



UA

## *Unidad 3: BBDD relacionales*

*BBDD01, Sesión 8:*  
*Álgebra relacional*  
*Cálculo relacional*

*Ignacio Olmeda*

*Josefa Gómez*

*Daniel Rodríguez García*

*Iván González Diego*

*Dept. Ciencias de la Computación*

*Universidad de Alcalá*



# *INDICE*

- Algebra relacional
- Algebra relacional extendida
- Modificación de la base de datos
- Vistas
- Cálculo relacional de tuplas
- Cálculo relacional de dominios.

Referencias: Silberschatz 4ª Ed. pp 53-82



# *Lenguajes de consulta*

- Lenguaje de consulta  $\Rightarrow$  usuario solicita información de la base de datos
- Dos tipos:
  - Procedimentales  $\Rightarrow$  usuario especifica las operaciones a realizar
    - Álgebra relacional
  - No procedimentales  $\Rightarrow$  usuario describe “lo que necesita”, no el modo de conseguirlo.
    - Cálculo relacional de tuplas y el de dominios
- Lenguaje también incluye componentes para modificación de la base de datos.



# *Algebra relacional*

- *Lenguaje consulta procedimental  $r(R)$*
- *El resultado de cualquier operación (monaria o binaria) es una nueva relación  $\rightarrow$  operaciones cerradas  $\rightarrow$  se pueden componer*
- *Seis operadores básicos*
  - *Selección*
  - *Proyección*
  - *Unión*
  - *Diferencia de conjuntos*
  - *Producto Cartesiano*
  - *Renombrado*



# Operación selección

- Notación:  $\sigma_p(r)$
- $p$  se llama predicado de la selección
- Se define como:

$$\sigma_p(r) = \{t \mid t \in r \text{ y } p(t)\}$$

Donde  $p$  es una expresión lógica = Términos lógicos conectados por  $\wedge$  (**and**),  $\vee$  (**or**),  $\neg$  (**not**)

Cada término es:

$\langle \text{atributo} \rangle$  op  $\langle \text{atributo} \rangle$  o  $\langle \text{constante} \rangle$

donde op es:  $=, \neq, >, \geq, <, \leq$

- Comparación implicando a un valor nulo  $\Rightarrow$  falsa
- Ejemplo:



# Operación selección

- *Ejemplo: Información sobre préstamos de la sucursal de Navacerrada*

## **prestamo**

número-préstamo	nombre-sucursal	importe
P-11	Collado Mediano	900
P-14	Centro	1.500
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300
P-17	Centro	1.000
P-23	Moralzarzal	2.000
P-93	Becerril	500

$\sigma$  nombre\_sucursal="Navacerrada"(prestamo)

número-préstamo	nombre-sucursal	importe
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300



# Operación proyección

- Notación:  $\Pi_{A_1, A_2, \dots, A_k}(r)$   
donde  $A_1, A_2$  son atributos y  $r$  es la relación
- El resultado es una relación de  $k$  columnas eliminando de  $R$  las que no están en la lista
- Las filas duplicadas se eliminan. Son conjuntos
- Ejemplo: Importe de cada uno de los préstamos

prestamo

número-préstamo	nombre-sucursal	importe
P-11	Collado Mediano	900
P-14	Centro	1.500
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300
P-17	Centro	1.000
P-23	Moralzarzal	2.000
P-93	Becerril	500

$\Pi_{\text{numero\_prestamo}, \text{importe}}(\text{prestamo})$

número-préstamo	importe
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500



# *Composición de operaciones relacionales*

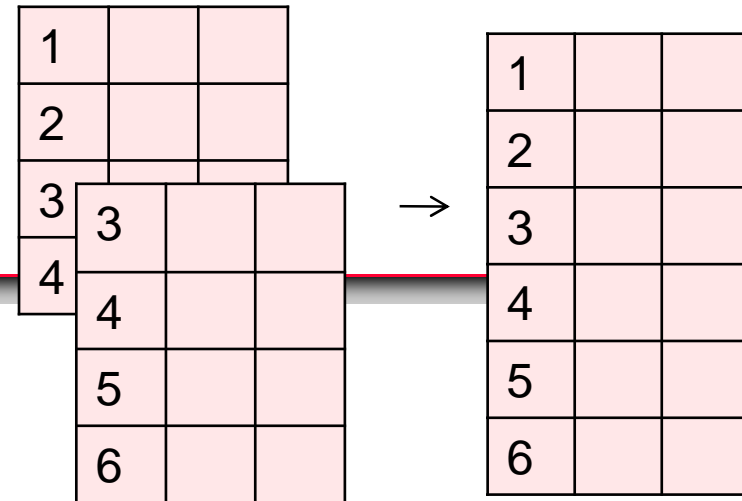
- *Resultado de una operación relacional es otra relación*
- *→ Se pueden componer expresiones*
- *Ejemplo:*
  - *Encontrar los clientes que viven en Peguerinos*

$\Pi_{\text{nombre\_cliente}} (\sigma_{\text{ciudad\_cliente}=\text{"Peguerinos"}}(\text{cliente}))$





## Operación unión



■ Notación:  $r \cup s$

■ Se define como:

$$r \cup s = \{t \mid t \in r \text{ o } t \in s\}$$

■ Para  $r \cup s$  sea válida  $\rightarrow r$  y  $s$  compatibles:

1.  $r, s$  debe tener el mismo número de atributos
2. El dominio de los atributos debe de ser compatible

■ No hay valores duplicados

*Ejemplo: Nombre de todos los clientes, ya sea que tengan un préstamo o una cuenta*

$$\Pi_{\text{nombre\_cliente}} (\text{prestatario}) \cup \Pi_{\text{nombre\_cliente}} (\text{impositor})$$



## Operación unión

Ejemplo:: Nombre de todos los clientes, ya sea que tengan un préstamo o una cuenta

*prestatario*

<i>nombre cliente</i>	<i>número préstamo</i>
Fernández	P-16
Gómez	P-93
Gómez	P-15
López	P-14
Pérez	P-17
Santos	P-11
Sotoca	P-23
Valdivieso	P-17

*impositor*

<i>nombre cliente</i>	<i>número cuenta</i>
Abril	C-102
Gómez	C-101
González	C-201
González	C-217
López	C-222
Rupérez	C-215
Santos	C-305

$$\Pi_{\text{nombre\_cliente}}(\text{prestatario}) \cup \Pi_{\text{nombre\_cliente}}(\text{impositor})$$

<i>nombre-cliente</i>
Abril
Fernández
Gómez
González
López
Pérez
Rupérez
Santos
Sotoca
Valdivieso



# Operación diferencia de conjuntos

- Notación  $r - s$
- Definido como:  $r - s = \{t \mid t \in r \text{ y } t \notin s\}$
- Las relaciones deben de ser **compatibles**
- Ejemplo: Clientes que tienen un préstamo pero no una cuenta impositor

1		
2		

1		
2		
3		
4	3	
	4	
5		
6		

*impositor*

nombre cliente	número cuenta
Abril	C-102
Gómez	C-101
González	C-201
González	C-217
López	C-222
Rupérez	C-215
Santos	C-305

*prestatario*

nombre cliente	número préstamo
Fernández	P-16
Gómez	P-93
Gómez	P-15
López	P-14
Pérez	P-17
Santos	P-11
Sotoca	P-23
Valdivieso	P-17

$$\Pi_{\text{nombre\_cliente}}(\text{prestatario}) - \Pi_{\text{nombre\_cliente}}(\text{impositor})$$

nombre-cliente
Abril
González
Rupérez



# Operación producto cartesiano

- Notación  $r \times s$
- Se define como:
$$r \times s = \{t \mid t \in r \text{ y } t \in s\}$$
- Argumentos de producto cartesiano
- Si  $r$  tiene  $a$  atributos y  $s$  tiene  $b$  atributos  $\rightarrow r \times s$  tiene  $a+b$  con nombre, el de los atributos originales
- Ejemplo:
  - $r = \text{prestatario} \times \text{préstamo}$
  - $(\text{prestatario.nombre-cliente}, \text{prestatario.número-préstamo}, \text{préstamo.nombre-sucursal}, \text{préstamo.número-préstamo}, \text{préstamo.importe})$
  - Si no hay ambigüedad  $\rightarrow$  simplificar:  $(\text{nombre-cliente}, \text{prestatario.número-préstamo}, \text{nombre-sucursal}, \text{préstamo.número-préstamo}, \text{importe})$
- Si hay  $n_1$  tuplas en  $r$  y  $n_2$  tuplas en  $s$ , el resultado es  $n_1 * n_2$  tuplas.



# Operación producto cartesiano

prestamo x prestatario

nombre-cliente	prestatario.número-préstamo	préstamo.número-préstamo	nombre-sucursal	importe
Santos	P-17	P-11	Collado Mediano	900
Santos	P-17	P-14	Centro	1.500
Santos	P-17	P-15	Navacerrada	1.500
Santos	P-17	P-16	Navacerrada	1.300
Santos	P-17	P-17	Centro	1.000
Santos	P-17	P-23	Moralzarzal	2.000
Santos	P-17	P-93	Becerril	500
Gómez	P-23	P-11	Collado Mediano	900
Gómez	P-23	P-14	Centro	1.500
Gómez	P-23	P-15	Navacerrada	1.500
Gómez	P-23	P-16	Navacerrada	1.300
Gómez	P-23	P-17	Centro	1.000
Gómez	P-23	P-23	Moralzarzal	2.000
Gómez	P-23	P-93	Becerril	500
López	P-15	P-11	Collado Mediano	900
López	P-15	P-14	Centro	1.500
López	P-15	P-15	Navacerrada	1.500
López	P-15	P-16	Navacerrada	1.300
López	P-15	P-17	Centro	1.000
López	P-15	P-23	Moralzarzal	2.000
López	P-15	P-93	Becerril	500
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Valdivieso	P-17	P-11	Collado Mediano	900
Valdivieso	P-17	P-14	Centro	1.500
Valdivieso	P-17	P-15	Navacerrada	1.500
Valdivieso	P-17	P-16	Navacerrada	1.300
Valdivieso	P-17	P-17	Centro	1.000
Valdivieso	P-17	P-23	Moralzarzal	2.000
Valdivieso	P-17	P-93	Becerril	500
Fernández	P-16	P-11	Collado Mediano	900
Fernández	P-16	P-14	Centro	1.500
Fernández	P-16	P-15	Navacerrada	1.500
Fernández	P-16	P-16	Navacerrada	1.300
Fernández	P-16	P-17	Centro	1.000
Fernández	P-16	P-23	Moralzarzal	2.000
Fernández	P-16	P-93	Becerril	500



# Operación producto cartesiano

*Ejemplo: Averiguar los nombres de todos los clientes con un préstamo en Navacerrada.*

- 1º - producto cartesiano

$$\sigma_{\text{nombre-sucursal}} = \text{«Navacerrada»} (\text{prestatario} \times \text{préstamo})$$

- 2º -Seleccionar solo los de “Navacerrada”

nombre-cliente	prestatario.número-préstamo	préstamo.número-préstamo	nombre-sucursal	importe
Santos	P-17	P-15	Navacerrada	1.500
Santos	P-17	P-16	Navacerrada	1.300
Gómez	P-23	P-15	Navacerrada	1.500
Gómez	P-23	P-16	Navacerrada	1.300
López	P-15	P-15	Navacerrada	1.500
López	P-15	P-16	Navacerrada	1.300
Sotoca	P-14	P-15	Navacerrada	1.500
Sotoca	P-14	P-16	Navacerrada	1.300
Pérez	P-93	P-15	Navacerrada	1.500
Pérez	P-93	P-16	Navacerrada	1.300
Gómez	P-11	P-15	Navacerrada	1.500
Gómez	P-11	P-16	Navacerrada	1.300
Valdivieso	P-17	P-15	Navacerrada	1.500
Valdivieso	P-17	P-16	Navacerrada	1.300
Fernández	P-16	P-15	Navacerrada	1.500
Fernández	P-16	P-16	Navacerrada	1.300



## Operación producto cartesiano

- 3º - Eliminar tuplas que no corresponden al mismo préstamo

$$\sigma_{\text{prestatario.número-préstamo} = \text{préstamo.número-préstamo}} (\sigma_{\text{nombre-sucursal} = \text{«Navaceñada»}} (\text{prestatario} \times \text{préstamo}))$$

- 4º - Proyección para eliminar los atributos no necesarios

$$\Pi_{\text{nombre-cliente}} (\sigma_{\text{prestatario.número-préstamo} = \text{préstamo.número-préstamo}} (\sigma_{\text{nombre-sucursal} = \text{«Navaceñada»}} (\text{prestatario} \times \text{préstamo})))$$

nombre-cliente
Fernández López



## *Operación renombrado*

- *Permite poner nombres a los resultados de las expresiones del álgebra relacional*
- *Permite referir a una relación por más de un nombre*

*Ejemplo:*

$$\rho_X(E) \quad \text{o} \quad X \leftarrow E \text{ (esto es asignación)}$$

*Devuelve resultado de la expresión E bajo el nombre X*

*Si la expresión E tiene n atributos, entonces*

$$\rho_{X(A_1, A_2, \dots, A_n)}(E) \quad \text{o} \quad X(A_1, A_2, A_n) \leftarrow E$$

*Devuelve el resultado de la expresión E bajo el nombre X y con los atributos renombrados como A1, A2, ..., An.*





# Operación renombrado

■ *Ejemplo: Buscar el máximo saldo de cuenta del banco impositor*

número-cuenta	nombre-sucursal	saldo
C-101	Centro	500
C-215	Becerril	700
C-102	Navacerrada	400
C-305	Collado Mediano	350
C-201	Galapagar	900
C-222	Moralzarzal	700
C-217	Galapagar	750

- 1º - *Calcular una relación intermedia que contiene todos los saldos que no son el máximo.*
- 2º - *Realizar la diferencia entre la proyección del saldo de las cuentas y esta relación intermedia*
- 3º *Renombrar*



# Operación renombrado

- 1º - Calcular una relación intermedia que contiene todos los saldos que no son el máximo. Aplicar renombrado a cuenta y construir la relación intermedia

$$\Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d (\text{cuenta})))$$

saldo
500
400
700
750
350

- 2º - Calcular el máximo

$$\Pi_{\text{saldo}} (\text{cuenta}) - \Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d (\text{cuenta})))$$

saldo
900

- 3º Renombrar

$$\rho_{\text{maximo\_saldo (máximo)}} ( \Pi_{\text{saldo}} (\text{cuenta}) - \Pi_{\text{cuenta.saldo}} (\sigma_{\text{cuenta.saldo} < d.\text{saldo}} (\text{cuenta} \times \rho_d (\text{cuenta}))) )$$



## *Expresiones del álgebra relacional*

■ Si  $E_1$  y  $E_2$  son expresiones del álgebra relacional, las siguientes son también expresiones del álgebra relacional

- $E_1 \cup E_2$
- $E_1 - E_2$
- $E_1 \times E_2$
- $\sigma_p(E_1)$ ,  $P$  un predicado sobre atributos de  $E_1$
- $\Pi_s(E_1)$ ,  $S$  es una lista de atributos de  $E_1$
- $\rho_x(E_1)$ ,  $x$  es el nuevo nombre para el resultado  $E_1$



## *Operaciones adicionales*

*No dan más potencia, pero simplifican las expresiones*

- *Intersección de conjuntos*
- *Reunión natural*
- *División*
- *Asignación.*

*Se pueden expresar en función de las anteriores*



# Operación intersección

■ *Notación:*  $r \cap s$

■ *Definido como:*

■  $r \cap s = \{ t \mid t \in r \text{ y } t \in s \}$

■ *r y s han de ser compatibles (para la unión):*

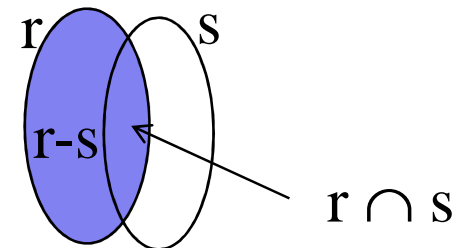
- *r, s tienen el mismo número de atributos*
- *Los atributos de r y s tienen el mismo dominio*

■ *Advertid:*  $r \cap s = r - (r - s)$

3		
4		

1		
2		
3		
4	3	
5	4	
6		

←





# Operación intersección

■ Ejemplo: Nombre de clientes que tienen préstamos y cuentas impositor

nombre cliente	número préstamo
Fernández	P-16
Gómez	P-93
Gómez	P-15
López	P-14
Pérez	P-17
Santos	P-11
Sotoca	P-23
Valdivieso	P-17

nombre cliente	número cuenta
Abril	C-102
Gómez	C-101
González	C-201
González	C-217
López	C-222
Rupérez	C-215
Santos	C-305

$$\Pi_{\text{nombre-cliente}}(\text{prestatarario}) \cap \Pi_{\text{nombre-cliente}}(\text{impositor})$$

nombre-cliente
Gómez
Pérez
Santos



# Operación reunión natural

■ Notación:  $r \bowtie s$

■ Sean  $r$  y  $s$  relaciones con esquema  $R$  y  $S$  respectivamente  
Entonces,  $r \bowtie s$  es una relación en el esquema  $R \cup S$  obtenida como:

- Considerar tuplas  $t_r$  de  $r$  y  $t_s$  de  $s$ .
- Si  $t_r$  y  $t_s$  tienen el mismo valor en cada uno de los atributos de  $R \cap S$ , añadir una tupla  $t$  al resultado donde
  - $t$  tiene el mismo valor que  $t_r$  en  $r$
  - $t$  tiene el mismo valor que  $t_s$  en  $s$

■ Ejemplo:

$R = (A, B, C, D)$

$S = (E, B, D)$

- Esquema resultante =  $(A, B, C, D, E)$
- $r \bowtie s$  se define como:

$$\Pi_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$



## Operación reunión natural

- Definición formal: dos relaciones  $r(R)$  y  $s(S)$ ,  $r \bowtie s$  es una relación del esquema  $R \cup S$  definida por:

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

donde  $R \cap S = \{A_1, A_2, \dots, A_n\}$

- Es asociativa:

$(cliente \bowtie cuenta) \bowtie impositor$   
 $cliente \bowtie (cuenta \bowtie impositor)$

- Si  $R \cap S = \emptyset$ ,  $\Rightarrow r \bowtie s = r \times s$





# Operación reunión natural

■ Ejemplo: Información de todos los préstamos, especificando el nombre del cliente.

prestatario

nombre cliente	número préstamo
Fernández	P-16
Gómez	P-93
Gómez	P-15
López	P-14
Pérez	P-17
Santos	P-11
Sotoca	P-23
Valdivieso	P-17

préstamo

número-préstamo	importe
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500

$\Pi_{\text{nombre-cliente, número-préstamo, importe}} (\text{prestatario} \bowtie \text{préstamo})$

nombre-cliente	número-préstamo	importe
Fernández	P-16	1.300
Gómez	P-23	2.000
Gómez	P-11	900
López	P-15	1.500
Pérez	P-93	500
Santos	P-17	1.000
Sotoca	P-14	1.500
Valdivieso	P-17	1.000



## *Operación reunión zeta*

■ Notación:  $r \bowtie_{\theta} s$

■ Combina selección y producto cartesiano en una operación.

$$r \bowtie_{\theta} s = \sigma_{\theta}(r \times s)$$



## Operación división

- *Válida para las consultas que tienen “para todos”.*
- *Sea  $r$  y  $s$  relaciones en esquemas  $R$  and  $S$  respectivamente cuando*

- $R = (A_1, \dots, A_m, B_1, \dots, B_n)$
- $S = (B_1, \dots, B_n)$

*El resultado de  $r \div s$  es una relación del esquema  $R - S = (A_1, \dots, A_m)$*

*Una tupla  $t$  está en  $r \div s$  si se cumple:*

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

- *Se puede definir en términos del álgebra relacional*

$$r \div s = \Pi_{R-S} (r) - \Pi_{R-S} ( (\Pi_{R-S} (r) \times s) - (r) )$$



## Operación división

- *Ejemplo:* los clientes que tengan abierta una cuenta en *todas* las sucursales ubicadas en Arganzuela

sucursal

nombre de la sucursal	ciudad de la sucursal	activos
Galapagar	Arganzuela	7.500
Centro	Arganzuela	9.000.000
Becerril	Aluche	2.000
Segovia	Cerceda	3.700.000
Navacerrada	Aluche	1.700.000
Navas de la Asunción	Alcalá de Henares	1.500
Moralzarzal	La Granja	2.500
Collado Mediano	Aluche	8.000.000

- 1) Sucursales de Arganzuela

$$r_1 = \Pi_{\text{nombre-sucursal}} (\sigma_{\text{ciudad-sucursal} = \text{«Arganzuela»}} (\text{sucursal}))$$

nombre-sucursal
Centro
Galapagar



## Operación división

### 2) Sucursal de los clientes que tienen cuenta

impositor

nombre cliente	número cuenta
Abril	C-102
Gómez	C-101
González	C-201
González	C-217
López	C-222
Rupérez	C-215
Santos	C-305

cuenta

número-cuenta	nombre-sucursal	saldo
C-101	Centro	500
C-215	Becerril	700
C-102	Navacerrada	400
C-305	Collado Mediano	350
C-201	Galapagar	900
C-222	Moralzarzal	700
C-217	Galapagar	750

$$r_2 = \Pi_{\text{nombre-cliente, nombre-sucursal}} (\text{impositor} \bowtie \text{cuenta})$$

nombre-cliente	nombre-sucursal
Abril	Collado Mediano
Gómez	Becerril
González	Centro
González	Galapagar
López	Navacerrada
Rupérez	Moralzarzal
Santos	Galapagar
Valdivieso	Navacerrada



## Operación división

3) todos los clientes que tengan abierta una cuenta en todas las sucursales ubicadas en Arganzuela

<i>nombre-cliente</i>	<i>nombre-sucursal</i>
Abril	Collado Mediano
Gómez	Becerril
González	Centro
González	Galapagar
López	Navacerrada
Rupérez	Moralzarzal
Santos	Galapagar
Valdivieso	Navacerrada

<i>nombre-sucursal</i>
Centro
Galapagar

$$\Pi_{\text{nombre-cliente, nombre-sucursal}}(\text{impositor} \bowtie \text{cuenta}) \div \Pi_{\text{nombre-sucursal}}(\sigma_{\text{ciudad-sucursal} = \text{«Arganzuela»}(\text{sucursal}))$$

El resultado es una tupla: González en el esquema nombre\_cliente



# Operación de asignación

- Operación de asignación ( $\leftarrow$ ) una manera conveniente de expresar consultas complejas
  - Escribir consulta como un programa secuencial consistiendo de
    - Una serie de asignaciones
    - Seguidos por una expresión cuyo valor se muestra como un resultado de una consulta
  - Asignación se debe de realizar a una variable temporal
- Ejemplo: escribir  $r \div s$  como
$$\begin{aligned}\text{temp1} &\leftarrow \Pi_{R-S}(r) \\ \text{temp2} &\leftarrow \Pi_{R-S}((\text{temp1} \times s) - \Pi_{R-S,S}(r)) \\ \text{result} &= \text{temp1} - \text{temp2}\end{aligned}$$
  - El resultado de la derecha de  $\leftarrow$  se asigna a la variable relación temporal de la izquierda.



## *Operaciones del algebra relacional extendida*

- *Nuevas operaciones* que dan más potencia al algebra relacional:
  - Proyección generalizada
  - Funciones agregadas
  - Reunión externa





# Proyección generalizada

- *Extiende la operación de proyección permitiendo funciones aritméticas en la lista de proyección*

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

- *E expresión del álgebra relacional*
- *Cada  $F_1, F_2, \dots, F_n$  son expresiones aritméticas que involucran constantes y atributos en el esquema de E*

- *Ejemplo:*

## Informacion\_credito

<i>nombre-cliente</i>	<i>límite</i>	<i>saldo-crédito</i>
Gómez	2.000	400
López	1.500	1.500
Pérez	2.000	1.750
Santos	6.000	700

$\Pi_{\text{nombre-cliente, límite - saldo-credito}}$  as crédito-disponible  
(informacion\_credito)

<i>nombre-cliente</i>	<i>crédito-disponible</i>
Gómez	1.600
López	0
Pérez	250
Santos	5.300



# *Funciones agregadas y operaciones*

- **Función agregada** toma un conjunto de valores y devuelve uno sólo.

**avg:** valor medio

**min:** valor mínimo

**max:** valor máximo

**sum:** suma de valores

**count:** número de valores

- **Operación agregada en el álgebra relacional**

$G_1, G_2, \dots, G_n \quad \mathbf{g} \quad F_1(A_1), F_2(A_2), \dots, F_n(A_n) (E)$

- E es una expresión
- $G_1, G_2, \dots, G_n$  lista de atributos en los cuales se agrupa (puede ser vacío)
- Cada  $F_i$  es una función agregada
- cada  $A_i$  es un nombre de atributo



# *Funciones agregadas y operaciones*

■ **Ejemplo:** trabajo-por-horas

<i>nombre-empleado</i>	<i>nombre-sucursal</i>	<i>sueldo</i>
González	Centro	1.500
Díaz	Centro	1.300
Jiménez	Centro	2.500
Catalán	Leganés	1.600
Cana	Leganés	1.500
Cascallar	Navacerrada	5.300
Fernández	Navacerrada	1.500
Ribera	Navacerrada	1.300

$G_{\text{sum}(\text{sueldo})}(\text{trabajo-por-horas})$

■ **Resultado: una tupla de valor 16.500**



# Funciones agregadas y operaciones

- **Borrar los valores duplicados antes de aplicar la función de agregación  $\Rightarrow$  distinct**

$G_{\text{count-distinct}(\text{nombre-sucursal})}(\text{trabajo-por-horas})$

- **Resultado: una tupla de valor 3**
- **Realizar grupos por nombre\_sucursal**

nombre-empleado	nombre-sucursal	sueldo
González	Centro	1.500
Díaz	Centro	1.300
Jiménez	Centro	2.500
Catalán	Leganés	1.600
Cana	Leganés	1.500
Cascallar	Navacerrada	5.300
Fernández	Navacerrada	1.500
Ribera	Navacerrada	1.300

$\text{nombre-sucursal } G_{\text{sum}(\text{sueldo})}(\text{trabajo-por-horas})$

nombre-sucursal	suma de sueldos
Centro	5.300
Leganés	3.100
Navacerrada	8.100



# Funciones agregadas

- *Resultado de agregación no tiene nombre*
  - Se puede usar la operación de renombramiento
  - Se puede permitir el renombramiento de una función agregada

*nombre-sucursal*  $G_{\text{sum}(\text{sueldo})}$  as *suma-sueldo*,  $\text{max}(\text{sueldo})$  as  
*sueldo-máximo* (*trabajo-por-horas*)

<i>nombre-sucursal</i>	<i>suma-sueldo</i>	<i>sueldo-máximo</i>
Centro	5.300	2.500
Leganés	3.100	1.600
Navacerrada	8.100	5.300



## *Reunión externa*

- *Una extensión de la operación de reunión que evita la pérdida de información*
- *Realiza la reunión y añade las tuplas de una relación que no coincide con el atributo de la reunión*
- *Usa valores nulos (null):*
  - *null significa valor desconocido o que no existe*
  - *todas las comparaciones en que interviene null son false por definición*
- *Por la izquierda, por la derecha y completa*



# *Reunión externa*

## ■ *Ejemplo empleado*

<i>nombre-empleado</i>	<i>calle</i>	<i>ciudad</i>
Segura	Tebeo	La Loma
Domínguez	Viaducto	Villaconejos
Gómez	Bailén	Alcorcón
Valdivieso	Fuencarral	Móstoles

## *trabajo-a-tiempo-completo*

<i>nombre-empleado</i>	<i>nombre-sucursal</i>	<i>suelo</i>
Segura	Majadahonda	1.500
Domínguez	Majadahonda	1.300
Barea	Fuenlabrada	5.300
Valdivieso	Fuenlabrada	1.500

## ■ *Reunión empleado ⋈ trabajo-a-tiempo-completo*

<i>nombre-empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre-sucursal</i>	<i>suelo</i>
Segura	Tebeo	La Loma	Majadahonda	1.500
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500

## ■ *R externa completa (⋈) empleado ⋈ trabajo-a-tiempo-completo*

<i>nombre-empleado</i>	<i>calle</i>	<i>ciudad</i>	<i>nombre-sucursal</i>	<i>suelo</i>
Segura	Tebeo	La Loma	Majadahonda	1.500
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500
Gómez	Bailén	Alcorcón	<i>nulo</i>	<i>nulo</i>
Barea	<i>nulo</i>	<i>nulo</i>	Fuenlabrada	5.300



## Reunión externa

### ■ Reunión externa por la izquierda: (⌞)

empleado

nombre-empleado	calle	ciudad
Segura	Tebeo	La Loma
Domínguez	Viaducto	Villaconejos
Gómez	Bailén	Alcorcón
Valdivieso	Fuencarral	Móstoles

trabajo-a-tiempo-completo

nombre-empleado	nombre-sucursal	sueldo
Segura	Majadahonda	1.500
Domínguez	Majadahonda	1.300
Barea	Fuenlabrada	5.300
Valdivieso	Fuenlabrada	1.500

empleado ⌞ trabajo-a-tiempo-completo

nombre-empleado	calle	ciudad	nombre-sucursal	sueldo
Segura	Tebeo	La Loma	Majadahonda	1.500
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500
Gómez	Bailén	Alcorcón	nulo	nulo

### ■ Reunión externa por la derecha (⌞)

empleado ⌞ trabajo-a-tiempo-completo

nombre-empleado	calle	ciudad	nombre-sucursal	sueldo
Segura	Tebeo	La Loma	Majadahonda	1.500
Domínguez	Viaducto	Villaconejos	Majadahonda	1.300
Valdivieso	Fuencarral	Móstoles	Fuenlabrada	1.500
Barea	nulo	nulo	Fuenlabrada	5.300





## *Valores nulos*

- *Es posible tener valores nulos en ciertas tuplas para ciertos atributos*
- *null significa valor desconocido o que no existe*
- *El resultado de una expresión aritmética que conlleva null es null.*
- *Las funciones agregadas ignoran esos valores*
- *Para eliminación de duplicados y agrupamiento, null se trata como otro valor y dos nulos son el mismo valor.*



# Valores nulos

- *Comparaciones con valor nulo devuelven nulo*
- *Operadores OR, AND, NOT null = unknown:*
  - OR: (unknown **or** true) = true,  
(unknown **or** false) = unknown  
(unknown **or** unknown) = unknown
  - AND: (true **and** unknown) = unknown,  
(false **and** unknown) = false,  
(unknown **and** unknown) = unknown
  - NOT: (**not** unknown) = unknown
  - En SQL “P **is unknown**” puede ser true si P es unknown
- $\sigma_p(\mathbf{E})$  si p es falso o unknown no añade la tupla
- Reunión como la selección
- Proyección como otro valor
- Unión, Intersección, Diferencia como otro valor



# Valores nulos

## ■ *Proyección generalizada*

- *Tuplas duplicadas con valores nulos como en la proyección*
- *En operaciones agregadas  $\Rightarrow$  se borran los nulos antes de la agregación*

## ■ *Reunión externa*

- *Como las operaciones de reunión.*
- *Excepto con las tuplas que no aparecen en el resultado*
- *Se añaden con nulos dependiendo si es: izquierda, derecha o total.*



## *Borrado, Inserción y Modificación*

- Borrado se expresa en el álgebra relacional como:

$$r \leftarrow r - E$$

*donde  $r$  es la relación y  $E$  una expresión*

*Sólo se pueden borrar tuplas enteras. No valores de atributos*

*Ejemplo:*

$$\begin{aligned} \text{impositor} &\leftarrow \text{impositor} - \sigma_{\text{nombre-cliente}} \\ &= \text{«Gómez»}(\text{impositor}) \end{aligned}$$

- Inserción:

$$r \leftarrow r \cup E$$

*Ejemplo:*

$$\begin{aligned} \text{cuenta} &\leftarrow \text{cuenta} \cup \{(C-973, \text{«Navacerrada»}, 1200)\} \\ \text{impositor} &\leftarrow \text{impositor} \cup \{(\text{«Gómez»}, C-973)\} \end{aligned}$$



## Borrado, Inserción y Modificación

$$\begin{aligned}r_1 &\leftarrow (\sigma_{\text{nombre-sucursal} = \text{«Navacerrada»}}(\text{prestatario} \bowtie \text{préstamo})) \\r_2 &\leftarrow \Pi_{\text{nombre-sucursal}, \text{número-préstamo}}(r_1) \\ \text{cuenta} &\leftarrow \text{cuenta} \cup (r_2 \times \{(200)\}) \\ \text{impositor} &\leftarrow \text{impositor} \cup \Pi_{\text{nombre-cliente}, \text{número-préstamo}}(r_1)\end{aligned}$$

■ **Modificación:**

$$r \leftarrow \Pi_{F_1, F_2, \dots, F_n}(r)$$

donde  $F_i$  son expresiones que involucran constantes y atributos de  $r$

Para varias tuplas de  $r$ :

$$r \leftarrow \Pi_{F_1, F_2, \dots, F_n}(\sigma_P(r)) \cup (r - \sigma_P(r))$$

**Ejemplo:**

$$\text{cuenta} \leftarrow \Pi_{\text{nombre-sucursal}, \text{número-cuenta}, \text{saldo}, \text{saldo} * 1.05}(\text{cuenta})$$

$$\begin{aligned}\text{cuenta} &\leftarrow \Pi_{NS, NC, \text{saldo} * 1.06}(\sigma_{\text{saldo} > 10000}(\text{cuenta})) \cup \\ \text{cuenta} &\leftarrow \Pi_{NS, NC, \text{saldo} * 1.05}(\sigma_{\text{saldo} \leq 10000}(\text{cuenta}))\end{aligned}$$



# Vistas

- *Hasta ahora se ha operado en el nivel lógico (relaciones)*
- *En algunos casos no es deseable que todos los usuarios vean todo el modelo lógico de la base de datos*
- *Vista: relación que no forma parte del modelo conceptual pero que se hace visible al usuario como una relación virtual*

***create view v as <Expresión de consulta>***

*donde <Expresión de consulta> es cualquier expresión de consulta legal del álgebra relacional.*

```
create view todos-los-clientes as  
   $\Pi_{\text{nombre-sucursal, nombre-cliente}} (\text{impositor} \bowtie \text{cuenta})$   
   $\cup \Pi_{\text{nombre-sucursal, nombre-cliente}} (\text{prestatario} \bowtie \text{préstamo})$ 
```



# Vistas

- Una vez creada, se puede utilizar

$\Pi_{\text{nombre-cliente}} (\sigma_{\text{nombre-sucursal} = \text{«Navacerrada»}} (\text{todos-los-clientes}))$

- No se guarda el resultado
- Se guarda la definición de la vista
- Algunos SBGD permiten guardar el resultado  $\Rightarrow$  vistas materializadas/mantenimiento de vistas/instantanea/snapshot
- Actualizaciones sobre vistas  $\Rightarrow$  sobre las relaciones reales

**create view** préstamo-sucursal as  $\Pi_{\text{nombre-sucursal, número-préstamo}} (\text{préstamo})$        $\text{préstamo-sucursal} \leftarrow \text{préstamo-sucursal} \cup \{(P-37, \text{«Navacerrada»})\}$

- Se puede permitir, pero la tupla sería: (P-37, Navacerrada, null)
- No se puede permitir



# Vistas

## ■ Otro problema:

**create view** *información-crédito* **as**  $\Pi_{\text{nombre-cliente, importe}}(\text{prestatario} \bowtie \text{préstamo}) \cup \{(\text{«González»}, 1900)\}$

- *Habría que insertar: (González,nulo) y (nulo,nulo,1900)*
- *No se consigue la tupla deseada (González,1900)*
- *Generalmente no se permite actualización sobre vistas*
- *Se puede definir vistas sobre otras vistas, pero sin recursividad*

**create view** *todos-los-clientes* **as**  
 $\Pi_{\text{nombre-sucursal, nombre-cliente}}(\text{impositor} \bowtie \text{cuenta})$   
 $\cup \Pi_{\text{nombre-sucursal, nombre-cliente}}(\text{prestatario} \bowtie \text{préstamo})$

**create view** *cliente-navacerrada* **as**  
 $\Pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal} = \text{«Navacerrada»}}(\text{todos-los-clientes}))$





# Vistas

## ■ Procedimiento expansión de vistas

$$\sigma_{\text{nombre-cliente} = \text{«Martín»}}(\text{cliente-navacerrada})$$

$$\sigma_{\text{nombre-cliente} = \text{«Martín»}}(\Pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal} = \text{«Navacerrada»}}(\text{todos-los-clientes})))$$

$$\sigma_{\text{nombre-cliente} = \text{«Martín»}}(\Pi_{\text{nombre-cliente}}(\sigma_{\text{nombre-sucursal} = \text{«Navacerrada»}}(\Pi_{\text{nombre-sucursal, nombre-cliente}}(\text{impositor} \bowtie \text{cuenta}) \cup \Pi_{\text{nombre-sucursal, nombre-cliente}}(\text{prestatarario} \bowtie \text{préstamo}))))))$$



# *Cálculo relacional de tuplas*

- *Lenguaje de consultas no procedimental*

$$\{t \mid P(t)\}$$

- *Todas las tuplas  $t$  que cumplen el predicado  $P$*
- *$t[A]$ , valor de la tupla en el atributo  $A$*
- *$t \in r$ , tupla  $t$  contenida en la relación  $r$*
- *Ejemplo:*

$$\{t \mid t \in \text{préstamo} \wedge t[\text{importe}] > 1200\}$$

- *Constructor  $\exists \Rightarrow \exists t \in r (Q(t))$*

$$\{t \mid \exists s \in \text{préstamo} (t[\text{número-préstamo}] = s[\text{número-préstamo}] \wedge s[\text{importe}] > 1200)\}$$

- *La tupla  $t$  tiene solo el atributo `numero_prestamo`*



# Cálculo relacional de tuplas

## ■ Consultas con dos relaciones

- Operador  $\wedge$  (y) Préstamos de Navacerrada

$$\{t \mid \exists s \in \text{prestatario} (t[\text{número-préstamo}] = s[\text{número-préstamo}] \wedge \exists u \in \text{préstamo} (u[\text{número-préstamo}] = s[\text{número-préstamo}] \wedge u[\text{nombre-sucursal}] = \text{«Navacerrada»}))\}$$

- Operador  $\vee$  (o): Nombre de todos los clientes (impositores y prestatarios)

$$\{t \mid \exists s \in \text{prestatario} (t[\text{nombre-cliente}] = s[\text{nombre-cliente}]) \vee \exists u \in \text{impositor} (t[\text{nombre-cliente}] = u[\text{nombre-cliente}])\}$$

- Operador  $\neg$  (no): Nombre de los clientes que no tienen préstamos

$$\{t \mid \exists u \in \text{impositor} (t[\text{nombre-cliente}] = u[\text{nombre-cliente}]) \wedge \neg \exists s \in \text{prestatario} (t[\text{nombre-cliente}] = s[\text{nombre-cliente}])\}$$



## Cálculo relacional de tuplas

### ■ Consultas con dos relaciones.

- Operador  $\Rightarrow$ ,  $P \Rightarrow Q$  equivale a  $\neg (P) \vee Q$
- Constructor  $\forall$

$$\forall t \in r (Q(t))$$

- *Determinar todos los clientes que tienen una cuenta en todas las sucursales de Arganzuela.*
- *Reformulado, determinar los clientes tales que no existe una sucursal en Arganzuela, en la cual no tengan cuenta*

$$\{t \mid \exists r \in \text{cliente} (r[\text{nombre-cliente}] = t[\text{nombre-cliente}] \wedge (\forall u \in \text{sucursal} (u[\text{ciudad-sucursal}] = \text{«Arganzuela»} \Rightarrow \exists s \in \text{impositor} (t[\text{nombre-cliente}] = s[\text{nombre-cliente}] \wedge \exists w \in \text{cuenta} (w[\text{número-cuenta}] = s[\text{número-cuenta}] \wedge w[\text{nombre-sucursal}] = u[\text{nombre-sucursal}]))))))\}$$

## Diapositiva 52

---

**R1**

Determinar todos los clientes que tienen una cuenta en todas las sucursales de Arganzuela.  
Es decir, no existe una sucursal en Arganzuela para la que no tengan cuenta  
Rpr; 28/03/2011



# Cálculo relacional de tuplas

## ■ Definiciones

$$\{t \mid P(t)\}$$

- $P$  es una fórmula

$$t \in \text{préstamo} \wedge \exists s \in \text{cliente} (t[\text{nombre-sucursal}] = s[\text{nombre-sucursal}])$$

- $t$  es una variable libre
- $s$  es una variable ligada ( con  $\exists$  ó  $\forall$  )
- Las fórmulas se construyen con átomos:
  - $s \in r$
  - $s[x] \text{ op } u[y]$  , donde  $\text{op}$  es operador comparación.
  - $s[x] \text{ op } \text{cte}$  , donde  $\text{cte}$  es un valor del dominios de  $x$
- Fórmulas con formulas son fórmulas:  $P_1 \wedge P_2$



# Cálculo relacional de tuplas

## ■ Seguridad de las expresiones

- Las expresiones del cálculo relacional pueden generar expresiones infinitas

$$\{t \mid \neg (t \in \text{préstamo})\}$$

- Infinitas tuplas que no están en préstamo  $\Rightarrow$  dominio de la fórmula,  $\text{dom}(P)$
- $\text{dom}(P) \Rightarrow$  conjunto de todos los valores a los que  $P$  hace referencia

$$\text{dom}(t \in \text{préstamo} \wedge t[\text{importe}] > 1200)$$

- $\text{dom}(\neg (t \in \text{préstamo}))$ , conjunto de todos los valores que aparecen en préstamo
- $\{t \mid P(t)\}$  es segura, si todos los valores que aparecen en el resultado pertenecen a  $\text{dom}(P)$



## *Cálculo relacional de dominios*

- *Utiliza variables dominio  $\Rightarrow$  los valores del dominio de los atributos*
- *Relacionado con el cálculo relacional de tuplas*

$$\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$$

- *$x_1, x_2, \dots, x_n$  son variable dominio y  $P$  una fórmula*
- *Ejemplos:*

$$\{ \langle p, s, i \rangle \mid \langle p, s, i \rangle \in \text{préstamo} \wedge i > 1200 \}$$

$$\{ \langle p \rangle \mid \exists s, i (\langle p, s, i \rangle \in \text{préstamo} \wedge i > 1200) \}$$





## *Cálculo relacional de dominios*

$$\{ \langle n, c \rangle \mid \exists l (\langle n, p \rangle \in \text{prestatario} \wedge \exists s (\langle p, s, i \rangle \in \text{préstamo} \wedge s = \text{«Navacerrada»})) \}$$

$$\{ \langle n \rangle \mid \exists p (\langle n, p \rangle \in \text{prestatario} \wedge \exists s, i (\langle p, s, i \rangle \in \text{préstamo} \wedge s = \text{«Navacerrada»})) \vee \exists c (\langle n, c \rangle \in \text{impositor} \wedge \exists s, i (\langle c, s, i \rangle \in \text{cuenta} \wedge s = \text{«Navacerrada»})) \}$$

$$\{ \langle c \rangle \mid \exists s, t (\langle c, s, t \rangle \in \text{cliente}) \wedge \forall x, y, z (\langle x, y, z \rangle \in \text{sucursal}) \wedge y = \text{«Arganzuela»} \Rightarrow \exists a, b (\langle x, a, b \rangle \in \text{cuenta} \wedge (\langle c, a \rangle \in \text{impositor})) \}$$



# Cálculo relacional de dominios

## ■ Seguridad de las expresiones:

1. Todos los valores que aparecen en las tuplas de la expresión son valores de  $dom(P)$ .
2. Para cada subfórmula «existe» de la forma  $\exists x (P_1(x))$ , la subfórmula es cierta si y sólo si hay un valor  $x$  en  $dom(P_1)$  tal que  $P_1(x)$  es verdadero.
3. Para cada subfórmula «para todo» de la forma  $\forall x (P_1(x))$ , la subfórmula es verdadera si y sólo si  $P_1(x)$  es verdadero para todos los valores  $x$  de  $dom(P_1)$ .