Relational Algebra

Tutorial

SWEN 304

Trimester 2, 2017

Lecturer: Dr Hui Ma

Engineering and Computer Science

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

Slides by: Pavle Morgan & Hui Ma

Outline

- Natural join
- Natural join and Cartesian product
- Expressions for database updates
- Aggregates
- Outer Join

Join Operation

- 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
- Theta, equi, and natural join operations

Natural Join

$r\ (N_1)$

A	*B*
1	2
3	3
4	4

*

$r\ N_2)$

B	*C*
2	7
4	9
ω	0

=

$r(N)$

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

Cartesian Product

$r\ (N_1)$

A	*B*
1	2
3	3
4	4

×

$r\ N_2)$

B	*C*
2	7
4	9
ω	0

=

$r\ (N')$

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

Join and Cartesian Product

- Join is not a basic operation
- It can be composed by using select, project and set theoretic operations
- e.g.
  - Given relations $r(N_1)$ and $r(N_2)$, let $R_1 \cap R_2 = X$, then the natural join is

$$r\,(N_1) * r\,(N_2) = \pi_{(R1\ \cup R2\,)}(\sigma_{N1.X = N2.X}(r\,(N_1) \times r\,(N_2)))$$

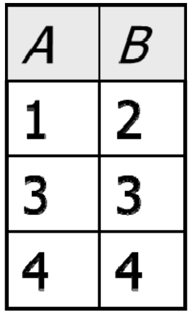
  - When $R_1 \cap R_2 = \varnothing$, then

$$|r(N_1) * r(N_2)| = |r(N_1) \times r(N_2)|$$

Natural Join and Cartesian Product

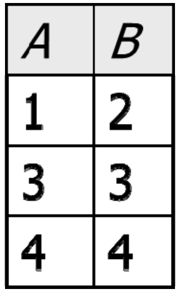
Equijoin: $\sigma\,(r(N_1) \bowtie_{N1.B\,=\,N2.B} r(N_2)) = \sigma_{N1.B\,=\,N2.B}\,(r(N_1) \times r(N_2))$

$r\,(N_1)$



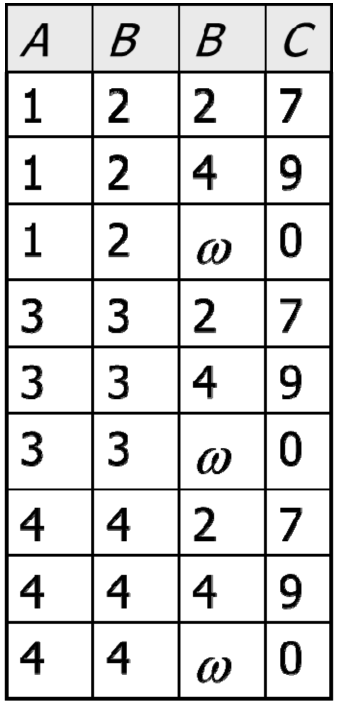
A	B
1	2
3	3
4	4

$r\,(N_1)$



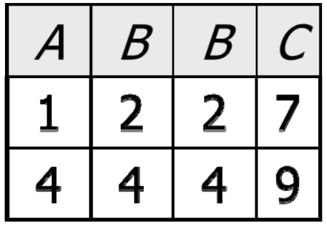
A	B
1	2
3	3
4	4

$r(N_1) \times r(N_2))$



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

$\sigma_{N1.B\,=\,N2.B}\,(r(N_1) \times r(N_2))$



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

Natural Join and Cartesian Product

Natural Join : $r\,(N1) \;*\; r\,(N2) =$

$r(N_1)$

A	B
1	2
3	3
4	4

*

$r(N_2)$

B	C
2	7
4	9
ω	0

$\pi_{\mathrm{A,B,C}}(\sigma_{N1.B\,=\,N2.B}\,(r(N_1) \times r(N_2)) =$

$r(N\,)$

A	B	C
1	2	7
4	4	9

Database Update Operations

- Database update operations are founded on the set theoretic operations:
  - Insert $t\,[R]$ into $r\,(N)$ over $N\,(R, C)$

$$r_1\,(N) = r(N) \cup \{t[R]\}$$

  - Delete $t\,[R]$ from $r(N)$

$$r_2(N) = r(N\,) - \{t\,[R\,]\,\}$$

  - Modify $t\,[R\,] = (a_1, \ldots, a_i, \ldots, a_n\,) \in r(N)$ into

$$u\,[R] = (a_1, \ldots, b_i, \ldots, a_n\,)$$

$$r_3(N) = (r(N) - \{t\,[R\,]\,\}) \cup \{u\,[R\,]\,\}$$

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

PName	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

A Closer Insight into Aggregates

- *Query*: for each student retrieve the his/her first name and the number of papers enrolled

- Query in relational algebra:

$$ {}_{\text{StudId, FName}}\mathcal{F}_{\text{(COUNT, CourId)}} \text{(Grades * Student)} $$

StudId	Fname	COUNT_CourId
010101	John	1
070707	James	4
131313	Susan	2
555555	Susan	2

Aggregate Functions and Grouping

- *Query:* For each student retrieve the number of courses that he/she passed
- Let Grades1 = $\sigma_{\text{Grade} \neq \omega}$ (Grades)

$$_{\text{StudId}}\,\mathcal{F}_{\text{(COUNT, CourId)}}(\text{Grades1})$$

StudId	COUNT_CourId
070707	4
131313	1
555555	1

Outer Join

Right Outer Join

$r(N_1)$

A	B
1	2
5	6

$⟖_{B=C}$

$r(N_2)$

C	D
2	7
2	9
ω	7

=

$r(N_1) ⟖_{B=C} r(N_2)$

A	B	C	D
1	2	2	7
1	2	2	9
ω	ω	ω	7

Full Outer Join

$r(N_1)$

A	B
1	2
5	6

$⟗_{B=C}$

$r(N_2)$

C	D
2	7
2	9
ω	7

=

$r(N_1) ⟗_{B=C} r(N_2)$

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 $\sigma_c(r(N))$: choose rows
- PROJECT $\pi_{A1,\dots,Ak}$ $(r(N))$: choose columns
- RENAME $\delta_{A1\rightarrow B1,\dots,Ak\rightarrow Bk}(r(N))$: rename attributes
- JOIN: combine tables
  - Natural Join ($r(N_1)$ * $r(N_2)$ or
  - Equi-Join $r(N_1) \bowtie_{A1=B1,\dots,Ak=Bk} r(N_2)$)
- CARTESIAN PRODUCT (x): combine tables
- Set operations
  - UNION ($\cup$),
  - INTERSECTION ($\cap$),
  - DIFFERENCE (or MINUS, **–**)
- Additional Relational Operations
  - OUTER JOINS, AGGREGATE FUNCTIONS, e.g., SUM, COUNT, AVG, MIN, MAX