Relational Algebra

DBI - Databases and Interfaces

Dr Matthew Pike & Dr Yuan Yao

This Lecture

- Relational algebra - Operators
    - Projection, Selection
    - Product, Join
    - Union, Intersection, Difference
    - Rename
    - Examples

- Chapters 4, 5 of the DB Book

Manipulate Database

English <-> Relational Algebra <->SQL queries

English: "Find all universities with > 20000 students"

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

SQL: SELECT uName FROM University WHERE University.Enrollment>20000

Recall: Relational model

Attributes are: ID, Name, Salary & Department

The degree of the relation is 4

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

Schema is { ID, Name, Salary, Department }

Tuples, e.g.
{ (ID, A368),
(Name, Jane Brown),
(Salary, 22,000),
(Department, Accounts) }

The cardinality of the relation is 5

Relations: mathematical definition

- A relation R of degree n, where values come from domains $A_1, \ldots, A_n$:

$$R \subseteq A_1 \times A_2 \times \ldots \times A_n$$

subset of the Cartesian product of domains

- Cartesian product:

$$A_1 \times A_2 \times \ldots \times A_n = \{(v_1, v_2, \ldots, v_n) \mid v_1 \in A_1, v_2 \in A_2, \ldots, v_n \in A_n\}$$

- Example:

$A_1 = \{1, 2\}$ and $A_2 = \{3, 4\}$

$A_1 \times A_2 = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$

Relational model: data manipulation

- Data → Relations
- Data manipulation → Operations on relations
- Relational Algebra: take relations as input and produce new relations
- Operators: +,-,*,\ for numbers, &, |, ! for boolean values
    - Common set-theoretic one
    - Specific to relations

Unary Operators

Example: University Applications

- The University/Student/Apply example.
- What is the key for Apply?
- Relations can be represented as rName(aName1, aName2, …)
- E.g., University(uName, County, Enrollment).

University

<u>uName</u>	County	Enrollment
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Student

<u>SID</u>	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Apply

<u>SID</u>	<u>uName</u>	<u>Subj</u>	Dec
0135	CAM	CS	‘A’
0135	NOTT	CS	‘A’
0423	NOTT	ENG	‘R’

Selection

- Selection works as filters.
- Let R be a relation with n columns and α is a property of tuples.
- Selection from R subject to condition α is defined as:

$$\sigma_{\alpha(R)} = \{(a_1, \dots, an) | (a_1, \dots, an) \in R, \alpha(a_1, \dots, an)\}$$

- σ: selection operator
- α: properties
- R: target relation

Student

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

What are properties?

- Properties are expressions connected with logical symbols (and, or, not)
- Each expression is either:
  - Attributes comparisons ($=, \neq, >, <, \geq, \leq$), e.g., Enrollment > HS
  - Or comparison between an attribute and a value, e.g., GPA > 15

Exercise: Selection

- 1. Find out all students with GPA more than 19.
- 2. Find out all students with GPA more than 19 and high school size less than 1000.
- 3. Find out all applications to University of Nottingham with subject CS

University

uName	County	Enrollment
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

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’

Exercise: Selection

- 1. Find out all students with GPA more than 19.

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

- 2. Find out all students with GPA more than 19 and high school size less than 1000.

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

- 3. Find out all applications to University of Nottingham with subject CS

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

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

Projection

- Projection works as slicing.
- Let R be a relation with n columns and X is a set of attributes.
- Projection of R on X is represented as:

$$\pi_X(R)$$

- π: projection operator
- X: a set of attributes
- R: target relation
- $\pi_X(R)$ is a new relation only contain attributes from X

Student

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

Exercise: Projection

- 1. Get IDs and decisions from all applications.
- 2. Get IDs and names of students with GPA greater than 19.

University

<u>uName</u>	County	Enrollment
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Student

<u>SID</u>	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

Apply

<u>SID</u>	<u>uName</u>	<u>Subj</u>	Dec
0135	CAM	CS	‘A’
0135	NOTT	CS	‘A’
0423	NOTT	ENG	‘R’

Exercise: Projection

- 1. Get IDs and decisions from all applications.

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

- 2. Get IDs and names of students with GPA greater than 19.

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

Set Operators

Union

- Standard set-theoretic definition:

$$A \cup B = \{x \mid x \in A \; or \; x \in B\}$$

- E.g., $\{a, b, c\} \cup \{a, d, e\} = \{a, b, c, d, e\}$

Union

- Standard set-theoretic definition:

$$A \cup B = \{x \mid x \in A \; or \; x \in B\}$$

- E.g., $\{a, b, c\} \cup \{a, d, e\} = \{a, b, c, d, e\}$

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

Union

- Standard set-theoretic definition:
  $$A \cup B = \{x \mid x \in A \; or \; x \in B\}$$
- E.g., $\{a, b, c\} \cup \{a, d, e\} = \{a, b, c, d, e\}$

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

ID	Name
M139	John Smith
M140	Mary Jones
A368	Jane Brown
P222	Mark Brown
A367	David Jones

Union-compatible

- Two relations are union compatible if they have the same number of column and corresponding columns have the same domain. (Partial operation)

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Age
M140	23
P222	31
A367	28

Set difference

- Standard set-theoretic definition:

$$A - B = \{x \mid x \in A \; or \; x \notin B\}$$

- E.g., $\{a, b, c\} - \{a, d, e\} = \{b, c\}$
- Partial Operation: requires union-compatible

Set difference

- Standard set-theoretic definition:

$$A - B = \{x \mid x \in A \; or \; x \notin B\}$$

- E.g., $\{a, b, c\} - \{a, d, e\} = \{b, c\}$
- Partial Operation: requires union-compatible

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

Set difference

- Standard set-theoretic definition:

$$A - B = \{x \mid x \in A \; or \; x \notin B\}$$

- E.g., $\{a, b, c\} - \{a, d, e\} = \{b, c\}$
- Partial Operation: requires union-compatible

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

ID	Name
M139	John Smith
A368	Jane Brown

Intersection

- Standard set-theoretic definition:

$$A \cap B = \{x \mid x \in A \; and \; x \in B\}$$

- E.g., $\{a, b, c\} \cap \{a, d, e\} = \{a\}$
- Partial Operation: requires union-compatible

Intersection

- Standard set-theoretic definition:

$$A \cap B = \{x \mid x \in A \ and \ x \in B\}$$

- E.g., $\{a, b, c\} \cap \{a, d, e\} = \{a\}$
- Partial Operation: requires union-compatible

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

Intersection

- Standard set-theoretic definition:

$$A \cap B = \{x \mid x \in A \; and \; x \in B\}$$

- E.g., $\{a, b, c\} \cap \{a, d, e\} = \{a\}$
- Partial Operation: requires union-compatible

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
P222	Mark Brown
A367	David Jones

ID	Name
A367	David Jones

Cartesian product

- Standard set-theoretic definition:

$$A \times B = \{(x, y) \mid x \in A, y \in B\}$$

- E.g., $\{a, b\} \times \{d, e\} = \{(a, d), (a, e), (b, d), (b, e)\}$
- Total Operation
- Extended Cartesian product:

$$A \times B = \{(c_1, \ldots, cn, d_1, \ldots, dm) \mid (c_1, \ldots, cn) \in A, (d_1, \ldots, dm) \in B\}$$

Example: Cartesian product

- The contents of the result are all combination of tuples from the 2 relations
- Note, some tuples are meaningless, e.g., S.SID=0135 is combined with A.SID = 0423
- How to solve it?

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’

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	‘R’
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’

Joint Operators

Natural Join Operator

- **Student ⋈ Apply** (bowtie)
  - Same as Cartesian 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	CAM	CS	‘A’
0135	John	18.5	100	NOTT	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”
  - $\pi_{GPA,sName}(\sigma_{HS>1000 \text{ and } subj=\text{'CS'} \text{ and } dec=\text{'Rej'}}(\textbf{Student} \bowtie \textbf{Apply}))$

- E.g. 2 “Names and GPAs of students with HS>1000 who applied to CS at Universities with enrolment > 20000 and were rejected”
  - $\pi_{GPA,sName}(\sigma_{HS>1000 \text{ and } subj=\text{'CS'} \text{ and } Enrollment>20000 \text{ and } dec=\text{'Rej'}}(\textbf{Student} \bowtie \textbf{Apply} \bowtie \textbf{Uni}))$
- Natural join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries

Theta Join Operator

- Cartesian Product satisfy certain properties
- Can be implemented via Cartesian Product and Select.
- The Theta Join operator is defined as
  - $R \bowtie_{\theta} S = \sigma_{\theta} (R \times S)$
- The result of this operation consists of all combinations of tuples in R and S that satisfy property θ
- Theta join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries

Rename Operator

- The rename operator has 3 forms. The first one is the most general
  - $\rho R(A_1, A_2, \ldots A_n)(E)$.
    - This should be read as: “Evaluate E, and get a relation as a result. Then call the result relation R with attributes $A_1, \ldots, An$.” 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
  - $\rho(A_1, A_2, \ldots A_n)(E)$.
    - “Use the same relation name but change the attribute names to $A_1, \ldots, An$.”

Rename Operator

- 2 main uses
  - Unifies schemas for the Union, Difference and Intersection operators
  - E.g. 1 “List all Student and University names”

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

  - Helps to disambiguation in self joins
  - E.g. 2 “Pairs of Universities in same County”

University

uName	**County**	**Enrollment**
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Rename Operator

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

- Step 1
  - $\rho U1(n1, c, e1)(University) \bowtie \rho U2(n2, c, e2)(University)$
- Step 2
  - $\sigma_{n1 \neq n2} (\rho U1(n1, c, e1)(University) \bowtie \rho U2(n2, c, e2)(University))$
- Step 3
  - $\sigma_{n1 > n2} (\rho U1(n1, c, e1)(University) \bowtie \rho U2(n2, c, e2)(University))$

Question

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).

1) (5, 1, 6, 4)
2) (6, 4, 3, 9)
3) (3, 3, 5, 8)
4) (2, 4, 3, 9)