

Sistemas de Información II

Tema 6. Álgebra relacional

Bibliografía:

Elmasri y Navathe: “Fundamentos de Sistemas de Bases de Datos”

3ª edición, 2002 (Capítulo 7).

Garcia-Molina, Ullman y Widom: “Database systems: the complete book”. Prentice-Hall (Capítulo 5).

Carlos Castillo

UPF – 2008

Operaciones

- Proyectar (π)
- Seleccionar (σ)
- Producto cartesiano (\times)
- Join ó Reunir (\bowtie)
- Operaciones de conjuntos
 - Unir (\cup)
 - Intersectar (\cap)
 - Restar ($-$)

Proyección (π)

- Selecciona el valor de ciertos atributos de todas las tuplas de una relación

$$\pi_{A_1, A_2, \dots, A_n}(R) = \{ t[A_1, A_2, \dots, A_n] : t \in R \}$$

- Selecciona columnas completas

Proyección (π) ejemplos

Película

ID_Película	Nombre	Año
1	La guerra de las galaxias	1977
2	El señor de los anillos 1	2001
3	Mar Adentro	2004
4	El viaje de Chihiro	2001

Actor

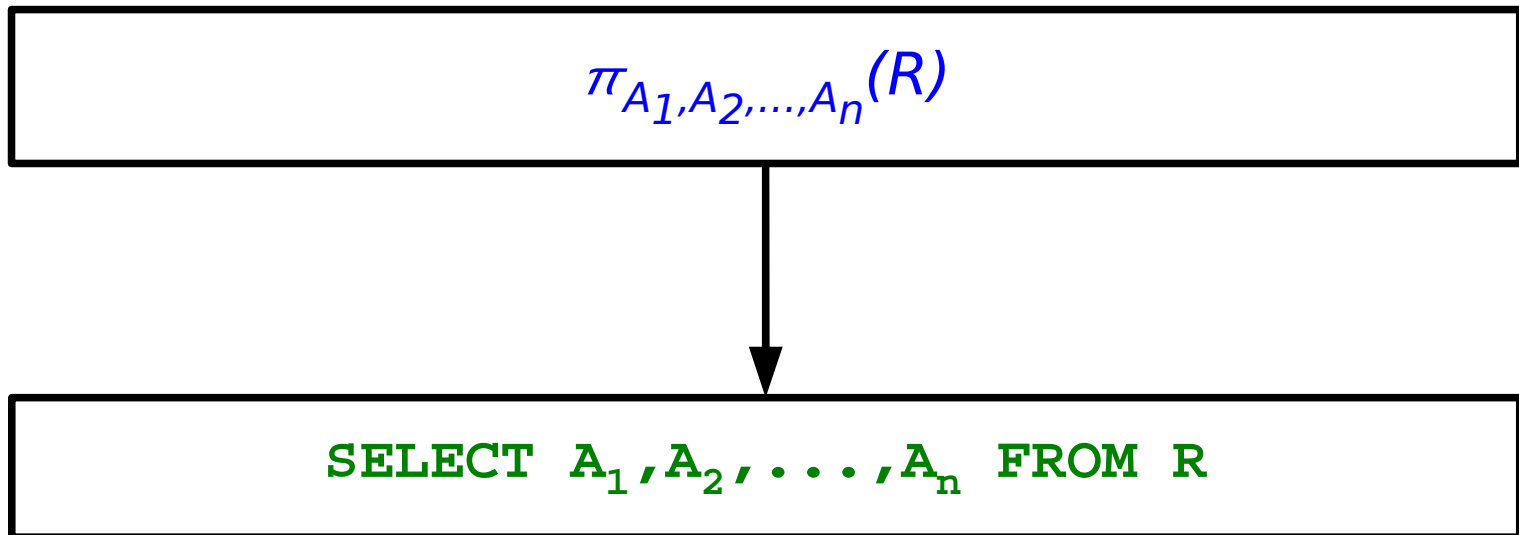
ID_Actor	Nombre	Apellido
1	Mark	Hamill
2	Cristopher	Lee
3	Javier	Bardem
4	Hugo	Weaving

$$\pi_{\text{Año}}(\text{Película}) = \{ \langle 1977 \rangle, \langle 2001 \rangle, \langle 2004 \rangle, \langle 2001 \rangle \}$$

$$\pi_{\text{ID_Película, Año}}(\text{Película}) = \{ \langle 1, 1977 \rangle, \langle 2, 2001 \rangle, \langle 3, 2004 \rangle, \langle 4, 2001 \rangle \}$$

$$\pi_{\text{Nombre}}(\text{Actor}) = \{ \langle \text{Mark} \rangle, \langle \text{Cristopher} \rangle, \langle \text{Javier} \rangle, \langle \text{Hugo} \rangle \}$$

Proyección (π) en SQL



Selección (σ)

- Selecciona el valor de ciertas tuplas

$$\sigma_{\text{condición}}(R) = \{ t \in R : \text{condición}(t) \text{ es cierto} \}$$

- Selecciona filas completas

Selección (σ) ejemplos

Película

ID_Película	Nombre	Año
1	La guerra de las galaxias	1977
2	La comunidad del anillo	2001
3	Mar Adentro	2004
4	El viaje de Chihiro	2001

Actor

ID_Actor	Nombre	Apellido
1	Mark	Hamill
2	Cristopher	Lee
3	Javier	Bardem
4	Hugo	Weaving

$\sigma_{\text{Apellido}=\text{Lee}}(\text{Actor}) =$
 $\{ \langle 2, \text{Cristopher}, \text{Lee} \rangle \}$

$\sigma_{\text{Año} > 2000}(\text{Película}) =$
 $\{ \langle 2, \text{La comunidad del anillo}, 2001 \rangle, \langle 4, \text{El viaje de Chihiro}, 2001 \rangle \}$

Selección (σ) en SQL

$\sigma_{condición}(R)$



`SELECT * FROM R WHERE condición`

Composición de selección y proyección σ, π

Película

ID_Película	Nombre	Año
1	La guerra de las galaxias	1977
2	La comunidad del anillo	2001
3	Mar Adentro	2004
4	El viaje de Chihiro	2001

Actor

ID_Actor	Nombre	Apellido
1	Mark	Hamill
2	Cristopher	Lee
3	Javier	Bardem
4	Hugo	Weaving

$\pi_{\text{Nombre}}(\sigma_{\text{Apellido}=\text{Lee}}(\text{Actor})) =$
 $\{ \langle \text{Cristopher} \rangle \}$

$\pi_{\text{Nombre}}(\sigma_{\text{Año} > 2000}(\text{Película})) =$
 $\{ \langle \text{La comunidad del anillo} \rangle, \langle \text{El viaje de Chihiro} \rangle \}$

Composición (σ y π) en SQL

$\pi_{A_1, A_2, \dots, A_n}(\sigma_{\text{condición}}(R))$



`SELECT A1, A2, ..., An FROM R WHERE condición`

Eliminar duplicados (δ)

Elimina tuplas
duplicadas en una
relación

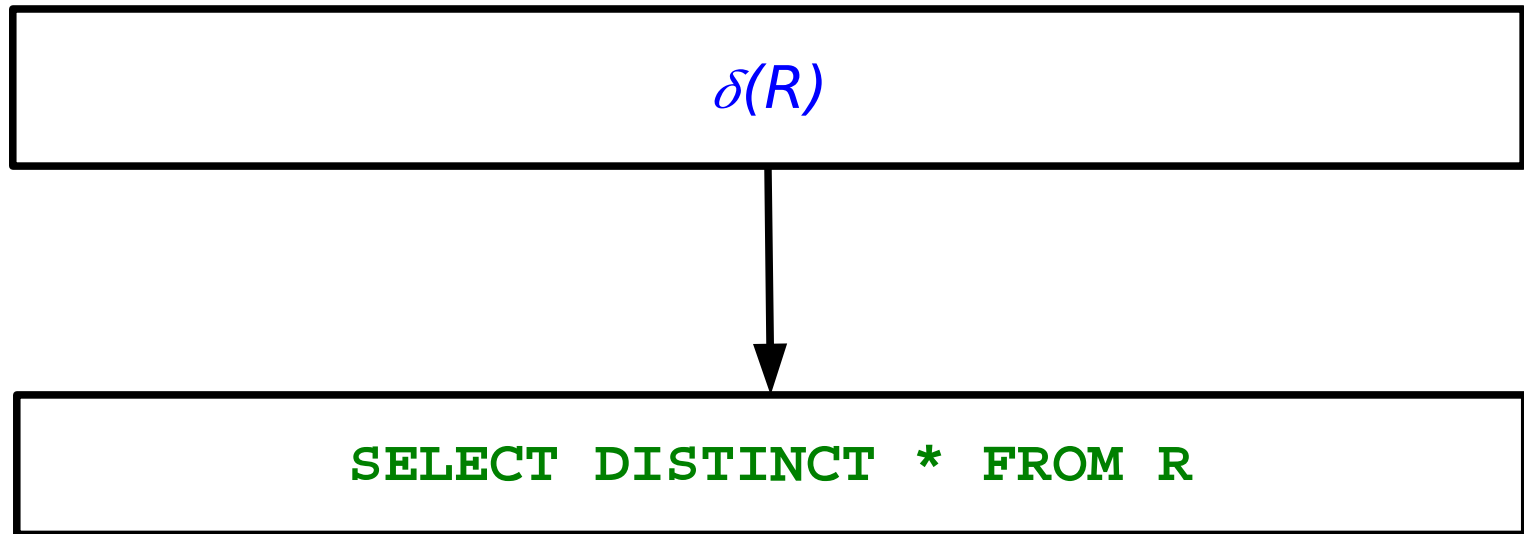
$\delta(R)$

Película

ID Película	Nombre	Año	ID Estudio
1	La guerra de las galaxias	1977	3
2	La comunidad del anillo	2001	2
3	Mar adentro	2004	4
4	El viaje de Chihiro	2001	1

$$\delta(\pi_{\text{Año}}(\text{Película})) = \{ 1997, 2001, 2004 \}$$

Operación delta en SQL



Producto cartesiano (\times)

$$\mathbf{A} \times \mathbf{B} = \{(\mathbf{a}, \mathbf{b}) : \mathbf{a} \in \mathbf{A} \wedge \mathbf{b} \in \mathbf{B}\}$$

Ejemplo:

- $A = \{s, t\}$
- $B = \{u, v, w\}$
- $A \times B = \{s, t\} \times \{u, v, w\}$
 $\{ (s, u), (s, v), (s, w), (t, u), (t, v), (t, w) \}$

La cardinalidad es $|A \times B| = |A||B|$

Producto cartesiano (×) ejemplos

Película

ID_Película	Nombre	Año	ID_Estudio
1	La guerra de las galaxias	1977	3
2	La comunidad del anillo	2001	2
3	Mar adentro	2004	4
4	El viaje de Chihiro	2001	1

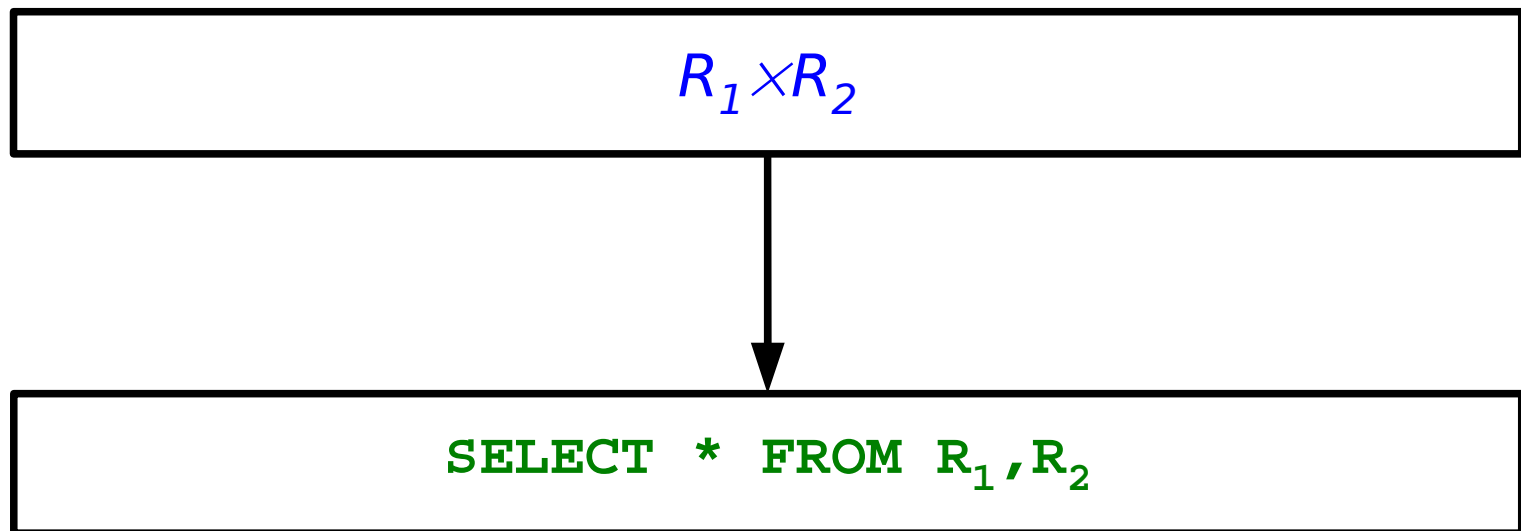
Estudio

ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema
3	Lucasfilms
4	Sogecine

Película × Estudio =

{
<1,La guerra de las galaxias,1977,3,1,Ghibli>,
<1,La guerra de las galaxias,1977,3,2,New Line Cinema>,
<1,La guerra de las galaxias,1977,3,3,Lucasfilms>,
<1,La guerra de las galaxias,1977,3,4,Sogecine>,
<2,La comunidad del anillo,2001,2,1,Ghibli>,
<2,La comunidad del anillo,2001,2,2,New Line Cinema>,
<2,La comunidad del anillo,2001,2,3,Lucasfilms>,
<2,La comunidad del anillo,2001,2,4,Sogecine>,
<3,Mar adentro,2004,4,1,Ghibli>,
<3,Mar adentro,2004,4,2,New Line Cinema>,
... }

Producto cartesiano (\times) en SQL



Seleccionar combinaciones correctas

Película

ID_Película	Nombre	Año	ID_Estudio
1	La guerra de las galaxias	1977	3
2	La comunidad del anillo	2001	2
3	Mar adentro	2004	4
4	El viaje de Chihiro	2001	1

Estudio

ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema
3	Lucasfilms
4	Sogecine

$\sigma_{Película.ID_estudio=Estudio.ID_Estudio}(Película \times Estudio) =$
{
<1,La guerra de las galaxias,1977,3,3,Lucasfilms>,
<2,La comunidad del anillo,2001,2,2,New Line Cinema>,
<3,Mar adentro,2004,4,4,Sogecine>,
<4,El viaje de Chihiro,2001,1,1,Ghibli>
}

Seleccionar combinaciones correctas en SQL

$$\sigma_{R_1.k=R_2.k}(R_1 \times R_2)$$



```
SELECT * FROM R1,R2 WHERE R1.k=R2.k
```

Notación, operación Reunir (*JOIN*)

$$\sigma_{R_1.k=R_2.k} (R_1 \times R_2)$$


$$R_1 \bowtie_k R_2$$

Operación JOIN en SQL

$R_1 \bowtie_k R_2$



`SELECT * FROM R1,R2 WHERE R1.k=R2.k`

Operación JOIN en MySQL

$R_1 \bowtie_k R_2$



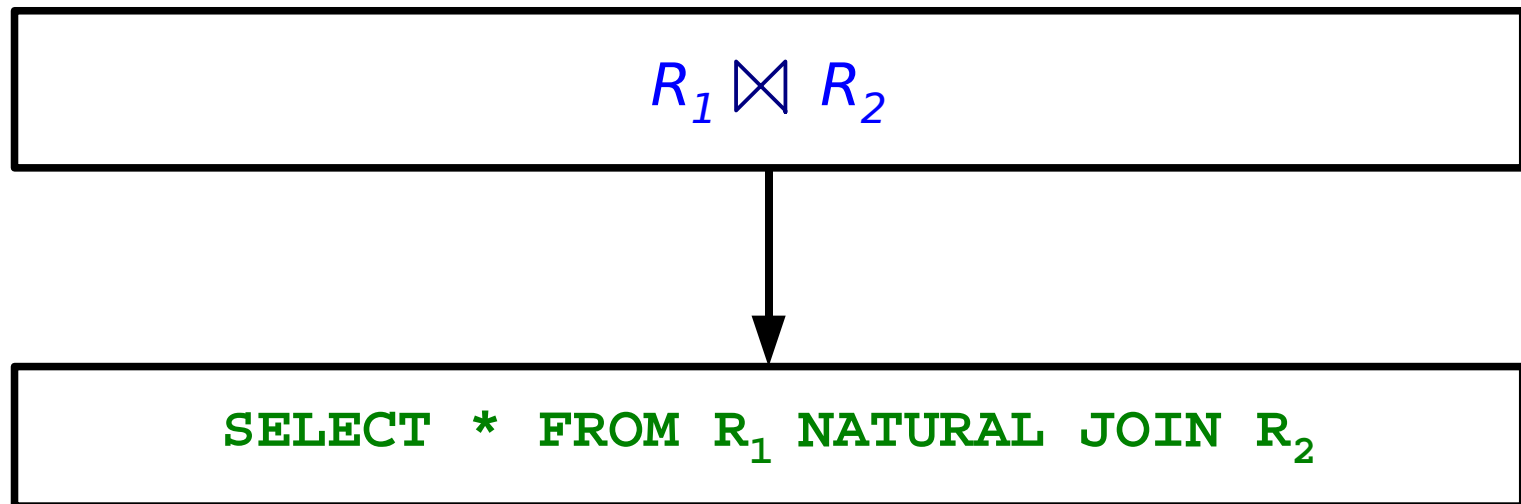
`SELECT * FROM R1 JOIN R2 USING(k)`

JOIN natural

$$R_1 \bowtie R_2$$

- Omitir el subíndice significa:
 - Unir según todos los atributos que tengan el mismo nombre en las dos tablas

Operación NATURAL JOIN en MySQL



Nota: esto usa todos los atributos que se llamen de la misma manera, a veces no es lo que nosotros queremos

Comunitat(`id_comunitat`, `nom`)

Municipi(`id_municipi`, `id_comunitat`, `nom`)

Queremos unir `id_comunitat` pero no `nom`

Ejemplo de NATURAL JOIN

```
mysql> select comunitat.nom, municipi.nom,  
municipi.superficie from comunitat natural join municipi;
```

nom	nom	superficie
Ceuta	Ceuta	19.52
Melilla	Melilla	13.96

2 rows in set (0.14 sec)

```
mysql> select comunitat.nom, municipi.nom,  
municipi.superficie from comunitat join municipi  
using(ca_id);
```

nom	nom	superficie
Andalucía	Abla	45.28
Andalucía	Abrucena	83.18
Andalucía	Adra	89.98

...

LEFT JOIN

- JOIN elimina algunos datos
 - Los que no están en las dos tablas
- LEFT JOIN reemplaza los eliminados por valores nulos en la tabla de la izquierda

Operación LEFT JOIN en MySQL

$R_1 \bowtie_k R_2$



`SELECT * FROM R1 LEFT JOIN R2 USING(k)`

Ejemplo LEFT JOIN

Película

ID_Película	Nombre	Año	ID_Estudio
1	La guerra de las galaxias	1977	3
2	La comunidad del anillo	2001	2
3	Mar adentro	2004	4
4	El viaje de Chihiro	2001	1

Estudio

ID_Estudio	Nombre
1	Ghibli
2	New Line Cinema
3	Lucasfilms
4	Sogecine
5	Nuevo Estudio

```
SELECT count(id_película) AS CNT  
FROM estudio JOIN película  
USING (id_estudio)
```

CNT	Nombre
1	Ghibli
1	New Line Cinema
1	Lucasfilms
1	Sogecine

```
SELECT count(id_película) AS CNT  
FROM estudio LEFT JOIN película  
USING (id_estudio)
```

CNT	Nombre
1	Ghibli
1	New Line Cinema
1	Lucasfilms
1	Sogecine
0	Nuevo Estudio

Otro ejemplo LEFT JOIN

Ciudad

id_ciudad	Nombre
1	Barcelona
2	Berlin
3	Roma
4	Paris
5	Budapest

Viaje

id_salida	id_llegada
1	2
1	4
5	3
5	4
5	2
5	1
2	4

```
SELECT
  ciudad.nombre,COUNT(viaje.id_salida)
FROM
  ciudad LEFT JOIN viaje ON
    (ciudad.id_ciudad=viaje.id_salida)
GROUP BY
  ciudad.nombre;
```

Nombre	CNT
Barcelona	2
Berlin	1
Budapest	4
Paris	0
Roma	0

Ejemplo múltiples JOIN

Ciudad

id_ciudad	Nombre
1	Barcelona
2	Berlin
3	Roma
4	Paris
5	Budapest

Viaje

id_salida	id_llegada
1	2
1	4
5	3
5	4
5	2
5	1
2	4

```
SELECT cs.nombre, cl.nombre
FROM viaje
  JOIN ciudad AS cs ON
(viaje.id_salida=cs.id_ciudad)
  JOIN ciudad AS cl ON
(viaje.id_llegada=cl.id_ciudad);
```

Nombre	Nombre
Barcelona	Berlin
Barcelona	Paris
Budapest	Roma
Budapest	Paris
Budapest	Berlin
Budapest	Barcelona
Berlin	Paris

Resumen

- Proyectar (π): elegir columnas
- Seleccionar (σ): criterio para las filas
- Producto cartesiano (\times): producto tablas
- Join ó Reunir (\bowtie): **combinar tablas**