



บทที่ 6 การจัดการกับข้อมูล (Data Manipulation) Part1 – Relational Algebra

โดย อ.อภิพงศ์ ปิงยศ

รายวิชา สธ312 ระบบการจัดการฐานข้อมูลทางธุรกิจ

Overview

- พีชคณิตเชิงสัมพันธ์ (Relational Algebra)
 - Traditional Set Operation คือ Union, Intersection, Difference, และ Cartesian Product
 - Special Relational Operation คือ Select, Project, Join และ Divide
 - คำสั่งเพิ่มเติมใน Relational Algebra คือ Rename, Extend, Summarize, Insert, Update และ Delete

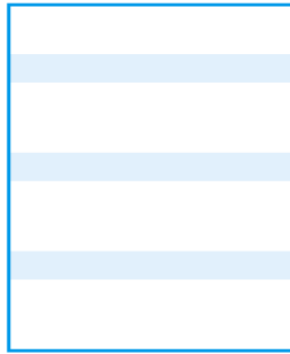
Relational Algebra

- คือ ทฤษฎีทางภาษาสำหรับปฏิบัติการระหว่างข้อมูลต่าง ๆ ในรีเลชันหนึ่งหรือหลาย ๆ รีเลชัน โดยผลของการปฏิบัติการจะไม่ส่งผลต่อรีเลชันหลัก
- พีชคณิตเชิงสัมพันธ์ (Relational Algebra) เป็นภาษาในรูปแบบของ Procedural Query Language ด้วยการพิจารณาว่าจะทำอย่างไรเพื่อให้ได้มาซึ่งผลลัพธ์ของข้อมูลตามที่ต้องการ

Traditional Set Operation

- เป็นการกระทำที่เกิดกับรีเลชันตามทฤษฎีของเซต โดยรีเลชันที่นำมากระทำในกลุ่มนี้จะต้องมีจำนวน Attribute ที่เท่ากัน และมีโดเมนเดียวกัน
- รูปแบบการสร้างประโยคภาษาใน Relational Algebra จะมี Operation พื้นฐานที่เกี่ยวกับเซตอยู่ 4 Operation คือ
 - Cartesian product
 - Union
 - Intersection
 - Difference

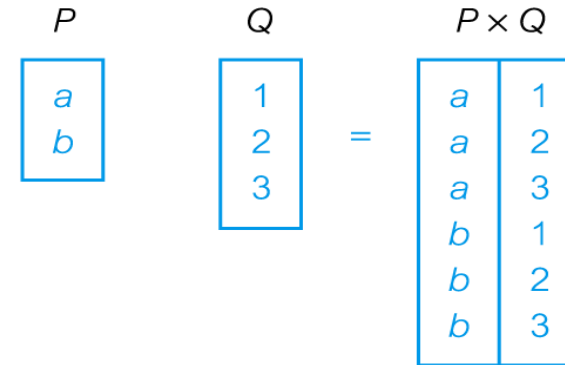
Relational Algebra Operations



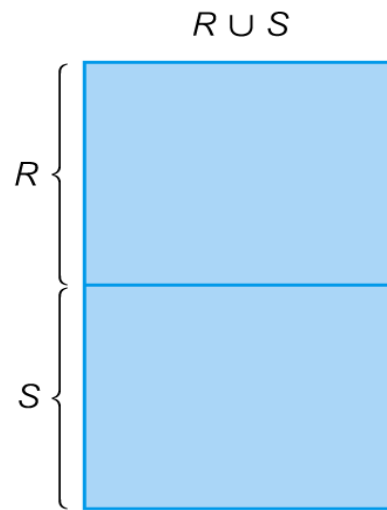
(a) Selection



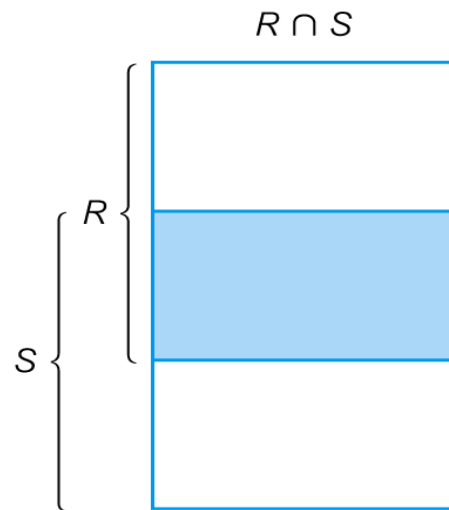
(b) Projection



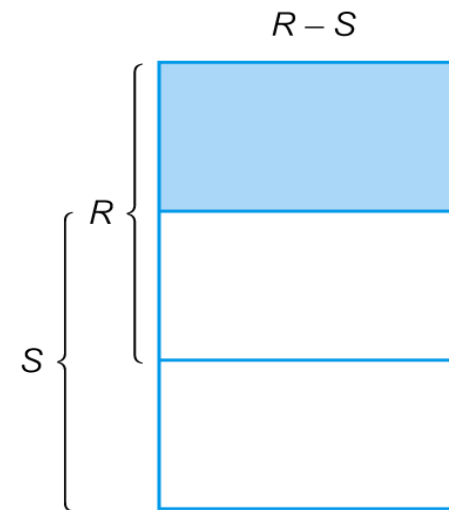
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference

Relational Algebra Operations

	T	
A	B	
a	1	
b	2	

U	
B	C
1	x
1	y
3	z

A	B	C
a	1	x
a	1	y

A	B
a	1

	$T \bowtie_c U$	
	A	B
C	a	x
	a	y
	b	2

(g) Natural join

(h) Semijoin

(i) Left Outer join

A diagram of a rectangle divided into four regions by a vertical line and a horizontal line. The top region is labeled R , the bottom region is labeled Remainder, and the left and right regions are unlabeled.

S

$$R \div S$$

V	
A	B
a	1
a	2
b	1
b	2
c	1

	W
B	
1	
2	

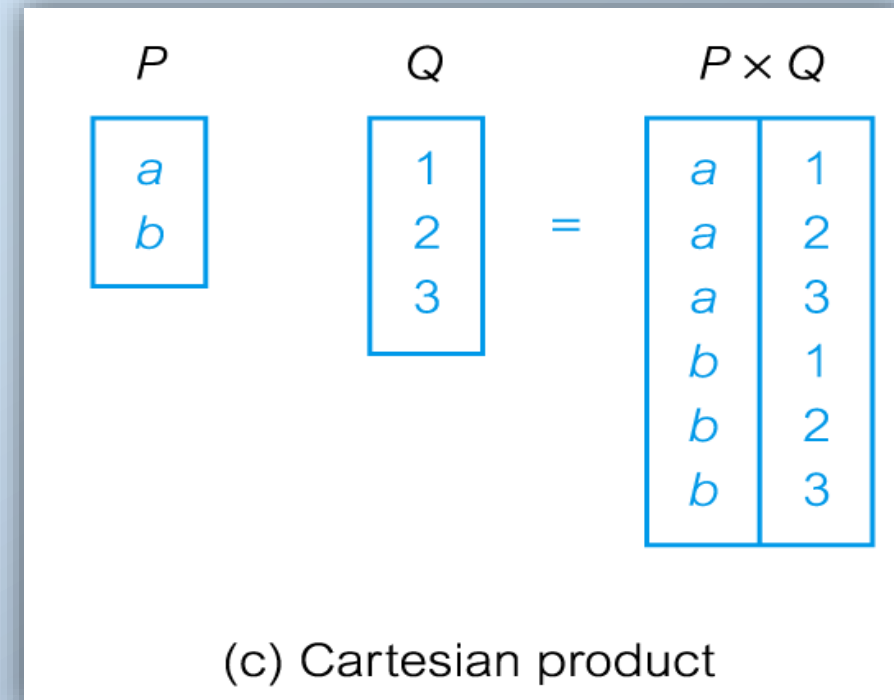
A
a b

(j) Divis on (shaded area)

Example of division

Cartesian product

- $R \times S$ หรือ $R \text{ TIMES } S$
 - เป็นการรวมความสัมพันธ์โดยเป็นผลคูณ Cartesian ระหว่าง 2 relations
 - จะได้จำนวน tuple ของ R ซึ่งทำการ join กับทุก ๆ tuple ใน S
 - R และ S ไม่จำเป็นต้อง union กันก็ได้



Example 1 - Cartesian product

R1

Name	Age	Sex
A	20	M
C	21	M

R2

Name	Age	Sex
D	20	F
E	21	F

$$R3 = R1 \times R2$$

Name	Age	Sex	Name	Age	Sex
A	20	M	D	20	F
C	21	M	D	20	F
A	20	M	E	21	F
C	21	M	E	21	F

Example 2 - Cartesian product

- List the names and comments of all clients who have viewed a property for rent.

$$(\Pi_{\text{clientNo}, \text{clientName}}(\text{CLIENT})) \times (\Pi_{\text{clientNo}, \text{propertyNo}, \text{comment}}(\text{VIEWING}))$$

clientNo	clientName
CR76	ยงยุทธ ชนเลิศ
CR56	สิราณี พรหมจรรย์
CR74	ศรีสมร หิรัญพงศ์
CR62	ตะวัน สงศรีสุข

×

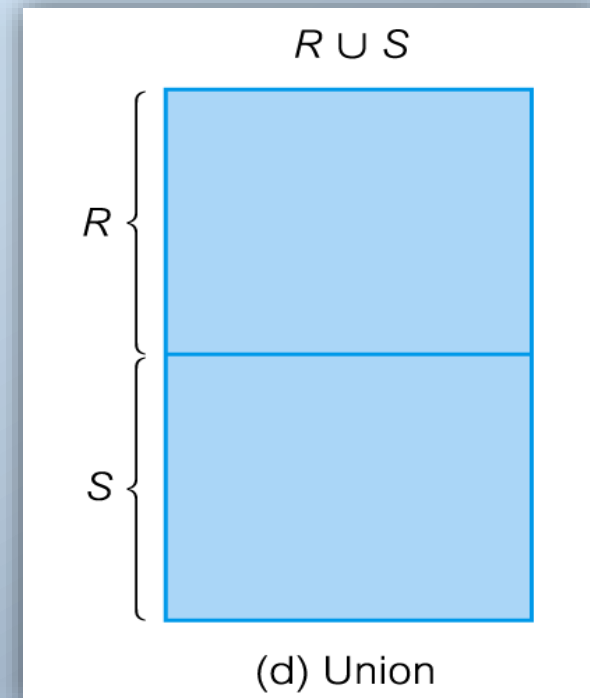
clientNo	propertyNo	comment
CR56	PA14	พื้นที่ขนาดเล็ก
CR76	PG04	ระยะทางไกลเกิน
CR56	PG04	
CR62	PA14	ไม่มีห้องครัว
CR56	PG36	

Result - Cartesian product

CLIENT.clientNo	clientName	VIEWING.clientNo	propertyNo	comment
CR76	ยงยุทธ ธนเลิศ	CR56	PA14	พื้นที่ขนาดเล็ก
CR76	ยงยุทธ ธนเลิศ	CR76	PG04	ระยะทางไกลเกิน
CR76	ยงยุทธ ธนเลิศ	CR56	PG04	
CR76	ยงยุทธ ธนเลิศ	CR62	PA14	ไม่มีห้องครัว
CR76	ยงยุทธ ธนเลิศ	CR56	PG36	
CR56	สิราณี พรหมจรรย์	CR56	PA14	พื้นที่ขนาดเล็ก
CR56	สิราณี พรหมจรรย์	CR76	PG04	ระยะทางไกลเกิน
CR56	สิราณี พรหมจรรย์	CR56	PG04	
CR56	สิราณี พรหมจรรย์	CR62	PA14	ไม่มีห้องครัว
CR56	สิราณี พรหมจรรย์	CR56	PG36	
CR74	ศรีสมร หิรัญพงศ์	CR56	PA14	พื้นที่ขนาดเล็ก
CR74	ศรีสมร หิรัญพงศ์	CR76	PG04	ระยะทางไกลเกิน
CR74	ศรีสมร หิรัญพงศ์	CR56	PG04	
CR74	ศรีสมร หิรัญพงศ์	CR62	PA14	ไม่มีห้องครัว
CR74	ศรีสมร หิรัญพงศ์	CR56	PG36	
CR62	ตะวัน สงศรีสุข	CR56	PA14	พื้นที่ขนาดเล็ก
CR62	ตะวัน สงศรีสุข	CR76	PG04	ระยะทางไกลเกิน
CR62	ตะวัน สงศรีสุข	CR56	PG04	
CR62	ตะวัน สงศรีสุข	CR62	PA14	ไม่มีห้องครัว
CR62	ตะวัน สงศรีสุข	CR56	PG36	

Union

- $R \cup S$ หรือ R UNION S
 - เป็นการเชื่อมความสัมพันธ์ด้วยการนำรีเลชัน R และ S มา union กัน
 - จะได้จำนวน tuple ทั้งหมดใน R และ S รวมกัน
 - Tuple ที่ซ้ำกันจะถูกขจัดออกไป



Example - Union

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

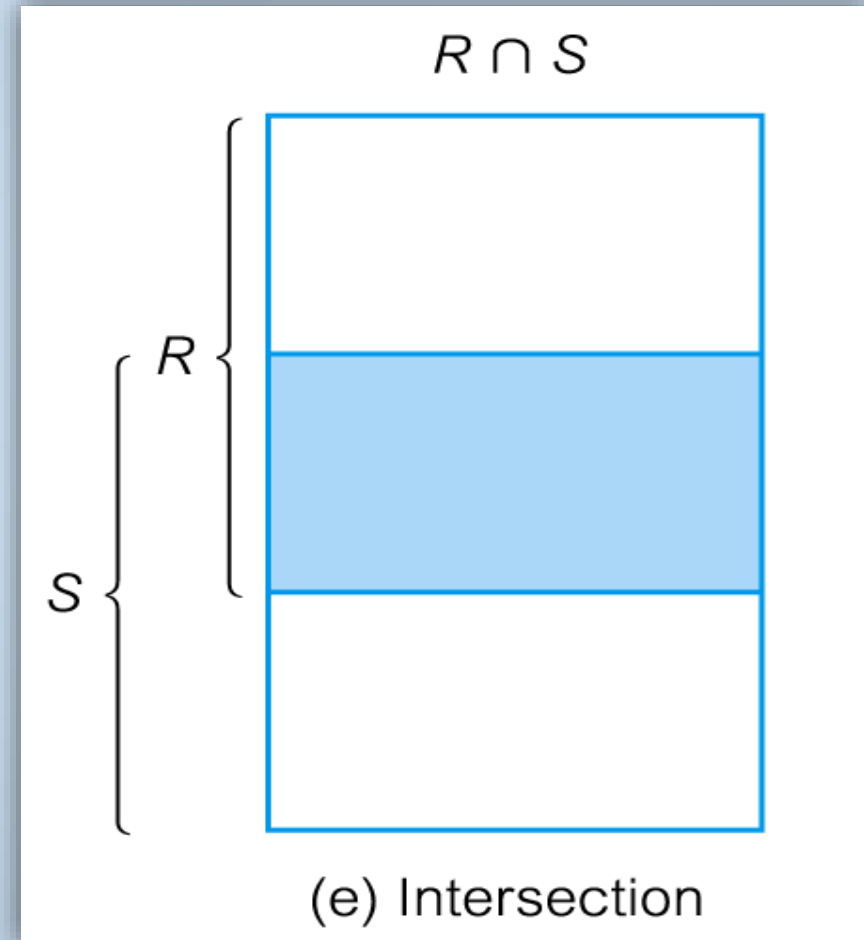
Name	Age	Sex
D	20	F
A	20	M
E	21	F

R3 = R1 U R2

Name	Age	Sex
A	20	M
C	21	M
B	21	F
D	20	F
E	21	F

Intersection

- $R \cap S$ หรือ $R \text{ INTERSECT } S$
 - จะได้กลุ่ม tuple ที่อยู่ทั้งใน R และ S



Example - Intersection

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

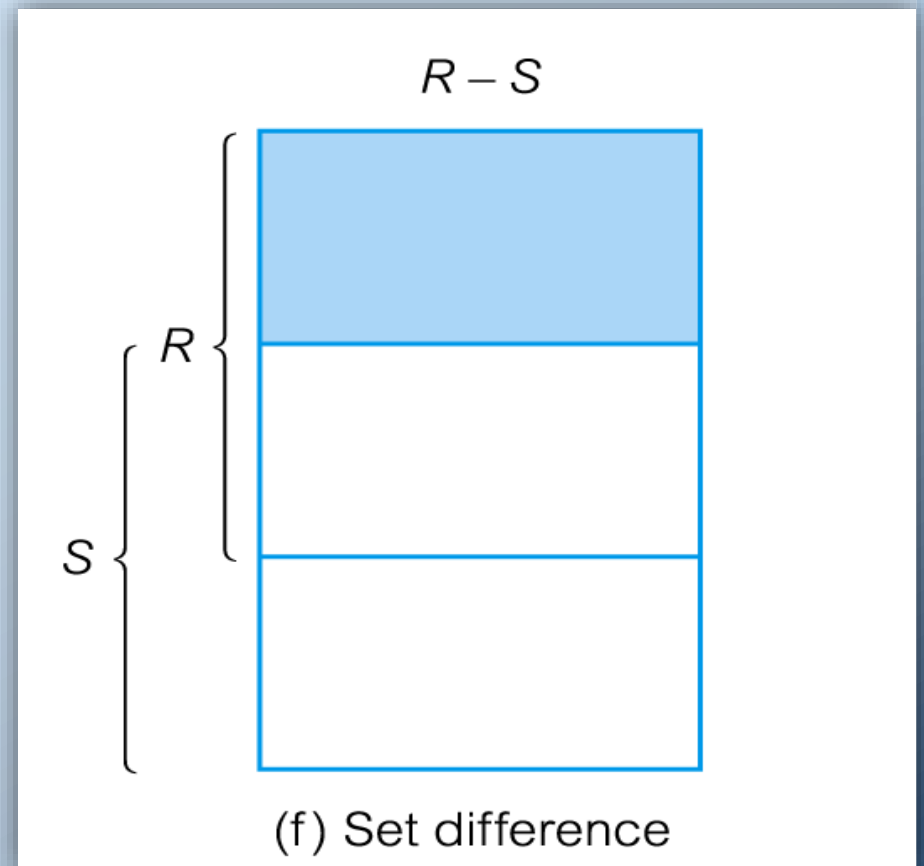
Name	Age	Sex
D	20	F
A	20	M
E	21	F

$R3 = R1 \cap R2$

Name	Age	Sex
A	20	M

Difference

- $R - S$ หรือ $R \text{ MINUS } S$
 - เป็นการหาความแตกต่างระหว่างสองความสัมพันธ์
 - จะได้ tuple ที่อยู่ใน R แต่ไม่อยู่ใน S



Example - Difference

R1

Name	Age	Sex
A	20	M
C	21	M
B	21	F

R2

Name	Age	Sex
D	20	F
A	20	M
E	21	F

$$R3 = R1 - R2$$

Name	Age	Sex
C	21	M
B	21	F

$$R3 = R2 - R1$$

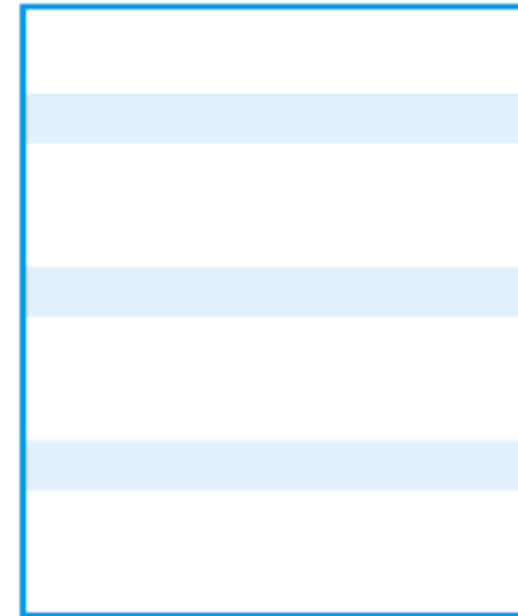
Name	Age	Sex
D	20	F
E	21	F

Special Relational Operation

- รีเลชันที่นำมากระทำในกลุ่มนี้ไม่จำเป็นต้องมีจำนวน Attribute ที่เท่ากัน แต่ Attribute ที่จะนำมากระทำกันจะต้องมีชื่อและโดเมนที่เหมือนกัน
- ตัวดำเนินการที่เกี่ยวข้องกับรีเลชันก็จะมีอยู่ 4 operation คือ
 - Select หรือ Restrict
 - Project
 - Join
 - Divide

Select (or Restrict)

- เป็นการเลือก Tuple ที่เป็นสับเซตในรีเลชัน เป็นการดำเนินการที่ใช้ในรีเลชันเดียว
- รูปแบบของการใช้ตัวดำเนินการคือ
- $\sigma_{\text{predicate}}(R)$
 - R คือรีเลชันที่ทำการ selection
 - Predicate คือ เงื่อนไขที่กำหนดขึ้น
- หรือ (R) **WHERE** predicate



(a) Selection

Example - Select (or Restrict)

R

Name	Age	Sex
A	20	M
M	21	F
B	20	F
F	19	M
A	20	F
R	21	F
C	21	M

$R1 = \sigma (\text{Age}=20)(R)$

Name	Age	Sex
A	20	M
B	20	F
A	20	F

R WHERE Age = 20

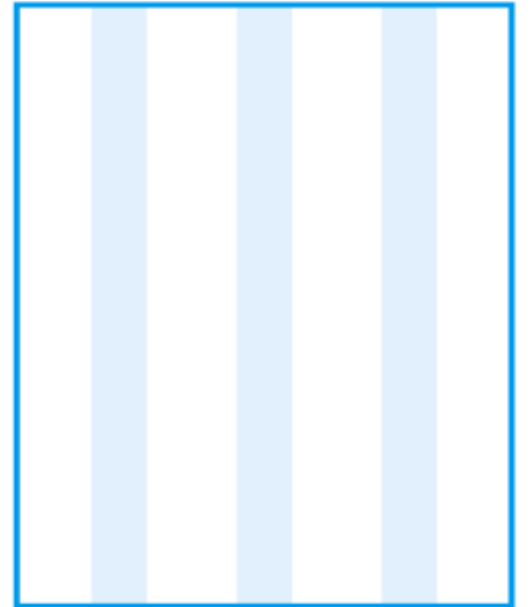
$R2 = \sigma (\text{Sex}=M \text{ AND } \text{Age}>19))(R)$

Name	Age	Sex
A	20	M
C	21	M

R WHERE Sex = M AND Age>19

Project

- เป็นการสร้างรีเลชันใหม่โดยการเลือกsubsetของ Attribute ในรีเลชัน โดยจะมีค่าของ Tuple ที่ไม่ซ้ำกัน
- $\Pi_{col_1, \dots, col_n}(R)$
 - เป็นการเลือกเฉพาะคอลัมน์หรือแอตทริบิวต์ที่ต้องการ
 - $col_1 \dots col_n$ คือ คอลัมน์หรือแอตทริบิวต์ที่เลือก
 - R คือ รีเลชัน
- หรือใช้รูปแบบ $R[attrib_1, \dots, attrib_n]$



(b) Projection

Example - Project

R

Name	Age	Sex
A	20	M
M	21	F
B	20	F
F	19	M
A	20	F
R	21	F
C	21	M

$R1 = \pi(\text{Name, Sex})(R)$

Name	Sex
A	M
M	F
B	F
F	M
A	F

$R2 = \pi(\text{Age, Sex})(R)$

Age	Sex
20	M
21	F
20	F
19	M

Join

- เป็นการรวม 2 รีเลชันไปเก็บไว้ในรีเลชันใหม่ ซึ่งทั้ง 2 รีเลชันจะต้องมี Attribute ที่มีชื่อและโดเมนที่เหมือนกัน
- เป็น operation ที่ถือกำเนิดมาจากผลคูณ Cartesian

T	
A	B
a	1
b	2

U	
B	C
1	x
1	y
3	z

$T \bowtie U$		
A	B	C
a	1	x
a	1	y

(g) Natural join

$T \bowtie_B U$	
A	B
a	1

(h) Semijoin

$T \bowtie_C U$		
A	B	C
a	1	x
a	1	y
b	2	

(i) Left Outer join

Join [cont.]

- $R \bowtie_F S$ หรือ $R \text{ JOIN } S$
- เป็นการหา tuple ที่เกิดจากผลคูณ Cartesian ของรีเลชัน R และ S ตามเงื่อนไขที่กำหนด
 - F คือ เงื่อนไขที่กำหนดขึ้น
 - โดยเงื่อนไขสามารถใช้ตัวดำเนินการในการเปรียบเทียบ ($<, \leq, >, \geq, =, \neq$)

Example - Join

S#	SNAME	STATUS	CITY
S1	ก.การช่าง	20	สมุทรปราการ
S2	ส.ชัยพลาย	10	สุพรรณบุรี
S3	แมชชีน	30	สุพรรณบุรี
S4	เอส.พี.	20	สมุทรปราการ

Relation: S

P#	PNAME	COLOR	WEIGHT	CITY
P1	เหล็กฉาก	แดง	12	สมุทรปราการ
P2	ปูนซีเมนต์	เขียว	17	สุพรรณบุรี
P4	ตะปู	แดง	14	สมุทรปราการ
P5	กระเบื้อง	น้ำเงิน	12	สุพรรณบุรี
P6	เหล็กเส้น	แดง	19	สมุทรปราการ

Relation: P

Example – Join (Result)

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT
S1	ก.การช่าง	20	สมุทรปราการ	P1	เหล็กฉาก	แดง	12
S1	ก.การช่าง	20	สมุทรปราการ	P4	ตะปู	แดง	14
S1	ก.การช่าง	20	สมุทรปราการ	P6	เหล็กเส้น	แดง	19
S2	ส.ซัพพลาย	10	สุพรรณบุรี	P2	ปูนซีเมนต์	เขียว	17
S2	ส.ซัพพลาย	10	สุพรรณบุรี	P5	กระเบื้อง	น้ำเงิน	12
S3	แมชชีน	30	สุพรรณบุรี	P2	ปูนซีเมนต์	เขียว	17
S3	แมชชีน	30	สุพรรณบุรี	P5	กระเบื้อง	น้ำเงิน	12
S4	เอส.พี	20	สมุทรปราการ	P1	เหล็กฉาก	แดง	12
S4	เอส.พี	20	สมุทรปราการ	P4	ตะปู	แดง	14
S4	เอส.พี	20	สมุทรปราการ	P6	เหล็กเส้น	แดง	19

Divide

- $R \div S$ หรือ R DIVIDE BY S
 - เป็นโอเปอเรชันที่นำรีเลชันสองรีเลชันมาแบ่งหรือหารกัน
 - เป็นการค้นหากลุ่มของ tuple ที่มีค่าแอตทริบิวต์เหมือนกับแอตทริบิวต์ของทุก ๆ tuple ในอีกกลุ่มหนึ่ง
 - มีกฎว่า รีเลชันที่เป็นตัวหาร ต้องเป็นสับเซตของรีเลชันที่เป็นตัวตั้ง
 - สมมติว่ารีเลชัน R มีดีกรีเป็น $(n+m)$ หารด้วยรีเลชัน S ที่มีดีกรีเท่ากับ m ผลลัพธ์ที่ได้คือ รีเลชันที่มีดีกรีเท่ากับ n

Divide [cont.]

- การดำเนินการคือ ทุก Tuple ในรีเลชัน R ที่มีค่าตรงกับทุก Tuple ในรีเลชัน S แต่ในการแสดงผลจะแสดงเฉพาะแอตทริบิวต์ที่ปรากฏใน R แต่ไม่ปรากฏใน S
- ตัวอย่าง $V \div W$

V		W	$V \div W$
A	B	B	A
a	1	1	a
a	2	2	b
b	1		
b	2		
c	1		

Example of division

Example - Divide

R1

Name	Sex
A	M
B	F
A	F
C	F
D	M
C	M

$R3 = R1 \div R2$

Name
A
C

R2

Sex
M
F

คำสั่งเพิ่มเติมใน Relational Algebra

- เป็นคำสั่งที่เพิ่มเติมขึ้นมาเพื่อช่วยให้คำสั่งใน Relational Algebra มีความคล่องตัวมากยิ่งขึ้น ประกอบด้วย
 - RENAME
 - EXTEND
 - SUMMARIZE
 - INSERT
 - UPDATE
 - DELETE

Rename

- เป็นคำสั่งสำหรับเปลี่ยนชื่อของ Attribute
- รูปแบบคือ R **RENAME** old-attribute **AS** new-attribute
- โดยที่
 - R คือ Relation ที่เป็นเจ้าของ Attribute
 - old-attribute คือชื่อเดิมของ Attribute
 - new-attribute คือชื่อใหม่ของ Attribute
- เช่น S **RENAME** CITY **AS** SCITY

EXTEND

- เป็นคำสั่งที่ใช้เพิ่ม Attribute ใหม่ให้กับรีเลชัน
- มีรูปแบบคือ **EXTEND R ADD value AS attribute**
- โดยที่
 - R คือ รีเลชันที่ต้องการเพิ่ม Attribute
 - value คือ ค่าที่กำหนดให้กับทุกๆ Tuple ใน Attribute ใหม่
 - attribute คือ ชื่อของ Attribute ใหม่ที่เพิ่มเข้าไป
- เช่น **EXTEND S ADD “Supplier” AS TAG**

Summarize

- เป็นคำสั่งที่ใช้สำหรับหาค่าของ Attribute ใด Attribute หนึ่ง ด้วยฟังก์ชันใดๆ
- รูปแบบคือ **SUMMARIZE R By (a_1, a_2, \dots, a_n) ADD f_n AS z**
- โดยที่
 - R คือชื่อของ Relation ที่จะ Summarize
 - a_i คือ ชื่อของ Attribute ที่จะใช้ Summarize
 - f_n คือ ฟังก์ชันที่จะนำมาใช้ เช่น SUM, COUNT หรือ AVG
 - z คือ ชื่อของ Attribute ที่ใช้เก็บผลลัพธ์ที่ได้จากการ Summarize
- เช่น **SUMMARIZE SP BY () ADD SUM (QTY) AS GRANTOTAL**

INSERT

- เป็นคำสั่งที่ใช้เพิ่ม Tuple ทุก Tuple จาก Relation หนึ่ง ให้กับอีก Relation หนึ่ง
- รูปแบบคือ **INSERT** source **INTO** target
- โดยที่
 - source คือ ชื่อ Relation ต้นทาง
 - target คือ Relation ที่รับค่าจาก source
- เช่น **INSERT (S WHERE CITY = “สมุทรปราการ”) INTO TEMP**

UPDATE

- เป็นคำสั่งที่ใช้สำหรับเปลี่ยนแปลงค่าของ Attribute ใด Attribute หนึ่ง ใน Relation
- รูปแบบคือ **UPDATE** target assignment-value
- โดยที่
 - target คือ ชื่อ Relation ที่ต้องการเปลี่ยนค่า
 - assignment-value คือ เงื่อนไขในการเปลี่ยนค่าให้กับ Attribute
- เช่น **UPDATE S WHERE COLOR = “แดง” CITY := “สุพรรณบุรี”**

DELETE

- เป็นคำสั่งใช้สำหรับลบ Tuple ออกจาก Relation
- มีรูปแบบคือ **DELETE** target
- โดยที่ target คือเงื่อนไขในการลบ
- เช่น **DELETE S WHERE STATUS<20**