

1 Algebraische Äquivalenzen

Zeigen Sie, wie man die folgenden Äquivalenzen durch eine Reihe von Transformationen unter Verwendung der in der Vorlesung (Folien 11–16) besprochenen Äquivalenzregeln ableitet.

1. $\pi_{A,B,C}(\sigma_{A=B \wedge A \neq C}(R)) = \sigma_{A=B \wedge A \neq C}(\pi_{A,B,C}(R))$
2. $\sigma_{\theta_1 \wedge \theta_2}(R_1 \bowtie_{\theta_3} R_2) = \sigma_{\theta_2}(R_2 \bowtie_{\theta_3} (\sigma_{\theta_1}(R_1)))$ wo θ_1 nur Attribute von R_1 verwendet.

2 Algebraisch Inäquivalenzen

Geben Sie für jedes der folgenden Paare von Ausdrücken Instanzen von Relationen an, die zeigen, dass die Ausdrücke NICHT äquivalent sind.

1. $\sigma_{B < 5}(\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(R)))$ and $\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(\sigma_{B < 5}(R)))$
2. $\pi_A(R_1 - R_2)$ and $\pi_A(R_1) - \pi_A(R_2)$
3. Wären die Ausdrücke in Frage 1. äquivalent, wenn beide Vorkommen von *max* durch *min* ersetzt würden?
4. $(R \bowtie P) \bowtie Q$ and $R \bowtie (P \bowtie Q)$
Mit anderen Worten, der Natural-Left-Outer-Join ist nicht assoziativ. (Hinweis: Nehmen Sie an, dass die Schemata der drei Relationen $R(a, b)$, $S(a, c)$ und $T(a, d)$ sind.)
5. $\sigma_{\theta}(R \bowtie S)$ and $R \bowtie \sigma_{\theta}(S)$, wobei θ nur Attribute von R_2 verwendet.

3 Query Optimization

- (a) Sehen Sie sich die “Phasen der Logical Query Optimization”, die wir in der Vorlesung besprochen haben (Folie 17), noch einmal an. Erklären Sie die Wirkung jedes Schrittes in Ihren eigenen Worten und beschreiben Sie, wie sie die Optimierung beeinflussen.
- (b) Was genau ist der Einfluss der Äquivalenzen in der relationalen Algebra, die wir auch in der Vorlesung besprochen haben (Folien 11–16)? Bitte geben Sie für jede der oben genannten Phasen an, welche Äquivalenzen verwendet werden.
- (c) Finden Sie äquivalente Ausdrücke in der relationalen Algebra, die effizienter in ihrer Ausführung sind als die gegebenen. Zur Veranschaulichung sind unten einige Tabellen mit Beispielinhalten dargestellt.

1. $\pi_{\text{name}}(\sigma_{\text{type}='ore' \wedge \text{name}='Zephyr'}(\text{Shp} \bowtie_{\text{pCID}=\text{CID}} \text{Crg}))$

1 Algebraische Äquivalenzen

Zeigen Sie, wie man die folgenden Äquivalenzen durch eine Reihe von Transformationen unter Verwendung der in der Vorlesung (Folien 11–16) besprochenen Äquivalenzregeln ableitet.

1. $\pi_{A,B,C}(\sigma_{A=B \wedge A \neq C}(R)) = \sigma_{A=B \wedge A \neq C}(\pi_{A,B,C}(R))$

2. $\sigma_{\theta_1 \wedge \theta_2}(R_1 \bowtie_{\theta_3} R_2) = \sigma_{\theta_2}(R_2 \bowtie_{\theta_3} (\sigma_{\theta_1}(R_1)))$ wo θ_1 nur Attribute von R_1 verwendet.

Vertauschen der Reihenfolge von σ und π
Falls die Selektion sich nur auf Attribute A_1, \dots, A_n der Projektionsliste bezieht, können die beiden Operationen vertauscht werden:

$$\pi_{A_1, \dots, A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1, \dots, A_n}(R))$$

1) $\pi_{A,B,C}(\sigma_{A=B \wedge A \neq C}(R))$

Wenn die Selektion σ nur auf Attribute in der Projektionsliste π zugreift, kann man σ und π vertauschen

σ verwendet nur A, B, C π beinhaltet A, B, C ✓

erlaubt

$\Rightarrow \pi_{A,B,C}(\sigma_{A=B \wedge A \neq C}(R)) = \sigma_{A=B \wedge A \neq C}(\pi_{A,B,C}(R))$

2) $\sigma_{\theta_1 \wedge \theta_2}(R_1 \bowtie_{\theta_3} R_2) = \sigma_{\theta_1}(\sigma_{\theta_2}(R_1 \bowtie_{\theta_3} R_2)) = \sigma_{\theta_2}(\sigma_{\theta_1}(R_1 \bowtie_{\theta_3} R_2))$

Vertauschen von σ und \bowtie
Falls das Selektionsprädikat c nur auf Attribute der Relation R zugreift, kann man die beiden Operationen vertauschen:

i) $\sigma_{\theta_1}(R_1 \bowtie_{\theta_3} R_2) = \sigma_{\theta_1}(R_1) \bowtie_{\theta_3} R_2$

$$\sigma_c(R \bowtie_j S) \equiv \sigma_c(R) \bowtie_j S$$

\cup, \cap und \bowtie sind kommutativ

$$R \bowtie_c S \equiv S \bowtie_c R$$

$\sigma_{\theta_2}(\sigma_{\theta_1}(R_1) \bowtie_{\theta_3} R_2) = \sigma_{\theta_2}(R_2 \bowtie_{\theta_3} \sigma_{\theta_1}(R_1))$

Join ist kommutativ, solange θ_3 entsprechend angepasst wird

$$R_1.id = R_2.proflD$$

$$R_1 \bowtie_{R_1.id = R_2.proflD} R_2$$

$$R_2.proflD = R_1.id$$

$$R_2 \bowtie_{R_2.proflD = R_1.id} R_1$$

↕ θ_3 anpassen

2 Algebraisch Inäquivalenzen

Geben Sie für jedes der folgenden Paare von Ausdrücken Instanzen von Relationen an, die zeigen, dass die Ausdrücke NICHT äquivalent sind.

- $\sigma_{B < 5}(\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(R)))$ and $\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(\sigma_{B < 5}(R)))$
- $\pi_A(R_1 - R_2)$ and $\pi_A(R_1) - \pi_A(R_2)$
- Wären die Ausdrücke in Frage 1. äquivalent, wenn beide Vorkommen von *max* durch *min* ersetzt würden?
- $(R \bowtie P) \bowtie Q$ and $R \bowtie (P \bowtie Q)$
Mit anderen Worten, der Natural-Left-Outer-Join ist nicht assoziativ. (Hinweis: Nehmen Sie an, dass die Schemata der drei Relationen $R(a, b)$, $P(a, c)$ und $Q(a, d)$ sind.)
- $\sigma_\theta(R \bowtie S)$ and $R \bowtie \sigma_\theta(S)$, wobei θ nur Attribute von S verwendet.

ρ := Umbenennung

γ := Gruppierung & Aggregation

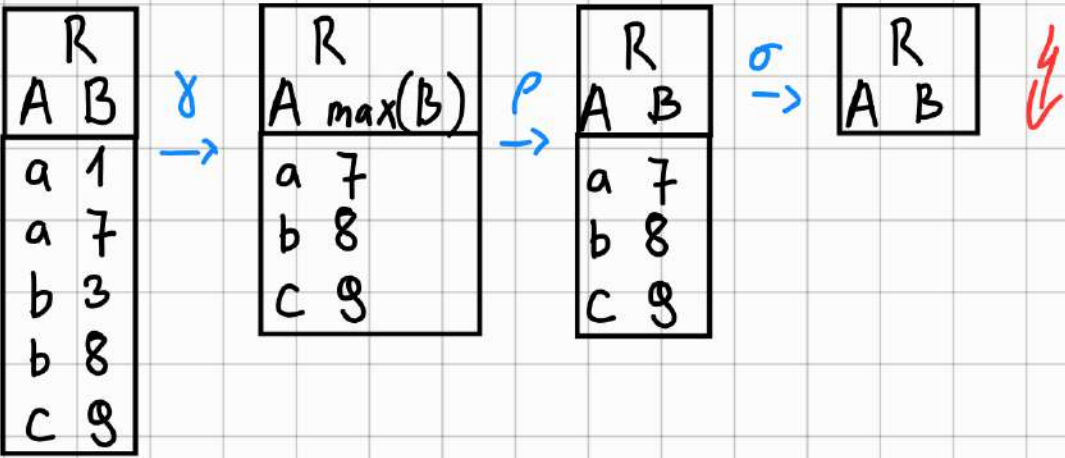
$\rho_{A \leftarrow B}(R)$

$\gamma_{L:F}(R)$

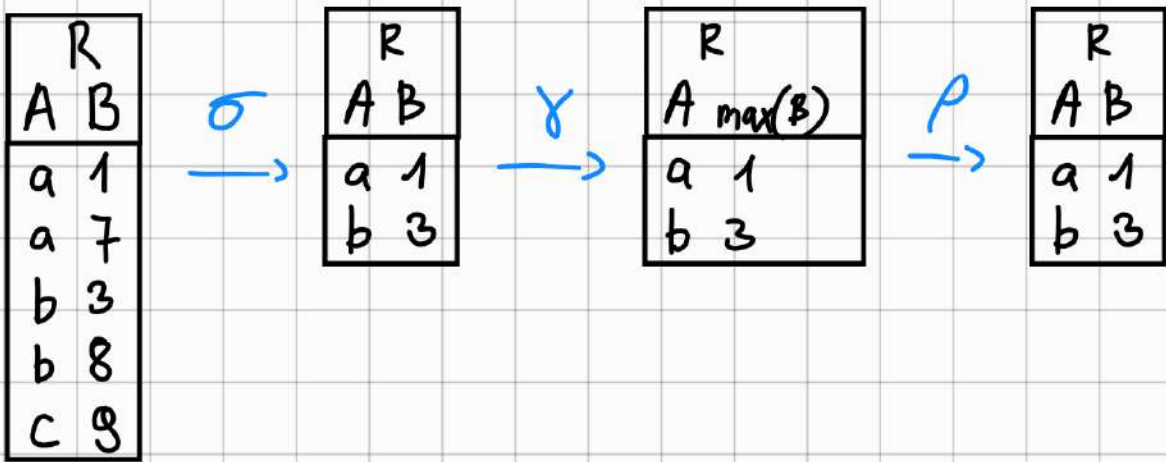
Umbenennung von Attribut B in A

L... Liste der Attribute für die Gruppierung
F... Aggregatfunktion

1) $\sigma_{B < 5}(\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(R)))$



$\rho_{B \leftarrow \max(B)}(\gamma_{A; \max(B)}(\sigma_{B < 5}(R)))$





$A - B$

2) $\pi_A(R_1 - R_2)$

! In der mathematischen Notation ist π immer SELECT DISTINCT

R_1		\neq	R_2		R		R
A	B		A	B	A	B	A
Anna	18		Anna	20	Anna	18	Anna
Bob	17				Anna	20	Bob
					Bob	17	

$\pi_A(R_1) - \pi_A(R_2)$

R_1		R_2		R_1'	$=$	R_2'	R
A	B	A	B	A		A	A
Anna	18	Anna	20	Anna		Anna	Bob
Bob	17			Bob			

3) $\sigma_{B < 5}(p_{B \leftarrow \min(B)}(\gamma_{A; \min(B)}(\sigma_{B < 5}(R)))$

R		γ	R		p	R		σ	R	
A	B		A	$\min(B)$		A	B		A	B
a	1		a	1		a	1		a	1
a	7		b	3		b	3		b	3
b	3		c	9		c	9			
b	8									
c	9									

$p_{B \leftarrow \min(B)}(\gamma_{A; \min(B)}(\sigma_{B < 5}(R)))$

R		σ	R		γ	R		p	R	
A	B		A	B		A	$\min(B)$		A	B
a	1		a	1		a	1		a	1
a	7		b	3		b	3		b	3
b	3									
b	8									
c	9									

Letztes Mal hat man über $\max(B)$ zu viele Zeilen im unteren Zahlenbereich ausgeschlossen.

$\left. \begin{matrix} A & 2 \\ A & 7 \end{matrix} \right\} A > 7 \Rightarrow A < 2 \text{ geht verloren}$

Über $\min(B)$ passieren diese Ausschlüsse nicht mehr

4) $(R \bowtie P) \bowtie Q$

R		P		Q	
name	age	name	salary	name	courseID
Bob	20	Bob	100	Bob	3
John	30	Anna	200	Anna	4
				John	7

$R \bowtie P$			$(R \bowtie P) \bowtie Q$		
name	age	salary	name	age	salary
Bob	20	100	Bob	20	100
John	30	—	John	30	—

courseID
3
7

$R \bowtie (P \bowtie Q)$

$P \bowtie Q$		
name	salary	courseID
Bob	100	3
Anna	200	4

$R \bowtie (P \bowtie Q)$			
name	age	salary	courseID
Bob	20	100	3
John	30	—	—

courseID geht verloren



nicht äquivalent



Left Outer Join
nicht assoziativ

5) $\sigma_{\theta}(R \bowtie S)$

$\theta := s.salary > 100$

R	
name	age
Bob	20
Anna	23
James	24
Don	33

S	
name	salary
Bob	50
Anna	200
James	70

R \bowtie S		
name	age	salary
Bob	20	50
Anna	23	200
James	24	70
Don	33	—

$\sigma_{\theta}(R \bowtie S)$		
name	age	salary
Anna	23	200



$R \bowtie \sigma_{\theta}(S)$

S'	
name	salary
Anna	200

$R \bowtie \sigma_{\theta}(S)$		
name	age	salary
Bob	20	—
Anna	23	200
James	24	—
Don	33	—

3 Query Optimization

- Sehen Sie sich die "Phasen der Logical Query Optimization", die wir in der Vorlesung besprochen haben (Folie 17), noch einmal an. Erklären Sie die Wirkung jedes Schrittes in Ihren eigenen Worten und beschreiben Sie, wie sie die Optimierung beeinflussen.
- Was genau ist der Einfluss der Äquivalenzen in der relationalen Algebra, die wir auch in der Vorlesung besprochen haben (Folien 11–16)? Bitte geben Sie für jede der oben genannten Phasen an, welche Äquivalenzen verwendet werden.
- Finden Sie äquivalente Ausdrücke in der relationalen Algebra, die effizienter in ihrer Ausführung sind als die gegebenen. Zur Veranschaulichung sind unten einige Tabellen mit Beispielinhalten dargestellt.

1. $\pi_{\text{name}}(\sigma_{\text{type}='ore' \wedge \text{name}='Zephyr'}(\text{Shp} \bowtie_{\text{pCID}=\text{CID}} \text{Crg}))$

Phasen der logischen Anfrageoptimierung

- Aufbrechen von Selektionen
- Verschieben der Selektionen so weit wie möglich nach unten (pushing selections)
- Joins einführen (Zusammenfassen von Selektionen und Kreuzprodukten)
- Join-Reihenfolge bestimmen, so dass möglichst kleine Zwischenergebnisse entstehen
Heuristik: Joins mit Input von Selektionen vor anderen Joins auswerten
- ggf. Einführen von Projektionen
- Verschieben der Projektionen so weit wie möglich nach unten
Nicht immer nötig

möglichst früh die Anzahl an Tupel kürzen
↳ Optimierung

1, 2) Man versucht so früh wie möglich Tupel zu kürzen. Es ist besser die Selektionen direkt auszuführen, damit die Zwischentabellen kleiner bleiben. Man eliminiert so den dead weight

3) $\sigma_{R.ID=S.ID} (R \times S)$ $R \bowtie_{R.ID=S.ID} S$
viel Zwischenspeicher JOIN vermeidet das gesamte $R \times S$ zu berechnen über Join-Algorithmen (Nested Loop, Merge, Hash)

In der Regel ist es effizienter JOIN zu nehmen

4) Auch über die Reihenfolge der JOINS kann man die Zwischenergebnisse kleiner halten und so den dead weight früher eliminieren

5) Falls nur wenige Attribute relevant sind, kann man die restlichen über Projektionen eliminieren und wieder so die Zwischenergebnisse möglichst frei von redundanten Informationen halten

6) Selbe Argumentation wie bei 1, 2) und 5)

b)

Phasen der logischen Anfrageoptimierung

- 1 Aufbrechen von Selektionen
- 2 Verschieben der Selektionen so weit wie möglich nach unten (pushing selections)
- 3 Joins einführen (Zusammenfassen von Selektionen und Kreuzprodukten)
- 4 Join-Reihenfolge bestimmen, so dass möglichst kleine Zwischenergebnisse entstehen
Heuristik: Joins mit Input von Selektionen vor anderen Joins auswerten
- 5 ggf. Einführen von Projektionen
- 6 Verschieben der Projektionen so weit wie möglich nach unten
Nicht immer nötig

1) $\sigma_{c_1 \wedge c_2 \wedge \dots \wedge c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(\dots(\sigma_{c_n}(R))))$ Aufbrechen

$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$ beliebige Reihenfolge

2) $\pi_{A_1, \dots, A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1, \dots, A_n}(R))$ Vertauschen π & σ

$\sigma_c(R \bowtie_j S) \equiv \sigma_c(R) \bowtie_j S$

$\sigma_c(R \bowtie_j S) \equiv \sigma_{c_1}(R) \bowtie_j \sigma_{c_2}(S)$ } Vertauschen σ & \bowtie

3) $\sigma_\theta(R \times S) \equiv R \bowtie_\theta S$

4) $R \bowtie_c S \equiv S \bowtie_c R$ $R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$

5) $\pi_{L_1}(\pi_{L_2}(\dots(\pi_{L_n}(R)))) \equiv \pi_{L_1}(R)$

6) Vertauschen π & σ

$\pi_{A_1, \dots, A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1, \dots, A_n}(R))$

Vertauschen π & \bowtie

$\pi_L(R \bowtie_c S) \equiv (\pi_{A_1, \dots, A_n}(R)) \bowtie_c (\pi_{B_1, \dots, B_n}(S))$

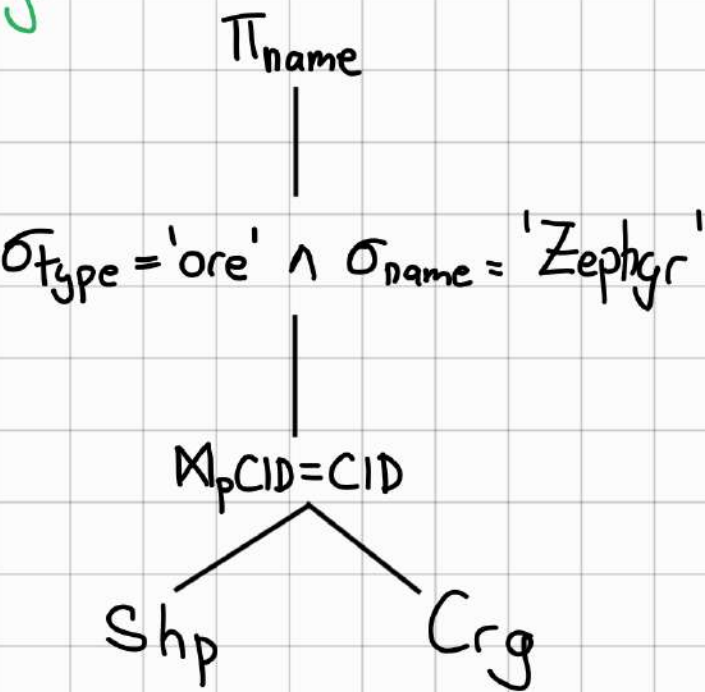
π distributiv mit \cup

$\pi_c(R \cup S) \equiv (\pi_c(R)) \cup (\pi_c(S))$

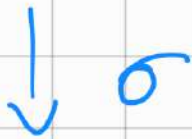
(c) Finden Sie äquivalente Ausdrücke in der relationalen Algebra, die effizienter in ihrer Ausführung sind als die gegebenen. Zur Veranschaulichung sind unten einige Tabellen mit Beispieldaten dargestellt.

1. $\pi_{name}(\sigma_{type='ore' \wedge name='Zephyr'}(Shp \bowtie_{pCID=CID} Crg))$

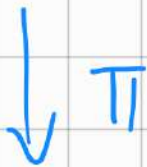
Original



SID	name	pCID	CID	type	risk	val
101	Zephyr	1	1	ore	7	450
102	Orion	2	2	fuel	3	200
103	Vega	3	3	meds	1	850
104	Sirius	3	3	meds	1	850
105	Draco	2	2	fuel	3	200



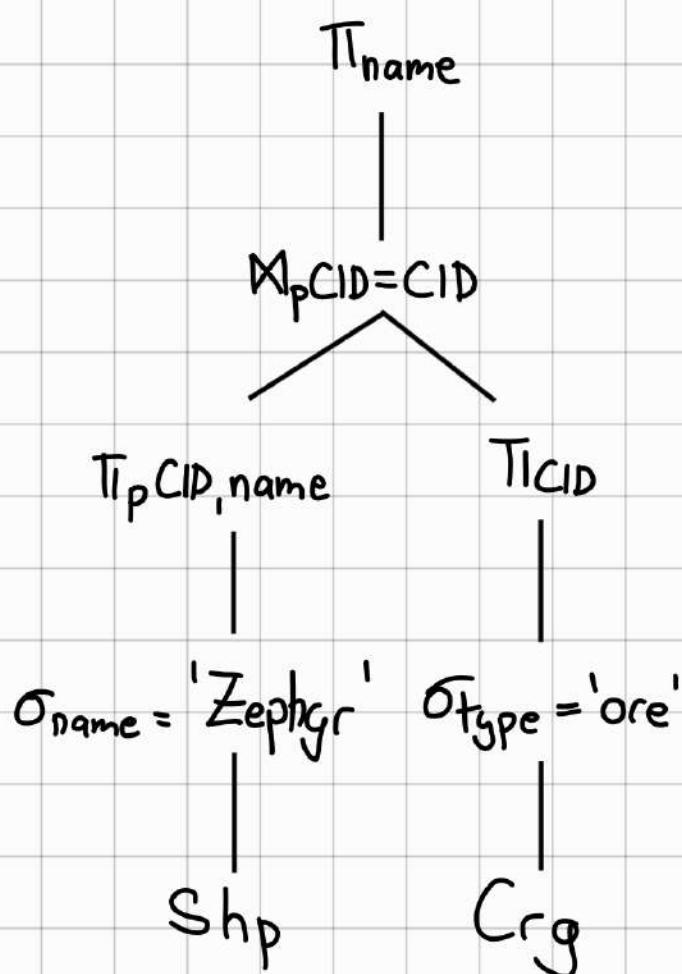
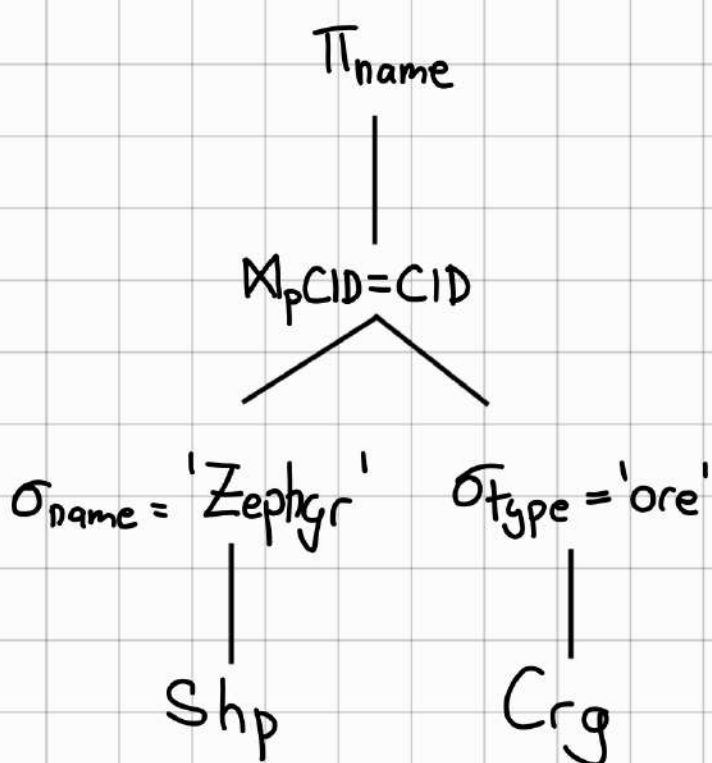
SID	name	pCID	CID	type	risk	val
101	Zephyr	1	1	ore	7	450



name
Zephyr

2) Push Selections down

5, 6) Add Projections and push them down



$\Pi_{name} \left(\Pi_{pCID, name} \left(\sigma_{name='Zephyr'}(Shp) \right) \right.$
 $\left. \bowtie_{pCID=CID} \left(\Pi_{CID} \left(\sigma_{type='ore'}(Crg) \right) \right) \right)$

$\downarrow \sigma$

SID	name	pCID
101	Zephyr	1

$\downarrow \Pi$

name	pCID
Zephyr	1

$\downarrow \sigma$

CID	type	risk	val
1	ore	7	450

$\downarrow \Pi$

CID
1

$\bowtie_{pCID=CID}$

name	pCID	CID
Zephyr	1	1

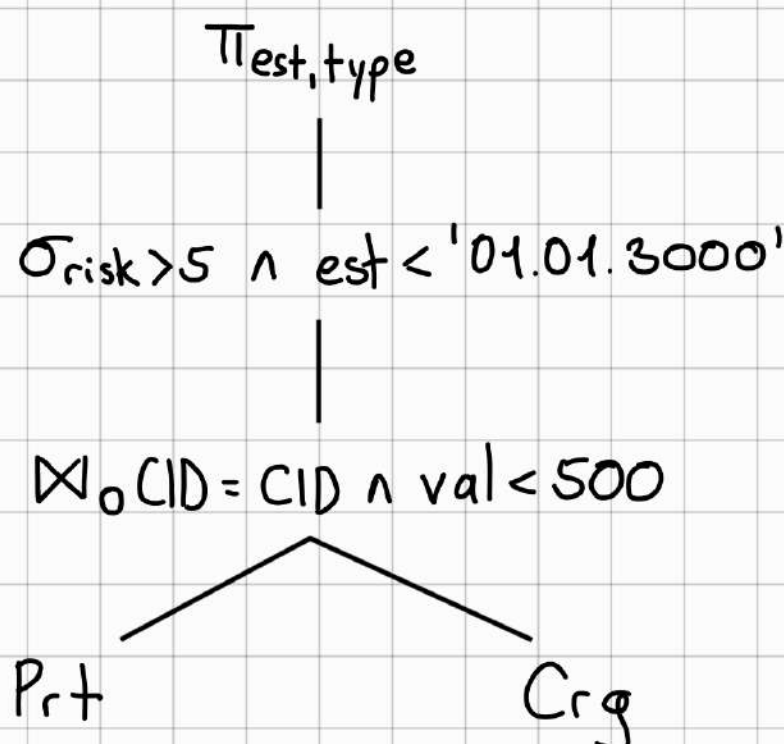
$\downarrow \Pi$

name
Zephyr

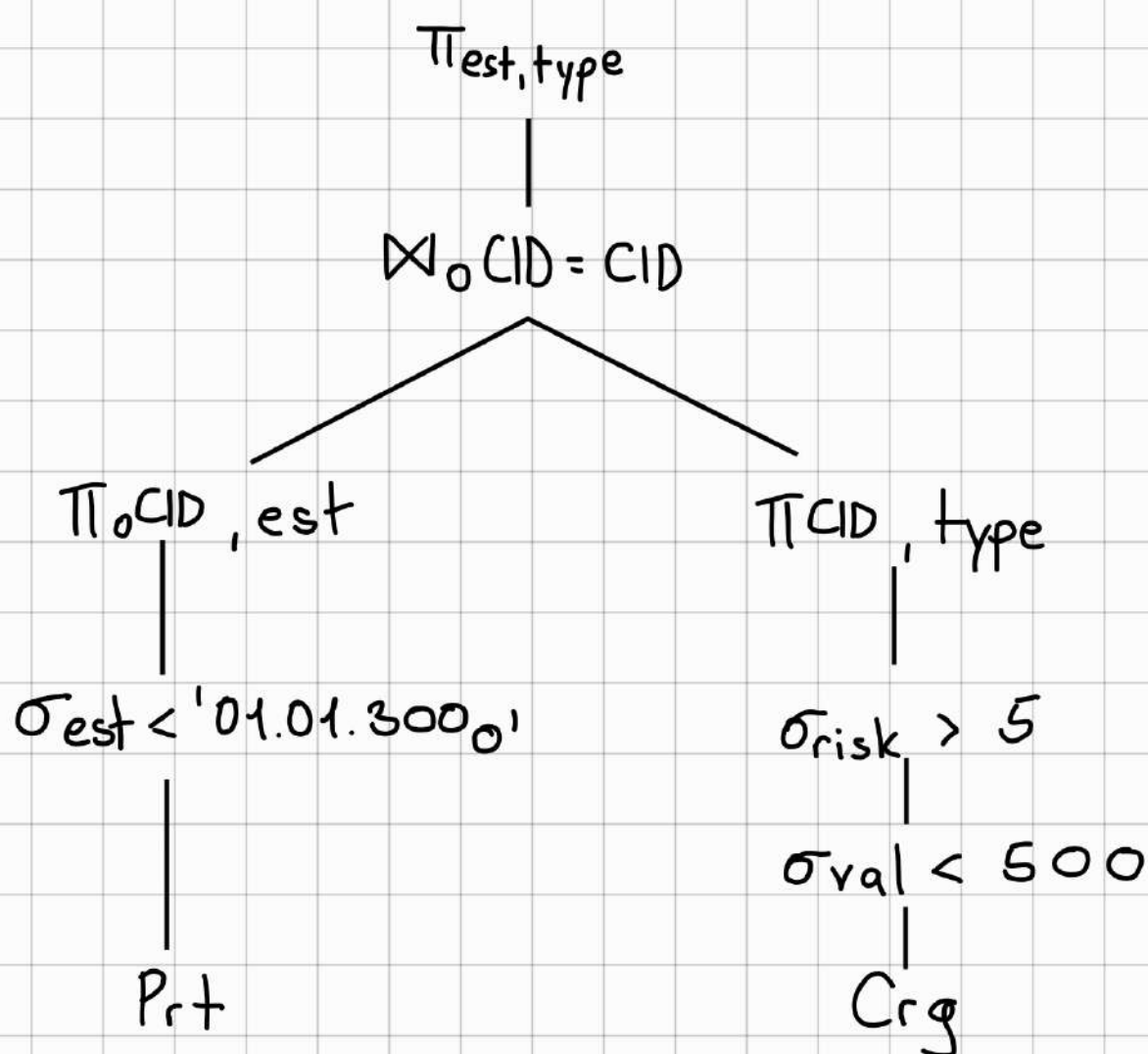
Im Vergleich zum Original keine redundante Informationen

$$2. \pi_{\text{est,type}}(\sigma_{\text{risk}} > 5 \wedge \text{est} < '01.01.3000' (\\ \text{Prt} \bowtie_{\text{CID}=\text{CID}} \wedge \text{val} < 500 \text{ Crg}))$$

Original



Improved

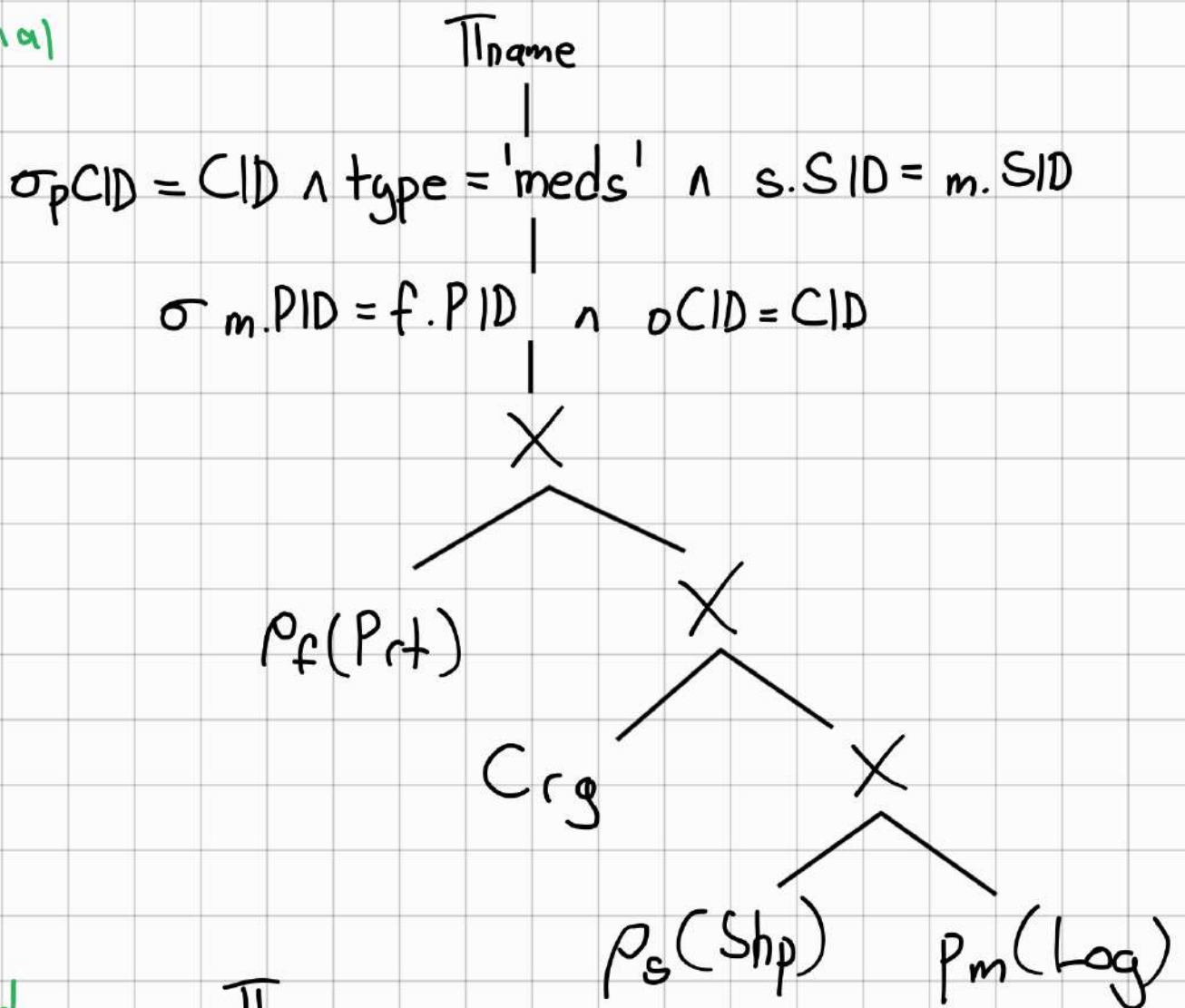


$$\pi_{\text{est,type}} \left(\pi_{\text{CID,est}} \left(\sigma_{\text{est}} < '01.01.3000' (\text{Prt}) \right) \right. \\ \left. \bowtie_{\text{CID}=\text{CID}} \right.$$

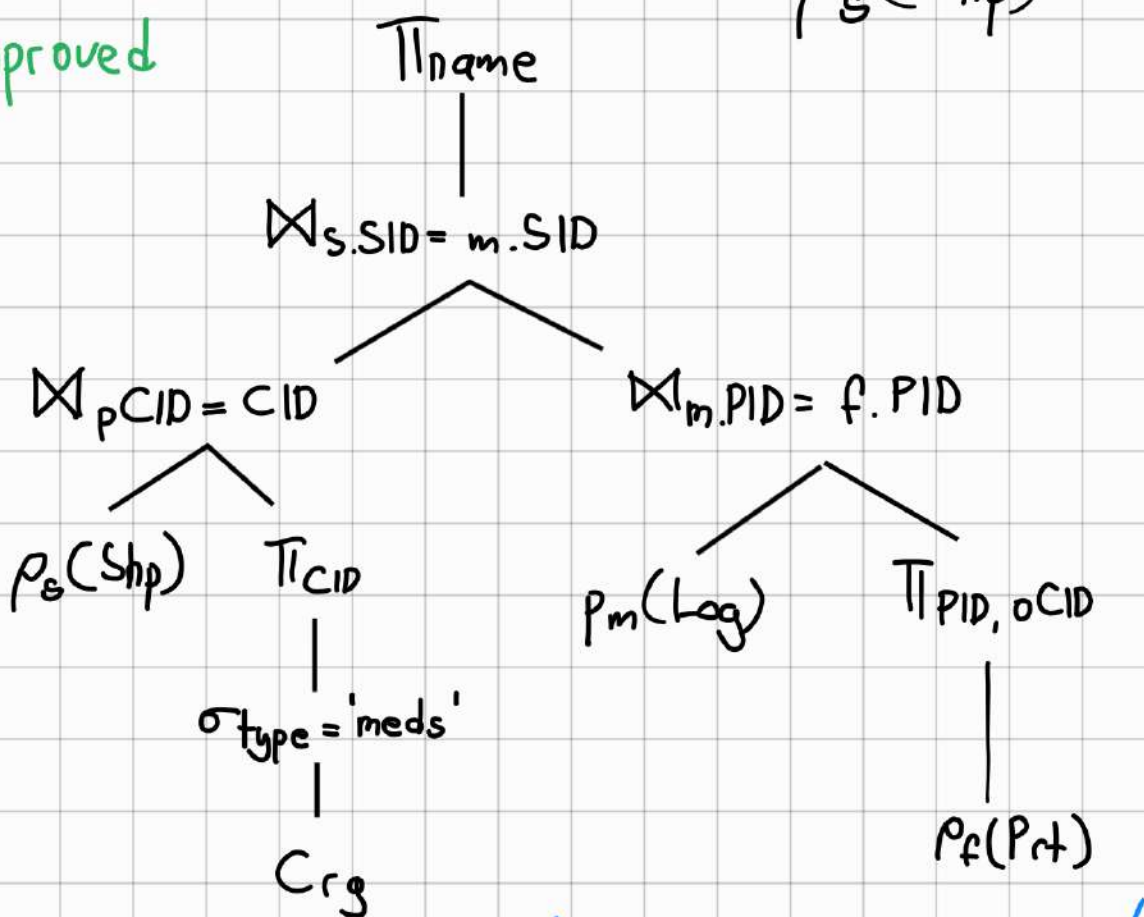
$$\left. \pi_{\text{CID,type}} \left(\sigma_{\text{risk}} > 5 \left(\sigma_{\text{val}} < 500 (\text{Crg}) \right) \right) \right)$$

3. $\pi_{\text{name}}(\sigma_{pCID=CID \wedge \text{type}='meds' \wedge s.SID = m.SID \wedge m.PID=f.PID \wedge oCID=CID}(\rho_f(\text{Prt}) \times \text{Crg} \times \rho_s(\text{Shp}) \times \rho_m(\text{Log})))$

Original



Improved



Kleinere Tabellen zuerst joinen
eventuell schneller

$\pi_{\text{name}} \left(\rho_s(\text{Shp}) \bowtie_{pCID=CID} \left(\pi_{CID} \left(\sigma_{\text{type}='meds'} \left(\text{Crg} \right) \right) \right) \right. \\ \left. \bowtie_{s.SID=m.SID} \left(\rho_m(\text{Log}) \bowtie_{m.PID=f.PID} \left(\pi_{PID, oCID} \left(\rho_f(\text{Prt}) \right) \right) \right) \right)$

2. $\pi_{\text{est,type}}(\sigma_{\text{risk} > 5 \wedge \text{est} < '01.01.3000'}(\text{Prt} \bowtie_{\text{oCID}=\text{CID} \wedge \text{val} < 500} \text{Crg}))$
3. $\pi_{\text{name}}(\sigma_{\text{pCID}=\text{CID} \wedge \text{type}='meds' \wedge \text{s.SID} = \text{m.SID} \wedge \text{m.PID}=\text{f.PID} \wedge \text{oCID}=\text{CID}}(\rho_f(\text{Prt}) \times \text{Crg} \times \rho_s(\text{Shp}) \times \rho_m(\text{Log})))$

Crg			
<u>CID</u>	type	risk	val
1	ore	7	450
2	fuel	3	200
3	meds	1	850

Shp		
<u>SID</u>	name	pCID → Crg
101	Zephyr	1
102	Orion	2
103	Vega	3
104	Sirius	3
105	Draco	2

Rte		
<u>RID</u>	desig	aPID → Prt
71	Sol-Mars	501
76	AlphaC-ProxB	500
73	SiriusRun	503
77	CoreWorlds	502

Prt		
<u>PID</u>	est	oCID → Crg
500	01.09.2904	3
501	01.01.2965	2
502	28.05.3009	1
503	01.10.2933	3

Log	
<u>SID</u>	<u>PID</u>
102	503
105	500
102	501

Man	
<u>SID</u>	<u>RID</u>
101	71
102	73
103	77
105	76
102	71

4 Logical Query Optimization

Gegeben ist die folgende SQL-Abfrage an eine Datenbank der Forschungslabors (labs) eines Instituts.

```
SELECT L1.lName
FROM Lab L1, Lab L2
WHERE L1.pubs > L2.pubs AND L2.city = "Kyoto"
```

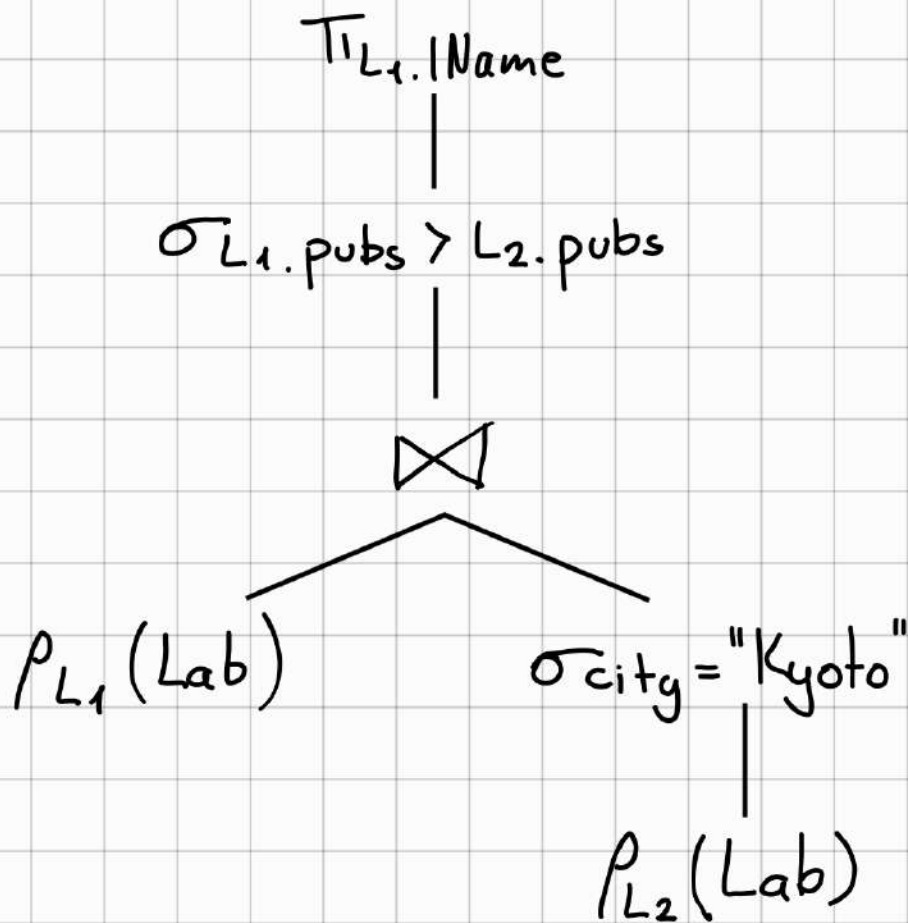
Schreiben Sie einen effizienten relationalen Algebra-Ausdruck, der dieser Abfrage entspricht. Begründen Sie Ihre Antwort.

4 Logical Query Optimization

Gegeben ist die folgende SQL-Abfrage an eine Datenbank der Forschungslabors (labs) eines Instituts.

```
SELECT L1.lName  
FROM Lab L1, Lab L2  
WHERE L1.pubs > L2.pubs AND L2.city = "Kyoto"
```

Schreiben Sie einen effizienten relationalen Algebra-Ausdruck, der dieser Abfrage entspricht. Begründen Sie Ihre Antwort.



$\pi_{L_1.lName} \left(\sigma_{L_1.pubs > L_2.pubs} \left(\rho_{L_1}(Lab) \bowtie \left(\sigma_{city = "Kyoto"}(\rho_{L_2}(Lab)) \right) \right) \right)$

Alle Labore L_1 , die mehr Publikationen als
den Laboren L_2 aus Kyoto

Besser: JOIN > Kreuzprodukt

σ Selektion so früh wie möglich

5 Left-Deep Trees vs. Bushy Trees

Neben der Reihenfolge der Joins müssen wir auch die allgemeine Struktur der Query Pläne berücksichtigen. Left-Deep und Bushy Join-Trees (Abbildung 1) sind die gängigsten Varianten, die während der Optimierung betrachtet werden.

Bitte diskutieren Sie die Vor- und Nachteile jeder Variante, insbesondere in Bezug auf die parallele Verarbeitung.

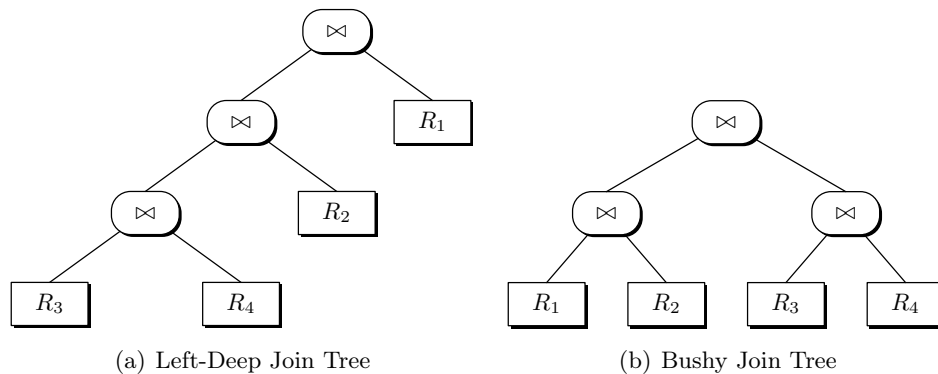


Figure 1: Join Trees

a) Die JOINS warten auf einander, weshalb parallele Verarbeitung nicht möglich ist \ominus

Leichter zu überlegen, da es sequentiell aufgebaut ist. \oplus

b) Die Subbäume können \oplus parallel verarbeitet werden

Potenziell kleinere Zwischenergebnisse durch geschickte Gruppierung \oplus

Komplexere Umsetzung \ominus (aber dafür bessere Performance)

Index Nested Join Algorithmus
effizient bei der Struktur \oplus