

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 (NOT-SQL, Not-Oonly-SQL, KNOW-SQL, http://www.youtube.com/watch?v=sh1YACOK_bo)

New Terminology (1)

Term	Comments
<u>Table</u>	Part of a database
<u>Relation</u>	A table where rows are unique. Operand in Relational Algebra/Calculus
<u>Tuple</u>	<u>single</u> , <u>double</u> , <u>triple</u> , <u>quadruple</u> , <u>quintuple</u> , <u>sextuple</u> ; Like a row in a table
<u>Arity</u>	<u>unary</u> , <u>binary</u> , <u>ternary</u> , <u>quaternary</u>
<u>Closure</u>	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
<u>Procedural</u>	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 <u>procedural</u> or <u>imperative</u>)
<u>Declarative</u>	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 <u>declarative</u>)
<u>Relational Algebra</u>	The algebra that describes relations as operands and results
<u>Relational Calculus</u>	The calculus that uses relations as operands and results (SQL)

New Terminology (3)

Operation	Symbols	Comments
<u>Selection</u>	σ (sigma); $\sigma_{\varphi}(R)$;	SELECT * FROM <table name> <u>WHERE</u> <u>Column1 = 1</u>
<u>Projection</u>	π (pi); $\pi_{c_1, c_2, \dots, c_n}(R)$	SELECT <u>Column1, Column 2</u> FROM <table name>
<u>Rename</u>	ρ (rho)	
<u>Union</u>	\cup	$A \cup B$; $A = \{1, 2, 3, 5\}$; $B = \{0, 2\}$; $\{1, 2, 3, 5\} \cup \{0, 2\} = \{0, 1, 2, 3, 5\}$
<u>Intersection</u>	\cap	$A \cap B$; $A = \{1, 2, 3, 5\}$; $B = \{0, 2\}$; $\{1, 2, 3, 5\} \cap \{0, 2\} = \{2\}$
<u>Difference</u>	\setminus , -	$B \setminus A = B - A$; $\{0, 2\} - \{1, 2, 3, 5\} = \{0\}$

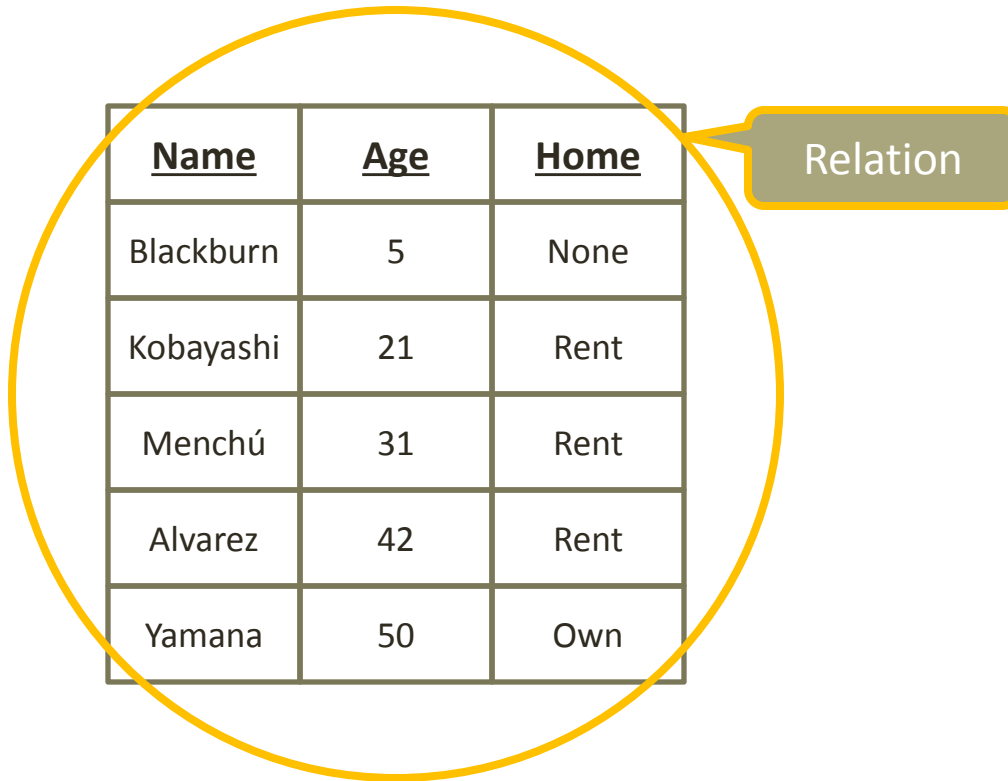
New Terminology (4)

Operation	Symbols	Comments
<u>Product</u>	\times	$A \times B$ $A=\{1,2,3,5\}$; $B=\{0,2\}$; $\{1,2,3, 5\} \times \{0,2\} = \{\{1,0\}, \{2,0\}, \{3,0\}, \{5,0\}, \{1,2\}, \{2,2\}, \{3,2\}, \{5,2\}\}$
<u>Join</u>	\bowtie_{φ}	$B \bowtie_{\varphi} A$; $\varphi: A > B$; $A=\{1,2,3,5\}$; $B=\{0,2\}$; $\{1,2,3,5\} \bowtie_{\varphi} \{0,2\} = \{\{1,0\}, \{2,0\}, \{3,0\}, \{3,2\}, \{5,0\}, \{5,2\}\}$
<u>Division</u>	\div	$A \div B = C$; 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

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

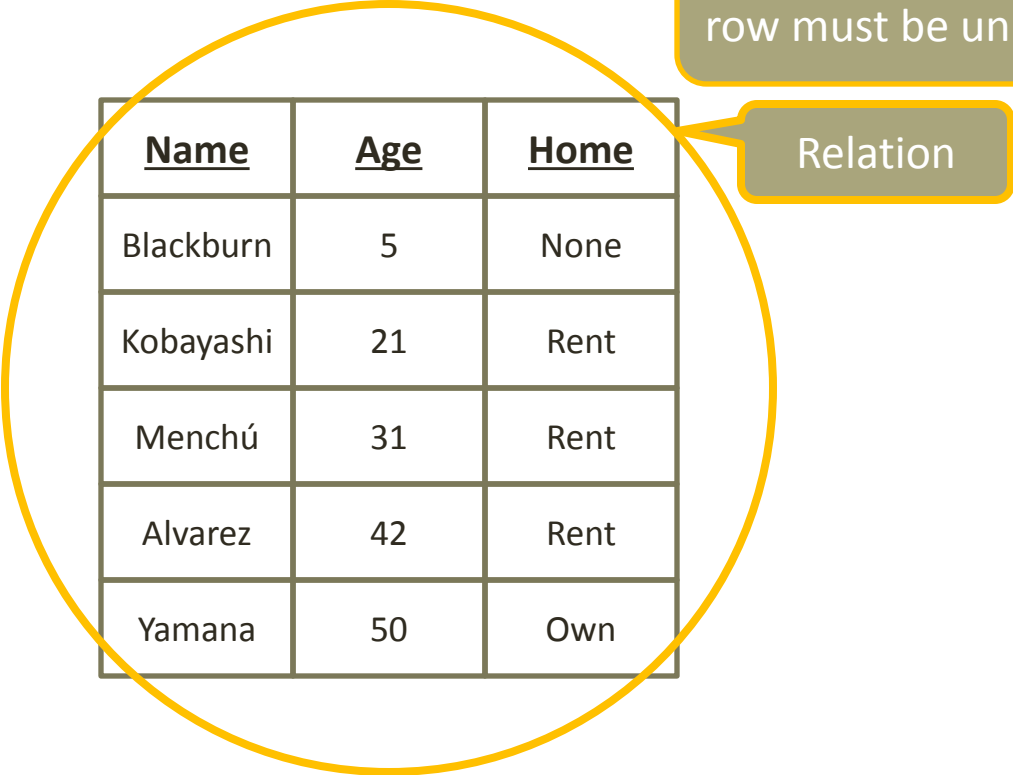
Relational Algebra: Relation



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

Relational Algebra: Relation

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



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

Relation

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

<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: Tuple

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



A diagram illustrating a tuple in a relational database. A yellow rectangular box highlights the row containing 'Kobayashi', '21', and 'Rent'. A yellow arrow points from this box to a separate yellow rounded rectangle labeled 'tuple'.

Relational Algebra: Tuple

tuple from: singlele, doublele, triplele,
quadruple, quinttuple
arity from: unary, binary, ternary

<u>Name</u>	<u>Age</u>	<u>Home</u>
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

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Union

Combine Relations

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:
 $R \cup S$

Relational Algebra: Union

Combine Relations

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

SQL Statement:

```
SELECT * FROM MyTableR UNION  
SELECT * FROM MyTableS
```

Relational Algebra Union:

$R \cup S$

Relational Algebra: Union

Combine Relations

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own



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

Relational Algebra Union:
 $R \cup S$

Relational Algebra: Intersect

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

Same Rows

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

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

Same Rows

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

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

Same Rows

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Intersection:
 $R \cap S$

Relational Algebra: Intersect

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

Same Rows

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

SQL Statement:

```
SELECT * FROM MyTableR  
INTERSECT  
SELECT * FROM MyTableS
```

Relational Algebra Intersection:
 $R \cap S$

Relational Algebra: Intersect

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

Same Rows

<u>Name</u>	<u>Age</u>	<u>Home</u>
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own



<u>Name</u>	<u>Age</u>	<u>Home</u>
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

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

Vertical partition

Relational Algebra: Project

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

Vertical partition

Relational Algebra Project:

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

where

$c1, c2, \dots, cn$: Age, Home

R: MyTable

Relational Algebra: Project

<u>Name</u>	<u>Age</u>	<u>Home</u>
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:

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


where

$c1, c2, \dots, cn$: Age, Home

R: MyTable

Relational Algebra: Project

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



<u>Age</u>	<u>Home</u>
5	None
21	Rent
31	Rent
42	Rent
50	Own

Relational Algebra Project:

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


where

$c1, c2, \dots, cn$: Age, Home

R: MyTable

Relational Algebra: Project

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



<u>Age</u>	<u>Home</u>
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:

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

where

c1, c2, ..., cn: Age, Home

R: MyTable

Relational Algebra: Select

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

Horizontal partition

Relational Algebra: Select

<u>Name</u>	<u>Age</u>	<u>Home</u>
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

SQL Statement:

```
SELECT * FROM MyTable WHERE  
Home = "Rent"
```

<u>Name</u>	<u>Age</u>	<u>Home</u>
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

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



<u>Name</u>	<u>Age</u>	<u>Home</u>
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"

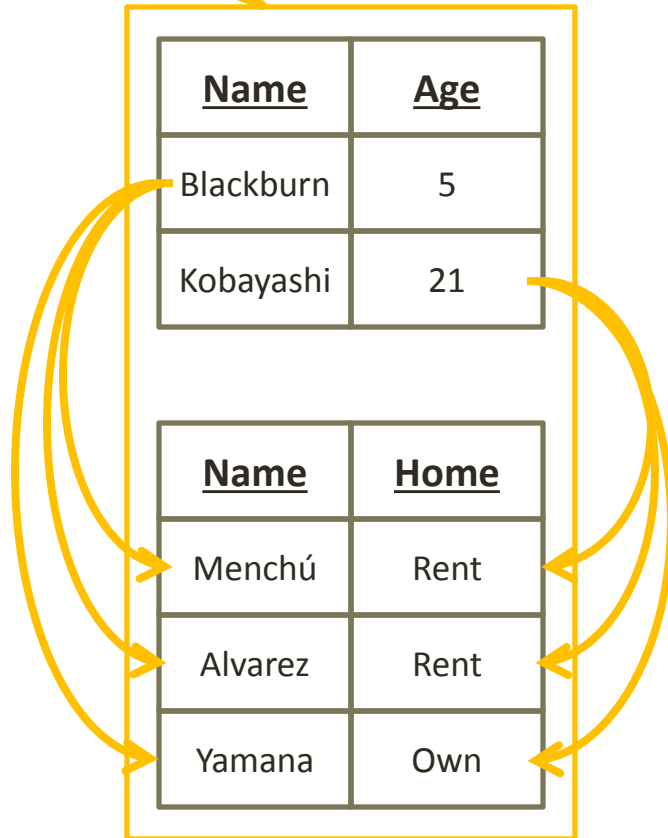
R: MyTable

Relational Algebra: Examples

- $\pi_{\text{Age, Home}}(R)$
 - SELECT Age, Home FROM MyTable
- $\sigma_{\text{Home}=\text{"Rent"}}(R)$
 - SELECT * FROM MyTable WHERE Home = "Rent"
- SELECT Age, Home FROM MyTable WHERE Home = "Rent"
 - $\pi_{\text{Age, Home}}(\sigma_{\text{Home}=\text{"Rent"}}(R))$ or $\sigma_{\text{Home}=\text{"Rent"}}(\pi_{\text{Age, Home}}(R))$
- 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

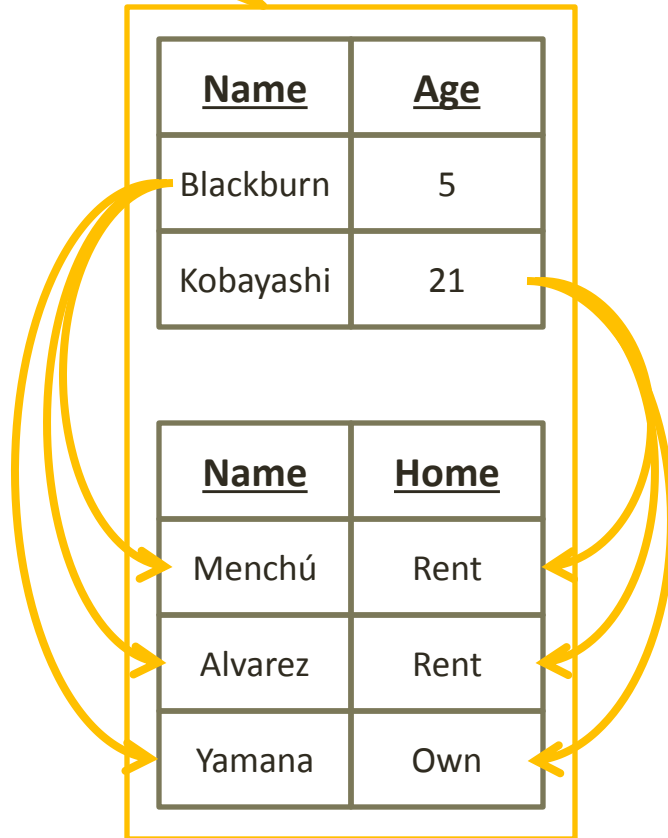
Combine Rows



Relational Algebra Product:
 $R \times S$

Relational Algebra: Product

Combine Rows



SQL Statement:

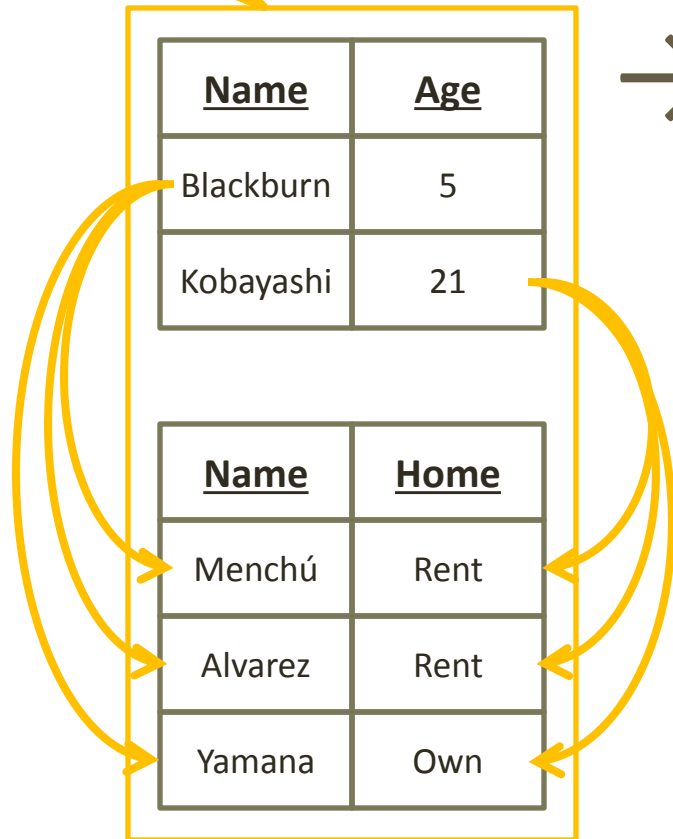
```
SELECT * FROM TableR, TableS
```

Relational Algebra Product:

$R \times S$

Relational Algebra: Product

Combine Rows



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
---------------	------------	---------------	-------------

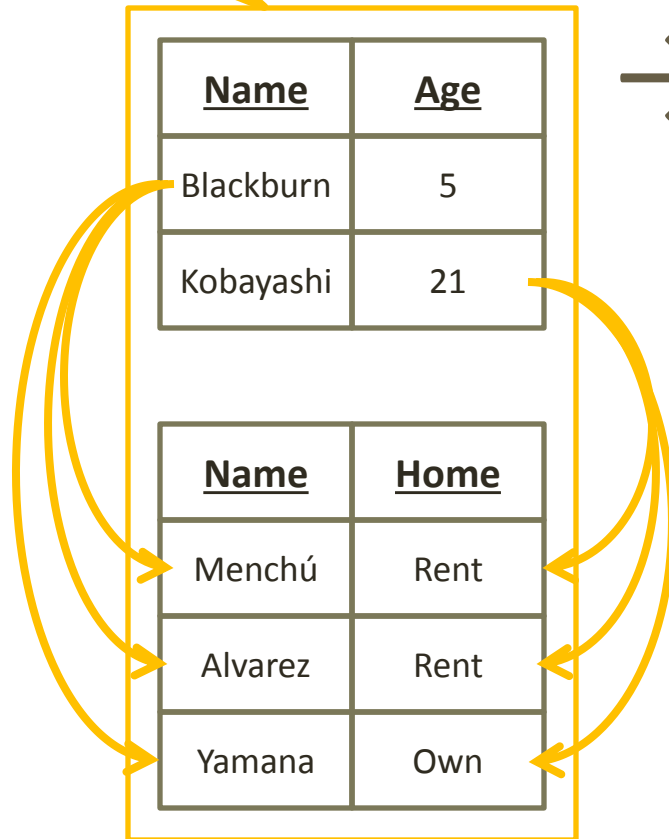
Blackburn	5	Menchú	Rent
		Alvarez	Rent
		Yamana	Own

Kobayashi	21	Menchú	Rent
		Alvarez	Rent
		Yamana	Own

Relational Algebra Product:
 $R \times S$

Relational Algebra: Product

Combine Rows



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
---------------	------------	---------------	-------------

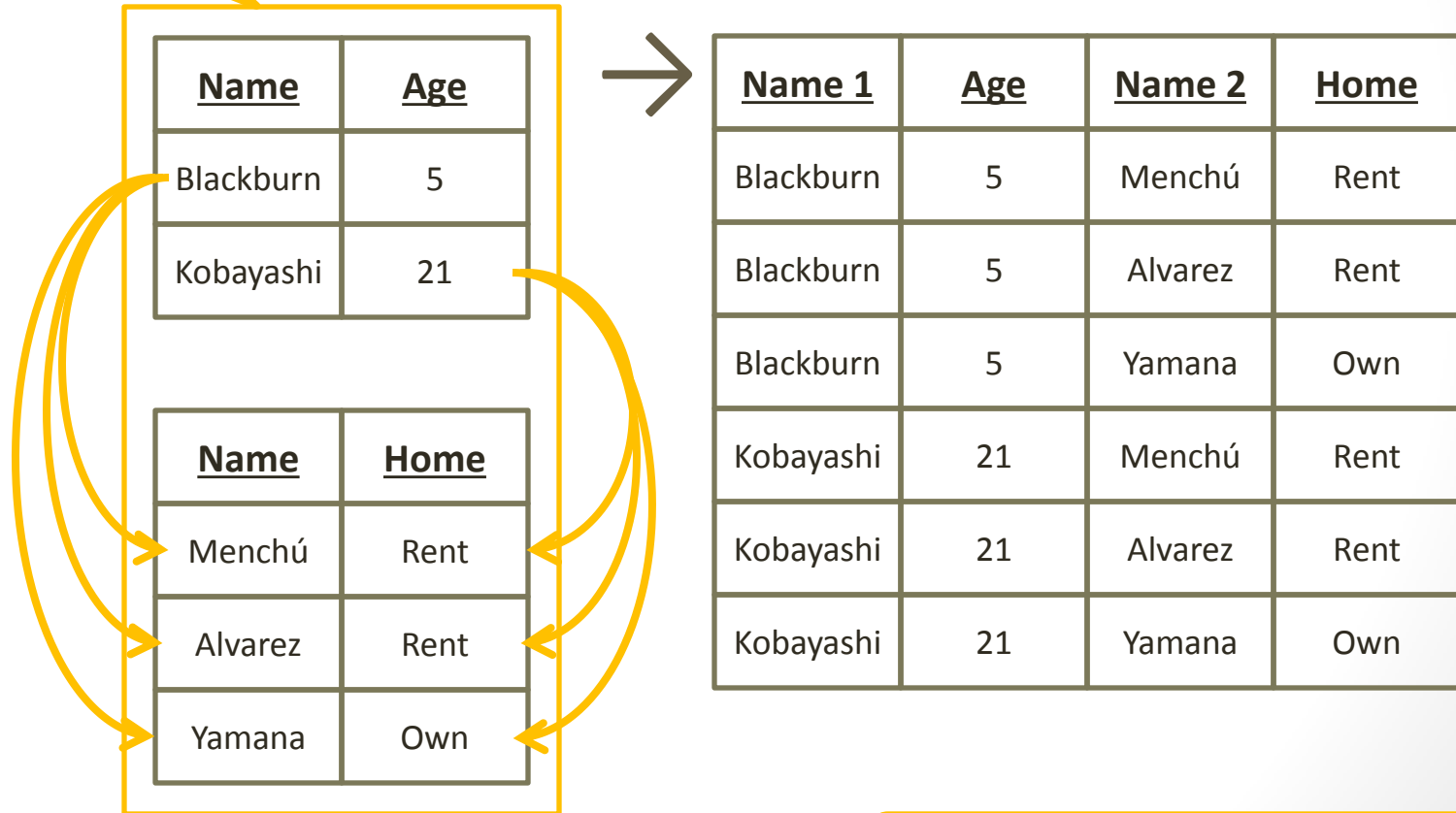
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own

Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
 $R \times S$

Relational Algebra: Product

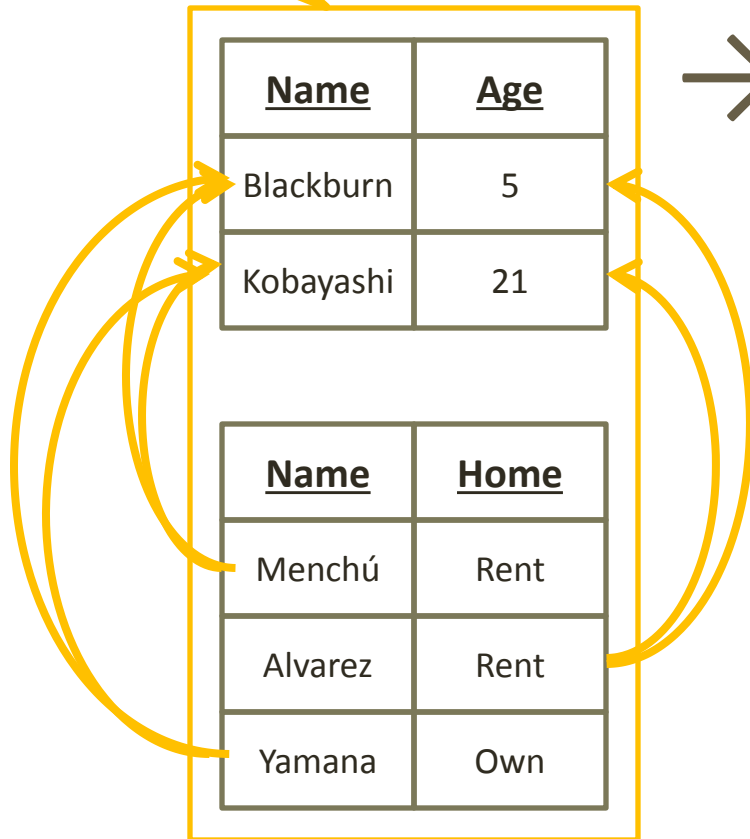
Combine Rows



Relational Algebra Product:
 $R \times S$

Relational Algebra: Product

Combine Rows



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
Blackburn	5	Menchú	Rent
Kobayashi	21		
Blackburn	5	Alvarez	Rent
Kobayashi	21		
Blackburn	5	Yamana	Own
Kobayashi	21		

Relational Algebra Product:
 $R \times S$

Relational Algebra: Product

Combine Rows

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
---------------	------------	---------------	-------------

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:
 $R \times S$

Relational Algebra: Product

Combine Rows

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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:
 $R \times S$

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

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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:
 $R \times S$

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

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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:
 $R \times S$

Relational Algebra: Join

Combine Rows

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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 $\varphi: \text{Home} = \text{"Rent"}$
Relational Algebra Join:
 $R \bowtie_{\varphi} S$ where $\varphi: \text{Home} = \text{"Rent"}$

Relational Algebra: Join

Combine Rows

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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 $\varphi: \text{Home} = \text{"Rent"}$
Relational Algebra Join:
 $R \bowtie_{\varphi} S$ where $\varphi: \text{Home} = \text{"Rent"}$

Relational Algebra: Join

Combine Rows

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21

<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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 $\varphi: \text{Home} = \text{"Rent"}$
Relational Algebra Join:
 $R \bowtie_{\varphi} S$ where $\varphi: \text{Home} = \text{"Rent"}$

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 $\varphi: \text{Home} = \text{"Rent"}$
- JOIN
 - `SELECT * FROM TableR JOIN TableS ON Home = "Rent"`
 - $R \bowtie_{\varphi} S$ where $\varphi: \text{Home} = \text{"Rent"}$

Relational Algebra: Division

A Division is sort of like the reverse of a Product

This was a Product
Operand

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21



<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own

This was a Product Operand

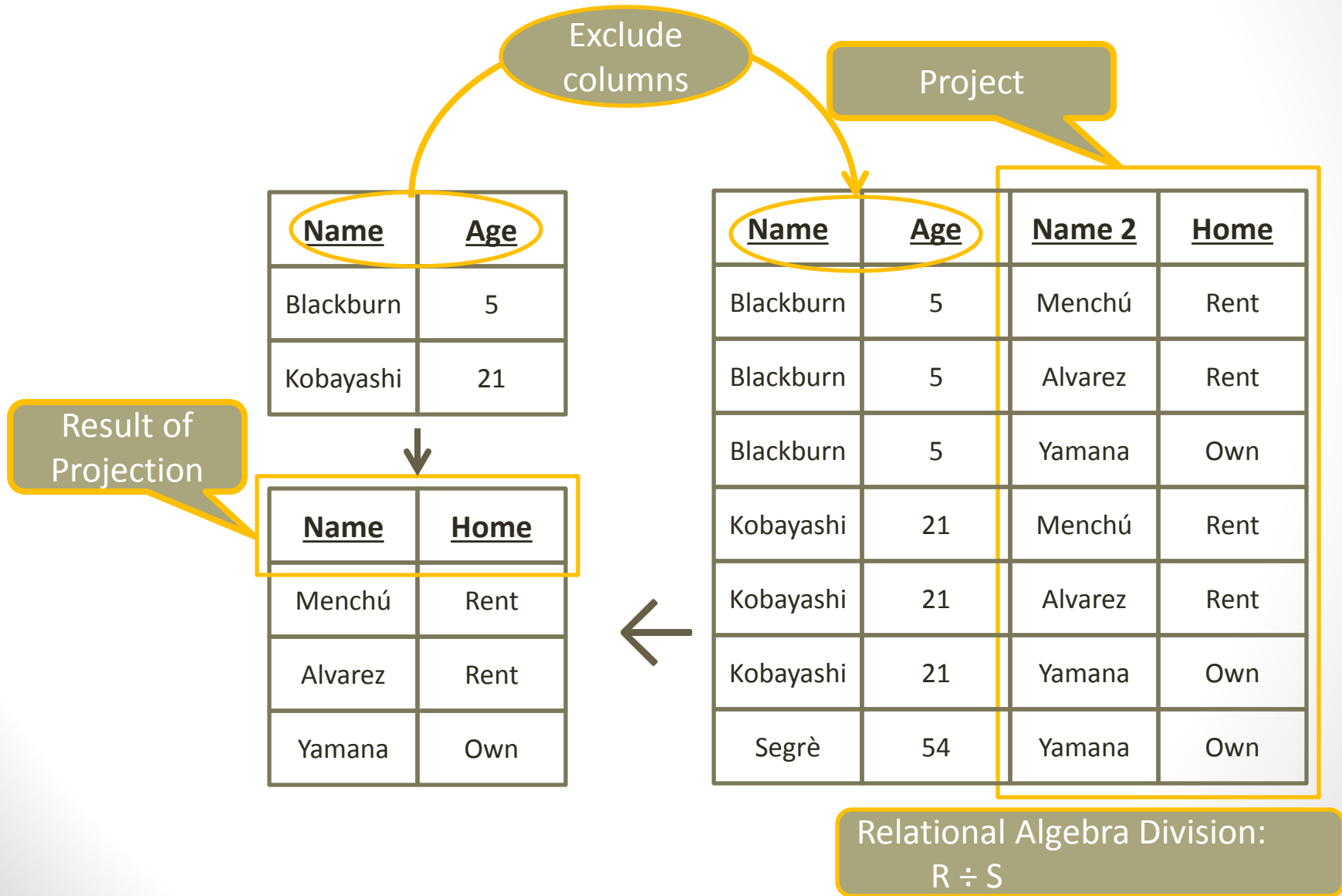
This was the result
of a Product

<u>Name 1</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

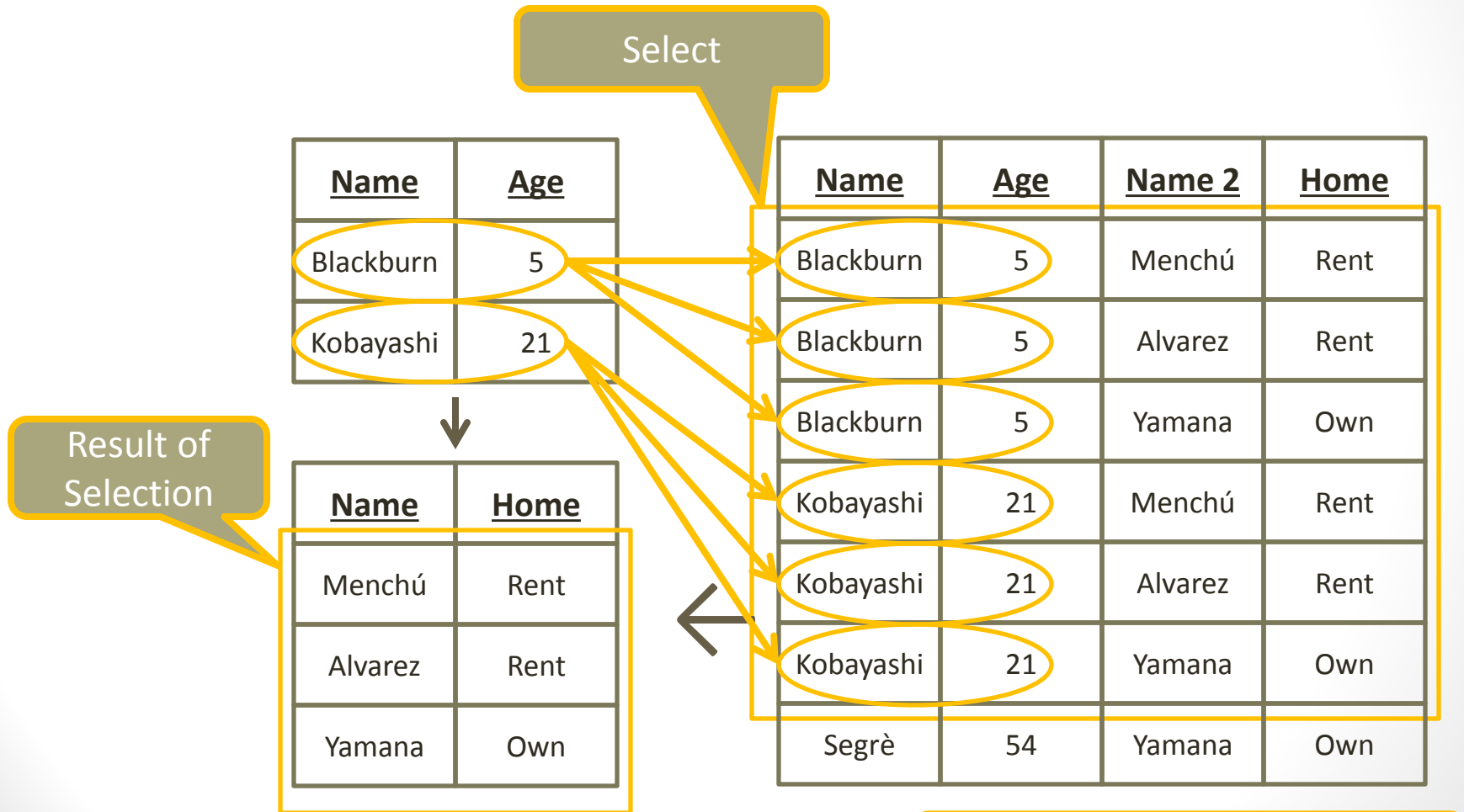


Relational Algebra Division:
 $R \div S$

Relational Algebra: Division



Relational Algebra: Division



Relational Algebra Division:
 $R \div S$

Relational Algebra: Division

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21



<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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 l where the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21



<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own



<u>Name</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
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.

<u>Name</u>	<u>Age</u>
Blackburn	5
Kobayashi	21



<u>Name</u>	<u>Home</u>
Menchú	Rent
Alvarez	Rent
Yamana	Own

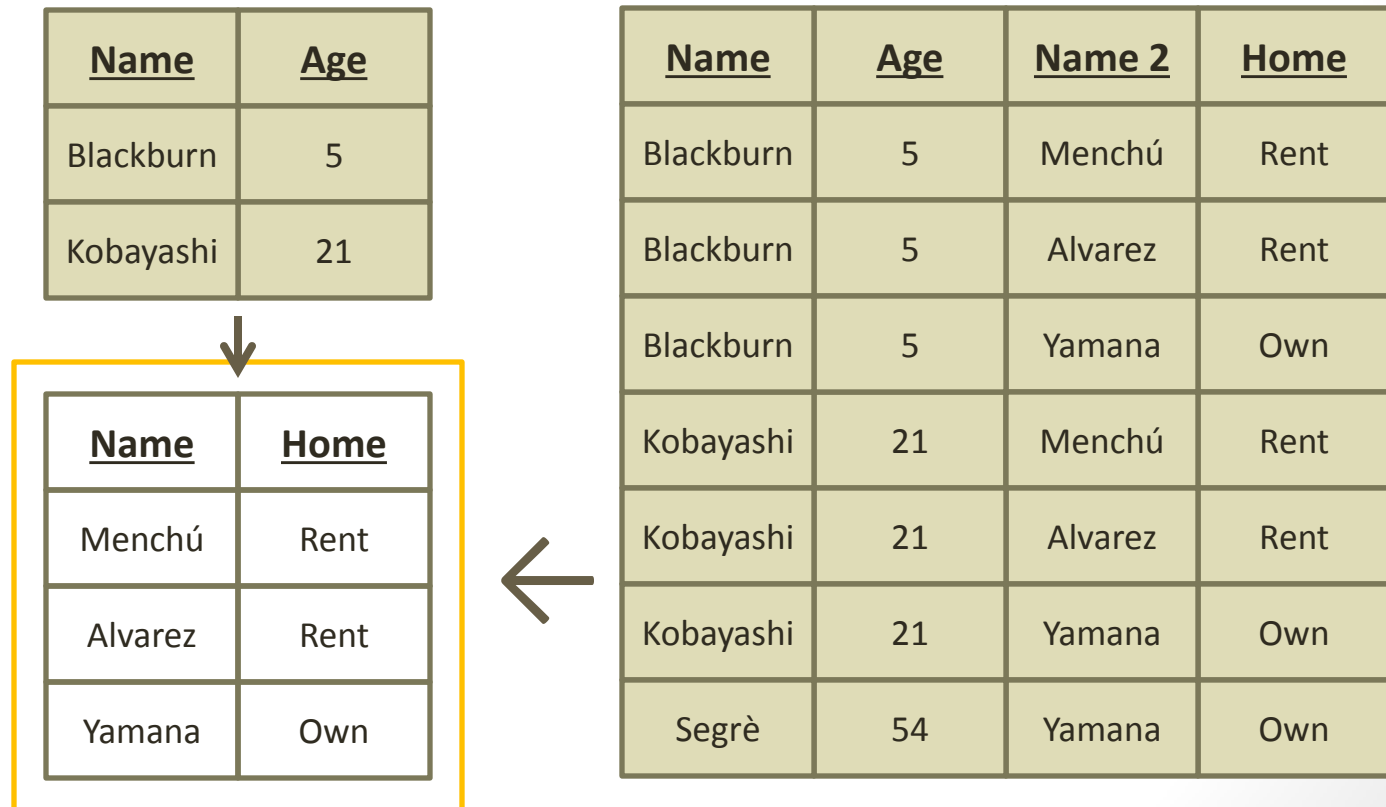


<u>Name</u>	<u>Age</u>	<u>Name 2</u>	<u>Home</u>
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
 $R \div S$

Relational Algebra: Division

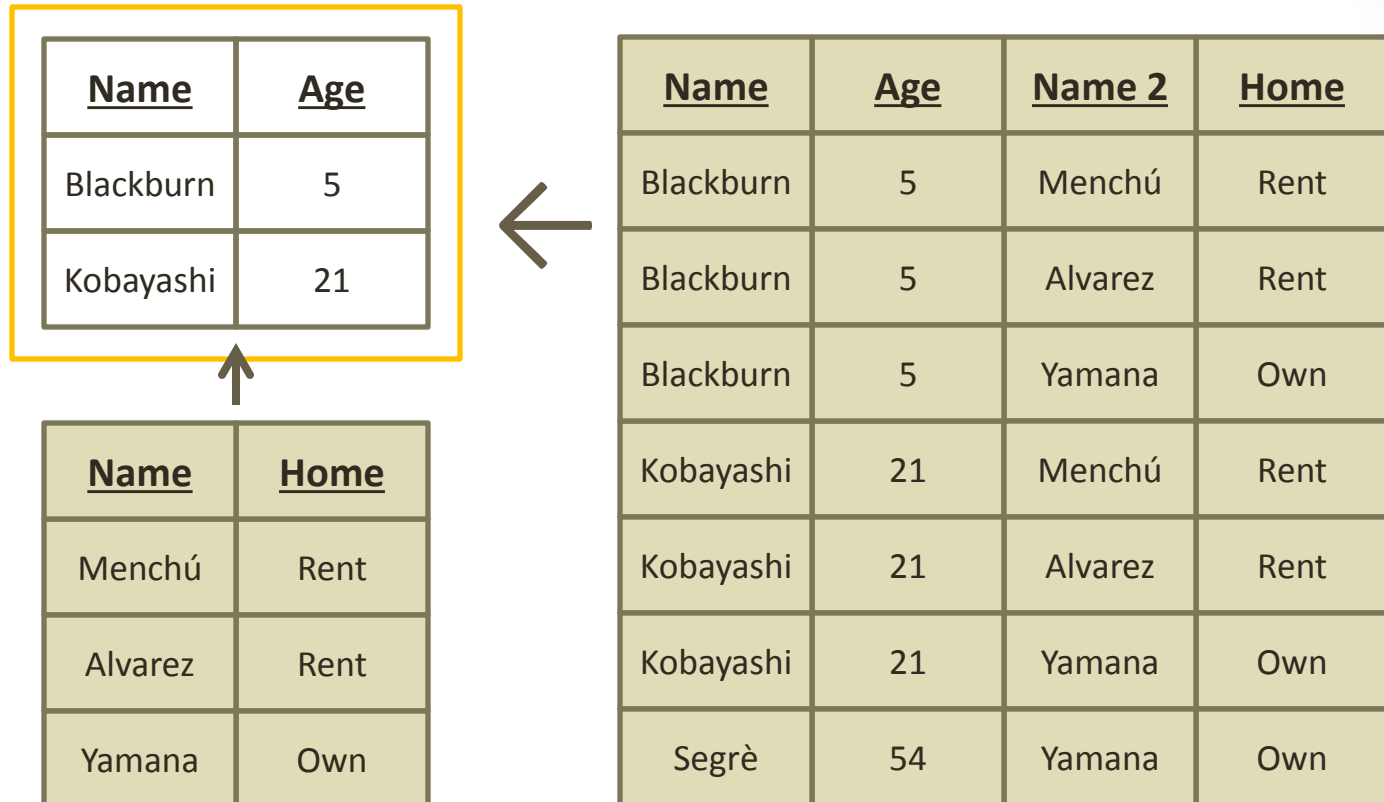
The result of a division is a relation with n tuples of arity i where the dividend operand has $n*m$ tuples of arity $i + j$ and the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.



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.



Relational Algebra Division:
 $R \div S$

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_\(mathematics\)](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. $\pi_{c1, c2}(\sigma_{\varphi1}(\sigma_{\varphi2}(\pi_{c1, c2, c3, c5}(R))))$
Where
 - $\varphi1: C1 = C5;$
 - $\varphi2: C5 = \text{"Test"};$
 - $R: \text{MyTable};$
 - 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. $\pi_{S.C1, R.C2}(\sigma_{\varphi1}(R) \bowtie_{\varphi2} S)$
 - where
 - $\varphi1 = (R.C2 = 'A')$
 - $\varphi2 = (R.C1 = S.C2)$
 - 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.