Relational Algebra

Operations in the Relational Model

These operation can be expressed in an algebra, called **“relational algebra.”**

In this algebra, relations are the operands and we apply operators on them.

Operations

Four broad classes:

1. Usual set operations

   union

   intersection

   difference

2. Operations that remove parts of a relation:

   selection eliminates some rows(tuples)

   projection eliminates some columns

3. Operations that combine the tuples of two relations:

   Cartesian product pairs tuples of two relations in all possible ways

   join selectively pairs tuples from two relations.

4. An operation called **“renaming”.**

Conditions for Set Operations on Relations

We can apply **union, intersection, difference-** on relations **R** and **S** provided that:

1. **R** and **S** must have schemas with identical sets of attributes.

2. Before applying the operations, the columns of **R** and **S** must be ordered so that the order of attributes is the same for both relations.

Set Operations on Relations

R ∪ **S**, the **union** of **R** and **S**, is the set of tuples that are in **R** or **S** or both.

R ∩ **S**, the **intersection** of **R** and **S**, is the set of tuples that are in both **R** and **S**.

R – **S**, the **difference** of **R** and **S**, is the set of tuples that are in **R** but not in **S**.

Note that **R** – **S** is different from **S** – **R**.

Projection

$\pi_{A1,\ldots,An}(R)$

Produces from relation **R** a new relation that has only the A_1, …, A_n columns of **R**.

Example:

For $\pi_{\text{title, year, length}}$**(Movies)** on:

title	*year*	*length*	*filmType*	*studioName*	*producerC#*
Star wars	1977	124	color	Fox	12345
Mighty Ducks	1991	104	color	Disney	67890
Wayne's World	1992	95	color	Paramount	99999

Example (Continued)

We get:

title	*year*	*length*
Star wars	1977	124
Mighty Ducks	1991	104
Wayne's World	1992	95

What about π_{filmtype}**(Movies)** ?

Selection

σ_c(R)

Produces a new relation with those tuples of **R** which satisfy condition **C**.

Example:

For $\sigma_{length \geq 100}$**(Movies**) we have as result:

title	*year*	*length*	*filmType*	*studioName*	*producerC#*
Star wars	1977	124	color	Fox	12345
Mighty Ducks	1991	104	color	Disney	67890

Another Example

Suppose we want the movies by Fox which are at least 100 minutes long.

$\sigma_{length \geq 100 \text{ And } studioName='Fox'}(\text{Movies})$

Result is:

title	*year*	*length*	*filmType*	*studioName*	*producerC#*
Star wars	1977	124	color	Fox	12345

Cartesian Product

R×S

1. Set of tuples ***rs*** that are formed by choosing the first part (***r***) to be any tuple of **R** and the second part (***s***) to be any tuple of **S**.
2. Schema for the resulting relation is the union of schemas for **R** and **S**.
3. If **R** and **S** happen to have some attributes in common, then prefix those attributes by the relation name.

Example:

R

A	B
1	2
3	4

S

B	C	D
2	5	6
4	7	8
9	10	11

Example (Continued)

Resulting relation will be:

R×S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

Theta-Join

R $\bowtie_c$ **S**.

1. The result of this operation is constructed as follows:
   a) Take the Cartesian product of **R** and **S**.
   b) Select from the product only those tuples that satisfy the condition **C**.
2. Schema for the result is the union of the schema of **R** and **S** with, **“R”** or **“S”** prefix as necessary.

When the condition is equality, we call it “equijoin”.

Natural Join

Special case of equijoin when attributes we want to use in join have the same name in both tables

R ⋈ **S**

Let $\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_n}$ be the attributes in both the schema of **R** and the schema of **S**.

Then a tuple ***r*** from **R** and a tuple ***s*** from **S** are successfully paired if and only if ***r*** and ***s*** agree on each of the attributes $\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_n}$.

Example: The natural join of the relation **R** and **S** from previous example is:

A	B	C	D
1	2	5	6
3	4	7	8

Attributes with the same name have only one representative. Why?

Example

Compute the natural and theta join for relations U and V:

A	B	C
1	2	3
6	7	8
9	7	8

Relation U

B	C	D
2	3	4
2	3	5
7	8	10

Relation V

$U \bowtie V$ and $U \bowtie_{A<D} V$

Example

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

Result U ⋈ V

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

Result of U $\bowtie_{A<D}$ V

Combing Operations to Form Queries

"What are the **title** and **years** of movies made by **Fox** that are at least 100 minutes long?"

$\pi_{title,year}(\sigma_{length \geq 100 \text{ AND } studioName='Fox'}(\text{Movies}))$

```
SELECT title, year
FROM Movies
WHERE length>=100 AND studioName='Fox';
```

Another Example

Consider two relations **Movies** and **StarsIn**,

With schemas:

Movies(title, year, length, filmType, studioName)

StarsIn(title, year, starName)

Suppose we want to know:

"**Find the stars of the movies that are at least 100 minutes long**."

First we join the two relations: **Movies**, **StarsIn**

Second we select movies with length at least 100 min.

Then we project onto **starName**.

Another Example – Solution 1

$\pi_{\text{starName}}(\sigma_{\text{Movies.title=StarsIn.title AND Movies.year=StarsIn.year AND length>=100}}(\text{Movies} \times \text{StarsIn}))$

```
SELECT starName
FROM Movies, StarsIn
WHERE Movies.title=StarsIn.title AND Movies.year=StarsIn.year
    AND length>=100;
```

Another Example – Solution 2

$\pi_{starName}(\sigma_{length>=100}(Movies \triangleright\triangleleft_{Movies.title=StarsIn.title\ AND\ Movies.year=StarsIn.year} StarsIn))$

SELECT starName

FROM Movies JOIN StarsIn ON Movies.title=StarsIn.title AND
Movies.year=StarsIn.year

WHERE length>=100;

Another Example – Solution 3

$\pi_{starName}(\sigma_{length>=100}(Movies \triangleright\triangleleft StarsIn))$

```
SELECT starName
FROM Movies JOIN StarsIn USING (title, year)
WHERE length>=100;
```

Or (less safe):

```
SELECT starName
FROM Movies NATURAL JOIN StarsIn
WHERE length>=100;
```

Renaming Operator

$\rho_{S(A1,A2,\ldots,An)}$ **(R)**

1. Resulting relation has exactly the same tuples as **R**, but the name of the relation is **S**.
2. Moreover, the attributes of the result relation **S** are named A_1, A_2, …, A_n, in order from the left.

Problem

Product(maker, model, type)

PC(model, speed, ram, hd, rd, price)

Laptop(model, speed, ram, hd, screen, price)

Printer(model, color, type, price)

(Exercise 5.2.1)

maker	*model*	*type*
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2001	laptop
B	2002	laptop
B	2003	laptop
C	1007	pc
C	1008	pc
C	2008	laptop
C	2009	laptop
C	3002	printer
C	3003	printer
C	3006	printer
D	1009	pc
D	1010	pc
D	1011	pc
D	2007	laptop
E	1012	pc
E	1013	pc
E	2010	laptop
F	3001	printer
F	3004	printer
G	3005	printer
H	3007	printer

model	*speed*	*ram*	*hd*	*rd*	*price*
1001	700	64	10	48xCD	799
1002	1500	128	60	12xDVD	2499
1003	866	128	20	8xDVD	1999
1004	866	64	10	12xDVD	999
1005	1000	128	20	12xDVD	1499
1006	1300	256	40	16xDVD	2119
1007	1400	128	80	12xDVD	2299
1008	700	64	30	24xCD	999
1009	1200	128	80	16xDVD	1699
1010	750	64	30	40xCD	699
1011	1100	128	60	16xDVD	1299
1012	350	64	7	48xCD	799
1013	733	256	60	12xDVD	2499

(a) Sample data for relation PC

model	*speed*	*ram*	*hd*	*screen*	*price*
2001	700	64	5	12.1	1448
2002	800	96	10	15.1	2584
2003	850	64	10	15.1	2738
2004	550	32	5	12.1	999
2005	600	64	6	12.1	2399
2006	800	96	20	15.7	2999
2007	850	128	20	15.0	3099
2008	650	64	10	12.1	1249
2009	750	256	20	15.1	2599
2010	366	64	10	12.1	1499

(b) Sample data for relation Laptop

model	*color*	*type*	*price*
3001	true	ink-jet	231
3002	true	ink-jet	267
3003	false	laser	390
3004	true	ink-jet	439
3005	true	bubble	200
3006	true	laser	1999
3007	false	laser	350

(c) Sample data for relation Printer

Figure 5.10: Sample data for Product

Problem

a) Which PC models have a speed of at least 1000?

b) Which manufacturers make laptops with a hard disk of at least 30?

c) Find the model number and price of all products (of any type) made by manufacturer *B.*

d) Find the model numbers of all color laser printers.

e) Find those manufacturers that sell Laptops, but not PC's.

!f) Find those hard-disk sizes that occur in two or more PC's.

!g) Find those pairs of PC models that have both the same speed and RAM. A pair should be listed once.

!!h)Find those manufacturers of at least two different computers (PC or Laptops) with speed of at least 700.