IE301
Analysis and Design of Data Systems

Lecture 18

Relational Algebra 2

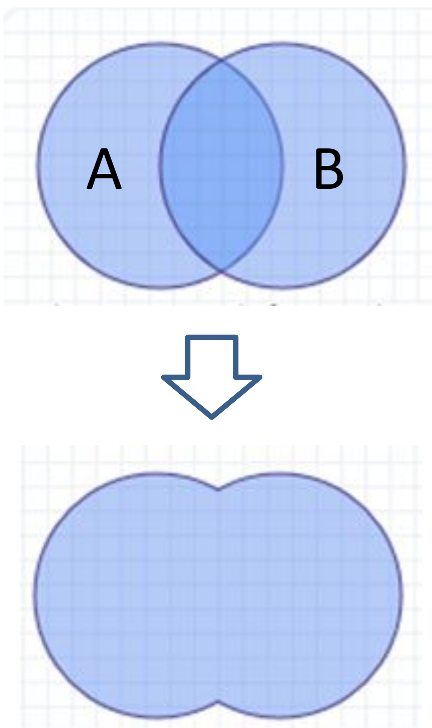
Aram Keryan

November 16, 2015

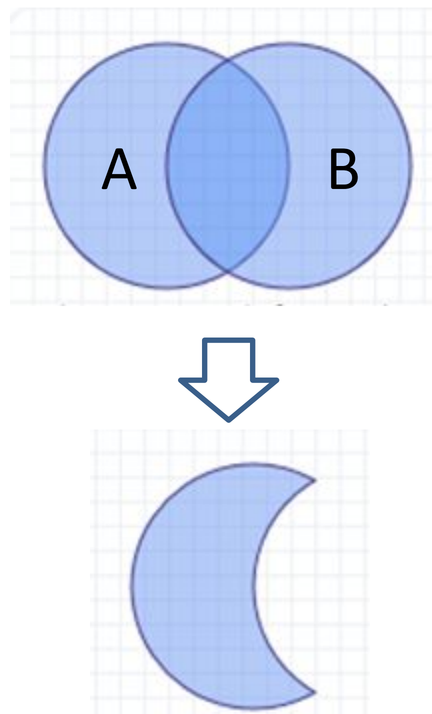
Set Operations
UNION, INTERSECTION, MINUS

The next group of relational algebra operations are the standard mathematical operations on sets.

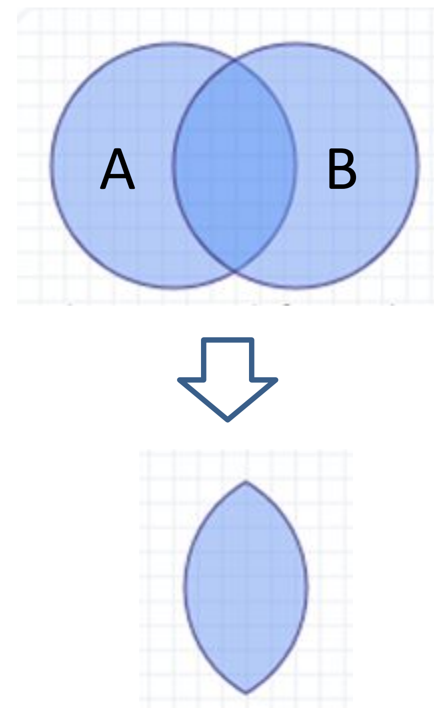
UNION



MINUS



INTERSECTION



Set Operations
UNION, INTERSECTION, MINUS

UNION, INTERSECTION, and MINUS are ***binary*** operations.

Let's $R(A_1, A_2, \dots, A_n)$ and $S(B_1, B_2, \dots, B_n)$ are relations.

The relations R and S on which any of set operations to be applied must satisfy to the next two conditions:

1) $Degree(R) = degree(S)$;
2) $dom(A_i) = dom(B_i) \qquad for\ 1 \le i \le n$;

- This means that the two relations have the same number of attributes and each corresponding pair of attributes has the same domain.

UNION

Example: Retrieve the Social Security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5.

Step 1: $\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$

Step 2: $\text{RESULT1} \leftarrow \pi_{\text{Ssn}}(\text{DEP5\_EMPS})$

Step 3: $\text{RESULT2(Ssn)} \leftarrow \pi_{\text{Super\_ssn}}(\text{DEP5\_EMPS})$

Step 4: $\text{RESULT} \leftarrow \text{RESULT1} \cup \text{RESULT2}$

or, as a single relational algebra expression:

$$\text{Result} \leftarrow \pi_{\text{Ssn}}\ (\sigma_{\text{Dno}=5}\ (\text{EMPLOYEE})\ ) \cup$$
$$\cup\ \ \pi_{\text{Super\_ssn}}\ (\sigma_{\text{Dno}=5}\ (\text{EMPLOYEE}))$$

✔ Set operations eliminate duplicates

RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Set operations (attributes naming)

STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

STUDENT ∪ INSTRUCTOR = **(a)**

STUDENT ∩ INSTRUCTOR = **(b)**

STUDENT − INSTRUCTOR = **(c)**

INSTRUCTOR − STUDENT = **(d)**

(a)

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(b)

Fn	Ln
Susan	Yao
Ramesh	Shah

(c)

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(d)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

CARTESIAN PRODUCT Operation

(CROSS PRODUCT)

Cartesian product is also a *binary set* operation denoted by $\times$.

Cartesian product operation produces a new element by combining every tuple from one relation with every tuple from the other relation.

$R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$ is a $n + m$ degree relation

$Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ with attributes in that order.

- ✔ The CARTESIAN PRODUCT operation by itself is generally meaningless, except when followed by a selection that matches values of attributes coming from the component relations.

CARTESIAN PRODUCT Operation

Example: retrieve a list of names of each female employee's dependents

Step 1: $\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{Sex}=\text{'F'}}(\text{EMPLOYEE})$

Step 2: $\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE\_EMPS})$

Step 3: $\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$

Step 4: $\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn}=\text{Essn}}(\text{EMP\_DEPENDENTS})$

Step 5: $\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Dependent\_name}}(\text{ACTUAL\_DEPENDENTS})$

Resulting Relations:

Step 1:

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

CARTESIAN PRODUCT Operation

Step 2:

EMPNAMES

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

Step 3:

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...

CARTESIAN PRODUCT Operation

Step 4:

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

Step 5:

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

- ✔ Because this sequence of CARTESIAN PRODUCT followed by SELECT is quite commonly used to combine *related tuples* from two relations, a special operation, called JOIN, was created to specify this sequence as a single operation.

Binary Operation θ-JOIN

θ-JOIN operation, denoted $\bowtie$, is used to combine *related tuples* from two relations into single tuples if join condition is satisfied.

Example: retrieve the name of the manager of each department.

SELECT d.Dname, e.Fname, e.Lname
FROM EMPLOYEE e, DEPARTMENT d
WHERE Ssn = Mgr_ssn;

$\Updownarrow$

$$\text{DEPT\_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr\_ssn=Ssn}} \text{EMPLOYEE}$$
$$\text{RESULT} \leftarrow \pi_{\text{Dname, Lname, Fname}}(\text{DEPT\_MGR})$$

General form of θ-JOIN

Let's $R(A_1, A_2, \ldots, A_n)$ and $S(B_1, B_2, \ldots, B_m)$ are relations.

The general form of a θ-JOIN operation on two relations R and S is:

$$R \bowtie_{\text{<join condition>}} S$$

where <join condition> is of the form:

<condition> **AND** <condition> **AND** … **AND** (condition)

where <condition> is of the form $A_i \theta B_j$

where A_i is an attribute of R, and B_j is from S

and θ is one of the comparison operators $\{=, <, \leq, >, \geq, \neq\}$.

NATURAL JOIN

NATURAL JOIN operation, denoted ⋈ (without join condition), is used to combine *related tuples* from two relations into single tuples if the join attributes have the same name in both relations.

Example: retrieve the name of the manager of each department.

DEPT_MGR ← EMPLOYEE ⋈ $\rho_{(Dname,Dnum,\boldsymbol{Ssn},Mgr\_srart\_date)}(DEPARTMENT)$

RESULT ← $\pi_{Dname,Lname,Fname}\ (DEPT\_MGR)$