

# Álgebra Relacional



Universidad Autónoma  
de Entre Ríos

---

# Álgebra Relacional

- Es un conjunto de operaciones definidas sobre relaciones.
- Estos operadores simples, combinados entre sí, permiten manipular las relaciones de una manera sencilla y útil.
- Cada operador representa una función que a partir de una o más relaciones obtiene otra relación.

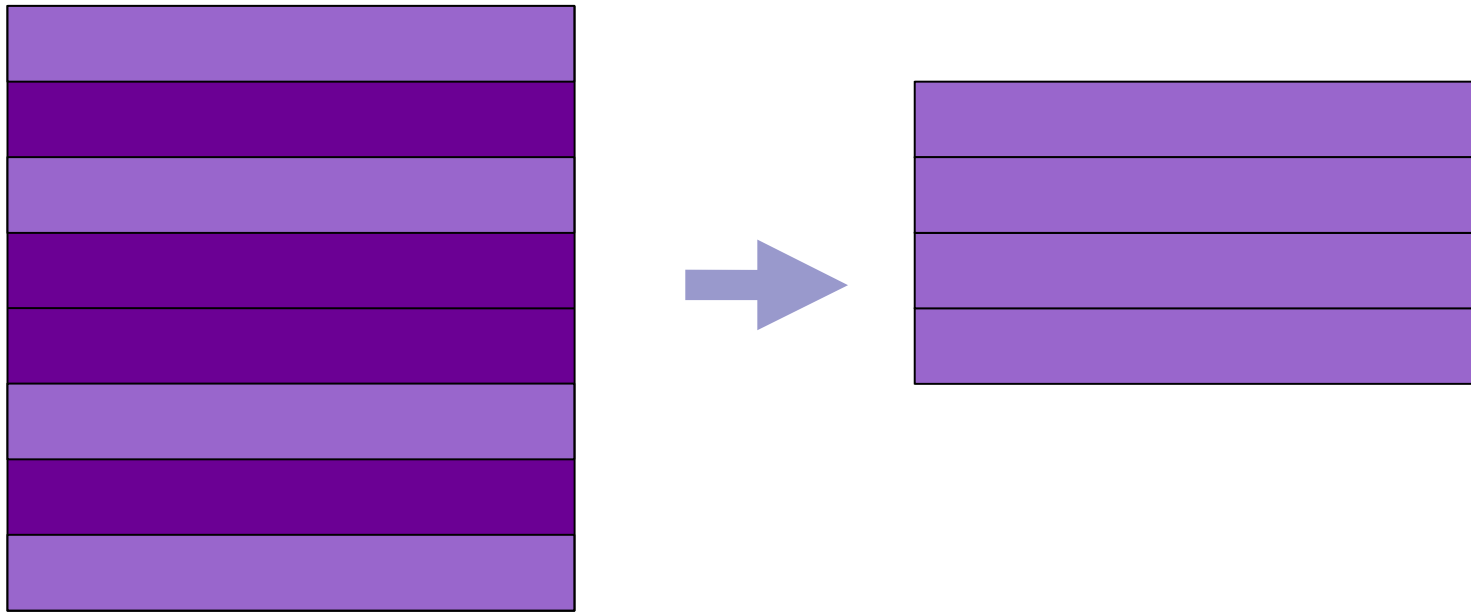
# Álgebra Relacional

**Las expresiones que el álgebra relacional permite construir constituyen una representación simbólica de alto nivel de aquella porción de la base datos con la que el usuario pretende operar.**

# Operaciones del Álgebra Relacional

# Operaciones del álgebra relacional

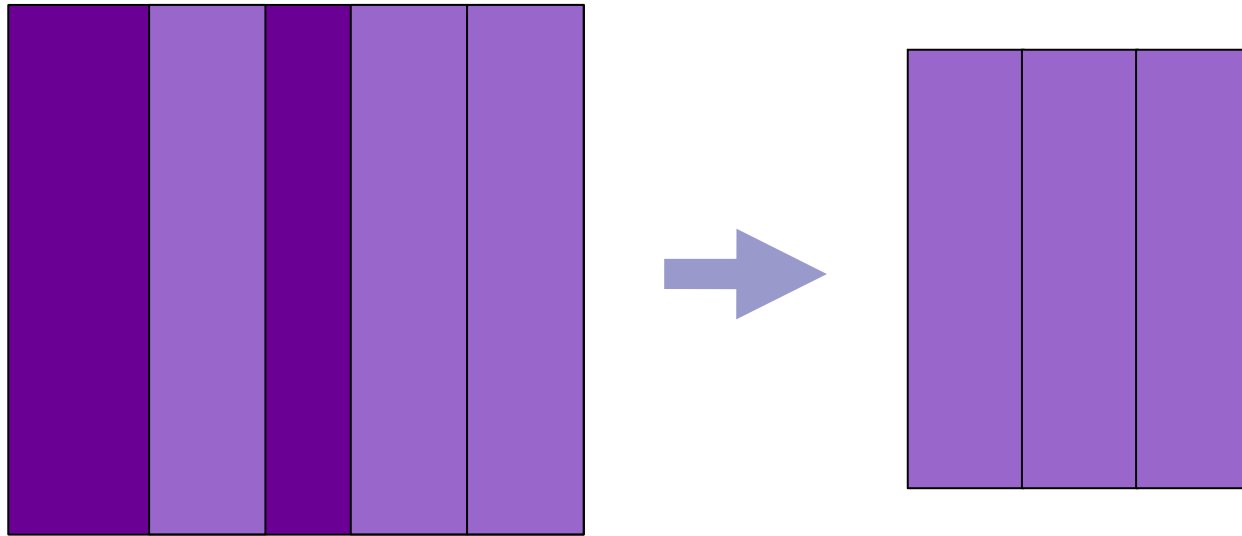
## Restricción



Devuelve una relación conteniendo todas las tuplas de una determinada relación que cumplen con una condición dada.

# Operaciones del álgebra relacional

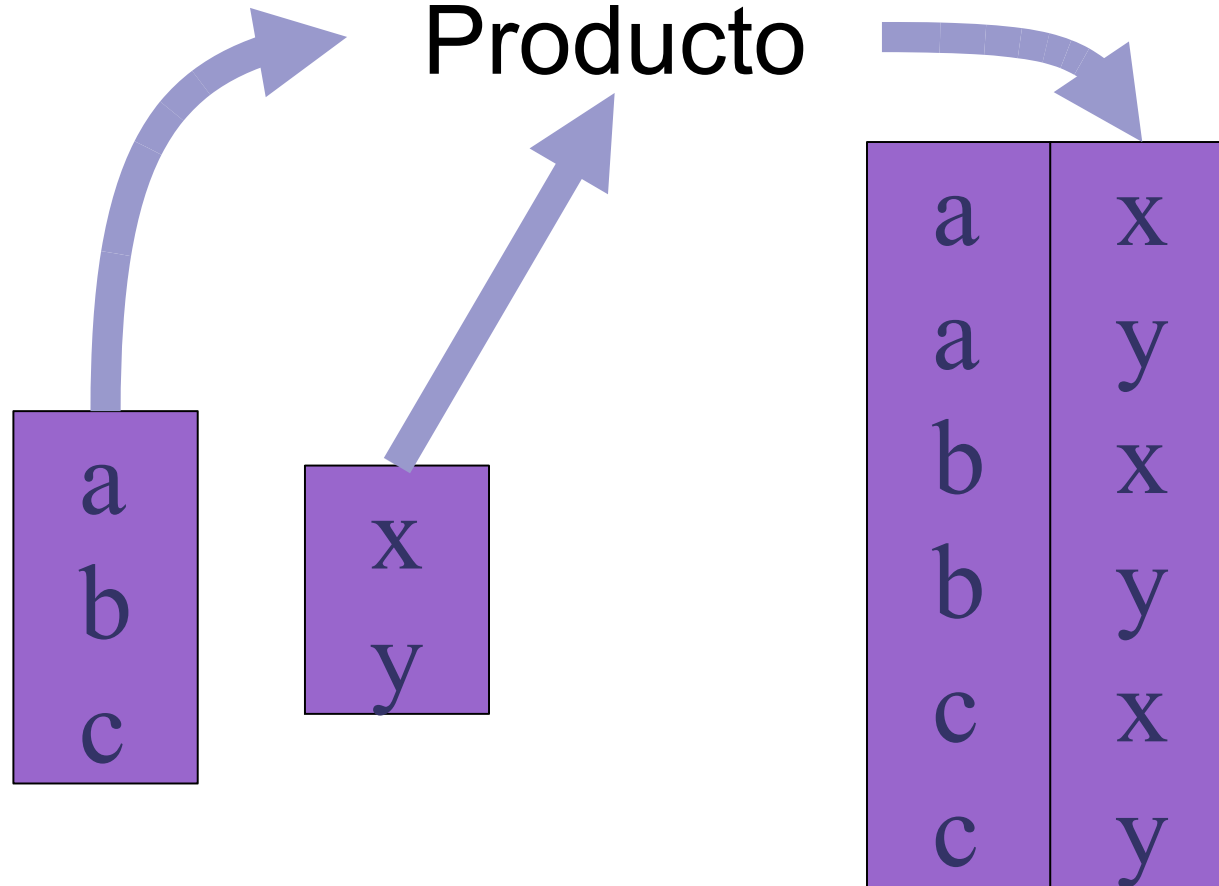
## Proyección



Devuelve una relación conteniendo todas las (sub)tuplas que quedan en una determinada relación luego de quitar determinados atributos

# Operaciones del álgebra relacional

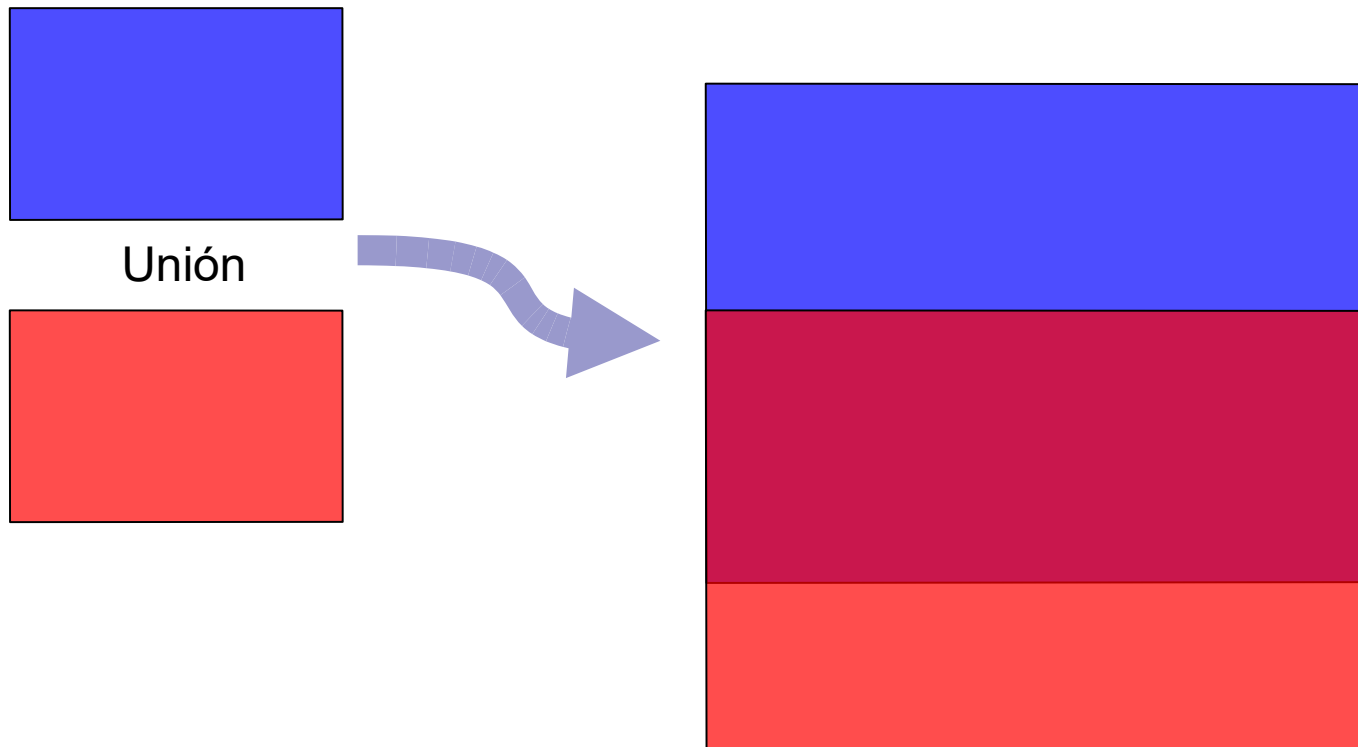
## Producto



Devuelve una relación que contiene todas las tuplas posibles que resultan de la combinación de dos tuplas cualquiera provenientes de sendas relaciones.

# Operaciones del álgebra relacional

## Unión

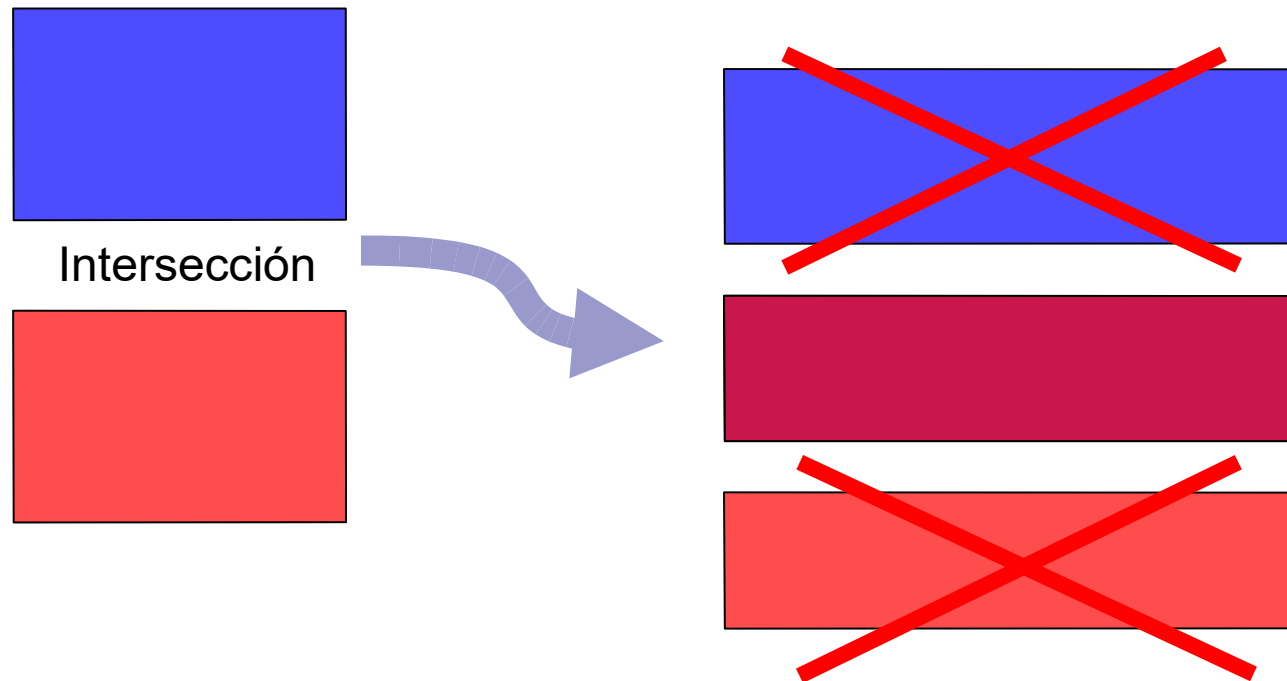


Devuelve una relación que contiene todas las tuplas que aparecen en cualquiera de dos relaciones dadas.



# Operaciones del álgebra relacional

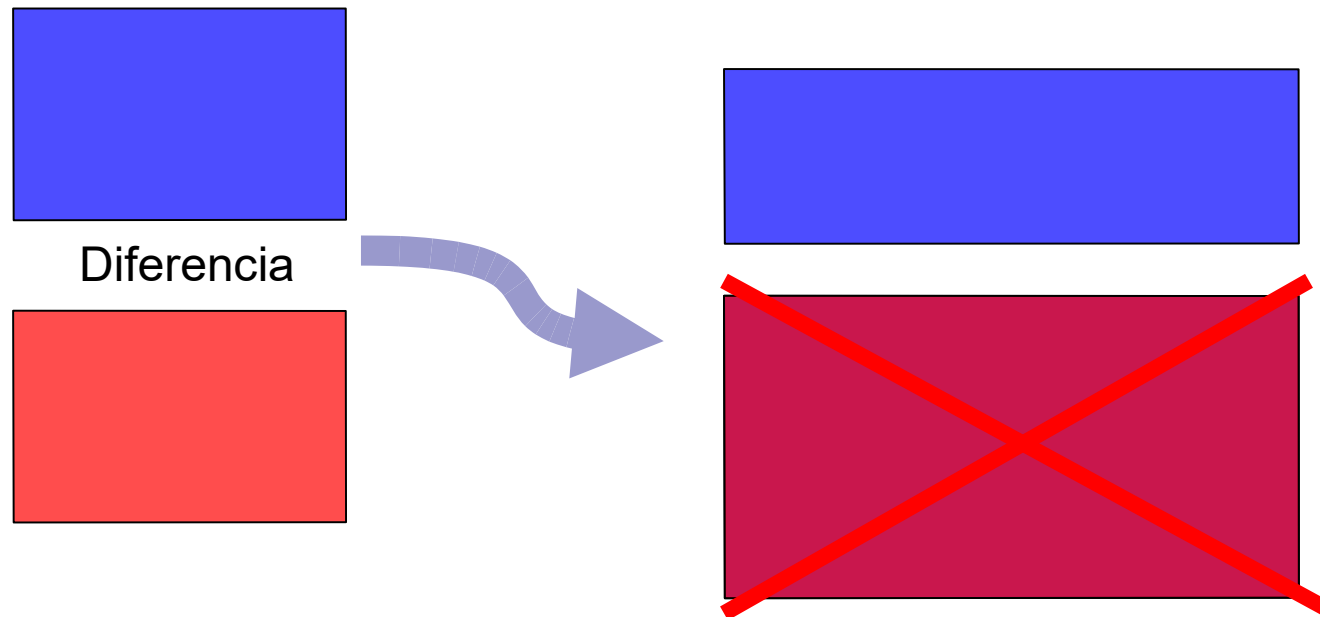
## Intersección



Devuelve una relación que contiene todas las tuplas que aparecen en ambas relaciones especificadas.

# Operaciones del álgebra relacional

## Diferencia



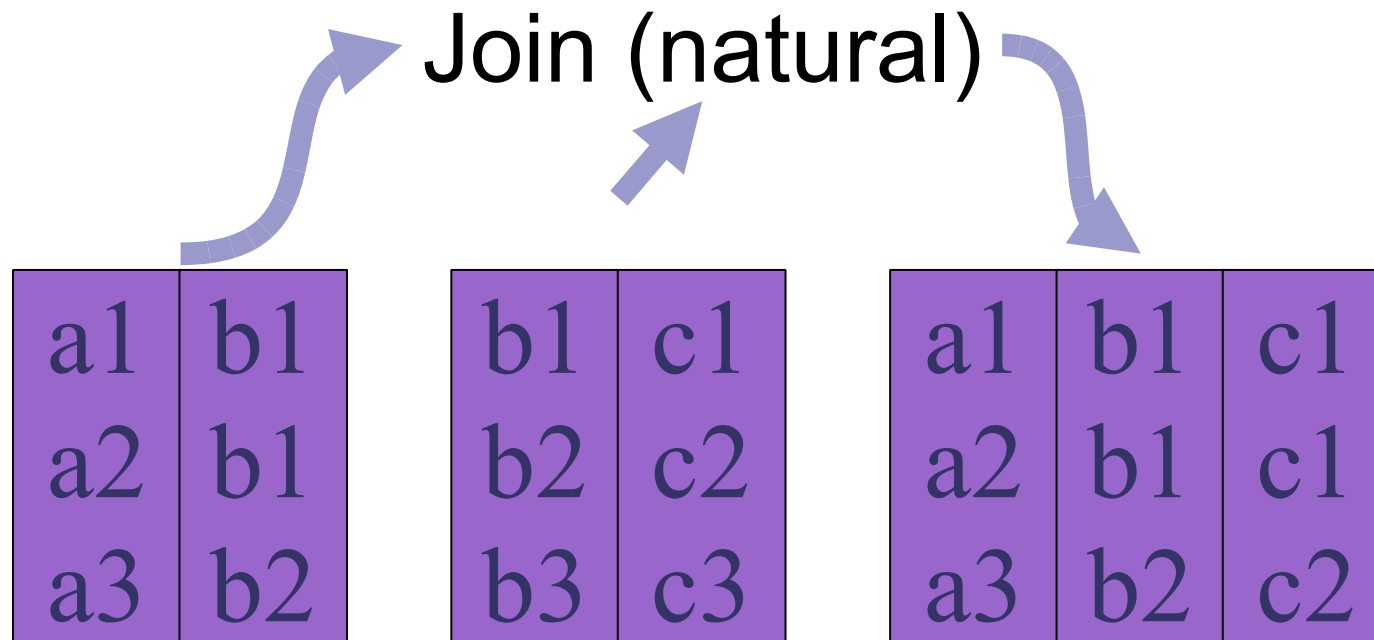
Devuelve una relación con todas las tuplas que aparecen en la primera, pero no en la segunda de dos relaciones dadas.

# Operaciones del álgebra relacional

No se define el operador

**complemento**

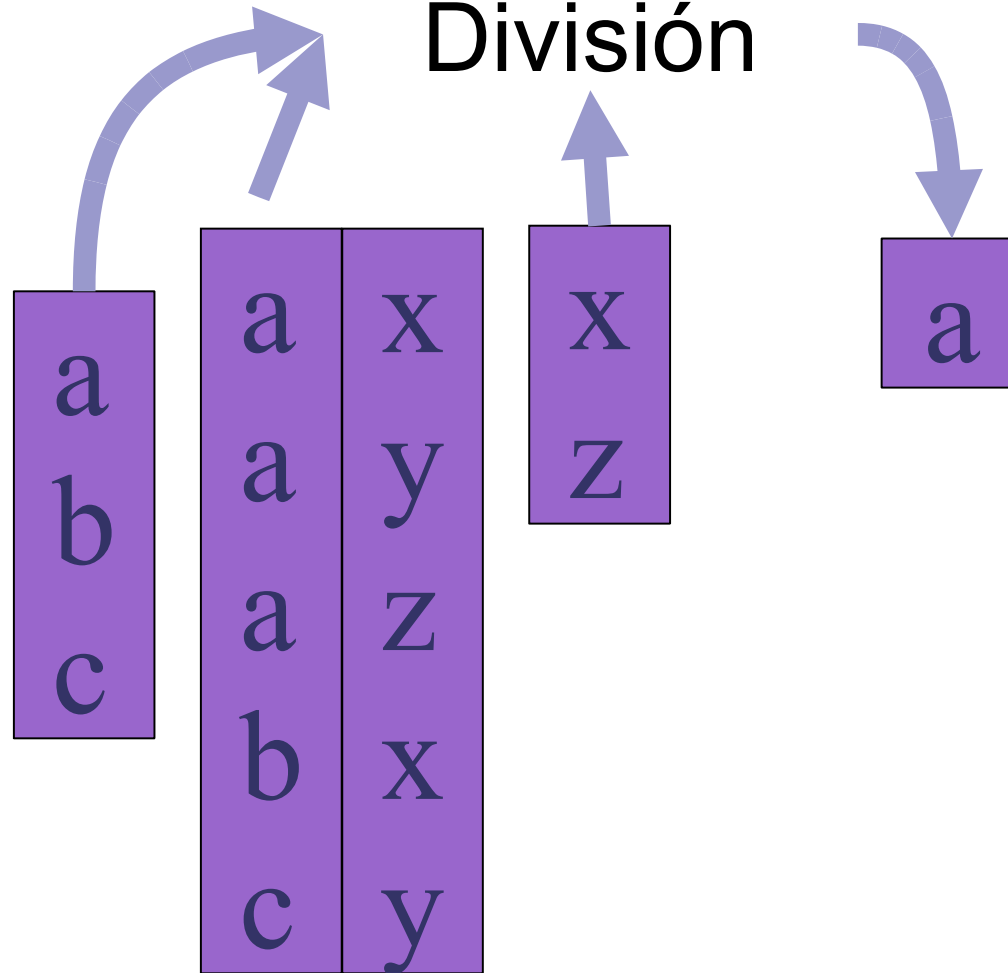
# Operaciones del álgebra relacional



Devuelve una relación que contiene todas las tuplas posibles que son combinación de dos tuplas provenientes de sendas relaciones, tal que las tuplas que contribuyen a una combinación dada tienen valores comunes para los atributos en común, y esos valores comunes aparecen una vez (y no dos) en las tuplas del resultado.

# Operaciones del álgebra

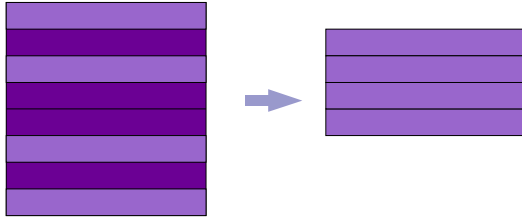
## relacional División



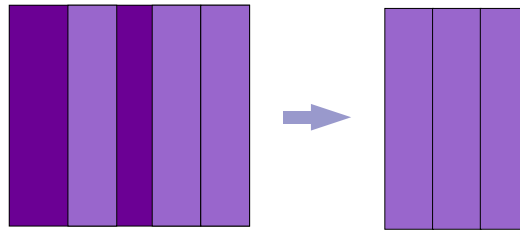
A partir de dos relaciones unarias y una binaria, devuelve una relación que contiene todas las tuplas de una relación unaria que aparecen en la relación binaria coincidiendo con todas las tuplas de la otra relación unaria.

# Operaciones del álgebra relacional

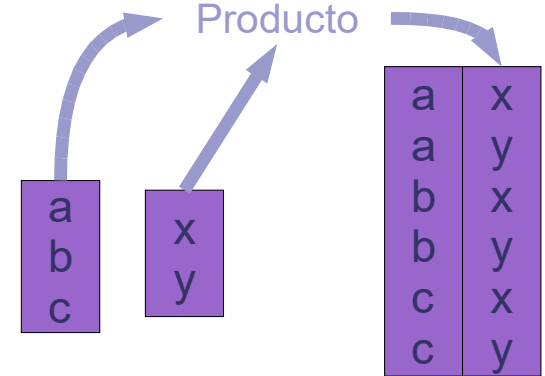
Restricción



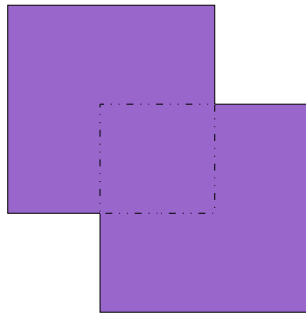
Proyección



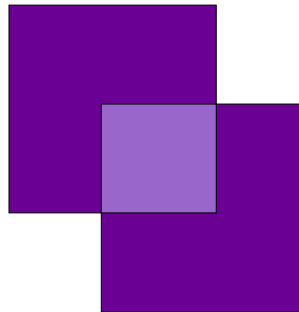
Producto



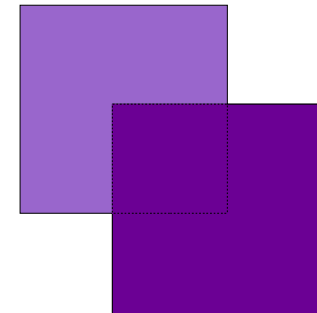
Unión



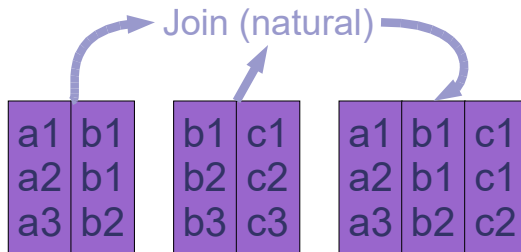
Intersección



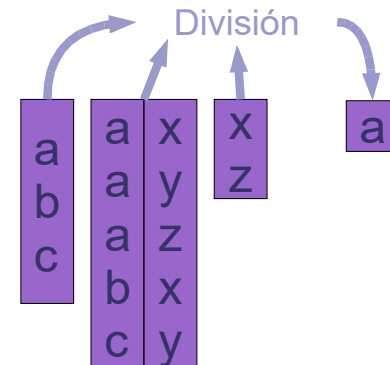
Diferencia



Join (natural)



División

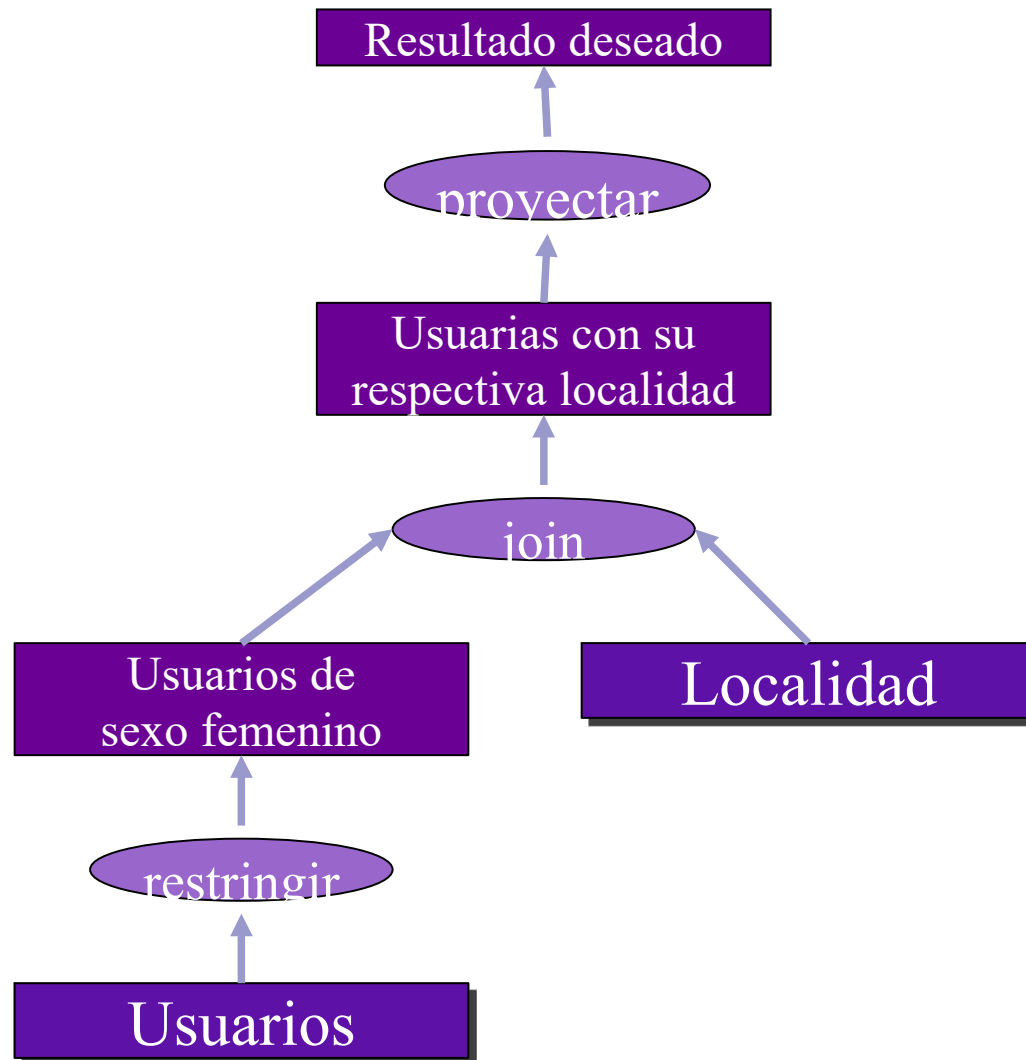


# Propiedad de Clausura

- El resultado de una operación es una relación
- Se puede utilizar el resultado de una operación en otra.

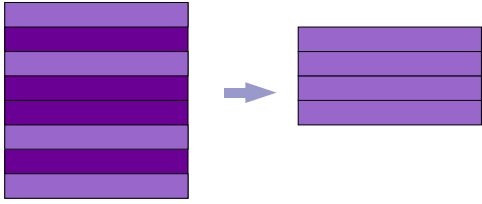
# Propiedad de Clausura

- Ejemplo: Hallar el nombre de las localidades donde viven usuarias de la biblioteca.





# Las operaciones del Álgebra Relacional **en detalle**



## Restricción ( $\sigma$ )

- **Sintaxis WinRDBI:**

`select <condición> (<expresión>)`

`donde`

`<condición>`

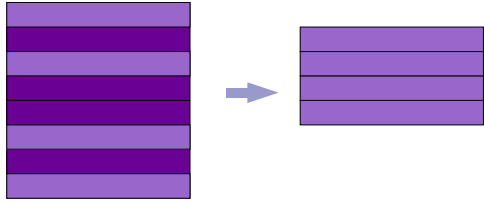
Es una expresión booleana que puede involucrar atributos de una tupla, constantes, operadores de comparación (<, >, =) y operadores lógicos (and, or)

`<expresión>`

Es cualquier expresión que denote una relación.

- **Semántica:**

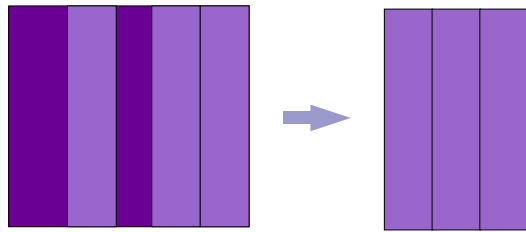
Sólo las tuplas para las cuales se cumple la condición se conservarán en la relación resultante.



## Restricción ( $\sigma$ )

- **Ejemplo:**  
`select cp='3260' (usuario);`
- **Observaciones**  
La expresión de comparación siempre opera con los atributos de una tupla o con valores constantes. No es posible realizar comparaciones de una tupla con otra.
- **Esquema resultante**  
Idéntico al esquema de la relación original.
- **Sintaxis original**

$\sigma_{\langle \text{condición} \rangle} (\langle \text{expresión} \rangle)$



## Proyección ( $\Pi$ )

- **Sintaxis WinRDBI:**

`project <lista> (<expresión>)`

**donde**

`<lista>`

Es una lista de uno o más atributos de la relación a proyectar, separando un elemento de otro con una coma.

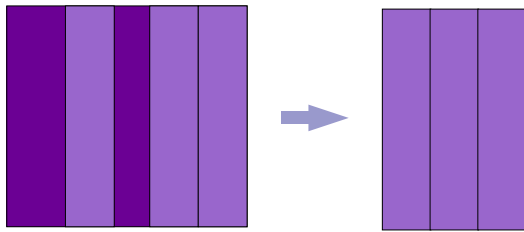
`<expresión>`

Es cualquier expresión que denote una relación.

- **Semántica:**

Obtiene un subconjunto “vertical” de una relación dada, es decir, otra relación en la que se han eliminado aquellos atributos no mencionados en la lista.

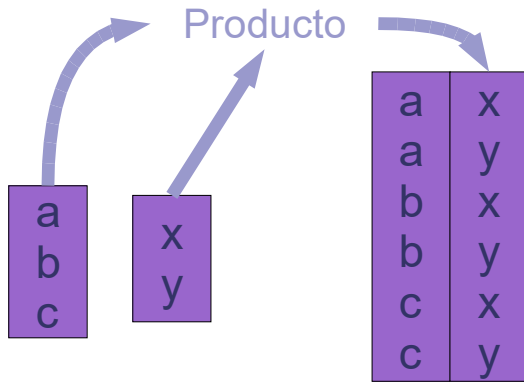
Proyección



## Proyección ( $\Pi$ )

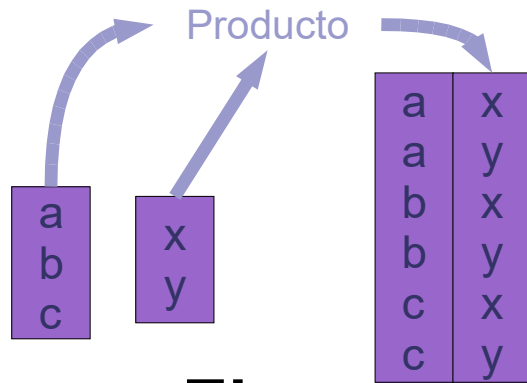
- **Ejemplo:**  
`project cp (usuario);`
- **Observaciones**  
Las tuplas duplicadas que pudieran resultar son eliminadas .
- **Esquema resultante**  
Conserva sólo los atributos especificados.
- **Sintaxis original**

$\Pi_{\langle \text{lista} \rangle} (\langle \text{expresión} \rangle)$



# Producto ( $\times$ )

- **Sintaxis WinRDBI:**  
(`<expresión>`) product (`<expresión>`)  
donde  
`<expresión>`  
denota cualquier relación.
- **Semántica:**  
Obtiene la combinación de todas las tuplas de una relación con todas las tuplas de la otra.
- **Restricciones**  
Las relaciones no pueden poseer atributos en común.

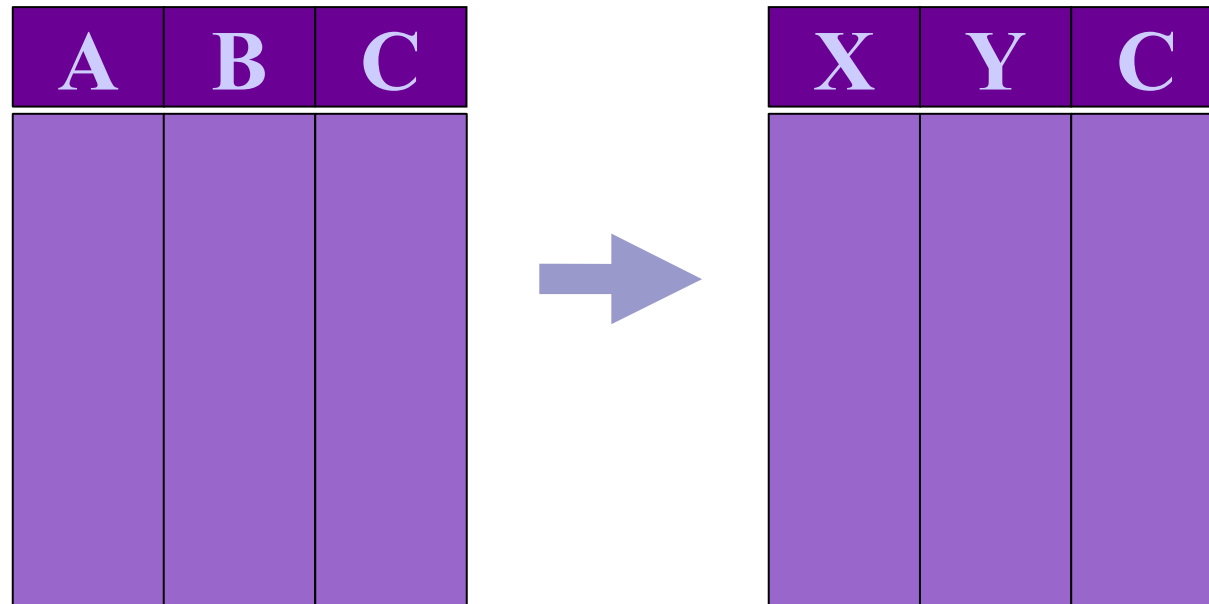


# Producto ( $\times$ )

- **Ejemplo:**  
`(libro) product (prestamo);`
- **Observaciones**  
También se lo denomina “producto cartesiano”.
- **Esquema resultante**  
La unión de los esquemas de las dos relaciones.
- **Sintaxis original**  
`<expresión>  $\times$  <expresión>`

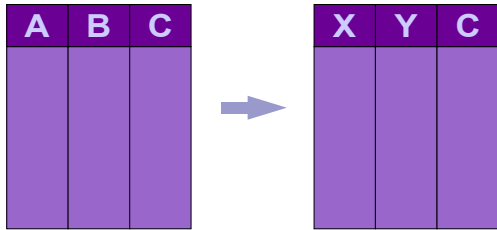
# Operaciones del álgebra relacional

## Renombrar



Operación especial que dada una relación obtiene otra que difiere solamente en el nombre de uno o más de sus atributos.

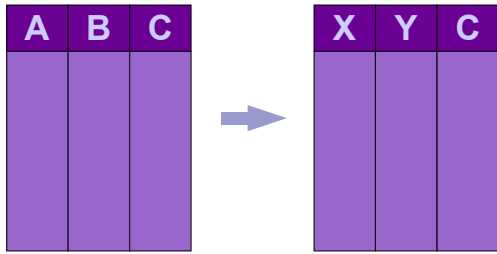




## Renombrar ( $\rho$ )

- **Sintaxis WinRDBI:** WinRDBI no implementa este operador. Aún así, es posible suplirlo al renombrar los atributos en una asignación:  
`<relvar>(<lista>) := <expr>;`  
donde  
`<relvar>` es el nombre de una variable relacional.  
`<lista>` es una lista de atributos, separados por comas  
`<expr>` denota una relación.
- **Semántica:** Se asigna a la variable el valor resultante de evaluar la expresión, pero forzando el esquema indicado por la lista de atributos.
- **Restricciones:** la lista de atributos debe contener una cantidad de elementos igual al grado de la relación.

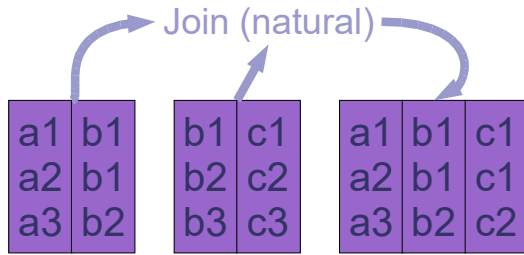
Renombar



## Renombrar ( $\rho$ )

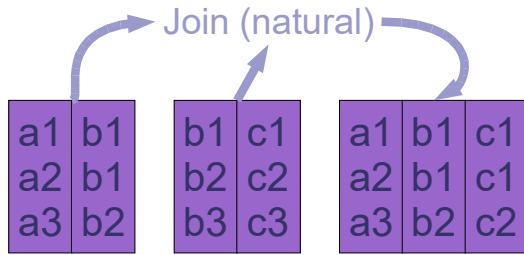
- **Ejemplo:**  
`doc(id):=project dni (usuario);`
- **Esquema resultante**  
Aquel designado por la lista de atributos que se desea obtener.
- **Observaciones**  
Este operador muchas veces resulta imprescindible ya que ciertas operaciones requieren nombres específicos de atributos
- **Sintaxis original**

$\rho_{\langle \text{lista} \rangle}(\langle \text{expr} \rangle)$



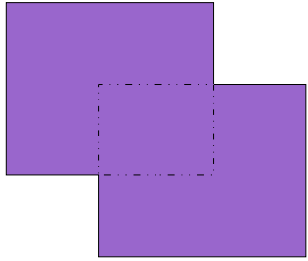
## Join natural ( $|><|$ )

- **Sintaxis WinRDBI:**  
`(<expresión>) njoin (<expresión>)`  
donde  
`<expresión>`  
denota cualquier relación.
- **Semántica:**  
Devuelve la relación resultante de aplicar el producto a dos relaciones y al resultado aplicarle una restricción que devuelva aquellas tuplas con igual valor para atributos de igual nombre.



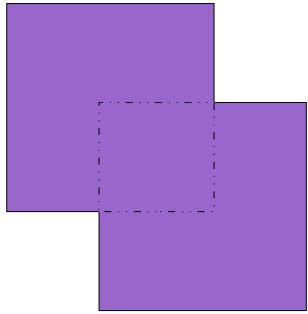
## Join natural ( $|><|$ )

- **Ejemplo:**  
`(libro) njoin (prestamo);`
- **Observaciones**
- Es la operación más usada.
- Equivale a un producto seguido de una restricción y una proyección.
- Si se aplica a relaciones sin ningún atributo en común, equivale a un producto cartesiano.
- **Esquema resultante**  
 La unión de los esquemas de las dos relaciones.
- **Sintaxis original**  
`<expresión> |><| <expresión>`



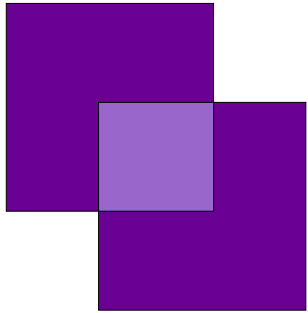
# Unión ( $\cup$ )

- **Sintaxis WinRDBI:**  
`(<expresión>) union (<expresión>)`  
donde  
`<expresión>`  
denota cualquier relación.
- **Semántica:**  
Obtiene las tuplas presentes en una relación, en la otra, o en ambas.
- **Restricciones**  
Los esquemas de las relaciones deben ser compatibles: deben poseer igual cantidad y nombres de atributos, y los atributos correspondientes deben estar definidos sobre dominios compatibles..



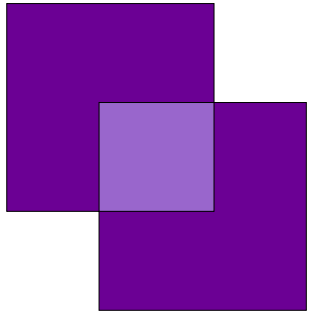
# Unión ( $\cup$ )

- **Ejemplo:**  
`(project dni (alumno))  
union  
(project dni (docente));`
- **Esquema resultante**  
**Mismo esquema que las relaciones originales.**
- **Sintaxis original**  
**`<expresión>  $\cup$  <expresión>`**



# Intersección ( $\cap$ )

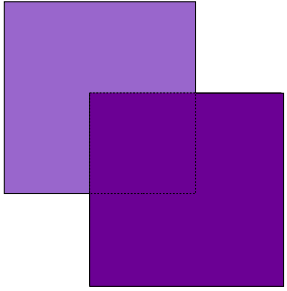
- **Sintaxis WinRDBI:**  
`(<expresión>) intersect (<expresión>)`  
donde  
`<expresión>`  
denota cualquier relación.
- **Semántica:**  
Obtiene las tuplas presentes en ambas relaciones.
- **Restricciones**  
Los esquemas de las relaciones deben ser compatibles: deben poseer igual cantidad y nombres de atributos, y los atributos correspondientes deben estar definidos sobre dominios compatibles..



# Intersección ( $\cap$ )

- **Ejemplo:**  
`(project dni (alumno))  
intersect  
(project dni (docente));`
- **Esquema resultante**  
**Mismo esquema que las relaciones originales.**
- **Sintaxis original**  
 **$\langle \text{expresión} \rangle \cap \langle \text{expresión} \rangle$**



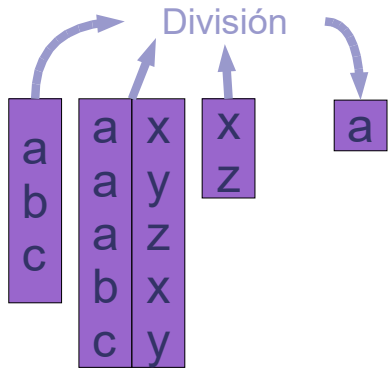


# Diferencia (-)

- **Sintaxis WinRDBI:**  
`(<expresión>) difference (<expresión>)`  
donde  
`<expresión>`  
denota cualquier relación.
- **Semántica:**  
Obtiene las tuplas presentes en la primera relación pero no en la segunda.
- **Restricciones**  
Los esquemas de las relaciones deben ser compatibles: deben poseer igual cantidad y nombres de atributos, y los atributos correspondientes deben estar definidos sobre dominios compatibles..

# Diferencia (-)

- **Ejemplo:**  
`(project dni (alumno))  
difference  
(project dni (docente)));`
- **Esquema resultante**  
**Mismo esquema que las relaciones originales.**
- **Sintaxis original**  
**<expresión> - <expresión>**



# División

- WinRDBI no incluye un operador para división

- Date lo define como

$\langle \text{expr}^1 \rangle \text{ DIVIDEBY } \langle \text{expr}^2 \rangle \text{ PER } \langle \text{expr}^3 \rangle$

donde

$\langle \text{expr} \rangle$

denota cualquier relación.

Llamamos a

$\langle \text{expr}^1 \rangle$ : dividendo

$\langle \text{expr}^2 \rangle$ : divisor

$\langle \text{expr}^3 \rangle$ : mediador

**Restricciones**

$H(\text{dividendo}) \cap H(\text{divisor}) = \emptyset,$

$H(\text{mediador}) = H(\text{dividendo}) \cup H(\text{divisor})$

# División

A	B	C		R
X	Y	X	Y	X
a	x	a	x	a
b	z	a	y	
c		a	z	
		b	x	
		c	y	

- Semántica**

Si en

$A \text{ DIVIDEBY } B \text{ PER } C$

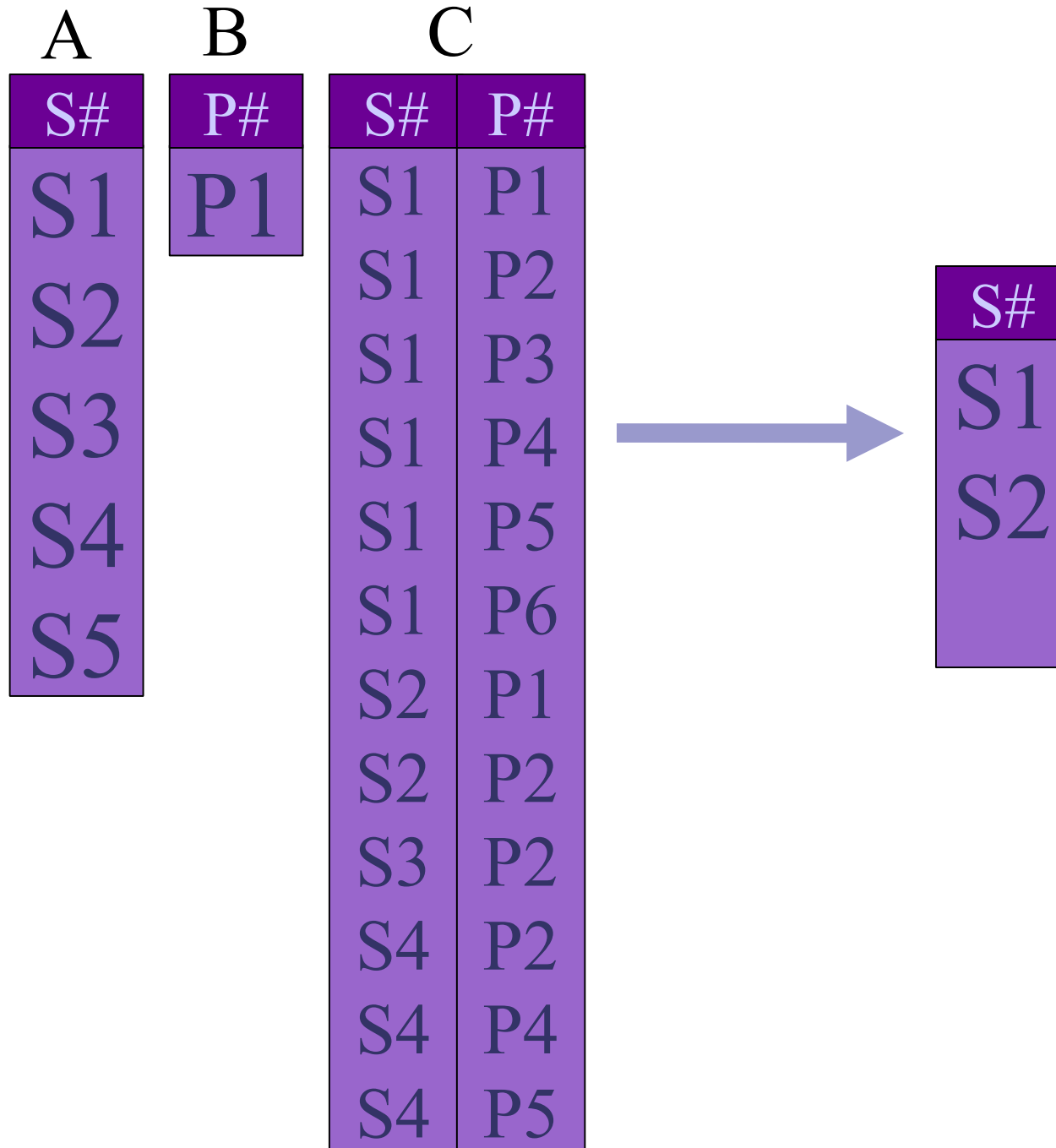
X es el conjunto de atributos de A e

Y es el conjunto de atributos de B

El resultado R será una relación donde  $H(R)=X$  y las Tuplas de R serán aquellas tuplas de A  $\{X:x\}$  para las que una tupla de C  $\{X:x,Y:y\}$  aparece en C para todas las tuplas  $\{Y:y\}$  que aparecen en B.

# Ejemplo


A DIVIDEBY B PER C



# Ejemplo

A **DIVIDE** BY B **PER** C

A		B		C			
S#		P#		S#	P#		
S1		P2		S1	P1		
S2		P4		S1	P2		
S3				S1	P3		
S4				S1	P4		
S5				S1	P5		
				S1	P6		
				S2	P1		
				S2	P2		
				S3	P2		
				S4	P2		
				S4	P4		
				S4	P5		




S#
S1
S4

# Ejemplo

A **DIVIDE** BY B **PER** C

A		B		C			
S#	P#	S#	P#	S#	P#		
S1	P1	S1	P1				
S2	P2	S1	P2				
S3	P3	S1	P3				
S4	P4	S1	P4				
S5	P5	S1	P5				
	P6	S2	P1				
		S2	P2				
		S3	P2				
		S4	P2				
		S4	P4				
		S4	P5				



S#
S1

# ¿ Dónde aplicar la división ?

La división equivale al cuantificador 'Para todo'

Ejemplo

¿ Existirá algún docente que haya pedido todos los libros ?

```
(project dni (docente))  
  divideby  
(project isbn (libro))  
  per  
(project isbn,dni ((ejemplar) njoin  
  (prestamo)))
```



# ¿ Dónde aplicar la división ?

**Ejemplo:**

**¿ Existirá algún libro que haya sido pedido por todos los docentes ?**

```
(project isbn (libro))  
  divideby  
(project dni (docente))  
  per  
(project isbn,dni ((ejemplar) njoin  
  (presmamo))
```

# Usos del Álgebra relacional

- Extraer información de la base de datos  
`<expresión>`
- Actualizar información  
`relvar ← <expresión>`
- Comparar  
`<expresión> * <expresión>`
- Comprobar la integridad  
`assert <expresión>`

# Usos del Álgebra relacional

- Definir relvars *derivadas* (**vistas e instantáneas**).
- Definir requerimientos de estabilidad para operaciones que precisen control de concurrencia.
- Definir restricciones de seguridad.

# Usos del Álgebra relacional

Servir de marco de referencia para la definición de lenguajes de consulta y manipulación de datos en Gestores de Bases de Datos relacionales.

Ejemplo: SQL