Relational Algebra

G51DBI – Databases and Interfaces

Yorgos Tzimiropoulos
yorgos.tzimiropoulos@nottingham.ac.uk

Overview of next 3-4 lectures

In the next couple of lectures we will see how to translate English to Relational Algebra and SQL queries **and** vice versa

English: “Find all universities with > 20000 students”

Relational Algebra: $\pi_{uName}(\sigma_{enr > 20000}(University))$

SQL: Select uName From University Where University.enr>20000

Theory is easy and simple

But a sequence of simple operations is not always so obvious!

This Lecture

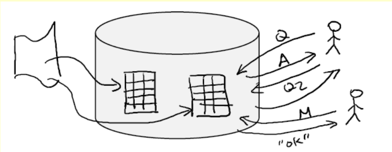
Relational algebra - Operators

- Projection, Selection
- Product, Join
- Union, Intersection, Difference
- Rename
- Examples

Create and Use Relational Databases

Basic steps

- Design Schema
- Insert Data
- Execute queries and modifications



Querying Relational Databases

University

uName	County	Enr
Notts	Nott/shire	18,000
Cam	Cam/shire	22,000
UCL	Greater Lon	20,000

Example with 3 Relations
- "All universities with > 20000 students"
- "All engineering depart. with < 2000 applicants"
- "Uni in Nott/shire with highest average GPA"

Student

SID	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Apply

SID	uName	Subj	Dec
0135	Cam	CS	'A'
0135	Notts	CS	'A'
0423	Notts	Eng	'R'

5

Querying Relational Databases

- Queries are expressed in high level language
- Query language in DS = DML (Data Manipulation Language: querying and data modification)
- Complicated queries expressed in a compact way
- Declarative (vs. Procedural): no need to specify how (the algorithm)
- Some queries are easy to pose
- Some queries easy to execute

6

2 properties

- When you ask a query over a relation you get back a relation
- This is called "**closure**"
- E.g. "Find all universities with >= 20000 students"

uName	County	Enr
Cam	Cam/shire	22,000
UCL	Greater Lon	20,000

- The result can be used as input to another query
- "Find all universities in Cambridgeshire from" ("Find all universities with > 20000 students") "
- This is called **composition**

7

Relational Algebra Vs SQL

2 Query Languages
- Relational Algebra is the formal language
- Provides the theoretical foundations for SQL
- SQL is the real language
- What runs an actual deployed database application

E.g. "Find all universities with > 20000 students"

RA:

$\pi_{uName}(\sigma_{enr > 2000}(University))$

SQL:

Select uName From University Where University.enr>2000

8

Relational Algebra Operators

- Operators in Relational Algebra allow us to filter, slice and combine relations
- We will use our Uni/Student/Apply example:

University

uName	county	Enr
Notts	Nott/shire	18,000
Cam	Cam/shire	22,000
UCL	Greater Lon	20,000

- Underlined are keys for each relation
- Q: What does it mean that SID, uName, Subj is a key for Apply?

Student

sID	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Apply

sID	uName	subj	dec
0135	Cam	CS	'A'
0135	Notts	CS	'A'
0423	Notts	Eng	'R'

Relational Algebra Operators

Operators in Relational Algebra allow us to filter, slice and combine relations

- Filter means row removal

Student

SID	sName	GPA	HS
0135	John	18.5	100
~~0025~~	~~Mary~~	~~19.3~~	~~1000~~
0423	Mary	17.5	300

Relational Algebra Operators

Operators in Relational Algebra allow us to filter, slice and combine relations

- Slice means column removal

Student

SID	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Relational Algebra Operators

Operators in Relational Algebra allow us to filter, slice and combine relations

- Combine means combine rows or columns

Student x Apply

S.SID	sName	GPA	HS	A.SID	uName	Subj	Dec
0135	John	18.5	100	0135	Cam	CS	'A'
0135	John	18.5	100	0135	Nott	CS	'A'
0135	John	18.5	100	0423	Nott	Eng	'A'
0423	Mary	17.5	300	0135	Cam	CS	'A'
0423	Mary	17.5	300	0135	Nott	CS	'A'
0423	Mary	17.5	300	0423	Nott	Eng	'R'

Select Operator

The Select Operator picks certain rows (filtering)

- E.g. 1: "(Find) Students with GPA > 19"

$$\sigma_{GPA > 19}(Student)$$

- Returns subset of the student table with rows such that GPA > 19

SID	sName	GPA	HS
0025	Mary	19.3	1000

E.g. 2: "Students with GPA > 19 and HS<1000"

$$\sigma_{GPA > 19 \text{ and } HS<1000}(Student)$$

E.g. 3: "Applications to Notts with subject CS"

$$\sigma_{uName = 'Notts' \text{ and } Subj='CS'}(Apply)$$

13

Select Operator

The Select Operator picks certain rows (filtering)

- General form

$$\sigma_{cond}\ (E)$$

- E is an Expression of Relational Algebra
- E could be a Relation
- OR E could be the result of applying an Operator (not necessarily only the Select) to a Relation
- E.g. $\sigma_{GPA > 19}((\sigma_{HS < 1000}(Student))$

14

Project Operator

The Project Operator picks certain attributes (slicing)

- E.g. 1: "(Get) the IDs and decisions from all applications

$$\pi_{SID,dec}(Apply)$$

SID	dec
0135	'A'
0135	'A'
0423	'R'

- There's no condition in the Project Operator
- General form

$$\pi_{A_1,A_2,\ldots,A_n}\ (E)$$

- E is an Expression of Relational Algebra
- $A_1, A_2, \ldots, A_n$ are the attributes to be kept

15

Select combined with Project

The Select and Project Operators can be naturally combined

- E.g. 1: "(Get) the IDs and names of students with GPA > 19"

$$\pi_{SID,sName}(\sigma_{GPA> 19}(Student))$$

- You can compose as much as you like select, project, select, select, project,….
- $\sigma_{GPA> 19}\ (\pi_{SID,sName}\ (Student))$?

16

Duplicates

A frequent result of applying Project is the creation of duplicates

- E.g. : "List the subjects and decisions of all applications"

$\pi_{subj,dec}(Apply)$

subj	dec
CS	'A'
CS	'A'
Eng	'R'
CS	'A'
CS	'A'
Eng	'R'

Vs

Subj	dec
CS	'A'
Eng	'R'

- In Relational Algebra all duplicates are eliminated. **NOT THE CASE** for SQL
- Relational Algebra is based on Sets. SQL is based on multi-Sets

17

Cross Product Operator

- Also known as Cartesian Product (operator x)
- It is applied between 2 or more relations
- The schema of the result is the union of the schema of the two relations
- The contents of the result are all combination of tuples from the 2 relations
- E.g. Student x Apply

Student

SID	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Apply

SID	uName	subj	dec
0135	Cam	CS	'A'
0135	Nott	CS	'A'
0423	Nott	Eng	'R'

18

Cross Product Operator

- Student x Apply

Student x Apply

S.SID	sName	GPA	HS	A.SID	uName	subj	dec
0135	John	18.5	100	0135	Cam	CS	'A'
0135	John	18.5	100	0135	Nott	CS	'A'
0135	John	18.5	100	0423	Nott	Eng	'A'
0025	Mary	19.3	1000	0135	Cam	CS	'A'
0025	Mary	19.3	1000	0135	Nott	CS	'A'
0025	Mary	19.3	1000	0423	Nott	Eng	'R'
0423	Mary	17.5	300	0135	Cam	CS	'A'
0423	Mary	17.5	300	0135	Nott	CS	'A'
0423	Mary	17.5	300	0423	Nott	Eng	'R'

19

Cross Product Operator

Student x Apply

- If S tuples from Student and A tuples from Apply, in total S x A tuples (3 x 3 = 9 in this example)
- If attributes share the same name, they should be renamed

e.g. S.SID and A.Apply

- Notice that some rows make little sense! For example S.SID = 0135 is combined with A.SID = 0423
- E.g. 2 "Names and GPAs of students with HS>1000 who applied to CS and were rejected

20

Cross Product Operator

Student x Apply

- If S tuples from Student and A tuples from Apply, in total S x A tuples (3 x 3 = 9 in this example)
- If attributes share the same name, they should be renamed

e.g. S.SID and A.Apply

- Notice that some rows make little sense! For example S.SID = 0135 is combined with A.SID = 0423
- E.g. 2 "Names and GPAs of students with HS>1000 who applied to CS and were rejected

$\pi_{GPA,sName}(\sigma_{S.SID=A.SID \text{ and } HS>1000 \text{ and } subj='CS' \text{ and } dec='Rej'}(St \times Ap))$

Natural Join Operator

Student ⋈ Apply (bowtie)

- Same as Cross product but enforces equality on all attributes with the same name (S.SID and A.SID in our case)
- Automatically sets values equal when attribute names are the same
- Gets rid of multiple copies of the attributes with the same name (there will be only one common SID attribute in the result)

Student ⋈ Apply

SID	sName	GPA	HS	uName	subj	dec
0135	John	18.5	100	Nott	CS	'A'
0135	John	18.5	100	Cam	CS	'A'
0423	Mary	17.5	300	Nott	Eng	'R'

Natural Join Operator

- E.g. 1 "Names and GPAs of students with HS>1000 who applied to CS and were rejected"
- E.g. 2 "Names and GPAs of students with HS>1000 who applied to CS at Universities with enrolment > 20000 and were rejected"
- Natural join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries

Natural Join Operator

- E.g. 1 "Names and GPAs of students with HS>1000 who applied to CS and were rejected"

$\pi_{GPA,sName}(\sigma_{HS>1000 \text{ and } subj = 'CS' \text{ and } dec='Rej'}(Student \bowtie Apply))$

- E.g. 2 "Names and GPAs of students with HS>1000 who applied to CS at Universities with enrolment > 20000 and were rejected"

$\pi_{GPA,sName}(\sigma_{HS>1000 \text{ and } subj = 'CS' \text{ and } dec='Rej' \text{ and } enr>20000}(St \bowtie (Ap \bowtie Uni)))$

- Natural join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries

Theta Join Operator

- Similar to Cartesian Product and Natural Join
- Can be implemented via Cartesian Product and Select. The Theta Join operator is defined as

$$\text{Student} \bowtie_{\theta} \text{Apply} = \sigma_{\theta}(\text{Student} \times \text{Apply})$$

- The result of this operation consists of all combinations of tuples in Student and Apply that satisfy condition θ
- Theta join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries

25

Union Operator

Union operator combines information between 2 relations vertically (cross product or join combines information horizontally)

- E.g. "List all Student and University names"

π_{sName} (Student) U π_{uName} (University)

Name
John
Mary
Cam
Nott

- Only relations with same schemas can be combined using the Union operator (not the exactly the case above!)
- Duplicates are always eliminated!

26

Difference Operator

Better illustrated with example!

- E.g. 1 "IDs of all students who didn't apply anywhere"

π_{SID} (Student) - π_{SID} (Apply)

- E.g. 2 "IDs and Names of all students who didn't apply anywhere"

27

Difference Operator

Better illustrated with example!

- E.g. 1 "IDs of all students who didn't apply anywhere"

π_{SID} (Student) - π_{SID} (Apply)

- E.g. 2 "IDs and Names of all students who didn't apply anywhere"

$\pi_{sName,SID}(\pi_{SID}$ (Student) - π_{SID} (Apply)) $\bowtie$ Student)

(this is called join back)

28

Intersection Operator

Better illustrated with example!

- E.g. 1 "Names that are both a University name and Student name"

$$\pi_{sName}(Student) \cap \pi_{uName}(Apply)$$

Intersection does not add expressive power

- $E1 \cap E2 = E1 - (E1 - E2)$

Rename Operator

The rename operator has 3 forms. The first one is the most general

- $\rho R_{(A1,\ A2,\ \ldots An)}(E)$. This should be read as: "Evaluate E, and get a relation as a result. Then call the result relation R with attributes A1,…,An." From now on, we can use this schema to describe the result of E.
- $\rho R(E)$. "Use the same attribute names but change the relation name to R"
- $\rho_{(A1,\ A2,\ \ldots An)}(E)$. "Use the same relation name but change the attribute names to A1,…,An."

Rename Operator

2 main uses

- Unifies schemas for the Union, Difference and Intersection operators

E.g. 1 "List all Student and University names"

$$(\rho C(name)(\pi_{sName}(Student)) \cup (\rho C(name)(\pi_{uName}(University))$$

- Helps to disambiguation in self joins

E.g. 2 "Pairs of Universities in same County"

Rename Operator

E.g. 2 "Pairs of Universities in same County"

University

uName	county	Enr
Notts	Nott/shire	18,000
Trent	Nott/shire	22,000
UCL	Greater Lon	20,000

-

Rename Operator

E.g. 2 “Pairs of Universities in same County”

- Step 1

$\sigma_{c1=c2}$ (ρU1(n1, c1, e1)(University)x ρU2(n2, c2, e2)(University))

OR ρU1(n1, c, e1)(University) ⋈ ρU2(n2, c, e2)(University)

- Step 2

$\sigma_{n1 \neq n2}$ (ρU1(n1, c, e1)(University) ⋈ ρU2(n2, c, e2)(University))

- Step 3

$\sigma_{n1>n2}$ (ρU1(n1, c, e1)(University) ⋈ ρU2(n2, c, e2)(University))

33

Question 1

- Assume that we are given relations R(A,C) and S(B,C,D):

R

A	C
3	3
6	4
2	3
3	5
7	1

S

B	C	D
5	1	6
1	5	8
4	3	9

Compute the natural join of R and S. Which of the following tuples is in the result? Assume each tuple has schema (A,B,C,D).

I. (5, 1, 6, 4)
II. (6, 4, 3, 9)
III.(3, 3, 5, 8)
IV.(2, 4, 3, 9)

34

Question 3

- Consider a relation R(A,B) with r tuples, all unique within R, and a relation S(B,C) with s tuples, all unique within S. Let t represent the number of tuples in R natural-join S. Which of the following triples of values (r, s, t) is possible?

I. (5, 2, 10)
II. (2, 3, 9)
III.(5, 10, 500)
IV.(5, 10, 250)

35

Question 4

- Consider a relation R(A) with r tuples, all unique within R, and a relation S(A) with s tuples, all unique within S. Let t represent the number of tuples in R minus S. Which of the following triples of values (r, s, t) is possible?

I. (5, 10, 10)
II. (5, 10, 15)
III.(5, 3, 4)
IV.(10, 5, 2)

36

Question 5

➢ Assume that we are given relations R(A,B) and S(B,C,D):

R

A	B
1	2
3	4
5	6

S

B	C	D
2	4	6
4	6	8
4	7	9

Compute the natural join of R and S. Which of the following tuples is in the result? Assume each tuple has schema (A,B,C,D).

I. (3, 4, 7, 8)
II. (3, 4, 6, 8)
III.(1, 2, 4, 8)
IV.(3, 4, 4, 6)

37

Question 11

Statement A: It can be useful to compose two selection operators.

$\sigma_{county='London'}(\ \sigma_{enr>8000}(University)\ )$

Statement B: It can be useful to compose two projection operators.

$\pi_{GPA}(\pi_{SID,GPA,HS}(Student))$

I. Both statements are true.
II. Statement A is true but Statement B is false.
III. Statement B is true but Statement A is false.
IV. Both statements are false.

38

Question 12

Which of the following expressions does NOT return the names and GPAs of students with HS>1000 who applied to CS and were rejected?

I. $\pi_{sName,GPA}($

$\sigma_{S.SID=A.SID}(\sigma_{HS>1000}(St) \times \sigma_{Subj='CS' \text{ and } dec='R'}(Ap))\ )$

II. $\pi_{sName,GPA}($

$\sigma_{S.SID=A.SID \text{ and } HS>1000 \text{ and } subj='CS' \text{ and } dec='R'}(St \times \pi_{SID,subj,dec}(Ap))\ )$

III. $\sigma_{S.SID=A.SID}($

$\pi_{sName,GPA}(\sigma_{HS>1000}(St) \times \sigma_{subj='CS' \text{and } dec='R'}(Ap))\ )$

39

Question 13

Which of the following expressions finds the IDs of all students such that some University bears the student's name?

I. $\pi_{SID}(Uni \bowtie Student)$

II. $\pi_{SID}(\sigma_{uName=sName}(Uni \times Student)\ )$

III. $\pi_{SID}(\ \pi_{uName}(Uni) \bowtie \pi_{uName}(\sigma_{sName=uName}(Student)))$

IV. $\pi_{SID}(\sigma_{uName=sName}(\pi_{SID}(Student \times Uni \times Student)\ )$

40

Thanks for your attention!

Any questions??

Question 2

➢ Assume that we are given relations R(A,C) and S(B,C,D):

R

A	B
1	a
7	t
2	g
4	c
9	t

S

B	C	D
c	5	6
a	7	8
t	8	9

Compute the theta-join of R and S with the condition R.B = S.B AND R.A < S.C Which of the following tuples is in the result? Assume each tuple has schema (A, R.B, S.B, C, D).

I. (2, g, g, 7, 8)
II. (4, c, c, 7, 8)
III.(2, g, t, 8, 9)
IV.(4, c, c, 5, 6)

Question 6

➢ Assume that we are given relations R(A,C) and S(B,C,D):

R

A	B
1	2
3	4
5	6

S

B	C	D
2	4	6
4	6	8
4	7	9

Compute the theta-join of R and S with the condition R.A < S.C AND R.B < S.D. Which of the following tuples is in the result? Assume each tuple has schema (A, R.B, S.B, C, D).

I. (5,6,2,4,6)
II. (3,4,5,7,9)
III. (1,2,2,6,8)
IV. (3,4,4,6,8)

Question 7

➢ Assume that we are given relation R(A,B,C) :

R

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

Compute the projection $\pi_{C,B}$ (R). Which of the following tuples is in the result?

I. (1, 2, 6)
II. (6, 5)
III. (2,6)
IV. (5,6)

Question 8

- Assume that we are given relations R(A,B,C) and S(A,B,C)

R

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

S

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3
2	5	3

Compute the union of R and S. Which of the following tuples DOES NOT appear in the result?

I. (1, 2, 3)
II. (4, 5, 3)
III. (4, 5, 6)
IV. (2, 5, 4)

45

Question 9

- Assume that we are given relations R(A,B,C) and S(A,B,C)

R

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

S

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3
2	5	3

Compute the intersection of R and S. Which of the following tuples appears in the result?

I. (2, 5, 3)
II. (2, 5, 4)
III. (4, 2, 3)
IV. (1, 2, 6)

46

Question 10

- Assume that we are given relations R(A,B,C) and S(A,B,C)

R

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

S

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3
2	5	3

Compute (R - S) union (S - R). Which of the following tuples is in the result?

I. (1, 5, 6)
II. (4, 5, 6)
III. (2, 5, 4)
IV. (4, 5, 3)

47

Question 14

Which of the following describes the result of this expression:

$\pi_{sName,uName}(\sigma_{HS>enr}(\sigma_{county='London'}(Uni \bowtie St \bowtie \sigma_{subj='CS'}(Ap))))$

I. All Student-University name pairs, where the student is applying to CS at the University, the University is in London, and the University is smaller than some High School

II. Students paired with all London Universities to which the Student applied to CS, where at least one of those Universities is smaller than the Student's High School

III. Students paired with all Universities smaller than the Student's high school to which the Student applied to CS, where at least one of those Universities is in London

IV. Students paired with all London Universities smaller than the Student's High School to which the Student applied to CS

48

Question 15

Which of the following describes the result of this expression:

$\pi_{uName}(Uni)$ –

$\pi_{uName}(Ap \bowtie (\pi_{sID}(\sigma_{GPA>19} St) \cap \pi_{SID}(\sigma_{subj='CS'}(Ap)))$

- All Universities with no GPA>19 Applicants who applied for CS at that University
- All Universities with no GPA>19 Applicants who applied for CS at any University
- All Universities where all Applicants either have GPA>19 or applied for CS at that University
- All Universities where no Applicants have GPA>19 or no Applicants applied for CS at that University

49