Relational Algebra

Query Languages and the Relational Model

- **Query Languages allow retrieval and manipulation of data from a Database.**

- **In particular, we will look at the Relational Model:**
  - Logical foundation based on mathematical logic.
  - High query optimization possible

- **A Query language is DIFFERENT from a programming language:**
  - it is not in general Turing Complete.
  - must allow easy, fast, efficient access to (large) datasets.
  - it is not designed for complex calculations.

Query Languages and the Relational Model

Formally, there are two mathematically defined languages upon which real query languages are built:

- **RELATIONAL ALGEBRA:**
  Used for describing the execution plan
- **RELATIONAL CALCULUS**
  Used by users to describe **what** they want, but not the **how**.

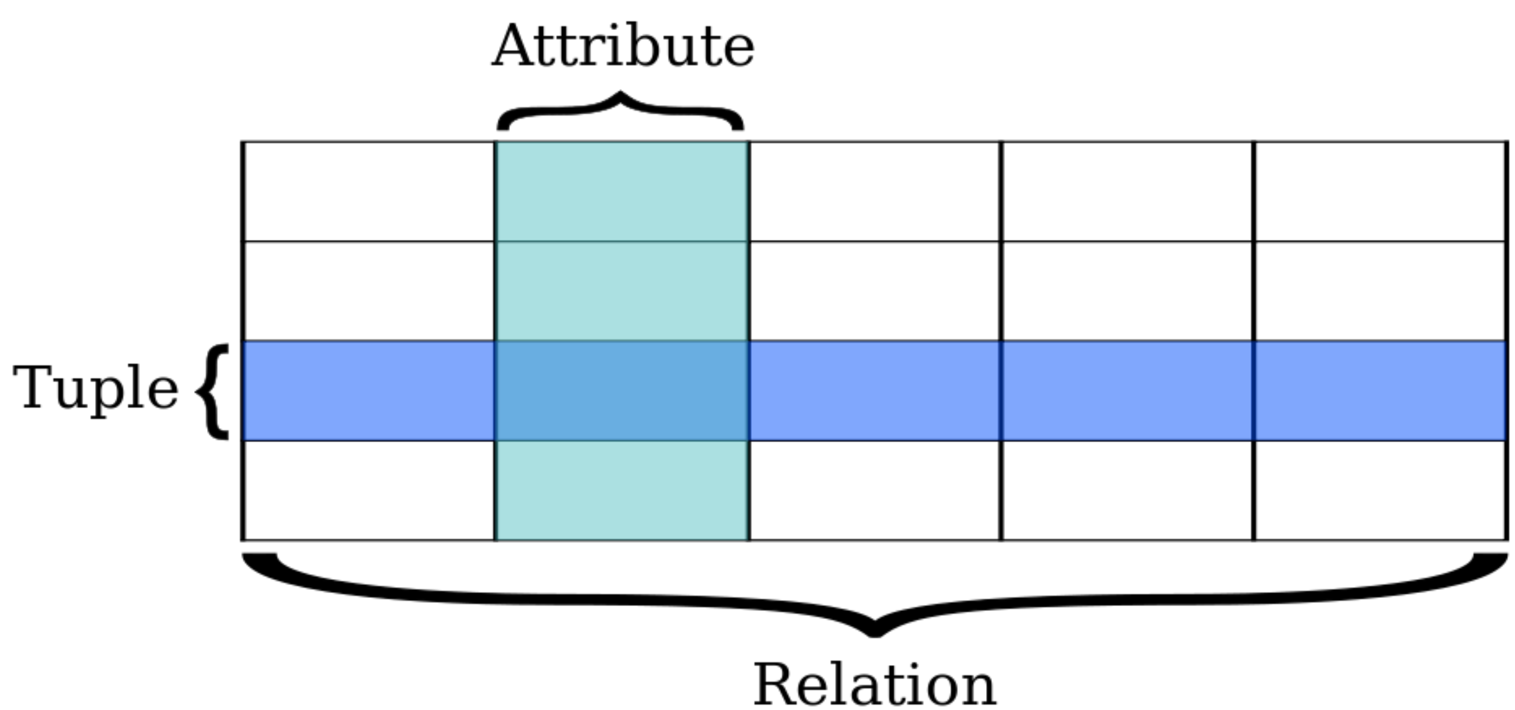
A real implementation of the above concepts, is e.g. the SQL language. (Standard Query Language).

Relations and Queries

In a relational database model, all the data is represented in **tuples**.
The tuples are organized in **relations**.

Definition:

- A relation is a set of tuples where each element of them is a member of a data domain. (or it is an attribute).



In the SQL language, an instance of a relation, is called a **Table**.

A **query** applied to a relation instance, produces a new relation instance.

Relational Algebra

- **Selection**: σ selects a subset of rows from a relation
- **Projection**: π Deletes columns from a relation
- **Cross Product**: $\times$ Combines two relations
- **Set Difference**: $-$ Tuples in relation1 and not in relation 2.
- **Union/Inters.**: $\cup$ $\cap$ Tuples in relation1 and in relation 2.
- **Join**: $\bowtie$ Composite Operation
- **Division: /** Composite Operation

Every basic operation maps a relation into another relation, therefore the operations can be composed.
In more mathematical words, the relational algebra is closed.

Selection σ

Selects the rows which satisfy a condition.
No duplicates in the result.

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
3	Simon Chan	21	Vancouver
4	Yuri Kirillov	57	Novosibirsk
5	Antonio Esposito	19	Napoli
6	Pierre Daveaux	76	Marseille
7	Antonia Caballero	33	Madrid
8	Alexa Fredrikson	37	Oslo
9	Jack Post	22	New York
10	Shintaro Ito	25	Osaka

$$\sigma_{Age<26}(T) =$$

ID	Name	Age	City
3	Simon Chan	21	Vancouver
5	Antonio Esposito	19	Napoli
9	Jack Post	22	New York
10	Shintaro Ito	25	Osaka

Projection π

Deletes attributes not present in the projection list.

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
3	Simon Chan	21	Vancouver
4	Yuri Kirillov	57	Novosibirsk
5	Antonio Esposito	19	Napoli
6	Pierre Daveaux	76	Marseille
7	Antonia Caballero	33	Madrid
8	Alexa Fredrikson	37	Oslo
9	Jack Post	22	New York
10	Shintaro Ito	25	Osaka

$$\sigma_{City,Age}(T) =$$

ID	Age	City
1	44	Kansas City
2	32	Portland
3	21	Vancouver
4	57	Novosibirsk
5	19	Napoli
6	76	Marseille
7	33	Madrid
8	37	Oslo
9	22	New York
10	25	Osaka

Cross-Product $\times$

Each row of T1 is paired with each row of T2.
If the tables have fields with the same name, it is possible to rename them with the renaming operator ρ.

T1

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
8	Alexa Fredrikson	37	Oslo

T2

ID	Date
1	29 Feb 2016
2	01 Feb 2016

$$\rho\left[1 \to id1, 5 \to id2, T1 \times T2\right] =$$

ID1	Name	Age	City	ID2	
1	John Smith	44	Kansas City	1	29 Feb 2016
1	John Smith	44	Kansas City	2	01 Feb 2016
2	Steven Miller	32	Portland	1	29 Feb 2016
2	Steven Miller	32	Portland	2	01 Feb 2016
8	Alexa Fredrikson	37	Oslo	1	29 Feb 2016
8	Alexa Fredrikson	37	Oslo	2	01 Feb 2016

Union, Intersection, Difference : $\cup \quad \cap \quad -$

Tables must be union-compatible:

- Same # of fields
- Corresponding fields with the same name.

T1

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
8	Alexa Fredrikson	37	Oslo
10	Shintaro Ito	25	Osaka
5	Antonio Esposito	19	Napoli

T2

ID	Name	Age	City
9	Jack Post	22	New York
8	Alexa Fredrikson	37	Oslo

$$T1 \cap T2$$

ID	Name	Age	City
8	Alexa Fredrikson	37	Oslo

$$T1 \cup T2$$

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
8	Alexa Fredrikson	37	Oslo
10	Shintaro Ito	25	Osaka
5	Antonio Esposito	19	Napoli
9	Jack Post	22	New York

$$T1 - T2$$

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
10	Shintaro Ito	25	Osaka
5	Antonio Esposito	19	Napoli

Join $\bowtie$

Conditional (or theta) Join: $T1 \bowtie_C T2 = \sigma_C(T1 \times T2)$

T1

ID	Name	Age	City
1	John Smith	44	Kansas City
2	Steven Miller	32	Portland
8	Alexa Fredrikson	37	Oslo

T2

ID	Date
1	29 Feb 2016
2	01 Feb 2016

$$T1 \bowtie_A ge < 40T2 = \sigma_{Age<40}(T1 \times T2) =$$

ID1	Name	Age	City	ID2	
2	Steven Miller	32	Portland	1	29 Feb 2016
2	Steven Miller	32	Portland	2	01 Feb 2016
8	Alexa Fredrikson	37	Oslo	1	29 Feb 2016
8	Alexa Fredrikson	37	Oslo	2	01 Feb 2016

Division

$$T1/T2_{x} = \{x | \exists (x, y) \in T1 \forall y \in T2\}$$

Where x, y are fields and T1, T2 are tables.

T1

x	y
a	1
a	2
c	1
d	3

T2

y
1
2

T1/T2 =

x
a
c

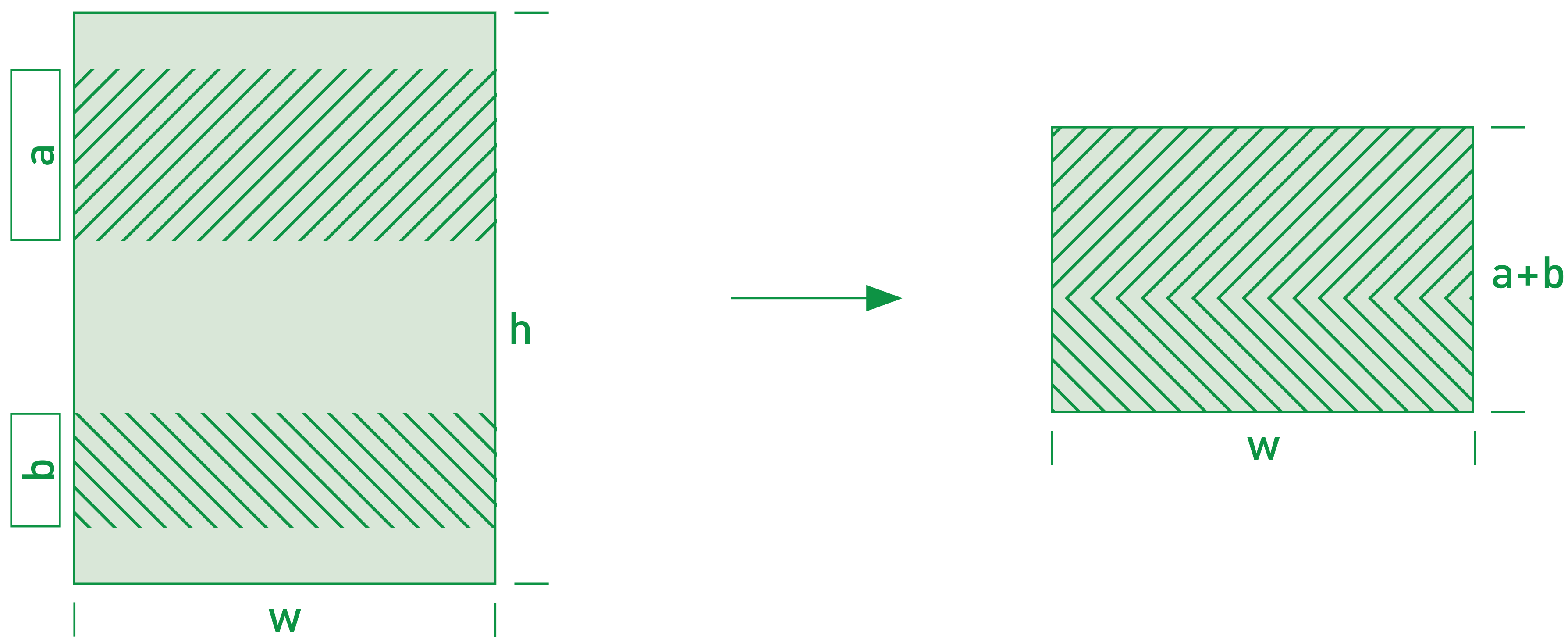
The division can be rewritten in terms of basic relational operators:

$$T1/T2_{x} = \pi_x(T1) - \pi_x((\pi_x(T1) \times T2) - T1)$$

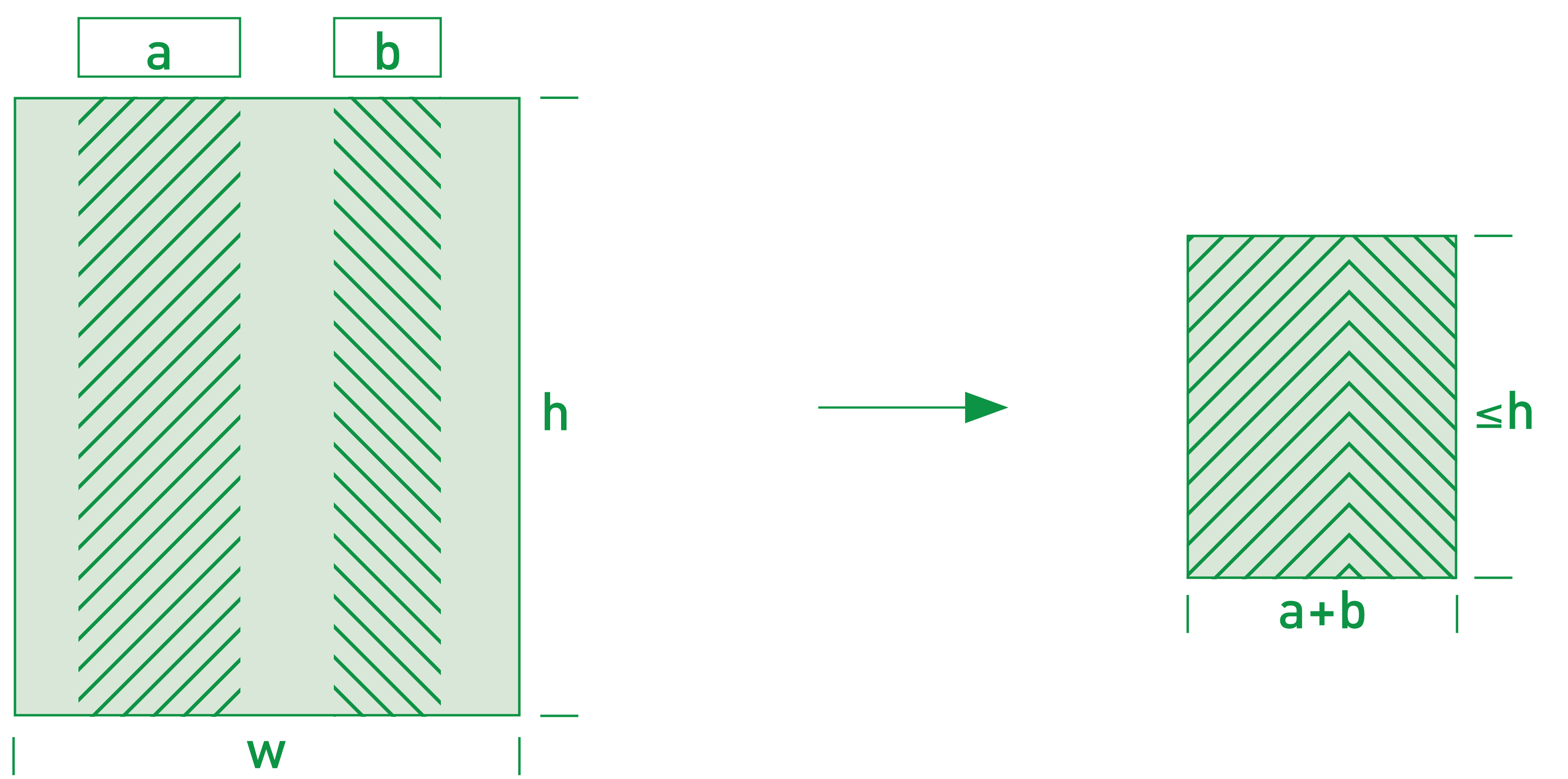
The division is obtained subtracting the disqualified tuples from x in T1.

Disqualified values are x fields where by attaching y from T2 we obtain an xy tuple not in T1.

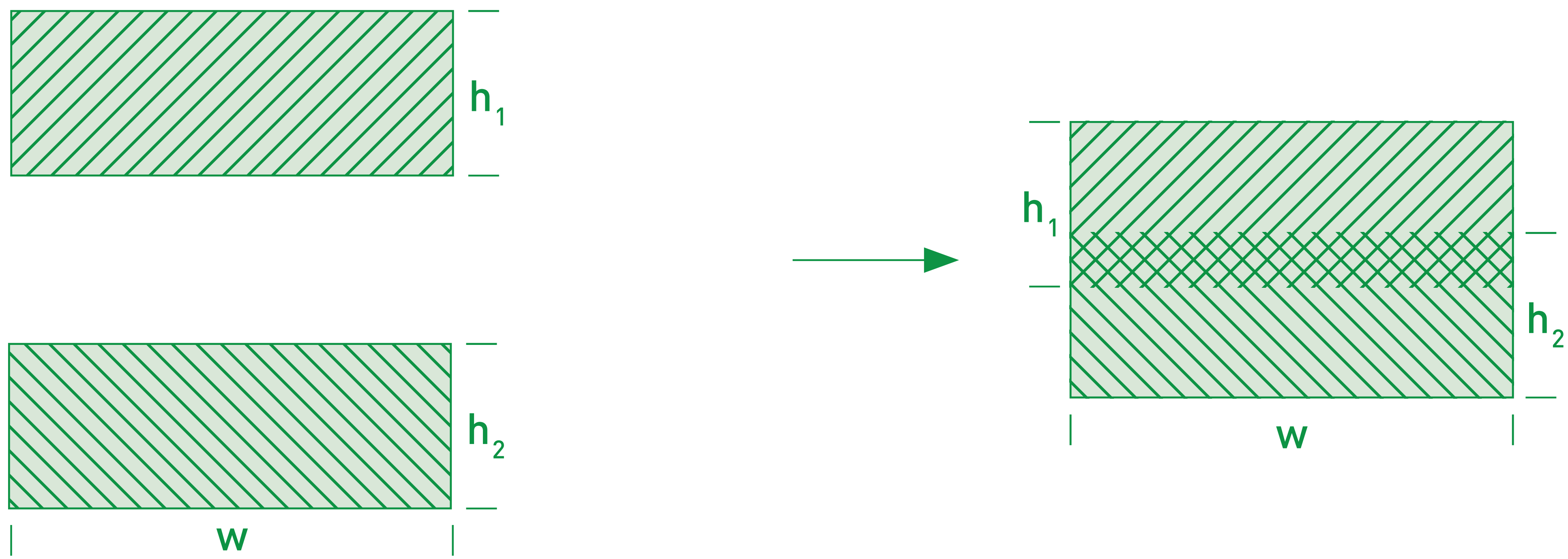
Select



Project



Union



Relational Algebra Equivalences

$\sigma_{c_1 \wedge c_2 \wedge \ldots} = \sigma_{c_1}(\sigma_{c_2}(\sigma_{c_3}\ldots)))$ cascading select

$\sigma_{c1}(\sigma_{c2}(T)) = \sigma_{c2}(\sigma_{c1}(T))$ select commutativity

$\pi_{f1}(\pi_{f2}(\pi_{f3}\ldots)) = \pi_{f1}(T)$ only the last projection matters

$T1 \bowtie_\theta T2 = T2 \bowtie_\theta T1$ theta-join commutativity

$(T1 \bowtie T2) \bowtie T3 = T1 \bowtie (T2 \bowtie T3)$ natural-join associativity

$(T1 \bowtie_{c_1} T2) \bowtie_{c_2 \wedge c_3} T3 = T1 \bowtie_{c_1 \wedge c_3} (T2 \bowtie_{c_2} T3)$ theta-join associativity

Union and Intersection are commutative and associative