Relational Algebra

The Theory behind Relational Databases

Relational Algebra: What and Why

- Ted Codd introduced relational algebra to databases and created the relational model.
- Relational algebra provides a theoretical foundation for relational databases, and particularly for query languages like SQL.
- Why do you want a theoretical foundation?
  - If you want to optimize a query or a database
  - If you are thinking about using NOSQL, then you should be aware of the limitations and advantages of NOSQL data management. In other words, relational algebra assists in comparing SQL with NOSQL (**NO**T-SQL, **N**ot-**O**nly-SQL, K**NO**W-SQL, http://www.youtube.com/watch?v=sh1YACOK_bo)

New Terminology (1)

Term	Comments
Table	Part of a database
Relation	A table where rows are unique. Operand in Relational Algebra/Calculus
Tuple	single, double, triple, qudruple, quintuple, sextuple; Like a row in a table
Arity	unary, binary, ternary, quaternary
Closure	Operation on a type produces a value of that same type. Natural Numbers have closure under + and * (3 * 5 = 15) Natural Numbers do not have closure under – or /; 5 – 3 = -2

New Terminology (2)

Term	Comments
Procedural	Step-by-step solution to solving problem or achieving goal. I will drive to Bellevue, enter the class room and listen to the lecture. (Relational Algebra is procedural or imperative)
Declarative	Stating what one wants in non-ambiguous terms without describing how one is to achieve ones goal. Example: I want to know what was said in class last week. I don't care if you use the slide deck, your memory, or the recording to get me that information. (SQL is declarative)
Relational Algebra	The algebra that describes relations as operands and results
Relational Calculus	The calculus that uses relations as operands and results (SQL)

New Terminology (3)

Operation	Symbols	Comments
Selection	σ (sigma); $\sigma_{\varphi}(R)$;	SELECT * FROM <table name> WHERE Column1 = 1
Projection	π (pi); $\pi_{c1,\, c2,\, \ldots,\, cn}(R)$	SELECT Column1, Column 2 FROM <table name>
Rename	P (rho)	
Union	∪	A∪B; A={1,2,3, 5}; B={0,2}; {1,2,3, 5}∪{0,2}={0,1,2,3,5}
Intersection	∩	A∩B; A={1,2,3, 5}; B={0,2}; {1,2,3, 5} ∩{0,2}= {2}
Difference	\, -,	B\A = B-A; {0,2} - {1,2,3,5} = {0}

New Terminology (4)

Operation	Symbols	Comments
Product	X	AXB A={1,2,3,5}; B={0,2}; {1,2,3, 5} X{0,2}= {{1,0}, {2,0}, {3,0}, {5,0}, {1,2}, {2,2}, {3,2}, {5,2}}
Join	$\bowtie_{\varphi}$	$B \bowtie_{\varphi} A$; φ: A > B; A={1,2,3,5}; B={0,2}; {1,2,3,5} $\bowtie_{\varphi}${0,2} = {{1,0},{2,0},{3,0},{3,2},{5,0},{5,2}}
Division	÷	A÷B = C; Project to show me the columns in A that are not in B; Select to show me the tuples in A that are a superset of the a tuple in B.

Relational Algebra

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Relation

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relation

Relational Algebra: Relation

Relation is like a table except that each row must be unique like in a set

Relation

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Attribute

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Attribute

Relational Algebra: Attribute

Attribute:

Must be of the same data type.

Have a name

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Attribute

Relational Algebra: Tuple

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

tuple

Relational Algebra: Tuple

tuple from: sing**le**, doub**le**, tri**ple**, quadr**uple**, quin**tuple**
arity from: un**ary**, bin**ary**, tern**ary**

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

tuple with arity of 3

Relational Algebra: Operands and Simple Operations

- Operand
  - Relation (Table)
- Operations
  - UNION
  - INTERSECT
  - PROJECT
  - SELECT
  - PRODUCT
  - DIVISION

Relational Algebra: Union

Combine Relations

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Union

Combine Relations

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:

R ∪ S

Relational Algebra: Union

Combine Relations

SQL Statement:

SELECT * FROM MyTableR UNION
SELECT * FROM MyTableS

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent

Name	**Age**	**Home**
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:

$R \cup S$

Relational Algebra: Union

Combine Relations

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent

Name	**Age**	**Home**
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:

R ∪ S

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Intersection:
$R \cap S$

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Same Rows

SQL Statement:
SELECT * FROM MyTableR
INTERSECT
SELECT * FROM MyTableS

Relational Algebra Intersection:
$R \cap S$

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	Age	Home
Menchú	31	Rent
Yamana	50	Own

Relational Algebra Intersection:
$R \cap S$

Relational Algebra: Examples

- $R \cup S$
  - SELECT * FROM MyTableR UNION SELECT * FROM MyTableS

- SELECT * FROM MyTableR UNION SELECT * FROM MyTableS
  - $R \cup S$ or $S \cup R$

- $R \cap S$
  - SELECT * FROM MyTableR INTERSECT SELECT * FROM MyTableS

- SELECT * FROM MyTableR INTERSECT SELECT * FROM MyTableS
  - $R \cap S$ or $S \cap R$

- In General:
  - An operation with $\cup$ or $\cap$ produces a relation
  - $R \cup S = S \cup R$
  - $R \cap S = S \cap R$
  - $(R \cup S) \cap T = (R \cap T) \cup (S \cap T)$
  - $(R \cap S) \cup T = (R \cup T) \cap (S \cup T)$

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Vertical partition

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Vertical partition

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

SQL Statement:

SELECT Age, Home FROM MyTable

Vertical partition

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Age	Home
5	None
21	Rent
31	Rent
42	Rent
50	Own

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Age	Home
5	None
21	Rent
31	Rent
42	Rent
50	Own

The result of a projection is a relation with 0 to n attributes where n is the number of attributes in the operand

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Horizontal partition

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Horizontal partition

Relational Algebra Select:

$\sigma_{\varphi}(R)$

where

φ: Home = "Rent"

R: MyTable

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

SQL Statement:
SELECT * FROM MyTable WHERE Home = “Rent”

Horizontal partition

Relational Algebra Select:
$\sigma_{\varphi}(R)$
where
φ: Home = “Rent”
R: MyTable

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	Age	Home
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent

The result of a selection is a relation with 0 to n tuples where n is the number of tuples in the operand

Relational Algebra Select:

$\sigma_{\varphi}(R)$

where

φ: Home = "Rent"

R: MyTable

Relational Algebra: Examples

- $\pi_{Age,Home}(R)$
  - SELECT Age, Home FROM MyTable
- $\sigma_{Home=\text{“Rent”}}(R)$
  - SELECT * FROM MyTable WHERE Home = “Rent”
- SELECT Age, Home FROM MyTable WHERE Home = “Rent”
  - $\pi_{Age,Home}(\sigma_{Home=\text{“Rent”}}(R))$ or $\sigma_{Home=\text{“Rent”}}(\pi_{Age,Home}(R))$
- In General:
  - An operation with σ produces a relation
  - An operation with π produces a relation
  - $\sigma_{\varphi 1}(\sigma_{\varphi 2}(R)) = \sigma_{\varphi 2}(\sigma_{\varphi 1}(R))$
  - $\pi_{[c1]}(\pi_{[c2]}(R)) = \pi_{[c2]}(\pi_{[c1]}(R))$
  - $\pi_{[c]}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{[c]}(R))$ (**only if** φ is not dependent on [c])

Relational Algebra: Product

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

SQL Statement:
SELECT * FROM TableR, TableS

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
		Alvarez	Rent
		Yamana	Own
Kobayashi	21	Menchú	Rent
		Alvarez	Rent
		Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21		
Blackburn	5	Alvarez	Rent
Kobayashi	21		
Blackburn	5	Yamana	Own
Kobayashi	21		

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

The result of a product is a relation with n*m tuples where n and m are the number of tuples in the operands. The arity of the result is i + j where i and j are the arities of the operands

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

The result of a product is a relation with n*m tuples where n and m are the number of tuples in the operands. The arity of the result is i + j where i and j are the arities of the operands

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Join

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product with Select:

$\sigma_{\varphi}(R \times S)$ where φ: Home = “Rent”

Relational Algebra Join:

$R \bowtie_{\varphi} S$ where φ: Home = “Rent”

Relational Algebra: Join

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product with Select:

σ_{φ}(R X S) where φ: Home = “Rent”

Relational Algebra Join:

R $\bowtie_{\varphi}$ S where φ: Home = “Rent”

Relational Algebra: Join

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent

Relational Algebra Product with Select:

$\sigma_{\varphi}(R \times S)$ where φ: Home = “Rent”

Relational Algebra Join:

$R \bowtie_{\varphi} S$ where φ: Home = “Rent”

Relational Algebra: Join

- A Join is a Product with a select statement

- Product followed by Select
  - SELECT * FROM TableR, TableS WHERE Home = “Rent”
  - $\sigma_{\varphi}(R \times S)$ where φ: Home = “Rent”

- JOIN
  - SELECT * FROM TableR JOIN TableS ON Home = “Rent”
  - $R \bowtie_{\varphi} S$ where φ: Home = “Rent”

Relational Algebra: Division

A Division is sort of like the reverse of a Product

This was a Product Operand

This was the result of a Product

Name	**Age**
Blackburn	5
Kobayashi	21

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

This was a Product Operand

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

Exclude columns

Name	Age
Blackburn	5
Kobayashi	21

Result of Projection

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Project

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

Select

Name	Age
Blackburn	5
Kobayashi	21

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Result of Selection

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity i where the dividend operand contains n*m tuples of arity i + j that are a superset of the result tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:

$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the dividend operand has n*m tuples of arity i + j and the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the dividend operand has n*m tuples of arity i + j and the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Resources

- Relational Algebra and SQL
  - RelationalAlgebraAndSQL.pdf
  - RelationalAlgebraAndSQL.sql

- http://en.wikipedia.org/wiki/Cartesian_product
- http://en.wikipedia.org/wiki/Commutative_property
- http://en.wikipedia.org/wiki/Associative_property
- http://en.wikipedia.org/wiki/Closure_(mathematics)

Relational Algebra

Assignment (1)

1. {a, b, c} is a relation that contains the tuples a, b, and c. In the following cases the tuples have arity of 1. Calculate the following:
   a. $(\{1, 2, 3\} \cup \{5, 7, 11\}) \cap \{2, 4, 6, 8, 10\}$
   b. $(\{1, 2, 3\} \cap \{2, 4, 6, 8, 10\}) \cup (\{5, 7, 11\} \cap \{2, 4, 6, 8, 10\})$
2. Use formal notation to write an algebraic example of the following SQL:
   a. SELECT Column1, Column3 FROM MyTable WHERE Column2 = Column3
   b. Reverse the order of projection and selection in your algebraic formulation. What happened?
3. $\pi_{c1,\, c2}(\sigma_{\varphi 1}(\sigma_{\varphi 2}(\pi_{c1,\, c2,\, c3,\, c5}(R))))$

   Where
   - $\varphi 1$: C1 = C5;
   - $\varphi 2$: C5 = "Test";
   - R: MyTable;

   a. Write a SQL statement that declares the intent of the algebraic notation
   b. Simply the algebraic statement

Assignment (2)

4. SELECT * FROM T1 JOIN T2 ON T1.C1 = T2.C1
   a. Write out an equivalent in relational algebra using the join operator
   b. Write out an equivalent in relational algebra without using the join operator
5. $\pi_{S.C1,\ R.C2}(\sigma_{\varphi 1}(R) \bowtie_{\varphi 2} S)$
   - where
     - $\varphi 1 = (R.C2 = \text{'A'})$
     - $\varphi 2 = (R.C1 = S.C2)$
   - Write out equivalent SQL and test this SQL using relations R and S that you create for this example. The relations R and S in RelationalAlgebraAndSQL.pdf and RelationalAlgebraAndSQL.sql don’t quite work because their column types do not match for this assignment.