

Database Management Systems

Part II: The Relational Model

Lecture 6

Relational Operators I: Relational Algebra

Contents

- Relational Algebra
- Traditional Set Operations
- Special Relational Operations
- Extend and Summarize Operations
- Relational Comparisons

Relational Algebra

- The third and last part of the relational model is **the operators**.
- The principal component of the model is **the relational algebra**.
- Relational algebra essentially consists of a collection of operators, such as join, that **take relations as their operands** and **return relations as their result**.
- There are two groups of relational operators: **Traditional Set Operators** and **Special Relational Operators**.

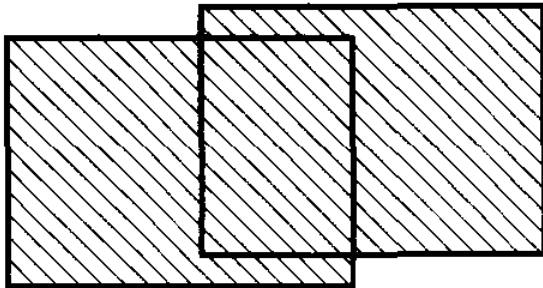
Traditional Set Operators

- Traditional set operator includes four operators:
 1. Union
 2. Intersection
 3. Difference
 4. Cartesian Product

Traditional Set Operators (Cont.)

1. Union

- The **union** of two **type-compatible relations** (A and B) returns a relation consisting of all tuples appearing in either or both of two specified relations.



A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

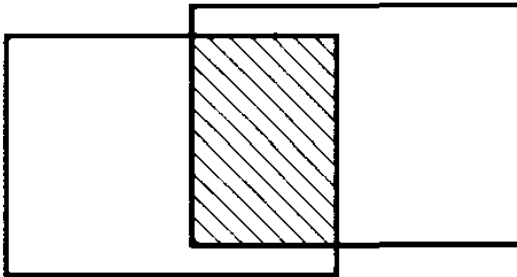
A UNION B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London
S2	Jones	10	Paris

Traditional Set Operators (Cont.)

2. Intersection

- The **intersection** of two **type-compatible relations** (A and B) returns a relation consisting of all tuples appearing in both of two specified relations.



A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

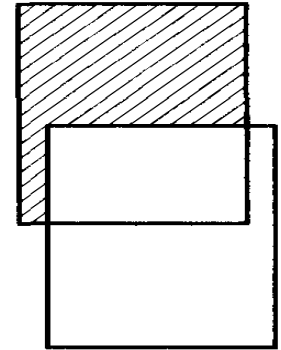
A INTERSECT B

S#	SNAME	STATUS	CITY
S1	Smith	20	London

Traditional Set Operators (Cont.)

3. Difference

- The **difference** between two **type-compatible relations** (A and B) returns a relation consisting of all tuples appearing in the first and not the second of two specified relations.



A

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris

A MINUS B

S#	SNAME	STATUS	CITY
S4	Clark	20	London

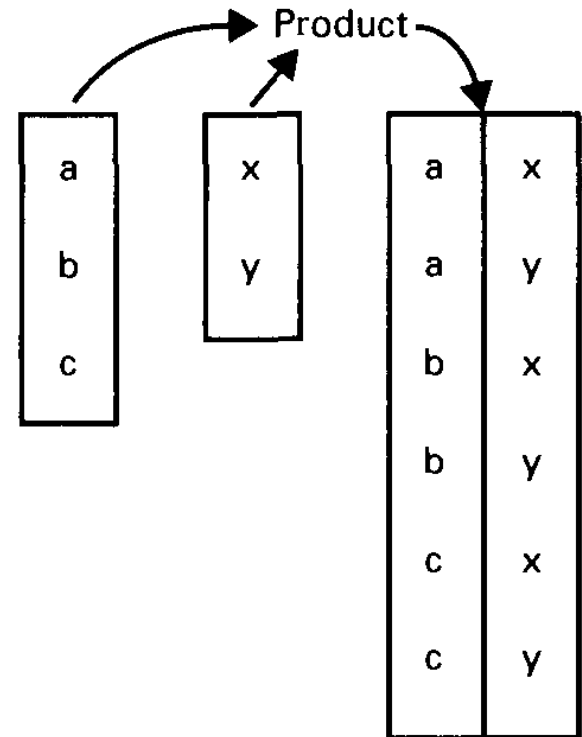
B MINUS A

S#	SNAME	STATUS	CITY
S2	Jones	10	Paris

Traditional Set Operators (Cont.)

4. Product

- The **product operation** returns a relation consisting of all possible tuples that are a combination of two tuples, one from each of two specified relations.



Traditional Set Operators (Cont.)

4. Product

- Figure shows an example of product operation.

A

S#
S1
S2
S3
S4
S5

B

P#
P1
P2
P3
P4
P5
P6

S#	P#
S1	P1	S2	P1	S3	P1	S4	P1	S5	P1
S1	P2	S2	P2	S3	P2	S4	P2	S5	P2
S1	P3	S2	P3	S3	P3	S4	P3	S5	P3
S1	P4	S2	P4	S3	P4	S4	P4	S5	P4
S1	P5	S2	P5	S3	P5	S4	P5	S5	P5
S1	P6	S2	P6	S3	P6	S4	P6	S5	P6
..

Special relational operators

- Special relational operators includes four operators:
 1. Restrict
 2. Project
 3. Join
 4. Divide

Special relational operators (Cont.)

1. Restrict

- The **restrict operation** returns a relation consisting of all tuples from a specified relation that satisfy a specified condition.

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

S WHERE CITY = 'LONDON'

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

Special relational operators (Cont.)

1. Restrict

- Some restriction examples are as follows.

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

SP WHERE S# = 'S1' AND P# = 'P1'

S#	P#	QTY
S1	P1	300

P WHERE WEIGHT < 14

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P5	Cam	Blue	12	Paris

Special relational operators (Cont.)

2. Project

- The **project operation** returns a relation consisting of all tuples that remain as tuples in a specified relation after specified attributes have been eliminated.

S	S#	SNAME	STATUS	CITY	S [CITY]
	S1	Smith	20	London	CITY
	S2	Jones	10	Paris	London
	S3	Blake	30	Paris	Paris
	S4	Clark	20	London	Athens
	S5	Adams	30	Athens	

Special relational operators (Cont.)

2. Project

- Some project operation examples are as follows.

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P [COLOR, CITY]

COLOR	CITY
Red	London
Green	Paris
Blue	Rome
Blue	Paris

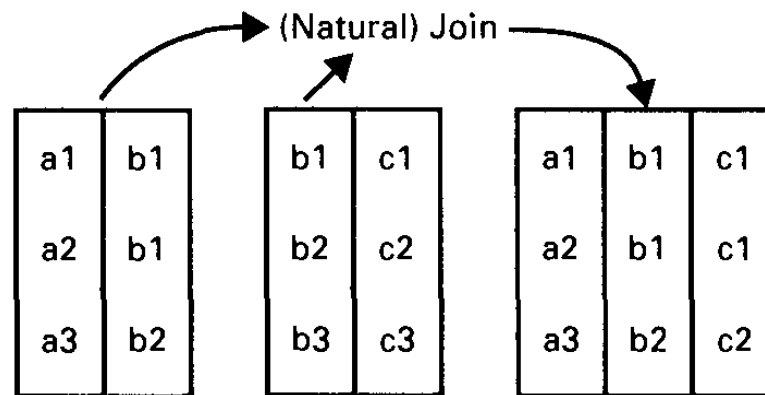
(S WHERE CITY = 'PARIS') [S#]

S#
S2
S3

Special relational operators (Cont.)

3. Join

- The **natural join operation** returns a relation consisting of all possible tuples that are a combination of two tuples, one from each of two specified relations, such that the two tuples contributing to any given combination have a common value for the common attribute(s) of the two relations.



Special relational operators (Cont.)

3. Join

- Figure shows an example of join operation.

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

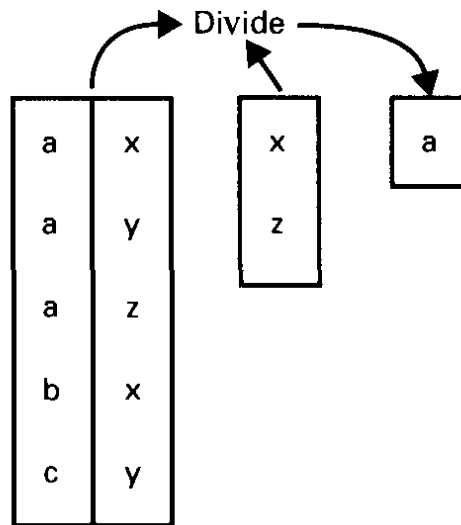
S JOIN P

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT
S1	Smith	20	London	P1	Nut	Red	12
S1	Smith	20	London	P4	Screw	Red	14
S1	Smith	20	London	P6	Cog	Red	19
S2	Jones	10	Paris	P2	Bolt	Green	17
S2	Jones	10	Paris	P5	Cam	Blue	12
S3	Blake	30	Paris	P2	Bolt	Green	17
S3	Blake	30	Paris	P5	Cam	Blue	12
S4	Clark	20	London	P1	Nut	Red	12
S4	Clark	20	London	P4	Screw	Red	14
S4	Clark	20	London	P6	Cog	Red	19

Special relational operators (Cont.)

4. Divide

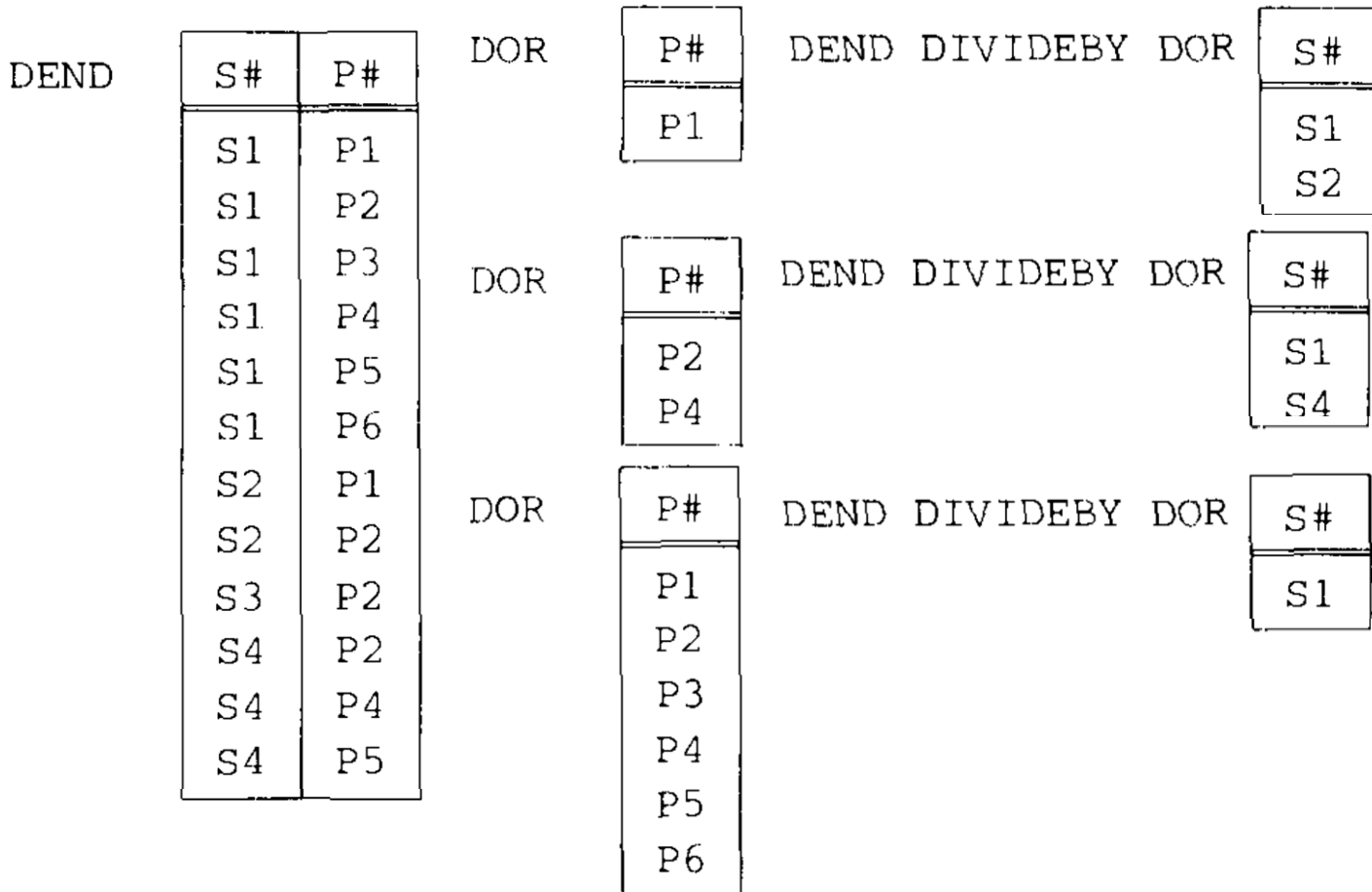
- The **divide operation** takes two relations, one binary and one unary, and returns a relation consisting of all values of one attribute of the binary relation that match all values in the unary relation.



Special relational operators (Cont.)

4. Divide

- Figures show some examples of divide operation.



Extend Operation

- **Extend** takes a specified relation and **makes a new relation** that is similar to the original relation but **includes an additional attribute**, values of which are obtained **by evaluating some scalar computational expression**.

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

EXTEND P ADD (WEIGHT * 454) AS GMWT

P#	PNAME	COLOR	WEIGHT	CITY	GMWT
P1	Nut	Red	12	London	5448
P2	Bolt	Green	17	Paris	7718
P3	Screw	Blue	17	Rome	7718
P4	Screw	Red	14	London	6356
P5	Cam	Blue	12	Paris	5448
P6	Cog	Red	19	London	8626

Extend Operation (Cont.)

- The result of the following example looks like this:

SP	S#	P#	QTY
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200
	S4	P4	300
	S4	P5	400

S	S#	SNAME	STATUS	CITY
	S1	Smith	20	London
	S2	Jones	10	Paris
	S3	Blake	30	Paris
	S4	Clark	20	London
	S5	Adams	30	Athens

EXTEND S ADD COUNT ((SP RENAME S# AS X) WHERE X=S#) AS NP

S#	SNAME	STATUS	CITY	NP
S1	Smith	20	London	6
S2	Jones	10	Paris	2
S3	Blake	30	Paris	1
S4	Clark	20	London	3
S5	Adams	30	Athens	0

Summarize Operation

- The **summarize operation** performs the analogous function for vertical or column-wise computations.

SP	S#	P#	QTY
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200
	S4	P4	300
	S4	P5	400

SUMMARIZE SP BY (P#) ADD SUM (QTY) AS TOTQTY

P#	TOTQTY
P1	600
P2	1000
P3	400
P4	500
P5	500
P6	100

Summarize Operation (Cont.)

- The result of the following example looks like this:

SP	S#	P#	QTY
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200
	S4	P4	300
	S4	P5	400

P	P#	PNAME	COLOR	WEIGHT	CITY
	P1	Nut	Red	12	London
	P2	Bolt	Green	17	Paris
	P3	Screw	Blue	17	Rome
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

SUMMARIZE (P JOIN SP) BY (CITY) ADD COUNT AS NSP

CITY	NSP
London	5
Paris	6
Rome	1

Summarize Operation (Cont.)

- The result of this second example looks like this:

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

SUMMARIZE SP BY (S#) ADD COUNT AS NP

S#	NP
S1	6
S2	2
S3	1
S4	3

Relational Comparisons

- The relational algebra includes the following **comparison operators**:

= (equals)

≠ (not equals)

≤ (subset of)

< (proper subset of)

≥ (superset of)

> (proper superset of)

Relational Comparisons (Cont.)

- Here are some examples:

S [CITY] = P [CITY]

S [S#] > SP [S#]

S WHERE ((SP RENAME S# AS X) WHERE X = S#) [P#] = P [P#]

Summary

- The original algebra consisted of **eight operations**.
- Traditional set operations are **union, intersection, difference and product**.
- Special relational operations are **restrict, project, join and divide**.
- For certain of these operations, the two operands are required to be **type-compatible**.

Next Lecture

Part II: The Relational Model

Relational Operators II

- Relational Calculus
- Tuple-Oriented Relational Calculus
- Domain-Oriented Relational Calculus

Textbook and References

Textbook

- C. J. Date, “An Introduction to Database Systems”, 6th Edition, 1994.

Additional References

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, 6th Edition, 2011.
- Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, 6th Edition, 2010.
- <https://www.guru99.com>
- [https:// www.includehelp.com](https://www.includehelp.com)
- [https:// www.tutorialspoint.com](https://www.tutorialspoint.com)