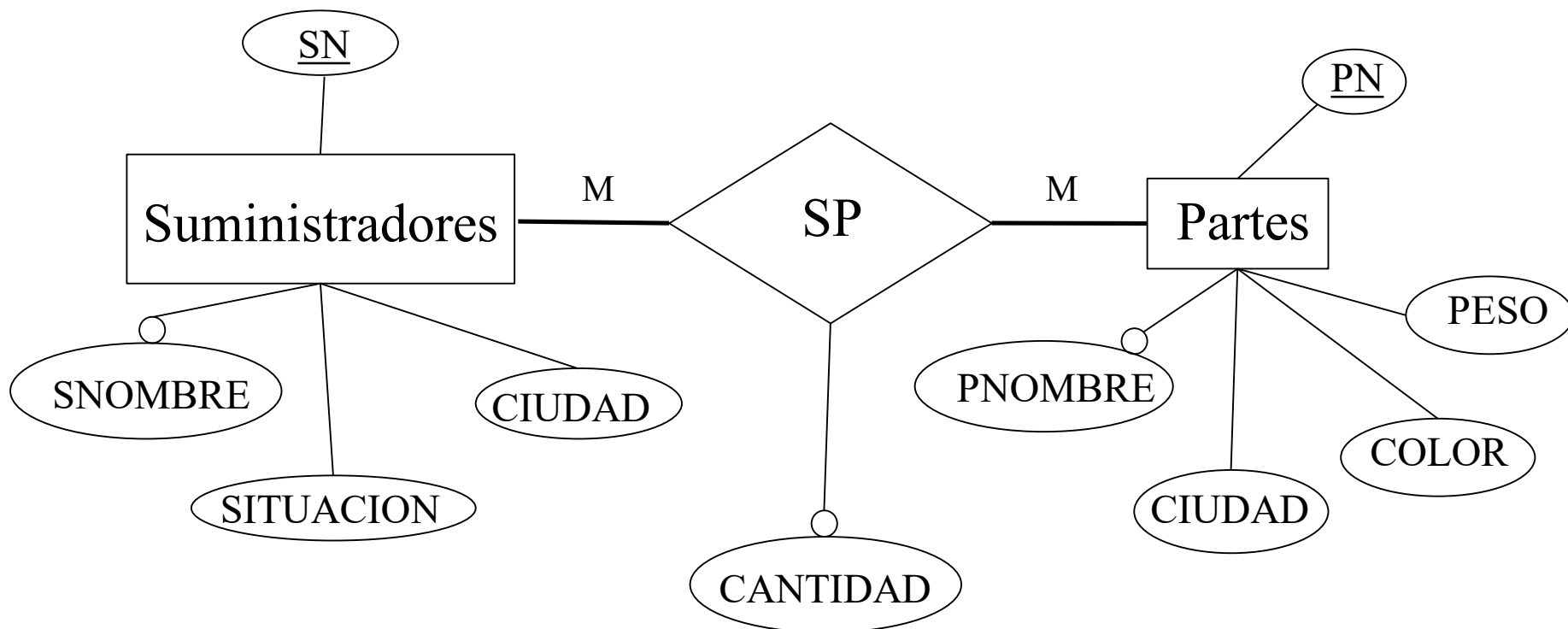


# Tema 3.2 .-Álgebra relacional. (2 de 4)

# BD para los ejemplos

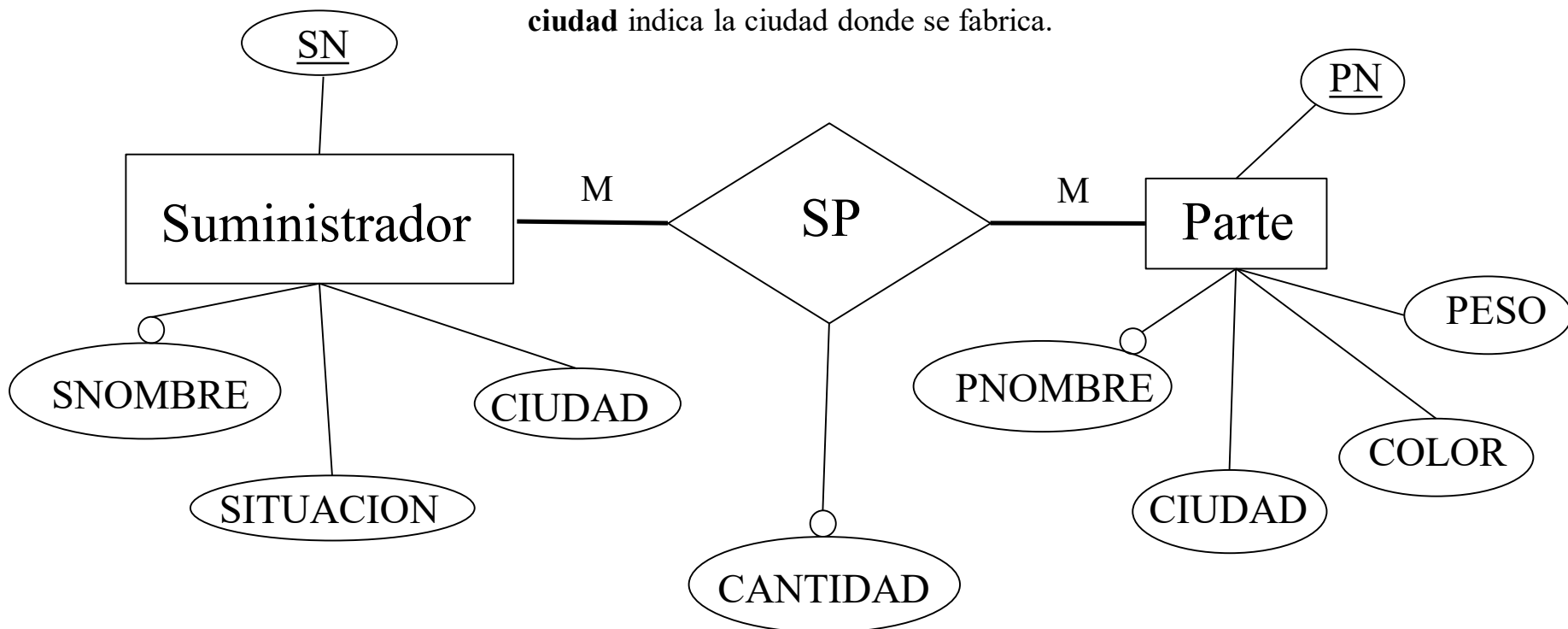
# Esquema Entidad/Interrelación

- Este esquema es parcial, pero de momento nos basta:
  - Representa una interrelación binaria en una empresa, entre las **partes** usadas como materia prima y los **proveedores** que las suministran
  - Una ocurrencia de **SP** une a un **suministrador** y una **parte** si dicho proveedor ha vendido esa parte alguna vez a la empresa
    - Indicando en la propiedad **cantidad** la suma total de la cantidad de todas las ventas en las que ese proveedor me ha vendido ese producto
    - Esquivando con ello el problema de historial que aparecería si pretendiese representar cada factura de venta individualmente



# Esquema Entidad/Interrelación

- En la entidad **suministrador**:
  - La propiedad **sn** es un código identificador, **snombre** es el nombre del proveedor, **situacion** es un valor numérico que cuantifica la importancia que dicho suministrador tiene para la empresa y **ciudad** indica la ciudad donde reside el proveedor.
- En la entidad **parte**:
  - La propiedad **pn** es un código identificador establecido por la empresa, **pnombre**, **peso** y **color** son el nombre, el peso y el color del producto y **ciudad** indica la ciudad donde se fabrica.



# Esquema Relacional

❖ Si llevamos el esquema E/IR al modelo relacional se obtienen tres relaciones expresadas en pseudo-SQL:

CT s (sn char(3),  
snombre char(25) NN,  
situacion int,  
ciudad char(25))  
PK (sn);

CT p (pn char(3),  
pnombre char(25) NN,  
peso int,  
color char(25),  
ciudad char(25))  
PK (pn);

CT sp (sn char(3),  
pn char(3),  
cantidad int NN)  
PK (sn, pn)  
FK (sn) REF s DR UR  
FK (pn) REF p DR UR;

# Esquema Entidad/Interrelación

La situación ejemplo de la BD sobre la que vamos a trabajar es:

| SN | SNOMBRE | SITUACION | CIUDAD      |
|----|---------|-----------|-------------|
| S1 | Luis    | 100       | Telde       |
| S2 | Carlos  | 200       | Agaete      |
| S3 | Pedro   | 200       | Galdar      |
| S4 | Antonio | 300       | Telde       |
| S5 | Juan    | 100       | Guía        |
| S6 | Juan    | 100       | Firgas      |
| S7 | María   | 500       | <b>null</b> |

| PN | PNOMBRE  | PESO | COLOR       | CIUDAD      |
|----|----------|------|-------------|-------------|
| P1 | Tornillo | 10   | Acero       | Telde       |
| P2 | Tuerca   | 5    | <b>null</b> | Agaete      |
| P3 | Arandela | 12   | <b>null</b> | <b>null</b> |
| P4 | Bombilla | 20   | Blanco      | Telde       |
| P5 | Enchufe  | 16   | Negro       | Galdar      |

| SN | PN | CANTIDAD |
|----|----|----------|
| S1 | P1 | 100      |
| S3 | P1 | 50       |
| S4 | P1 | 200      |
| S1 | P2 | 100      |
| S2 | P2 | 200      |
| S1 | P3 | 50       |
| S2 | P3 | 100      |
| S3 | P3 | 150      |
| S4 | P3 | 200      |
| S5 | P3 | 150      |
| S6 | P3 | 150      |

| SN | PN | CANTIDAD |
|----|----|----------|
| S7 | P3 | 50       |
| S1 | P4 | 100      |
| S2 | P4 | 150      |
| S4 | P4 | 200      |
| S1 | P5 | 250      |
| S2 | P5 | 100      |
| S3 | P5 | 150      |
| S4 | P5 | 50       |
| S5 | P5 | 50       |
| S6 | P5 | 150      |
| S7 | P5 | 200      |

# Operadores Relacionales Especiales

# Operador de Restricción

- Tiene el formato:
  - *'Relación' WHERE 'condición'.*
- *'Relación'* hace referencia a la relación sobre la que se aplica la restricción
- *'condición'* hace referencia a cualquier expresión booleana formada por:
  - los atributos de **'Relación'** y/o constantes
  - los operadores relacionales  $<$ ,  $>$ ,  $>=$ ,  $=<$ , etc,
  - los operadores lógicos AND, OR y NOT y
  - los paréntesis cuando sea necesario.
- El operador de **restricción** produce como resultado una relación cuya cabecera es idéntica a la de **'Relación'** y su cuerpo está restringido, de forma que sólo tiene aquellas filas de **'Relación'** que cumplen **'condición'**
  - Constituye una *'selección de tuplas': Selección Horizontal*





GRUPO DE ESTRUCTURAS DE DATOS

# Ejemplo de restricción

 Ejemplo:

s

| SN | SNOMBRE | SITUACION | CIUDAD      |
|----|---------|-----------|-------------|
| S1 | Luis    | 100       | Telde       |
| S2 | Carlos  | 200       | Agaete      |
| S3 | Pedro   | 200       | Galdar      |
| S4 | Antonio | 300       | Telde       |
| S5 | Juan    | 100       | Guía        |
| S6 | Juan    | 100       | Firgas      |
| S7 | María   | 500       | <b>null</b> |

s where situacion>100

| SN | SNOMBRE | SITUACION | CIUDAD      |
|----|---------|-----------|-------------|
| S2 | Carlos  | 200       | Agaete      |
| S3 | Pedro   | 200       | Galdar      |
| S4 | Antonio | 300       | Telde       |
| S7 | María   | 500       | <b>null</b> |



GRUPO DE ESTRUCTURAS DE DATOS

# Ejemplo de restricción

 Ejemplo:

p

| PN | PNOMBRE  | PESO | COLOR       | CIUDAD      |
|----|----------|------|-------------|-------------|
| P1 | Tornillo | 10   | Acero       | Telde       |
| P2 | Tuerca   | 5    | <b>null</b> | Agaete      |
| P3 | Arandela | 12   | <b>null</b> | <b>null</b> |
| P4 | Bombilla | 20   | Blanco      | Telde       |
| P5 | Enchufe  | 16   | Negro       | Galdar      |

p where peso<14:

| PN | PNOMBRE  | PESO | COLOR       | CIUDAD      |
|----|----------|------|-------------|-------------|
| P1 | Tornillo | 10   | Acero       | Telde       |
| P2 | Tuerca   | 5    | <b>null</b> | Agaete      |
| P3 | Arandela | 12   | <b>null</b> | <b>null</b> |

# Ejemplo de restricción

 Ejemplo:

s

| SN | SNOMBRE | SITUACION | CIUDAD      |
|----|---------|-----------|-------------|
| S1 | Luis    | 100       | Telde       |
| S2 | Carlos  | 200       | Agaete      |
| S3 | Pedro   | 200       | Galdar      |
| S4 | Antonio | 300       | Telde       |
| S5 | Juan    | 100       | Guía        |
| S6 | Juan    | 100       | Firgas      |
| S7 | María   | 500       | <b>null</b> |

(s where situacion>100) where ciudad='Telde'

| SN | SNOMBRE | SITUACION | CIUDAD |
|----|---------|-----------|--------|
| S4 | Antonio | 300       | Telde  |

s where situacion>100 and ciudad='Telde'

| SN | SNOMBRE | SITUACION | CIUDAD |
|----|---------|-----------|--------|
| S4 | Antonio | 300       | Telde  |

# Operador de Proyección

- ❑ Tiene el formato:
  - ❑ **'Relación'**[Atributo1, Atributo2, ..., AtributoN] .
- ❑ **'Relación'** hace referencia al nombre de la relación sobre la que se aplica la proyección.
- ❑ Los atributos especificados son aquellos seleccionados para formar parte de la cabecera de la relación resultante.
- ❑ El operador de proyección produce como resultado una relación cuya cabecera tiene sólo los atributos seleccionados y cuyo cuerpo está formado por las tuplas de la relación original eliminando aquellas tuplas duplicadas
  - ❑ Se consideran duplicadas aquellas tuplas en las que todos los atributos seleccionados son iguales.
- ❑ Constituye una **'selección de atributos': Selección Vertical**.

# Ejemplo de proyección

## Ejemplo

**S**

| SN | SNOMBRE | SITUACION | CIUDAD      |
|----|---------|-----------|-------------|
| S1 | Luis    | 100       | Telde       |
| S2 | Carlos  | 200       | Agaete      |
| S3 | Pedro   | 200       | Galdar      |
| S4 | Antonio | 300       | Telde       |
| S5 | Juan    | 100       | Guía        |
| S6 | Juan    | 100       | Firgas      |
| S7 | María   | 500       | <b>null</b> |

**s [snombre, situacion]**

| SNOMBRE | SITUACION |
|---------|-----------|
| Luis    | 100       |
| Carlos  | 200       |
| Pedro   | 200       |
| Antonio | 300       |
| Juan    | 100       |
| María   | 500       |

# Ejemplo de proyección

## Ejemplo

**sp**

| SN | PN | CANTIDAD | SN | PN | CANTIDAD |
|----|----|----------|----|----|----------|
| S1 | P1 | 100      | S7 | P3 | 50       |
| S3 | P1 | 50       | S1 | P4 | 100      |
| S4 | P1 | 200      | S2 | P4 | 150      |
| S1 | P2 | 100      | S4 | P4 | 200      |
| S2 | P2 | 200      | S1 | P5 | 250      |
| S1 | P3 | 50       | S2 | P5 | 100      |
| S2 | P3 | 100      | S3 | P5 | 150      |
| S3 | P3 | 150      | S4 | P5 | 50       |
| S4 | P3 | 200      | S5 | P5 | 50       |
| S5 | P3 | 150      | S6 | P5 | 150      |
| S6 | P3 | 150      | S7 | P5 | 200      |

**(sp where cantidad <100) [sn, pn]**

| SN | PN |
|----|----|
| S3 | P1 |
| S1 | P3 |
| S7 | P3 |
| S4 | P5 |
| S5 | P5 |

# Derivados del Producto Cartesiano

- El operador binario **reunión- $\theta$**  (se lee reunión-theta) se define de la siguiente forma:
  - Sean  $a$  y  $b$  dos relaciones compatibles para el producto, y  $\theta$  una condición que compara atributos de ambas tablas
  - La operación  $(a \text{ reunión-}\theta b)$  equivale a;
    - $(a \text{ times } b) \text{ where } \theta$
    - De tal forma que resulta ser un producto cartesiano restringido.
- Si el único operador de comparación usado en la restricción es la igualdad, entonces se denomina **equi-reunión**.



GRUPO DE ESTRUCTURAS DE DATOS

# Operador de Reunión natural

- ✓ Sean **a** y **b** dos relaciones **no necesariamente** compatibles para el producto, de tal forma que:
  - ✓ llamaremos **A** al atributo compuesto por los atributos de la relación **a** que no están en la relación **b**
  - ✓ llamaremos **B** al atributo compuesto por los atributos de la relación **b** que no están en la relación **a**
  - ✓ y llamaremos **C** al atributo compuesto por los atributos de la relación **a** que también están en la relación **b** (o viceversa, da igual)
- ✓ de forma que la cabecera de la relación **a** está compuesta por (**A,C**) y la cabecera de la relación **b** está compuesta por (**B,C**)
  - ✓ Supóngase que **C** está formado por atributos simples  $C_i$  ( $i=1..n$ )
- ✓ La operación de reunión natural se expresa sintácticamente como:
  - ✓ **a join b**
  - ✓ y produce como resultado una relación que tiene como cabecera los atributos, posiblemente compuestos, **A**, **B** y **C**
  - ✓ y como cuerpo el de:
    - ✓  $((a \text{ times } (b \text{ rename } C_{i(i=1..n)} \text{ as } D_{i(i=1..n)})) \text{ where } C_i=D_i (i=1..n)) [A, B, C]$
  - ✓ Por tanto, la reunión natural es una equi-reunión con los atributos duplicados eliminados y en la que la equi-condición se establece siempre por defecto sobre las parejas de atributos con el mismo nombre en ambas relaciones
- ✓ Si el conjunto de atributos **C** es vacío, la reunión natural degenera en un mero producto cartesiano



# Derivados del Producto Cartesiano

- Obsérvese que la **reunión natural** trabaja de forma diferente a los derivados del producto cartesiano presentados previamente:
  - La **reunión- $\theta$**  y la **equi-reunión** exigen que las relaciones operandos sean compatibles para el producto
  - La reunión natural no exige dicha compatibilidad y solventa el problema de la repetición de nombres de atributos en la cabecera de la relación resultante mediante una proyección
    - Aparte, por supuesto, del criterio por defecto para la equi-condición de restricción

## Reuniones internas

- Lo que sí comparten los tres es la existencia de la condición de restricción del cuerpo del producto cartesiano
  - causando que si algún tuple de una de las relaciones operandos no cumple dicha condición frente a ningún tuple de la otra relación operando,
    - entonces dicho tuple desaparece del cuerpo de la relación resultado
  - En otras palabras: “el que no encuentra pareja, no sale en la foto”
- Eso se expresa denominando a estas reuniones:
  - **Reuniones internas**

# Reuniones externas

- En las *reuniones externas*, los tuples de una de las relaciones operandos que no cumplen la condición de restricción frente a ningún tuple de la otra relación operando, podrían permanecer en el cuerpo de la relación resultado emparejados con valores nulos en los atributos correspondientes a la otra relación operando:
  - **Reunión natural izquierda:**
    - (a left-join b) tiene el mismo cuerpo que (a join b) y además cada tuple de **a** que no resulta emparejado con ningún tuple de **b** se incluye en el cuerpo del resultado con los atributos procedentes de **b** con valor nulo.
  - **Reunión natural derecha:**
    - (a right-join b) tiene el mismo cuerpo que (a join b) y además cada tuple de **b** que no resulta emparejado con ningún tuple de **a** se incluye en el cuerpo del resultado con los atributos procedentes de **a** con valor nulo.
  - **Reunión natural completa:**
    - (a full-join b) tiene el mismo cuerpo que (a join b) y además cada tuple de **a** que no resulta emparejado con ningún tuple de **b** se incluye en el cuerpo del resultado con los atributos procedentes de **b** con valor nulo y cada tuple de **b** que no resulta emparejado con ningún tuple de **a** se incluye en el cuerpo del resultado con los atributos procedentes de **a** con valor nulo.
- Se han definido los tres tipos de **reuniones naturales externas** pero de forma similar se podrían definir:
  - los tres tipos de **reuniones- $\theta$  externas** y
  - los tres tipos de **equi-reuniones externas**

# Propiedades del operador de reunión natural

- En adelante, si no se indica lo contrario, siempre se entenderá que se trata de reuniones internas, así pues la reunión natural:
  - Es conmutativa:
    - $a \text{ join } b = b \text{ join } a$
  - Es asociativa:
    - $(a \text{ join } b) \text{ join } c = a \text{ join } (b \text{ join } c)$
    - por tanto se puede escribir  $a \text{ join } b \text{ join } c$ , sin ambigüedad

# Ejemplo de reunión natural

S



| SN | SNOMBRE | SITUACION | CIUDAD |
|----|---------|-----------|--------|
| S1 | Luis    | 100       | Telde  |
| S2 | Carlos  | 200       | Agaete |
| S3 | Pedro   | 200       | Galdar |
| S4 | Antonio | 300       | Telde  |
| S5 | Juan    | 100       | Guía   |
| S6 | Juan    | 100       | Firgas |
| S7 | María   | 500       | null   |

p



| PN | PNOMBRE  | PESO | COLOR  | CIUDAD |
|----|----------|------|--------|--------|
| P1 | Tornillo | 10   | Acero  | Telde  |
| P2 | Tuerca   | 5    | null   | Agaete |
| P3 | Arandela | 12   | null   | null   |
| P4 | Bombilla | 20   | Blanco | Telde  |
| P5 | Enchufe  | 16   | Negro  | Galdar |

s join p

| SN | SNOMBRE | SITUACION | CIUDAD | PN | PNOMBRE  | PESO | COLOR  |
|----|---------|-----------|--------|----|----------|------|--------|
| S1 | Luis    | 100       | Telde  | P1 | Tornillo | 10   | Acero  |
| S1 | Luis    | 100       | Telde  | P4 | Bombilla | 20   | Blanco |
| S2 | Carlos  | 200       | Agaete | P2 | Tuerca   | 5    | null   |
| S3 | Pedro   | 200       | Galdar | P5 | Enchufe  | 16   | Negro  |
| S4 | Antonio | 300       | Telde  | P1 | Tornillo | 10   | Acero  |
| S4 | Antonio | 300       | Telde  | P4 | Bombilla | 20   | Blanco |

# Operador de División

- Sean **a** una relación con una cabecera formada por los atributos **A** y **B**, posiblemente compuestos, y denominada en adelante relación **dividendo**
- y **b** otra relación con una cabecera formada por el atributo **B**, posiblemente compuesto, y denominada en adelante relación **divisor**
- El operador de **división** relacional
  - se expresa sintácticamente como: **a div b**
  - y produce como resultado una relación denominada relación **cociente** cuya cabecera está formada por el atributo **A**
    - o sea, los atributos del **dividendo** que no están en el **divisor**
  - y cuyo cuerpo está formado por
    - los valores de **A** que aparecen en los tuples del **dividendo** asociados a todos y cada uno de los valores de **B** que aparecen en los tuples del divisor



GRUPO DE ESTRUCTURAS DE DATOS

# Ejemplo de división

↪ dividendo div divisor

↪ Si las relaciones son:

dividendo

| SN | PN |
|----|----|
| S1 | P1 |
| S1 | P2 |
| S1 | P3 |
| S1 | P4 |
| S1 | P5 |
| S1 | P6 |
| S2 | P1 |
| S2 | P2 |
| S3 | P2 |
| S4 | P2 |
| S4 | P4 |
| S4 | P5 |

divisor

| PN |
|----|
| P1 |

cociente

| SN |
|----|
| S1 |
| S2 |



GRUPO DE ESTRUCTURAS DE DATOS

# Ejemplo de división

⌞ dividendo div divisor

⌞ Si las relaciones son:

dividendo

| SN | PN |
|----|----|
| S1 | P1 |
| S1 | P2 |
| S1 | P3 |
| S1 | P4 |
| S1 | P5 |
| S1 | P6 |
| S2 | P1 |
| S2 | P2 |
| S3 | P2 |
| S4 | P2 |
| S4 | P4 |
| S4 | P5 |

divisor

| PN |
|----|
| P2 |
| P4 |

cociente

| SN |
|----|
| S1 |
| S4 |





GRUPO DE ESTRUCTURAS DE DATOS

# Ejemplo de división

⌞ dividendo div divisor

⌞ Si las relaciones son:

dividendo

| SN | PN |
|----|----|
| S1 | P1 |
| S1 | P2 |
| S1 | P3 |
| S1 | P4 |
| S1 | P5 |
| S1 | P6 |
| S2 | P1 |
| S2 | P2 |
| S3 | P2 |
| S4 | P2 |
| S4 | P4 |
| S4 | P5 |

divisor

| PN |
|----|
| P1 |
| P2 |
| P3 |
| P4 |
| P6 |

cociente

| SN |
|----|
| S1 |

# Operadores básicos

- Los operadores expuestos se pueden dividir en dos grupos:
  - **Operadores básicos:**
    - Renombrado de atributos: (rename) ►  $r_1$  rename  $A_i$  as  $B_j$
    - Unión:  $\cup$  (union) ►  $r_1$  union  $r_2$ 
      - Exige compatibilidad para la unión
    - Diferencia:  $-$  (minus) ►  $r_1$  minus  $r_2$ 
      - Exige compatibilidad para la unión
    - Producto cartesiano:  $\times$  (times) ►  $r_1$  times  $r_2$ 
      - Exige compatibilidad para el producto
    - Restricción: (where) ►  $r$  where condición
      - Cardinalidad del resultado entre 0 y cardinalidad de  $r$
    - Proyección:  $([])$  ►  $r [A_i (i=1..n)]$ 
      - ( $n \leq m$ ) siendo  $r$  una relación de grado  $m$
      - Elimina tuples duplicados

# Operadores derivados

- **Operadores derivados**, que pueden expresarse como combinaciones de los operadores básicos:
  - Intersección:  $\cap$  (intersect) ►  $r1 \text{ intersect } r2$ 
    - Exige compatibilidad para la unión
    - $r1 \text{ intersect } r2 = r1 \text{ minus } (r1 \text{ minus } r2) = r2 \text{ minus } (r2 \text{ minus } r1)$
  - Reunión natural: (join) ►  $r1 \text{ join } r2$ 
    - $r1 \text{ join } r2 = ((r1 \text{ times } (r2 \text{ rename } C_{i(i=1..n)} \text{ as } D_{i(i=1..n)})) \text{ where } C_i=D_i (i=1..n)) [A, B, C]$ 
      - Siendo (A,C) la cabecera de **r1** y (B,C) la cabecera de **r2**
      - y correspondiendo los atributos **A**, **B** y **C** a las definiciones dadas en la diapositiva 16 que presenta al operador de reunión natural.
  - División: (div) ►  $r1 \text{ div } r2$ 
    - $r1 \text{ div } r2 = r1[A] \text{ minus } ((r1[A] \text{ times } r2) \text{ minus } r1)[A]$ 
      - Siendo A el atributo compuesto por los atributos de la cabecera del dividendo que no están en la cabecera del divisor
- Pueden combinarse todos ellos, básicos y derivados, para crear expresiones complejas.
- En caso de necesidad pueden emplearse los paréntesis para especificar precedencias entre los operadores.

# Metodología básica

- Llegados a este punto, podemos reconocer la utilidad del estudio del álgebra relacional como lenguaje sobre las relaciones de una base de datos que permite traducir un enunciado de búsqueda en lenguaje natural:
  - y expresar la forma en que se va a realizar dicha búsqueda
  - a diferencia del SQL en el cual sólo se define lo que se quiere obtener y no cómo se desea obtenerlo
- La primera acción a tomar ante un enunciado de búsqueda complejo en lenguaje natural es desambiguarlo
  - es decir, averiguar con exactitud el significado que el enunciante da a sus palabras en lenguaje natural
- Una vez desambiguado, hay que decidir qué relaciones están implicadas en la búsqueda
  - bien por contener información solicitada por el resultado
  - bien por contener información usada para evaluar condiciones
- Por último habrá que decidir los operadores algebraicos a aplicar
  - y la precedencia entre ellos, que será expresada con paréntesis
- Una de las características del álgebra relacional es que se pueden encontrar diversas soluciones para el mismo enunciado
  - que corresponden a diferentes estrategias de búsqueda
- Evaluar cual de las soluciones es la óptima no es materia de estudio de éste tema

# Ejemplos básicos

1. Obtener las ciudades dónde se fabrican partes rojas:
  1. (p where color='Rojo')[ciudad]
  2. ~~p[ciudad] where color='Rojo'~~
2. Obtener los nombres de proveedores y/o de partes cuya ciudad es Telde:
  1. ((s where ciudad='Telde')[snombre] rename snombre as nombre)  
union  
((p where ciudad='Telde')[pnombre] rename pnombre as nombre)
3. Obtener los códigos de los proveedores que venden la parte 'P2':
  1. (sp where pn='P2')[sn]
4. Obtener los nombres de los proveedores que venden la parte 'P2':
  1. ((sp join s) where pn='P2')[snombre]
  2. ((sp where pn='P2') join s)[snombre]
  3. ((sp where pn='P2')[sn] join s)[snombre]

## Ejemplos básicos

5. Obtener los códigos de los proveedores que venden partes que no son 'P2':
  1.  $(sp \text{ where } pn \neq 'P2')[sn]$
6. Obtener los códigos de los proveedores que no venden la parte 'P2':
  1.  $s[sn] \text{ minus } (sp \text{ where } pn = 'P2')[sn]$
7. Obtener los nombres de los proveedores que no venden la parte 'P2':
  1.  $((s[sn] \text{ minus } (sp \text{ where } pn = 'P2')[sn]) \text{ join } s)[snombre]$
  2.  ~~$s[snombre] \text{ minus } ((sp \text{ where } pn = 'P2')[sn] \text{ join } s)[snombre]$~~
8. Obtener los códigos de los proveedores que venden todas las partes:
  1.  $sp[sn,pn] \text{ div } p[pn]$
9. Obtener los nombres de los proveedores que venden todas las partes:
  1.  $((sp[sn,pn] \text{ div } p[pn]) \text{ join } s)[snombre]$
  2.  ~~$(s \text{ join } sp)[snombre,pn] \text{ div } p[pn]$~~
  3.  $((s \text{ join } sp)[sn,snombre,pn] \text{ div } p[pn])[snombre]$

## Ejercicios básicos

1. Obtener los nombres de los proveedores que venden partes rojas
2. Obtener las ciudades dónde se venden partes rojas
3. Obtener todos los pares de códigos de proveedores de la misma ciudad
4. Obtener los nombres de los proveedores que no venden todas las partes
5. Obtener las ciudades donde se fabrican partes que no se venden