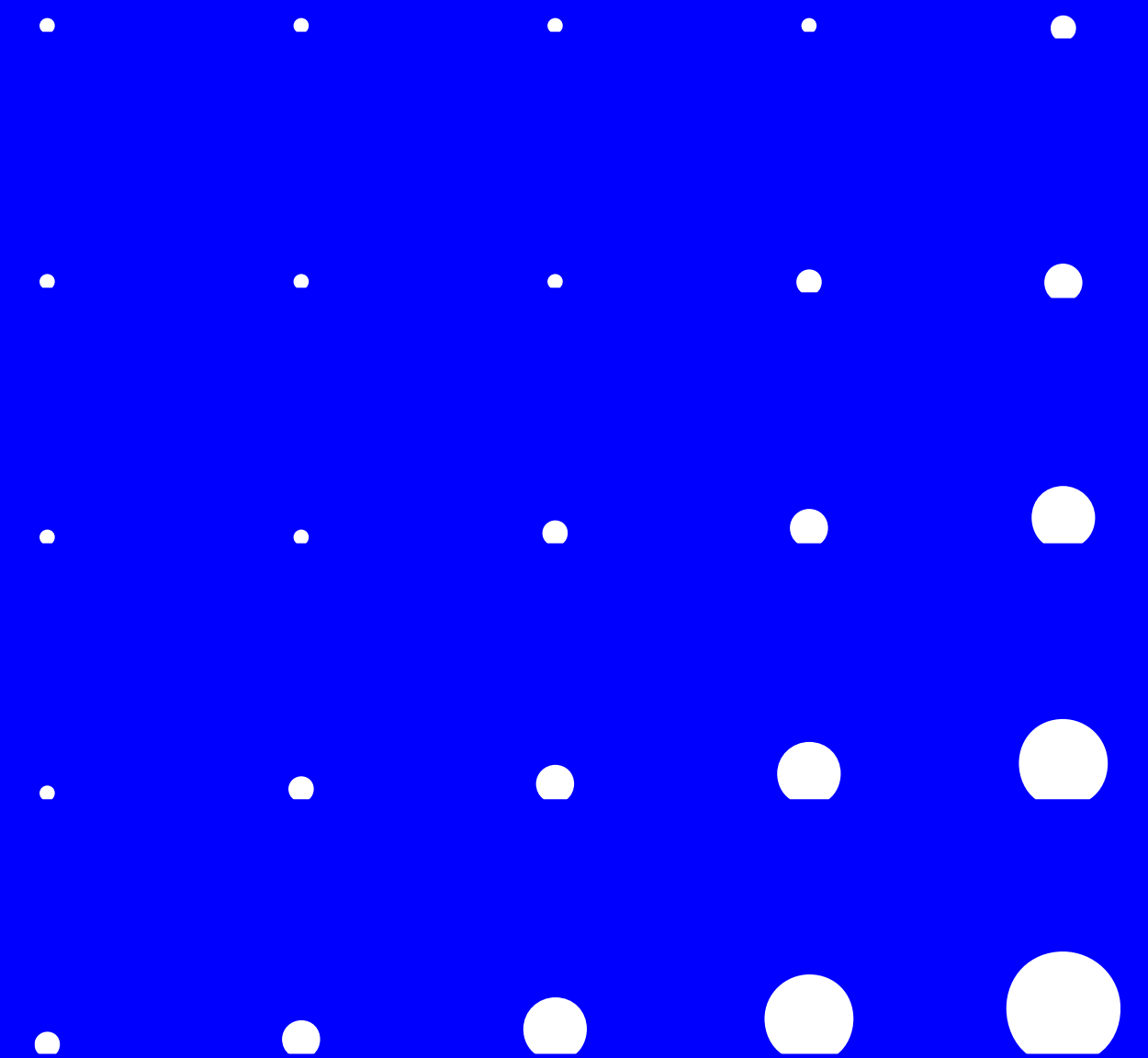


Relationenalgebra: Operatorbaum & Beispiele



Warum Baumdarstellung?

- Komplexere Ausdrücke können auch als Baum dargestellt werden!
- Sie werden dann von unten nach oben gelesen.
- Unäre Operationen haben einen Input-Zweig mit Relationen, binäre Operationen entsprechend zwei.
- Die Baumdarstellung ist oft besser nachvollziehbar.

Beispiel für Operatorbäume

- Aufgabe:
Finde die Titel und Nummern aller Vorlesungen, die der Student Xenokrates gehört hat!

Vorlesungen			
VorlNr	Titel	SWS	Gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Beispiel für Operatorbäume

- Aufgabe:
Finde die Titel und Nummern aller Vorlesungen, die der Student Xenokrates gehört hat!

Vorlesungen			
VorlNr	Titel	SWS	Gelesen Von

hören	
MatrNr	VorlNr

Studenten		
MatrNr	Name	Semester

```
SELECT VorlNr, Titel FROM Vorlesungen
```

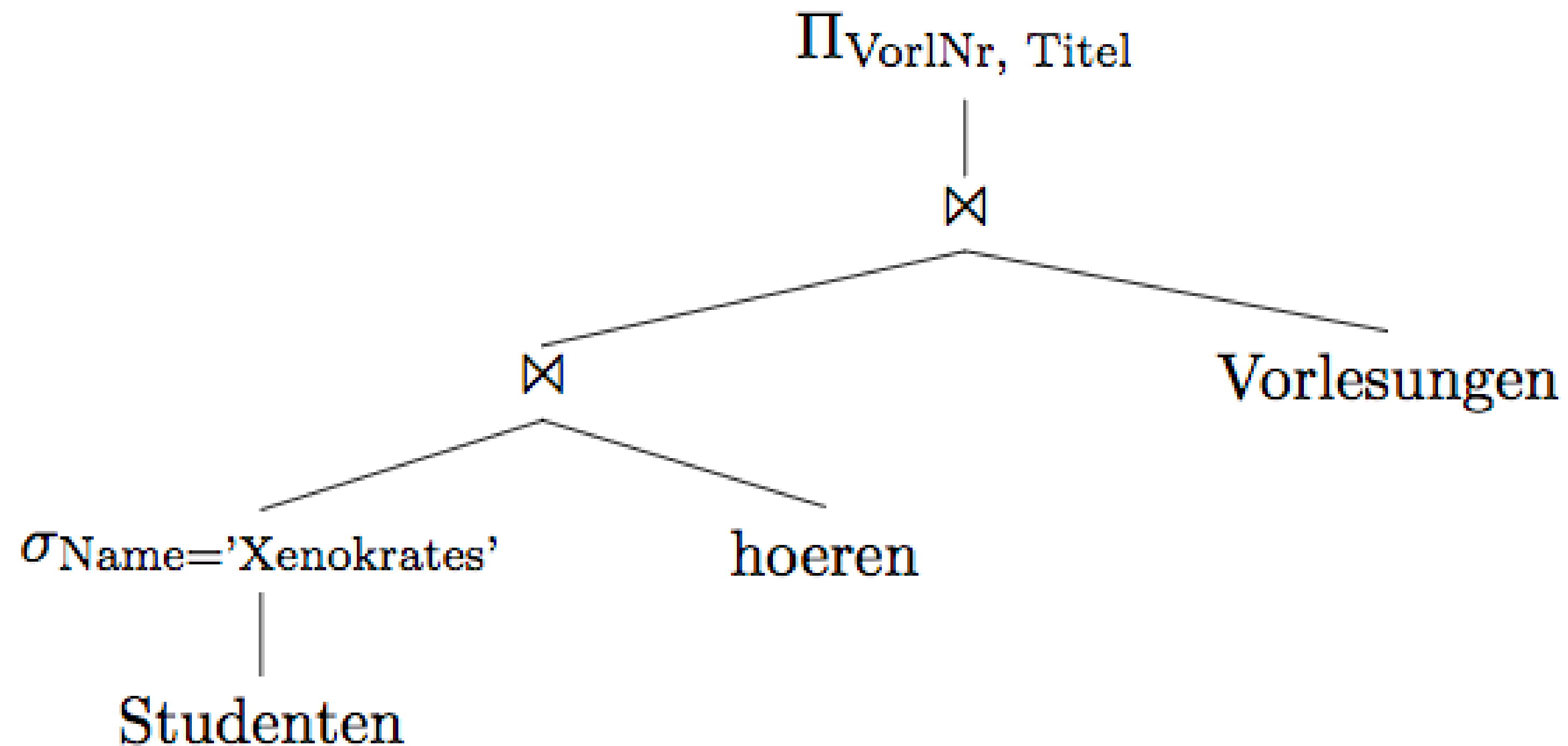
```
NATURAL JOIN hören
```

```
NATURAL JOIN (SELECT * FROM Studenten WHERE Name="Xenokrates")
```

```
 $\Pi_{\text{VorlNr, Titel}}(\text{Vorlesungen} \bowtie (\text{hören} \bowtie (\sigma_{\text{Name}='Xenokrates'}(\text{Studenten}))))$ 
```

Beispiel für Operatorbäume

- Aufgabe:
Finde die Titel und Nummern aller Vorlesungen, die der Student Xenokrates gehört hat!

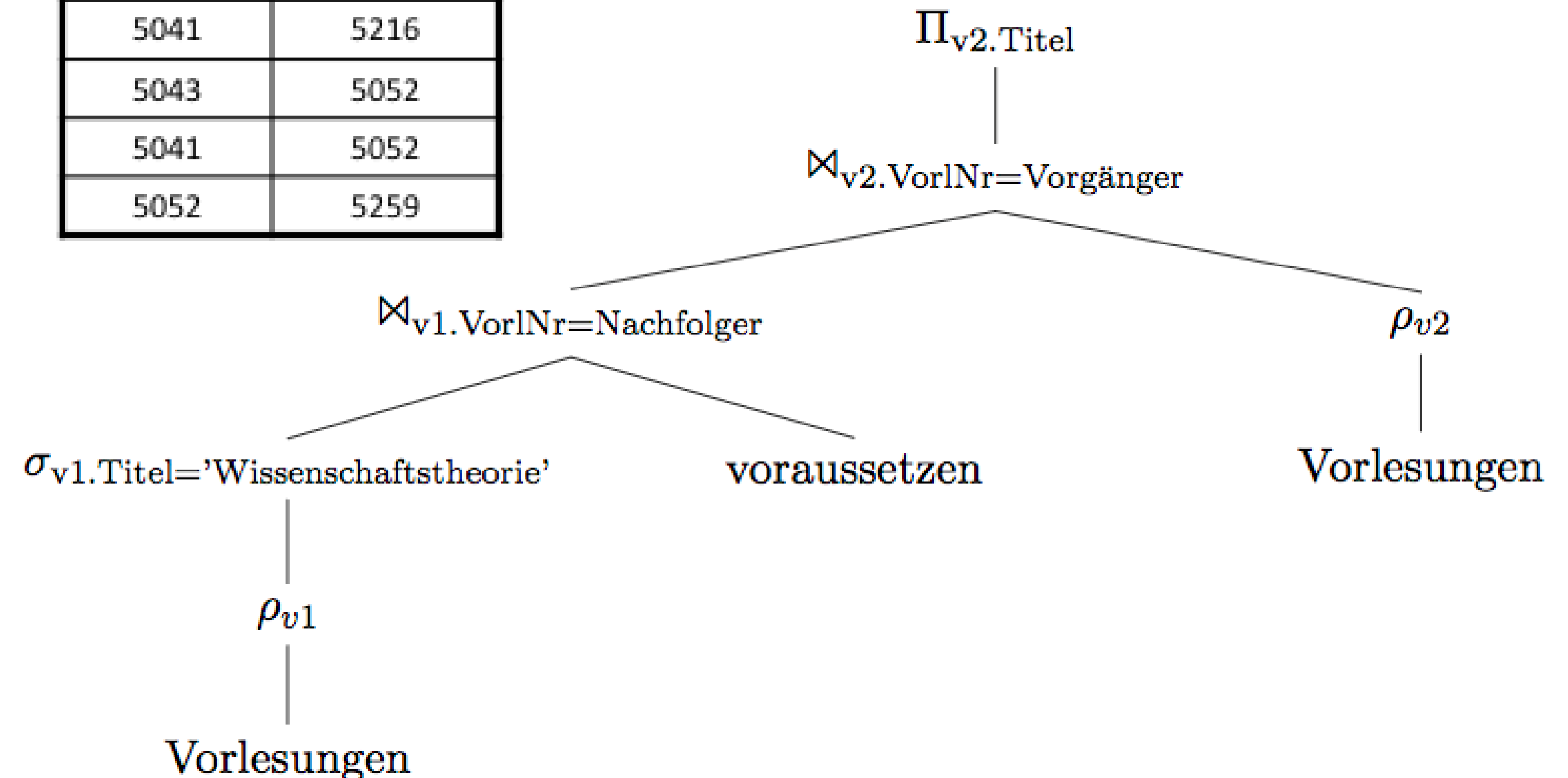


$\Pi_{\text{VorlNr}, \text{Titel}}(\text{Vorlesungen} \bowtie (\text{ hoeren} \bowtie (\sigma_{\text{Name}='Xenokrates'}(\text{Studenten}))))$

Was soll hier gesucht werden? Und wie lautet hier der SQL-Befehl?

Vorlesungen			
VorlNr	Titel	SWS	Gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259



Beispiele zur Relationenalgebra

Beispiele zu Abfragen: Kunde, Lieferant und Ware

- Als Relationenschemata für die Beispiele verwenden wir eine Beispieldatenbank bestehend aus den Schemata Kunde, Lieferant und Ware.
- Die Schemata seien:
 - KUNDE(Kundenr, Name, Wohnort, Kontostand)
 - LIEFERANT(Lieferantenr, Name, Ort, Telefon)
 - WARE(Warenr, Bezeichnung, Lieferantenr, Preis)

Beispiele zu Abfragen: Kunde, Lieferant und Ware

- Grundoperationen der relationalen Algebra werden dann so benutzt:
 - Die Preise aller Waren:
 $\pi_{\text{Bezeichnung, Preis}}(\text{WARE})$
 - Alle Lieferanten aus Bremen:
 $\sigma_{\text{Ort} = \text{'Bremen'}}(\text{LIEFERANT})$
 - Kunden mit negativem Kontostand:
 $\sigma_{\text{Kontostand} < 0}(\text{KUNDE})$
 - Ort von LIEFERANT umbenennen:
 $\rho_{[\text{Wohnort} \leftarrow \text{Ort}]}(\text{LIEFERANT})$

Beispiele zu Abfragen: Kunde, Lieferant und Ware

- Da die Ergebnisse der relationalen Algebra wieder Relationen sind, können die Operationen wieder auf die Ergebnisse von Operationen angewendet werden.
 - Dies erlaubt komplexe Abfragen.
- Für eine einfachere Schreibweise nehmen wir an, dass das Kreuzprodukt eine implizite Umbenennung der Attribute vornimmt, so dass die neuen Attributnamen mit dem Relationennamen qualifiziert sind, d. h. aus Lieferantennr aus der Relation WARE wird WARE.Lieferantennr.

Beispiele zu Abfragen: Kunde, Lieferant und Ware

- Die Telefonnummern aller Lieferanten, die Gemüse in Bremen liefern:
 - $\pi_{\text{Telefon}}(\sigma_{\text{Bezeichnung}='Gemüse' \wedge \text{Ort}='Bremen' \wedge \text{LIEFERANT.Lieferantenrnr}=\text{WARE.Lieferantenrnr}}(\text{LIEFERANT} \times \text{WARE}))$
- Alle Orte, die wenigstens einen Lieferanten und wenigstens einen Kunden enthalten:
 - $\pi_{\text{Ort}}(\rho_{\text{Ort} \leftarrow \text{Wohnort}}(\text{KUNDE})) \cap \pi_{\text{Ort}}(\text{LIEFERANT})$

Beispiele zu Abfragen: Mitarbeiter & Abteilungen

Mitarbeiter

PNr	Name	Vorname	Abteilung
001	Huber	Erwin	01
002	Mayer	Hugo	01
003	Müller	Anton	02

$\sigma_{\text{Abteilung}=01}(\text{Mitarbeiter})$

PNr	Name	Vorname	Abteilung
001	Huber	Erwin	01
002	Mayer	Hugo	01

Beispiele zu Abfragen: Mitarbeiter & Abteilungen

Mitarbeiter \times Abteilungen

PNr	Name	Vorname	Abteilung	ANr	Abteilungsname
001	Huber	Erwin	01	01	Buchhaltung
001	Huber	Erwin	01	02	Produktion
002	Mayer	Hugo	01	01	Buchhaltung
002	Mayer	Hugo	01	02	Produktion
003	Müller	Anton	02	01	Buchhaltung
003	Müller	Anton	02	02	Produktion

$\sigma_{\text{Abteilung}=\text{ANr}}$ (Mitarbeiter \times Abteilungen)

PNr	Name	Vorname	Abteilung	ANr	Abteilungsname
001	Huber	Erwin	01	01	Buchhaltung
002	Mayer	Hugo	01	01	Buchhaltung
003	Müller	Anton	02	02	Produktion

Beispiele zu Abfragen: Mitarbeiter & Abteilungen in SQL

select * from Mitarbeiter m, Abteilungen a...

PNr	Name	Vorname	m.ANr	a.ANr	Abteilungsname
001	Huber	Erwin	01	01	Buchhaltung
001	Huber	Erwin	01	02	Produktion
002	Mayer	Hugo	01	01	Buchhaltung
002	Mayer	Hugo	01	02	Produktion
003	Müller	Anton	02	01	Buchhaltung
003	Müller	Anton	02	02	Produktion

...where m.ANr = a.ANr

PNr	Name	Vorname	m.ANr	a.ANr	Abteilungsname
001	Huber	Erwin	01	01	Buchhaltung
002	Mayer	Hugo	01	01	Buchhaltung
003	Müller	Anton	02	02	Produktion

Beispiele zu Abfragen: Städte, Länder und Parteien

Gegeben sei folgendes Relationenschema:

Städte (SName: String, SEinw: Integer, Land: String)
Länder (LName: String, LEinw: Integer, Partei*: String)

* bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

-
- Bestimme alle Großstädte (≥ 500.000) und ihre Einwohner

$$\pi_{\text{SName}, \text{SEinw}}(\sigma_{\text{SEinw} \geq 500.000}(\text{Städte}))$$

- In welchem Land liegt die Stadt Passau?

$$\pi_{\text{Land}}(\sigma_{\text{SName}=\text{Passau}}(\text{Städte}))$$

- Bestimme die Namen aller Städte, deren Einwohnerzahl die eines beliebigen Landes übersteigt:

$$\pi_{\text{SName}}(\sigma_{\text{SEinw} > \text{LEinw}}(\text{Städte} \times \text{Länder}))$$

Beispiele zu Abfragen: Städte, Länder und Parteien

Gegeben sei folgendes Relationenschema:

Städte (SName: String, SEinw: Integer, Land: String)

Länder (LName: String, LEinw: Integer, Partei*: String)

* bei Koalitionsregierungen: jeweils eigenes Tupel pro Partei

-
- Finde alle Städtenamen in CDU-regierten Ländern

$$\pi_{\text{SName}}(\sigma_{\text{Land=LName}}(\text{Städte} \times \sigma_{\text{Partei=CDU}}(\text{Länder})))$$

oder auch:

$$\pi_{\text{SName}}(\sigma_{\text{Land=LName} \wedge \text{Partei=CDU}}(\text{Städte} \times \text{Länder}))$$

- Welche Länder werden von der SPD *allein* regiert

$$\pi_{\text{LName}}(\sigma_{\text{Partei=SPD}}(\text{Länder})) - \pi_{\text{LName}}(\sigma_{\text{Partei} \neq \text{SPD}}(\text{Länder}))$$

Einfache Beispiele zu unserer Aufgabenstellung

- Alle Rechnungen eines bestimmten Kunden
 - **SELECT * FROM Rechnung WHERE kundennummer = 101;**
 - $\sigma_{\text{kundennummer}=101}(\text{Rechnung})$
- Alle Bestellungen, die vor 2023 aufgegeben wurden
 - **SELECT * FROM Bestellung WHERE datum < '2023-01-01';**
 - $\sigma_{\text{datum}<'2023-01-01'}(\text{Bestellung})$
- Alle Produkte mit einem Preis unter 50 Euro
 - **SELECT * FROM Produkt WHERE preis < 50;**
 - $\sigma_{\text{preis}<50}(\text{Produkt})$
- Alle offenen Rechnungen für den Kunden mit kundennummer = 101
 - **SELECT * FROM Rechnung
WHERE rechnungsstatus = 'offen' AND kundennummer = 101;**
 - $\sigma_{\text{Rechnungsstatus}='offen' \wedge \text{Kundennummer}=101}(\text{Rechnung})$

Projektion

- Namen und Preise aller Produkte
 - **SELECT DISTINCT name, preis FROM Produkt;**
 - $\pi_{\text{name,preis}}(\text{Produkt})$
- Kunden-IDs und zugehörige Adressen
 - **SELECT DISTINCT kundennummer, id_adresse FROM Kunde;**
 - $\pi_{\text{kundennummer, id_adresse}}(\text{Kunde})$
- Alle Produkte mit einem Preis unter 50 Euro
 - **SELECT * FROM Produkt WHERE preis < 50;**
 - $\sigma_{\text{preis} < 50}(\text{Produkt})$

- Kunden mit ihren Rechnungen (Natural Join)
 - **SELECT k.kundennummer, k.adresse_id, r.rechnungsnummer, r.status**
FROM Kunde k JOIN Rechnung r
ON k.kundennummer = r.kundennummer;
 - $\text{Kunde} \bowtie_{\text{Kunde.kundennummer=Rechnung.kundennummer}} (\text{Rechnung})$
- Rechnungspositionen mit Produktdetails
 - **SELECT rp.rechnungsnummer, rp.menge, p.name, p.preis**
FROM Rechnungsposition rp JOIN Produkt p
ON rp.produktnummer = p.produktnummer;
 - $\text{Rechnungsposition} \bowtie_{\text{Rechnungsposition.produktnummer=Produkt.produktnummer}} (\text{Produkt})$

Kreuzprodukt

- Kombination aus Produkten und Lieferanten
 - **SELECT * FROM Produkt, Lieferant;**
 - Produkt × Lieferant
- Kombination aus Rechnungen und Bestellungen
 - **SELECT * FROM Rechnung, Bestellung;**
 - Rechnung × Bestellung