Relational Algebra

Five Fundamental Operators

Selection $\sigma_\theta(r) = \{t \mid t \in r \text{ and } \theta\}$

cse_majors

id	name	class
1	stu1	FR
2	stu2	SO
3	stu3	JR
4	stu4	SR
5	stu5	GR

WinRDBI: select class = 'SR' (cse_majors);

id	name	class
4	stu4	SR

The selection operation $\sigma_\theta(r)$ retrieves some of the tuples from the operand relation r that satisfy the selection condition θ.

E.g. a query over the Employee Training enterprise that retrieves those employee whose salary is greater than 100 000:

√Qσ: $\sigma_{eSalary>100\,000}(employee)$;

Projection $\pi_A(r) = \{t.A \mid t \in r\}$

WinRDBI: project id, name (cse_majors);

id	name
1	stu1
2	stu2
3	stu3
4	stu4
5	stu5

The projection operation $\pi_A(r)$ retrieves some of the attributes of a relation as specified by the list of attributes given by the subscript A.

E.g. a query over the Employee Training enterprise that retrieves the last and first names and titles of all employees in the database.

√Qπ: $\pi_{eLast, eFirst, eTitle}(employee)$;

Union $r \cup s = \{t \mid t \in r \text{ or } t \in s\}$

eee_majors

id	name	class
2	stu2	SO
4	stu4	SR
6	stu6	SR

WinRDBI: cse_majors union eee_majors;

id	name	class
1	stu1	FR
2	stu2	SO
3	stu3	JR
4	stu4	SR
5	stu5	GR
6	stu6	SR

The binary operator union (∪) retrieves the union of the tuples from its operand relations.

E.g. a query over the Employee Training enterprise that retrieves the identification number of employees who are managers or coaches:

√Q∪: $\pi_{eID}(\sigma_{eTitle='Manager'}(employee)) \cup \pi_{eID}(\sigma_{eTitle='Coach'}(employee))$;

≡ √ $\pi_{eID}(\sigma_{eTitle='Manager' \text{ or } eTitle='Coach'}(employee))$;

Difference $r - s = \{t \mid t \in R \text{ and } t \notin s\}$

WinRDBI: cse_majors difference eee_majors;

id	name	class
1	stu1	FR
3	stu3	JR
5	stu5	GR

The binary operator difference (−) retrieves the tuples from the first operand relation that are not in the second operand relation.

Relation Algebra

E.g. a query over the employee Training enterprise that retrieves the identification number of employees who are managers that have not taken any training courses:

$\checkmark Q_{-} : \pi_{eID}(\sigma_{eTitle='Manager'}(employee)) - \pi_{eID}(takes);$

Cartesian Product $q \times r = \{ t_q t_r \mid t_q \in q \text{ and } t_r \in r \}$

cse_profs

name	office
Prof1	Off1
Prof2	Off2

cse_courses

crsid	crstitle
CSE410	IP
CSE412	DM
CSE556	ES

WinRDBI: cse_profs product cse_courses;

name	office	crsid	crstitle
Prof1	Off1	CSE410	IP
Prof1	Off1	CSE412	DM
Prof1	Off1	CSE556	ES
Prof2	Off2	CSE410	IP
Prof2	Off2	CSE412	DM
Prof2	Off2	CSE556	ES

The binary cartesian product operator (x) retrieves all possible combination of the tuples from the operand relations.

E.g. a query over the Employee Training enterprise that retrieves all possible combinations of employees (eID) with training courses (cID):

$\checkmark Q_x : \pi_{eID}(employee) \times \pi_{cID}(training\ Course);$

Relational Algebra

Intersection

The intersection operator ($\cap$) returns those tuples that are in both operand relation.

$$r \cap s = \{t \mid t \in r \text{ and } t \in s\}$$

This is equivalent to the following relational algebra expression in terms of only the fundamental operators:

$$r \cap s = r - (r - s)$$

cse_majors $\cap$ eee_majors

id	name	class
1	stu2	SO
4	stu4	SR

E.g. an example query over the Employee Training enterprise that retrieves the identification number of employees who are managers and have taken at least one training course:

$$Q_1 : \pi_{eID}(\sigma_{eTitle='Manager'}(employee)) \cap \pi_{eID}(takes);$$

Join

The join operator ($\bowtie_\theta$) is defined as a cartesian product of the operand relations followed by a selection using the θ condition:

$$P \bowtie_\theta q = \sigma_\theta(P \times q)$$

P

p_1	p_2
x_1	y_1
x_2	y_2

q

q_1	q_2
y_1	z_1
y_2	z_2
y_1	z_3
y_3	z_4

P × q

p_1	p_2	q_1	q_2
x_1	y_1	y_1	z_1
x_1	y_1	y_2	z_2
x_1	y_1	y_1	z_3
x_1	y_1	y_3	z_4
x_2	y_2	y_1	z_1
x_2	y_2	y_2	z_2
x_2	y_2	y_1	z_3
x_2	y_2	y_3	z_4

$P \bowtie_{p_2=p_1} q$

p_1	p_2	q_1	q_2
x_1	y_1	y_1	z_1
x_1	y_1	y_1	z_3
x_2	y_2	y_2	z_2

Natural Join $\bowtie$

$$P \bowtie q = \pi_{PUQ}(\sigma_{p.a_i = q.a_i \text{ and } \ldots \text{ and } p.a_j = q.a_j}(P \times q)), \text{ where } P \cap Q = \{a_i, \ldots, a_j\}$$

P

p_1	p_2	q_1
x_1	y_1	y_1
x_2	y_2	y_2

q

p_2	q_1	q_2
y_1	y_1	z_1
y_2	y_2	z_2
y_3		z_3

P ⋈ q

p_1	p_2	q_1	q_2
x_1	y_1	y_1	z_1
x_2	y_2	y_2	z_2

Division

$$P \div q = \pi_{P-Q}(P) - \pi_{P-Q}((\pi_{P-Q}(P) \times q) - P)$$

abTable ÷ bTable

a	b
a_1	b_1
a_1	b_2
a_2	b_2
a_3	b_1
a_3	b_2
a_3	b_3

b
b_1
b_2

⇒

a
a_1
a_3

(a_1, a_3 contains b_1, b_2)