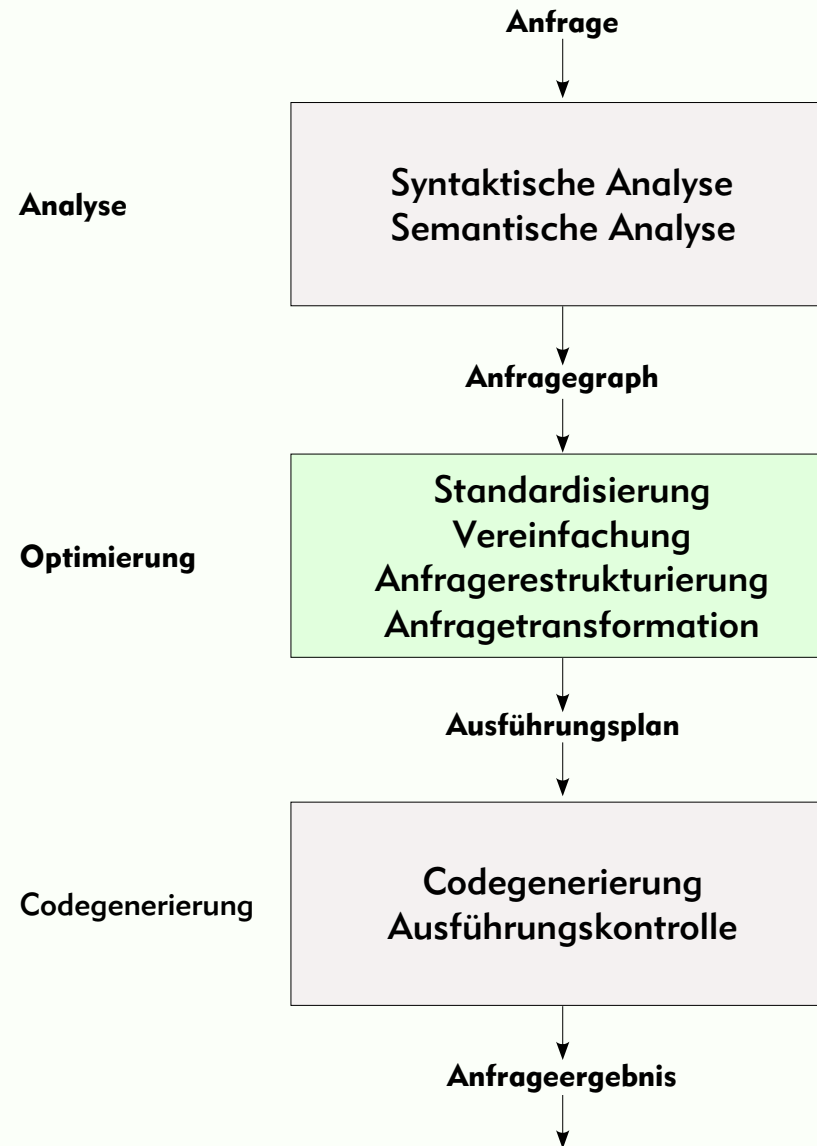
vs.

Minimierung der Ressourcennutzung bei gegebenem Output

Verarbeitungsschritte bei der Anfrageverarbeitung



Verarbeitungsschritte bei der Anfrageverarbeitung



Standardisierung

Vorgehen

- Überführung des Prädikates in **DNF** bzw. **KNF**
- Anwendung der Umformungsregeln für Boole'sche Ausdrücke

Beispiel

```
((Alter <= 24 OR (Beruf = 'Informatiker' OR Beruf = 'Lehrer'))  
  AND (Beruf = 'Lehrer' OR NOT(Alter <= 24)))
```

≡

```
((Beruf = 'Informatiker' AND NOT(Alter <= 24))  
  OR Beruf = 'Lehrer')
```


Vereinfachung

Ziel

- Aufdeckung und Auflösung von Redundanzen
- Überprüfung der Erfüllbarkeit des Prädikates
- Integritätsprüfung

Beispiel

```
(ALTER >= 30 OR (Alter >= 30 AND (Beruf = 'Informatiker')))  
≡  
(ALTER >= 30)
```

Anfragerestrukturierung

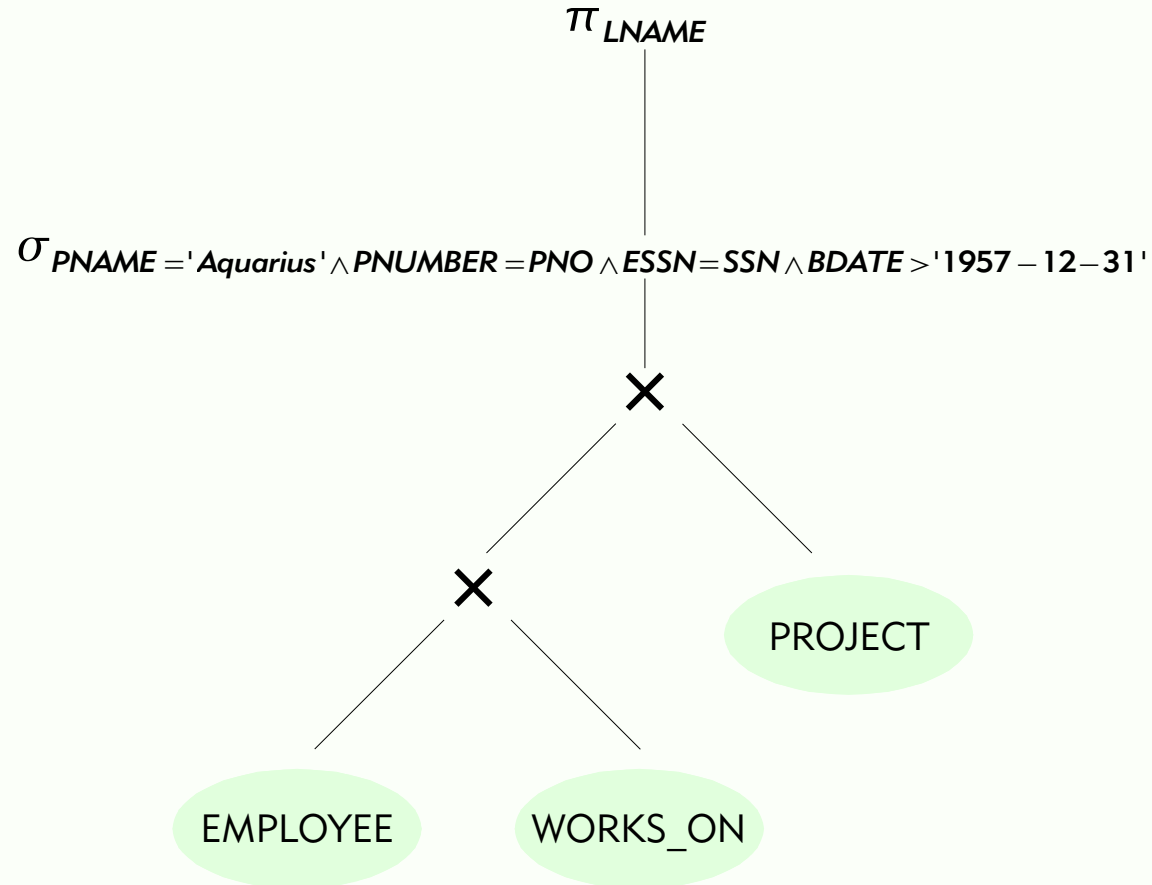
Regeln

- (1) Führe Selektionen und Projektionen so früh wie möglich aus
- (2) Fasse Folge von unären Operatoren auf einer Relation zu einer Operation mit komplexerem Prädikat zusammen
- (3) Fasse Selektionen und Projektionen, die eine Relation betreffen, so zusammen, dass jedes Tupel nur einmal verarbeitet werden muss
- (4) Minimiere die Zwischenergebnisse bei Folgen von binären Operatoren
- (5) Werte gleiche Teile im Anfragegraphen nur einmal aus

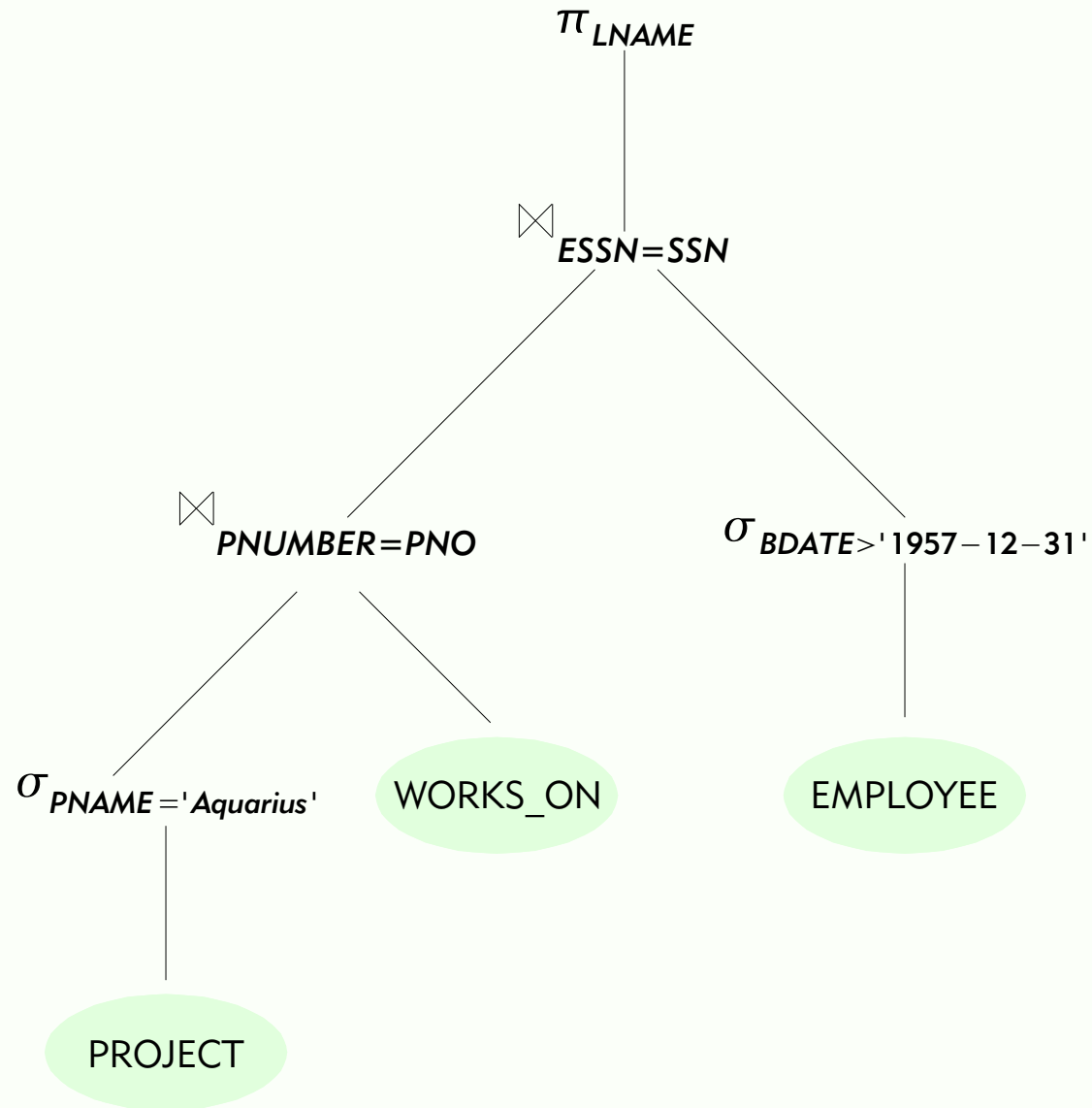
Beispiel zur Anfragerestrukturierung

```
SELECT      LNAME
FROM        EMPLOYEE, WORKS_ON, PROJECT
WHERE        PNAME = 'Aquarius'           AND
              PNUMBER = PNO               AND
              ESSN = SSN                   AND
              BDATE > '1957-12-31';
```

Anfragegraph vor der Restrukturierung



Anfragegraph nach der Restrukturierung



Anfragetransformation

Nicht-algebraische Optimierung

Betrachte zusätzlich die Eigenschaften der physischen Operatoren (Planoperatoren)

Vorgehen

- Ersetzung der logischen Operatoren durch Planoperatoren (Selektion, Projektion, Sortierung, Aggregation etc.)
- Ersetzung direkt benachbarter Operatoren durch einen speziellen Planoperator
- Bestimmung der Verknüpfungsreihenfolge bei Verbund-Operatoren
- Erkennung von gemeinsamen Teilbäumen

Kostenmodelle und -abschätzungen

regelbasierte Optimierer \longleftrightarrow kostenbasierte Optimierer

Entscheidung über Auswahl des physischen Zugriffsplans
Kostenmodell für Speicherungsstrukturen und Zugriffsoperationen

Grundannahmen

- Attributwerte aller Attribute sind gleichverteilt
- Attributwerte sind voneinander stochastisch unabhängig

Kostenarten

- Kommunikationskosten
- Berechnungskosten
- E/A-Kosten
- Speicherkosten

Kostenmodelle und -abschätzungen (Forts.)

Der Selektivitätsfaktor

gibt an, wieviele Tupel höchstwahrscheinlich ein Prädikat p erfüllen werden.

Annahme: Gleichverteilung

$$Card(\sigma_p(R)) = SF(p) \cdot Card(R)$$

↑
↑
 Selektivitätsfaktor Kardinalität von R

Selektivitätsfaktoren komplexer Ausdrücke

$$SF(p(A) \wedge p(B)) = SF(p(A)) \cdot SF(p(B))$$

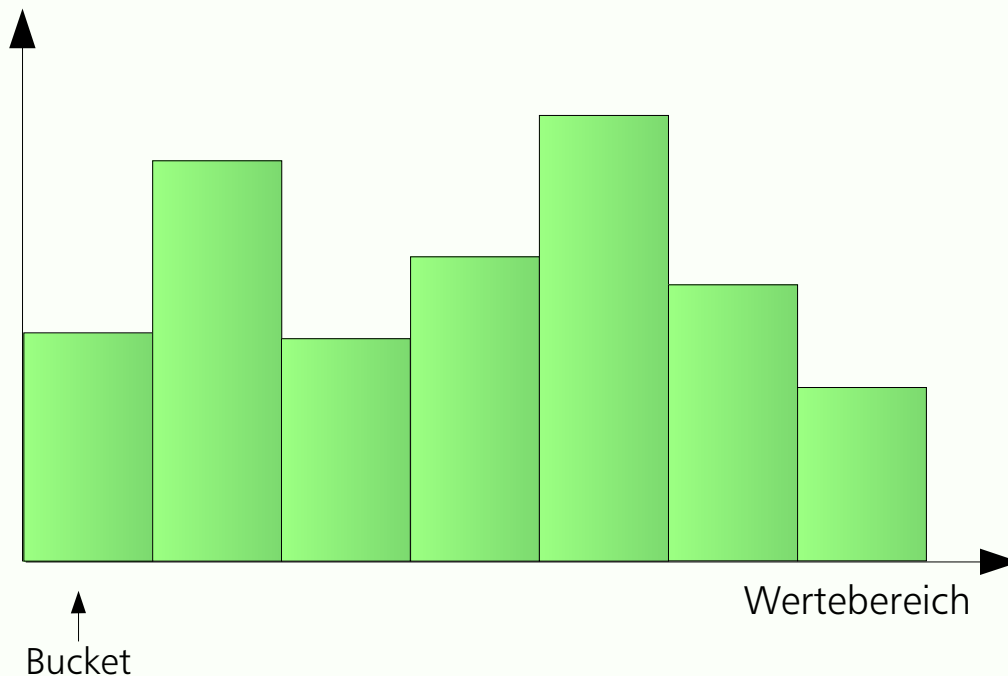
$$SF(p(A) \vee p(B)) = SF(p(A)) + SF(p(B)) - SF(p(A)) \cdot SF(p(B))$$

$$SF(\neg p(A)) = 1 - SF(p(A))$$

Verwaltung von Selektivitätswerten

Histogramme ...

- ermöglichen die Darstellung der Häufigkeitsverteilung von Messwerten
- partitionieren den Wertebereich in disjunkte Teilmengen (*Buckets*)
- gestatten die effiziente Verwaltung von Selektivitätswerten
- erhöhen die Güte von Selektivitätsschätzungen



Kritische Betrachtung der Annahmen

Beispiel

```
(GEHALT >= 100000 AND (ALTER BETWEEN 21 AND 25))
```



Wertebereich(0, 10⁶)



Wertebereich(16, 65)

- **keine** Gleichverteilung bei Gehalt oder Alter
- **keine** stochastische Unabhängigkeit der Attribute

2. Adaptive Anfrageoptimierung

Motivation

Plan-first-execute-next-Methode

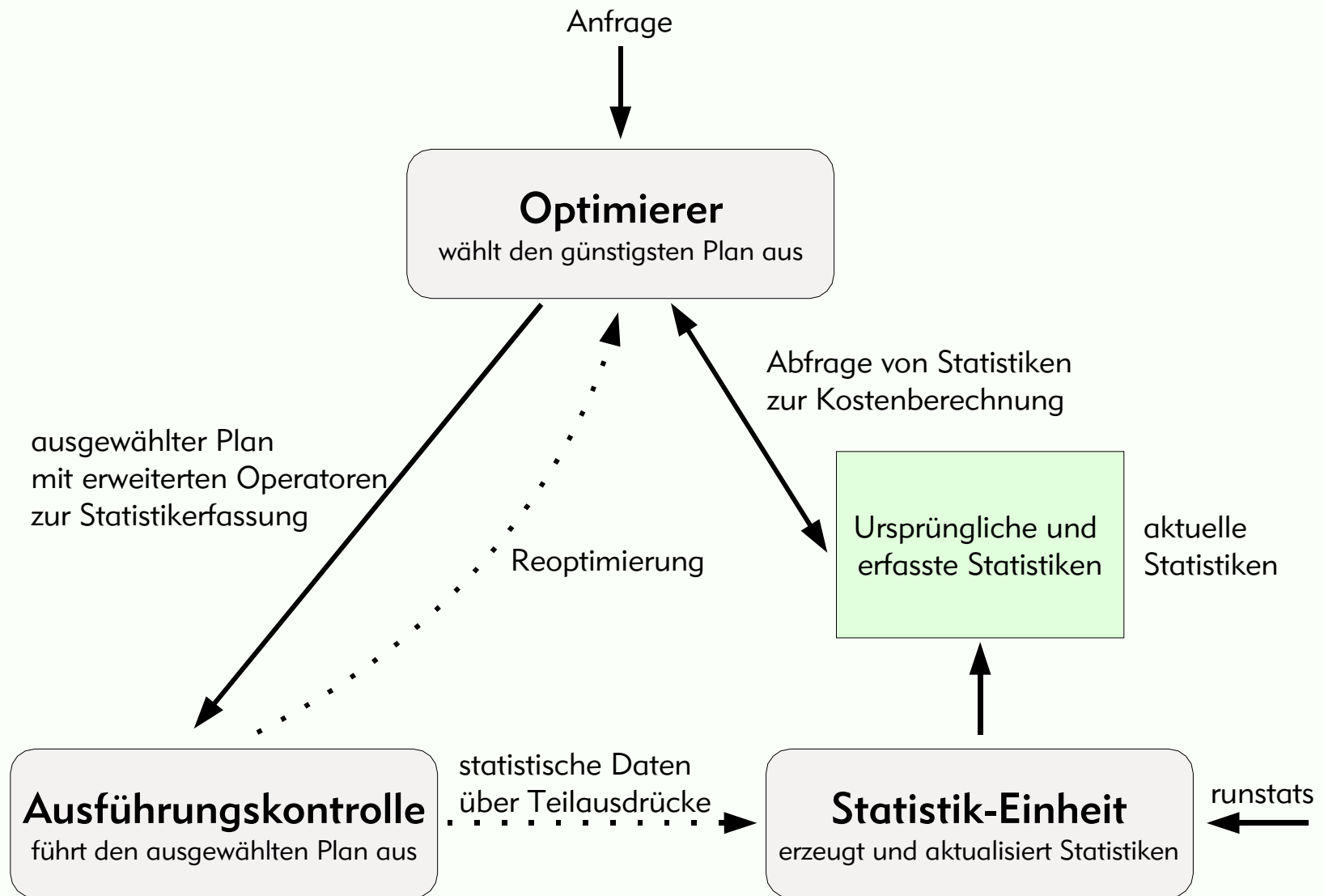
- Optimierung
- Ausführung
- Aktualisierung der Statistiken

Aber

- keine automatische Aktualisierung der Statistiken
- Annahmen problematisch

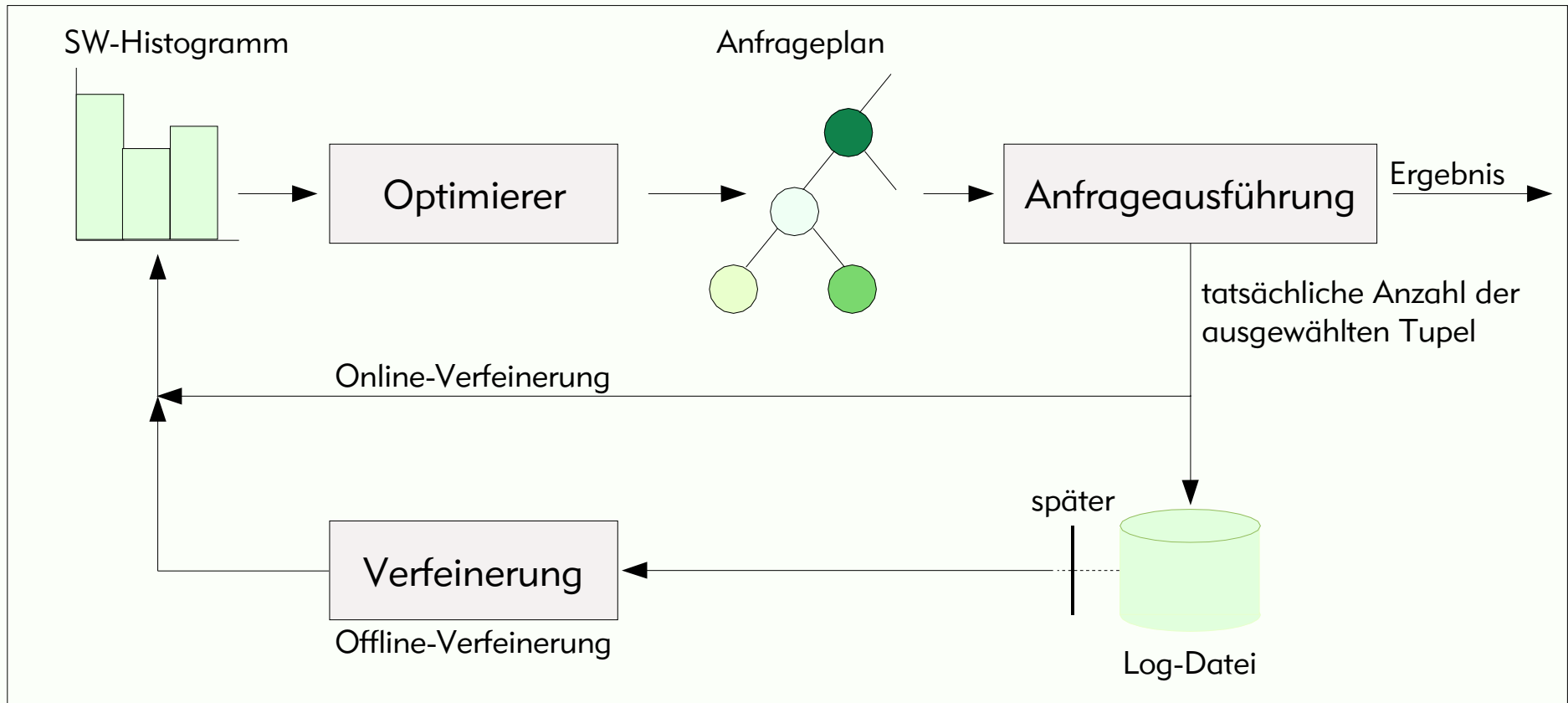
⇒ *Leistung des DBVS wird negativ beeinflusst*

Adaptive Anfrageoptimierung



3. Selbstwartende Histogramme

Selbstwartende Histogramme



- Adaptiver Ansatz zur Wartung von Histogrammen
- Aktualisierung der Histogramme durch Feedback-Mechanismus

Selbstwartende Histogramme (Forts.)

- Ein SW-Histogramm besteht aus einer Menge von Buckets
- Bucket b speichert die Werte im Intervall $[\text{low}(b), \text{high}(b)]$
- **Bucket-Häufigkeit:** # Elemente die im Bucket enthalten sind

Lebenszyklus eines SW-Histogramms

- Initialisierung
- Aktualisierung der Bucket-Häufigkeiten
- Restrukturierung

Aktualisierung der Bucket-Häufigkeiten

Vorgehen

- Bestimme Buckets, die den zu aktualisierenden Wertebereich partiell oder vollständig beinhalten
- Berechne den Schätzfehler: $esterr = act - est$
- Berechne für jedes Bucket den ihm zurechenbaren Fehleranteil
- Korrigiere die Bucket-Häufigkeit

Aktualisierung der Bucket-Häufigkeiten reicht alleine nicht aus, da manche Werte häufiger vorkommen können als andere.

⇒ Restrukturierung notwendig

Restrukturierung von SW-Histogrammen

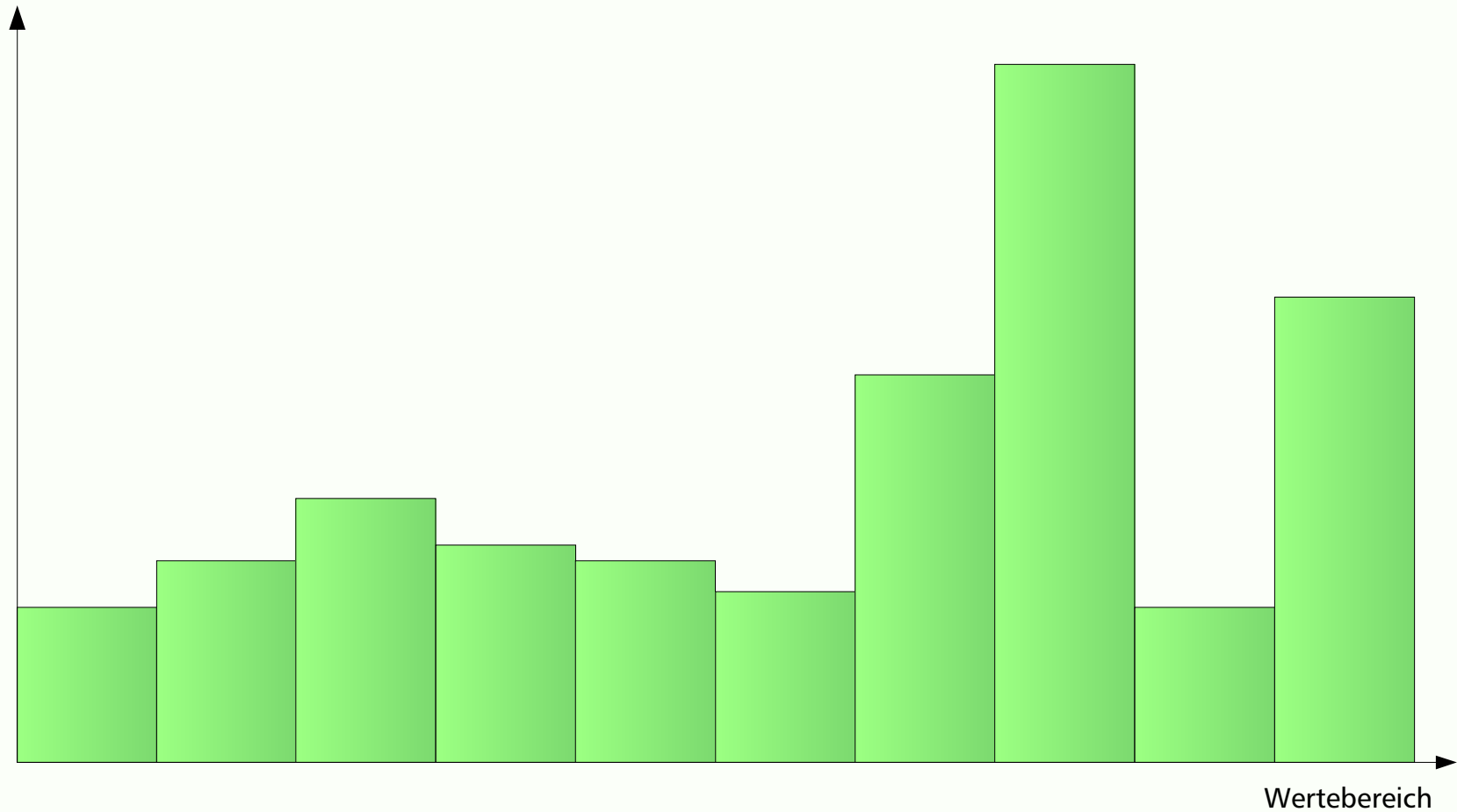
Verschmelzung von Buckets mit ähnlichen Bucket-Häufigkeiten

- Bestimme Folge von benachbarten Buckets mit ähnlichen Bucket-Häufigkeiten
- Verschmelze diejenigen Folgen von Buckets mit minimaler Differenz in den Bucket-Häufigkeiten

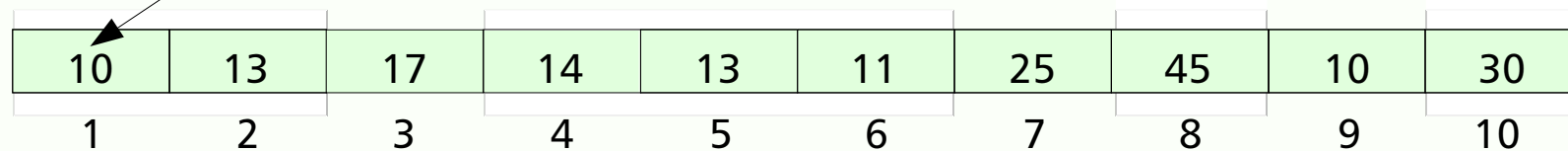
Split-Vorgang

Teile die Wertebereiche der Buckets mit den höchsten Bucket-Häufigkeiten auf die freigesetzten Buckets auf.

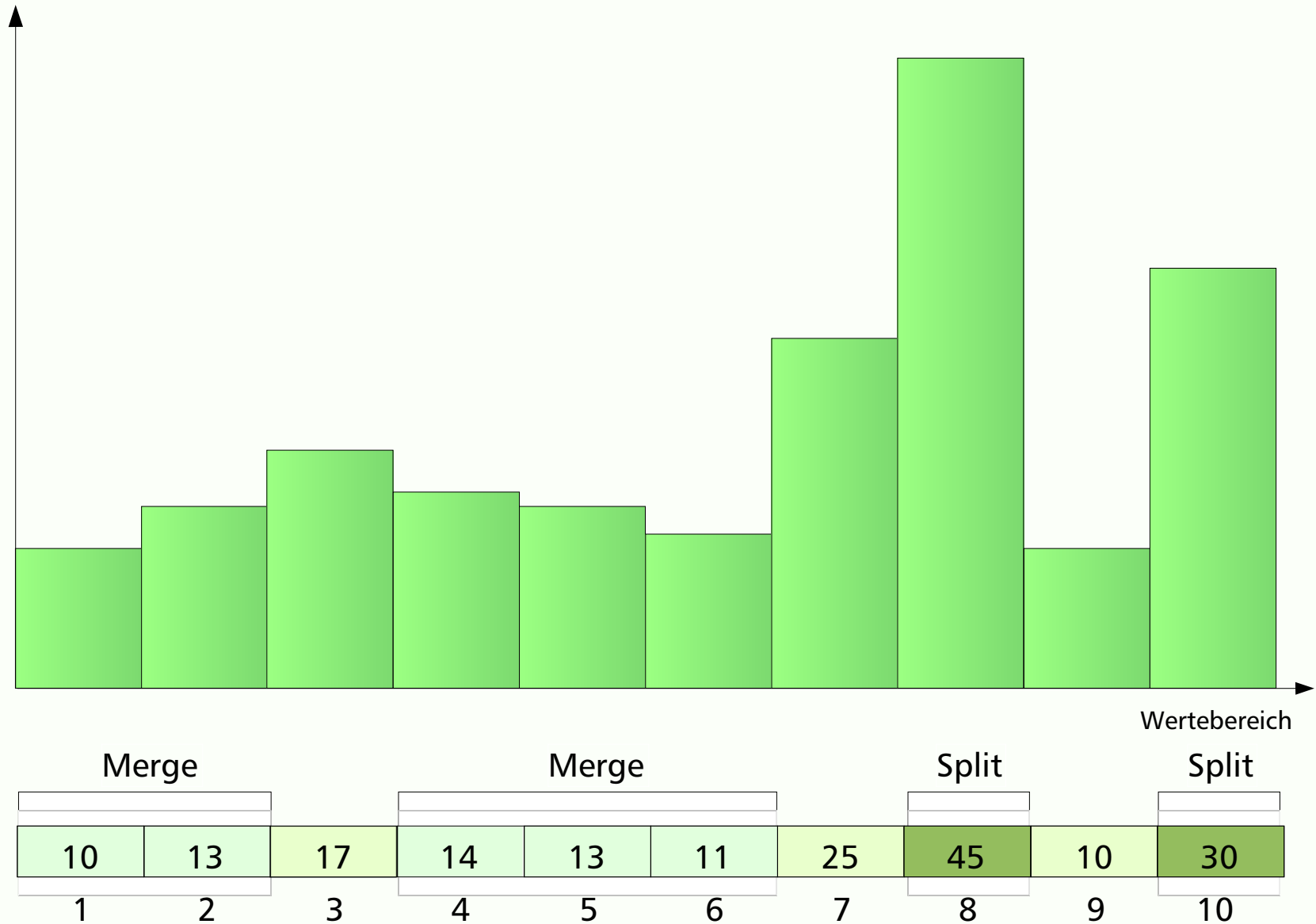
SW-Histogramm vor der Restrukturierung



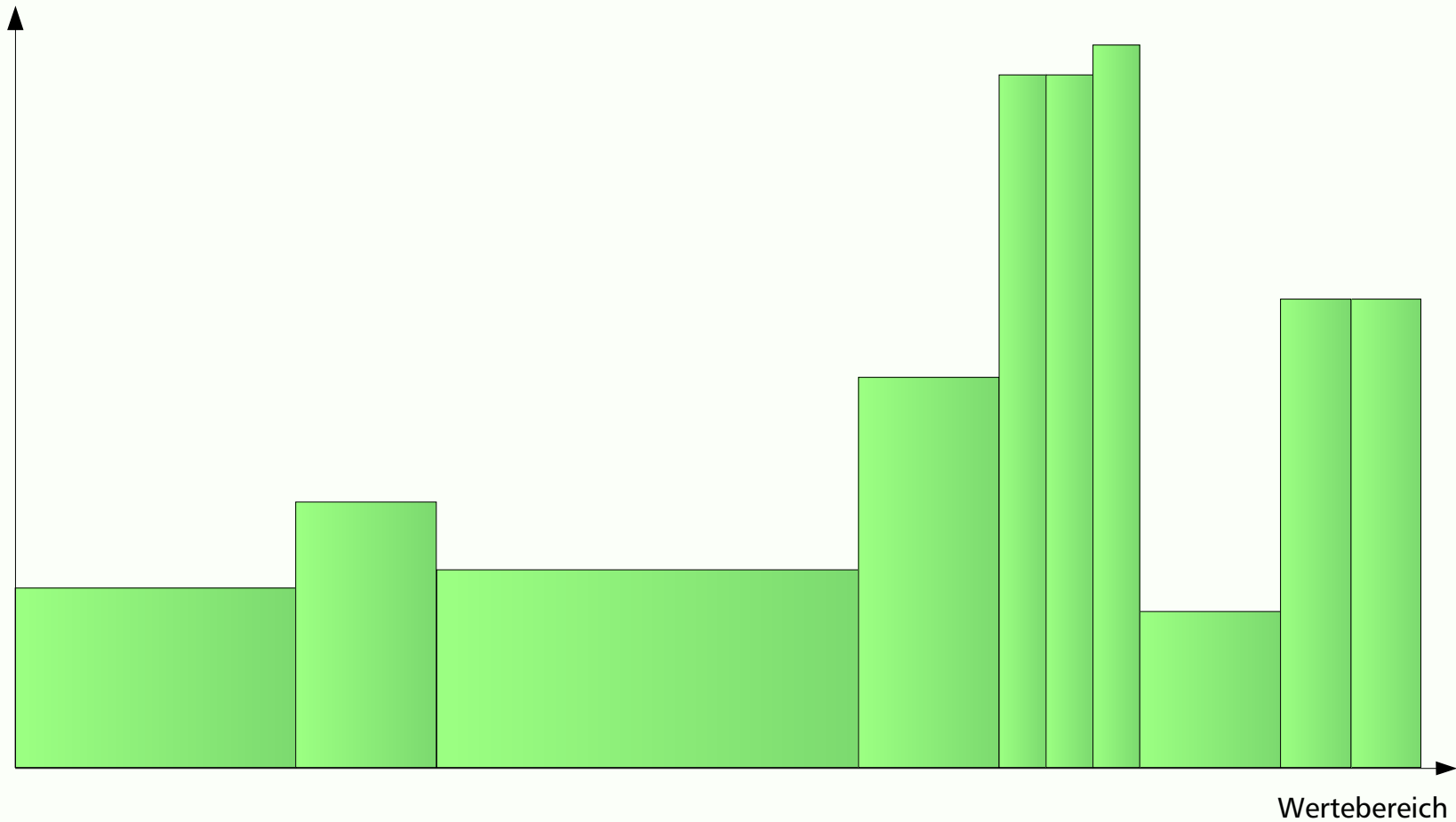
Bucket-Häufigkeit



SW-Histogramm vor der Restrukturierung



SW-Histogramm nach der Restrukturierung



23	17	38	25	14	14	17	10	15	15
1	2	3	4	5	6	7	8	9	10

4. Adaptive Selektivitätsabschätzung

Adaptive Selektivitätsabschätzung

Idee

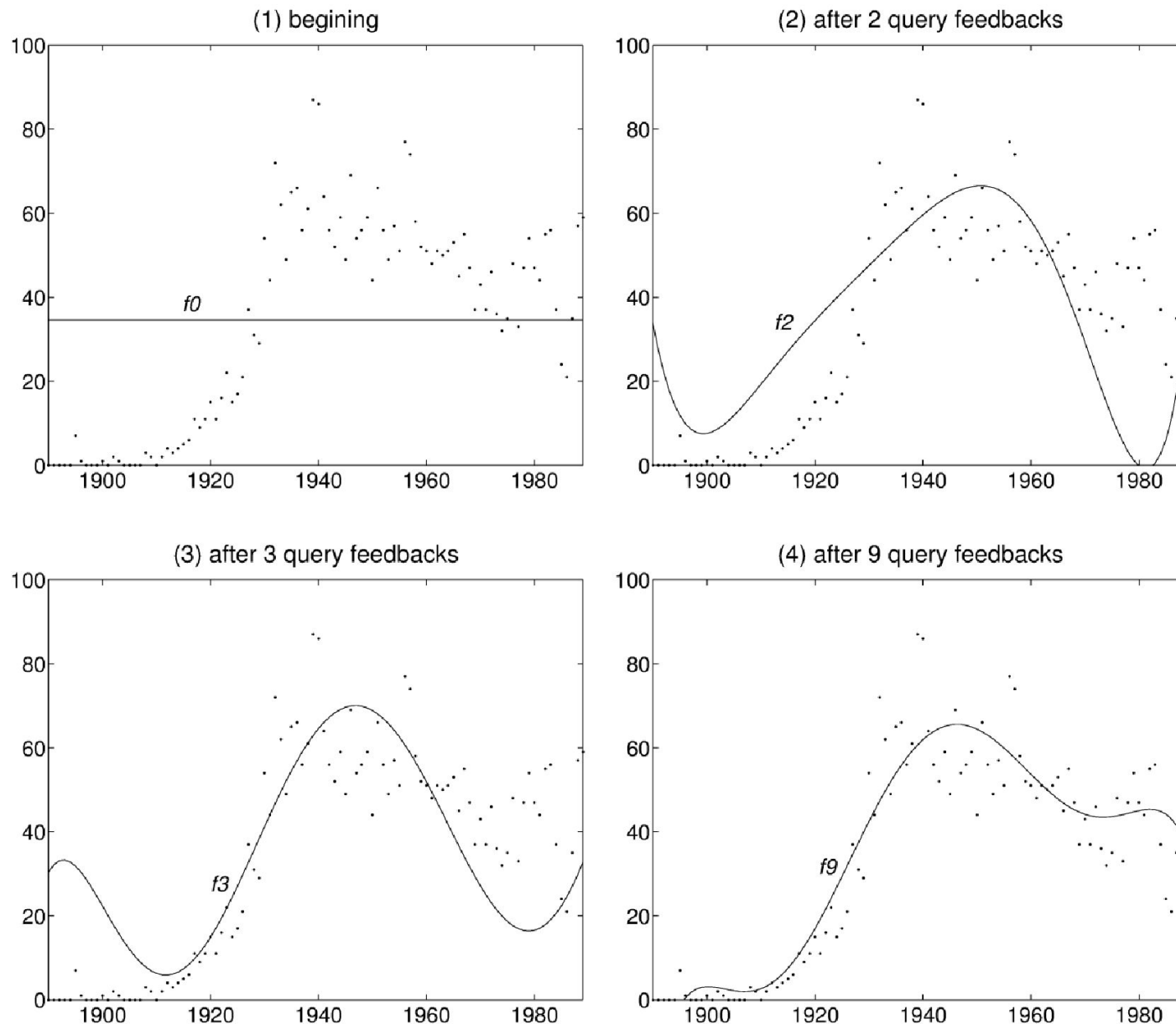
- Approximation der Werteverteilung
- Verwende Feedback des Laufzeitsystems
- „Lernen“ der Selektivität durch Analyse vorhergehender Anfragen
- Verwendet die sog. *Recursive-Least-Square-Error-Technik* zur Anpassung der Werteverteilung an die aktuellen Gegebenheiten

Beispiel

query sequence	1	2	3	4	5
$[l_i, h_i]$	[1935,1966]	[1925,1950]	[1904,1939]	[1890,1923]	[1908,1913]
\hat{s}_i	1073	1138	1248	567	2
s_i	1872	1399	890	136	14

6	7	8	9
[1948,1989]	[1957,1980]	[1964,1989]	[1916,1981]
1956	1103	1041	3173
2033	1130	1134	3045

Adaptive Selektivitätsabschätzung (Forts.)



5. Reoptimierung während der Anfrageverarbeitung

Reoptimierung während der Anfrageverarbeitung

Idee

Veränderung des Anfrageausführungsplans (AP) zur Laufzeit

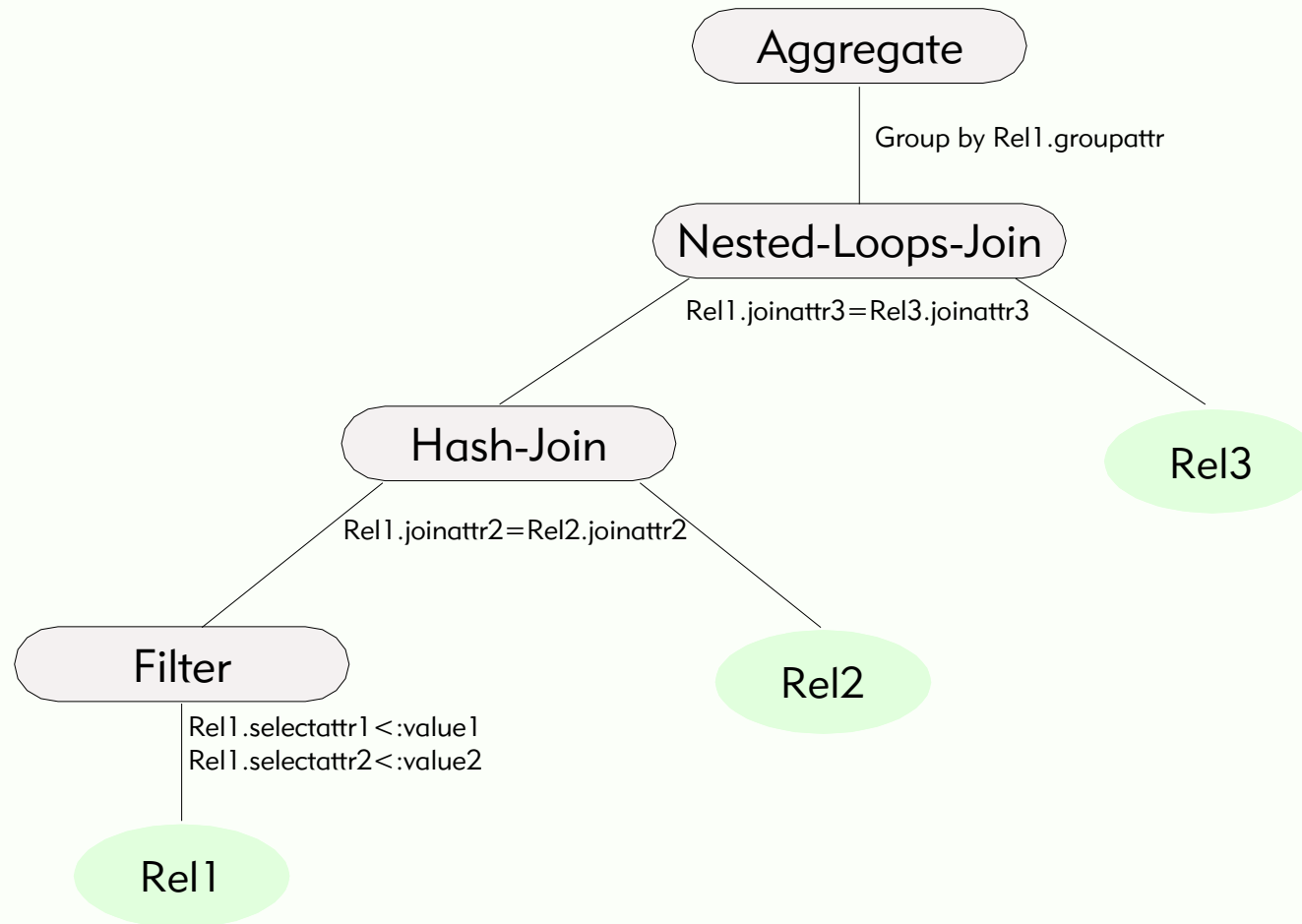
Vorgehen

- Reichere den AP mit statistischen Daten an
- Zeichne statistische Daten zur Laufzeit auf
- Ist die Differenz zwischen den tatsächlichen und geschätzten Werten hinreichend groß \Rightarrow AP suboptimal \Rightarrow Reoptimierung

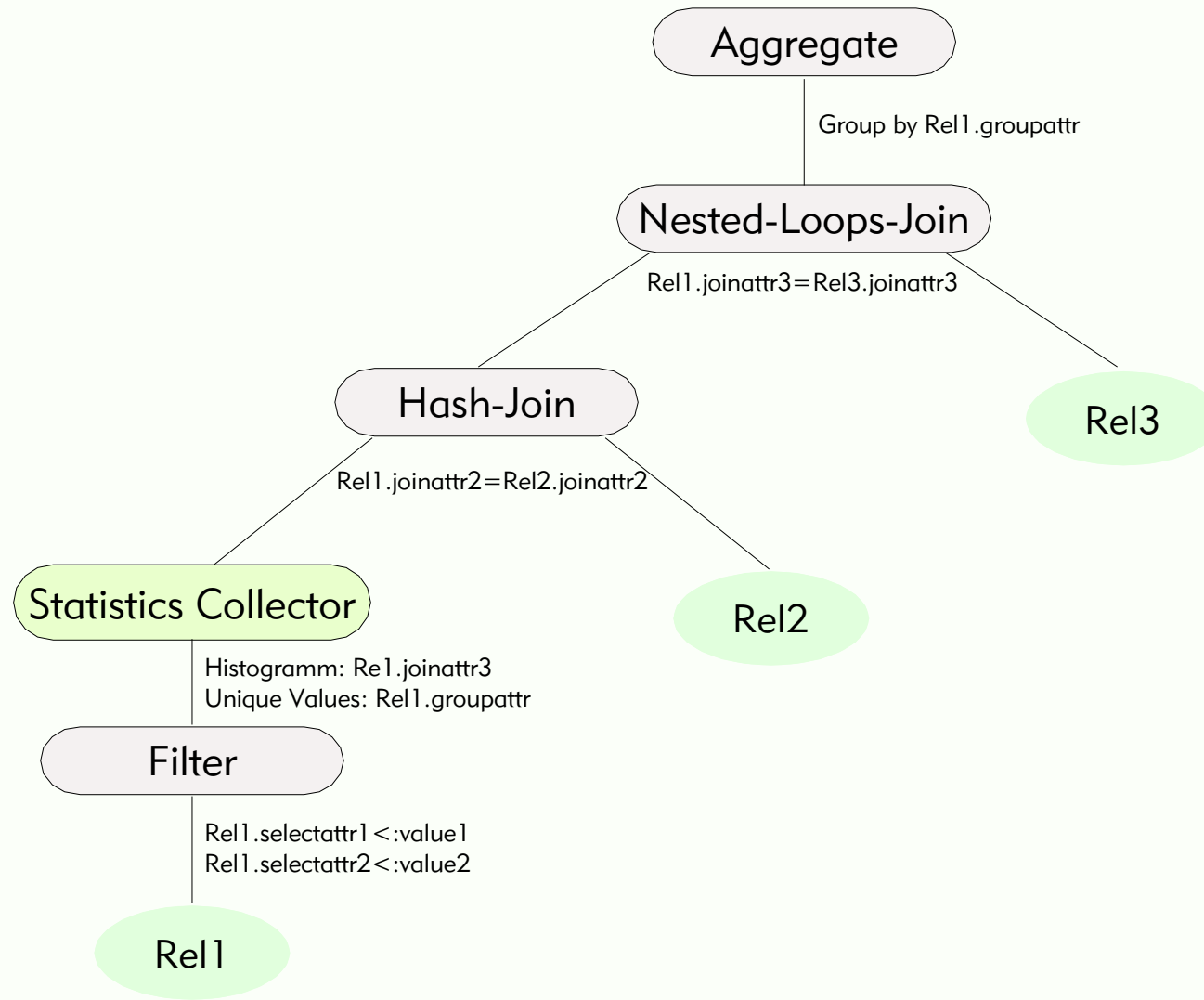
Beispiel

```
SELECT    AVG(Rel1.selectattr1),  
           AVG(Rel1.selectattr2),  
           Rel1.groupattr  
FROM      Rel1, Rel2, Rel2  
WHERE     Rel1.selectattr1 < :value1           AND  
           Rel1.selectattr2 < :value2           AND  
           Rel1.joinattr2 = Rel2.joinattr2      AND  
           Rel1.joinattr3 = Rel3.joinattr3  
GROUP BY Rel1.groupattr;
```

Anfrageausführungsplan ohne Annotationen



Anfrageausführungsplan mit Annotationen

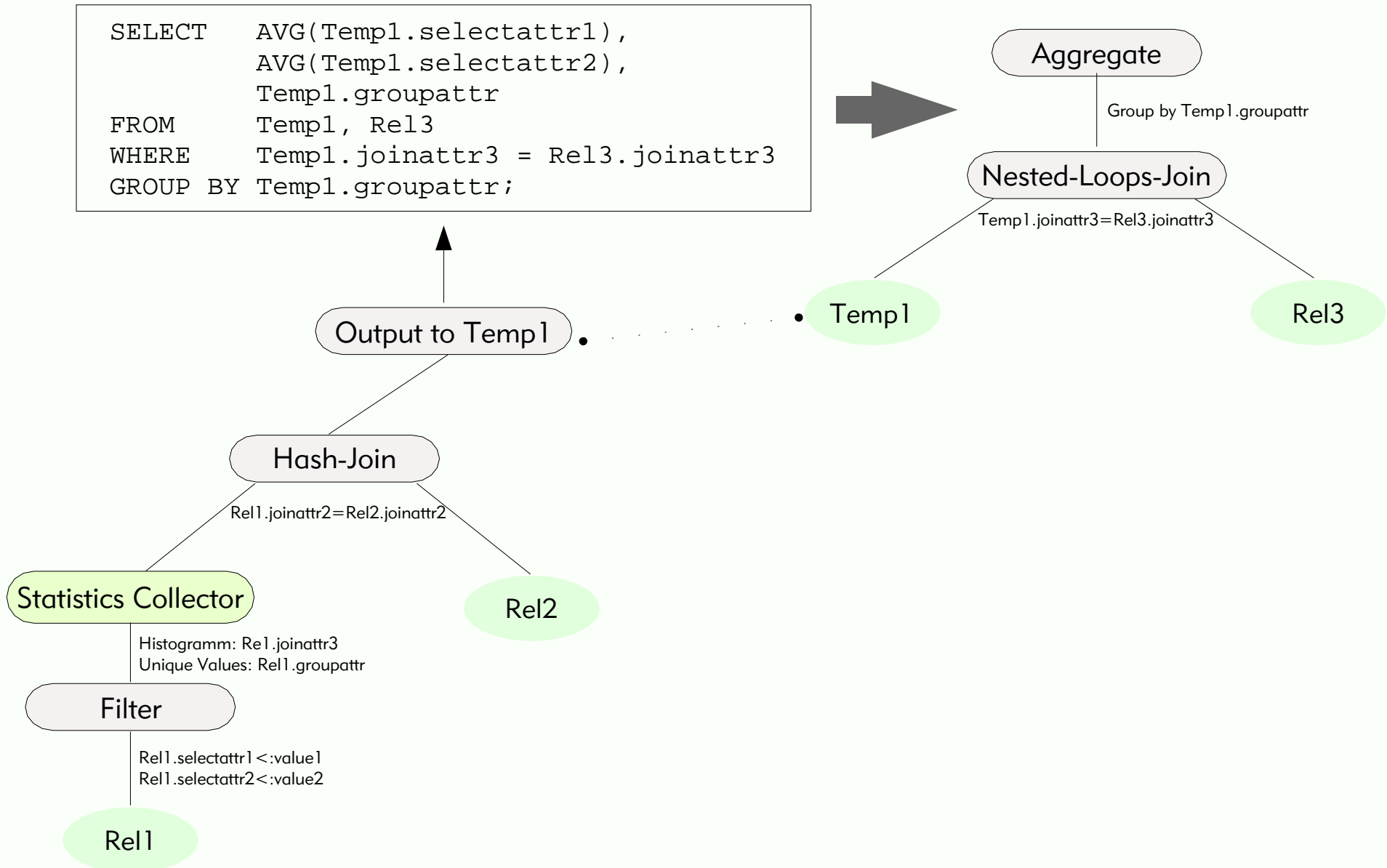


Reoptimierung zur Laufzeit

```

SELECT  AVG(Temp1.selectattr1),
        AVG(Temp1.selectattr2),
        Temp1.groupattr
FROM    Temp1, Rel3
WHERE   Temp1.joinattr3 = Rel3.joinattr3
GROUP BY Temp1.groupattr;

```



6. LEO – Der lernende Optimierer

Der LEarning Optimizer (LEO)

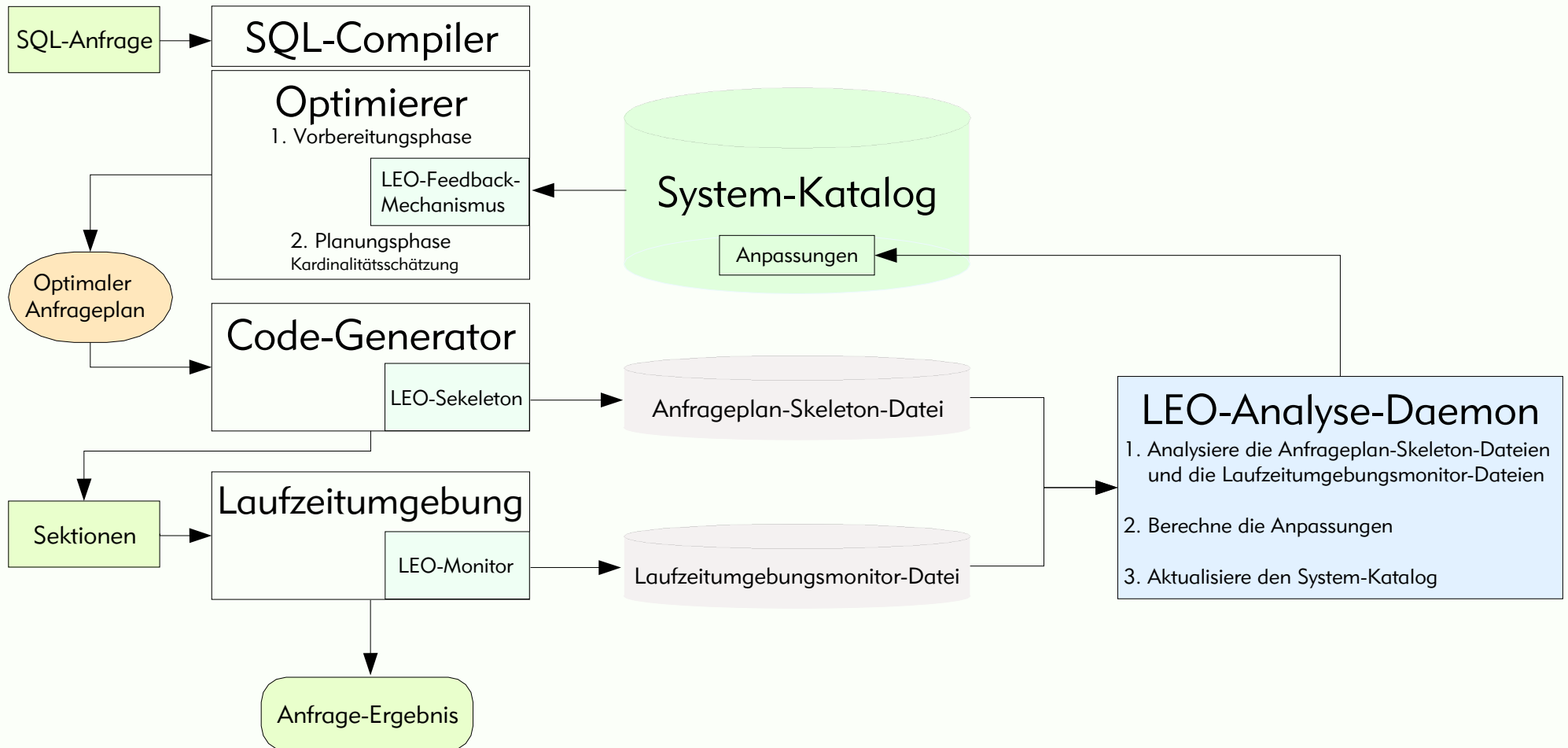
LEO ...

- ist Teil der DB2 Universal Database von IBM
- erkennt Fehler bei der Kardinalitätsabschätzung
- korrigiert falsche Statistiken zur Laufzeit
- kann den AP nicht zur Laufzeit modifizieren
- „lernt“ mit Hilfe eines Feedback-Mechanismus aus Fehlern

Der Feedback-Mechanismus von LEO

- Code-Generator erzeugt Sektion aus optimalem AP
- Während der Übersetzungsphase wird ein sog. *Plan Skeleton* in der DB gespeichert
- Jeder Planoperator wird mit einem Zähler versehen
- Ableitung eines Anpassungsfaktors zur Korrektur fehlerhafter Werte
- Statistiken werden mit Hilfe des Anpassungsfaktors korrigiert

Die Architektur von LEO



Berechnung des Anpassungsfaktors

- Vergleiche geschätzte mit der tatsächlichen Selektivität
- Mit hoher Wahrscheinlichkeit liegt ein Fehler vor, wenn gilt:

$$\frac{|\text{est}_{\text{old}} - \text{act}|}{\text{act}} > 0.05$$

geschätzte Selektivität
tatsächliche Selektivität

Berechnung des Anpassungsfaktors

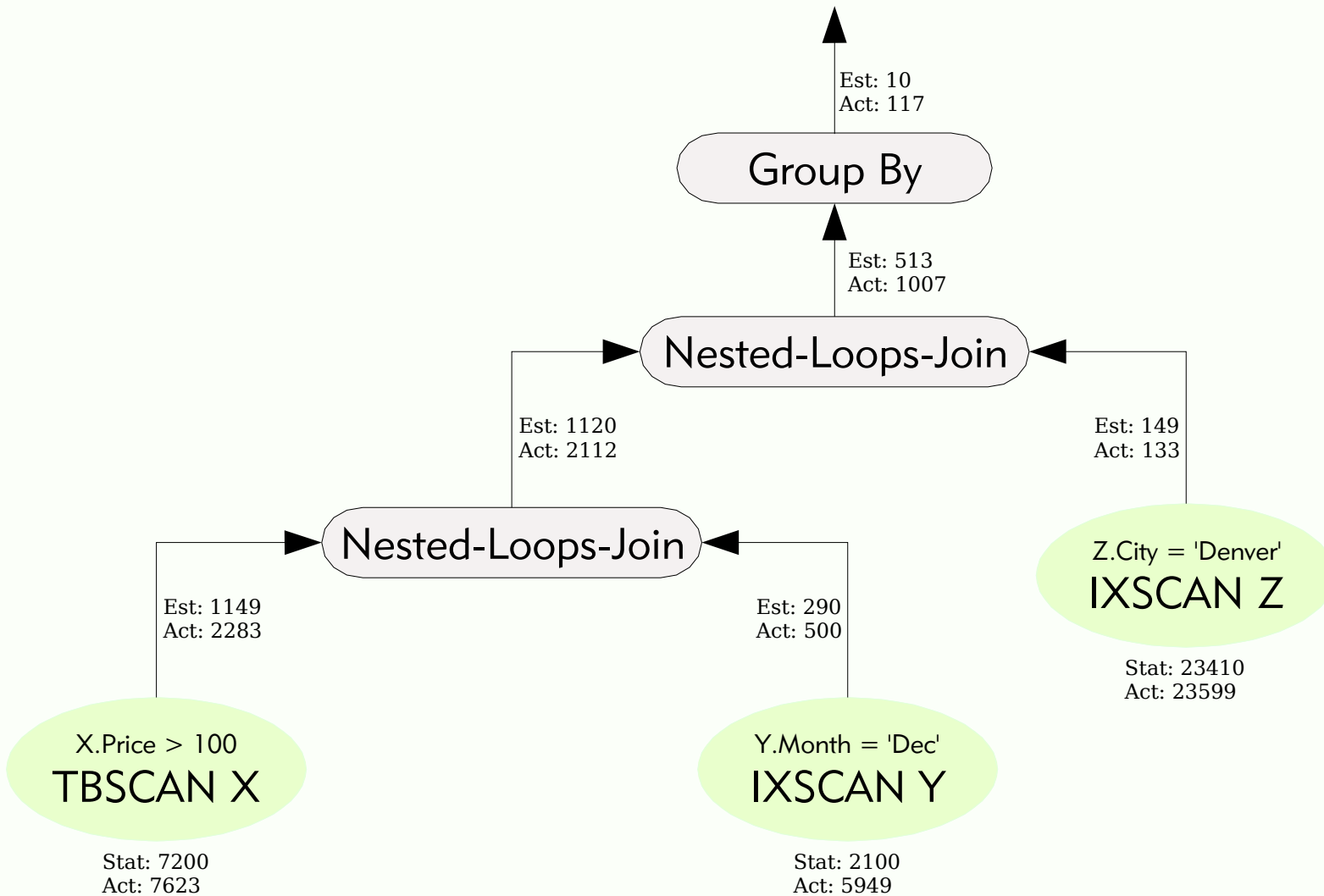
$$\text{adj} = \frac{\text{act} \cdot \text{adj}_{\text{old}}}{\text{est}_{\text{old}}}$$

neuer Anpassungsfaktor
alter Anpassungsfaktor

Beispiel

```
SELECT      *  
FROM        X, Y, Z  
WHERE       X.Price >= 100      AND  
             Z.City   = 'Denver' AND  
             Y.Month  = 'Dec'    AND  
             X.ID     = Y.ID     AND  
             Y.NR     = Z.NR  
GROUP BY   A;
```

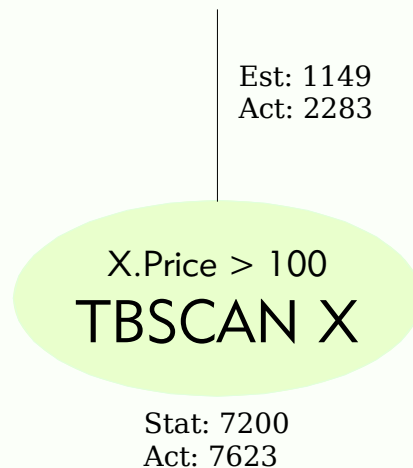
Beispiel (Forts.)



Stat # Tupel aus dem Katalog
Est Schätzwert für die Kardinalität
Act Tatsächliche Kardinalität

Beispiel (Forts.)

Bestimme den Anpassungsfaktor für das Prädikat ($X > 100$)



$$\text{act} = \frac{2283}{7623} = 0,2994$$

$$\text{est}_{\text{old}} = \frac{1149}{7200} = 0,1595$$

$$\text{adj} = \frac{0,2994}{0,1595} = 1,877$$

7. Fazit und Ausblick

Fazit und Ausblick

- Grundlagen der Anfrageoptimierung
- Adaptive Anfrageoptimierung
- Selbstwartende Histogramme
- Adaptive Selektivitätsabschätzung
- Reoptimierung während der Anfrageverarbeitung
- IBM Learning Optimizer

Bisher

Verbesserung der Adaptionsfähigkeit einzelner DBVS

Zukünftig

Adaptive Anfrageoptimierung im Kontext föderierter DBVS

Vielen Dank für Ihre Aufmerksamkeit!

Haben Sie noch Fragen?

