Relational Algebra → query languages of a Relational Model

every operator in the algebra accepts one or two relation instances as arguments and returns a relation instance as a result.

Manipulate data in a single relation

Selection operator σ specifies the tuples to retain through a selection condition

Projection operator π allows us to extract columns from a relation

<u>union</u>

union compatible R and S

schema of result relation is identical to the schema of R

They have same number of fields

Corresponding fields, taken in order from left to right, have same domains.

R×S returns a relation instance whose schema contains all the fields of R in the same order they appear in R followed by all

same order as they
the fields of S in the same order they appear
in S.

A join is a cross product followed by selection and projection

<u>Condition Joins</u>

accepts a join condition C and a pair of relations instances as arguments and returns a relation instance.

$$R \bowtie_C S = \sigma_C(R \times S)$$

<u>Equijoin</u>

A special case of the join operation $R \bowtie S$ is when the join condition consists solely of equalities of the form R.name1 = S.name2
Join operation is defined by doing an additional projection in which S.name2 is dropped.
This join operation with this refinement is

called equijoin

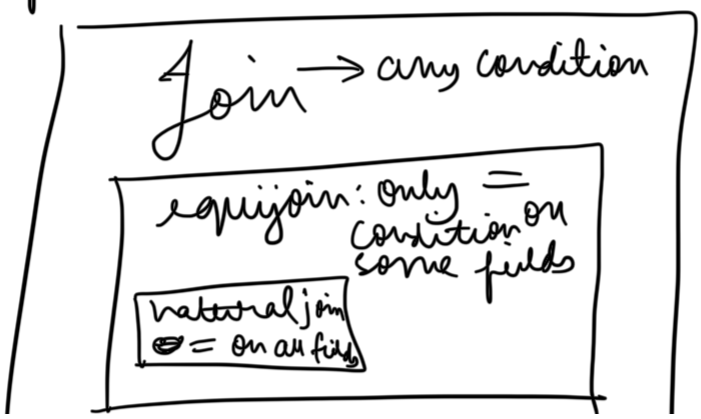
If this set of fields in the result contains 2 fields that inherit the same name from R and S → they are unnamed in the result relation

Natural Join

A special case of join condition R ⋈ S is an equijoin in which equalities are specified on all fields having the same name in R and S

we simply omit the join condition → the default is that the join condition is a collection of equalities on all common fields

result is always guaranteed to have distinct field names



Join can have any condition on any fields

equijoin uses = condition on some fields

natural join uses = condition on all fields

thus we don't have any more fields left with equal names and values

so some fields can still be equal but we didn't take them into account in the equijoin's condition

Thus equi and normal joins have some condition while natural joins have no condition

2 options of condition formulas →

1. decide which operator to use : >, <, <=, >=, =
2. decide which fields to use from both relations

normal join	equijoin	natural join
freely choose from 1 and 2	→ 1 = only used 2 can choose any field	none allowed to be specified by user 1 = used 2 choose all fields' combinations with same name

Division

consider two relation instances A and B in which A has exactly 2 fields x and y and B has just

one field y with the same domain as in A
we define A/B as the set of x values in
the form of unary tuples such that for
every y value in B, there is a tuple
<x, y> in A.

For each x value in the first column of A,
consider the set of y values that appear in
the second field of tuples of A with that
x value. If this set contains all the y values
in B, then this x value is in A/B's result

For relation instances A and B, A/B is the
largest relation instance Q, such that
$Q \times B \subseteq A$

A is like listing the parts supplied by
suppliers and B as listing the parts
A/B computes the suppliers who supply
all parts

Compute all x values which are not

disqualified

An x value is disqualified if by attaching a y value from B, we obtain a tuple $\langle x, y \rangle$ that is not in A

$$\pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

$$\pi_x(A) - \underbrace{\pi_x((\pi_x(A) \times B) - A)}_{\text{compute disqualified tuples}}$$

Declarative → allows us to describe the set of answers we want without being explicit about how they are computed

Tuple relational calculus, ~~tuple~~ on tuples as values

Domain relational calculus, range over field values

TRC, DRC

↓ ↓

SQL QBE

Query by example

A tuple variable is a variable that takes on

tuples of a relation as values
every value assigned to a tuple variable has the same number and types of fields

Domain Relational Calculus

form every $\{ < x_1, x_2 \ldots x_n > \mid P(< x_1, x_2, \ldots x_n >) \}$

each x_i is either a domain variable or constant and $P(< x_1, x_2 \ldots x_n >)$ denotes a DRC formula whose only free variables are among the x_i : $1 \leq i \leq n$

examples of Comparison b/w RA VS TRC VS DRC

(1) $\pi_{sname}(\sigma_{bid=103} \text{Reserves} \bowtie \text{Sailors})$

$\{ P \mid \exists S \in \text{Sailors} \exists R \in \text{Reserves} (R.sid = S.sid \wedge R.bid = 103 \wedge P.sname = S.sname) \}$

$\{ < N > \mid \exists I, T, A (< I, N, T, A > \in \text{Sailors} \wedge \exists Ir, Br, D (< Ir, Br, D > \in \text{Reserves} \wedge Ir = I$

$\wedge Br = 103)) \}$

(2) $\Pi_{sname, age}(\sigma_{rating > 7}(Sailors))$

$\{ P \mid \exists S \in Sailors (S.rating > 7 \wedge P.sname = S.sname \wedge P.age = S.age) \}$

$\{ \langle N, A \rangle \mid \exists I, T (\langle I, N, T, A \rangle \in Sailors \wedge T > 7 \}$

(3) Red Boat

$\{ \langle N \rangle \mid \exists I, T, A (\langle I N T A \rangle \in Sailors \wedge \exists \langle I, Br, D \rangle \in Reserves \wedge \exists \langle Br, BN, 'Red' \rangle \in Boats) \}$

$\{ P \mid \exists S \in Sailors \exists R \in Reserves (R.sid = S.sid \wedge P.sname = S.sname) \wedge \exists B \in Boats (B.bid = R.bid \wedge B.color = 'red')) \}$

$\Pi_{sname}(\sigma_{color='red'}(Boats) \bowtie Reserves \bowtie Sailors)$

---

(4) $\Pi_{Color}\left(\Pi_{sid}\left(\sigma_{\substack{\cancel{boat}\ Sailors \\ sname='lubber'}}\right) \bowtie Reserves \bowtie Boats\right)$

(5) $\Pi_{sname}(Sailors \bowtie Reserves)$

(6) $\rho(Temp, (\sigma_{color='red'}^{Boats}) \cup (\sigma_{color='green'}^{Boats}))$

$\Pi_{sname}(\cancel{Boats}\ Jump \bowtie Reserves \bowtie Sailors$

(7) $\rho(Temp, (\sigma_{color='red' \wedge color='green'}^{Boats}))$

$\Pi_{sname}(Temp \bowtie Reserves \bowtie \cancel{Boat}\ Sailors)$

Correct Solution $\longrightarrow$ find them separately

(8) $\rho(Reservations, \Pi_{sid, sname, bid}(Sailors \bowtie Reserves))$

$\rho(Reservationspair(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow bid1, 4 \rightarrow sid2, 5 \rightarrow sname2, 6 \rightarrow bid2),$

$Reservations \bowtie Reservations)$

$\Pi_{sname}\ \sigma_{(sid1 = sid2) \wedge (bid1 \neq bid2)}^{Reservations}$

$\{ P \mid \exists S \in Sailors\ \exists R1 \in Reserves\ \exists R2 \in Reserves(S.sid = R1.sid \wedge R1.sid = R2.sid$

$\wedge R1.bid \neq R2.bid \wedge S.sname = P.$

$\wedge R1.bid \neq R2.bid \wedge P.sname = S.sname)\}$

$\{\langle N \rangle \mid \exists ITA(\langle ITNA \rangle \in Sailors \wedge$
$\exists Br1\, Br2\, D1\, D2(\langle I\, Br1\, D1 \rangle \in Reserves$
$\wedge \langle I\, Br2\, D2 \rangle \in Reserves \wedge Br1 \neq Br2))\}$

(9) $\cancel{\pi_{sname}}$
$\pi_{sid}(\sigma_{age > 20}\, Sailors) - \pi_{sid}((\sigma_{color='red'}$
$Boats) \bowtie Reserves \bowtie Sailors)$

(10) $\rho(Temp, (\pi_{sid,bid}\, Reserves) / (\pi_{bid}\, \cancel{Reserves}\, Boats))$
$\pi_{sname}(Temp \bowtie Sailors)$

$\{P \mid \exists S \in \cancel{Boats}\, Sailors\, \forall B \in Boats(\exists R \in Reserves$
$(R.sid = S.sid \wedge R.bid = B.bid \wedge P.sname$
$= S.sname))\}$

$\{\langle N \rangle \mid \exists ITA(\langle INTA \rangle \in Sailors \wedge$
$\forall B, BN, C(\neg(\langle B, BN, C \rangle \in Boats)$
$\vee \exists I\, Br\, D \rangle \in Reserves(I = I_p$

$\vee (\exists \langle Sr, Br, D \rangle$

$\wedge Br = B)))\}$

(11) $\rho(Temp, (\Pi_{sid, bid} Reserves) / \Pi_{bid}(\sigma_{bname} = \text{'interlake'} Boats)))$

$\Pi_{sname}(Temp \bowtie Sailors)$

(12) $\{S \mid S \in Sailors \wedge \forall B \in Boats(B.color = \text{'red'}$

$\Rightarrow (\exists R \in Reserves(S.sid = R.sid$

$\wedge R.bid = B.bid)))\}$

$\{\langle INTA \rangle \mid \langle INTA \rangle \in Sailors \wedge \forall \langle B, BN, C \rangle$

$\in Boats(C = \text{'red'} \Rightarrow \exists \langle Ir\ Br\ D \rangle \in Reserved$

$(I = Ir \wedge Br = B))\}$