Relational Algebra

Tutorial

SWEN304/435

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Outline

- Unary Operation: Select, Project, Rename
- Binary Operation: Join, Cartesian Product, Outer Join, Union, Interaction, Difference
- Relational algebra exercises

Unary Operations

- Project: $\pi_{AL}(N)$
  - Example: $\pi_{\text{LName, FName}}(\text{Student})$
- Select: $\sigma_{c}(N)$
  - Example: $\sigma_{\text{FName} = \text{'Susan'}}(\text{Student})$
- Rename: $\rho_{A1 \rightarrow B1,\ldots,Ak \rightarrow Bk}(N)$
  - Example: $\rho_{\text{FName} \rightarrow \text{FirstName,LName} \rightarrow \text{LastName}}(\text{Student})$

Binary Operations

- Union: $N_1 \cup N_2$
  - Example: $\pi_{StudId}(Student) \cup \pi_{StudId}(Grades)$
- Interaction: $N_1 \cap N_2$
  - Example: $\pi_{StudId}(Student) \cap \pi_{StudId}(Grades)$
- Difference: N_1 - N_2
  - Example: $\pi_{StudId}(Student)$ - $\pi_{StudId}(Grades)$

Binary Operation: Join Operations

- Join operation joins two relations by merging those tuples from two relations that satisfy a given condition
  - The condition is defined on attributes belonging to relations to be joined
- Equijoin, natural join operations

Equijoin Operation

- Notation: $N = N_1 \bowtie_{JC} N_2$

  where $JC = jc_1 \wedge \ldots \wedge jc_n$

  $jc_i \equiv A = B,\ A \in R_1,\ B \in R_2,$

- For example,

  $\text{Student} \bowtie_{\text{StudId = StudId}} \text{Grades}$

  In SQL:

```
SELECT *
FROM Student s, Grades g
WHERE s.StudId = g.StudId;
```

Equijoin

Equijoin: $N_1 \bowtie_{N1.B = N2.B} N_2$

N_1

A	B
1	2
3	3
4	4

$\bowtie_{N1.B = N2.B}$

N_2

B	C
2	7
4	9
ω	0

=

A	B	B	C
1	2	2	7
4	4	4	9

Natural Join

Natural Join : $N = N_1 * N_2$

N_1

A	B
1	2
3	3
4	4

*

N_2

B	C
2	7
4	9
ω	0

=

N

A	B	C
1	2	7
4	4	9

Cartesian Product

N_1

A	B
1	2
3	3
4	4

×

N_2

B	C
2	7
4	9
ω	0

=

$N_1 \times N_2$

A	B	B	C
1	2	2	7
1	2	4	9
1	2	ω	0
3	3	2	7
3	3	4	9
3	3	ω	0
4	4	2	7
4	4	4	9
4	4	ω	0

Outer Join

Right Outer Join

N_1

A	B
1	2
5	6

$\bowtie_{B=C}$

N_2

C	D
2	7
2	9
ω	7

=

$N_1 \bowtie_{B=C} N_2$

A	B	C	D
1	2	2	7
1	2	2	9
ω	ω	ω	7

Full Outer Join

N_1

A	B
1	2
5	6

$\bowtie_{B=C}$

N_2

C	D
2	7
2	9
ω	7

=

$N_1 \bowtie_{B=C} N_2$

A	B	C	D
1	2	2	7
1	2	2	9
5	6	ω	ω
ω	ω	ω	7

Summary or Relational Operations

- SELECT $\sigma_c(N)$: choose rows
- PROJECT $\pi_{A1,\ldots,Ak}(N)$: choose columns
- RENAME $\rho_{A1\rightarrow B1,\ldots,Ak\rightarrow Bk}(N)$: rename attributes
- JOIN: combine tables
  - Natural Join $N_1 * N_2$ or
  - Equi-Join $N_1 \bowtie_{A1=B1,\ldots,Ak=Bk} N_2$
- CARTESIAN PRODUCT (x): combine tables
- Set operations
  - UNION ($\cup$),
  - INTERSECTION ($\cap$),
  - DIFFERENCE (or MINUS, –)
- Additional Relational Operations
  - OUTER JOINS

A Sample Relational Database

Student

LName	FName	StudId	Major
Smith	Susan	131313	Comp
Bond	James	007007	Math
Smith	Susan	555555	Comp
Cecil	John	010101	Math

Course

CName	CourId	Points	Dept
DB Sys	C302	15	Comp
SofEng	C301	15	Comp
DisMat	M214	22	Math
Pr&Sys	C201	22	Comp

Grades

StudId	CourId	Grade
007007	C302	A+
555555	C302	ω
007007	C301	A
007007	M214	A+
131313	C201	B-
555555	C201	C
131313	C302	ω
007007	C201	A
010101	C201	ω

Exercises

- Suppose we are given the university database instance as in slide 9. Write queries in relational algebra for the following queries
  1. Find all students with their ID who got at least one 'A+'
  2. Find students with their ID, FName, who have enrolled in C302
  3. Find students with their IDs who have enrolled in 'C201' but not 'C302'
  4. Find students who have enrolled in both 'M214' and 'C302'
  5. Find students who have neither enrolled in 'M214' nor in 'C302'
  6. Find students who major in 'Math' and got 'A+' in at least one course offered by computer science department
  7. Find students who always take courses from Comp department.