Álgebra Relacional

①

Proveedor S (codpro , Nompro , status, ciudad)

Pieza P (Codpie , Nompie , color , Peso , ciudad)

Proyecto J (codpj , Nompj , ciudad)

ventas SPJ (Codpro , codpie , codpj , cantidad , Fecha)

a) Códigos de proveedores que suministran alguna pieza P1

$$\pi_{codpro} (\sigma_{codpie = P1} (ventas))$$

b) Suministros cuya cantidad supere las 100 unidades

$$\sigma_{cantidad > 100} (ventas)$$

c) Encontrar nombre de proveedores, piezas y proyectos que estén en la misma ciudad

$$\pi_{nompro, codpj, codpie} ((S \bowtie P) \bowtie J)$$

como el atributo en común es ciudad, se unen por esa columna

d) Encontrar las piezas suministradas por los proveedores de Londres

$$\pi_{p.nompie} (\sigma_{ciudad = Londres} (S \bowtie SPJ)) \bowtie_{codpie} P)$$

e) Encontrar parejas de ciudades tales que la primera sea de un proveedor y la segunda la de un proyecto entre los cuales haya algún suministro

$$\pi_{s.ciudad, j.ciudad} \left(S \bowtie_{codpro} \left(\left(\pi_{codpro, codpj} (SPJ) \right) \bowtie J \right) \right)$$

f) Encontrar los códigos de las piezas suministradas a algún proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto

$$\pi_{codpie} \left(\sigma_{s.ciudad = j.ciudad} (S \bowtie_{codpro} (SPJ \bowtie J)) \right)$$

g) Encontrar los códigos de los proyectos que tienen al menos un proveedor que no se encuentra en su ciudad.

$$\pi_{codpj} \left(\sigma_{s.ciudad \neq j.ciudad} \left(\left((SPJ \bowtie J) \bowtie_{codpro} S \right) \right) \right)$$

h) Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos

$$\pi_{ciudad}(P) \cup \pi_{ciudad}(J)$$

i) Mostrar todas las ciudades de los proveedores que no fabriquen piezas

$$\pi_{ciudad}(S) - \pi_{ciudad}(P)$$

j) Mostrar todas las ciudades de los proveedores en las que además se fabriquen piezas

$$\pi_{ciudad}(S) \cap \pi_{ciudad}(P)$$

k) Encontrar los códigos de los proyectos que usan una pieza que vende S1

$$\pi_{codpj}\left(\underbrace{SPJ \bowtie}_{②} \underbrace{\left(\pi_{codpie}\left(\sigma_{codpro=S1}(SPJ)\right)\right)}_{①}\right)$$

② Hacemos la reunión natural con SPJ, obteniendo las tuplas que tienen piezas que vende S1

① códigos de pieza que vende S1

l) Encontrar la cantidad más pequeña enviada en algún suministro

$\rho(SPJ) = \boxed{A = B}$ → Hacemos dos copias de ventas

$$\underbrace{\pi_{cantidad}(A)}_{} - \underbrace{\pi_{cantidad}\left(\sigma_{A.cantidad > B.cantidad}(A \times B)\right)}_{}$$

Tomamos las cantidades de la 1ª copia

Le quitamos las cantidades más grande recurrentemente. Llegará un momento en el que a A solo le quede un valor (que es el mínimo)

m) Encontrar los códigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres.

$$\underbrace{\pi_{codpj}(SPJ)}_{} - \pi_{codpj}\left(\underbrace{\sigma_{\substack{p.color = Rojo \\ s.ciudad = Londres}}\left(S \underset{codpro}{\bowtie} (SPJ \bowtie P)\right)}_{}\right)$$

② Se la quitas al total

① Buscas las piezas rojas suministradas por proveedores de Londres

n) Encontrar los códigos de los proyectos que tienen como único proveedor a S1.

$$\underbrace{\pi_{codpj}(SPJ)} - \underbrace{\pi_{codpj}(\sigma_{codpro \neq S1}(SPJ))}$$

se lo restamos al total ②

Buscamos los códigos de los proyectos que tienen proveedores que no sean s1 ①

ñ) Encontrar los códigos de las piezas que se suministran a todos los proyectos de París

$$\pi_{codpie}\left(\pi_{codpie, codpj}(SPJ) \div (\underbrace{\pi_{codpj}(\sigma_{ciudad=Paris}(J))})\right)$$

si no hiciéramos esto, también miraría cantidad y fecha

Obtenemos los códigos de los proyectos de paris ①

o) Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos

$$\pi_{codpro}(\underbrace{\pi_{codpro, codpie, codpj}(SPJ)} \div \underbrace{\pi_{codpj}(SPJ)})$$

② Mediante la división miramos que la dupla (codpro, codpie) se relaciona con todos los proyectos, es decir, que un mismo proveedor venda una misma pieza a todos los proyectos

① Tomamos código de los proyectos

p) Encontrar los códigos de los proyectos a los que S1 suministra todas las piezas existentes

$$\pi_{codpj}\left(\underbrace{\sigma_{codpro=S1}}(\underbrace{\pi_{codpro, codpie, codpj}}(SPJ) \div \underbrace{\pi_{codpie}(P)})\right)$$

③ solo nos interesa las de S1

② Mediante la división buscamos duplas (codpro, codpj) que estén relacionadas con todas las piezas existentes

① Tomamos todas las piezas existentes

q) Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos

$$\pi_{codpro}\left(\underline{\pi_{codpro,\,codpie,\,codpj}(SPJ)} \div \left(\underline{\pi_{codpie}(P) \times \pi_{codpj}(J)} \right) \right)$$

② Mediante la división miramos que proveedores suministran todas las piezas a todos los proyectos

Duplas de todas las piezas con todos los proyectos ①

②

L Lista_Boda (Ref# , Descripción , Precio) → Lista de todos los regalos
I Invitaciones (Nombre , Dirección , ciudad)
C confirman (Nombre , Numero)
R Reserva_Regalo (Nombre , Ref# , fecha) → Regalos Revisados

a) Encontrar los regalos (descripción) que no han sido revisados

$$\underline{\pi_{Descripción}} \left(L \bowtie \underline{\left(\pi_{Ref\#}(L) - \pi_{Ref\#}(R) \right)} \right)$$

② Lo resno para obtener su descripción

① Le quito a todos los regalos los que están revisados. Luego, nos quedan los que no están revisados

b) Encontrar la dirección de los invitados que confirman la asistencia de más de dos personas

$$\pi_{Dirección}\left(I \bowtie \pi_{Nombre}\left(\sigma_{Numero > 2}(C) \right) \right)$$

c) Encontrar el nombre y la referencia del regalo más caro ya reservado

$A = B = L \bowtie R$

$$\pi_{Ref\#,\,nombre}(A) - \pi_{A.Ref\#.\,A.nombre}\left(\sigma_{B.precio > A.precio}(A \times B) \right)$$

A = B

Ref#	Nombre	Precio
1	a	100
2	b	50

A.Ref#	A.Nombre	A.Precio	B.Ref#	B.Nombre	B.Precio
~~1~~	~~a~~	~~100~~	~~1~~	~~a~~	~~100~~
~~1~~	~~a~~	~~100~~	~~2~~	~~b~~	~~50~~
2	b	50	1	a	100
~~2~~	~~b~~	~~50~~	~~2~~	~~b~~	~~50~~

1 a
2 b − 2 b = 1 a

③

H Hombre (NomH, Edad)

M Mujeres (NomM, Edad)

HM HsimM (NomH, NomM) NomH cae simpático a NomM

MH MsimH (NomM, NomH) NomM cae simpático a NomH

C Matrim (NomH, NomM) casados

a) Hallar las parejas de hombres y mujeres que se caen mutuamente simpáticos, con edades entre 20 y 30 años y que no estén casados entre sí

$$A = \sigma_{20 \leq H.Edad \leq 30,\ 20 \leq M.Edad \leq 30} (H \times M)$$ → ① Parejas de hombres y mujeres entre 20 y 30 años

$$\pi_{NomH, NomM} (A \bowtie HM) \bowtie MH - \pi_{NomH, NomM} (C)$$

② Parejas que se caen mutuamente simpáticos

③ Le quitas las casadas

b) Hallar las mujeres casadas a los que no cae simpático su marido

$$\pi_{NomM} (C - (C \bowtie HM))$$

② Se lo quitas a las casadas y quedan los que no caen simpático

① casados donde el hombre cae simpático a la mujer

c) Hallar los hombres a los que no les cae simpática ninguna mujer

$$\pi_{NomH} (H) - \pi_{NomH} (MH)$$

d) Hallar las mujeres casadas a las que no les cae simpático ningún hombre casado

$$\pi_{NomM} (C) - \pi_{NomM} (\pi_{NomH} (C) \bowtie HM)$$

Nombre de las mujeres que les cae bien un hombre casado

4

Conductor (DNI , Nombre, Direc, Prov)
Vehículo (Matrícula , Carga_Max, Fecha_compra)
Ruta (Ruta# , ciudad_Sal , ciudad_Lleg , KM)
Viaje (Viaje# , DNI , Matrícula)
Prog-Viaje (Viaje# , Ruta# , Dia_Sem , Hora_Sal , Hora_Lleg)

a) Encontrar entre qué dos ciudades se realiza el viaje más largo

$\rho(\text{Ruta}) = A = B$

$$\pi_{\text{Ruta\#}, \text{ciudad\_sal}, \text{ciudad\_lleg}}(A) - \left(\pi_{B.\text{Ruta\#}, B.\text{ciudad\_sal}, B.\text{ciudad\_lleg}}\left(\sigma_{B.KM < A.KM}(A \times B)\right)\right)$$

A	B
100	100
10	10

A x B	
~~100~~	~~100~~
100	$\boxed{10}$
~~10~~	~~100~~
~~10~~	~~10~~

b) Listar los nombres de los conductores que hayan llevado todos los camiones de la empresa

$$\pi_{\text{Nombre}}\left(\text{conductor} \bowtie \left(\pi_{\text{DNI}, \text{Matrícula}}(\text{Viaje}) \div \pi_{\text{Matrícula}}(\text{Vehículo})\right)\right)$$

c) Encontrar qué días de la semana se hacen viajes entre Granada y Sevilla por la mañana (antes de las 13 h)

$$\text{Viajes} = \sigma_{\text{ciudad\_Sal} = \text{Granada}, \text{ciudad\_Lleg} = \text{Sevilla}}(\text{Ruta}) \cup \sigma_{\text{ciudad\_sal} = \text{Sevilla}, \text{ciudad\_lleg} = \text{Granada}}(\text{Ruta})$$

$$\pi_{\text{Dia\_sem}}\left(\sigma_{\text{Hora\_Sal} < 13}(\text{Prog-Viaje} \bowtie \text{Viajes})\right)$$

d) Encontrar las rutas que se hacen todos los días de la semana, suponiendo que hay rutas todos los días

$$\pi_{\text{Ruta\#}}\left(\pi_{\text{Ruta\#}, \text{Dia\_sem}}(\text{Prog-Viaje}) \div \pi_{\text{Dia\_sem}}(\text{Prog-Viaje})\right)$$

⑥

Representante (<u>DNI</u> , Nombre , Direc , Provincia)

Zona_Rep (<u>DNI , cod_zona</u> , Poblacion , Provincia)

Pedidos (<u>DNI , Cod_Art</u> , cantidad , poblacion)

Articulo (<u>Cod_Art</u> , Nombre , color , Prov_Fab)

a) Listar todas las provincias que son visitadas por todos los representantes

$$\pi_{provincia}\left(\pi_{DNI,\ provincia}(Zona\_Rep) \div \pi_{DNI}(Representante)\right)$$

b) Encontrar los representantes que venden fuera de su provincia artículos fabricados en su provincia

$$A = \sigma_{Representante.Provincia = Articulo.Prov\_Fab}(Representante \times Articulo)$$

$$B = Pedidos \bowtie_{poblacion} \pi_{poblacion,\ provincia}(Zona\_Rep)$$

$$\pi_{DNI}\left(\sigma_{B.provincia \,!= A.prov\_fab}\left(B \bowtie \pi_{DNI,\ cod\_Art,\ prov\_Fab}(A)\right)\right)$$

c) obtener las poblaciones de Granada que hayan superado los 50.000 euros de facturación y quién realizó el pedido

$$Pob\_Gr = \pi_{Poblacion}\left(\sigma_{provincia = Granada}(Zona\_Rep)\right)$$

$$\pi_{DNI,\ poblacion}\left(\sigma_{cantidad > 50.000}\left(Pedidos \bowtie Pob\_Gr\right)\right)$$

d) Mostrar las zonas que incluyen una sola población

$$\rho(Zona\_Rep) = A = B$$

$$\underbrace{\pi_{cod\_zona}(A)}_{②} - \pi_{A.cod\_zona}\left(\underbrace{\sigma_{\substack{A.cod\_zona = B.cod\_zona \\ \wedge \\ A.poblacion \neq B.poblacion}}(A \times B)}_{①}\right)$$

② se los quitas al total y te quedan los que solo tienen una población

① zonas con más de una población

e) Encontrar el código del artículo vendido en mayor cantidad

$$\rho(Pedidos) = A = B$$

$$\pi_{cod\_art}(A) - \left(\pi_{A.cod\_Art}\left(\sigma_{B.cantidad > A.cantidad}(A \times B)\right)\right)$$

A	B	A × B
100	100	~~100 100~~
10	10	~~100 10~~
		$\boxed{10}\ \boxed{100}$
		~~10 10~~

⑥

Inscripciones (INS # , Nombre, Procedencia)

Sesiones (SES # , Titulo, $\boxed{\text{Coord \#}}$) ⟶ son INS # renombrados

Articulos (ART # , Titulo)

Programa (SES # , ART # , Fecha, Hora-Inicio, Hora-Fin, sala, $\boxed{\text{Ponente \#}}$)

a) Mostrar los nombres de los ponentes que coordinan su propia sesión

$$\pi_{Nombre}\left(Inscripciones \bowtie \left(\sigma_{Sesiones.Coord\# = Programa.Ponente\#}(Sesiones \bowtie Programa)\right)\right)$$

b) Seleccionar los coordinadores que coordinan una única sesión

$\rho(sesiones) = A = B$

$$\pi_{coord\#}(A) - \underbrace{\pi_{A.cord\#}\left(\sigma_{A.SES\# \neq B.SES\# \ \wedge\ A.cord\# = B.cord\#}(A \times B)\right)}_{\text{coordinadores con más de una sesión}}$$

c) Mostrar el título de los artículos que se exponen en primer y último lugar

Supongamos que un artículo solo se expone una vez

$\rho(programa) = P1 = P2$

$$A = \pi_{art\#}(P1) - \pi_{P1.art\#}\left(\sigma_{P1.fecha > P2.fecha \ \vee\ (P1.fecha = P2.fecha \ \wedge\ P1.inicio > P2.inicio)}(P1 \times P2)\right)$$

$$B = \pi_{art\#}(P1) - \pi_{P1.art\#}\left(\sigma_{P1.fecha < P2.fecha \ \vee\ (P1.fecha = P2.fecha \ \wedge\ P1.inicio < P2.inicio)}(P1 \times P2)\right)$$

$$\pi_{Titulo}(articulos \bowtie (A \cup B))$$