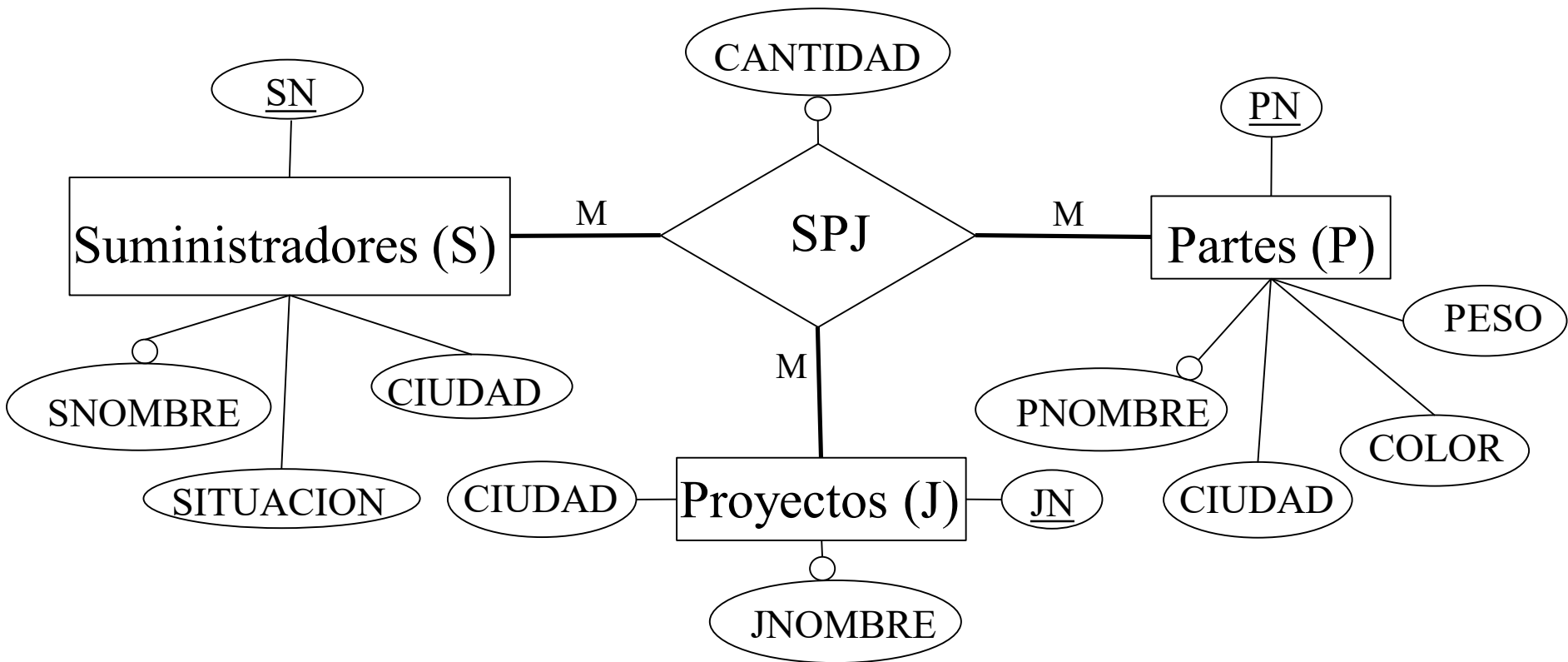


Tema 3.2 .-Álgebra relacional. (4 de 4)

Ejemplos ternarios

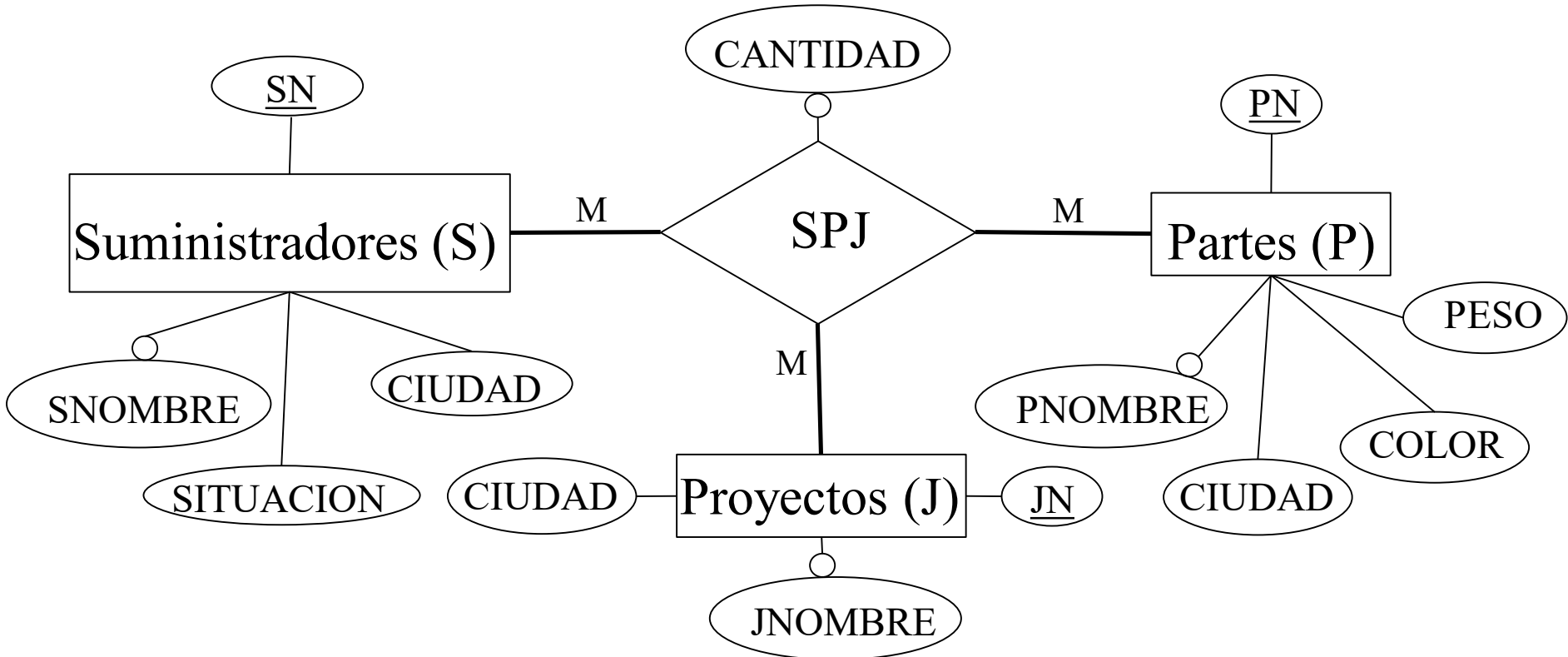
Esquema Entidad/Interrelación

- Representa una interrelación ternaria en una empresa, entre las **partes** usadas como materia prima, los **proveedores** que las suministran y los **proyectos** en los que se usan
- Una ocurrencia de **SPJ** une a un **suministrador**, una **parte** y un **proyecto** si dicho proveedor ha vendido a la empresa alguna vez esa parte para ser usada en ese proyecto
 - 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 para ser usado en ese proyecto
 - 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 dónde 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 dónde se fabrica.
- En la entidad **proyecto**:
 - La propiedad **jn** es un código identificador, **jnombre** es el nombre del proyecto, y **ciudad** indica la ciudad donde se monta el proyecto.



Esquema Relacional

❖ Si llevamos el esquema E/IR al modelo relacional se obtienen cuatro 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 j (jn char(3),
 jnombre char(25) NN,
 ciudad char(25))
 PK (jn);

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



GRUPO DE ESTRUCTURAS DE DATOS

Esquema Entidad/Interrelación

Una situación ejemplo de la BD sobre la que vamos a trabajar, sería:

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

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

JN	JNOMBRE	CIUDAD
J1	Lavadora	Telde
J2	Secadora	Agaete
J3	Nevera	Telde
J4	Lavadora	Arucas

SN	PN	JN	CANTIDAD
S1	P1	J1	100
S1	P1	J2	200
S1	P1	J3	100
S1	P1	J4	300
S1	P2	J1	200
S1	P4	J2	100
S1	P5	J4	200
S2	P1	J1	300
S2	P2	J1	200
S2	P2	J2	100
S2	P3	J2	200
S2	P3	J3	200
S2	P4	J3	200
S3	P1	J1	100
S3	P1	J2	300
S3	P1	J3	100
S3	P1	J4	200
S3	P5	J2	100

SN	PN	JN	CANTIDAD
S4	P1	J2	100
S4	P2	J2	200
S4	P3	J2	300
S4	P4	J2	100
S4	P5	J2	200
S5	P5	J1	300
S5	P5	J2	100
S5	P5	J3	200
S5	P5	J4	300
S6	P5	J1	100
S6	P5	J2	200
S6	P5	J3	300
S6	P5	J4	100
S6	P1	J3	100
S6	P2	J3	100
S6	P3	J3	100
S6	P4	J3	100
S6	P4	J4	100

Ejemplo 1

➤ Obtener todas las tripletas SN, PN, JN tales que el suministrador, la parte y el proyecto indicados estén cosituados en la misma ciudad.

$(S \text{ JOIN } P \text{ JOIN } J) [SN, PN, JN]$

1. Reunimos las tablas S, P y J.
2. Proyectamos la tabla anterior por los atributos buscados.

Ejemplo 2

➤ Obtener todas las tripletas SN, PN, JN tales que el suministrador, la parte y el proyecto indicados no estén todos cosituados.

(((S RENAME CIUDAD AS SCIUDAD)	TIMES
(P RENAME CIUDAD AS PCIUDAD)	TIMES
(J RENAME CIUDAD AS JCIUDAD)	
) WHERE SCIUDAD < > PCIUDAD OR SCIUDAD < > JCIUDAD	
) [SN, PN, JN]	

1. Renombramos el atributo CIUDAD como SCIUDAD en la tabla S.
2. Renombramos el atributo CIUDAD como PCIUDAD en la tabla P.
3. Renombramos el atributo CIUDAD como JCIUDAD en la tabla J.
4. Obtenemos el producto cartesiano de las tablas obtenidas en 1, 2 y 3.
5. Seleccionamos sólo aquellas en la que al menos un par de los antiguos atributos CIUDAD sean diferentes.
6. Proyectamos la tabla obtenida en 5 por los atributos que buscamos.

Ejemplo 3

➤ Obtener todas las tripletas SN, PN, JN tales que el suministrador, la parte y el proyecto indicados estén todos en diferente ciudad.

```
(( ( S RENAME CIUDAD AS SCIUDAD ) TIMES
  ( P RENAME CIUDAD AS PCIUDAD ) TIMES
  ( J RENAME CIUDAD AS JCIUDAD )
) WHERE SCIUDAD <> PCIUDAD
   AND SCIUDAD <> JCIUDAD
   AND PCIUDAD <> JCIUDAD
) [SN, PN, JN]
```

1. Renombramos el atributo CIUDAD como SCIUDAD en la tabla S.
2. Renombramos el atributo CIUDAD como PCIUDAD en la tabla P.
3. Renombramos el atributo CIUDAD como JCIUDAD en la tabla J.
4. Obtenemos el producto cartesiano de las tablas obtenidas en 1, 2 y 3.
5. Seleccionamos sólo aquellas en la que todos los antiguos atributos CIUDAD sean diferentes.
6. Proyectamos la tabla obtenida en 5 por los atributos que buscamos.

Ejemplo 4

➤ Obtener los códigos de las partes suministradas por algún suministrador de ‘Telde’.

(SPJ JOIN (S WHERE CIUDAD = ‘Telde’)) [PN]
--

1. Obtenemos todos los suministradores de ‘Telde’.
2. Reunimos las tablas SPJ y la obtenida en el paso 1.
3. Proyectamos la tabla anterior por el atributo buscado.

Ejemplo 5

➤ Obtener los códigos de las partes suministradas por algún suministrador de 'Telde' a un proyecto de 'Agaete'.

```
(( ( SPJ JOIN  
  ( S WHERE CIUDAD = 'Telde' )  
 ) [PN, JN]  
 ) JOIN  
 ( J WHERE CIUDAD = 'Agaete' )  
 ) [PN]
```

1. Seleccionamos los suministradores de 'Telde'.
2. Reunimos las tablas SPJ y la obtenida en el paso 1.
3. Proyectamos la tabla anterior en los atributos PN y JN.
4. Seleccionamos los proyectos de 'Agaete'.
5. Reunimos las tablas obtenidas en los pasos 3 y 4.
6. Proyectamos en el atributo buscado.

Ejemplo 6

➤ Obtener todas las parejas de ciudades tales que un proveedor de la primera ciudad suministre partes a un proyecto de la segunda ciudad.

```
( ( S RENAME CIUDAD AS SCIUDAD )  
  JOIN SPJ  
  JOIN (J RENAME CIUDAD AS JCIUDAD )  
) [SCIUDAD, JCIUDAD]
```

1. Renombramos el atributo CIUDAD como SCIUDAD en la tabla S.
2. Renombramos el atributo CIUDAD como JCIUDAD en la tabla J.
3. Reunimos las tablas SPJ y las obtenidas en los pasos 1 y 2.
4. Proyectamos en los atributos que buscamos.

Ejemplo 7

➤ Obtener los códigos de partes suministradas a un proyecto por un suministrador situado en la misma ciudad que el proyecto.

$(J \text{ JOIN } SPJ \text{ JOIN } S) [PN]$

1. Reunimos las tablas J, S y SPJ.
2. Proyectamos el atributo PN.

Ejemplo 8

➤ Obtener los códigos de los proyectos a los cuales suministra partes por lo menos un suministrador situado en una ciudad distinta.

```
(( ( J RENAME CIUDAD AS JCIUDAD )  
  JOIN SPJ  
  JOIN ( S RENAME CIUDAD AS SCIUDAD ) )  
 WHERE SCIUDAD < > JCIUDAD )  
[JN]
```

1. Renombramos el atributo CIUDAD como JCIUDAD en la tabla J.
2. Renombramos el atributo CIUDAD como SCIUDAD en la tabla S.
3. Reunimos las tablas SPJ y las obtenidas en los pasos 1 y 2.
4. Seleccionamos aquellas tuplas con diferentes valores para SCIUDAD y JCIUDAD.
5. Proyectamos por el atributo buscado.

Ejemplo 9

➤ Obtener todas las parejas de códigos de parte tales que algún suministrador suministre las dos partes indicadas.

```
(( ( SPJ[SN, PN] RENAME PN AS KPN )  
  JOIN  
  ( SPJ[SN, PN] RENAME PN AS YPN )  
 )  
 WHERE KPN < YPN  
 )  
 [KPN, YPN]
```

1. Proyectamos la tabla SPJ por los atributos SN y PN.
2. Renombramos el atributo PN como KPN en la tabla anterior
3. Proyectamos la tabla SPJ por los atributos SN y PN.
4. Renombramos el atributo PN como YPN en la tabla anterior
5. Reunimos las tablas resultantes de las puntos 2 y 4.
6. Seleccionamos las tuplas en las que $KPN < YPN$
7. Proyectamos en los atributos buscados.