



# *Bases de Datos*

## Tema 5: Álgebra y cálculo relacional

*Dpto. de Ingeniería Informática*

# Contenidos

- Introducción
- Relaciones de ejemplo
- Álgebra
- Cálculo
- Referencias

# Introducción

- El modelo de datos relacional tiene asociado dos partes:
  - *Estática*: formada por las estructuras que almacenan a los datos y las restricciones que soportan estos datos
  - *Dinámica*: transformaciones entre estados de la Base de Datos
    - Si  $O$  es un operador, se puede pasar de un estado origen ( $BD_i$ ) a un estado destino ( $BD_j$ ) de la Base de Datos como sigue:

$$O(BD_i) = BD_j$$

- Estos estados deben satisfacer las restricciones de integridad estática, y la transformación ha de cumplir las restricciones de integridad dinámica (entre estados)

# Introducción

- El modelo relacional actúa sobre conjuntos de tuplas mediante lenguajes de manipulación relacionales, que asocian una sintaxis concreta a las operaciones
- Se dividen en dos tipos:
  - *Algebraicos*: los cambios de estado se especifican mediante operaciones, los operandos son relaciones y el resultado es una relación. Se conoce como álgebra relacional a «una colección de operaciones que sirven para manipular relaciones enteras»
  - *Predicativos*: los cambios de estado se especifican mediante predicados que definen el estado objetivo sin indicar las operaciones que hay que realizar para llegar al mismo. Basados en el cálculo de predicados

# Relaciones

- En los ejemplos usamos la BD *Biblioteca*.  
Relaciones:
  - Autores (Cód\_Aut, Nombre, Apellido)
  - Editores (Cód\_Edit, Nombre, Apellido)
  - Libros (Cód\_Lib, Nombre, Cód\_Edit, Cód\_Aut)
  - Socios (Cód\_Soc, Nombre, Apellido)
  - Préstamos (Cód\_Lib, Cód\_Soc, Fecha)
- ¿Qué restricciones impone esta estructura y sus claves primarias?

# Relaciones

- Instancia de ejemplo de la BD *Biblioteca*:

- Autores:

<u>Cód_Aut</u>	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

- Editores:

<u>Cód_Edit</u>	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

- Libros:

<u>Cód_Lib</u>	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Relaciones

- Instancia de ejemplo de la BD *Biblioteca*:

- Libros

(*again*):

<u>Cód_Lib</u>	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

- Socios:

<u>Cód_Soc</u>	Nombre	Apellido
1	Eric	Manrique
2	Mar	García

- Préstamos:

<u>Cód_Lib</u>	<u>Cód_Soc</u>	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

# Álgebra relacional

- El álgebra relacional consta de un conjunto de operadores que aplicados a las relaciones dan como resultado nuevas relaciones (se denomina *propiedad de cierre*)
- Una operación consiste en aplicar un operador  $O$  a una relación  $R$  para obtener otra relación  $R'$ , a la cual se puede volver a aplicar otro operador
- El operador se aplica a una extensión de  $R$ ,  $r(R)$ , no al esquema  $R$ , pero se acostumbra a utilizar la notación siguiente:

$$O_n (\cdot \cdot \cdot (O_1 (R))) = R'$$

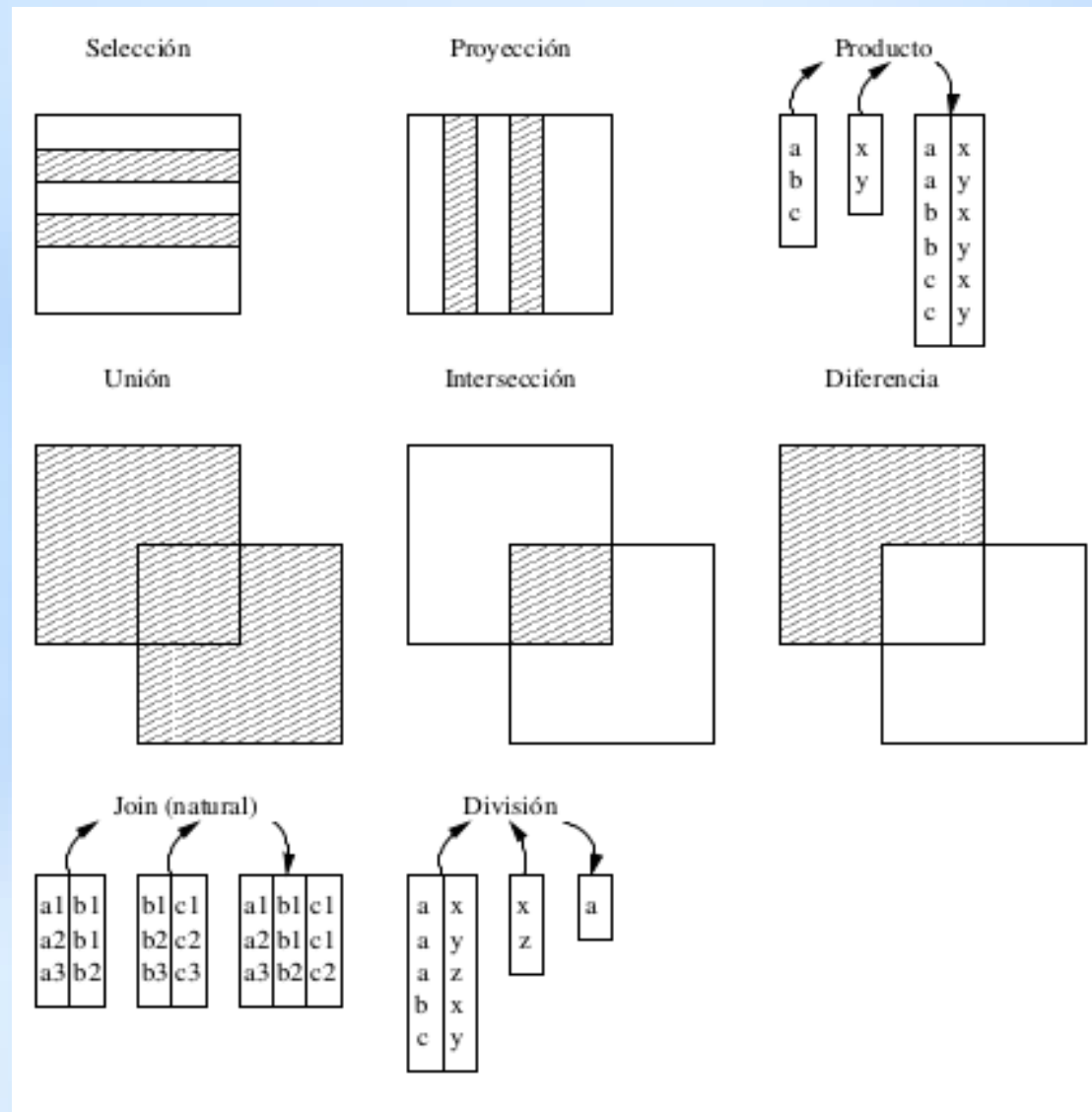
- Se dice que el operador  $O$  se aplica a la variable de relación (esquema) y que en el momento de aplicarlo tendrá un cierto valor (ocurrencia)



# Álgebra relacional

- Codd definió ocho operadores divididos en dos grupos de cuatro:
  - Los operadores conjuntistas tradicionales:
    - Unión
    - Intersección
    - Diferencia
    - Producto cartesiano
  - Los operadores relacionales especiales:
    - Selección
    - Proyección
    - Producto natural
    - División

# Álgebra relacional



# Álgebra relacional

- Clasificación de operadores:
  - Operadores *fundamentales*: son esenciales y no pueden obtenerse a partir de otros
    - Sin ellos, el álgebra relacional no sería un lenguaje completo
  - Estos operadores fundamentales son:
    - Proyección
    - Selección
    - Producto cartesiano
    - Unión de conjuntos
    - Diferencia de conjuntos

# Álgebra relacional

- Clasificación de operadores:
  - Los operadores, atendiendo al número de relaciones sobre las que se aplican, se divide en:
    - Operadores *unarios*
      - Proyección
      - Selección
    - Operadores *binarios*
      - Producto cartesiano
      - Unión de conjuntos
      - Diferencia de conjuntos

# Álgebra relacional

- Clasificación de operadores:
  - Operadores *adicionales*: son operadores binarios que pueden simplificar las expresiones
    - Las operaciones que ellos representan siempre pueden ser expresadas por combinación de operadores fundamentales
  - Estos operadores son:
    - Intersección de conjuntos
    - Productos de reunión: producto theta, producto natural, autounión, uniones externas, ...
    - División (cuantificador existencial)

# Operadores unarios

- Se aplican a una sola relación
- Sea  $R(A)$  un esquema de relación con el conjunto de atributos  $A$  definidos sobre el conjunto de dominios  $D$ :

$$R(A) = R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$$

- la relación  $r(R)$ , definida sobre el esquema  $R$ , de grado  $n$  y cardinalidad  $m$  estará constituida por un conjunto de  $m$  tuplas:

$$r(R) = \{t_i\}_{i=1}^m$$

- Donde

$$t_i = \{v_{i1}, v_{i2}, \dots, v_{in} \mid v_{ij} \in D_j\}$$

# Proyección

- Operación: el operador de *Proyección* selecciona ciertos atributos del esquema de R para obtener R'
- Si X es un subconjunto estricto y no vacío del conjunto de atributos A de la relación r(R), el resultado de esta operación será otra relación r' cuyo esquema (R') estará formado por el conjunto de atributos X, es decir:

$$r'(R') = \{t_i \in r(R(X)) \mid X \subset A \wedge X \neq \emptyset\}$$

- Trabaja sobre atributos (la parte *estática* de R)

# Proyección

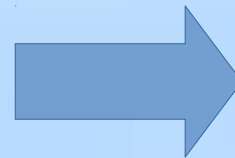
- Representación ( $\pi$ ): los atributos se escriben separados por comas, de izquierda a derecha en el mismo orden en que se desean

$\Pi_{\text{atributos}}(\text{relación})$

- Ejemplo: «obtener los apellidos de los autores»

$\Pi_{\text{Apellido}}(\text{Autores})$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Apellido
Stephenson
Lessig
Chomsky



# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado (~núm. de columnas) de  $R'$ ?

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado ( $\sim$  núm. de columnas) de  $R'$ ?
  - $n'$  es igual al número de elementos en la serie de *atributos* de  $\Pi$

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado ( $\sim$  núm. de columnas) de  $R'$ ?
  - $n'$  es igual al número de elementos en la serie de *atributos* de  $\Pi$
  - ¿y  $n'$  respecto a  $n$ ?

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado ( $\sim$  núm. de columnas) de  $R'$ ?
  - $n'$  es igual al número de elementos en la serie de *atributos* de  $\Pi$
  - $1 \leq n' \leq n$ 
    - Esta operación no puede incrementar el grado de  $R$
    - ¿y tiene sentido que no lo decrezca?

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado ( $\sim$  núm. de columnas) de  $R'$ ?
  - $n'$  es igual al número de elementos en la serie de *atributos* de  $\Pi$
  - $1 \leq n' \leq n$ 
    - Esta operación no puede incrementar el grado de  $R$
    - ¿y tiene sentido que no lo decrezca? NO, porque la lista de atributos es estática (la conozco antes de hacer la  $\Pi$ ), es absurdo pedir todos los atributos que sé que  $R$  tiene

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La proyección  $\Pi$  selecciona atributos, por lo que no afecta al número de tuplas  $\rightarrow m' = m$ 
    - ¿Seguro?

# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La proyección  $\Pi$  selecciona atributos, por lo que no afecta al número de tuplas  $\rightarrow m' = m$ 
    - Si tengo una relación de personas y mediante proyección selecciono el atributo “casado”, ¿cuál es el máximo posible?



# Proyección

- Operación:  $\Pi_{\text{atributos}}(\text{relación})$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La proyección  $\Pi$  selecciona atributos, por lo que no afecta al número de tuplas  $\rightarrow m' = m$ 
    - Si tengo una relación de personas y mediante proyección selecciono el atributo “casado”, ¿cuál es el máximo posible?
      - 2: verdadero o falso (lo impone el dominio)

# Proyección

- Pero si en una tabla de alumnos proyecto el atributo “medio de acceso”:
  - ¿cuál es el máximo de tuplas del resultado?

# Proyección

- Pero si en una tabla de alumnos proyecto el atributo “medio de acceso”:
  - Es 4 (si sólo hay 4 medios de acceso a la universidad: selectividad, CFGS, mayores de 25 años y extranjeros)
    - Lo impone el contexto
  - ¿Puedo en una relación con un sólo atributo (medio de acceso) tener más de 4 tuplas (filas)?
    - No, porque las tuplas forman un conjunto (por lo que no admiten repeticiones)

# Proyección

- Entonces ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R$  y  $R'$ ?

# Proyección

- Entonces ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R$  y  $R'$ ?

$$m' \leq m$$

- Esta operación no incrementa la cardinalidad de  $R$

# Proyección

- Y ¿en qué caso  $m'$  llegaría al máximo, esto es  $m' = m$ ?

# Proyección

- Y ¿en qué caso  $m'$  llegaría al máximo, esto es  $m' = m$ ?
  - Si  $X$  es *superclave* de  $R$  (o contiene un atributo UNIQUE)

# Proyección

- Y en otro caso, ¿cuál sería el máximo si se proyectan 2 atributos?



# Proyección

- Y en otro caso, ¿cuál sería el máximo si se proyectan 2 atributos?
  - Por ejemplo sexo y provincia española:

# Proyección

- Y en otro caso, ¿cuál sería el máximo si se proyectan 2 atributos?
  - Por ejemplo sexo y provincia española:
    - Serían  $2 * 50$  (hombre y mujer de cada provincia)

# Proyección

- Y en otro caso, ¿cuál sería el máximo si se proyectan 2 atributos?
  - Por ejemplo sexo y provincia española:
    - Serían  $2 * 50$  (hombre y mujer de cada provincia)
  - Pero si la BD sólo es para Andalucía ¿cuánto sería el máximo?

# Proyección

- Y en otro caso, ¿cuál sería el máximo si se proyectan 2 atributos?
  - Por ejemplo sexo y provincia española:
    - Serían  $2 * 50$  (hombre y mujer de cada provincia)
  - Pero si la BD sólo es para Andalucía ¿cuánto sería el máximo?
    - $2 * 8$  (hombre y mujer de cada provincia de Andalucía)

# Proyección

- Y en caso general, ¿cuál sería el máximo si se proyectan  $n$  atributos?

# Proyección

- Y en caso general, ¿cuál sería el máximo si se proyectan  $n$  atributos?
  - Todas las combinaciones de los valores posibles de cada atributo seleccionado ( $m_1 * m_2 * \dots * m_n$ )
  - Si consideramos NULL un valor aceptable más en el campo 1, sería  $((m_1+1) * m_2 * \dots * m_n)$

# Proyección

- ¿Y cuál sería el mínimo?

# Proyección

- ¿Y cuál sería el mínimo?
  - Un sólo valor (que podría incluso ser NULL)



# Proyección

- ¿Hemos visto la proyección en SQL?

# Proyección

- ¿Hemos visto la proyección en SQL?
  - Sí, es sencilla, indicando las columnas en el SELECT
    - Ej: `SELECT nombre, apellidos FROM clientes;`
    - ¿Y elimina los duplicados?

# Proyección

- ¿Hemos visto la proyección en SQL?
  - Sí, es sencilla, indicando las columnas en el SELECT
    - Ej: SELECT nombre, apellidos FROM clientes;
    - ¿Y elimina los duplicados?
      - Según se le indique (DISTINCT)

# Proyección

- ¿Hemos visto la proyección en SQL?
  - Sí, es sencilla, indicando las columnas en el SELECT
    - Ej: SELECT nombre, apellidos FROM clientes;
    - ¿Y elimina los duplicados?
      - Según se le indique (DISTINCT)
  - ¿Por qué opera así nuestro SGBD?
    - ¿Despiste de los desarrolladores? ¿Es difícil de controlar? ...

# Proyección

- ¿Hemos visto la proyección en SQL?
  - Sí, es sencilla, indicando las columnas en el SELECT
    - Ej: SELECT nombre, apellidos FROM clientes;
    - ¿Y elimina los duplicados?
      - Según se le indique (DISTINCT)
  - ¿Por qué opera así nuestro SGBD?
    - Para dar más libertad al usuario: se permite tratar el resultado como conjunto o como una lista
    - El SGBD implementa un modelo que viola las leyes de Codd: si quiero mantener la coherencia con el modelo teórico/matemático tengo que hacer un uso controlado de las herramientas que ofrece

# Selección

- Operación: selecciona las tuplas que cumple un determinado criterio de selección
- Sea la relación  $r(R)$ , si le aplicamos un criterio de selección  $p$ , obtendremos la relación  $r'(R)$ :

$$r'(R) = \{t_i \in r(R) \mid p(t_i) = \text{“cierto”}\}$$

- Representación: (símbolo *sigma*)

$$\sigma_{\text{criterio-de-selección}}(R)$$

- Trabaja sobre tuplas (la parte *dinámica* de  $R$ )

# Selección

- Representación:  $\sigma_{\text{criterio-de-selección}}(R)$
- Evidentemente el predicado/criterio de selección debe ser computable, no se aceptan:
  - Predicados sobre sucesos futuros
  - Sucesos indecidibles (ej: *el programa p para*)
  - ...
- En los criterios de selección se aplican los operadores aritméticos, de comparación y lógicos a constantes y atributos (formando expresiones sobre las que operar a su vez)

# Selección

- Ejemplo: «obtener todos los datos del autor cuyo código es igual a tres»

<u>Cód_Aut</u>	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

$\sigma_{\text{Cód\_Aut}=3}(\text{Autores})$

<u>Cód_Aut</u>	Nombre	Apellido
3	Noam	Chomsky



# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado (~núm. de columnas) de  $R'$ ?
  - El grado será el mismo para  $R$  y  $R'$  ¿siempre?

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $n'$ , el grado (~núm. de columnas) de  $R'$ ?
  - El grado será el mismo para  $R$  y  $R'$  ¿siempre?
    - Sí, siempre ;)

$$n' = n$$

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La cardinalidad cumple:  $m' \leq m$  (estamos *seleccionando tuplas*)

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La cardinalidad cumple:  $m' \leq m$  (estamos *seleccionando tuplas*)
  - ¿En qué caso  $m=m'$ ?

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La cardinalidad cumple:  $m' \leq m$  (estamos *seleccionando tuplas*)
  - ¿En qué caso  $m=m'$ ?
    - Cuando el predicado se cumpla para toda fila
    - O sea siempre cierto (Ej: tautología  $2>1$ )

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La cardinalidad cumple:  $m' \leq m$  (estamos *seleccionando tuplas*)
  - ¿Tiene sentido el caso  $m=m'$ ? Porque con la proyección era raro ...

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- ¿Qué podemos afirmar sobre  $m'$ , la cardinalidad ( $\sim$  núm. de filas) de  $R'$ ?
  - La cardinalidad cumple:  $m' \leq m$  (estamos *seleccionando tuplas*)
  - ¿Tiene sentido el caso  $m=m'$ ? Sí
    - En la proyección era raro porque los atributos son estáticos y conocidos, pero los valores concretos son dinámicos en el tiempo (cambian en su dominio)



# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- No obstante, se cumple siempre que
$$r'(R') \subseteq r(R)$$
- ¿Y podría ser vacío el mínimo?

# Selección

- Operación:  $\sigma_{\text{criterio-de-selección}}(R)$ 
  - Sean  $n$  el grado y  $m$  la cardinalidad de  $r(R)$
  - Sean  $n'$  el grado y  $m'$  la cardinalidad de  $r'(R')$
- No obstante, se cumple siempre que
$$r'(R') \subseteq r(R)$$
- ¿Y podría ser vacío el mínimo?
  - Cuando el predicado nunca sea cierto (Ej:  $\text{edad} < 0$ )
  - Cuando el predicado no se cumpla para ninguna tupla
    - Y es un resultado totalmente válido

# Proyección y selección

- Acción: normalmente las operaciones de proyección y selección aparecen combinadas en los SGBD.
- Ejemplo: «obtener el nombre y el apellido del autor(es) cuyo código es igual a tres».

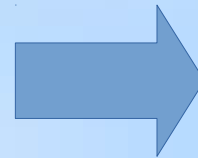
Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

# Proyección y selección

- Acción: normalmente las operaciones de proyección y selección aparecen combinadas en los SGBD.
- Ejemplo: «obtener el nombre y el apellido del autor(es) cuyo código es igual a tres».

$\Pi_{\text{Nombre,Apellido}}(\sigma_{\text{Cód\_Aut}=3}(\text{Autores}))$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Nombre	Apellido
Noam	Chomsky

# Proyección y selección

- ¿Da igual resultado proyectar y seleccionar que seleccionar y proyectar?

$\Pi_{\text{Nombre,Apellido}}(\sigma_{\text{Apellido=Palomo}}(\text{Autores}))$

$\sigma_{\text{Apellido=Palomo}}(\Pi_{\text{Nombre,Apellido}}(\text{Autores}))$

# Proyección y selección

- ¿Da igual resultado proyectar y seleccionar que seleccionar y proyectar?

$$\Pi_{\text{Nombre,Apellido}}(\sigma_{\text{Apellido=Palomo}}(\text{Autores}))$$

$$\sigma_{\text{Apellido=Palomo}}(\Pi_{\text{Nombre,Apellido}}(\text{Autores}))$$

- Parece que sí, pero cuidado si la proyección elimina campos usados en la selección

$$\Pi_{\text{Nombre,Cód_Aut}}(\sigma_{\text{Apellido=Palomo}}(\text{Autores}))$$

$$\sigma_{\text{Apellido=Palomo}}(\Pi_{\text{Nombre,Cód_Aut}}(\text{Autores}))$$

# Asignación y renombrado

- Acción: nos permite cambiar el nombre de una relación, o bien, cambiar el nombre de los atributos para hacer que su esquema sea idéntico al de otra relación con objeto de poder realizar ciertas operaciones, como las de conjuntos
- Representación:

$$R' \leftarrow R$$

- Ejemplo: cambiar el nombre del atributo *Nombre* de la relación Autores por *Nombre\_pila* y que, además, la relación se denomine *Autores1*

$\text{Autores1}(\text{Cód\_Aut}, \text{Nombre\_pila}, \text{Apellido}) \leftarrow \text{Autores}(\text{Cód\_Aut}, \text{Nombre}, \text{Apellido})$

$\text{Existe Autores1}(\text{Cód\_Aut}, \text{Nombre\_pila}, \text{Apellido})$

# Asignación y renombrado

- Esta acción es MUY IMPORTANTE en los ejercicios
  - Sobre todo en el examen
  - ¿Por qué? (todavía no lo hemos explicado, pero ...)



# Asignación y renombrado

- Esta acción es MUY IMPORTANTE en los ejercicios
  - Sobre todo en el examen
  - Porque es la única forma de indicar los atributos sobre los que se hacen operaciones entre tablas (salvo que coincidan los nombres originales de los atributos)
    - O indicar aquellos por los que no ...

# Operadores binarios

- Estos operadores se aplican a dos relaciones
- En algunos casos se exige que las relaciones tengan idéntico conjunto de atributos y en el mismo orden en sus esquemas
- Dados: dos esquemas

$$R_1 (A1_i : D1_i) \text{ y } R_2 (A2_j : D2_j)$$

son idénticos a efectos de dichos operadores, si ambos están definidos sobre el mismo conjunto de dominios, y se cumple que:

$$\forall A1_i \exists A2_j \mid \text{dom}(A1_i) = \text{dom}(A2_j) \wedge$$

$$\forall A2_j \exists A1_i \mid \text{dom}(A2_j) = \text{dom}(A1_i)$$

# Producto cartesiano

- Operación: sean las relaciones  $r_1$  y  $r_2$  con esquemas  $R_1$  y  $R_2$ , grado  $n_1$  y  $n_2$  y cardinalidad  $m_1$  y  $m_2$ , respectivamente. El producto cartesiano es otra relación  $r$  de grado  $n = n_1 + n_2$  cuyo esquema  $R$  estará formado por los  $n_1 + n_2$  atributos de  $R_1 \cup R_2$  es decir:

$$R(A1_i : D1_i, \dots, A1_{n1} : D1_{n1}, A2_i : D2_i, \dots, A2_{n2} : D2_{n2})$$

$$R_1 (A1_i : D1_i, \dots, A1_{n1} : D1_{n1})$$

$$R_2 (A2_i : D2_i, \dots, A2_{n2} : D2_{n2})$$

- El resultado es el producto cartesiano matem.: por cada elemento de uno de los conjuntos se incluyen todas las posibles combinaciones con los elementos del otro

# Producto cartesiano

- Representación: se utiliza el mismo símbolo que en matemáticas

$$R_1 \times R_2$$

- A veces se llama *producto* a secas
- ¿Qué cardinalidad tiene el resultado?

# Producto cartesiano

- Representación: se utiliza el mismo símbolo que en matemáticas

$$R_1 \times R_2$$

- A veces se llama *producto* a secas
- ¿Qué cardinalidad tiene el resultado?

La cardinalidad total será  $m_1 * m_2$ , expresada de la siguiente manera:

$$\{V1_{i1}, \dots, V1_{in1}, V2_{j1}, \dots, V2_{jn2} \mid \forall i \forall j (V1_{i1}, \dots, V1_{in1} \in r_1 \wedge V2_{j1}, \dots, V2_{jn2} \in r_2)\}$$

# Producto cartesiano

- Ejemplo: obtener una relación con los datos de los socios y los libros

## Socios × Libros

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García

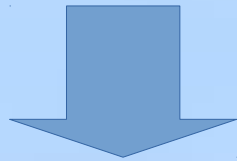
Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	2	3

# Producto cartesiano

Socios × Libros

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	2	3



Cód_Soc	Socios. Nombre	Apellido	Cód_Lib	Libros. Nombre	Cód_Edit	Cód_Aut
1	Eric	Manrique	1	SSOO	1	1
1	Eric	Manrique	2	BBDD	2	2
1	Eric	Manrique	3	ABBDD	2	3
2	Mar	García	1	SSOO	1	1
2	Mar	García	2	BBDD	2	2
2	Mar	García	3	ABBDD	2	3

# Producto cartesiano

- En concreto, ¿qué significado tiene este producto cartesiano en el ejemplo?

Cód_Soc	Socios. Nombre	Apellido	Cód_Lib	Libros. Nombre	Cód_Edit	Cód_Aut
1	Eric	Manrique	1	SSOO	1	1
1	Eric	Manrique	2	BBDD	2	2
1	Eric	Manrique	3	ABBDD	2	3
2	Mar	García	1	SSOO	1	1
2	Mar	García	2	BBDD	2	2
2	Mar	García	3	ABBDD	2	3



# Producto cartesiano

- En concreto, ¿qué significado tiene este producto cartesiano en el ejemplo?
  - Son todas las lecturas posibles: que todo socio consulte todo libro
  - Normalmente la realidad en un momento dado suele ser un subconjunto del producto cartesiano

Cód_Soc	Socios. Nombre	Apellido	Cód_Lib	Libros. Nombre	Cód_Edit	Cód_Aut
1	Eric	Manrique	1	SSOO	1	1
1	Eric	Manrique	2	BBDD	2	2
1	Eric	Manrique	3	ABBDD	2	3
2	Mar	García	1	SSOO	1	1
2	Mar	García	2	BBDD	2	2
2	Mar	García	3	ABBDD	2	3

# Unión de conjuntos

- Operación: dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1 = R_2$  se pueden unir en una sola relación con el mismo esquema ( $R$ ) y cuya extensión está formada por el conjunto de tuplas ( $t_i$ ) que pertenecen a  $r_1$  o a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \vee t_i \in r_2\}$$

- ¿Qué condiciones deben cumplir ...?
  - Los grados
  - Los dominios

# Unión de conjuntos

- Operación: dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1 = R_2$  se pueden unir en una sola relación con el mismo esquema ( $R$ ) y cuya extensión está formada por el conjunto de tuplas ( $t_i$ ) que pertenecen a  $r_1$  o a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \vee t_i \in r_2\}$$

- Condiciones: se tienen que cumplir:
  - Las relaciones  $R_1$  y  $R_2$  deben tener el mismo número de atributos, o sea, ser del mismo grado
  - El dominio del atributo  $i$ -ésimo de  $r_1$  y de  $r_2$  debe ser el mismo

# Unión de conjuntos

- Representación: se utilizan el mismo símbolo que para la unión de conjuntos.

$$r_1 \cup r_2$$

- ¿Qué grado y cardinalidad tendrá el resultado?

# Unión de conjuntos

- Representación: se utilizan el mismo símbolo que para la unión de conjuntos.

$$r_1 \cup r_2$$

- ¿Qué grado y cardinalidad tendrá el resultado?
  - Grado es el mismo que  $r_1$  y  $r_2$

# Unión de conjuntos

- Representación: se utilizan el mismo símbolo que para la unión de conjuntos.

$$r_1 \cup r_2$$

- ¿Qué grado y cardinalidad tendrá el resultado?
  - Grado es el mismo que  $r_1$  y  $r_2$
  - Cardinalidad es menor o igual que la suma de las cardinalidades de  $r_1$  y  $r_2$ , porque si hay repeticiones se eliminan del resultado
  - Y la cardinalidad será igual o mayor al máximo de la cardinalidad de  $r_1$  y de  $r_2$

# Unión de conjuntos

- Ejemplo: obtener una relación que contenga a todos los autores y a todos los editores

Autores  $\cup$  Editores

- ¿Qué resultado dará?

# Unión de conjuntos

- Ejemplo: obtener una relación que contenga a todos los autores y a todos los editores

Autores U Editores

- ¿Qué resultado dará?

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig



# Unión de conjuntos

- Hemos exigido que  $R_1$  y  $R_2$  tengan el mismo número de atributos y con dominios  $i$ -ésimos iguales. Pero no que sus atributos se llamen igual: ¿qué nombre tendrán en el resultado?
  - Convención: por defecto el de la primera relación (a la izquierda del  $\cup$ )
  - Si se hace una asignación, el de la nueva relación
- Propiedades:
  - Conmutativa:  $r_1 \cup r_2 = r_2 \cup r_1$
  - Asociativa:  $(r_1 \cup r_2) \cup r_3 = r_1 \cup (r_2 \cup r_3)$

# Unión de conjuntos

- Ejemplo: obtener una relación que contenga a todos los autores y a todos los editores

Autores U Editores

- ¿Podemos evitar depender de esa convención?
  - Es decir, igualar sus esquemas:
    - Autores (Cód\_Aut, Nombre, Apellido)
    - Editores (Cód\_Edit, Nombre, Apellido)

# Unión de conjuntos

- Ejemplo: obtener una relación que contenga a todos los autores y a todos los editores

Autores U Editores

- ¿Podemos evitar depender de esa convención?
  - Es decir, igualar sus esquemas:
    - Autores (Cód\_Aut, Nombre, Apellido)
    - Editores (Cód\_Edit, Nombre, Apellido)
  - Solución: renombrar

Editores(Cód\_Aut, Nombre\_pila, Apellido) ← Editores(Cód\_Edit,  
Nombre, Apellido)

# Unión de conjuntos

- Ejemplo: obtener una relación que contenga a todos los autores y a todos los editores

Autores U Editores

- ¿Qué resultado dará?

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

# Diferencia de conjuntos

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1$  y  $R_2$ , podemos obtener otra relación definida sobre dicho esquema ( $R$ ) y cuya extensión estará constituida por el conjunto de tuplas ( $t_i$ ) que pertenezcan a  $r_1$  pero no a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \wedge t_i \notin r_2\}$$

- Representación: el símbolo (  $-$  )
  - Cuidado porque a veces en matemática discreta se usa  $\setminus$

$$r_1 - r_2$$

- ¿Qué condiciones se exigen?

# Diferencia de conjuntos

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1$  y  $R_2$ , podemos obtener otra relación definida sobre dicho esquema ( $R$ ) y cuya extensión estará constituida por el conjunto de tuplas ( $t_i$ ) que pertenezcan a  $r_1$  pero no a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \wedge t_i \notin r_2\}$$

- Representación: el símbolo (  $-$  )
  - Cuidado porque a veces en matemática discreta se usa  $\setminus$

$$r_1 - r_2$$

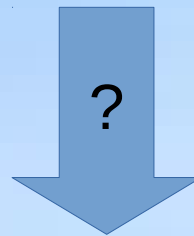
- Condiciones: las mismas que para la unión

# Diferencia de conjuntos

- Ejemplo: encontrar a todos los autores que no sean editores

## Autores – Editores

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

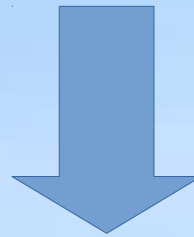
Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
3	Noam	Chomsky

# Diferencia de conjuntos

- Ejemplo: encontrar a todos los autores que no sean editores

## Autores – Editores

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



# Diferencia de conjuntos

- Ejemplo: encontrar a todos los autores que no sean editores

## Autores – Editores

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

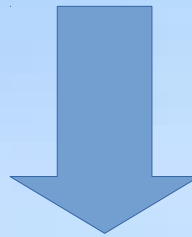
- Problema: una persona puede aparecer en cada relación con un código distinto

# Diferencia de conjuntos

- Solución: encontrar a todos los autores que no sean editores

$$(\Pi_{\text{nombre,apellido}}(\text{Autores})) - (\Pi_{\text{nombre,apellido}}(\text{Editores}))$$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

Nombre	Apellido
Neal	Stephenson
Noam	Chomsky

# Diferencia de conjuntos

- Ejemplo: encontrar a todos los autores que no sean editores

$$(\Pi_{\text{nombre,apellido}}(\text{Autores})) - (\Pi_{\text{nombre,apellido}}(\text{Editores}))$$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig

- Problema: puede haber dos personas distintas con igual nombre y apellido/s (sobre todo si sólo tienen un apellido)

# Diferencia de conjuntos

- Ejemplo: encontrar a todos los autores que no sean editores

$$(\Pi_{\text{nombre,apellido}}(\text{Autores})) - (\Pi_{\text{nombre,apellido}}(\text{Editores}))$$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig

- Problema: puede haber dos personas distintas con igual nombre y apellido/s (sobre todo si sólo tienen un apellido)
  - Entonces la solución válida es la inicial

# Diferencia de conjuntos

- Ahora bien, ¿sería un buen diseño?

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig

# Diferencia de conjuntos

- Ahora bien, ¿sería un buen diseño?
  - Evidentemente no: si cambia de nombre (porque cambia de estado civil) hay que actualizar dos tablas. Igual si se borra ...
  - ¿Solución?

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig

# Diferencia de conjuntos

- Ahora bien, ¿sería un buen diseño?
  - Evidentemente no: si cambia de nombre (porque cambia de estado civil) hay que actualizar dos tablas. Igual si se borra ...
  - Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig
4	Lawrence	Lessig



Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE



# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
- Y ahora ¿cómo encuentro a todos los autores que no sean editores?

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
- Y ahora ¿cómo encuentro a todos los autores que no sean editores?

$\sigma_{(\text{Autor}=\text{TRUE}) \text{ AND } (\text{Editor}=\text{FALSE})}(\text{Personas})$

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
  - ¿De qué estructura de E/R puede provenir este tipo de relación?

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
  - ¿De qué estructura de E/R puede provenir este tipo de relación?
    - De una especialización con solapamiento

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
  - ¿De qué estructura de E/R puede provenir este tipo de relación?
    - De una especialización con solapamiento
    - ¿Y si la especialización fuera disjunta o con completitud total?

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución: unir en una sola relación y poner un booleano por cada rol que pueda tener
  - ¿De qué estructura de E/R puede provenir este tipo de relación?
    - De una especialización con solapamiento
    - ¿Y si la especialización fuera disjunta o con completitud total? Probablemente también sería mejor opción (aunque requiriera control externo con elementos del SGBD o aplicaciones)

Código	Nombre	Apellido	Autor	Editor
1	Neal	Stephenson	TRUE	FALSE
2	Lawrence	Lessig	TRUE	TRUE
3	Noam	Chomsky	TRUE	FALSE
4	Santiago	Ceri	FALSE	TRUE
5	Antón	De Miguel	FALSE	TRUE
6	Lawrence	Lessig	FALSE	TRUE

# Diferencia de conjuntos

- Solución alternativa: ¿podría ahorrarme los booleanos y sacar quien es autor y/o editor viendo en la tabla libros?

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
4	Santiago	Ceri
5	Antón	De Miguel
6	Lawrence	Lessig

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Diferencia de conjuntos

- Solución alternativa: ¿podría ahorrarme los booleanos y sacar quien es autor y/o editor viendo en la tabla libros?
  - Pregunta: ¿puedo tener dos claves foráneas que apunten a la misma CP de otra tabla?

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
4	Santiago	Ceri
5	Antón	De Miguel
6	Lawrence	Lessig

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



# Diferencia de conjuntos

- Solución alternativa: ¿podría ahorrarme los booleanos y sacar quien es autor y/o editor viendo en la tabla libros?
  - Pregunta: ¿puedo tener dos claves foráneas que apunten a la misma CP de otra tabla? Sí

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
4	Santiago	Ceri
5	Antón	De Miguel
6	Lawrence	Lessig

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Diferencia de conjuntos

- Solución alternativa: ¿podría ahorrarme los booleanos y sacar quien es autor y/o editor viendo en la tabla libros?
  - Problema: sólo se considerarían autores o editores quienes tengan libros ya publicados
    - ¿Qué estructura sería en un diagrama E/R?

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
4	Santiago	Ceri
5	Antón	De Miguel
6	Lawrence	Lessig

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Diferencia de conjuntos

- Solución alternativa: ¿podría ahorrarme los booleanos y sacar quien es autor y/o editor viendo en la tabla libros?
  - Problema: sólo se considerarían autores o editores quienes tengan libros ya publicados
    - En E/R: particip. total con *Edita* y *Escribe* (en sub-tipos)

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky
4	Santiago	Ceri
5	Antón	De Miguel
6	Lawrence	Lessig

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Operadores adicionales

- Por ahora se han visto los operadores fundamentales
- Veremos los adicionales: simplifican las expresiones que se pueden formar utilizando los fundamentales
- Los operadores adicionales son:
  - Intersección de conjuntos
  - Productos de reunión:
    - Producto theta
    - Producto natural
    - Autounión
    - Unión externa
  - División

# Intersección de conjuntos

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1$  y  $R_2$ , podemos obtener otra relación definida sobre el mismo esquema de relación ( $R$ ) y cuya extensión estará constituida por el conjunto de tuplas ( $t_i$ ) que pertenezcan a  $r_1$  y a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \wedge t_i \in r_2\}$$

- Representación: se utilizan el mismo símbolo que para la intersección de conjuntos

$$r_1 \cap r_2$$

- ¿Condiciones?

# Intersección de conjuntos

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas idénticos  $R_1$  y  $R_2$ , podemos obtener otra relación definida sobre el mismo esquema de relación ( $R$ ) y cuya extensión estará constituida por el conjunto de tuplas ( $t_i$ ) que pertenezcan a  $r_1$  y a  $r_2$

$$r(R) = \{t_i \mid t_i \in r_1 \wedge t_i \in r_2\}$$

- Representación: se utilizan el mismo símbolo que para la intersección de conjuntos

$$r_1 \cap r_2$$

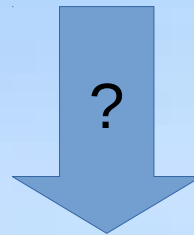
- Condiciones: las mismas que para la unión

# Intersección de conjuntos

- Ejemplo: encontrar a todos los autores que también sean editores

**Autores  $\cap$  Editores**

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

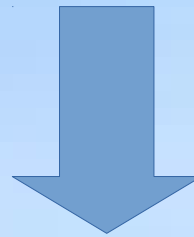
Código	Nombre	Apellido
2	Lawrence	Lessig

# Intersección de conjuntos

- Ejemplo: encontrar a todos los autores que también sean editores

**Autores  $\cap$  Editores**

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

Código	Nombre	Apellido
--------	--------	----------

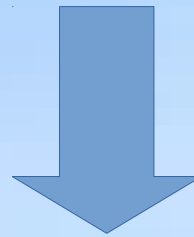


# Intersección de conjuntos

- Ejemplo: encontrar a todos los autores que también sean editores

$$(\Pi_{\text{nombre,apellido}}(\text{Autores})) \cap (\Pi_{\text{nombre,apellido}}(\text{Editores}))$$

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

Nombre	Apellido
Lawrence	Lessig

# Productos de reunión

- La operación de *reunión* es una de las más usadas en el álgebra relacional: combina en una sola operación información de varias relaciones
- Se puede definir como una secuencia de operaciones: producto cartesiano seguido de una selección y proyección
  - Permite elegir mediante las condiciones lógicas que deseemos elementos concretos de un producto cartesiano (que produce todas las combinaciones posibles, no las reales para mi problema)

# Productos de reunión

- La relación resultante de una operación de reunión tiene una cardinalidad menor o igual que la de un producto cartesiano
- Existen diferentes tipos de productos de reunión:
  - Producto theta (*join*)
  - Producto natural (*natural join*)
  - Autounión (*auto join*)
  - Unión externa (*outer join*)

# Producto theta

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas  $R_1$  y  $R_2$  y una condición de selección ( $\theta$ ) formada por atributos del esquema  $R_1 \cup R_2$ , da una relación  $R$  constituida por el conjunto de tuplas  $t_i$ , del producto cartesiano de las relaciones que cumplen el criterio de selección
- Representación: se representa por el símbolo del producto de reunión y su subíndice indica (aparte si se desea) el criterio de selección

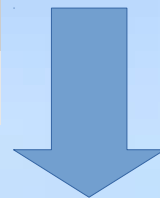
$$R_1 \bowtie_{\theta} R_2, \theta = \textit{condición}$$

# Producto theta

- Ejemplo: obtener una relación con los datos de los socios y los libros y donde el apellido del socio sea Manrique

Socios  $\bowtie_{\theta}$  Libros,  $\theta = (\text{Apellido} = \text{'Manrique'})$

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García



Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

Cód_Soc	Socios.N ombre	Apellido	Cód_Lib	Libros.N ombre	Cód_Edit	Cód_Aut
1	Eric	Manrique	1	SSOO	1	1
1	Eric	Manrique	2	BBDD	2	2
1	Eric	Manrique	3	ABBDD	3	2

# Producto natural

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas  $R_1$  y  $R_2$ , podemos obtener otra relación  $R$  definida sobre el esquema de relación  $R_1 \cup R_2$  y constituida por el conjunto de tuplas  $t_i$  tal que

$$r(R) = \{t_i \mid t_i \in (\Pi_{R_1 \cup R_2} (\sigma_{\theta} (r_1 \times r_2)))\}$$

$$\theta = (r_1.A_1 = r_2.A_1 \wedge r_1.A_2 = r_2.A_2 \wedge \dots r_1.A_n = r_2.A_n)$$

$$A_1, A_2, \dots, A_n = R_1 \cap R_2$$

- Representación: se representa por el símbolo del producto de reunión

$$r_1 \bowtie r_2$$

# Producto natural

- Consideraciones: si

$$R_1 \cap R_2 = \emptyset \Rightarrow r_1 \bowtie r_2 = r_1 \times r_2$$

- El producto cartesiano es un tipo de producto natural
  - En concreto, el producto natural de dos relaciones que no comparten atributos de su esquema

# Producto natural

- Consideraciones: si

$$R_1 \cap R_2 = \emptyset \Rightarrow r_1 \bowtie r_2 = r_1 \times r_2$$

- El producto cartesiano es un tipo de producto natural
  - En concreto, el producto natural de dos relaciones que no comparten atributos de su esquema
- Pregunta: ¿y si tienen un atributo de igual nombre pero distinto dominio?



# Producto natural

- Consideraciones: si

$$R_1 \cap R_2 = \emptyset \Rightarrow r_1 \bowtie r_2 = r_1 \times r_2$$

- El producto cartesiano es un tipo de producto natural
  - En concreto, el producto natural de dos relaciones que no comparten atributos de su esquema
- Pregunta: ¿y si tienen un atributo de igual nombre pero distinto dominio?
  - Da error la operación, no procede

# Producto natural

- La selección se realiza sobre la intersección de atributos
  - Los atributos que se usan de criterio de producto (aquellos con igual nombre) sólo aparecen una vez en el resultado
- Es importante usar el renombrado para que:
  - Poner igual nombre a los atributos sobre los que se hará la selección
  - Poner distinto nombre a los atributos que no proceda

# Producto natural

- El producto natural es una restricción del producto theta en que se evita decir que seleccione las tuplas en que los campos de igual nombre tengan igual valor
  - Restringe los campos
  - No considera otro operador que no sea el igual (=)
  - No permite usar constantes, etc
- Se usa muchísimo (claves foráneas)
- Las tuplas con NULL en los campos comunes se descartan del resultado

# Producto natural

- ¿Qué cardinalidad tendrá un producto natural?

# Producto natural

- ¿Qué cardinalidad tendrá un producto natural?
  - Como mínimo cero elementos (no hay concordancia entre los valores de los campos comparados)

# Producto natural

- ¿Qué cardinalidad tendrá un producto natural?
  - Como mínimo cero elementos (no hay concordancia entre los valores de los campos comparados)
  - Como máximo la misma que el producto cartesiano ( $m_1 * m_2$ )
- El ratio entre la cardinalidad de un producto natural y su máximo posible es la *selectividad del producto* (*join selectivity*)
- Se pueden concatenar varios prod. naturales

# Producto natural

- Ejemplo: obtener los datos de los socios que tienen un libro en préstamo así como el código del libro y la fecha del préstamo

Socios ⋈ Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García

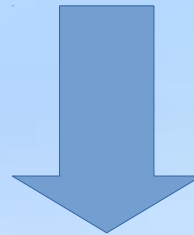
Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

# Producto natural

- Ejemplo: Socios ⋈ Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017



Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017



# Autounión

- Operación: es un producto theta de una tabla sobre sí misma
- Da como resultado una relación en la que los (conjuntos de) campos coincidentes pertenecen a una misma tabla
- ¿No se puede hacer con una selección?

# Autounión

- Operación: es un producto theta de una tabla sobre sí misma
- Da como resultado una relación en la que los (conjuntos de) campos coincidentes pertenecen a una misma tabla
- ¿No se puede hacer con una selección?
  - No, porque las condiciones son sobre el contenido de una tupla individualmente (no respecto a otras)
- ¿Es esto algo común?

# Autounión

- Operación: es un producto theta de una tabla sobre sí misma
- Da como resultado una relación en la que los (conjuntos de) campos coincidentes pertenecen a una misma tabla
- ¿No se puede hacer con una selección?
  - No, porque las condiciones son sobre el contenido de una tupla individualmente (no respecto a otras)
- ¿Es esto algo común?
  - Ej.: alumnos que mentoriza cada alumno mentor

# Autounión

- La auto-unión está muy poco documentada en álgebra. Nosotros la implementaremos con un renombrado y producto theta

$\text{Alumnos1} \leftarrow \text{Alumnos}$

$R \leftarrow \text{Alumnos1} \bowtie_{\theta} \text{Alumnos}$

$\theta = (\text{Alumnos1.mentor} = \text{Alumnos.DNIe})$

# Autounión

- La auto-uniión ¿se puede hacer en SQL?

# Autounión

- La auto-uniión ¿se puede hacer en SQL?
- Sí, usando sinónimos/alias
- Ejemplo: Trabajadores que ganen más que su supervisor
  - Trabajadores(SegSoc, nombre, salario, supervisor)

# Autounión

- La auto-uni3n ¿se puede hacer en SQL?
- SÍ, usando sin3nimos/alias
- Ejemplo: Trabajadores que ganen m3s que su supervisor
  - Trabajadores(SegSoc, nombre, salario, supervisor)  
SELECT T.EmpName  
FROM Trabajadores T, Trabajadores SUP  
WHERE T.supervisor = SUP.SegSoc  
AND T.salario > SUP.salario
  - *Ejemplo tomado de “SQL: Structured Query Language Data Manipulation Language”*

# Unión externa

- Se define para solventar el problemas de los valores no emparejados en los productos. En concreto, se apoya en el uso de un valor  $\omega$  fuera de rango (que en SQL es NULL)
- Hay tres tipos:
  - Unión externa por la izquierda (left outer join  $\bowtie$ )
  - Unión externa por la derecha (right outer join  $\bowtie$ )
  - Unión externa completa o unión externa “a secas” (full outer join  $\bowtie$ )



# Unión externa

- Unión externa por la izquierda (left outer join  $\bowtie$ )
  - Incluye en el resultado las tuplas de la relación a la izquierda del símbolo ( $R_1$  si es  $R_1 \bowtie R_2$ ) que no tienen valor que concuerde, poniendo  $\omega$  como valor del campo de join

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

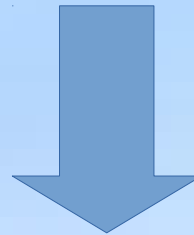
# Unión externa

- Unión externa por la izquierda (left outer join  $\bowtie$ )

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017



Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
8	Gregorio	Sánchez	$\omega$	$\omega$

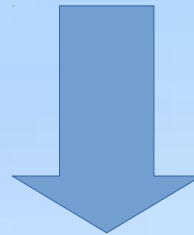
# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios ⋈ Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017



Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
8	Gregorio	Sánchez	ω	ω

# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios ⋈ Préstamos

- Listado de todos los socios, incluyendo los libros que han sacado en préstamo (hayan sacado o no libros)

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
8	Gregorio	Sánchez	ω	ω

# Unión externa

- Unión externa por la derecha (right outer join  $\bowtie$ )
  - Incluye en el resultado las tuplas de la relación a la derecha del símbolo ( $R_2$  si es  $R_1 \bowtie R_2$ ) que no tienen valor que concuerde, poniendo  $\omega$  como valor del campo de join

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez

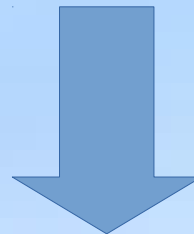
Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017
1	18	5/1/2017

# Unión externa

- Unión externa por derecha (right outer join ⋈)

Socios ⋈ Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez



Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017
1	18	5/1/2017

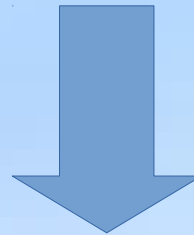
Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017

# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez



Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017
1	18	5/1/2017

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017

# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios  $\bowtie$  Préstamos

- Listado de todos los préstamos, indicando el socio que lo ha sacado

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017



# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios  $\bowtie$  Préstamos

- Listado de todos los préstamos, indicando el socio que lo ha sacado
- ¿Y qué indica el  $\omega$ ?

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	$\omega$	$\omega$	1	5/1/2017

# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios  $\bowtie$  Préstamos

- Listado de todos los préstamos, indicando el socio que lo ha sacado
- ¿Y qué indica el  $\omega$ ? Una inconsistencia (CF con valor de CP que no existe)

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	$\omega$	$\omega$	1	5/1/2017

# Unión externa

- ¿Tiene sentido este producto en este caso?

Socios  $\bowtie$  Préstamos

- Listado de todos los préstamos, indicando el socio que lo ha sacado
- ¿Y qué indica el  $\omega$ ? Una inconsistencia (violación de la regla de integridad referencial, *CF no en CP*)
  - Según contexto: dato erróneo, no actualizó en cascada...

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	$\omega$	$\omega$	1	5/1/2017

# Unión externa

- ¿Y qué indica el  $\omega$ ? Una inconsistencia (violación de la regla de integridad referencial, *CF no en CP*)
  - Según contexto: dato erróneo, no actualizó en cascada...
  - ¿Cómo podría solucionarlo?

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	$\omega$	$\omega$	1	5/1/2017

# Unión externa

- La inconsistencia (violación de la regla de integridad referencial, *CF no en CP*)
  - ¿Cómo podría solucionarla?
    - Dos alternativas

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017

# Unión externa

- La inconsistencia (violación de la regla de integridad referencial, *CF no en CP*)
  - ¿Cómo podría solucionarla?
    - Dos alternativas
      - Poniendo los Cod\_Soc de Préstamos a NULL
      - Poniendo todos los campos de Socios a NULL (excepto Cod\_Soc)

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017

# Unión externa

- La inconsistencia (violación de la regla de integridad referencial, *CF no en CP*)
  - ¿Cómo podría solucionarla?
    - Dos alternativas
      - Poniendo los Cod\_Soc de Préstamos a NULL
      - Poniendo todos los campos de Socios a NULL (excepto Cod\_Soc)
    - ¿Qué implicaciones tiene una u otra opción?

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
18	ω	ω	1	5/1/2017

# Unión externa

- Unión externa por la completa (outer join  $\bowtie$ )
  - Incluye en el resultado las tuplas de ambas relaciones ( $R_1$  y  $R_2$ ) que no tienen valor que concuerde, poniendo  $\omega$  como valor del campo de join

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017
1	18	5/1/2017

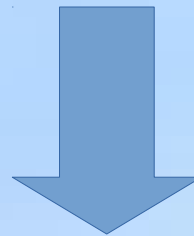


# Unión externa

- Unión externa por la completa (outer join  $\bowtie$ )

Socios  $\bowtie$  Préstamos

Cód_Soc	Nombre	Apellido
1	Eric	Manrique
2	Mar	García
8	Gregorio	Sánchez



Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017
1	18	5/1/2017

Cód_Soc	Nombre	Apellido	Cód_Lib	Fecha
1	Eric	Manrique	1	19/3/2010
1	Eric	Manrique	2	9/10/2010
1	Eric	Manrique	3	18/12/2010
2	Mar	García	1	5/1/2017
8	Gregorio	Sánchez	$\omega$	$\omega$
18	$\omega$	$\omega$	1	5/1/2017

# Unión externa

- ¿Qué grados y cardinalidades tendrán *outer*?
  - Sea  $R_1$  y  $R_2$  de grado  $n_1$  y  $n_2$  y card.  $m_1$  y  $m_2$

Operador	Grado min.	Grado max.	Card. min.	Card. max.
left $\bowtie$				
right $\bowtie$				
full $\bowtie$				

# Unión externa

- ¿Qué grados y cardinalidades tendrán *outer*?
  - Sea  $R_1$  y  $R_2$  de grado  $n_1$  y  $n_2$  y card.  $m_1$  y  $m_2$
  - ¿Cuándo se llegaría a esos valores?

Operador	Grado min.	Grado max.	Card. min.	Card. max.
left $\bowtie$	$\max(n_1, n_2)$	$(n_1 + n_2) - 1$	$m_1$	$m_1 + m_2 - 1$
right $\bowtie$	$\max(n_1, n_2)$	$(n_1 + n_2) - 1$	$m_2$	$m_1 + m_2 - 1$
full $\bowtie$	$\max(n_1, n_2)$	$(n_1 + n_2) - 1$	$\max(m_1, m_2)$	$m_1 + m_2$

# Outer union

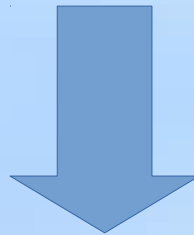
- Operación: hace la “unión” de dos relaciones con esquemas *no compatibles*
- El resultado tiene como esquema el conjunto de atributos de ambas relaciones
  - Incluye en el resultado las tuplas de ambas relaciones ( $R_1$  y  $R_2$ ) que no tienen valor que concuerde, poniendo  $\omega$  como valor de los campos que no concuerdan

# Outer union

- Ejemplo: listar todos los autores o editores

Autores OUTERUNION Editores

Cód_Aut	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky



Cód_Edit	Nombre	Apellido
1	Santiago	Ceri
2	Antón	De Miguel
3	Lawrence	Lessig

Cód_Aut	Cód_Edit	Nombre	Apellido
1	ω	Neal	Stephenson
2	ω	Lawrence	Lessig
3	ω	Noam	Chomsky
ω	1	Santiago	Ceri
ω	2	Antón	De Miguel
ω	3	Lawrence	Lessig

# División

- Operación: dadas dos relaciones  $r_1$  y  $r_2$  con esquemas  $R_1$  y  $R_2$ , donde  $R_2 \subset R_1$ , podemos obtener otra relación  $R$  definida sobre el esquema de relación  $(R_1 - R_2)$  y cuya extensión será:

$$t_i(R) = \{(v_{i1} , \dots , v_{iR}) \mid \\ \forall (v_{i(R+1)} , \dots , v_{iR_2}) \in r_2 \\ \exists (v_{i1} \dots v_{iR}, v_{i(R+1)} , \dots , v_{iR_2}) \in r_1\}$$

- Representación: se utilizan el mismo símbolo que en matemáticas.

$$r_1 / r_2$$

# División

- Ejemplo: encontrar a los socios que han tenido en préstamo un ejemplar de todos los libros

$$(\Pi_{\text{Cód_Lib}, \text{Cód_Soc}}(\text{Préstamos})) / (\Pi_{\text{Cód_Lib}}(\text{Libros}))$$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

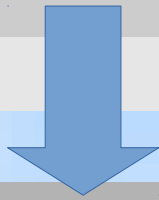
Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo:

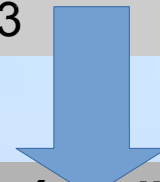
$$\Pi_{\text{Cód_Lib, Cód_Soc}}(\text{Préstamos}) / (\Pi_{\text{Cód_Lib}}(\text{Libros}))$$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

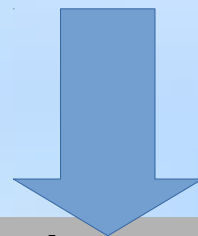


Cód_Lib	Cód_Soc
1	1
2	1
3	1
1	2

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



Cód_Lib
1
2
3



Cód_Soc
1



# División

- La división  $r_1/r_2$  comprueba que toda tupla de  $r_2$  aparezca en  $r_1$
- Por lo tanto todos los atributos de  $r_2$  tienen que estar en  $r_1$ 
  - Y dichos atributos no aparecerán en el resultado
    - Por lo que  $r_1$  debe tener algún atributos adicional
- Hemos visto un ejemplo más sencillo, pero sin  $r_2$  tuviera dos atributos se comprobarían que apareciera cada par de valores presentes con el resto de atributos de  $r_1$

# División

- Ejemplo2: ¿qué da la siguiente operación?

## Préstamos/Libros

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo2: ¿qué da la siguiente operación?

## Préstamos/Libros

Error: hay atributos en Libros que no están en Préstamo

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo3: ¿qué da la siguiente operación?

$$(\Pi_{\text{Cód_Lib}}(\text{Préstamos})) / (\Pi_{\text{Cód_Lib}}(\text{Libros}))$$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo3: ¿qué da la siguiente operación?

$$(\Pi_{\text{Cód\_Lib}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$$

Error: no hay atributos en la relación de la izquierda que no estén en la de la derecha

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo3: ¿qué da la siguiente operación?

$$(\Pi_{\text{Cód\_Lib}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$$

Error: no hay atributos en la relación de la izquierda que no estén en la de la derecha

*Se puede entender que daría una relación vacía (sin atributos), porque el atributo de la relación de la izquierda lo perdería al hacer la división*

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo4: ¿qué da la siguiente operación?

Préstamos/ $(\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# División

- Ejemplo4: ¿qué da la siguiente operación?

Préstamos/ $(\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Esta operación sí es correcta. ¿Qué atributos tendrá su resultado?

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



# División

- Ejemplo4: ¿qué da la siguiente operación?

Préstamos/ $(\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Esta operación sí es correcta. ¿Qué atributos tendrá su resultado? → Cód\_Soc, Fecha

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



Cód_Lib
1
2
3

# División

- Ejemplo4: ¿qué da la siguiente operación?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

¿Qué atributos tendrá?  $\rightarrow$  Cód\_Soc, Fecha

¿Qué tuplas contendrá?

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib
1
2
3

# División

- Ejemplo4: ¿qué da la siguiente operación?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

¿Qué atributos tendrá?  $\rightarrow$  Cód\_Soc, Fecha

¿Qué tuplas contendrá? Aquellas en las que para todo valor de  $\langle \text{Cód\_Soc}, \text{Fecha} \rangle$  existen todos los Cód\_Lib

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib
1
2
3

# División

- Ejemplo4: ¿qué da la siguiente operación?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

¿Qué atributos tendrá?  $\rightarrow$  Cód\_Soc, Fecha

¿Qué tuplas contendrá?

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib
1
2
3

# División

- Ejemplo4: ¿qué da la siguiente operación?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

¿Qué atributos tendrá?  $\rightarrow$  Cód\_Soc, Fecha

¿Qué tuplas contendrá? Aquellas en las que para todo valor de  $\langle \text{Cód\_Soc}, \text{Fecha} \rangle$  existen todos los Cód\_Lib

Es decir, aquellos socios que los días que realizaban un préstamo, sacaban todos los libros

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

Cód_Lib
1
2
3



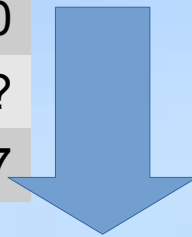
Cód_Soc	Fecha
---------	-------

# División

- Ejemplo5: ¿qué tuplas tendría que añadirse a Préstamos para dar un resultado no vacío?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
?	?	?
1	2	5/1/2017



Cód_Lib
1
2
3

Cód_Soc	Fecha
xxx	yyy

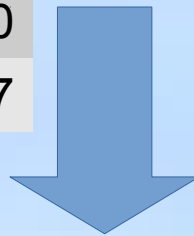
# División

- Ejemplo5: ¿qué tuplas tendría que añadirse a Préstamos para dar un resultado no vacío?

$\text{Préstamos} / (\Pi_{\text{Cód_Lib}}(\text{Libros}))$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
2	1	18/12/2010
1	1	18/12/2010
1	2	5/1/2017

Cód_Lib
1
2
3



Cód_Soc	Fecha
1	18/12/2010

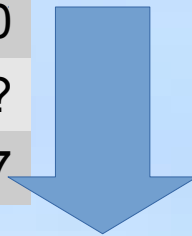
# División

- Ejemplo6: ¿qué tuplas tendría que añadirse a Préstamos para dar dos tuplas de resultado?

Préstamos/ $(\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
2	1	18/12/2010
1	1	18/12/2010
?	?	?
1	2	5/1/2017

Cód_Lib
1
2
3



Cód_Soc	Fecha
1	18/12/2010



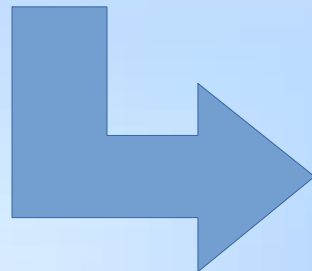
# División

- Ejemplo6: ¿qué tuplas tendría que añadirse a Préstamos para dar dos tuplas de resultado?

$\text{Préstamos} / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
2	1	18/12/2010
1	1	18/12/2010
3	2	5/1/2017
2	2	5/1/2017
1	2	5/1/2017

Cód_Lib
1
2
3



Cód_Soc	Fecha
1	18/12/2010
2	5/1/2017

# División

- Si tenemos dos relaciones  $r_1$  y  $r_2$  con esquemas  $R_1$  y  $R_2$  y calculamos  $R_1/R_2$ 
  - Los atributos de  $R_2$ :
    - No aparecerán en el resultado
  - Los atributos de  $R_1$  que también están en  $R_2$ 
    - No aparecerán en el resultado
  - Los atributos de  $R_1$  que no están en  $R_2$ 
    - Son los únicos que aparecerán en el resultado
    - Cada instancia suya son la unidad a la que se le comprobará el “para todo”: cuidado a más atributos, mayor “exigencia”

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
    - ¿y si quiero saber quién sacó cada libro?

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
    - ¿y si quiero saber quién sacó cada libro?
      - Pues como ya tengo las fechas, hago un producto natural con Préstamos (no se puede hacer sólo con la división)

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\sigma_{\text{Cód\_Lib} > 10}((\Pi_{\text{Cód\_Lib}}(\text{Libros}))))$

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\sigma_{\text{Cód\_Lib} > 10}((\Pi_{\text{Cód\_Lib}}(\text{Libros}))))$ 
    - Fechas en las que sacaron todos los libros con código mayor que 10
  - ¿Cuál de estas dos expresiones puede dar más tuplas de resultados?

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\sigma_{\text{Cód\_Lib} > 10}((\Pi_{\text{Cód\_Lib}}(\text{Libros}))))$ 
    - Fechas en las que sacaron todos los libros con código mayor que 10
  - ¿Cuál de estas dos expresiones puede dar más tuplas de resultados?
    - La parte izquierda es igual, pero en la derecha una exige menos que la otra ....



# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\sigma_{\text{Cód\_Lib} > 10}(\Pi_{\text{Cód\_Lib}}(\text{Libros})))$ 
    - Fechas en las que sacaron todos los libros con código mayor que 10
  - $(\sigma_{\text{Cód\_Lib} > 10}(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos}))) / ((\Pi_{\text{Cód\_Lib}}(\text{Libros})))$

# División

- Interpretación de otras divisiones:
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\Pi_{\text{Cód\_Lib}}(\text{Libros}))$ 
    - Fechas en las que sacaron todos los libros
  - $(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos})) / (\sigma_{\text{Cód\_Lib} > 10}((\Pi_{\text{Cód\_Lib}}(\text{Libros})))$ 
    - Fechas en las que sacaron todos los libros con código mayor que 10
  - $(\sigma_{\text{Cód\_Lib} > 10}(\Pi_{\text{Cód\_Lib, Fecha}}(\text{Préstamos}))) / ((\Pi_{\text{Cód\_Lib}}(\text{Libros})))$ 
    - Fechas en las que sacaron todos los libros, pero comprobando sólo los préstamos de los libros con código mayor que 10
      - Es decir, si hay un libro con código  $\leq 10$  en la derecha, el resultado es vacío

# División

- Interpretación de otras divisiones2:
  - $\text{Préstamos} / (\prod_{\text{Cód\_Lib}}(\text{Libros}) \times \prod_{\text{Cód\_Soc}}(\text{Socios}))$

# División

- Interpretación de otras divisiones2:
  - $\text{Préstamos} / (\prod_{\text{Cód\_Lib}}(\text{Libros}) \times \prod_{\text{Cód\_Soc}}(\text{Socios}))$ 
    - Fechas en las que todos los socios sacaron todos los libros
  - $(\prod_{\text{Cód\_Lib}, \text{Cód\_Edit}}(\text{Libros})) / \prod_{\text{Cód\_Edit}}(\text{Editores})$

# División

- Interpretación de otras divisiones2:
  - Préstamos/ $(\prod_{\text{Cód\_Lib}}(\text{Libros}) \times \prod_{\text{Cód\_Soc}}(\text{Socios}))$ 
    - Fechas en las que todos los socios sacaron todos los libros
  - $(\prod_{\text{Cód\_Lib}, \text{Cód\_Edit}}(\text{Libros})) / \prod_{\text{Cód\_Edit}}(\text{Editores})$ 
    - Libros que han sido publicados por todos los editores (se supone que se re-editarían)
  - $(\prod_{\text{Cód\_Aut}, \text{Cód\_Edit}}(\text{Libros})) / \prod_{\text{Cód\_Edit}}(\text{Editores})$

# División

- Interpretación de otras divisiones2:
  - Préstamos/ $(\prod_{\text{Cód\_Lib}}(\text{Libros}) \times \prod_{\text{Cód\_Soc}}(\text{Socios}))$ 
    - Fechas en las que todos los socios sacaron todos los libros
  - $(\prod_{\text{Cód\_Lib}, \text{Cód\_Edit}}(\text{Libros})) / \prod_{\text{Cód\_Edit}}(\text{Editores})$ 
    - Libros que han sido publicados por todos los editores (se supone que se re-editarían)
  - $(\prod_{\text{Cód\_Aut}, \text{Cód\_Edit}}(\text{Libros})) / \prod_{\text{Cód\_Edit}}(\text{Editores})$ 
    - Autores que han publicado con todos los editores

# Asignación a variables

- Operación: asigna una relación temporal a una variable. Esta acción simplifica las operaciones con expresiones complejas
- Representación: se representa por medio de una flecha ( $\leftarrow$ ) y actúa de igual forma que en los lenguajes de programación
- Ejemplo: encontrar a todos los socios que tienen en préstamo un ejemplar de todos los libros

$\text{var} \leftarrow \Pi_{\text{Cód\_Lib}}(\text{Libros})$

$\Pi_{\text{Cód\_Lib}, \text{Cód\_Soc}}(\text{Préstamos})/\text{var}$

# Expresión con operadores fundamentales

- Vimos que todos los operadores se podían expresar en función de los fundamentales:
  - Proyección
  - Selección
  - Producto cartesiano
  - Unión de conjuntos
  - Diferencia de conjuntos



# Expresión con operadores fundamentales

- Expresión en base de los operadores fund.:
  - Intersección de conjuntos
  - Productos de reunión
    - Producto theta
    - Producto natural
    - Autounión
    - Unión externa por la izquierda
    - Unión externa por la derecha
    - Unión externa completa
  - División

# Expresión con operadores fundamentales

- Expresión en base de los operadores fund.:

- Intersección de conjuntos

$$R \cap S = (R \cup S) - ((R - S) \cup (S - R))$$

- Producto theta

$$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta} (r_1 \times r_2)$$

- Producto natural

Renombro los atributos comunes en S para no perderlos

$$R \bowtie S = \Pi_{\text{atrib1R}, \dots} (\sigma_{\text{atrib1R}=\text{atrib1S}} (\sigma_{\text{atrib2R}=\text{atrib2S}} \dots (R \times S)))$$

- Autounión

- No lo vimos como operador propio

# Expresión con operadores fundamentales

- Expresión en base a operadores fund.:

- Unión externa por la izquierda

$$(R \bowtie S) \cup ((R - \pi_{r_1, r_2, \dots, r_n}(R \bowtie S)) \times \{(\omega, \dots, \omega)\})$$

- Unión externa por la derecha

$$(R \bowtie S) \cup (\{(\omega, \dots, \omega)\} \times (S - \pi_{s_1, s_2, \dots, s_n}(R \bowtie S)))$$

- Unión externa completa

$$R \bowtie S = (R \bowtie S) \cup (R \bowtie S)$$

- División

- $R \div S = \Pi_{a_1, \dots, a_n}(R) - (\Pi_{a_1, \dots, a_n}(((\Pi_{a_1, \dots, a_n}(R) \times S) - R)))$

# Criterio en consultas

- Hemos visto que se puede obtener un mismo resultado mediante distintas expresiones
  - ¿Podemos afirmar algo sobre la eficiencia de las distintas expresiones?

# Criterio en consultas

- Hemos visto que se puede obtener un mismo resultado mediante distintas expresiones
  - ¿Podemos afirmar algo sobre la eficiencia de las distintas expresiones?
    - Difícilmente, porque trabajamos en el nivel conceptual, no en el físico: desconocemos cómo están los datos en memoria/disco
  - ¿Y sobre su la complejidad de las expresiones?

# Criterio en consultas

- Hemos visto que se puede obtener un mismo resultado mediante distintas expresiones
  - ¿Podemos afirmar algo sobre la eficiencia de las distintas expresiones?
    - Difícilmente, porque trabajamos en el nivel conceptual, no en el físico: desconocemos cómo están los datos en memoria/disco
  - ¿Y sobre su la complejidad de las expresiones?
    - Ahí sí (y se valorará en el examen):
      - Evitar incluir relaciones/atributos innecesarios para una consulta
      - Evitar rehacer expresiones ya realizadas o volver a obtener información ya disponible
      - Son malas prácticas que complican modificar, depurar, etc

# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Clientes (Cód\_Cli, Nombre, Apellido)
  - Empleados (N\_seg\_soc, Nombre, Apellido)
  - Préstamos (Cód\_Cli, N\_seg\_soc, Cantidad, Vence)
  - AsuntosPropios (N\_seg\_soc, Fecha)

<u>Cód_Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód_Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Con la estructura actual:
    - ¿Cuántos préstamos puede pedir cada cliente?
    - ¿Cuántos préstamos pueden existir en la base de datos?

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023



# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Con la estructura actual:
    - ¿Cuántos días de asuntos propios se puede coger cada empleado? ¿Podría limitarlo a 3 máximo?

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Con la estructura actual:
    - ¿Cuántos días de asuntos propios se puede coger cada empleado? ¿Podría limitarlo a 3 máximo?:
      - AsuntosPropios (N\_seg\_soc, Fecha1, Fecha2, Fecha3)
        - O añadir los campos fechaX a la relación Empleado

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Con la estructura actual:
    - ¿Cuántos empleados pueden cogerse un día de asuntos propios en la misma fecha? ¿Podría poner 1 máximo?:

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un par ejemplos con esta otra BBDD:
  - Con la estructura actual:
    - ¿Cuántos empleados pueden cogerse un día de asuntos propios en la misma fecha? ¿Podría poner 1 máximo?:
      - AsuntosPropios (N\_seg\_soc, Fecha)

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Apellido de los empleados de nombre “Lola” que han dado préstamos de más de 2.000€

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Apellido de los empleados de nombre “Lola” que han dado préstamos de más de 2.000€
  - $A \leftarrow \text{Préstamos} \bowtie \text{Empleados}$
  - $B \leftarrow \sigma_{\text{cant} > 2000 \text{ AND nombre} = \text{“Lola”}}(\pi_{\text{N\_seg\_soc}, \text{Nombre}, \text{cant}}(A))$
  - $\pi_{\text{apellido}}(B \bowtie \text{Empleados})$

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Apellido de los empleados de nombre “Lola” que han dado préstamos de más de 2.000€
  - $A \leftarrow \text{Préstamos} \bowtie \text{Empleados}$
  - $B \leftarrow \sigma_{\text{cant} > 2000 \text{ AND nombre} = \text{“Lola”}}(\pi_{\text{N\_seg\_soc}, \text{Nombre}, \text{cant}}(A))$
  - $\pi_{\text{apellido}}(B \bowtie \text{Empleados})$
- Se hacen dos productos con la relación Empleados
  - $A \leftarrow \text{Empleados} \bowtie \text{Préstamos}$
  - $B \leftarrow \sigma_{\text{cant} > 2000 \text{ AND Nombre} = \text{“Lola”}}(\pi_{\text{N\_seg\_soc}, \text{Nombre}, \text{Apellido}, \text{cant}}(A))$
  - $\pi_{\text{Apellido}}(B)$

# Criterio en consultas

- Apellido de los empleados de nombre “Lola” que han dado préstamos de más de 2.000€
  - $A \leftarrow \text{Préstamos} \bowtie \text{Empleados}$
  - $B \leftarrow \sigma_{\text{cant} > 2000 \text{ AND nombre} = \text{“Lola”}}(\pi_{\text{N\_seg\_soc}, \text{Nombre}, \text{cant}}(A))$
  - $\pi_{\text{apellido}}(B \bowtie \text{Empleados})$
- Se hacen dos productos con la relación Empleados
  - $A \leftarrow \text{Empleados} \bowtie \text{Préstamos}$
  - $B \leftarrow \sigma_{\text{cant} > 2000 \text{ AND Nombre} = \text{“Lola”}}(A)$
  - $\pi_{\text{Apellido}}(B)$





# Criterio en consultas

- Eso no quiere decir que nunca se deba usar dos veces una relación
  - Pero en este caso hay que evitar eliminar algo de una expresión si después nos hace falta
  - Y lo mismo para recalcular

# Criterio en consultas

- Apellido de los empleados que nunca han dado un préstamo de más de 2.000 euros

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Apellido de los empleados que nunca han dado un préstamo de más de 2.000 euros
  - $A \leftarrow \Pi_{N\_seg\_soc}(Empleados) - \Pi_{N\_seg\_soc}(\sigma_{cant > 2000}(Préstamos))$
  - $\Pi_{apellido}(A \bowtie Empleados)$

<u>Cód Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?

<u>Cód_Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód_Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_cli}}(\text{Clientes})$
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_Cli}}(\text{Préstamos})$

<u>Cód_Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód_Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_cli}}(\text{Clientes})$  😊
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_Cli}}(\text{Préstamos})$

<u>Cód_Cli</u>	Nombre	Apellido
1	José	Pérez
2	Dolores	Gómez
3	Ambrosio	Álvarez

<u>N_seg_soc</u>	Nombre	Apellido
11	Pepe	Gades
22	Lola	Peláez
33	Anselmo	Muñoz

<u>Cód_Cli</u>	<u>N_seg_soc</u>	Cantidad	Vence
1	11	100.000	1/10/2010
1	22	1.000	1/11/2026
2	22	5.000	1/3/2027

<u>N_seg_soc</u>	<u>Fecha</u>
11	1/10/2018
11	21/3/2019
33	1/12/2023

# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_cli}}(\text{Clientes})$  😊
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_Cli}}(\text{Préstamos})$ 
    - Esta segunda expresión obtendría los empleados que han dado préstamos a todo el que alguna vez ha pedido un préstamo
      - Pero puede haber clientes que nunca hayan pedido un préstamos, y no se comprobarían

# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_cli}}(\text{Clientes})$  😊
  - ¿Y qué tal así?
    - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos} \bowtie \text{AsuntosPropios}) / \Pi_{\text{Cód\_Cli}}(\text{Clientes})$



# Criterio en consultas

- Veamos un ejemplo con esta otra BBDD:
  - Empleados que hayan dado préstamos a todo cliente ¿De qué relación cojo el Cód\_Cli?
  - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos}) / \Pi_{\text{Cód\_cli}}(\text{Clientes})$  😊
  - ¿Y qué tal así?
    - $\Pi_{\text{Cód\_Cli}, \text{N\_seg\_soc}}(\text{Préstamos} \bowtie \text{AsuntosPropios}) / \Pi_{\text{Cód\_Cli}}(\text{Clientes})$
    - Este sería un ejemplo en el se obtiene el mismo resultado pero con una expresión más complicada difícil de justificar

# Funciones de agregación

- Al igual que los OUTER JOIN y OUTER UNION, los operadores de agregación de SQL que no se pueden expresar en álgebra relacional:
  - Contar el número de clientes que cumplen una condición
  - Sumar los salarios de todos los empleados
  - Calcular el coste medio de los artículos de una determinada categoría
  - Indicar el mínimo/máximo de ...

# Funciones de agregación

- Para estos casos se define una función de agregación  $G$  (a veces se usa *script*  $F$ )

$$\text{atributos\_agrupamiento} G_{\text{invocaciones\_funciones}}(R)$$

- Las funciones clásicas son: sumatorio, media, mínimo y máximo y contar
- Una invocación de una función es una función seguida de un sólo atributo al que se aplica
- La salida tiene los atributos de agrupamiento más un atributo por cada invocación (estos últimos de la forma *función\_atributo*)

# Funciones de agregación

- Consideraciones:
  - Si no se indican atributos de agrupamiento, se aplica la función/es a todas las tuplas (y sale una única tupla en el resultado)
    - Ej: empleado de más edad (da igual su Dpto, puesto, ...)

$G_{\text{MÁXIMO edad}}(\text{Empleados})$

- Las funciones de agregación NO eliminan duplicados para ser coherentes en cálculos aritméticos (SUMATORIO y MEDIA)
- La salida es una Relación y no un escalar, aunque sea una única tupla con un único atributo

# Funciones de agregación

- Ejemplo: contar los libros publicados por cada autor

Cód\_Aut <sup>G</sup>CONTAR Cód\_Lib (Libros)

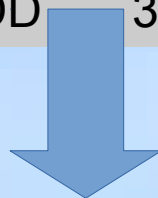
Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Funciones de agregación

- Ejemplo: contar los libros publicados por cada autor

Cód\_Aut <sup>G</sup>CONTAR Cód\_Lib (Libros)

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



Cód_Aut	CONTAR_Cód_lib
1	1
2	2

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente

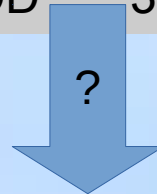
Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente

Cód\_Lib  $G_{MÁXIMO}$  Cód\_Lib (Libros)

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



MÁXIMO_Cód_lib
3

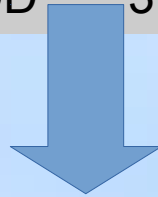


# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente

Cód\_Lib  $G_{MÁXIMO}$  Cód\_Lib (Libros)

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



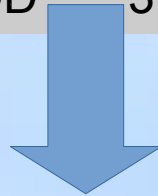
Cód_Lib	MÁXIMO_Cód_lib
1	1
2	2
3	3

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente

$G_{\text{MÁXIMO Cód_Lib}}(\text{Libros})$

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2



MÁXIMO_Cód_lib
3

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente ¿Y si quiero saber su nombre?

$A \leftarrow G_{\text{MÁXIMO Cód_Lib}}(\text{Libros})$

$\sigma_{\text{Cód_Lib}=A}(\text{Libros})$

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

MÁXIMO_Cód_lib
3

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente ¿Y si quiero saber su nombre?

$A \leftarrow G_{\text{MÁXIMO Cód_Lib}}(\text{Libros})$

$\sigma_{\text{Cód_Lib}=A}(\text{Libros})$

ERROR, A es una relación (con una sola columna y una sola tupla), no una constante

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

MÁXIMO_Cód_lib
3

# Funciones de agregación

- Ejemplo2: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el libro más reciente ¿Y si quiero saber su nombre?

$$A \leftarrow G_{\text{MÁXIMO Cód_Lib}}(\text{Libros})$$

$$A1(\text{Cód_Lib}) \leftarrow A(\text{MÁXIMO\_Cód_Lib})$$

$$\Pi_{\text{nombre}}(\text{Libros} \bowtie A1)$$

Cód_Lib	Nombre	Cód_Edit	Cód_Aut
1	SSOO	1	1
2	BBDD	2	2
3	ABBDD	3	2

MÁXIMO_Cód_lib
3

# Funciones de agregación

- Ejemplo3: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el último socio que sacó el libro más antiguo en Préstamo

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017

# Funciones de agregación

- Ejemplo3: suponiendo que Cód\_Lib indica lo reciente que es un libro, obtener el último socio que sacó el libro más antiguo en Préstamo

$$A(\text{Cód\_Lib}) \leftarrow G_{\text{MÍNIMO Cód\_Lib}}(\text{Libros})$$

$$B \leftarrow A \bowtie \text{Préstamos}$$

$$C \leftarrow G_{\text{MÁXIMO Fecha}}(B)$$

$$\Pi_{\text{cód\_Soc}}(\text{Préstamo} \bowtie C \bowtie A)$$

Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
2	1	9/10/2010
3	1	18/12/2010
1	2	5/1/2017



Cód_Lib	Cód_Soc	Fecha
1	1	19/3/2010
1	2	5/1/2017

Cód_Soc
2



# Clausura transitiva

- Es un operador que se define para aplicar operaciones recursivamente a tuplas de un mismo tipo
  - Ej: supongamos que todo alumno de la Universidad tiene un mentor, que es del curso superior
    - En álgebra se puede calcular fácilmente los mentores de los alumnos de primero, y los mentores de los alumnos de segundo, etc.
    - Pero no se pueden calcular los mentores de los alumnos de todos los cursos
    - Para ello hace falta la clausura transitiva
- No lo estudiaremos, sólo saber que existe



# Qué ofrece SQL

- SQL ofrece los operadores de álgebra, cálculo y otros adicionales (como hacer operaciones aritméticas con los resultados, ordenar, ...)
- Además cada fabricante “interpreta” el álgebra/cálculo como cree más adecuado
  - Más eficiente
  - Más cómodo para el administrador de BD
  - Más cómodo para el desarrollador
  - ...

# ¿Para qué sirve el álgebra?

- Procesado de datos utilizando sus operaciones
- Ayuda a escribir expresiones simplificando algunas de ellas
- Permite realizar estimaciones de tiempo de respuesta basadas en evidencias
  - Escapa de esta asignatura (ABDD)
- Patrón para otros lenguajes: podemos decir que un lenguaje es relacionalmente completo si cualquier expresión se puede expresar en álgebra

# Referencias

- Apuntes Esther Gadeschi
- Libro Elmasri, 3ª ed.
- [https://en.wikipedia.org/wiki/Relational\\_algebra#Outer\\_joins](https://en.wikipedia.org/wiki/Relational_algebra#Outer_joins)
- [https://en.wikipedia.org/wiki/Mathematical\\_operators\\_and\\_symbols\\_in\\_Unicode](https://en.wikipedia.org/wiki/Mathematical_operators_and_symbols_in_Unicode)
- SQL as a Data Manipulation Language  
<http://www-inf.it-sudparis.eu/COURS/BD/PRIVATE/EM2ENG/SLIDES/4SQL-DMLEM2ENG.pdf>
- A Simpler (and Better) SQL Approach to Relational Division

Gracias por la atención  
*¿Preguntas?*