

Relational Algebra and SQL

2.4 and 6.1.

Recall:

Relational Algebra (RA)

• Operations on Relations.

Projection

$\pi_{\langle \text{List Expr} \rangle} R$

list of expressions on the attributes of a relation.

Ex. $R(a, b)$

a	b
1	9
3	3

① $\pi_a R$

a
1
3

② $\pi_{a+5, -b} R$

a+5	-b
6	-9
8	-3

name of attributes

③ $\pi_{b,a} R$

b	a
9	1
3	3

④ $\pi_{-1,a} R$

"-1", a
-1
-1

SQL:

select <list expr> from R

① SELECT a FROM R

② SELECT a+5, -b FROM R

③ SELECT b, a FROM R

④ SELECT -1, a FROM R

Name of Relation optional!!

SELECT 3;

3
3

 Creates table of

one tuple!!

SELECT 'abc', 5.2

=>

abc	5.2
'abc'	5.2

name of attribute.

Tupler.

The result of SELECT is always a relation

Renaming Relations and their attributes.

Sometimes we need to rename tables or their attributes.

$\rho_{\langle \text{new schema} \rangle} R$

Ex:

$R(a,b)$ $\rho_{S(c,d)} R$
renames $R(a,b)$ to
 $S(c,d)$

ding notation: you can rename during the projection.

If we want to rename the projected expression we can do it:

$\Pi_{a \rightarrow c, b \rightarrow d} R \rightarrow S$
Result schema $S(c,d)$

Ex: ① $\Pi_{a+5 \rightarrow x, -b \rightarrow y} R$

x	y
6	-9
8	-3

SQL.

Given $R(a,b)$ $\rho_{S(c,d)} R$

SELECT a, b FROM R as S(c,d)

or

SELECT a as c, b as d FROM R

①

SELECT a+5 AS X, -b AS Y FROM R

SELECTION

$$\sigma_p R$$

p is a predicate on attributes of R

Expressions:

$<, >, <=, >=, =$

\uparrow different \uparrow equal

AND, NOT and many others.

Ex:

	a	b
$R(a,b)$	3	2
	1	\emptyset

p evaluated at each tuple.

$$\textcircled{1} \sigma_{a>1 \text{ or } b>1} R$$

a	b
3	2

SQL.

SELECT * FROM R WHERE p

\uparrow original attributes of R

Ex:

① SELECT * FROM R
WHERE $a > 1$ OR $b > 1$

We can combine Π and σ :

Ex: $\Pi_a \sigma_{a > 1 \text{ OR } b > 1} R$

SELECT a FROM R
WHERE $a > 1$ OR $b > 1$

NOT equivalent to.

$\sigma_{a > 1 \text{ OR } \underline{b > 1}} \Pi_a R$

b is not part of $\Pi_a R$.

Questions

What does this return?

1) $\sigma_{\text{FALSE}} R$

2) $\sigma_{\text{TRUE}} R$

Other expressions in predicates.

IN $att \text{ IN } (List)$

Ex.:

$a \text{ IN } (3, 2, 5)$

\Rightarrow equivalent to $(a = 3 \text{ or } a = 2 \text{ or } a = 5)$

But we can also use a query:

$a \text{ in } (\Pi_c S)$

SQL:

$a \text{ IN } (\text{SELECT } c \text{ FROM } S)$

EXISTS

$\text{EXISTS } (R)$ true if R not empty

Ex:

$\text{EXISTS } (\sigma_{ass} R)$

Operations on 2 Relations.

Union	\cup
Intersection	\cap
Difference (Except)	$-$

Union Compatible

R and S are "union compatible" iff
 $|\text{attrs}(R)| = |\text{attrs}(S)|$

and the type of the i -th attribute of S. is **type compatible** with the type of the i -th attribute of R.

One type t_1 is type compatible with type t_2 if t_1 can be converted to type t_2 .

$A \cup B$
 $A \cap B$
 $A - B$ } Defined only iff
A & B are
union compatible.

UNION

$$t \in R \cup S \Leftrightarrow t \in R \text{ and } t \in S$$

$$t \in R \cap S \Leftrightarrow t \in R \text{ or } t \in S$$

$$t \in R - S \Leftrightarrow t \in R \text{ and } t \notin S$$

Schema of result is schema of first relation.

Ex:

$R(a, b)$	<table><tr><th>a</th><th>b</th></tr><tr><td>1</td><td>a</td></tr><tr><td>3</td><td>x</td></tr></table>	a	b	1	a	3	x	$S(c, d)$	<table><tr><th>c</th><th>d</th></tr><tr><td>1</td><td>e</td></tr><tr><td>3</td><td>x</td></tr><tr><td>4</td><td>f</td></tr></table>	c	d	1	e	3	x	4	f
a	b																
1	a																
3	x																
c	d																
1	e																
3	x																
4	f																

$R \cup S$

a	b
1	a
3	x
1	e
4	f

$R \cap S$

a	b
3	x

$R - S$

a	b
1	a

$S - R$

c	d
1	e
4	f

SQL

TABLE R { UNION
INTERSECT } TABLE S
EXCEPT

or

<QUERY> { UNION
INTERSECT } <QUERY>
EXCEPT

SELECT a, b FROM R

UNION

SELECT c, d FROM S;