DMD HW3 - Relational algebra

Nikolay Troshkov

September 2015

1 ER to relational data model

Employee(EID, E_name, E_address, salary)
Nurse(EID, Room_id, E_name, E_address, salary, shift)
Doctor(EID, E_name, E_address, salary, speciality)
Trainee(EID, E_name, E_address, salary, speciality)
Permanent(EID, E_name, E_address, salary, speciality)
visiting(EID, E_name, E_address, salary, speciality)
Receptionist(EID, E_name, E_address, salary)
Record(RecordN, appointment, Patient_id)
Patient(PID, name, address)
Rooms(Room_id, period)
Treatment(PID, dosage, indications)
Medicine(PID, code, price)

assigned(PID, Room_id)
attends(PID, EID)
maintains(EID, RecordN)
govern(EID, Room_id) *// an excess relation, but it is useful for the query 2.a*

2 Queries in relational algebra

a) Find all employees who are taking care of patients in room 107

$$Employee \ltimes (\sigma_{Room\_id=107}(govern))$$

b) Find all the nurses that Dr.Alex is not working with

$$Nurse - (Nurse \ltimes (Rooms \ltimes (assigned \ltimes (Patient \ltimes (attends \ltimes (\sigma_{E\_name=Alex}(Doctor)))))))$$

c) Find all employees who has more salary than at least one doctor

$$Employee \ltimes (Employee \bowtie_{salary>doctor\_salary} (\rho_{doctor\_salary \leftarrow salary}(Doctor)))$$

d) Find all the rooms that has at least one patient

$$Rooms \ltimes assigned$$

3 Queries with results

a) Find an employee with the third highest salary.

$$Employees \ltimes \gamma_{max(salary) \to salary}((Employees - (Employees \ltimes \gamma_{max(salary) \to salary}(Employees))) - (Employees \ltimes \gamma_{max(salary) \to salary}(Employees - (Employees \ltimes \gamma_{max(salary) \to salary}(Employees)))))$$

eid	**ename**	**salary**
3	Alex	$60,000

b) Find employees who can fly the flight100.

$$Employees \ltimes (Certified \ltimes (Aircrafts \bowtie_{range \geq distance} \sigma_{flight=100}(Flights)))$$

eid	**ename**	**salary**
3	Alex	$60,000
4	Sam	$40,000

c) Find flights that are certified only by employees with salary more than $70,000.

$$Flights \ltimes (Flights \bowtie_{distance \leq range} (Aircrafts \ltimes (\Pi aid(Certified \ltimes \sigma_{salary > 70000}(Employees)) - \Pi aid(Certified \ltimes \sigma_{salary \leq 70000}(Employees)))))$$

flight	**from**	**to**	**distance**

4

The lower bound is 0. The upper bound is $m \times n$.

An example of the lower bound

No matches:

R(foo, bar)
S(bar, baz)

R = {(1, 'a'), (2, 'b'), (3, 'c')}
S = {('d', 12), ('e', 13)}

R $\bowtie$ S = $\emptyset$
|R $\bowtie$ S| = 0

An example of the upper bound

Completely different sets of attributes:

R(id, foo)
S(bar, baz)

|R $\bowtie$ S| = $m \times n$

5

$\Pi_N(R - S)$ is not equivalent to $\Pi_N(R) - \Pi_N(S)$. An example:

R(N, foo)
S(N, bar)

R = {(1, 'a'), (2, 'b'), (3, 'c')}
S = {(1, 'a'), (1, 'b'))}

$R - S$ = R
$\Pi_N(R - S)$ = {(1), (2), (3)}

$\Pi_N(R)$ = {(1), (2), (3)}
$\Pi_N(S)$ = {(1)}
$\Pi_N(R) - \Pi_N(S)$ = {(2), (3)}

6

R(A, B, C)
S(C. D)

1) $R \bowtie S$ *is not* equal to $\sigma_{R.C=S.C}(R \times S)$

Cartesian product results in a relation that includes attributes from both sources, even if there are the same attributes, while natural join combines them.

(R×S)(R.A, R.B, R.C, S.C, S.D)
(R $\bowtie$ S)(A, B, C, D)

2) $\Pi_C(R \bowtie S)$ *is* equal to $\Pi_C(R) \bowtie \Pi_C(S)$

In both cases we are looking for tuples that share values in the same attribute.