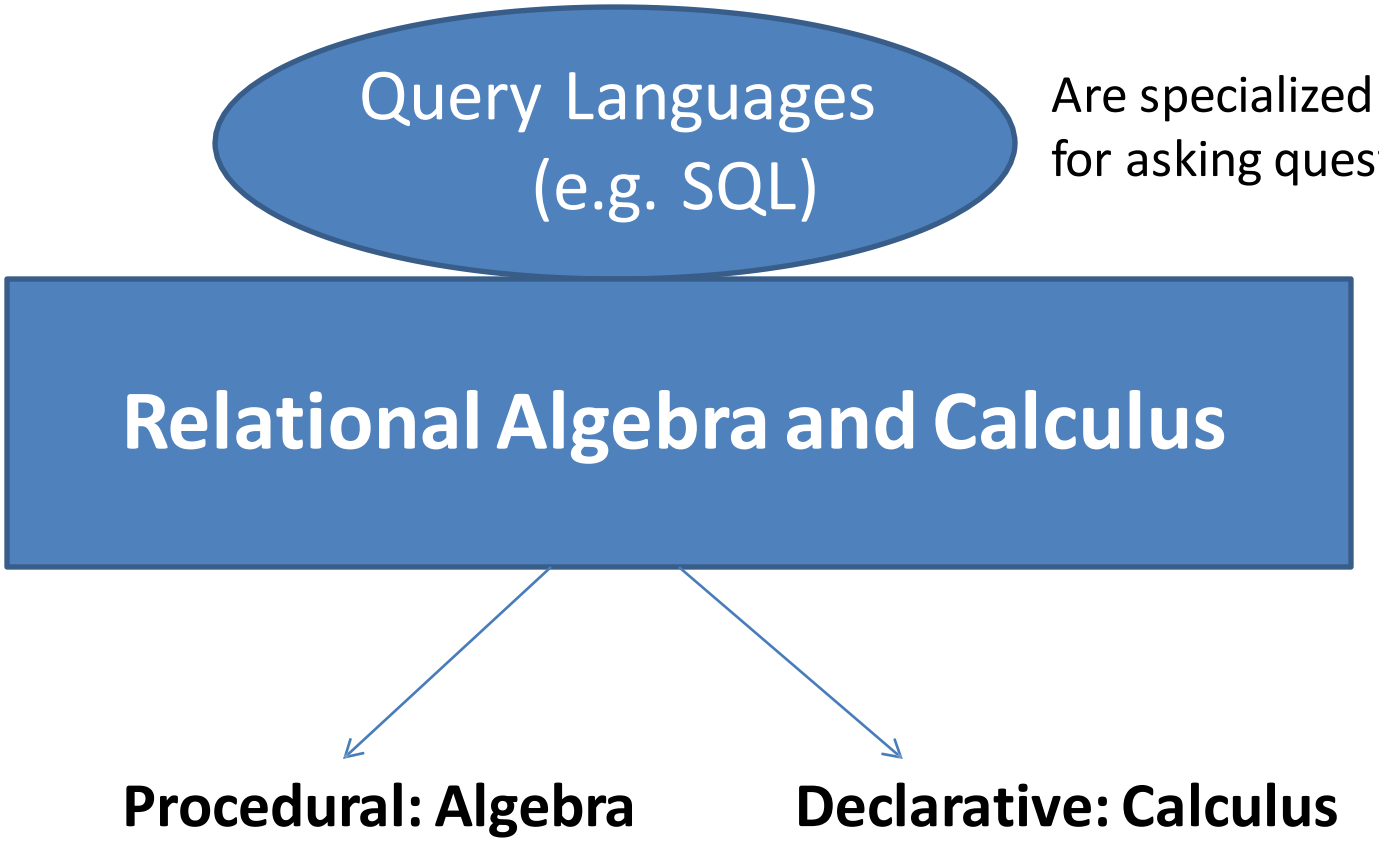
UNIT 2

RELATIONAL ALGEBRA AND CALCULUS



Are specialized languages for asking questions.


Instances of Relations
Query
Instances of a Relation

Relational Algebra

- Queries are composed using a collection of operators.
- Every operator:
  - Accepts one or two relation instances
  - Returns a relation instance.
- Compose relational algebra expression
- Each query describes a step-by-step procedure for computing the desired answer.

Relational Algebra

- Basic operators
  - Selection
  - Projection
  - Set operations
    - Union
    - Intersection
    - Difference
    - Cross-product
  - Renaming
  - Join
  - Division

Selection

$$\sigma_{Selection\_Criteria}(Input)$$

A relation instance

Manipulates data in a single relation

The selection operator specifies the tuples to retain through selection criteria.

A boolean combination (i.e. using V and Λ) of terms

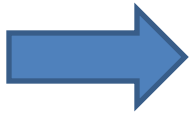
Attribute op constant or attribute1 op attribute2

< , <=, =, ≠, >=, or >

Selection

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0



$$\sigma_{rating>8}(S2)$$

sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

Projection

$$\pi_{fields}(Input)$$

Allows us to extract columns from a relation

Example:

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$$\pi_{age}(S2)$$

age
35.0
55.5

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$$\pi_{sname,rating}(\sigma_{rating>8}(S2))$$

sname	rating
yuppy	9
rusty	10

Set Operations

- Takes as input two relation instances
- Four standard operations
  - Union
  - Intersection
  - Set-difference
  - Cross-product
- Union, intersection, and difference require the two input set to be union compatible
  - They have the same number of fields
  - corresponding fields, taken in order from left to right, have the same domains

Set Operation: Union

- **R U S** returns relation instance containing all tuples that occur in either relation instance R or S, or both.
- R and S must be union compatible.
- Schema of the result is defined to be that of R.

Set Operation: Union

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

S1 U S2

Set Operation: Intersection

- R ∩ S: returns a relation instance containing all tuples that occur in both R and S.
- R and S must be union compatible.
- Schema of the result is that of R.

Set Operation: Intersection

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

S1 ∩ S2

Set Operation: Set-Difference

- **R – S**: returns a relation instance containing all tuples that occur in R but not in S.
- R and S must be union-compatible.
- Scheme of the result is the schema of R.

Set Operation: Set-Difference

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S1 – S2

sid	sname	rating	age
22	dustin	7	45.0

Set Operation: Cross-Product

- R x S: Returns a relation instance whose scheme contains:
  - All the fields of R (in the same order as they appear in R)
  - All the fields os S (in the same order as they appear in S)
- The result contains one tuple <r,s> for each pair with r ∈ R and s ∈ S
- Basically, it is the Cartesian product.
- Fields of the same name are unnamed.

Set Operation: Cross-Product

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

R1

sid	bid	day
22	101	10/10/96
58	103	11/12/96

S1 x R1

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Renaming

- Name conflict can arise in some situations
- It is convenient to be able to give names to the fields of a relation instance defined by a relational algebra expression.

$$\rho(R(\overline{F}),E)$$

- Take arbitrary relational expression E
- Returns an instance of a new relation R
- R is the result of E except that some fields are renamed
- Renaming list has the form (oldname → newname or position → newname)

Renaming

$$\rho(C(1 \to sid1, 5 \to sid2), S1 \times R1)$$

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

sid1	sname	rating	age	sid2	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96