

Chapter 6: Relational Algebra

Content:

- Relational algebra: querying the relational schema

Next:

- SQL

Relational Algebra

π Projection

σ Selection

ρ Renaming

\times Cartesian Product

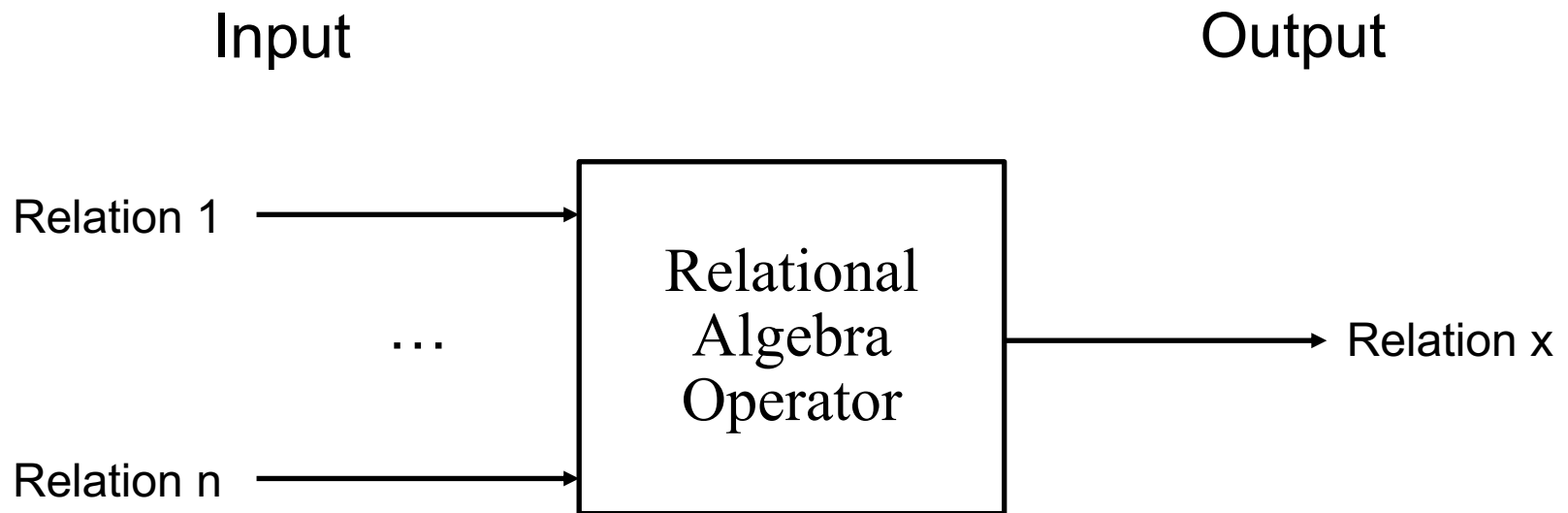
\bowtie Join

\cup Union

\cap Intersection

Relational Algebra

Each relational algebra operator takes a number of relations as an input and outputs one relation.

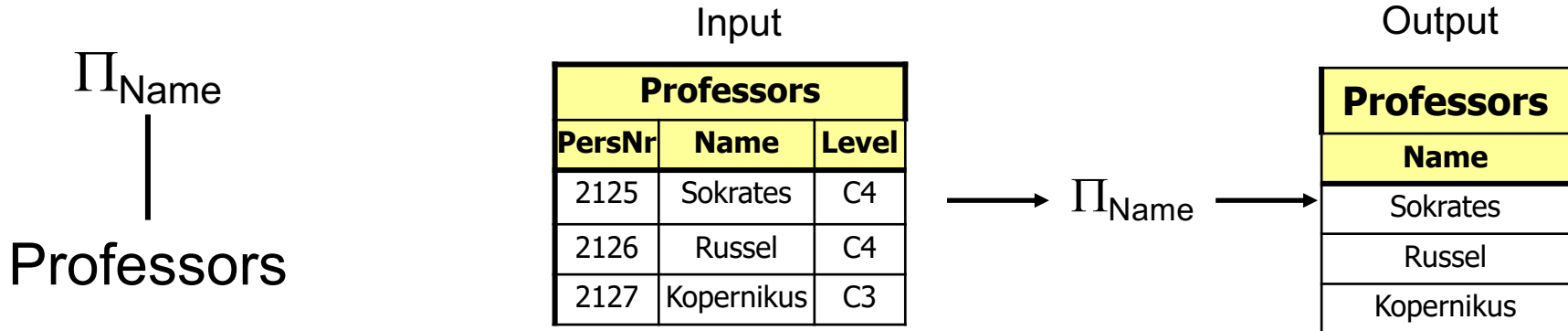


Projection π

- Takes one relation as an input.
- Alters the schema of the input relation by only producing the specified attributes $A_1, A_2.. A_n$ in the output relation.

Example: Find the *Name* of all professors.

Solution: $\Pi_{\text{Name}}(\text{Professors})$

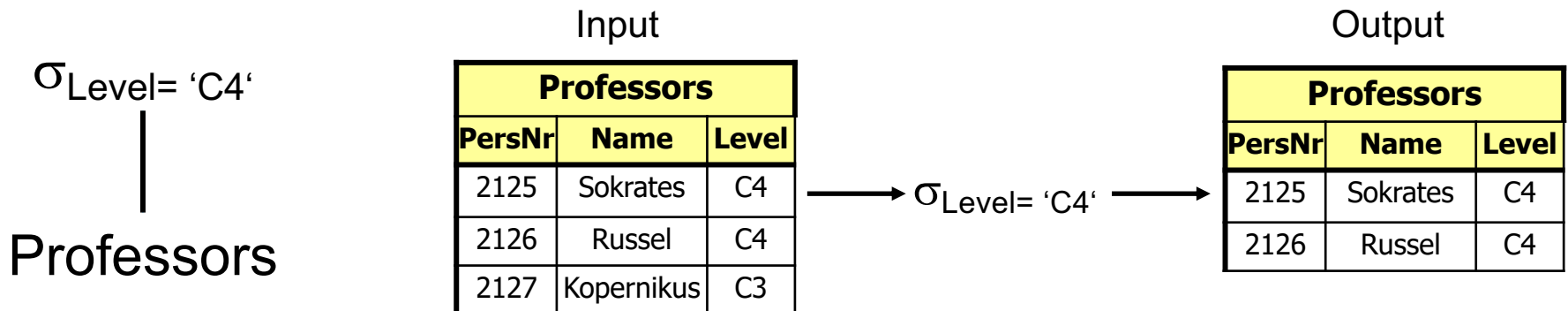


Selection σ

- Takes one relation as an input.
- The output relation only contains tuples fulfilling the specified condition.

Example: Find all professors with level 'C4'.

Solution: $\sigma_{\text{Level}='C4'}(\text{Professors})$

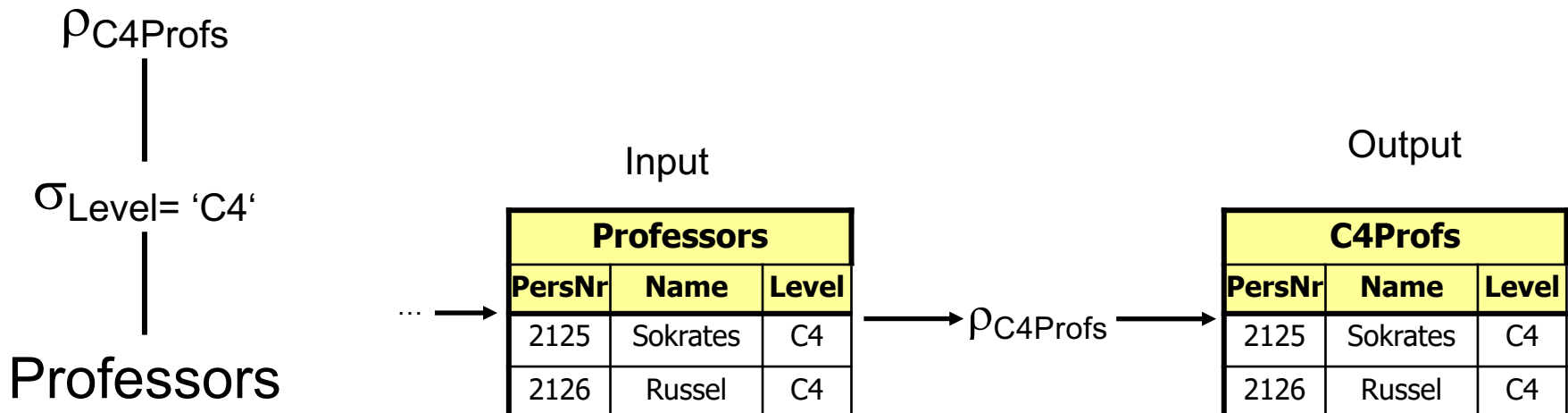


Renaming ρ

- Takes one relation as an input.
- Outputs the same relation with a different name.

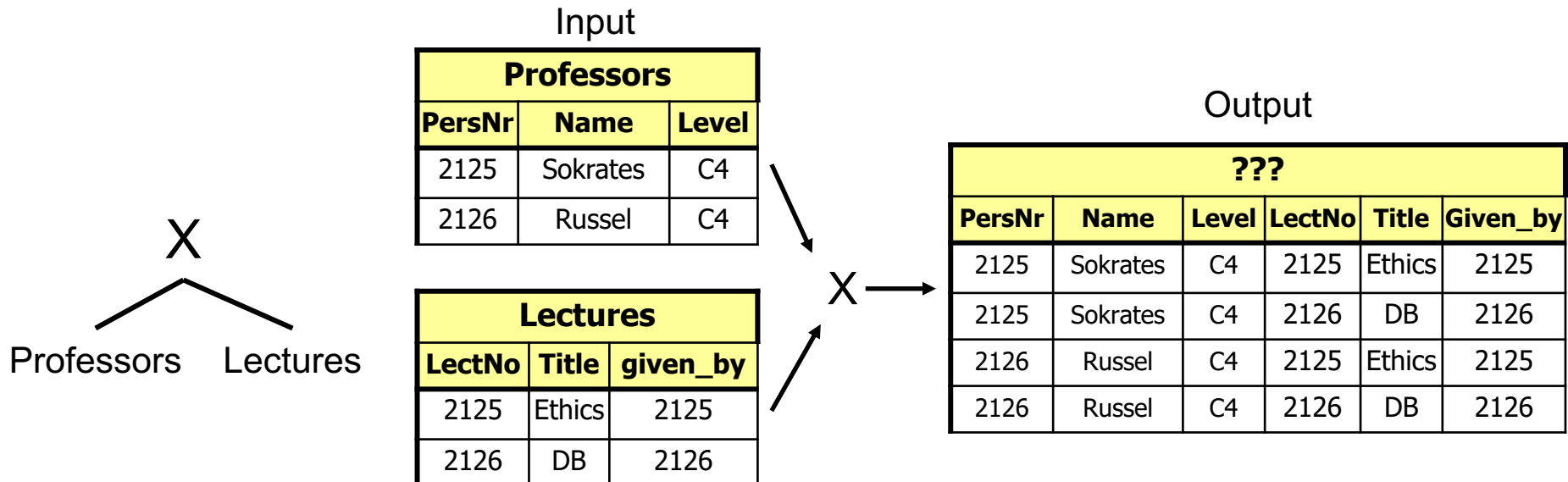
Example: Output all C4 professors in a relation named C4Profs.

Solution: $\rho_{C4Profs}(\sigma_{Level='C4'}(Professors))$



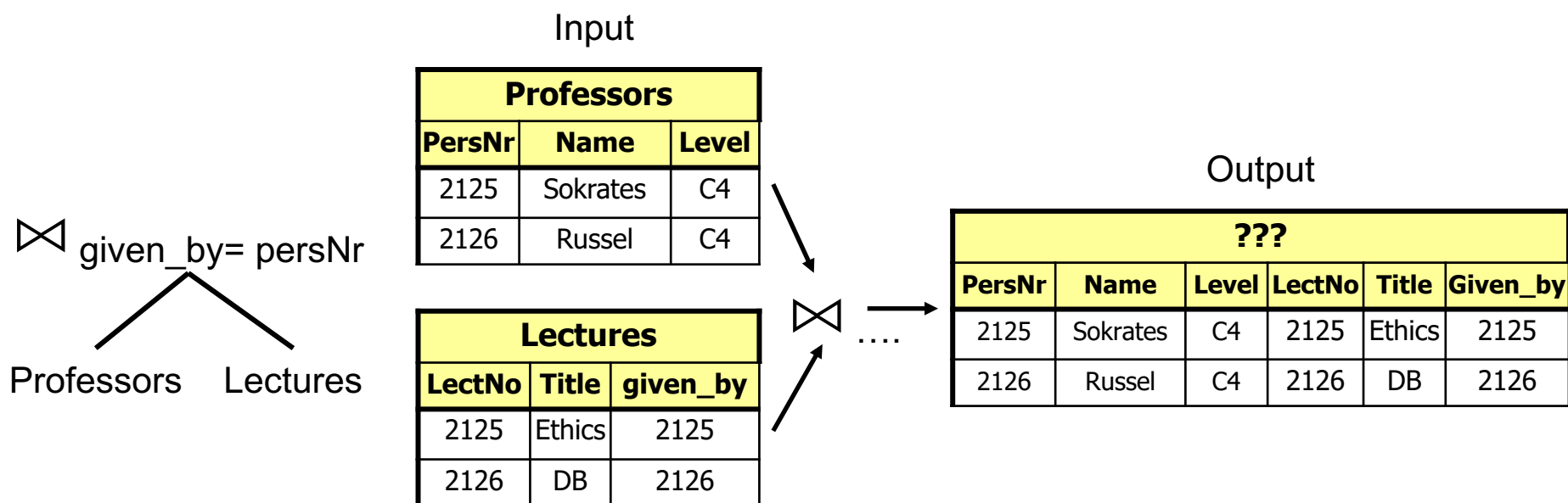
Cross/Cartesian Product X

- Takes n relations as an input.
- Outputs the cartesian product of the input relations
- $A \times B = \{(a, b) \mid a \in A, b \in B\}$



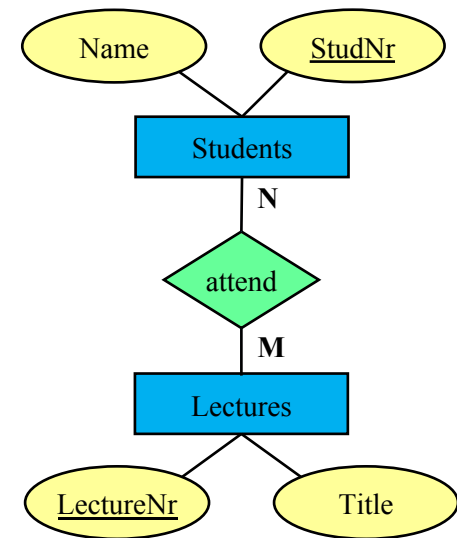
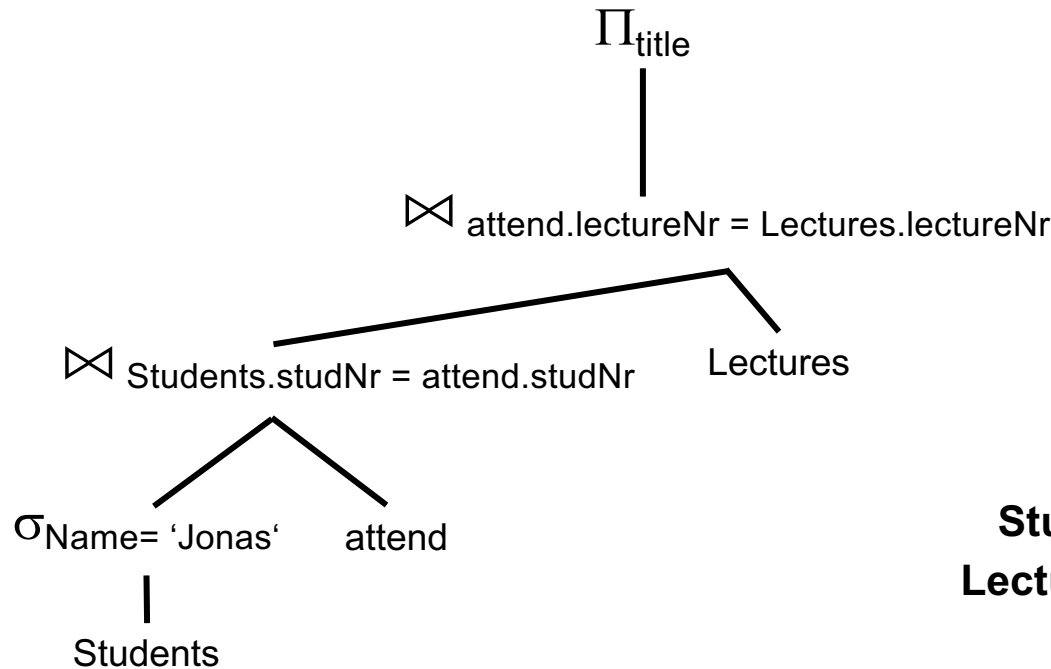
Join ⋈

- Takes n relations as an input.
- Cross product is very common in combination with a selection, the join is a short cut for that:
- $\sigma_{\text{given_by}=\text{persNr}}(\text{Professors} \times \text{Lectures}) = \text{Professors} \bowtie_{\text{given_by}=\text{persNr}} \text{Lectures}$



Relational Algebra: Example

Find the name all lectures the student named Jonas is attending.



Students: $\{[\text{StudNr}, \text{Name}, \text{Semester}]\}$
Lectures: $\{[\text{LectureNr}, \text{Title}, \text{WeeklyHours}]\}$
attend: $\{[\text{StudNr}, \text{LectureNr}]\}$

Other Relational Languages

- Relational Algebra
- Relational Tuple Calculus
- Relational Domain Calculus

Relational Tuple Calculus (Example)

Students who attend at least one lecture of Curie

$$\{s \mid s \in \text{Students} \\ \wedge \exists h \in \text{attend}(s.\text{StudNr}=h.\text{StudNr} \\ \wedge \exists v \in \text{Lectures}(h.\text{LectureNr}=v.\text{LectureNr} \\ \wedge \exists p \in \text{Professors}(p.\text{PersNr}=v.\text{Given_by} \\ \wedge p.\text{Name} = \text{'Curie'})))))\}$$

Relational Domain Calculus (Example)

Query in the domain calculus is of the form:

$$\{[v1, v2, \dots, vn] \mid P(v1, \dots, vn)\}$$

with $v1, \dots, v2$ domain variables and P predicate

Example:

StudNr and Name of the testees of Sokrates

$$\{[m, n] \mid \exists ([m, n, s] \in \text{Students} \\ \wedge \exists p, v, g ([m, p, v, g] \in \text{test} \\ \wedge \exists a, r, b ([p, a, r, b] \in \text{Professors} \\ \wedge a = \text{'Sokrates'})))]\}$$

Expressive Power

The three languages

- Relational Algebra
- Tuple Relational Calculus (restricted to safe expressions)
- Domain Relational Calculus (restricted to safe expressions)

are equal in their expressive power

$\{n \mid \neg(n \in \text{Professors})\}$ e.g. is not safe, as the result is infinite