

# **Modelo Relacional**

## **Lenguajes de manejo de datos (DMLs)**

**Bibliografía: Fundamentos de bases de datos  
Korth , Silberschatz**

# Lenguajes de manejo de datos

- **DML** es una notación para expresar
  - **consultas**,
  - actualización,
  - inserción y
  - borrado.
- Un **lenguaje de consulta** es un lenguaje en el que un usuario solicita información de la BD.

# Lenguajes de manejo de datos

Los **Lenguajes** se clasifican en:

- **algebraicos:** aplican operadores sobre relaciones (**ISBL**)
- **de cálculo:** especifican un predicado que las tuplas deben satisfacer
  - **relacional de tuplas (QUEL)**
  - **relacional de dominios (QBE)**
- **Intermedios: SQUARE y SEQUEL (SQL)**

# Lenguajes de manejo de datos

Pueden clasificarse en:

- **procedimentales**

- el usuario **da instrucciones** al sistema para que realice una secuencia de operaciones en la BD para calcular el resultado.
- **Álgebra relacional**

- **no procedimentales**

- el usuario **describe la información** deseada sin dar un procedimiento específico para obtener esa información
- **Cálculo relacional**

# Álgebra relacional

- Es un lenguaje de consulta **procedimental**.
- Consta de un **conjunto de operaciones** que toman una o dos relaciones como entrada y producen una nueva relación como resultado.
- Las operaciones fundamentales del AR son suficientes para expresar cualquier consulta.

# Álgebra relacional

Las **operaciones fundamentales** son:

- **selección**
  - **proyección**
  - **producto cartesiano** (de relaciones)
  - **unión**
  - **diferencia ó resta**
  - **renombrar**
- 
- Si nos restringimos **sólo** a estas operaciones algunas **consultas** son **largas de expresar**  $\Rightarrow$  se definen operaciones **adicionales**.

# Álgebra relacional

Las operaciones **adicionales** son:

- **intersección**
  - **producto natural**
  - **división**
  - **asignación**
- 
- Se definen en términos de las operaciones fundamentales.
  - No añaden potencia al AR pero simplifican consultas comunes.

# AR: Operaciones fundamentales

Las operaciones **seleccionar, proyectar y renombrar** se llaman operaciones **unarias**, ya que **operan sobre una relación**.

Las otras tres operaciones: **unión, resta y producto cartesiano**, operan **sobre pares de relaciones** y, por tanto, se llaman operaciones **binarias**.



# AR: Operaciones fundamentales

Sean  $r$  y  $s$  las siguientes relaciones:

A	B	C
a	b	c
d	a	f
c	b	d

Relación  $r$

D	E	F
b	g	a
d	a	f

Relación  $s$

# AR: Operaciones fundamentales

Unión:

**r**  $\cup$  **s**

a	b	c
d	a	f
c	b	d
b	g	a

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

# AR: Operaciones fundamentales

Diferencia:

**r - s**

a	b	c
c	b	d

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

# AR: Operaciones fundamentales

Proyección:

$\pi_{A,C}(r)$

A	C
a	c
d	f
c	d

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

# AR: Operaciones fundamentales

Selección:

$\sigma_{B = \text{"b"}}(r)$

A	B	C
a	b	c
c	b	d

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

# AR: Operaciones fundamentales

Producto  
cartesiano:

**r x s**

	A	B	C	D	E	F
a	b	c	b	g	a	
a	b	c	d	a	f	
d	a	f	b	g	a	
d	a	f	d	a	f	
c	b	d	b	g	a	
c	b	d	d	a	f	

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

# AR: Operación selección

- **Selecciona tuplas que satisfacen un predicado dado.**

$$\sigma_F(r)$$

- Se usa la letra griega minúscula sigma ( $\sigma$ ) para indicar la selección.
- El **predicado F** aparece como subíndice de  $\sigma$ .
- La **relación** argumento (**r**) se escribe a continuación de  $\sigma$  entre paréntesis

# AR: Operación selección

## Ejemplos

- Seleccionar las tuplas de la relación préstamo en las que la sucursal es Perryridge

$\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}}(\text{préstamo})$



# AR: Operación selección

## Ejemplos

La relación Préstamo es:

nombre-sucursal	número-préstamo	nombre-cliente	cantidad
Downtown	17	Jones	1000
Mianus	93	Curry	500
Perryridge	15	Hayes	1500
Round Hill	11	Turner	900
Perryridge	25	Glenn	2500
Redwood	23	Smith	2000
Brighton	10	Brooks	2200
Downtown	14	Jackson	1500
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000

# AR: Operación selección

## Ejemplos

La relación que resulta de la consulta

$\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}}(\text{préstamo})$

es:

nombre-sucursal	número-préstamo	nombre-cliente	cantidad
Perryridge	15	Hayes	1500
Perryridge	25	Glenn	2500

# AR: Operación selección

## Ejemplos

- Encontrar todas las tuplas en las que la cantidad prestada es más de 1200 dólares

$\sigma_{\text{cantidad} > 1200}$  (**préstamo**)

En general, se permiten las comparaciones usando

$= \neq < \leq > \geq$

en el predicado de selección.

# AR: Operación selección

## Ejemplos

- Encontrar las tuplas en préstamo de más de 1200 dólares hechas por la sucursal Perryridge

$\sigma$  nombre-sucursal = "Perryridge"  $\wedge$  cantidad > 1200  
**(préstamo)**

Se pueden combinar varios predicados usando los conectores

$\wedge$

$\vee$

# AR: Operación selección

## Ejemplos

Supongamos que se tiene el esquema de relación que indica

que un cliente tiene un “banquero personal”

servicio (nombre-cliente, nombre-banquero)

Nombre-cliente	Nombre-banquero
Turner	Johnson
Hayes	Jones
Johnson	Johnson

- Encontrar los clientes que tienen el mismo nombre que su banquero personal

# AR: Operación selección

## Ejemplos

- Encontrar los clientes que tienen el mismo nombre que su banquero personal.

$\sigma$  nombre-cliente = nombre-banquero (**servicio**)

Nombre-cliente	Nombre-banquero
Johnson	Johnson

El predicado de selección puede incluir  
**comparaciones entre dos atributos.**

# AR: Operación proyección

- Es una operación unaria que devuelve su relación argumento con ciertas columnas omitidas.

$$\pi_{i_1, \dots, i_k}(r)$$

- Se listan los **atributos** que se desea que aparezcan en el resultado como **subíndices** de  $\pi$ .
- Dado que el resultado es una relación, se eliminan todas las filas duplicadas.

# AR: Operación proyección

## Ejemplos

- Obtener una relación que muestre los **clientes** y las **sucursales** en las que tienen préstamos, pero no la cantidad del préstamo, ni el número del préstamo.

$\pi$  nombre-sucursal, nombre-cliente (**préstamo**)



# AR: Operación proyección

## Ejemplos

El resultado es:

nombre-sucursal	nombre-cliente
Downtown	Jones
Mianus	Curry
Perryridge	Hayes
Round Hill	Turner
Perryridge	Glenn
Redwood	Smith
Brighton	Brooks
Downtown	Jackson
Pownal	Williams
North Town	Adams
Downtown	Johnson

# AR: Operación proyección

## Ejemplos

- Encontrar a los clientes que tienen el mismo nombre que su banquero personal.

Como vimos, esta consulta produce el siguiente resultado:

Nombre-cliente	Nombre-banquero
Johnson	Johnson

- Es **redundante** listar dos veces el nombre de la persona.

# AR: Operación proyección

## Ejemplos

- Sería preferible una relación de un atributo sobre **(nombre-cliente)** que liste todos aquellos que tengan el mismo nombre que su banquero personal.
- La operación **proyectar** nos permite producir esta relación.

$\pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-cliente}=\text{nombre-banquero}}(\text{servicio}))$

- El **argumento** de la proyección puede ser el **resultado de otra operación**.

# AR: Operación producto cartesiano

- **Es una operación binaria que permite combinar relaciones.**

**$r \times s$**

# AR: Operación producto cartesiano

Queremos encontrar a todos los **clientes** del banquero Johnson, y las **ciudades** en las que viven estos clientes.

- Se necesita información de la relación servicio y de la relación cliente.

**u = servicio x cliente**

# AR: Operación producto cartesiano

u (servicio.nombre-cliente, servicio.nombre-banquero, cliente.nombre-cliente, cliente.calle, cliente.ciudad-cliente)

- Se **listan** todos los **atributos de las dos relaciones**, con el nombre de la relación de la que cada atributo procede.
- Se necesita el nombre de la relación para **evitar ambigüedad**:  
**servicio.nombre-cliente  $\neq$  cliente.nombre-cliente**
- Para aquellos atributos que sólo aparecen en uno de los dos esquemas, no es necesario el prefijo.

El resultado de

**servicio x  
cliente**

es:

Servicio. Nombre-cliente	Nombre- banquero	Cliente. Nombre-cliente	Calle	Ciudad-cliente
Turner	Johnson	Jones	Main	Harrison
Turner	Johnson	Smith	North	Rye
Turner	Johnson	Hayes	Main	Harrison
Turner	Johnson	Curry	North	Rye
Turner	Johnson	Lindsay	Park	Pittsfield
Turner	Johnson	Turner	Ptitnam	Starriford
Turner	Johnson	Williams	Nassau	Princeton
Turner	Johnson	Adams	Spring	Pittsfield
Turner	Johnson	Johnson	Alma	Palo Alto
Turner	Johnson	Glenn	Sand Hill	Woodside
Turner	Johnson	Brooks	Senator	Brooklyn
Turner	Johnson	Green	Walnut	Stamford
Hayes	Jones	Jones	Main	Harrison
Hayes	Jones	Smith	North	Rye
Hayes	Jones	Hayes	Main	Harrison
Hayes	Jones	Curry	North	Rye
Hayes	Jones	Lindsay	Park	Pittsfield
Hayes	Jones	Turner	Putnam	Stamford
Hayes	Jones	Williams	Nassau	Princeton
Hayes	Jones	Adams	Spring	Pittsfield
Hayes	Jones	Johnson	Alma	Palo Alto
Hayes	Jones	Glenn	Sand Hill	Woodside
Hayes	Jones	Brooks	Senator	Brooklyn
Hayes	Jones	Green	Walnut	Stamford
Johnson	Johnson	Jones	Main	Harrison
Johnson	Johnson	Smith	North	Rye
Johnson	Johnson	Hayes	Main	Harrison
Johnson	Johnson	Curry	North	Rye
Johnson	Johnson	Lindsay	Park	Pittsfield
Johnson	Johnson	Turner	Putnam	Stamford
Johnson	Johnson	Williams	Nassau	Princeton
Johnson	Johnson	Adams	Spring	Pittsfield
Johnson	Johnson	Johnson	Alma	Palo Alto
Johnson	Johnson	Glenn	Sand Hill	Woodside
Johnson	Johnson	Brooks	Senator	Brooklyn
Johnson	Johnson	Grcen	Wainut	Stamford

# AR: Operación producto cartesiano

¿Qué tuplas aparecen en  $r \times s$  ?

- Se construye **una tupla por cada par posible de tuplas**
  - Para el ejemplo una de la relación servicio y otra de la relación cliente.
- Si se tienen  **$n_1$  tuplas en servicio y  $n_2$  en cliente**. Entonces existen  $n_1 n_2$  formas de elegir un par de tuplas; de forma que hay  **$n_1 n_2$  tuplas en  $r$** .



# AR: Operación producto cartesiano

**Grado** de una relación es la **cantidad de atributos**.

- Si  $\text{gr}(r)=n$  y  $\text{gr}(s)=m$   $\rightarrow$   $\text{gr}(r \times s)=n+m$

En general, dadas dos relaciones  $r(R)$  y  $s(S)$

**$r \times s$**

es una relación cuyo **esquema** es  
la **concatenación de R y S**

# AR: Operación producto cartesiano

**Continuamos con el ejemplo:**

- Encontrar a todos los clientes del **banquero Johnson** y la **ciudad en la que viven**.

**Lo resolvemos por etapas:**

$\sigma_{\text{nombre-banquero} = \text{"Johnson"}}$  (**servicio x cliente**)

- El producto cartesiano toma todos los pares posibles de una tupla de servicio con una tupla de cliente.
- **cliente.nombre-cliente** puede contener clientes de banqueros que no sean Johnson.
- **servicio.nombre-cliente** contiene sólo clientes de Johnson.

Servicio. Nombre-cliente	Nombre-banquero	Cliente. Nombre-cliente	Calle	Ciudad-cliente
Turner	Johnson	Jones	Main	Harrison
Turner	Johnson	Smith	North	Rye
Turner	Johnson	Hayes	Main	Harrison
Turner	Johnson	Curry	North	Rye
Turner	Johnson	Lindsay	Park	Pittsfield
Turner	Johnson	Turner	Putnam	Stamford
Turner	Johnson	Williams	Nassau	Princeton
Turner	Johnson	Adams	Spring	Pittsfield
Turner	Johnson	Johnson	Alma	Palo Alto
Turner	Johnson	Glenn	Sand Hill	Woodside
Turner	Johnson	Brooks	Senator	Brooklyn
Turner	Johnson	Green	Walnut	Stamford
Johnson	Johnson	Jones	Main	Harrison
Johnson	Johnson	Smith	North	Rye
Johnson	Johnson	Hayes	Main	Harrison
Johnson	Johnson	Curry	North	Rye
Johnson	Johnson	Lindsay	Park	Pittsfield
Johnson	Johnson	Turner	Putnam	Stamford
Johnson	Johnson	Williams	Nassau	Princeton
Johnson	Johnson	Adams	Spring	Pittsfield
Johnson	Johnson	Johnson	Alma	Palo Alto
Johnson	Johnson	Glenn	Sand Hill	Woodside
Johnson	Johnson	Brooks	Senator	Brooklyn
Johnson	Johnson	Green	Walnut	Stamford

- Para obtener la **ciudad** del cliente, interesarán las tuplas en las que:

**servicio.nombre-cliente = cliente.nombre-cliente**

Así si se escribe

$\sigma_{\text{servicio.nombre-cliente=cliente.nombre-cliente}}$   
 $(\sigma_{\text{nombre-banquero="Johnson"}}(\text{servicio x cliente}))$

- Finalmente, como sólo se desea nombre-cliente y ciudad-cliente, se hace una **proyección**:

$$\pi_{\text{servicio.nombre-cliente, ciudad-cliente}} \left( \sigma_{\text{servicio.nombre-cliente} = \text{cliente.nombre-cliente}} \left( \sigma_{\text{nombre-banquero} = \text{"Johnson"}} (\text{servicio x cliente}) \right) \right)$$

- El resultado de esta expresión es la respuesta correcta a la pregunta.

Servicio. Nombre-cliente	Ciudad-cliente
Turner	Stamford
Johnson	Palo Alto

# AR: Operación unión

- Es una operación **binaria** cuyo resultado es una relación
  - cuyo **esquema** es idéntico al de  $r$  ó  $s$  y
  - cuyo **cuerpo** está formado por todas las **tuplas  $t$  pertenecientes a  $r$  ó a  $s$  ó a las dos.**

$$r \cup s$$

- Para que sea **válida** se exige que se cumplan dos **condiciones**:
  - $r$  y  $s$  deben tener el **mismo número de atributos.**
  - **Dominios** del  $i$ -ésimo atributo de  $r$  y  $s$  **deben ser iguales**

# AR: Operación unión

## Ejemplo

- Encontrar a todos los **clientes** de la sucursal Perryridge.

Es decir, encontrar a las personas que tienen un **préstamo**, una **cuenta**, o **ambos**.

Se necesita información de

- Préstamo
- Depósito

# AR: Operación unión

## Ejemplo

- Para encontrar a los clientes con **préstamo** en Perryridge:

$\pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}}(\text{préstamo}))$

- Y para encontrar a los clientes con **cuenta** en Perryridge:

$\pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}}(\text{depósito}))$

Para contestar la consulta, se necesita la **unión** de estos 2 conjuntos



# AR: Operación unión

## Ejemplo

Es decir, todos los clientes que aparecen en **cualquiera de las dos relaciones o en ambas.**

$$\pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}} (\text{préstamo})) \cup \pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-sucursal} = \text{"Perryridge"}} (\text{depósito}))$$

- La relación que resulta es:

Nombre-cliente
Hayes
Glenn
Williams

- Las relaciones son conjuntos  $\Rightarrow$  se **eliminan** valores **duplicados.**

# AR: Operación renombre

- Asigna un **nuevo nombre x** a una relación r

$$\rho_x(r)$$

Para eliminar ambigüedades se nombraron atributos mediante

**nombre-relación.nombre-atributo**

**Otra** forma de **ambigüedad** surge cuando **la misma relación aparece más de una vez** en una pregunta.

# AR: Operación renombrar

## Ejemplo

- Encontrar los clientes que viven en la misma calle y en la misma ciudad que Smith.

Se puede obtener la calle y la ciudad de Smith escribiendo

$\pi$  calle, ciudad-cliente ( $\sigma$  nombre-cliente = "Smith" (cliente))

# AR: Operación renombrar

## Ejemplo

Para encontrar otros clientes con esa calle y esa ciudad, debemos hacer **referencia una segunda vez** a la relación cliente:

$$\sigma_P(\text{cliente } x$$
$$(\pi_{\text{calle, ciudad-cliente}}(\sigma_{\text{nombre-cliente} = \text{"Smith"}}(\text{cliente})))$$
$$)$$

donde **P** es un predicado de selección que requiere que los valores de **calle** y de **ciudad-cliente** sean **iguales**.

# AR: Operación renombrar

## Ejemplo

Para especificar a qué valor de calle nos referimos, no podemos usar **cliente.calle = cliente.calle**, ya que ambos valores de calle se toman de la **misma** relación **cliente**.

Una dificultad parecida existe para **cliente.ciudad**.

Este problema **se resuelve** usando la operación  $\rho$  para cambiar el nombre una vez a cliente y así poder hacer referencia a la relación dos veces sin ambigüedad.

# AR: Operación renombrar

## Ejemplo

- Se asigna el nombre **cliente2** a la relación cliente,
- y se **refiere** a **cliente2** al calcular la **calle** y la **ciudad** de Smith.

$$\pi_{\text{cliente.nombre-cliente}} \left( \sigma_{\text{cliente2.calle=cliente.calle} \wedge \text{cliente2.ciudad-cliente=cliente.ciudad-cliente}} \left( \text{cliente} \times \left( \pi_{\text{calle,ciudad-cliente}} \left( \sigma_{\text{nombre-cliente="Smith"}} \left( \rho_{\text{cliente2}}(\text{cliente}) \right) \right) \right) \right) \right)$$

El resultado de esta consulta es:

Nombre-cliente
Smith
Curry

# AR: Operación resta

Permite encontrar **tuplas** que estén **en una relación pero no en otra.**

$$r - s$$

- La relación resultante contiene aquellas tuplas que **están en r pero no en s.**
- Para que sea **válida** se exige que se cumplan dos **condiciones:**
  - r y s deben tener el **mismo número de atributos.**
  - Los **dominios** del i-ésimo atributo de r y de s **deben ser los mismos.**

# AR: Operación resta

## Ejemplo

- Encontrar los clientes de la sucursal Perryridge que tienen una cuenta allí, pero no un préstamo

$\pi$  nombre-cliente ( $\sigma$  nombre-sucursal = "Perryridge"  
**(depósito)**)

-

$\pi$  nombre-cliente ( $\sigma$  nombre-sucursal = "Perryridge"  
**(préstamo)**)

La relación que resulta para esta consulta es:

Nombre-cliente
Williams



# AR: Otro ejemplo

- Encontrar el mayor saldo de cuenta en el banco.
  - Esta consulta puede expresarse usando **funciones de agregación**, por ejemplo MAX().
  - Pero podemos escribirla usando **solamente las operaciones fundamentales**:

1. Calcular una relación con saldos que no son los más grandes:

$$\pi_{\text{depósito.saldo}}(\sigma_{\text{depósito.saldo} < \text{d.saldo}}(\text{depósito} \times \rho_d(\text{depósito})))$$

El resultado contiene todos los saldos **excepto el mayor** de todos.

- El resultado contiene todos los saldos excepto el mayor de todos.

Es decir:

Saldo
500
700
400
350
750
850

2. Tomar la diferencia de conjuntos entre la relación depósito y el resultado anterior.

$\pi_{\text{saldo}}(\text{depósito})$

-

$\pi_{\text{depósito.saldo}}(\sigma_{\text{depósito.saldo} < \text{d.saldo}}(\text{depósito} \times \rho_d(\text{depósito})))$

siendo el resultado de esta consulta:

Saldo
900

# Definición formal del álgebra relacional

- Una **expresión general** en el AR se construye a **partir de subexpresiones**.

Sean  $E1$  y  $E2$  expresiones del AR.

Entonces las siguientes son todas expresiones del AR:

- $E1 \cup E2$
- $E1 - E2$
- $E1 \times E2$
- $\sigma_P(E1)$ , donde  $P$  es predicado con atributos de  $E1$
- $\pi_S(E1)$ , donde  $S$  es una lista de atributos de  $E1$
- $\rho_x(E1)$ , donde  $x$  es el nuevo nombre de  $E1$ .

# Operaciones adicionales

- Las **operaciones fundamentales** del AR son **suficientes para expresar cualquier consulta.**
- Si nos restringimos solo a las fundamentales algunas consultas son largas de expresar,
- por tanto se definen **operaciones adicionales** que **no añaden potencia** pero que **simplifican consultas.**
- Para cada nueva operación se da una **expresión equivalente** usando sólo las fundamentales.

# AR: Operación intersección

$$\mathbf{r} \cap \mathbf{s}$$

- Es una relación
  - cuya **cabecera** es idéntica a la de  $r$  ó  $s$  y
  - cuyo **cuerpo** está formado por **tuplas  $t$  pertenecientes tanto a  $r$  como a  $s$ .**
- Es una operación binaria.
- Para que sea válida, las relaciones deben tener el mismo **número de atributos**, definidos en los **mismos dominios**.
- La **intersección** equivale a:

$$\mathbf{r} \cap \mathbf{s} = \mathbf{r} - (\mathbf{r} - \mathbf{s})$$

# AR: Operación intersección

A	B	C
a	b	c
d	a	f
c	b	d

Relación r

D	E	F
b	g	a
d	a	f

Relación s

d	a	f
---	---	---

$r \cap s$

# AR: Operación intersección

- Encontrar a todos los clientes con un préstamo y una cuenta en la sucursal Perryridge.

$$\pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal}=\text{"Perryridge"}}(\text{préstamo}))$$
$$\cap$$
$$\pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal}=\text{"Perryridge"}}(\text{depósito}))$$

La relación que resulta es:

Nombre-cliente
Hayes



# AR: Operación producto natural

## Producto $\alpha$ ó $\alpha$ -join

$$r \underset{i \alpha j}{\bowtie} s = \sigma_{i \alpha (n+j)} (r \times s)$$

donde:

$\alpha$  es operador de comparación ( $>$   $<$   $=$  etc.)

$i$  = atributos de  $r$

$j$  = atributos de  $s$

$gr(r) = n$

# AR: Operación producto natural

**Equi-join** : Cuando  $\alpha$  es el operador igual (=).

**Natural-join ó producto natural** : Cuando se tienen dos relaciones con **una o más columnas con el mismo nombre**.

- Se representa  **$r \bowtie s$**
- Equivale a:
  - realizar un **producto cartesiano** de sus 2 argumentos,
  - realizar una **selección forzando la igualdad en atributos comunes**
  - y, finalmente, **quitar las columnas duplicadas**.

# AR: Operación producto natural

## Ejemplo

- Encontrar clientes que tienen un préstamo y las ciudades en las que viven.

$\pi$  préstamo.nombre-cliente, ciudad-cliente  
( $\sigma_{\text{préstamo.nombre-cliente} = \text{cliente.nombre-cliente}}$  (**préstamo x cliente**))

Equivalentemente usando el **producto natural** :

$\pi$  nombre-cliente, ciudad-cliente (**préstamo |x| cliente**)

Nombre-cliente	Ciudad-cliente
Jones	Harrison
Smith	Rye
Hayes	Harrison
Curry	Rye
Turner	Stamford
Williams	Princeton
Adams	Pittsfield
Johnson	Palo Alto
Glenn	Woodside
Brooks	Brooklyn

# Definición del producto natural

Sean las dos relaciones  $r(R)$  y  $s(S)$ ,

el producto natural  $r \bowtie s$  es:

- la proyección sobre  $R \cup S$
- de una selección, donde el predicado requiere que  $r.A=s.A$  para cada atributo  $A$  en  $R \cap S$ .
- en  $r \times s$

$$r \bowtie s = \pi_{R \cup S} (\sigma_{r.A_1=S.A_1 \wedge r.A_2=S.A_2 \wedge \dots \wedge r.A_n=S.A_n} (r \times s))$$

donde  $R \cap S = \{A_1, A_2, \dots, A_n\}$  y el esquema resultante es  $R \cup S$

# AR: Operación producto natural

## Ejemplo

- Encontrar el activo y el nombre de todas las sucursales que tienen depositantes
  - es decir, clientes con una cuenta, que viven en Stamford.

$\pi$  nombre-sucursal, activo (  $\sigma$  ciudad-cliente="Stamford"  
(cliente|x|depósito |x| sucursal) )

La relación que resulta es:

Nombre-sucursal	Activo
Round Hill	8000000
Brighton	7100000
Downtown	9000000

# AR: Operación producto natural

## Ejemplo

- Encontrar a todos los clientes que tienen una cuenta y un préstamo en la sucursal Perryridge.

$\pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal}=\text{"Perryridge"}}(\text{préstamo} \bowtie \text{depósito}))$

Esta expresión equivale a la intersección:

$\pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal}=\text{"Perryridge"}}(\text{préstamo}))$   
 $\cap$

$\pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal}=\text{"Perryridge"}}(\text{depósito}))$

En general en el AR es posible escribir varias **expresiones equivalentes** que son bastante diferentes entre sí.

# AR: Operación producto natural

## Observación:

Sean  $r(\mathbf{R})$  y  $s(\mathbf{S})$  relaciones  
sin **ningún atributo en común**,

es decir,  $\mathbf{R} \cap \mathbf{S} = \emptyset$  (conjunto vacío.)

Entonces

$$r \mid \mathbf{x} \mid s = r \times s.$$



# AR: Operación división

Sean los esquemas de relación:

$$r(X_1, \dots, X_m, Y_1, \dots, Y_n) \text{ y } s(Y_1, \dots, Y_n)$$

Llamemos  $X$  al atributo compuesto  $X_1, \dots, X_m$

$Y$  al atributo compuesto  $Y_1, \dots, Y_n$

La división  $r/s$  es una relación con

- **esquema**  $X$
- **cuerpo** formado por **todas** las tuplas  $\mathbf{x}$  tales que aparece una tupla  $(\mathbf{x}, \mathbf{y})$  en  $r$  para todas las tuplas  $\mathbf{y}$  de  $s$

# AR: Operación división

**$r / s$**

$x / (x,y)$  aparece en  $r \quad \forall y \quad \text{en } s$

- Se establece para aquellas consultas que incluyen la frase «**para todos**».
- **r** representa al Dividendo
- **s** representa al Divisor

# AR: Operación división

- Sean las relaciones **a** (dividendo) y **b** (divisor)

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

/

P#
P2
P4

=

S#
S1
S4

# AR: Operación división

- Sean las relaciones **a** (dividendo) y **b** (divisor)

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

/

P#
P1
P2
P3
P4
P5
P6

=

S#
S1

# AR: Operación división

## Ejemplo

- Encontrar los clientes que tienen una cuenta en **todas** las sucursales que están en Brooklyn.

Podemos obtener todas las **sucursales en Brooklyn** mediante :

$r1 = \pi_{\text{nombre-sucursal}}(\sigma_{\text{ciudad-sucursal}=\text{"Brooklyn"}}(\text{sucursal}))$

resultando:

Nombre-sucursal
Brighton
Downtown

Se puede encontrar todos los pares nombre-cliente, nombre-sucursal

con:

**$r2 = \pi_{\text{nombre-cliente, nombre-sucursal}}(\text{depósito})$**

resultando:

Nombre-cliente	Nombre-sucursal
Johnson	Downtown
Smith	Mianus
Hayes	Perryridge
Turner	Round Hill
Williams	Perryridge
Lindsay	Redwood
Green	Brighton
Green	Downtown

Ahora se desea encontrar a los clientes que aparecen en **r2** con **cada** nombre de sucursal en **r1**.

- La operación que proporciona exactamente esos clientes es la operación de **dividir**.
- Esto es, hacer : **r2 / r1**

$\pi_{\text{nombre-cliente, nombre-sucursal}}$  (**depósito**)  
/

$\pi_{\text{nombre-sucursal}}$  ( $\sigma_{\text{ciudad-sucursal="Brooklin"}}$  (**sucursal**))

- El resultado es una relación que tiene el esquema (nombre-cliente) y contiene la tupla (Green).

Ahora se desea encontrar a los clientes que aparecen en **r2** con **cada** nombre de sucursal en **r1**.

- La operación que proporciona exactamente esos clientes es la operación de **dividir**.
- Esto es, hacer : **r2 / r1**

$$\pi_{\text{nombre-cliente, nombre-sucursal}} (\text{depósito})$$
$$/$$
$$\pi_{\text{nombre-sucursal}} (\sigma_{\text{ciudad-sucursal}=\text{"Brooklin"}}(\text{sucursal}))$$

El resultado es una relación que

- tiene el esquema (nombre-cliente)
- contiene la tupla (Green).

Nombre-cliente
Green



# AR: Operación división

Formalmente:

Sean  $r(R)$  y  $s(S)$  relaciones, y  $S \subseteq R$ .

- La relación  $r / s$  tiene **esquema  $R - S$**
- Una tupla  $t$  está en  $r / s$  si para cada tupla  $t_s$  en  $s$  existe una tupla  $t_r$  en  $r$  que satisface las dos condiciones siguientes:

$$t_r [S] = t_s [S]$$

$$t_r [R - S] = t [R - S]$$

# AR: Operación división

- La operación división puede definirse **en términos de las operaciones fundamentales.**

Sea  $r(R)$  y  $s(S)$ , con  $S \subseteq R$ .

$$\mathbf{r / s = \pi_{R-S}(r) - \pi_{R-S}((\pi_{R-S}(r) \times s) - r)}$$

donde:

- $\pi_{R-S}(r)$  da todas las tuplas que satisfacen la segunda condición de la definición de división ( $t_r[R-S] = t[R-S]$ ).
- $\pi_{R-S}((\pi_{R-S}(r) \times s) - r)$  elimina aquellas tuplas que no satisfagan la primera condición ( $t_r[S] = t_s[S]$ ).

*Veamos cómo se hace esto ...*

$$r / s = \pi_{R-S}(r) - \pi_{R-S}(\pi_{R-S}(r) \times s) - r$$

$$\pi_{R-S}(r) \times s$$

es una relación de esquema R que empareja cada tupla de  $\pi_{R-S}(r)$  con cada tupla en **s**.

$$(\pi_{R-S}(r) \times s) - r$$

da los pares de tuplas de  $\pi_{R-S}(r)$  y **s** que **no aparecen** en **r**.

Si una tupla de **t** está en:

$$\pi_{R-S}(\pi_{R-S}(r) \times s) - r$$

entonces existe alguna tupla en **s** que no se combina con la tupla **t** para formar una tupla en **r**.

∴ **t** contiene un valor para los atributos **R - S** que no tiene que aparecer en **r / s**.

- Estos son valores que se eliminan (al restar) de  $\pi_{R-S}(r)$ .

# AR: Operación asignación

A veces conviene escribir una expresión del AR por partes usando la **asignación** a una **variable de relación temporal**.

Por ejemplo:  $r/s$  puede escribirse como

$$\mathbf{temp} \leftarrow \pi_{R-S}(r)$$

$$\mathbf{temp} - \pi_{R-S}((\mathbf{temp} \times s) - r)$$

- La evaluación de una asignación no da como resultado una relación que se presenta al usuario.
- El resultado de la expresión a la derecha de  $\leftarrow$  es asignado a la variable de relación a la izquierda
- Esta variable de relación puede usarse en subsiguientes expresiones

## **Definición:**

Un **lenguaje es relacionalmente completo** si es al menos **tan expresivo** como el **álgebra**.

Es decir,

si sus expresiones permiten la definición de cualquier relación que pueda definirse mediante expresiones del álgebra.