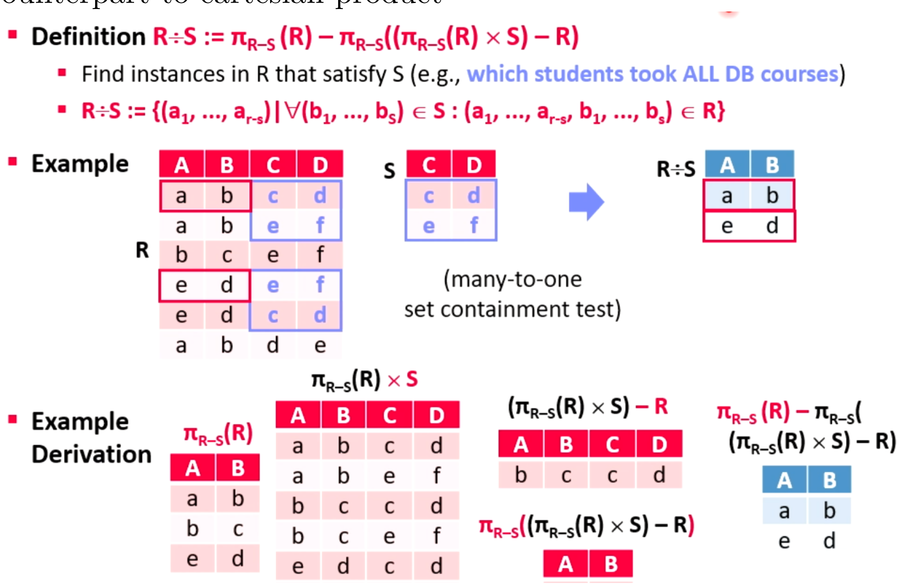
Intersection

- R∩S
- each tuple within R and S
- set intersection derived from difference
- bag intersection with element multiplicity min(R,S)
- require compatible schema

Division

- R÷S
- counterpart to cartesian product
-



- **Definition** $R \div S := \pi_{R-S}(R) - \pi_{R-S}((\pi_{R-S}(R) \times S) - R)$
  - Find instances in R that satisfy S (e.g., which students took ALL DB courses)
  - $R \div S := \{(a_1, \dots, a_{r-s}) \mid \forall (b_1, \dots, b_s) \in S : (a_1, \dots, a_{r-s}, b_1, \dots, b_s) \in R\}$
- **Example**

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	e	f
e	d	c	d
a	b	d	e

S

C	D
c	d
e	f

(many-to-one set containment test)

R÷S

A	B
a	b
e	d

- **Example Derivation**

$\pi_{R-S}(R)$

A	B
a	b
b	c
e	d

$\pi_{R-S}(R) \times S$

A	B	C	D
a	b	c	d
a	b	e	f
b	c	c	d
b	c	e	f
e	d	c	d

$(\pi_{R-S}(R) \times S) - R$

A	B	C	D
b	c	c	d

$\pi_{R-S}((\pi_{R-S}(R) \times S) - R)$

A	B

$\pi_{R-S}(R) - \pi_{R-S}((\pi_{R-S}(R) \times S) - R)$

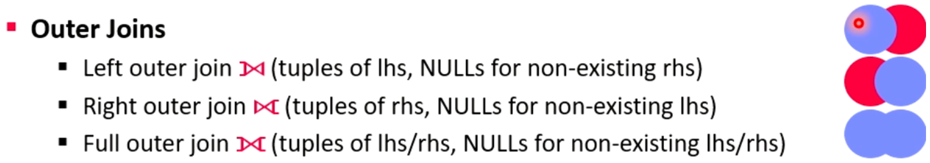
A	B
a	b
e	d

(Inner) Join

- R⋈S
- selection of tuples and attributes from R×S
  - equivalent in set/bag
- NULL never matches
- Theta Join
  - join by arbitrary condition
- Natural Join
  - equi join - join by equivalence
  - shared attributes only appearing once

Outer Join

- same as inner join but NULL matches
- left outer join

  - full left side, NULL for non-existing right side
- right outer join
  - full right side, NULL for non-existing right side
- full outer join
  - full left and right side, NULL for non-existing side
- 

  ▪ **Outer Joins**
  - Left outer join ⟕ (tuples of lhs, NULLs for non-existing rhs)
  - Right outer join ⟖ (tuples of rhs, NULLs for non-existing lhs)
  - Full outer join ⟗ (tuples of lhs/rhs, NULLs for non-existing lhs/rhs)

  - symbols different

Semi Join and Anti Join

- left semi join
  - join based on condition without left side
- right semi join
  - join based on condition without right side
- right/left anti join
  - returns tuples that do not appear within other side

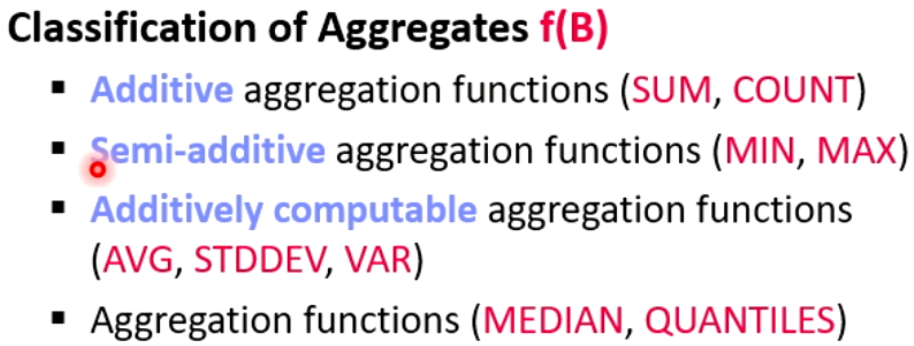
Deduplication, Sorting, Renaming

- deduplication
  - converts a bag into a set by removing duplicates
  - ALL/DISTINCT
    * indicate w/ or w/o duplicate elimination
- sorting
  - converts a bag into a sorted list of tuples
  - order lost if used in other ops
  - ASC/DESC
    * as/descending order
- Rename
  - defines new schema (new attribute/schema names)
  - tuples keep unchanged

Grouping and Aggregation

- grouping γ(R)
  - group input tuples R according to unique values in A
- aggregation
  - compute aggregate per group of tuples (created by grouping)

    - aggregation w/o grouping possible



Classification of Aggregates f(B)

- **Additive** aggregation functions (SUM, COUNT)
- **Semi-additive** aggregation functions (MIN, MAX)
- **Additively computable** aggregation functions (AVG, STDDEV, VAR)
- Aggregation functions (MEDIAN, QUANTILES)

    -

-

[[Relational Algebra]]

[[Relational Algebra]]