

## Material para a formación profesional inicial

### A03. Operacións do MR: álgebra e cálculo

Familia profesional	IFC	Informática e comunicacións
Ciclo formativo	CSIFC03 CSIFC02	Desenvolvemento de aplicacións web Desenvolvemento de aplicacións multiplataforma
Grao		Superior
Módulo profesional	MP0484	Bases de datos
Unidade didáctica	UD03	Deseño lóxico de base de datos
<b>Actividade</b>	<b>A03</b>	<b>Operacións do MR: álgebra e cálculo</b>
Autores		Marta Fernández García María del Carmen Fernández Lameiro Miguel Fraga Vila María Carmen Pato González Andrés del Río Rodríguez
Nome do arquivo		MP0484_V000303_UD03_A03_operAlxCal.docx
<p>© 2015 Xunta de Galicia. Consellería de Cultura, Educación e Ordenación Universitaria.</p> <p>Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón <a href="http://creativecommons.org/licenses/by-nc-sa/3.0/es/">http://creativecommons.org/licenses/by-nc-sa/3.0/es/</a>.</p>		



<b>1. A03. Operacións do MR: álgebra e cálculo .....</b>	<b>6</b>
1.1	Introdución .....6
1.2	Actividade.....6
1.2.1	Fundamentos da dinámica do modelo relacional.....6
1.2.2	Álgebra relacional .....6
1.2.2.1	Operación de asignación ( $\leftarrow$ ) é/ou renomear ( $\rho$ ) .....8
1.2.2.2	Operadores primitivos ou básicos.....8
	Unarias (Un operando).....8
	Restrición ou selección ( $\sigma$ ) .....8
	Proxección( $\Pi$ ).....11
	Binarias (Dous operandos) .....12
	Unión ( $\cup$ ).....12
	Diferenza ( $-$ ).....14
	Produto cartesiano ( $\times$ ) .....14
1.2.2.3	Operadores derivados.....16
	Intersección ( $\cap$ ).....17
	División ( $\div$ ) .....17
	Combinación, reunión ou join ( $\bowtie$ ).....18
	Reunión natural, produto natural ou natural join ( $*$ ) .....21
1.2.2.4	Resumo de propiedades dos operadores relacionais .....22
1.2.2.5	Operacións adicionais.....22
	Peche recursivo .....22
	Proxección xeneralizada .....25
	Agrupación e agregación.....25
	Reunión externa (outer-join) .....26
	Reunión externa esquerda ( $/^*$ ).....27
	Reunión externa dereita ( $^*/$ ) .....27
	Reunión externa completa ( $/^*/$ ).....28
1.2.2.6	Correspondencia da álgebra relacional coa linguaxe de consulta SQL.....29
1.2.3	Cálculo relacional.....30

1.2.3.1	Cálculo relacional de tuplas (CRT).....	30
	Expresión Xeral do CRT .....	31
	Os átomos.....	31
	Fórmula (ou condición) ben formada (fbf) .....	33
	Variable de tupla libre e ligada nunha fbf .....	33
	Fórmulas pechadas e abertas .....	34
	Expresións seguras .....	34
	Resultado dunha CRT .....	34
	Correspondencia do cálculo relacional coa linguaxe de consulta SQL .....	35
1.2.3.2	Cálculo relacional de dominios (CRD).....	35
1.2.4	Linguaxes relacionais comerciais .....	36
1.2.4.1	SQL.....	36
1.2.4.2	SQL dinámico .....	36
1.2.4.3	QUEL .....	36
1.2.4.4	QBE .....	36
1.3	Tarefas.....	36
1.3.1	Tarefa 1. Realizar consultas empregando o operador de selección .....	38
	Solución.....	38
1.3.2	Tarefa 2. Realizar consultas empregando o operador de proxección .....	38
	Solución.....	38
1.3.3	Tarefa 3. Realizar consultas empregando o operador de unión .....	39
	Solución.....	39
1.3.4	Tarefa 4. Realizar consultas empregando o operador de diferenza.....	39
	Solución.....	39
1.3.5	Tarefa 5. Realizar consultas empregando o operador de intersección .....	39
	Solución.....	39
1.3.6	Tarefa 6. Realizar consultas empregando o operador de división .....	40
	Solución.....	40
1.3.7	Tarefa 7. Realizar consultas empregando o operador de combinación .....	40
	Solución.....	40
1.3.8	Tarefa 8. Realizar consultas empregando o operador de combinación natural .....	41
	Solución.....	41
1.3.9	Tarefa 9. Realizar consultas empregando a agrupación e a agregación .....	43
	Solución.....	43

1.3.10	Tarefa 10 Realizar consultas empregando as reunións externas .....	44
	Solución:.....	45
1.3.11	Tarefa 11 Realizar consultas empregando cálculo relacional de tuplas .....	45
	Solución.....	46
<b>2.</b>	<b>Materiais .....</b>	<b>47</b>
2.1	Documentos de apoio ou referencia .....	47

# 1. A03. Operacións do MR: álgebra e cálculo

---

## 1.1 Introducción

Os obxectivos desta actividades son:

- Explicar as características e operacións que proporciona a álgebra e o cálculo relacional.
- Mostrar o paralelismo entra as linguaxes formais de álgebra e cálculo relacional e a linguaxe declarativa SQL.

## 1.2 Actividade

### 1.2.1 Fundamentos da dinámica do modelo relacional

O modelo relacional (MR), como todo modelo de datos (MD), leva asociado unha parte estática (estrutura e restricións) e unha dinámica que permite a transformación entre estados da base de datos. Esta transformación dun estado orixe a un estado obxectivo realízase aplicando un conxunto de operadores, mediante os cales se levan a cabo operacións de inserción, borrado, modificación e consulta de tuplas.

Se  $OP$  é un operador, o paso dun estado orixe da base de datos ( $BDi$ ) a un estado obxectivo ( $BDj$ ) pódese expresar como:  $OP(BDi) = BDj$ , ambos estados deben cumprir as restricións de integridade estáticas, e a transformación debe cumprir as restricións de integridade dinámicas (entre estados).

As linguaxes relacionais operan sobre conxuntos de tuplas, é dicir, non son linguaxes navegacionais (que manipulan rexistros, como C#, Java, etc.) senón de especificación. Divídense en dous tipos:

- **Algebraicos (álgebra relacional):** os cambios de estado especificanse mediante operacións, con operandos como relacións e o seu resultado é outra relación.
- **Predicativos (cálculo relacional):** os cambios de estado especificanse mediante predicados que definen o estado obxectivo sen indicar as operacións que hai que realizar para chegar ao mesmo. Divídense en dous subtipos:
  - orientados a tuplas.
  - orientados a dominios.

### 1.2.2 Álgebra relacional

A álgebra relacional (AR) constitúe a dinámica do modelo relacional (MR), foi definida por Codd en 1972, está composta por unha colección de operadores que toman relacións como operandos e devolven relacións como resultado (clausura relacional). Os operandos son relacións, e non conxuntos arbitrarios; as operacións baseadas na teoría de conxuntos están adaptadas ás relacións que son un tipo especial de conxuntos.

Sexan  $R_1, R_2, \dots, R_i$  e  $R_r$  relacións e  $OP$  un operador do AR, unha operación consiste en aplicar  $OP$  a(s) relación(s)  $R_1, \dots, R_i$ , obténdose  $R_r$ :  $OP(R_1 \dots R_i) = R_r$ .

Ao ser o resultado da operación outra relación, cúmprese a propiedade de cerre, é dicir, si  $OP_1 \dots OP_n$  representan operadores, cúmprese:  $OP_n ( \dots (OP_1 ( R ) ) ) = R_r$ .

Nesta propiedade de clausura relacional permítese aplicar unha operación tras outra establecéndose que a saída dunha operación pode ser entrada (operando) doutra e as expresións poden aniñarse formando a súa vez unha expresión con operandos que conteñen a súa vez outra expresión da álgebra en vez de nomes de relación.

Sexan  $R_1, R_2, R_3$ , relacións de tipos compatibles, podendo indicar:

- Unha única expresión dando lugar a expresións aniñadas:  $R \cap (S \cup T)$ .
- Varias expresións establecendo relacións intermedias con nome.
  - $A \leftarrow S \cup T$ .
  - $B \leftarrow R \cap A$ .

Os operadores do AR son 8, ver figura 3.1, e pódense clasificar de tres formas diferentes:

- **Segundo a súa orixe:**
  - Procedentes da teoría de conxuntos: unión, intersección, diferenza e produto cartesiano.
  - Relacionais especiais: restrición, proxección, combinación e división.
- **Segundo a completitude da linguaxe:**
  - Primitivos: operadores esenciais que non poden obterse doutros (sen eles, o AR non sería unha linguaxe completa).
  - Derivados: pódense obter aplicando varios dos operadores primitivos.
  - Aínda que se pode prescindir deles, son útiles para simplificar moitas operacións habituais.
- **Segundo o número de operandos:**
  - Unarios: actúan sobre unha única relación.
  - Binarios: o operador ten dúas relacións como operandos.

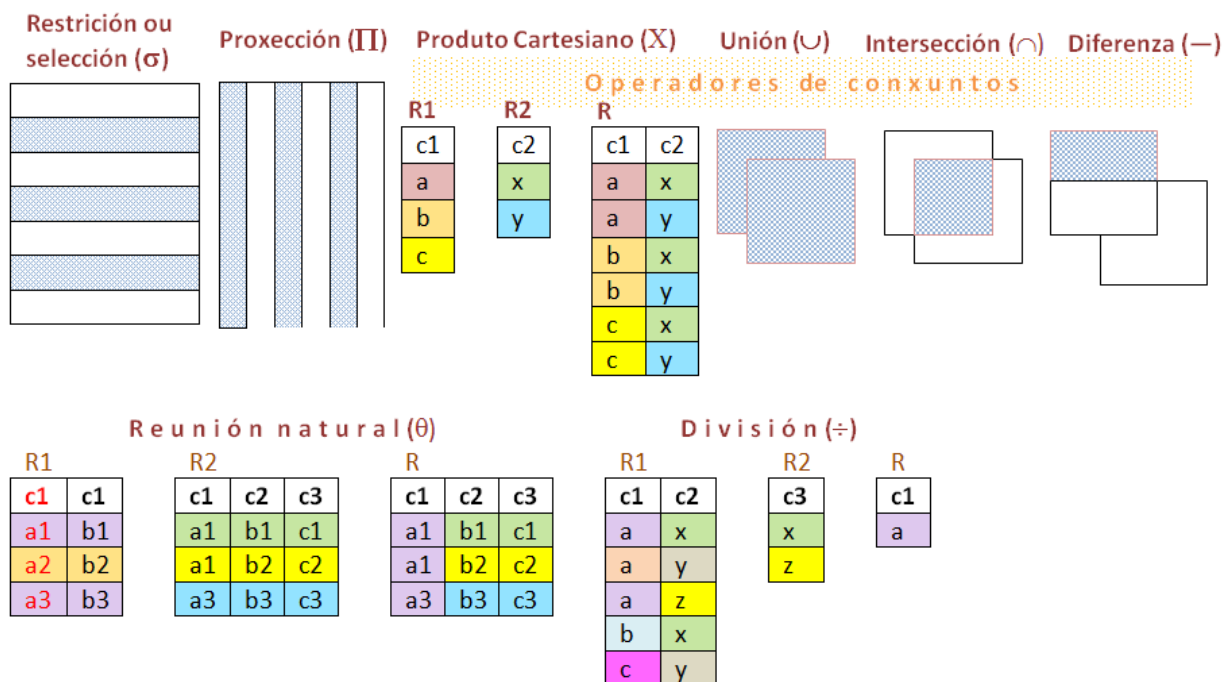


Figura 3.1. Operadores da álgebra relacional

### 1.2.2.1 Operación de asignación ( $\leftarrow$ ) é/ou renomear ( $\rho$ )

Pódense facer renomeamentos coma expresións:

- Renomear atributos:  $\rho B_1, B_2, \dots, B_n (R)$ , renomear atributos da relación R a  $B_1, B_2, \dots, B_n$ .
- Renomear só algúns atributos:  $\rho A_1/B_1, A_2/B_2, \dots, A_k/B_k (R)$ .
- Renomear esquemas e atributos:  $\rho S (B_1, B_2, \dots, B_n) (R)$ .

O que se refire á asignación, trátase dunha operación auxiliar que se utiliza para almacenar o resultado dunha consulta nunha nova relación ou para denominar resultados intermedios cando se desexa dividir unha operación complexa nunha secuencia de operacións máis simples. O símbolo para representala é unha frecha: **RELACION\_NOVA  $\leftarrow$  OP(R)**.

### 1.2.2.2 Operadores primitivos ou básicos.

#### Unarias (Un operando)

##### Restrición ou selección ( $\sigma$ )

A selección dunha relación aplícase soamente a unha única relación e mediante unha expresión lóxica (predicado de selección) dá como resultado unha relación formada polo subconxunto de tuplas que cumpre dita condición de selección.

Sintaxe:  $\sigma <\text{condición}> (<\text{RELACION}>)$ :

- **<condición>**: é unha expresión booleana (verdadeiro ou falso), especificada da seguinte forma:
  - Unha comparación simple  $O1 \text{ op } O2$  ou ben  $O1 \text{ op } C$  onde:
    - $OP1$  e  $OP2$  son atributos,  $C$  é unha constante, (valor pertencente o dominio do atributo sobre o que se compara ) e  $OP$  é un operador de comparación:  $\{=, <, \leq, >, \geq, \neq\}$ .
    - Operadores booleanos AND, OR, NOT aplicados a outras condicións formando unha cláusula.

O resultado será unha relación R (conxunto de tuplas) cos atributos de <relación> onde:

- O grao da relación resultado será o mesmo que o da relación orixe, é dicir, a relación orixe e a resultado teñen os mesmos atributos.
- O número de tuplas da relación resultado pode ser menor ou igual que o número de tuplas da relación orixe.

A selección constitúe o mecanismo de selección do sistema, aplicando <condición> a cada tupla individual de <relación>, substituíndo cada atributo polo seu valor na tupla; e se a <condición> é certa, a tupla selecciónase para o resultado.

No seguinte exemplo móstrase o contido máis o grafo relacional dunha base de datos que se emprega nun casting teatral de actores para o musical “A guerra das galaxias: unha nova esperanza”.



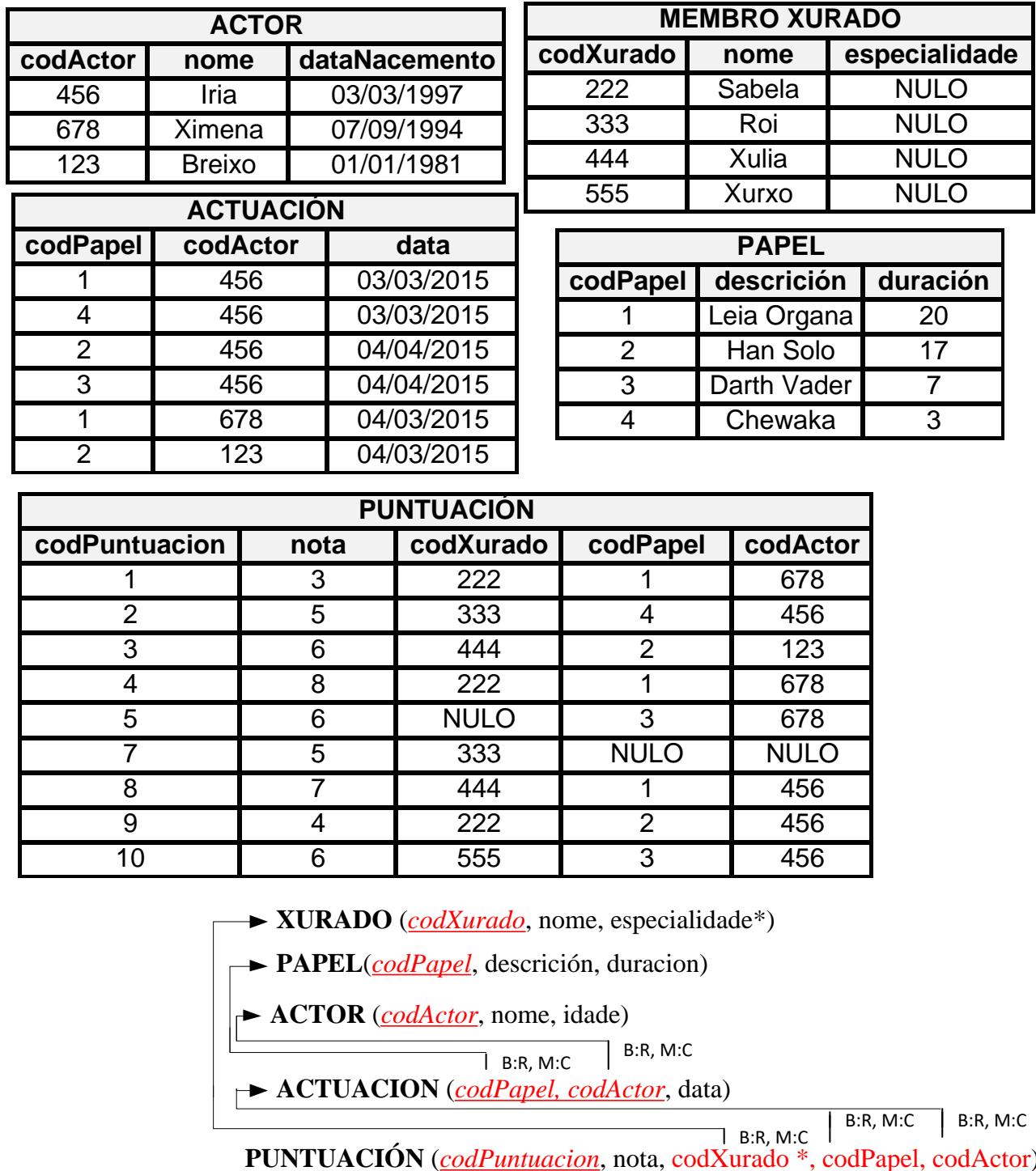


Figura 3.2. Grafo relacional e contido dunha base de datos

Supoñamos as seguintes consultas

- Tuplas de actores que se presentan para o papel de “Leia Organa”:

ACTUACIÓN		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

- $\sigma(\text{codPapel}=1)$  (ACTUACIÓN).
- Actores maiores de 20 anos:

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981

- $\sigma(\text{datNacemento}>01/01/1996)$  (ACTOR).
- Actores que tras realizar a proba para a personaxe de “Leia Organa” obtiveron unha nota igual ou superior a 5:

PUNTUACIÓN				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

- $\sigma((\text{codPApel} = 1)\text{AND}(\text{nota} \geq 5))$  (PUNTUACION).

A operación restrición é conmutativa,  $\sigma_{cond1}(\sigma_{cond2}(R)) \equiv \sigma_{cond2}(\sigma_{cond1}(R))$ , isto permite que a secuencia de restricións (selección) se realice en calquera orde e combinar unha secuencia de restricións nunha única restrición cunha condición conxuntiva (AND):  
 $\sigma_{cond1}(\sigma_{cond2}(\dots(\sigma_{condn}(R))\dots)) \equiv \sigma_{cond1} \text{ AND } cond2 \text{ AND } \dots \text{ AND } condn(R)$ .



Tarefa 1. Realizar consultas empregando o operador de selección.

### Proxección( $\Pi$ )

A proxección dunha relación sobre un conxunto dos seus atributos é outra relación definida sobre estes atributos, eliminando as tuplas duplicadas que puidesen resultar. Esta operación aplícase cando só interesan algúns atributos dunha relación e proxéctase a relación sobre eses atributos.

Restrición vs. Proxección :

- A selección ou restrición ( $\sigma$ ) selecciona algunhas tuplas da relación e descarta outras
- A proxección ( $\Pi$ ) selecciona certos atributos e descarta os demais

Sintaxe:  $\Pi\langle\text{listAtrib}\rangle(\langle\text{RELACION}\rangle)$

- $\langle\text{listAtrib}\rangle$ : lista de nomes de atributos de  $\langle\text{relación}\rangle$ . Se  $\langle\text{listAtrib}\rangle$  non contén atributos claves, as tuplas resultantes poderían estar duplicadas, no exemplo, obter “codXurado” e “codPapel” da relación PUNTUACION, o que obrigaría a unha eliminación implícita de duplicados para que o resultado sexa unha relación válida.

PUNTUACIÓN				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

–  $\Pi\langle\text{codXurado}, \text{codPapel}\rangle(\text{PUNTUACION})$ .

O resultado é unha relación R (conxunto de tuplas) cos atributos soamente de  $\langle\text{listAtrib}\rangle$  e en esa orde:

- O grado da relación resultado é igual ao número de atributos da lista de atributos especificada ( $\langle\text{listAtrib}\rangle$ ).
- O número de tuplas da relación resultado pode ser menor ou igual que o número de tuplas da relación orixe e sería igual se a lista de atributos indicada  $\langle\text{listAtrib}\rangle$  contén unha clave candidata.

Outro exemplo sería recuperar os códigos dos actores que xa foron puntuados no casting:

PUNTUACIÓN				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

A operación proxección non é conmutativa o que significa que  $\Pi_{\text{lista1}} (\Pi_{\text{lista2}} (R)) \neq \Pi_{\text{lista2}} (\Pi_{\text{lista1}} (R))$

Ademais, se sempre que  $\text{lista1} \subseteq \text{lista2}$ , entón  $\Pi_{\text{lista1}} (\Pi_{\text{lista2}} (R)) = \Pi_{\text{lista1}} (R)$ .



Tarefa 2. Realizar consultas empregando o operador de proxección

## Binarias (Dous operandos)

### Unión ( $\cup$ )

A operación de unión baséase na operación matemática de conxuntos na que  $A \cup B = \{ e / e \in A \text{ y-o } e \in B \}$ , a unión de dúas relacións R1 e R2, con esquemas R1 e R2 compatibles, é outra relación definida sobre o mesmo esquema de relación, nas que a súa extensión estará constituída polo conxunto de tuplas que pertencen a R1, a R2 ou a ambas (sen duplicar).

Sintaxe: **R1** $\cup$ **R2**, onde a relación resultado R contén as tuplas “t” que cumplen  $\{ t / t \in R1 \text{ e/ou } t \in R2 \}$ .

As relacións de entrada R1 e R2 deben:

- **Ser homoxéneas:** non poden conter mestura de tuplas de distintos tipos.
- **Manter a propiedade de Clausura:** o resultado da operación DEBE ser unha relación.
- **Ser de tipos compatibles,** onde a compatibilidade definírase da seguinte maneira:
  - R1 e R2 son compatibles se teñen o mesmo esquema definido por R1 ( t11, t12,..., t1n ), R2 ( t21, t22, ..., t2n ), é dicir:
    - Teñen o mesmo grado( igual número de atributos):  $\text{grado}(R1) = \text{grado}(R2) = n$ .
    - Os atributos correspondentes están definidos sobre o mesmo dominio:  $\text{dom}(t1i) = \text{dom}(t2i) ; i = 1, 2, \dots, n$ .

No exemplo da base de datos casting empregárase unha táboa histórica de actores que se presentaron con anterioridade a outros castings:

ACTOR HISTORICO		
codActor	nome	dataNacemento
321	Antón	19/11/1977
456	Iria	03/03/1997
215	Xenxo	23/10/1986
678	Ximena	07/09/1994
348	Lois	13/12/1978

Supoña que se desexa coñecer que actores en total presentáronse a este e outros castings no pasado:

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981



ACTOR HISTORICO		
codActor	nome	dataNacemento
321	Antón	19/11/1977
456	Iria	03/03/1997
215	Xenxo	23/10/1986
678	Ximena	07/09/1994
348	Lois	13/12/1978

#### – $ACTOR \cup ACTOR\_HISTORICO$

A relación resultante de  $R1 \cup R2$ , con  $R1$  (ACTOR) e  $R2$  (ACTOR\_HISTORICO) compatíbles en tipo, é una relación  $R$  tal que:

- O esquema (cabeceira) será o de  $R1$  (ou  $R2$ ).
- O estado (corpo) estará formado polo conxunto de tuplas que están en  $R1$ , en  $R2$  ou en ambas. As tuplas repetidas elimínanse (por definición).

codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981
321	Antón	19/11/1977
<del>456</del>	<del>Iria</del>	<del>03/03/1997</del>
215	Xenxo	23/10/1986
<del>678</del>	<del>Ximena</del>	<del>07/09/1994</del>
348	Lois	13/12/1978

A unión é unha operación conmutativa é asociativa.

A unión, intersección, diferenza necesitan operandos compatíbles en tipo mentres o produto cartesiano non necesita compatibilidade de tipo nos seus operandos.



### Tarefa 3. Realizar consultas empregando o operador de unión.

#### Diferenza ( $-$ )

A diferenza de dúas relacións  $R1$  e  $R2$ , compatibles no seu esquema, é outra relación definida sobre o mesmo esquema de relación, na que a súa extensión estará constituída polo conxunto de tuplas que pertencen a  $R1$  e non pertencen a  $R2$ .

Sintaxe:  $R1 - R2$ , con  $R1$  e  $R2$  compatibles en tipo, é unha relación  $R$  tal que:

- O esquema (cabeceira) corresponde ao de  $R1$  (ou  $R2$ ).
- O estado (corpo) é o conxunto de tuplas que están en  $R1$ , pero NON en  $R2$ .

Supoña que se desexa coñecer que actores de outros castings pasados se presentan este ano:

ACTOR HISTORICO		
codActor	nome	dataNacemento
321	Antón	19/11/1977
456	Iria	03/03/1997
215	Xenxo	23/10/1986
678	Ximena	07/09/1994
348	Lois	13/12/1978

—

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981

– **ACTOR\_HISTORIO — ACTOR.**

O resultado será:

ACTOR		
codActor	nome	dataNacemento
321	Antón	19/11/1977
215	Xenxo	23/10/1986
348	Lois	13/12/1978

Operación con “certa direccionalidade”, como a resta aritmética, polo que non cumpre a propiedade conmutativa  $(R1 - R2) \neq (R2 - R1)$ .



### Tarefa 4. Realizar consultas empregando o operador de diferenza

#### Produto cartesiano ( $\times$ )

O produto cartesiano de dúas relacións  $R1$  e  $R2$ , de cardinalidades  $m1$  e  $m2$  respectivamente, é unha relación definida sobre a unión dos atributos de ambas relacións e con extensión constituída polas  $m1 \times m2$  tuplas formadas concatenando cada tupla de  $R1$  con cada unha das tuplas de  $R2$ .

Esta operación baséase na operación matemática de conxuntos, onde, sendo  $A$  e  $B$  conxuntos,  $A \times B = \{ (a,b) / a \in A \text{ e } b \in B \}$ . Diferenciándose desta operación en que en lugar de concatenar pares de tuplas, o que se concatenan son os atributos das tuplas.

Sintaxe:  $R1 \times R2$  é unha relación  $R$  na que as tuplas das relacións cumpren que  $\{ (tr1, tr2) \}$

/  $t_{R1} \in R1$  e  $t_{R2} \in R2$  }.

O resultado será unha relación R formada por:

- O esquema, formado pola combinación (unión) dos esquemas de R1 e R2.
- O estado formado polo conxunto de todas as tuplas formadas polas posibles combinacións de cada tupla de R1 con cada tupla de R2, conservando a propiedade de clausura, é dicir, o resultado debe ser un conxunto de tuplas (non de pares de), ampliando a definición matemática de produto cartesiano, pois cada par ordenado é substituído pola tupla resultante da combinación das tuplas orixe.

No exemplo PAPEL X ACTUACION, obtense un conxunto de tuplas tales que cada unha é a combinación dunha tupla de PAPEL con cada unha das tuplas de ACTUACION.

PAPEL		
codPapel	nome	duración
1	Leia Organa	20
2	Han Solo	17
3	Darth Vader	7
4	Chewaka	3

X

ACTUACION		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

En ocasións na táboa resultante prodúcese unha colisión de nomes de atributos (no exemplo codPapel que aparece en ambos operandos), neste caso ambos renoméanse, polo xeral engadindo o nome da táboa de orixe (PAPEL.codPapel ou P.codPapel).

P.codPapel	nome	duración	A.codPapel	codActor	data
1	Leia Organa	20	1	456	03/03(2015
1	Leia Organa	20	4	456	03/03/2015
1	Leia Organa	20	1	678	04/03/2015
1	Leia Organa	20	2	123	04/03/2015
1	Leia Organa	20	3	456	04/04/2015
1	Leia Organa	20	1	678	04/03/2015
1	Leia Organa	20	2	123	04/03/2015
2	Han Solo	17	1	456	03/03(2015
2	Han Solo	17	4	456	03/03/2015
2	Han Solo	17	1	678	04/03/2015
2	Han Solo	17	2	123	04/03/2015
2	Han Solo	17	3	456	04/04/2015
2	Han Solo	17	1	678	04/03/2015
2	Han Solo	17	2	123	04/03/2015
3	Darth Vader	7	1	456	03/03(2015
3	Darth Vader	7	4	456	03/03/2015
3	Darth Vader	7	1	678	04/03/2015
3	Darth Vader	7	2	123	04/03/2015
3	Darth Vader	7	3	456	04/04/2015
3	Darth Vader	7	1	678	04/03/2015
3	Darth Vader	7	2	123	04/03/2015
4	Chewaka	3	1	456	03/03(2015
4	Chewaka	3	4	456	03/03/2015
4	Chewaka	3	1	678	04/03/2015
4	Chewaka	3	2	123	04/03/2015
4	Chewaka	3	3	456	04/04/2015
4	Chewaka	3	1	678	04/03/2015
4	Chewaka	3	2	123	04/03/2015

O produto cartesiano é unha operación sen demasiada importancia práctica, xa que non se ten máis información á saída que á entrada, pero é necesaria para definir a operación REUNIÓN (JOIN).

Esta operación é conmutativa e asociativa.

### 1.2.2.3 Operadores derivados.

As operacións de selección ( $\sigma$ ), proxección ( $\Pi$ ), unión ( $\cup$ ), diferenza ( $-$ ), e produto cartesiano ( $\times$ ) forman un conxunto completo de operacións, xa que as demais poden derivarse delas:

- $\cup$  e  $-$  constitúen a  $\cap$ .
- $\Pi$  e  $\times$  constitúen a  $\div$ .
- $\sigma$  e  $\times$  constitúen a  $\bowtie$  (condición).



- $\sigma$  e  $X$  constitúen a  $\bowtie$ .

### Intersección ( $\cap$ )

A intersección de dúas relacións  $R1$  e  $R2$ , compatibles no seu esquema, é outra relación definida sobre o mesmo esquema de relación, con extensión formada polo conxunto de tuplas que pertencen a  $R1$  e a  $R2$  á vez.


Por tratarse dunha operación derivada pode expresarse do seguinte xeito, sendo  $R1$  e  $R2$  dúas relacións compatibles do esquema:

- $R1 \cap R2 = R1 - (R1 - R2)$ .

Sintaxe:  $R1 \cap R2$ , onde a relación resultado  $R$  contén as tuplas “ $t$ ” que cumpren  $\{ t / t \in R1 \text{ e } t \in R2 \}$ .

Supoña que se desexa coñecer que actores de outros castings pasados se presentan este ano.

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981



ACTOR HISTORICO		
codActor	nome	dataNacemento
321	Antón	19/11/1977
456	Iria	03/03/1997
215	Xenxo	23/10/1986
678	Ximena	07/09/1994
348	Lois	13/12/1978

#### – $ACTOR \cap ACTOR\_HISTORICO$ .

A relación resultante de  $R1 \cap R2$ , con  $R1$  (ACTOR) e  $R2$  (ACTOR\_HISTORICO) compatibles en tipo, é una relación tal que:

- O esquema (cabeceira) será o de  $R1$  (ou  $R2$ ).
- O estado (corpo) será o conxunto de tuplas que están á vez en  $R1$  e  $R2$ .

codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994

A unión é unha operación conmutativa é asociativa.



Tarefa 5. Realizar consultas empregando o operador de intersección.

### División ( $\div$ )

A división dunha relación  $R1$  (dividendo) por outra relación  $R2$  (divisor) é unha relación  $R$  (cociente) tal que, ao realizarse a súa combinación co divisor, todas as tuplas resultantes atópanse no dividendo.

Por tratarse dunha operación derivada pode expresarse do seguinte xeito, sendo  $R1$  e  $R2$  dúas relacións esquema e onde se cumpre  $R2$  está contida en  $R1$ :

$$R1 \div R2 = \Pi \langle R2 - R1 \rangle (R1) - \Pi \langle R2 - R1 \rangle ((\Pi \langle R2 - R1 \rangle (R1) \times R2) - R1).$$

Sintaxe: **R1 ÷ R2**, onde R1(x1, x2, ... xn, y1, y2, ... ym) dividido por R2(y1, y2, ...ym) dan como resultado unha relación R tal que:

- O esquema estará formado polos atributos non comúns R1( x1, x2, ...xn ).
- O estado será un conxunto de tuplas (x) tal que existe en R2 unha tupla (x,y) para TODAS as tuplas (y ) de R2.

No exemplo, deséxase coñecer que actores foron puntuados por todos os membros do xurado.

DIVIDENDO

PUNTUACIÓN				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

DIVISOR

MEMBRO XURADO		
codXurado	nome	especialidade
222	Sabela	NULO
333	Roi	NULO
444	Xulia	NULO
555	Xurxo	NULO



Atributos non comúns  
que aparecen asociados  
a TODOS os valores  
do atributo común

#### – PUNTUACION ÷ MEMBRO\_XURADO.

Para que unha tupla t apareza no resultado, os valores de t deben aparecer en R1 en combinación con todas as tuplas de R2. Esta operación dará como resultado:

codPuntuacion	nota	codXurado	codPapel	codActor
2	5	333	4	456
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

A división é unha operación pouco común, útil para consultas especiais ocasionais.



Tarefa 6. Realizar consultas empregando o operador de división

#### Combinación, reunión ou join (⋈)

A combinación de dúas relacións respecto dunha certa condición, é outra relación constituída por todos os pares de tuplas t1i e t2i concatenadas, tales que, en cada par, as correspondentes tuplas cumpren a condición especificada. Desta forma combínase as tuplas relacionadas de dúas relacións nunha soa tupla, permitindo procesar vínculos entre relacións. Esta operación correspóndese cun produto cartesiano seguido dunha restrición.

Por tratarse dunha operación derivada, pode expresarse do seguinte xeito, sendo R1 e R2 dúas relacións do esquema:

- $R1 \bowtie_{\langle \text{condición de reunión} \rangle} R2 = \prod_{\langle A1, A2, \dots, An \rangle (\{=, <, \leq, >, \geq, \neq\} (R1 \times R2))}$ .

A sintaxe para dúas relacións R1 (t11, t12, t13, ..., t1n) e R2 (t21, t22, t23, ..., t2m) é:

#### **$R1 \bowtie_{\langle \text{condición de reunión} \rangle} R2$ :**

- A  $\langle \text{condición de reunión} \rangle$  é:
  - Unha expresión booleana especificada en termos de atributos de R1 e R2 e que ten a forma t1i OP t2i onde OP é un operador de comparación que cumpre  $\in \{=, <, \leq, >, \geq, \neq\}$ .
  - Avaliada para cada combinación (par) de tuplas, de tal xeito que as que a cumpren, forman unha nova tupla da relación resultado.
  - A condición especifícase da seguinte forma:  $\langle \text{condición} \rangle \text{ AND } \langle \text{condición} \rangle \text{ AND } \dots \text{ AND } \langle \text{condición} \rangle$  onde o dominio de t1i é igual ao dominio t2i.

O resultado será unha relación R con n+m atributos (t11, t12, ... t1n, t21, t22, ... t2m) na que:

- O esquema é a unión das cabeceiras de R1 e R2.
- O estado está formado polo conxunto de tuplas, unha tupla por cada combinación de tuplas (unha de R1 e outra de R2) que cumpre a condición.
- Se ningunha combinación de tuplas de R1 e R2 cumpre a condición, entón a relación R resultado será unha relación baleira (cero tuplas).
- Se NON se especifica condición, entón a condición é certa para todas as tuplas, e equivalería a un produto cartesiano (tamén chamado reunión ou reunión cruzada):

No exemplo, desexase coñecer toda a información sobre as personaxes que foron interpretadas por algún actor no proceso de casting.

PAPEL			ACTUACIÓN		
codPapel	nome	duración	codPapel	codActor	data
1	Leia Organa	20	1	456	03/03/2015
2	Han Solo	17	4	456	03/03/2015
3	Darth Vader	7	2	456	04/04/2015
4	Chewaka	3	3	456	04/04/2015
			1	678	04/03/2015
			2	123	04/03/2015



(PAPEL.CodPapel = ACTUACIÓN.CodPapel)

- **$PAPEL \bowtie_{\langle \text{PAPEL.codPapel} = \text{ACTUACION.codPapel} \rangle} ACTUACION$ :** Neste caso a condición será de igualdade entre os campos que identifican o papel en ambas táboas.

Esta operación fai nun primeiro paso un produto cartesiano de ambas táboas, e logo elimina as tuplas que non cumpren a condición:

P.codPapel	nome	duración	A.codPapel	codActor	data
1	Leia Organa	20	1	456	03/03/2015
1	Leia Organa	20	4	456	03/03/2015
1	Leia Organa	20	1	678	04/03/2015
1	Leia Organa	20	2	123	04/03/2015
1	Leia Organa	20	3	456	04/04/2015
1	Leia Organa	20	1	678	04/03/2015
1	Leia Organa	20	2	123	04/03/2015
2	Han Solo	17	1	456	03/03/2015
2	Han Solo	17	4	456	03/03/2015
2	Han Solo	17	1	678	04/03/2015
2	Han Solo	17	2	123	04/03/2015
2	Han Solo	17	3	456	04/04/2015
2	Han Solo	17	1	678	04/03/2015
2	Han Solo	17	2	123	04/03/2015
3	Darth Vader	7	1	456	03/03/2015
3	Darth Vader	7	4	456	03/03/2015
3	Darth Vader	7	1	678	04/03/2015
3	Darth Vader	7	2	123	04/03/2015
3	Darth Vader	7	3	456	04/04/2015
3	Darth Vader	7	1	678	04/03/2015
3	Darth Vader	7	2	123	04/03/2015
4	Chewaka	3	1	456	03/03/2015
4	Chewaka	3	4	456	03/03/2015
4	Chewaka	3	1	678	04/03/2015
4	Chewaka	3	2	123	04/03/2015
4	Chewaka	3	3	456	04/04/2015
4	Chewaka	3	1	678	04/03/2015
4	Chewaka	3	2	123	04/03/2015

Como se observa no exemplo, prodúcense colisións de nomes de atributos, neste caso actúase do mesmo xeito que no produto cartesiano, isto é, renomeando os atributos con nomes coincidentes.

O resultado final da operación será:

P.codPapel	nome	duración	A.codPapel	codActor	data
1	Leia Organa	20	1	456	03/03/2015
1	Leia Organa	20	1	678	04/03/2015
1	Leia Organa	20	1	678	04/03/2015
2	Han Solo	17	2	123	04/03/2015
2	Han Solo	17	2	123	04/03/2015
3	Darth Vader	7	3	456	04/04/2015
4	Chewaka	3	4	456	03/03/2015

Tipicamente os atributos aparellados son os atributos clave primaria e clave allea das entidades coas que se opera, como sucede neste exemplo.

A reunión máis común é a que implica comparación de igualdade que recibe o nome EQUI-REUNIÓN ou REUNIÓN NATURAL, e simbolízase co operador \*.



Tarefa 7. Realizar consultas empregando o operador de combinación.

### Reunión natural, produto natural ou natural join ( \* )

A operación de reunión natural é unha combinación por igualdade onde se eliminou, na relación resultante, un atributo de cada parella no que os valores son idénticos. É o caso máis utilizado de combinación para relacións que teñen atributos comúns. Esta operación correspóndese cun produto cartesiano seguido dunha restrición por igualdade, e despois una proxección (para eliminar a duplicación de atributos emparellados). Nos casos en que o atributo común ten o mesmo nome en ambas relacións, adóitase omitir a condición de combinación.

Por tratarse dunha operación derivada pode expresarse do seguinte xeito, sendo R1 e R2 dúas relacións do esquema:

$$R1 * R2 = R1 \bowtie_{\langle t1i=t2i \rangle} R2 = \sigma_{\langle t1i=t2i \rangle} R1 \times R2.$$

A sintaxe é **R1 \* R2**:

Constitúe un caso particular de reunión, o que implica comparacións de igualdade que é o caso de reunión máis común, é unha EQUI-REUNIÓN xunto coa eliminación de atributos superfluos.

Non necesita especificar condición de reunión, pois é igual a todos os pares de atributos con igual nome en R1 e R2 sendo estes os únicos atributos que conserva.

A definición estándar de reunión natural esixe que os atributos de reunión deben ter nome idéntico en ambas relacións operando, ademais os atributos de reunión deben ter o mesmo dominio.

No exemplo, deséxase coñecer toda a información sobre as personaxes que foron interpretadas por algún actor no proceso de casting.

PAPEL		
codPapel	nome	duración
1	Leia Organa	20
2	Han Solo	17
3	Darth Vader	7
4	Chewaka	3



(PAPEL.CodPapel =  
ACTUACION.CodPapel)

ACTUACIÓN		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

Son a mesma operación, pero empregando o JOIN NATURAL non precisamos indicar a condición xa que por defecto e a de igualdade entre a tributos co mesmo nome en ambas táboas

PAPEL		
codPapel	nome	duración
1	Leia Organa	20
2	Han Solo	17
3	Darth Vader	7
4	Chewaka	3



ACTUACIÓN		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

- **PAPEL \* ACTUACION:** Neste caso a condición será de igualdade entre os campos que identifican o papel en ambas táboas.

A táboa resultante é a seguinte:

P.codPapel	nome	duración	codActor	data
1	Leia Organa	20	456	03/03/2015
1	Leia Organa	20	678	04/03/2015
1	Leia Organa	20	678	04/03/2015
2	Han Solo	17	123	04/03/2015
2	Han Solo	17	123	04/03/2015
3	Darth Vader	7	456	04/04/2015
4	Chewaka	3	456	03/03/2015



Tarefa 8. Realizar consultas empregando o operador de combinación natural

#### 1.2.2.4 Resumo de propiedades dos operadores relacionais

R, S, T relaciones de tipos compatibles:

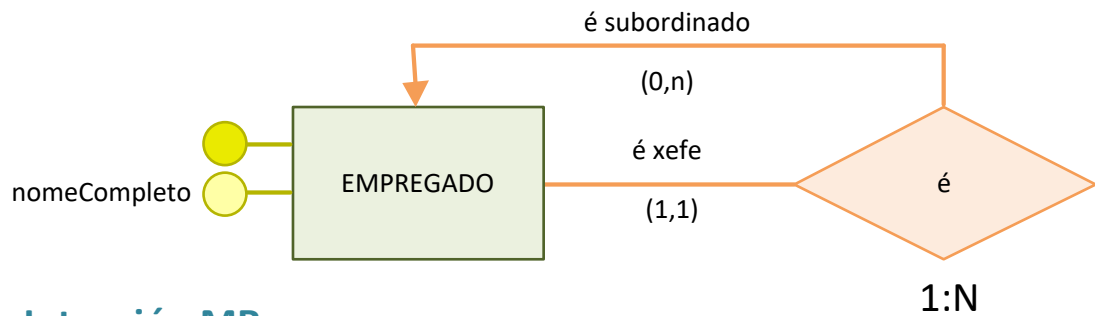
- Asociativa:
  - $(R \cup S) \cup T \equiv R \cup (S \cup T) \equiv R \cup S \cup T.$
  - $(R \cap S) \cap T \equiv R \cap (S \cap T) \equiv R \cap S \cap T.$
  - $(R \times S) \times T \equiv R \times (S \times T) \equiv R \times S \times T.$
- Conmutativa:
  - $R \cup S \equiv S \cup R.$
  - $R \cap S \equiv S \cap R.$
  - $R \times S \equiv S \times R.$
- A diferenza no cumpre ningunha destas propiedades.

#### 1.2.2.5 Operacións adicionais

Trátanse de extensións á álgebra propiamente dita e utilízanse por motivos prácticos. Algunhas delas aparecen pola necesidade de definir as necesidades de comportamento cos valores nulos que ata a súa aparición eran tratados como un valor máis. Estas operacións están incluídas na maioría das linguaxes de consultas comerciais.

#### Peché recursivo

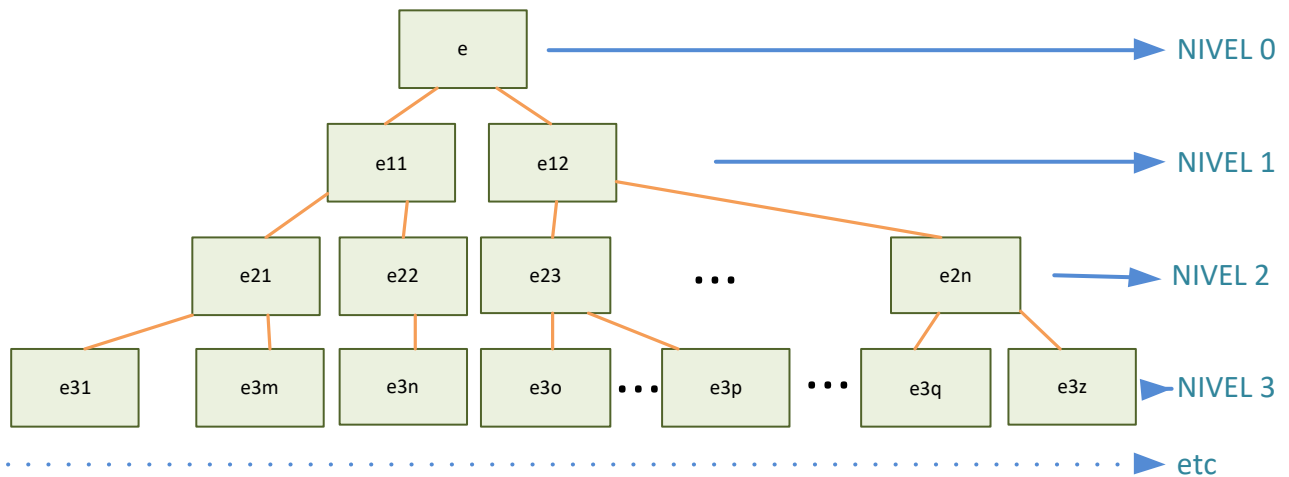
Estas operacións non poden expresarse no álgebra relacional. Aplícase a una referencia recursiva entre tuplas do mesmo tipo (empregado e xefe na relación empregado).



## Intensión MR

**EMPREGADO** (dni, nomeCompleto, xefe\*)  
 ↑  
 B:R, M:C

En álgebra relacional é sinxelo especificar os empregados dun xefe, pero é moi complicado chagar á raíz da árbore recursiva:



Supoñamos o seguinte contido para a relación **EMPREGADO** do exemplo:

<b>EMPREGADO</b>		
<b>CodEmpleado</b>	<b>nomeCompleto</b>	<b>xefe</b>
1	Mouteria Patiño, Patricia	NULO
1.1	Sarrapo Prado, Xoel	1
1.2	Penedo Almeida, Rosalía	1
1.1.1	Bouzas Piñoi, Praxeres	1.1
1.1.2	Losada Ardá, Urbán	1.1
1.1.3	Saburido Portomeñe, Xudite	1.1
1.2.1	Pedraio Bugallo, Xara	1.2
1.2.2	Regadas Pombo, Xelvira	1.2
1.2.3	Parcerio Figueroa, Uxío	1.2

Deséxase recuperar os códigos dos empregados cuxo xefe directo é “Mouteria Patiño, Patricia” xunto cos códigos dos empregados cuxos xefes empregados directos de “Mouteria Patiño, Patricia”.

Para obter esta información aplicarase a álgebra en varios pasos

- Nunha táboa chamada EMP\_E\_XEFES recupéranse os códigos de todos os empregados cos seus xefes:  $\text{EMP\_E\_XEFES}(\text{codEmpregado}, \text{xefe}) \leftarrow \Pi_{\langle \text{codEmpregado}, \text{xefe} \rangle} (\text{EMPREGADO})$ :

EMP_E_XEFES	
CodEmpregado	xefe
1	NULO
1.1	1
1.2	1
1.1.1	1.1
1.1.2	1.1
1.1.3	1.1
1.2.1	1.2
1.2.2	1.2
1.2.3	1.2

EMPREGADO		
CodEmpregado	nomeCompleto	xefe
1	Mouteria Patiño, Patricia	NULO
1.1	Sarrapo Prado, Xoel	1
1.2	Penedo Almeida, Rosalía	1
1.1.1	Bouzas Piñoi, Praxeres	1.1
1.1.2	Losada Ardá, Urbán	1.1
1.1.3	Saburido Portomeñe, Xudite	1.1
1.2.1	Pedraio Bugallo, Xara	1.2
1.2.2	Regadas Pombo, Xelvira	1.2
1.2.3	Parcerio Figueroa, Uxío	1.2

$\text{EMP\_E\_XEFES}(\text{codEmpregado}, \text{xefe}) \leftarrow \Pi_{\langle \text{codEmpregado}, \text{xefe} \rangle} (\text{EMPREGADO})$

- Recuperarase agora só os empregados directos do xefe có código “1.2” para elo seleccionamos as tuplas con ese código no campo xefe da táboa EMP\_E\_XEFES para posteriormente proxectar solo o código de empregado sobre a táboa:  $\text{EMP\_XEFE1}(\text{cod}) \leftarrow \Pi_{\text{codEmpregado}} (\sigma_{\langle \text{xefe}=1.2 \rangle} (\text{EMP\_E\_XEFES}))$ .

EMP_XEFE1	
cod	
1.1	
1.2	

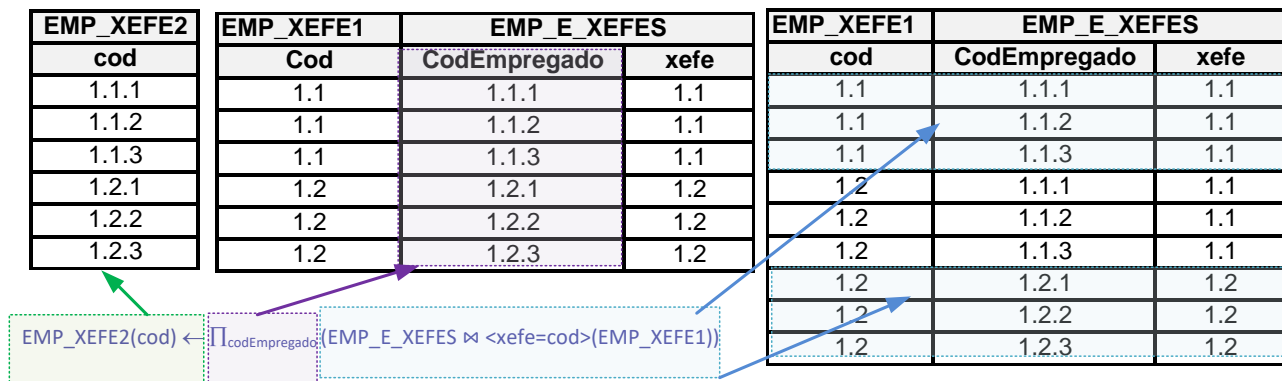
CodEmpregado	xefe
1.1	1
1.2	1

EMP_E_XEFES	
CodEmpregado	xefe
1	NULO
1.1	1
1.2	1
1.1.1	1.1
1.1.2	1.1
1.1.3	1.1
1.2.1	1.2
1.2.2	1.2
1.2.3	1.2

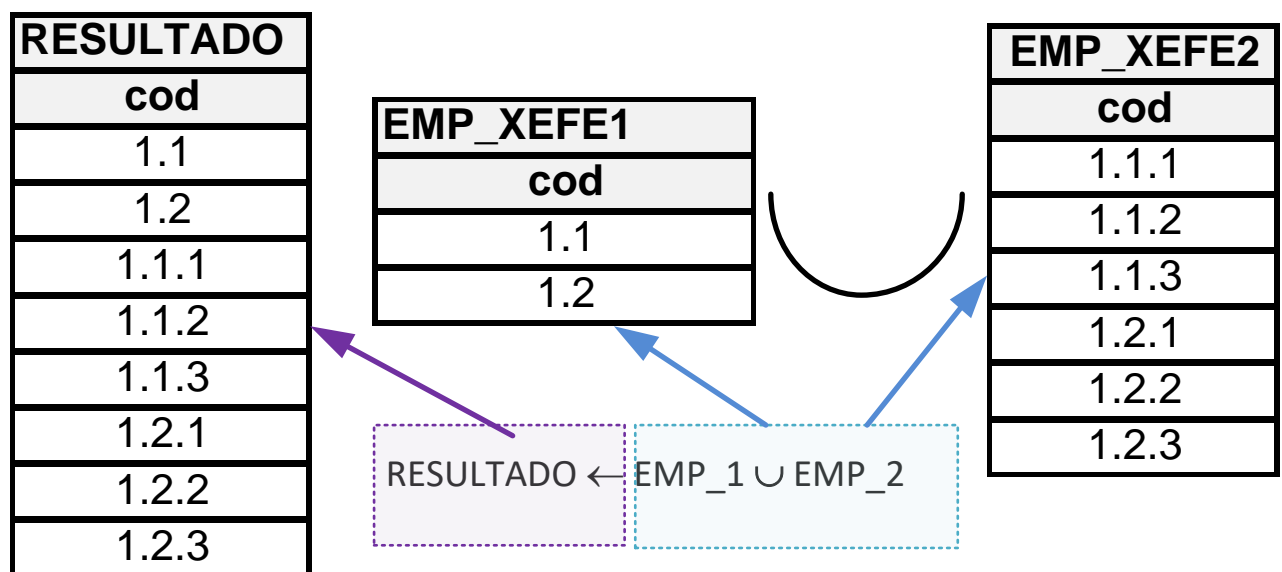
$\text{EMP\_XEFE1}(\text{cod}) \leftarrow \Pi_{\text{codEmpregado}} (\sigma_{\langle \text{xefe}=1.2 \rangle} (\text{EMP\_E\_XEFES}))$

Recuperarase agora os empregados directos dos empregados anteriores gardándoos nunha táboa que se chamará EMP\_XEFE2. Para elo recuperamos mediante a operación de reunión (JOIN) da táboa EMP\_E\_XEFES aqueles empregados que son empregados directos do empregado con código “1” e que aparecen na táboa EMP\_XEFE1. Do resultado anterior proxectamos só o código do empregado:  $\text{EMP\_XEFE2}(\text{cod}) \leftarrow \Pi_{\text{codEmpregado}} (\text{EMP\_E\_XEFES} \bowtie_{\langle \text{xefe}=\text{cod} \rangle} (\text{EMP\_XEFE1}))$ .





- Finalmente só queda unir as táboas resultantes que conteñen os empregados de segundo e terceiro nivel: **RESULTADO**  $\leftarrow$  **EMP\_1**  $\cup$  **EMP\_2**.



### Proxección xeneralizada

Admite a realización de operacións sobre os atributos proxectados

Sintaxe:  $\Pi_{\langle f_1(A_1), \dots, f_n(A_n) \rangle}(\langle RELACION \rangle)$  onde  $A_i$  son os atributos proxectados da relación e  $f_i$  son as funcións a empregar.

No exemplo deséxase coñecer as puntuacións obtidas en porcentaxes de rango 1 a 100:

- $RESULTADO \leftarrow \Pi_{\langle nota*10 \rangle}(PUNTUACIÓN).$

### Agrupación e agregación

As funcións de agregado son funcións matemáticas aplicadas aos valores dun conxunto de tuplas da BD e son:

- SUM: Suma de valores.
- AVG: Promedio de valores.
- MAX: Máximo dun conxunto de valores.
- MIN: Mínimo dun conxunto de valores.
- COUNT: Número de tuplas dunha relación.

Estas agrupacións de tuplas permiten aplicar unha función agregada a cada grupo por separado, segundo o valor dalgún dos atributos.

Sintaxe <atributos de agrupación> FG <lista de funcións>(<RELACION>) onde:

- <atributos de agrupación>: lista dos nomes de atributos de A1 a An da RELACION pola que se farán os grupos. De atoparse valeira, e dicir, non se especifican atributos de agrupación, o grupo farase con todos os atributos da relación.
- < lista de funcións>: lista de pares <función(atributo)> onde <función> é unha das seguintes {SUM, AVG, MAX, MIN, COUNT} e atributo é un atributo da relación.

O resultado é unha relación R tal que:

- A cabeceira está formada polos atributos de <atributos de agrupación> xunto cun atributo de cada elemento de <lista de funcións>.
- O corpo estará formado por un conxunto de tuplas (unha por cada grupo).

No exemplo poderíase calcular a puntuación media por actor:

- CodActorFG(SUM(nota))(PUNTUACION).



Tarefa 9. Realizar consulta empregando a agrupación e a agregación

### Reunión externa (outer-join)

As operacións de reunión externa son unha extensión da reunión natural e permite conservar TODAS as tuplas en R1, R2 ou ambas, aínda que:

- non teñan tuplas coincidentes.
- conteñan NULOS nos atributos de reunión.

Polo tanto, evítase que ao combinar R1 con R2, as tuplas dunha relación que non coinciden con ningunha tupla da outra desaparezan no resultado (comportamento da combinación normal interna).

Atendendo as tuplas que se conserven (as de R1, R2, ou ambas) distínguense tres tipos:

- reunión externa esquerda (/\*).
- reunión externa pola dereita (\*/\*).
- reunión externa simétrica (/\*/\*).

Para o noso exemplo de base de datos dun casting de actores engadiremos un novo actor, que non se presentou a ningún papel e un novo membro de xurado que non fixo ningunha nova puntuación.

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981
765	Xavier	28296

MEMBRO XURADO		
codXurado	nome	especialidade
222	Sabela	NULO
333	Roi	NULO
444	Xulia	NULO
555	Xurxo	NULO
666	Paula	NULO

Supoñamos que se realiza unha reunión interna entre as táboas MENBRO\_XURADO e PUNTUACIÓN:

MEMBRO XURADO		
codXurado	nome	especialidade
222	Sabela	NULO
333	Roi	NULO
444	Xulia	NULO
555	Xurxo	NULO
666	Paula	NULO

Tuplas non coincidentes  
que desaparecen cunha  
reunión natural interna

PUNTUACIÓN				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

### Reunión externa esquerda ( /\* )

A reunión externa pola esquerda toma todas as tuplas da relación da esquerda que non coincidan con ningunha tupla da relación da dereita, e completa con valores nulos os demais atributos da tupla da dereita, engadíndoas ao resultado da reunión natural.

Sintaxe: **R1 /\* R2** da como resultado unha relación R con todas as tuplas de R1 onde:

- O esquema é a concatenación das táboas.
- O estado está formado polo conxunto de tuplas de R1, unha por cada combinación de tuplas (unha de R1 e outra de R2) que cumpren a condición de reunión ou igualdade e as tuplas de R1 non coincidentes coas tuplas de R2 que se completa a NULOS.

No exemplo, deséxase coñecer a información de todos os membros do xurado xunto coas súas puntuacións si é que as teñen.

- **RESULTADO ← MEMBRO\_XURADO /\* PUNTUACION:**

PUNTUACION /* MEMBRO_XURADO							
codXurado	nome	especialidade	codPuntuacion	nota	codXurado	codPapel	codActor
222	Sabela	NULO	1	3	222	1	678
222	Sabela	NULO	4	8	222	1	678
222	Sabela	NULO	9	4	222	2	456
333	Roi	NULO	2	5	333	4	456
333	Roi	NULO	7	5	333	NULO	NULO
444	Xulia	NULO	3	6	444	2	123
444	Xulia	NULO	8	7	444	1	456
555	Xurxo	NULO	10	6	555	3	456
666	Paula	NULO	NULO	NULO	NULO	NULO	NULO

Non se perden os datos do membro do xurado por estar a esquerda

### Reunión externa dereita ( \*/ )

A reunión externa pola dereita toma todas as tuplas da relación da dereita que non coincidan con ningunha tupla da relación da esquerda, e completa con valores nulos os demais atributos da relación da esquerda e engadíndoas ao resultado da reunión natural.

Sintaxe: **R1 \*/ R2** da como resultado unha relación R con todas as tuplas de R2 onde:

- O esquema é a concatenación de ambas táboas.
- O estado está formado polo conxunto de tuplas de R2, unha por cada combinación de tuplas (unha de R1 e outra de R2) que cumpren a condición de reunión ou igualdade e as tuplas de R2 non coincidentes coas tuplas de R1 que se completan a nulos.

No exemplo, deséxase coñecer a información de todas as notas postas, xunto coa información do xurado que a calificou si é coñecido.

- $\text{RESULTADO} \leftarrow \text{PUNTUACION} */ \text{MEMBRO\_XURADO}$ .

PUNTUACION */ MEMBRO_XURADO							
codXurado	nome	especialidade	codPuntuacion	nota	codXurado	codPapel	codActor
222	Sabela	NULO	1	3	222	1	678
222	Sabela	NULO	4	8	222	1	678
222	Sabela	NULO	9	4	222	2	456
333	Roi	NULO	2	5	333	4	456
333	Roi	NULO	7	5	333	NULO	NULO
444	Xulia	NULO	3	6	444	2	123
444	Xulia	NULO	8	7	444	1	456
555	Xurxo	NULO	10	6	555	3	456
NULO	NULO	NULO	5	6	NULO	3	678

Non se perden os datos da do xuradpuntuación por estar a dereita

### Reunión externa completa (/\*/)

A reunión externa completa realiza as operacións da reunión externa pola esquerda e da reunión externa pola dereita.

Sintaxe:  $\mathbf{R1} */ \mathbf{R2}$  da como resultado unha relación R con todas as tuplas de R1 e R2 onde:

- O esquema é a cabeceira das relacións R1 e R2.
- O estado está formado polo conxunto de tuplas, unha por cada combinación de tuplas (unha de R1 e outra de R2) que cumpren a condición de reunión ou igualdade e as tuplas de R1 e R2 non coincidentes completaranse con NULO.

No exemplo, deséxase coñecer a información completa de todas as notas postas por todos os xurados.

- $\text{RESULTADO} \leftarrow \text{PUNTUACIÓN} */ \text{MEMBRO\_XURADO}$ .

PUNTUACION /*/ MEMBRO_XURADO							
codXurado	nome	especialidade	codPuntuacion	nota	codXurado	codPapel	codActor
222	Sabela	NULO	1	3	222	1	678
222	Sabela	NULO	4	8	222	1	678
222	Sabela	NULO	9	4	222	2	456
333	Roi	NULO	2	5	333	4	456
333	Roi	NULO	7	5	333	NULO	NULO
444	Xulia	NULO	3	6	444	2	123
444	Xulia	NULO	8	7	444	1	456
555	Xurxo	NULO	10	6	555	3	456
666	Paula	NULO	NULO	NULO	NULO	NULO	NULO
NULO	NULO	NULO	5	6	NULO	3	678

Non se perde ningha das tuplas non coincidentes



### Tarefa 10. Consulta empregando a reunión externa (outer-join)

### 1.2.2.6 Correspondencia da álgebra relacional coa linguaxe de consulta SQL

ÁLXEBRA RELACIONAL	SQL
$\pi_{\langle \text{atributos} \rangle} (\sigma_{\langle \text{condicion} \rangle} (\text{RELACION}))$	SELECT <atributos> FROM <RELACION> WHERE <condicion>
$\rho_{\langle \text{atributo1}, \text{atributo1}' \rangle} (\pi_{\langle \text{atributo1} \rangle, \langle \text{atributo2} \rangle} (\sigma_{\langle \text{condicion} \rangle} (\text{RELACION})))$	SELECT <atributo1> AS <atributo1'>, <atributo2> FROM <RELACION> WHERE <condicion>
$\text{RELACION}' \leftarrow (\pi_{\langle \text{atributos} \rangle} (\sigma_{\langle \text{condicion} \rangle} (\text{RELACION})))$	CREATE VIEW <RELACION'> AS SELECT <atributos> FROM <RELACION> WHERE <condicion>
$\pi_{\langle \text{atributos} \rangle} (\sigma_{\langle \text{condicion} \rangle} (\text{RELACION1} * \text{RELACION2}))$	SELECT <atributos> FROM <RELACION1> NATURAL JOIN <RELACION2> WHERE <condicion>
$\pi_{\langle \text{atributos} \rangle} (\text{RELACION1} \bowtie_{\langle \text{condicion} \rangle} \text{RELACION2})$	SELECT <atributos> FROM <RELACION1> JOIN <RELACION2> ON WHERE <condicion>
$\pi_{\langle \text{atributos} \rangle} (\sigma_{\langle \text{condicion} \rangle} (\text{RELACION1} \times \text{RELACION2}))$	SELECT <atributos> FROM <RELACION1>, <RELACION2> WHERE <condicion>
$\text{RELACION1} \cup \text{RELACION2}$	SELECT * FROM RELACION1 UNION SELECT * FROM RELACION2
$\text{RELACION1} \cap \text{RELACION2}$	SELECT * FROM RELACION1 INTERSECT SELECT * FROM *RELACION2
$\text{RELACION1} - \text{RELACION2},$	SELECT * FROM * RELACION1 EXCEPT SELECT FROM * RELACION2
$\langle \text{atributos} \rangle \text{ F GCount}(\text{atributo1}), \text{Sum}(\text{atributo2})(\text{RELACION})$	SELECT COUNT (<atributo1>), SUM(<atributo2>) FROM <RELACION> GROUP BY <atributo1>, <atributo2>
$\pi_{\langle \text{atributo comun} \rangle} (\text{RELACION1}) \div \pi_{\langle \text{atributo comun} \rangle} (\text{RELACION2})$	SELECT <atributos> FROM RELACION1 WHERE <atributo comun> NOT IN ( SELECT <atributo comun> FROM RELACION2 INNER JOIN RELACION1 WHERE <atributo comun> NOT IN ( SELECT <atributo comun> FROM RELACION2 ) )

### 1.2.3 Cálculo relacional.

O cálculo relacional (CR) foi proposto por Codd, como alternativa á álgebra relacional que resultaba demasiado matemática para o usuario final, co fin de obter unha linguaxe que se parecera máis as linguaxes de programación.

A diferenza entre ambas é que mentres a álgebra relacional ofrece o conxunto de operacións para construír a relación que se quere obter a partir dunhas relacións dadas, indica o procedemento que se debe utilizar para obter o resultado; o cálculo relacional ofrece unha notación para indicar como debe ser a relación resultado en función dunhas relacións determinadas, e dicir, o cálculo só plantexa o problema (só é preciso indicar o resultado que se quere obter).

As diferenzas entre a álgebra e o cálculo relacional son totalmente equivalentes, por cada expresión do cálculo relacional existe unha expresión equivalente da álgebra relacional e viceversa, e ambas son a parte dinámica do modelo relacional.

Existen dous tipos de cálculo relacional que adáptanse ao Cálculo de Predicados de Primeira Ode, no que se basean, para crear una Linguaxe de Consultas para Bases de Datos Relacionais (BDR):

- O proposto por Codd, en 1972, chamado cálculo relacional orientado á tupla (CRT), no mesmo, unha variable interprétase como si representase as tuplas dunha relación.
- O proposto por Lacroix e Pirotte, en 1977, chamado cálculo relacional orientado ao dominio, no que unha variable interprétase como si representase os valores dun dominio.

Cálculo Relacional	Álgebra Relacional
<ul style="list-style-type: none"><li>▪ Expresións Declarativas (linguaxe non procedimental)</li></ul>	<ul style="list-style-type: none"><li>▪ <b>Secuencias</b> de Operacións</li></ul>
<ul style="list-style-type: none"><li>▪ Non se indica CÓMO avaliar a consulta, senón <b>QUÉ</b> se desexa obter</li><li>▪ <b>Describe</b> a información desexada sen dar un procedemento específico para obtela</li></ul>	<ul style="list-style-type: none"><li>▪ Aínda que se aniñan para formar unha soa expresión, <b>sempre</b> se indica explicitamente certo <b>orde</b> das operacións</li><li>▪ <b>Estratexia parcial de avaliación</b> da consulta (linguaxe procedimental de alto nivel)</li></ul>

#### 1.2.3.1 Cálculo relacional de tuplas (CRT)

O CRT baséase na especificación de variables de tupla ou variable de recorrido, que son aquelas nas que os valores permitidos son tuplas da relación ou relacións sobre a que está definida dita variable.

Sintaxe dunha consulta sinxela: { **t** | **COND(t)** }, onde:

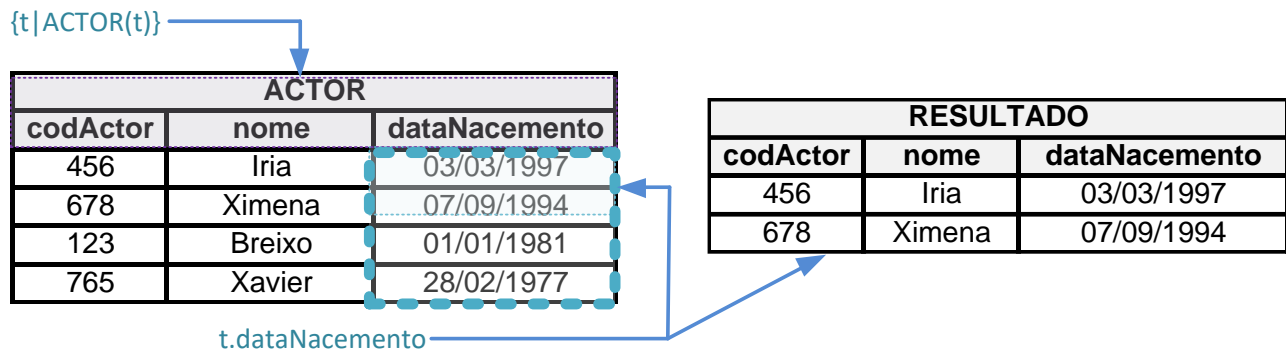
- **t**: representa unha variable de tupla.
- **COND(t)**: é unha expresión condicional na que intervéñ a variable de tupla **t**.

O resultado é o conxunto de tuplas **t** que cumpren a condición **COND(t)**.

Exemplo:

Actores nados con posterioridade a 1985:

- { **t** | **ACTOR(t)** and **t.dataNacemento**>31/12/1984 }.
  - **ACTOR(t)**: indica que **ACTOR** é a Relación de Intervalo que **t** recorre.
  - **t.dataNacemento**: fai referencia ao atributo **dataNacemento** da variable de tupla **t**.

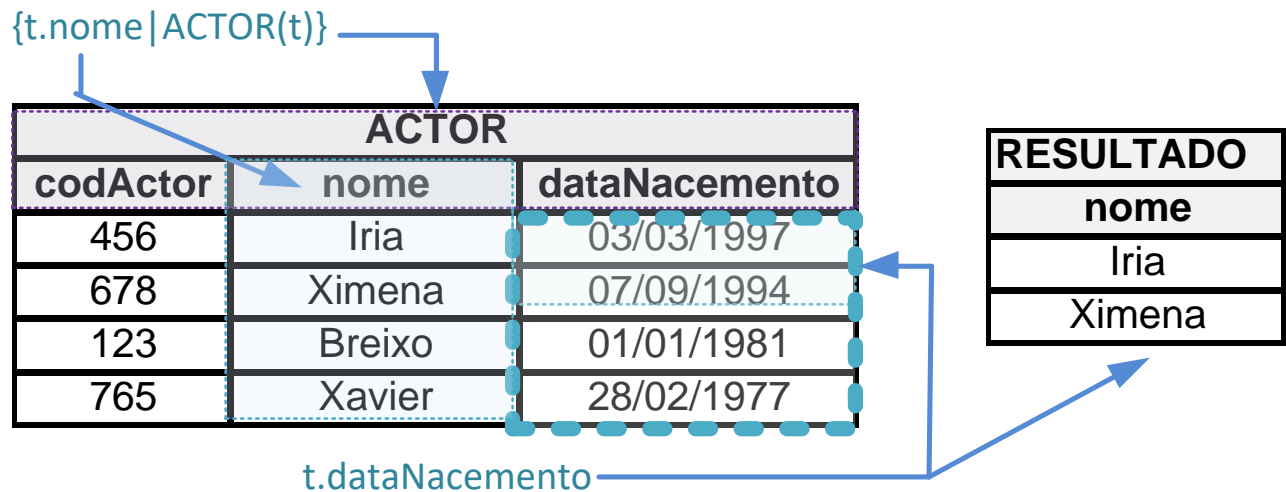


A obtención dalgúns atributos das tuplas seleccionadas especificase coa seguinte sintaxe:

$\{ \langle \text{lista atributos} \rangle | \text{COND}(t) \}$ .

No exemplo anterior, se quere recuperar só Actores nados con posterioridade a 1985:

- $\{ t.\text{nome} | \text{ACTOR}(t) \text{ and } t.\text{dataNacemento} > 31/12/1984 \}$ .



## Expresión Xeral do CRT

A sintaxe xeral para unha expresión CRT é como sigue:

- $\{ t1.a1, t2.a2, \dots, tn.ap | \text{COND}(t1, t2, \dots, tn, tn+1, tn+2, \dots, tn+m) \}$ , onde:
  - $t1, t2, \dots, tn, tn+1, tn+2, \dots, tn+m$  son variables de tupla.
  - pódense identificar atributos específicos “ai” nas variables tuplas “ti”, podendo mesurar variables tupla con e sen atributos.
  - $\text{COND}(\dots)$ : fórmula (ben formada) do Cálculo Relacional de Tuplas constituída por átomos do Cálculo de Predicados.

## Os átomos

A sintaxe dunha fórmula atómica é:

- $R(t)$  onde  $R$  é un relación e  $t$  unha variable de tupla.
- $t1.a$  op  $t2.b$  ou  $t1.a$  op  $ctte$ :
  - $t1$  e  $t2$  variables de tupla.
  - $a$  atributo da relación que  $t$ .
  - $b$  atributo da relación que  $t$ .

- Cte é un valor constante.
- Op e un operador contido en  $\{=, <, \leq, >, \geq, \neq\}$ .

No exemplo, quérese recuperar as notas por papel e actor asignadas polo membro do xurado con código “222”.

- $\{ t.nota, t.codPapel, t.codActor \mid \text{PUNTUACION}(t) \text{ and } (t.CodXurado = 222) \}$ .

PUNTUACION				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

PUNTUACIÓN		
nota	codPapel	codActor
3	1	678
8	1	678
4	2	456

$\{ t.nota, t.codPapel, t.codActor \mid \text{PUNTUACION}(t) \text{ and } (t.CodXurado = 222) \}$

Os átomos están unidos mediante operadores lóxicos and, or, not e  $\rightarrow$ .

No exemplo, quérese recuperar as notas superiores a 4 por papel e actor asignadas polo membro do xurado con código “222”.

- $\{ t.nota, t.codPapel, t.codActor \mid \text{PUNTUACION}(t) \text{ and } (t.CodXurado = 222) \text{ and } (t.nota > 4) \}$ .

PUNTUACION				
codPuntuacion	nota	codXurado	codPapel	codActor
1	3	222	1	678
2	5	333	4	456
3	6	444	2	123
4	8	222	1	678
5	6	NULO	3	678
7	5	333	NULO	NULO
8	7	444	1	456
9	4	222	2	456
10	6	555	3	456

PUNTUACIÓN		
nota	codPapel	codActor
8	1	678

$\{ t.nota, t.codPapel, t.codActor \mid$   
 $\text{PUNTUACION}(t) \text{ and}$   
 $(t.CodXurado = 222) \text{ and}$   
 $(t.nota > 4) \}$

As formulas atómicas pódense cuantificar cos operadores:

- **“Existe”** ( $\exists$ ): exemplo: Recuperar o nome dos actores que non teñen unha nota pola súa actuación.
  - $\{ a.nome \mid \text{ACTOR}(a) \text{ and } (\text{not } \exists (\text{PUNTUACION}(p) \text{ and } (p.codPapel) = (a.codPapel))) \}$
- **“Para todo”** ou universal ( $\forall$ ): No exemplo, deséxase incluír a restrición “Non pode existir ningún membro do xurado que puntúe a todos os actores”.
  - $\{ \forall \text{MEMBRO\_XURADO}(mx) \rightarrow \text{not } \exists \text{ACTOR}(a) \text{ and } \forall \text{PUNTUACION}(p) \rightarrow (p.codActor = a.codActor) \text{ and } (p.codXurado = mx.codXurado) \}$ .

Nunha condición só poden aparecer dous tipos de variables tupla ou “t”:



- As variables propias da consulta.
- As variables cuantificadas con  $\forall$  e  $\exists$  dentro da condición.
- Ademais, como se mostra nos exemplos anteriores, todas as variables deben levar unha condición de tipo de esquema  $R(t)$ .

A avaliación do átomo para unha combinación específica de tuplas pode devolver o valor verdadeiro (TRUE) ou falso (FALSE).

- $R(t)$  tomará o valor TRUE se asigna unha tupla de  $R$  a  $t$ , e dicir existe na relación unha tupla que cumpre a condición, senón, tomará o valor FALSE.
- “ $t1.a \text{ op } t2.b$ ”, “ $t1.a \text{ op } cte$ ”, ou “ $cte \text{ op } t2.b$ ” tomará o valor TRUE se  $t1$  e  $t2$  asígnanse a tuplas tales que os atributos especificados ( $a$  e  $b$ ) cumpren a condición, senón é así, será FALSE.

### Fórmula (ou condición) ben formada (fbf)

Unha fórmula está ben formada cando é correcta sintacticamente (está ben escrita). Para que unha fórmula sexa ben formada debe cumprir:

- Todo átomo é unha fórmula ben formada: “ $R(t)$ ”, “ $t1.a \text{ op } t1.b$ ”, “ $t1.a \text{ op } cte$ ”, “ $c \text{ op } t2.b$ ” son polo tanto fbf.
- Se  $F1$  e  $F2$  fórmulas atómicas ben formadas, tamén serán fbf:
  - $(F1 \text{ and } F2)$ ,  $(F1 \text{ or } F2)$ ,  $\text{not}(F1)$ ,  $\text{not}(F2)$  e os valores lóxicos derivados  $F1$  e  $F2$ , según a lóxica booleana.

### Variable de tupla libre e ligada nunha fbf

Dende un punto de vista informal  $t$  está ligada se está cuantificada, isto é, aparece en cláusulas  $(\forall t)$  o  $(\exists t)$ , senón está libre.

Dende un punto de vista formal a variable de tupla é libre ou ligada nunha fbf:

- Se  $F$  = átomo, calquera ocorrencia dunha variable de tupla  $t$ , está libre.
- En  $(F1 \text{ and } F2)$ ,  $(F1 \text{ or } F2)$ ,  $\text{not}(F1)$ ,  $\text{not}(F2)$  unha ocorrencia de  $t$  está libre ou ligada segundo o estea en  $F1$  o  $F2$  ( $t$  é libre se  $F$  tamén o é).
- Toda ocorrencia libre de  $t$  en  $F$  está ligada en  $F'$ , se
  - $F' = (\forall t)F$  ou ben  $F' = (\exists t)F$ , e  $t$  estará ligada ao cuantificador especificado en  $F'$ .

No exemplo:  $\{ \forall \text{MEMBRO\_XURADO}(mx) \rightarrow \text{not } \exists \text{ACTOR}(a) \text{ and } \forall \text{PUNTUACION}(p) \rightarrow (p.\text{codActor} = a.\text{codActor}) \text{ and } (p.\text{codXurado} = mx.\text{codXurado}) \}$ , as variables “ $mx$ ”, “ $a$ ” e “ $p$ ” son variables ligadas.

No exemplo,  $\{ t.\text{nota}, t.\text{codPapel}, t.\text{codActor} \mid \text{PUNTUACION}(t) \text{ and } (t.\text{CodXurado} = 222) \}$  a variable “ $t$ ” é libre.

Se  $F$  é unha fbf, tamén o é  $(\exists t)F$ , onde  $t$  é unha variable de tupla:

- $(\exists t)F$  é TRUE si  $F$  é TRUE para ao menos unha tupla asignada a ocorrencias libres de  $t$  en  $F$ , do contrario é FALSE.

Se  $F$  é unha fbf, tamén o é  $(\forall t)F$ , onde  $t$  é unha variable de tupla:

- $(\forall t)F$  é TRUE si  $F$  é TRUE para toda tupla (no universo) asignada a ocorrencias libres en  $t$  en  $F$  do contrario é FALSE.

## Fórmulas pechadas e abertas

Unha fórmula é pechada cando as súas variables están ligadas. Se a fórmula é pechada, entón representa unha expresión que será TRUE ou FALSE. Empréganse para declarar restricións.

No exemplo:  $\{ \forall \text{MEBRO\_XURADO}(mx) \rightarrow \text{not } \exists \text{ACTOR}(a) \text{ and } \forall \text{PUNTUACION}(p) \rightarrow (p.\text{codActor} = a.\text{codActor}) \text{ and } (p.\text{codXurado} = mx.\text{codXurado}) \}$ , as variables “mx”, “a” e “p” son variables ligadas e retorna o valor TRUE.

Se a fórmula é aberta (ten variables libres), entón representa unha consulta na que a súa avaliación devolverá os valores das súas variables libres que fan TRUE a fórmula.

No exemplo,  $\{ t.\text{nota}, t.\text{codPapel}, t.\text{codActor} \mid \text{PUNTUACION}(t) \text{ and } (t.\text{CodXurado} = 222) \}$  a variable “t” é libre e retorna dúas tuplas que fan TRUE a fórmula. Empréganse para facer consultas sobre a información.

## Expresións seguras

Unha expresión é segura cando o seu resultado é un número finito de tuplas. Débese ter en conta que ao usar cuantificadores ( $\forall, \exists$ ) ou negación (not), a expresión ha de ter sentido, isto é, **ser segura e non xerar unha relación infinita**

- Dominio dunha expresión do CRT: O dominio dunha expresión do cálculo relacional de tuplas son valores constantes na expresión ou que existen en calquera tupla das relacións ás que se referencia na expresión
  - Dominio  $(E = \{t \mid \text{not}(\text{ACTOR}(t))\}) =$  todos os valores de atributos de tuplas ACTOR.

Ademais, unha expresión é segura si todo valor do resultado pertence ao dominio da expresión e é insegura, cando no resultado inclúense tuplas (e, por tanto, valores) que non están na relación ACTOR (é dicir, que non pertencen o seu dominio).

## Resultado dunha CRT

O resultado dunha consulta en cálculo relacional é un conxunto de tuplas cuxos atributos son a unión dos atributos de todas as variables de tupla, máis os atributos indicados directamente.

No exemplo, sobre o seguinte modelo relacional:

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981
765	Xavier	28/02/1977

► ACTOR (codActor, nome, idade)

B:R, M:C

ACTUACION (codPapel, codActor, data)

ACTUACIÓN		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

Se realiza a seguinte consulta  $\{ a.\text{nome}, \text{act}.\text{data} \mid \text{ACTOR}(a) \text{ and } \text{ACTUACIÓN}(\text{act}) \text{ and } a.\text{CodActor} = \text{act}.\text{codActor} \}$ .

Nesta consulta os atributos de tuplas consultadas son nome. dataNacemento.

A condición filtra qué tuplas son incluídas nos conxunto resultado.

ACTOR		
codActor	nome	dataNacemento
456	Iria	03/03/1997
678	Ximena	07/09/1994
123	Breixo	01/01/1981
765	Xavier	28/02/1977

► ACTOR (codActor, nome, idade)

B:R, M:C

ACTUACION (codPapel, codActor, data)

{a.nome, act.data |  
 ACTOR(a) anda ACTUACIÓN(act)  
 and a.CodActor=act.codActor}

ACTUACIÓN		
codPapel	codActor	data
1	456	03/03/2015
4	456	03/03/2015
2	456	04/04/2015
3	456	04/04/2015
1	678	04/03/2015
2	123	04/03/2015

codActor	nome	data
456	Iria	03/03/2015
456	Iria	03/03/2015
456	Iria	04/04/2015
456	Iria	04/04/2015
678	Ximena	04/03/2015
123	Breixo	04/03/2015



Tarefa 11. Realizar consultas empregando cálculo relacional de tuplas

### Correspondencia do cálculo relacional coa linguaxe de consulta SQL

CALCULO RELACIONAL	SQL
<b>Variables de consulta</b>	Os termos que siguen a SELECT (con DISTINCT), salvo que en SELECT non existan variables de tuplas, só campos
<b>Condiciones de tipo R(t)</b>	Equivale a 'R as t' na cláusula FROM, pero non se concreta si é JOIN, producto cartesiano, etc.
<b>Condiciones con <math>\exists</math> e <math>\forall</math></b>	Se poden expresar con EXISTS, SOME e ALL –A maioría das veces $\exists$ se traduce simplemente nun esquema en FROM, que non aparece en SELECT –Si é difícil expresar un $\forall$ , se puede xugar con $\exists$ e a súa negación
<b>Outras condiciones</b>	Aparecen outras WHERE, ON, etc. –Poden volverse implícitas nun NATURAL JOIN
En principio non é común contemplar directamente no cálculo relacional (pero se puede): –Operacións de conxuntos: unión, intersección, diferenza, pertenza –Operacións de agregación: COUNT, AVG, MAX, etc. –Vistas e consultas anidadas ORDER BY non ten sentido xa que a orden das tuplas non existe no modelo relacional	

#### 1.2.3.2 Cálculo relacional de dominios (CRD)

O CR de dominio baséase na utilización das variables de dominio, que se diferencian das variables de tuplas porque toman valores dentro dun dominio en vez dunha relación.

A diferenza máis notable respecto ao cálculo relacional orientado á tuplas é que manexa unha forma de comparación chamada “condición de pertenza”, que ten o seguinte formato:

- Relación (atributo1:variable de dominio, atributo2:variable de dominio,...).

No exemplo, ACTOR ( $\text{dataNacemento} > 1980$ ,  $\text{dataNacemento} < 2000$ ) avaliarase como certa se na relación existe unha ou máis tuplas que teñan en  $\text{dataNacemento}$  un valor maior a “1980” e menor a “2000”.

A linguaxe máis característica baseada no cálculo relacional orientado ao dominio é o QBE ( Query ByExample), que dispón dun editor de pantallas que axuda a formular as consultas e que se converteu nunha paradigma de interface de usuario.

## 1.2.4 Linguaxes relacionais comerciais

### 1.2.4.1 SQL

A linguaxe SQL (polas súas siglas en inglés Structured Query Language- ou linguaxe de consulta estruturada) é unha linguaxe declarativa de acceso a bases de datos relacionais que permite especificar diversos tipos de operacións nelas. Unha das súas características é o manexo da álgebra e o cálculo relacional que permiten efectuar consultas co fin de recuperar, de forma sinxela, información de bases de datos, así como facer cambios nelas.

En 1970 os laboratorios de IBM definiron a linguaxe SEQUEL (Structured English Query Language) que mais tarde foi amplamente implementado polo sistema de xestión de bases de datos (SXBD) experimental System R, desenvolvido en 1977 tamén por IBM. Con todo, foi Oracle quen o introduciu por primeira vez en 1979 nun produto comercial.

O SEQUEL rematou sendo o predecesor de SQL, que é unha versión evolucionada do primeiro. SQL pasa a ser a linguaxe por excelencia dos diversos SXBD relacionais xurdidos nos anos seguintes e foi por fin estandarizado en 1986 polo ANSI, dando lugar á primeira versión estandar desta linguaxe, (SQL-86 ou SQL1). Ao ano seguinte este estándar é tamén adoptado pola ISO. Na actualidade o estándar é SQL2008.

A linguaxe SQL pódese utilizar de forma interactiva ou como linguaxe hóspede (embebido dentro doutra linguaxe de propósito xeral). En SQL Embebido, as sentenzas SQL vanse intercalando entre as sentenzas da linguaxe anfitrión, precedidas normalmente da partícula “EXEC SQL” para que poidan ser distinguidas polos precompiladores.

### 1.2.4.2 SQL dinámico

O concepto central do SQL Dinámico é non codificar de forma estática unha sentenza SQL. No seu lugar é o programa anfitrión que constrúe o texto da sentenza SQL para a súa execución.

### 1.2.4.3 QUEL

Linguaxe relacional baseado no cálculo relacional de tuplas, foi introducido polo SXBD Ingres da universidade de Berkley. A diferenza de outras linguaxes relacionais non inclúe os operadores alxebraicos.

### 1.2.4.4 QBE

Query-By-Example é o nome de outra linguaxe de manipulación de datos esta vez baseada no cálculo relacional de dominios. Desenvolta polos laboratorios de IBM a principios dos 70 está na actualidade fora de uso, pero foi o xerme de linguaxes empregadas por diferentes SXBD xa que emprega unha interface gráfica.

## 1.3 Tarefas

As tarefas propostas son as seguintes:

- Tarefa 1. Realizar consultas empregando o operador de selección.
- Tarefa 2. Realizar consultas empregando o operador de proxección.
- Tarefa 3. Realizar consultas empregando o operador de unión.
- Tarefa 4. Realizar consultas empregando o operador de diferenza.
- Tarefa 5. Realizar consultas empregando o operador de intersección.
- Tarefa 6. Realizar consultas empregando o operador de división.
- Tarefa 7. Realizar consultas empregando o operador de combinación.
- Tarefa 8. Realizar consultas empregando o operador de combinación natural.
- Tarefa 9. Realizar consultas empregando a agrupación e a agregación.
- Tarefa 10. Realizar consultas empregando a reunión externa (outer-join).
- Tarefa 11. Realizar consultas empregando cálculo relacional de tuplas.

Realice as tarefas sobre a seguinte base de datos formada polas táboas PROVEDORES, COMPOÑENTES, ARTIGOS, ENVIOS. En dita base de datos está almacenada a seguinte información:

ENVIOS			
codProveedor	codCompoñenete	codArtigo	Catidade
P1	C1	T1	200
P1	C1	T4	700
P2	C3	T1	400
P2	C3	T2	200
P2	C3	T3	200
P2	C3	T4	500
P2	C3	T5	600
P2	C3	T6	400
P2	C3	T7	800
P2	C5	T2	100
P3	C3	T1	200
P3	C4	T2	500
P4	C6	T3	300
P4	C6	T7	300
P5	C2	T2	200
P5	C2	T4	100
P5	C5	T4	500
P5	C5	T7	100
P5	C6	T2	200
P5	C1	T4	100
P5	C3	T4	200
P5	C4	T4	800
P5	C5	T5	400
P5	C6	T4	500

PROVEDOR			
codProveedor	nomeProveedor	categoria	cidade
P1	CARLOS	20	Lugo
P2	JUAN	10	Madrid
P3	JOSE	30	Ourense
P4	INMA	20	Vigo
P5	EVA	30	Ferrol

COMPONENTES				
codCompoñente	nomeCompoñente	cor	PESO	cidade
C1	X3A	vermello	12	Lugo
C2	B85	verde	17	Madrid
C3	C4B	azul	17	Pontevedra
C4	C4B	vermello	14	Lugo
C5	VT8	azul	12	Marid
C6	C30	vermello	19	Ourense

ARTIGO		
codArtigo	nomeArtigo	cidade
T1	CLASIFICADORA	Madrid
T2	PERFORADORA	Prontevedra
T3	LECTORA	Lugo
T4	CONSOLA	Ferrol
T5	MEZCLADORA	Vigo
T6	TERMINAL	Ourense
T7	CINTA	Ourense

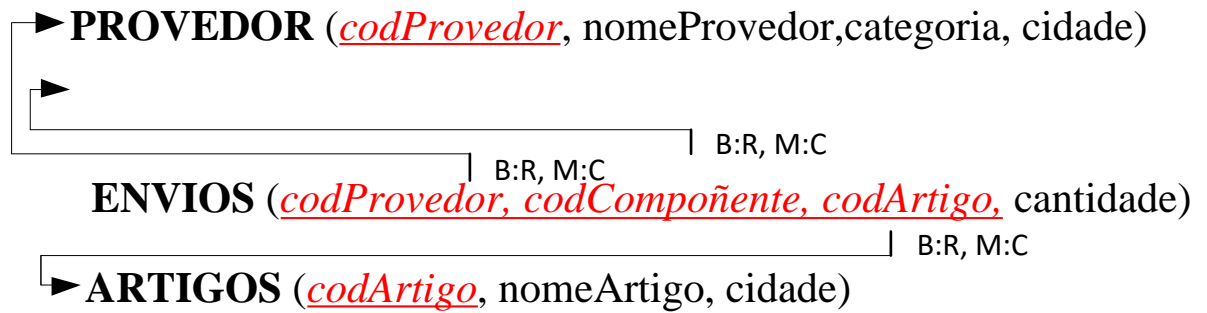
**PROVEDOR:** Representan os datos de provedores de compoñentes para a fabricación de artigos e a súa cidade de operación

**COMPOÑENTES:** Indica a información de pezas empregadas na fabricación dos artigos, indicándose o lugar de fabricación do compoñente.

**ARTIGO:** Información sobre os diferentes artigos que se fabrican e o lugar de montaxe do mesmos.

**ENVIOS:** Suministros realizados polos diferentes provedores de determinada cantidade de compoñentes, asignados para a elaboración do artigo correspondente.

O seu grafo relacional é como segue:



### 1.3.1 Tarefa 1. Realizar consultas empregando o operador de selección

Empregando o operado de selección obter todos os detalles dos artigos de “Ourense”

#### Solución

$\sigma(\text{cidade}=\text{Ourense})(\text{ARTIGO})$

codArtigo	nomeArtigo	cidade
T6	TERMINAL	Ourense
T7	CINTA	Ourense

### 1.3.2 Tarefa 2. Realizar consultas empregando o operador de proxección

- Tarefa 2.1 Empregando o operador de proxección e os xa vistos nos apartados anteriores obter todos os códigos de todos os provedores que abastecen do artigo con código T1
- Tarefa 2.2 Empregando o operador de proxección e os xa vistos nos apartados anteriores obter a lista de atributos ( cor e cidade) da táboa de COMPOÑENTES eliminando as tuplas repetidas
- Tarefa 2.3. Empregando o operador de proxección e os xa vistos nos apartados anteriores obter os valores de código de provedores que fornecen para o articulo T1 o compoñente C1

#### Solución

##### Tarefa 2.1

$\Pi_{\text{codProvedor}}(\sigma(\text{codArtigo} = \text{'T1'})(\text{ENVIOS}))$

codProvedor
P1
P2

##### Tarefa 2.2

$\Pi_{\text{cidade, cor}}(\text{COMPOÑENTES})$

cor	cidade
vermello	Lugo
verde	Madrid
azul	Pontevedra
azul	Marid
vermello	Ourense

### Tarefa 2.3

$\Pi \text{ CodProveedor}(\sigma((\text{codArtigo} = 'T1') \text{ and } (\text{codCompoñente} = 'C1'))(\text{ENVIOS}))$

codProveedor
P1

### 1.3.3 Tarefa 3. Realizar consultas empregando o operador de unión

Empregando o operador de unión e os xa vistos nos apartados anteriores codifique unha consulta que devolva todas as cidades coas que se traballa por ter algún proveedor, compoñente ou artigo.

#### Solución

$\Pi \text{ cidade}(\text{PROVEDOR}) \cup \Pi \text{ cidade}(\text{ARTIGO}) \cup \Pi \text{ cidade}(\text{COMPOÑENTE})$

### 1.3.4 Tarefa 4. Realizar consultas empregando o operador de diferenza

Empregando o operador de diferenza e os xa vistos nos apartados anteriores realice unha consulta que mostre as cidades coas que traballamos exclusivamente por ter compoñentes

#### Solución

$\Pi \text{ cidade}(\text{COMPOÑENTE}) - \Pi \text{ cidade}(\text{ARTIGO}) - \Pi \text{ cidade}(\text{PROVEDOR})$

### 1.3.5 Tarefa 5. Realizar consultas empregando o operador de intersección

Empregando o operador de intersección os xa vistos nos apartados anteriores obter

- Tarefa 5.1. valores para os provedores que fornecen os artigos T1 e T2
- Tarefa 5.2. mostrar as claves dos compoñentes que se fornecen para os artigo T1 e T2

#### Solución

##### Tarefa 5.1

$(\Pi \text{ codProveedor}(\sigma \text{ codArtigi} = 'T1')(\text{ENVIOS}))$

$\cap$

$(\Pi \text{ codProveedor}(\sigma \text{ codArtigo} = 'T1')(\text{ENVIOS}))$

<b>codProveedor</b>
P2
P3

Tarefa 5.2.

$(\Pi \text{ codCompñente}(\sigma \text{ codArtigo} = 'T1'(\text{ENVIOS})))$

$\cap$

$(\Pi \text{ codCompoñenet}(\sigma \text{ codArtigo} = 'T1'(\text{ENVIOS})))$

<b>codCompoñenete</b>
C3

### 1.3.6 Tarefa 6. Realizar consultas empregando o operador de división

Empregando o operador de división e os xa vistos nos apartados anteriores obter os códigos dos artigos para os que se provean envíos de todos os compoñentes existentes na base de datos

**Solución**

$\Pi \text{ codArtigo} ( \text{ENVIOS} \div (\Pi \text{ codCompoñente}(\text{COMPOÑENTES})))$

<b>codArtigo</b>
T4

### 1.3.7 Tarefa 7. Realizar consultas empregando o operador de combinación

- Tarefa 7.1 Empregando o operador de combinación e os xa vistos nos apartados anteriores obter valores de nome para os artigos do proveedor P1.
- Tarefa 7.2. Empregando o operador de combinación e os xa vistos nos apartados anteriores obter os valores de código para os compoñentes fornecidos por calquera artigo de “Madrid”

**Solución**

Tarefa 7.1

$\Pi \text{ nomeArugo}(\text{ARTIGOS} \bowtie \text{codProveedor} = \text{codProveedor}(\sigma \text{ codProveedor} = 'P1'(\text{ENVIOS})))$

<b>nomeArtigo</b>
CLASIFICADORA
CONSOLA



### Tarefa 7.2.

$\Pi_{\text{codCompoñente}}(\text{ENVIOS} \bowtie_{\text{codCompoñente}=\text{codCompoñente}}(\sigma_{\text{cidade}=\text{'Madrid'}}(\text{ARTIGOS})))$

codCompoñete
C1
C3

## 1.3.8 Tarefa 8. Realizar consultas empregando o operador de combinación natural

Empregando o operador de combinación natural e os xa vistos nos apartados anteriores obter

- Tarefa 8.1. Valores clave de Provedores que fornecen para un artigo de Lugo ou Madrid o compoñente vermello
- Tarefa 8.2. Valores de código dos compoñentes fornecidos para algún articulo de Ourense por un provedores de Madrid
- Tarefa 8.3. valores clave para os artigos que usan ao menos un compoñente que se pode obter có provedor P1
- Tarefa 8.4. Obter os códigos de provedores que fornecen o menos un compoñente fornecidos por un provedor que fornece polo menos un compoñente vermello
- Tarefa 8.5. Mostrar os datos dos envíos realizados de compoñentes con cor vermello
- Tarefa 8.6 mostrar os datos de envío e nome de cidade de aqueles envíos que cumbran que artigo, provedor,e compoñente son da mesma cidade

### Solución

#### Tarefa 8.1

$\Pi_{\text{codProvedor}}(\sigma_{\text{cor}=\text{'vermello'}}(\text{COMPOÑENTES}) * \text{ENVIOS} * \Pi_{\text{codArtigo}}(\sigma_{(\text{cidade}=\text{'Madrid'}} \text{ or } \text{cidade}=\text{'Lugo'})}(\text{ARTIGOS})))$

codProvedor
P1
P4

#### Tarefa 8.2.

$\Pi_{\text{codCompoñente}}((\sigma_{\text{cidade}=\text{'Madrid'}}(\text{PROVEDORES})) * \Pi_{\text{codCompoñente, codProvedor}}(\text{ENVIOS} * (\sigma_{\text{cidade}=\text{'Madrid'}}(\text{ARTIGOS}))))$

codCompoñete
C3

### Tarefa 8.3

$\Pi_{\text{codArtigo}} \square \text{ENVIOS} * \Pi_{\text{codCompoñente}} (\sigma_{\text{codProvedor}='P1'} (\text{ENVIOS}))$

<b>codArtigo</b>
T1
T4

### Tarefa 8.4

$\Pi_{\text{codProvedor}} \square \text{ENVIOS} * ((\Pi_{\text{codCompoñente}} \text{ENVIOS} * (\Pi_{\text{codProvedor}} (\text{ENVIOS} * (\sigma_{\text{color}='Vermello'} (\text{COMPOÑENETES}))))))$

<b>codProvedor</b>
P1
P2
P3
P4
P5

### Tarefa 8.5

$\Pi \text{ENVIOS} * (\text{ENVIOS} * \square (\sigma_{\text{color} \neq 'Vermello'} (\text{COMPOÑENETES})))$

<b>ENVIOS</b>			
<b>codProvedor</b>	<b>codCompoñenete</b>	<b>codArtigo</b>	<b>Catidade</b>
P5	C2	T4	100
P2	C3	T1	400
P3	C3	T1	200
P2	C3	T2	200
P2	C3	T3	200
P2	C3	T4	500
P5	C3	T4	200
P2	C3	T5	600
P2	C3	T6	400
P2	C3	T7	800
P2	C5	T2	100
P5	C5	T4	500
P5	C5	T5	400

#### Tarefa 8.6

$\Pi_{\text{codCompoñentes, codProvedores, cidades}} (\text{ARTIGOS} * (\text{PROVEDORES} * \text{ENVIOS} * \text{COMPOÑENETES})))$

### 1.3.9 Tarefa 9. Realizar consultas empregando a agrupación e a agregación

Empregando a agrupación e agregación xunto cos operadores xa vistos nos apartados anteriores obter:

- Tarefa 9.1: Valores de código de compoñentes tales que ningún outro teña un valor de peso inferior
- Tarefa 9.2: Para cada artigo e compoñente fornecido obter os códigos de compoñente e artigo e a cantidade total correspondente.

#### Solución

##### Tarefa 9.1

$\Pi_{\text{codCompoñenete}} (\text{COMPOÑENTES} \bowtie_{\text{peso=p}} (\text{FG}(\text{MIN}(\text{peso}))(\text{COMPOÑENETES}))$

codCompoñenete
C1
C5

##### Tarefa 9.2

$\Pi_{\text{codCompoñenete, codArtigo}} (\text{FG}(\text{SUM}(\text{Cantidad}))(\text{ENVIOS}))$

codCompoñente	codArtigo	Total
C1	T1	200
C1	T4	800
C2	T2	200
C2	T4	100
C3	T1	600
C3	T2	200
C3	T3	200
C3	T4	700
C3	T5	600
C3	T6	400
C3	T7	800
C4	T2	500
C4	T4	800
C5	T2	100
C5	T4	500
C5	T5	400
C5	T7	100
C6	T2	200
C6	T3	300
C6	T4	500

### 1.3.10 Tarefa 10 Realizar consultas empregando as reunións externas

Supoña que o estado actual da base de datos é a seguinte:

ENVIOS			
codProvedor	codCompoñente	codArtigo	Catidade
P1	C1	T1	200
P1	C1	T4	700
P2	C3	T1	400
P2	C3	T2	200
P2	C3	T3	200
P2	C3	T4	500
P2	C3	T5	600
P2	C3	T6	400
P2	C5	T2	100
P3	C3	T1	200
P3	C4	T2	500
P4	C6	T3	300

PROVEDOR			
codProvedor	nomeProvedor	categoria	cidade
P1	CARLOS	20	Lugo
P2	JUAN	10	Madrid
P3	JOSE	30	Ourense
P4	INMA	20	Vigo
P5	EVA	30	Ferrol

► **PROVEDOR** (*codProvedor*, nomeProvedor, categoria, cidade)  
 ► **COMPOÑENTE** (*codCompoñente*, nomeCompoñente, cor, peso, cidade)  
     B:R, M:C  
 ENVIOS (*codProvedor*, *codCompoñente*, *codArtigo*, cantidade)  
     B:R, M:C  
 ► **ARTIGOS** (*codArtigo*, nomeArtigo, cidade)

ARTIGO		
codArtigo	nomeArtigo	cidade
T1	CLASIFICADORA	Madrid
T2	PERFORADORA	Prontevedra
T3	LECTORA	Lugo
T4	CONSOLA	Ferrol
T5	MEZCLADORA	Vigo
T6	TERMINAL	Ourense
T7	CINTA	Ourense

COMPONENTES				
codCompoñente	nomeCompoñente	cor	PESO	cidade
C1	X3A	vermello	12	Lugo
C2	B85	verde	17	Madrid
C3	C4B	azul	17	Pontevedra
C4	C4B	vermello	14	Lugo
C5	VT8	azul	12	Marid
C6	C30	vermello	19	Ourense

- Tarefa 10.1: Indique a cláusula en álgebra relacional para obter a seguinte táboa resultado

codArtigo	nomeArtigo	cidade	codProvedor	codCompoñenete	codArtigo	Catidade
T1	CLASIFICADORA	Madrid	P1	C1	T1	200
T1	CLASIFICADORA	Madrid	P2	C3	T1	400
T1	CLASIFICADORA	Madrid	P3	C3	T1	200
T2	PERFORADORA	Prontevedra	P2	C3	T2	200
T2	PERFORADORA	Prontevedra	P2	C5	T2	100
T2	PERFORADORA	Prontevedra	P3	C4	T2	500
T3	LECTORA	Lugo	P2	C3	T3	200
T3	LECTORA	Lugo	P4	C6	T3	300
T4	CONSOLA	Ferrol	P1	C1	T4	700
T4	CONSOLA	Ferrol	P2	C3	T4	500
T5	MEZCLADORA	Vigo	P2	C3	T5	600
T6	TERMINAL	Ourense	P2	C3	T6	400
T7	CINTA	Ourense	NULO	NULO	NULO	NULO

- Tarefa 10.2.: Indique a cláusula en álgebra relacional para obter a seguinte táboa resultado

codProvedor	codCompoñenete	codArtigo	Catidade	codCompoñente	nomeCompoñente	cor	PESO	cidade
P1	C1	T1	200	C1	X3A	vermello	12	Lugo
P1	C1	T4	700	C1	X3A	vermello	12	Lugo
P2	C3	T1	400	C3	C4B	azul	17	Pontevedra
P2	C3	T2	200	C3	C4B	azul	17	Pontevedra
P2	C3	T3	200	C3	C4B	azul	17	Pontevedra
P2	C3	T4	500	C3	C4B	azul	17	Pontevedra
P2	C3	T5	600	C3	C4B	azul	17	Pontevedra
P2	C3	T6	400	C3	C4B	azul	17	Pontevedra
P2	C5	T2	100	C5	VT8	azul	12	Marid
P3	C3	T1	200	C3	C4B	azul	17	Pontevedra
P3	C4	T2	500	C4	C4B	vermello	14	Lugo
P4	C6	T3	300	C6	C30	vermello	19	Ourense
NULO	NULO	NULO	NULO	C2	B85	verde	17	Madrid

### Solución:

#### Tarefa 10.1

ARTIGO \*/ ENVIOS

#### Tarefa 10.2

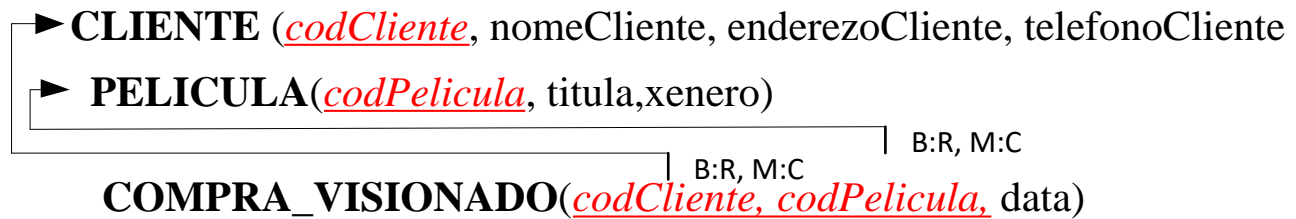
ENVIOS /\* COMPOÑENTES

### 1.3.11 Tarefa 11 Realizar consultas empregando cálculo relacional de tuplas

Para o mesmo esquema da tarefa 1 resolver as seguintes consultas mediante o cálculo relacional de tuplas.

- Tarefa 11.1: Nome dos provedores que fornecen o compoñente C2
- Tarefa 11.2: Nome dos provedores que fornecidos polo menos un compoñente vermello
- Tarefa 11.3.: Obter o nome dos compoñentes de cor vermella fornecidos por provedores da cidade de Lugo
- Tarefa 11.4: Obter o código dos provedores que tamén fornecen algún compoñente fornecido polo provedor P2
- Tarefa 11.5: Obter os datos dos envíos de máis de 100 unidades, mostrando tamén o nome de provedor e compoñente
- Tarefa 11.6: Obter o nome dos provedores que fornecen todas as pezas

- Tarefa 11.7: Para o grafo relacional que mostra unha versión reducida dunha compañía dedicada á venda de visionado de películas on-line resolver , obter o título das películas prestadas a todos os clientes, mediante o cálculo relacional de tuplas.



## Solución

### Tarefa 11.1

$\{e.codCompoñente, e.codProvedor | \exists ENVIOS(e) \rightarrow (ENVIOS.codProvedor = PROVIDOR.codProvedor \text{ and } ENVIOS.codProvedor = 'P2')\}$

### Tarefa 11.2

$\{p.codProvedor, p.nomeProvedor | PROVIDOR(p) \text{ and } \exists ENVIOS(e) \rightarrow (ENVIOS.codProvedor = PROVIDOR.codProvedor \text{ and } (\exists COMPOÑENTE(c) \text{ and } ENVIOS.codCompoñente = c.codCompoñente \text{ and } c.cor = 'Vermello'))\}$

### Tarefa 11.3

$\{c.codProvedor, c.nomeProvedor | COMPOÑENETE(c) \text{ and } c.cor = 'Vermello' \text{ and } \exists ENVIOS(e) \rightarrow (e.codProvedor = PROVIDOR.codProvedor \text{ and } (\exists PROVIDOR(p) \rightarrow PROVIDOR.codProvedor = c.codProvedor \text{ and } c.cidade = 'Lugo'))\}$

### Tarefa 11.4

$\{e.codProvedor | ENVIOS(e) \text{ and } \exists ENVIOS(e2) \rightarrow (e2.codProvedor = 'P2' \text{ and } e.CodCompoñente)\}$

### Tarefa 11.5

$\{e.codProvedor, p.nomeProvedor, c.codCompoñente, c.nomeCompoñente, e.cantidad | ENVIOS(e) \text{ and } PROVIDOR(p) \text{ and } COMPOÑENET(c) \text{ and } (e.conCompoñente = c.codCompoñente \text{ and } e.codProvedor = p.codProvedor \text{ and } e.cantidad > 100)\}$

### Tarefa 11.6

$\{p.codProvedor, p.nomeProvedor | PROVIDOR(p) \text{ and } ENVIOS(e) \forall p \text{ and } \exists e \rightarrow (e.codCompoñente = p.codProvedor \text{ and } e.codCompoñente = c.codCompoñente)\}$

Outra posible solución:

$\{p.codProvedor, p.nomeProvedor | PROVIDOR(p) \text{ and } ENVIOS(e) \text{ and not } \exists p \text{ and not } \exists e \rightarrow (e.codProvedor = p.codProvedor \text{ and } e.codCompoñente = c.codCompoñente)\}$

### Tarefa 11.7

Outro xeito de expresalo será: título das películas que para todos e cada un dos clientes exista polo menos un préstamo (correspondente a cada socio) de esa película:

$\{pe.codPelícula, pe.título | PELICULA(pe) \text{ and } \forall CLIENTE(c) \text{ and } \exists PRESTAMO(p) \text{ and } (c.codCliente = p.codCliente \text{ and } p.codPelícula = pe.codPelícula)\}$

## 2. Materiais

---

### 2.1 Documentos de apoio ou referencia

- [EN 2002] ELMASRI, R.;NAVATHE, S.B.*Fundamentos de Sistemas de Bases de Datos* Addison-Wesley, 2002.
- [MPM 1999] DE MIGUEL, A; PIATTINI, M; MARCOS, E. *.Diseño de base de datos relacionales*. Ra-MA. 1999
- CONNOLLY, T; BEGG, C; STRACHAN, A. *Database system: A practical aproach de-sing, implementation and magnagement*.Addisson-Wesley, 1998
- SILBERSCHATZ,A; KORTH. H; SUDARSHAN, S; CONNOLLY, T; BEGG, C; STRACHAN, A. *Fundamentos de bases de datos*. McGraw-Hill, 1998
- DATE,C.J. *Introducción a los sistemas de bases de datos*. Addisson-Wesley, 1992
- DE MIGUEL, A; PIATTINI, M. *Concepción y diseño de bases de datos*. Ra-Ma, 1993
- DE MIGUEL, A; PIATTINI, M. *Fundamentos y modelos de bases de datos*. Ra-Ma, 1993
- Métrica:
  - [http://administracionelectronica.gob.es/pae/Home/pae/Documentacion/pae/Metodolog/pae/Metrica\\_v3.html#.ViS6eGThBz8](http://administracionelectronica.gob.es/pae/Home/pae/Documentacion/pae/Metodolog/pae/Metrica_v3.html#.ViS6eGThBz8)