

# Fundamentos de Bases de Datos



Los Del DGIIM, [losdelldgiim.github.io](https://losdelldgiim.github.io)

Doble Grado en Ingeniería Informática y Matemáticas  
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

# Fundamentos de Bases de Datos

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Arturo Olivares Martos

Granada, 2024-2025



# Índice general

<b>1. Seminarios</b>	<b>5</b>
1.1. Álgebra Relacional . . . . .	6



# 1. Seminarios

## 1.1. Álgebra Relacional

**Ejercicio 1.1.1.** Considere el esquema de base de datos relacional de la Figura 1.1.

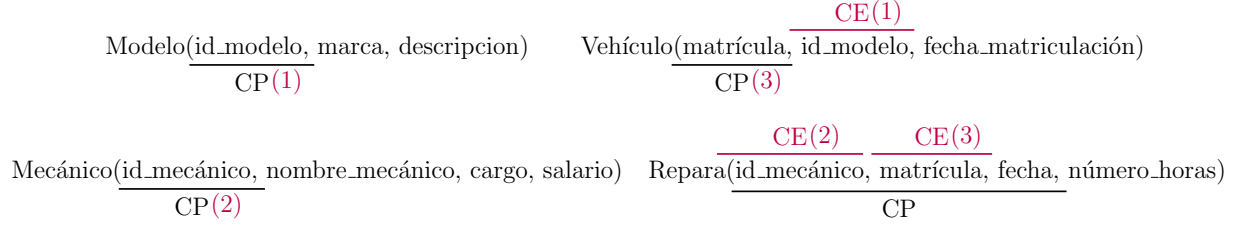


Figura 1.1: Esquema de base de datos relacional del Ejercicio 1.1.1.

Cada vehículo tiene asignado un modelo de una marca determinada. La tabla *Repara* registra reparaciones indicando qué mecánico repara qué vehículo en qué fecha y cuántas horas dura la reparación.

Indiqué qué consulta debe realizar en álgebra relacional para obtener la información solicitada en cada uno de los siguientes puntos:

1. Reparaciones de más de 20 horas.

$$\sigma_{\text{número\_horas} > 20}(\text{Repara})$$

2. Códigos de mecánicos que han reparado el vehículo de matrícula 1234ABC.

$$\pi_{\text{id\_mecánico}}(\sigma_{\text{matrícula}=1234ABC}(\text{Repara}))$$

3. Parejas de mecánicos que se pueden hacer en la empresa.

$$\begin{aligned} \rho(\text{Mecánico}) &= \text{Mecánico}_1, \text{Mecánico}_2 \\ \sigma_{\text{Mecánico}_1.\text{id\_mecánico} < \text{Mecánico}_2.\text{id\_mecánico}}(\text{Mecánico}_1 \times \text{Mecánico}_2) \end{aligned}$$

Notemos que, si tenemos dos mecánicos,  $A$  y  $B$ , debemos:

- Evitar las parejas de un mismo mecánico repetido, como  $A, A$  o  $B, B$ . Esto se consigue obligando a que los *id* sean distintos.
- Evitar parejas duplicadas, ya que  $A, B$  y  $B, A$  en realidad son distintas. Debido a que el orden en  $\mathbb{N}$  es un orden total<sup>1</sup>, esto se consigue obligando a que el código de uno de ellos (sea este el mecánico 1) sea menor que el del otro.

4. Marca de los vehículos matriculados después del 1/1/20.

$$\pi_{\text{marca}}(\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/20)}(\text{Vehículo}) \bowtie \text{Modelo})$$

<sup>1</sup>Suponemos que el dominio del código es  $\mathbb{N}$ . Si fuese alfanumérico, el orden alfabético también es total.



5. Parejas  $\langle \text{carga}, \text{marca} \rangle$  entre las que se ha dado alguna reparación.

$$\pi_{\text{carga}, \text{marca}} [\pi_{\text{id\_mecánico}, \text{carga}} (\text{Mecánico}) \bowtie \pi_{\text{id\_mecánico}, \text{matrícula}} (\text{Repara}) \bowtie \pi_{\text{matrícula}, \text{id\_modelo}} (\text{Vehículo}) \bowtie \pi_{\text{id\_modelo}, \text{marca}} (\text{Modelo})]$$

Notemos que esto sería equivalente a la siguiente consulta, solo que anteriormente hemos ido reduciendo el número de tuplas a manejar:

$$\pi_{\text{carga}, \text{marca}} (\text{Mecánico} \bowtie \text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})$$

6. Vehículos que o tienen una fecha de matriculación posterior al 1/1/22 o han sido reparados con posterioridad a esa misma fecha.

$$\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/22)} (\text{Vehículo}) \cup \pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}} (\text{Vehículo} \bowtie \sigma_{\text{fecha} > \text{fecha}(1/1/22)} (\text{Repara}))$$

7. Vehículos con fecha de matriculación posterior al 1/1/22 que han sido reparados alguna vez.

$$\pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}} (\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/22)} (\text{Vehículo}) \bowtie \text{Repara})$$

8. Marca de los vehículos que no han tenido ninguna reparación en el año 2022.

$$\pi_{\text{marca}} [(\text{Vehículo} - \pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}} (\text{Vehículo} \bowtie \sigma_{\text{fecha}(01/01/22) \leq \text{fecha} \leq \text{fecha}(31/12/22)} (\text{Repara}))) \bowtie \text{Modelo}]$$

9. Código de los mecánicos que han reparado vehículos de, al menos, dos marcas distintas.

Obtenemos en primer lugar todas las parejas de mecánicos con las marcas de los vehículos que han reparado; es decir:

$$\rho(\pi_{\text{id\_mecánico}, \text{marca}} (\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = A, B$$

La consulta sería:

$$\pi_{\text{id\_mecánico}} (\sigma_{A.\text{marca} \neq B.\text{marca}} (A \bowtie B))$$

10. Vehículos que tienen una sola reparación.

Buscamos en primer lugar los vehículos con más de una reparación. Para ello, tras hacer el producto cartesiano, exigimos que sea el mismo vehículo pero, o bien haya sido reparado en fechas distintas, o bien haya sido reparado por mecánicos distintos.

$$\begin{aligned} \rho(\text{Repara}) &= A \\ \rho(\text{Repara}) &= B \\ \rho \left[ \pi_{\text{matrícula}} \left( A \bowtie \begin{array}{c} A.\text{matrícula} = B.\text{matrícula} \\ (A.\text{fecha} > B.\text{fecha} \vee A.\text{id\_mecánico} > B.\text{id\_mecánico}) \end{array} \bowtie B \right) \right] &= C \end{aligned}$$

Notemos que tras proyectar habríamos llegado a la misma tabla estableciendo solo  $\neq$  en vez de  $>$ , pero usamos  $>$  ya que así obtenemos la mitad de las tuplas antes de proyectar.

Por tanto, en  $C$  tenemos las matrículas de los coches que tienen más de una reparación. Por tanto, las matrículas de los vehículos que tienen una sola reparación serían:

$$\pi_{\text{matrícula}}(\text{Repara}) - C$$

#### 11. Vehículos que han sufrido las reparaciones con la duración más alta.

Este caso es obtener un máximo, algo que es muy común en álgebra relacional. Obtener el máximo no es fácil, pero podemos obtener los que *no* cumplen la condición de ser el máximo.

$$\begin{aligned}\rho(\text{Repara}) &= A, B \\ \rho[\sigma_{A.\text{número\_horas} < B.\text{número\_horas}}(A \times B)] &= C \\ \rho(\pi_{A.\text{número\_horas}}(C)) &= D\end{aligned}$$

Por tanto, en  $D$  obtenemos los valores de *número\_horas* para los que hay otro valor que es mayor; es decir, los que no son el máximo. Por tanto, los que son el máximo son:

$$\pi_{\text{número\_horas}}(\text{Repara}) - D$$

Con esa consulta tan solo hemos obtenido el máximo del atributo *número\_horas*, pero ahora necesitamos obtener los vehículos que corresponden con esas reparaciones. Estos son:

$$\pi_{\text{matrícula}}(\text{Repara} \bowtie (\pi_{\text{número\_horas}}(\text{Repara}) - D))$$

#### 12. Mecánicos que tienen el salario más bajo.

En primer lugar, obtenemos los que no son candidatos a mínimo:

$$\begin{aligned}\rho(\text{Mecánico}) &= A, B \\ \rho[\sigma_{A.\text{salario} > B.\text{salario}}(A \times B)] &= C \\ \rho(\pi_{A.\text{salario}}(C)) &= D\end{aligned}$$

En  $D$ , obtenemos los salarios que no son el mínimo, ya que hay otro salario menor. Por tanto, el salario mínimo es:

$$\pi_{\text{salario}}(\text{Mecánico}) - D$$

Por tanto, los mecánicos que tienen el salario más bajo son:

$$\pi_{\text{id\_mecánico}}(\text{Mecánico} \bowtie (\pi_{\text{salario}}(\text{Mecánico}) - D))$$

13. Mecánicos cuyo salario es uno de los dos salarios más bajos.

Este apartado es similar al anterior, solo que hemos de repetir el procedimiento dos veces. Sea  $A$  la relación que contiene el resultado de la consulta anterior, es decir, los mecánicos con el salario más bajo. Consideramos ahora el resto de mecánicos:

$$\rho(\text{Mecánico} - (\text{Mecánico} \bowtie A)) = C, D$$

Por tanto, a esta nueva lista de mecánicos que no tienen el salario mínimo, le aplicamos el mismo procedimiento que en el apartado anterior para obtener los mecánicos con el segundo salario más bajo.

$$\begin{aligned} \rho[\sigma_{C.\text{salario} > D.\text{salario}}(C \times D)] &= E \\ \rho(\pi_{C.\text{salario}}(E)) &= F \end{aligned}$$

Por tanto, el segundo salario más bajo es:

$$\pi_{\text{salario}}(C) - F$$

Por tanto, los mecánicos cuyo salario es el segundo más bajo son:

$$\pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{salario}}(C) - F))$$

Por tanto, los mecánicos cuyo salario es uno de los dos salarios más bajos son:

$$A \cup \pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{salario}}(C) - F))$$

14. Vehículos que han sido reparados alguna vez por cada uno de los mecánicos.

Vemos que se trata de una consulta de división. Sean los siguientes alias:

$$\begin{aligned} \rho\left(\pi_{\substack{\text{matrícula} \\ \text{id\_mecánico}}}(\text{Repara})\right) &= R \\ \rho(\pi_{\text{id\_mecánico}}(\text{Mecánico})) &= S \end{aligned}$$

La consulta sería:

$$R \div S$$

Esta consulta devuelve una instancia  $w$  de la relación  $W[\text{matrícula}]$  que cumple que, para cada  $u \in w$ ,  $v \in s$  instancia de  $S$ , si  $r$  es una instancia de  $R$  se tiene:

$$\exists t \in r \mid t[\text{matrícula}] = u \wedge t[\text{id\_mecánico}] = v$$

Es decir, contiene las matrículas de los coches que han sido reparados por todos los mecánicos.

15. Mecánicos que han reparado vehículos de todas las marcas.

Obtenemos en primer lugar el par  $\langle \text{id\_mecánico}, \text{marca} \rangle$  de los vehículos que ha reparado cada mecánico.

$$\rho(\pi_{\text{id\_mecánico}, \text{marca}}(\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = R$$

El divisor sería la relación que contiene todas las marcas posibles.

$$\rho(\pi_{\text{marca}}(\text{Modelo})) = S$$

Por tanto, la consulta sería:

$$R \div S$$

16. Vehículos a los que el mecánico de id 123 les ha hecho todas las reparaciones.

De todas las reparaciones, no consideramos aquellas que han sido realizadas otro mecánico:

$$\pi_{\text{matrícula}}(\text{Repara}) - \pi_{\text{matrícula}}(\sigma_{\text{id\_mecánico} \neq 123}(\text{Repara}))$$

17. Marcas para las que todos sus vehículos han sido reparados alguna vez por un empleado con un salario superior a 30000.

De todas las marcas, no consideramos aquellas para las cuales hay un vehículo de esa marca que, o bien no ha sido reparado, o ha sido reparado por un mecánico con salario inferior a 30000. Obtenemos por tanto en primer lugar los vehículos que no han sido reparados y los que han sido reparados por mecánicos con salario inferior a 30000.

$$\begin{aligned} \rho(\pi_{\text{matrícula}}(\text{Vehículo}) - \pi_{\text{matrícula}}(\text{Repara})) &= A \\ \rho(\pi_{\text{matrícula}}(\text{Repara} \bowtie \sigma_{\text{salario} \leq 30000}(\text{Mecánico}))) &= B \end{aligned}$$

En  $A$  están los vehículos que no han sido reparados, y en  $B$  los que han sido reparados por mecánicos con salario inferior a 30000. Por tanto, las marcas que cumplen la condición son:

$$\pi_{\text{marca}}(\text{Modelo}) - \pi_{\text{marca}}(\text{Modelo} \bowtie \text{Vehículo} \bowtie (A \cup B))$$

*Observación.* En este caso, el enunciado puede llevar a pensar que se trata de una consulta de división, pero no es así porque no se trata de todos los vehículos, sino de todos los vehículos de una marca.

18. Vehículos que, para todos los cargos que hay en la empresa, han tenido al menos una reparación de más de 2 horas de duración con un empleado de ese cargo.

El dividendo  $R$  ha de ser los vehículos que han tenido al menos una reparación de más de 2 horas, con el cargo del mecánico que ha realizado la reparación.

$$\rho(\pi_{\text{matrícula}, \text{cargo}}(\sigma_{\text{número\_horas} > 2}(\text{Repara} \bowtie \text{Mecánico}))) = R$$

El divisor  $S$  sería la relación que contiene todos los cargos posibles.

$$\rho(\pi_{\text{cargo}}(\text{Mecánico})) = S$$

Por tanto, la consulta sería:

$$R \div S$$

19. Marcas para las que todos sus vehículos han sido reparados alguna vez por el mismo mecánico.

De nuevo, como no se trata de *todos los vehículos* sino de *todos los vehículos de una marca*, no se trata de una consulta de división.

20. Mecánico más joven que ha reparado vehículos de todas las marcas.

*Observación.* Para esto, supondremos que la tabla *Mecánico* tiene un atributo *fecha\_nacimiento*.

Obtengamos los mecánicos que han reparado vehículos de todas las marcas.

$$\rho(\pi_{\text{id\_mecánico,marca}}(\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = R$$

El divisor sería la relación que contiene todas las marcas posibles.

$$\rho(\pi_{\text{marca}}(\text{Modelo})) = S$$

Por tanto, los mecánicos que han reparado vehículos de todas las marcas son:

$$\rho(\pi_{\text{id\_mecánico}}(R \div S)) = C$$

Obtenemos ahora el mecánico más joven de entre estos. Para ello, obtenemos en primer lugar los mecánicos que no son candidatos a ser el más joven.

$$\rho(\text{Mecánico} \bowtie C) = A, B$$

$$\rho[\sigma_{A.\text{fecha\_nacimiento} < B.\text{fecha\_nacimiento}}(A \times B)] = D$$

$$\rho(\pi_{A.\text{fecha\_nacimiento}}(D)) = E$$

Por tanto, en  $E$  tenemos las fechas de nacimiento de los mecánicos de  $C$  que no son el más joven. Por tanto, la fecha de nacimiento del mecánico más joven es:

$$\pi_{\text{fecha\_nacimiento}}(C) - E$$

Por tanto, los mecánicos más jóvenes que han reparado vehículos de todas las marcas son:

$$\pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{fecha\_nacimiento}}(C) - E))$$

**Ejercicio 1.1.2.** Considere el esquema de base de datos relacional de la Figura 1.2.

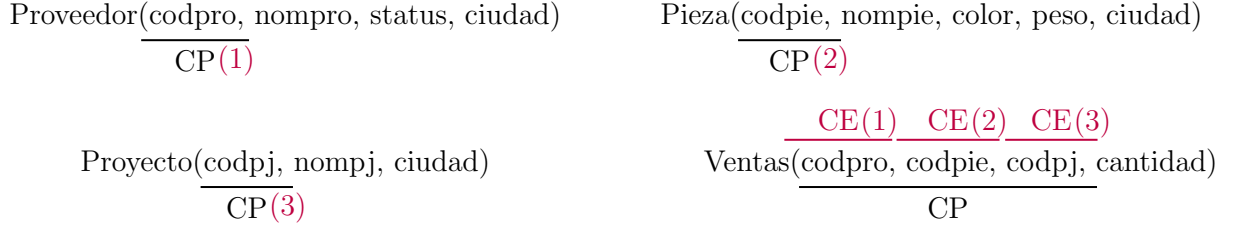


Figura 1.2: Esquema de base de datos relacional del Ejercicio 1.1.2.

Indique qué consulta debe realizar en álgebra relacional para obtener la información solicitada en cada uno de los siguientes puntos:

1. Encontrar todas las parejas de ciudades tales que la primera sea la de un proveedor y la segunda la de un proyecto entre los cuales haya algún suministro.

$$\pi_{\text{Proveedor.ciudad} \quad \text{Proyecto.ciudad}} \left[ \left( \pi_{\text{codpro}}(\text{Ventas}) \bowtie \text{Proveedor} \right) \bowtie_{\text{Ventas.codpj}=\text{Proyecto.codpj}} \text{Proyecto} \right]$$

Notemos que la segunda reunión realiza no puede ser natural, ya que ambas relaciones comparten el mismo nombre para un atributo *Ciudad*.

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

$$\pi_{\text{codpie}} [(\text{Ventas} \bowtie \text{Proveedor}) \bowtie \text{Proyecto}]$$

Notemos que la segunda impone que la ciudad sea la misma, ya que se realiza una reunión natural y ambos atributos comparten nombre.

3. Encontrar los códigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.

$$\pi_{\text{codpj}} \left[ (\text{Ventas} \bowtie \text{Proveedor}) \bowtie_{\substack{\text{Ventas.codpj}=\text{Proyecto.codpj} \\ \text{Proveedor.ciudad} \neq \text{Proyecto.ciudad}}} \text{Proyecto} \right]$$

4. Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos.

$$\pi_{\text{ciudad}}(\text{Pieza}) \cup \pi_{\text{ciudad}}(\text{Proyecto})$$

5. Mostrar todas las ciudades de los proveedores en las que no se fabriquen piezas.

$$\pi_{\text{ciudad}}(\text{Proveedor}) - \pi_{\text{ciudad}}(\text{Pieza})$$

6. Mostrar todas las ciudades de los proveedores en las que además se fabriquen piezas.

$$\pi_{\text{ciudad}}(\text{Proveedor}) \cap \pi_{\text{ciudad}}(\text{Pieza})$$

7. Encontrar los códigos de los proyectos que usan una pieza que vende S1.

$$\pi_{\text{codpj}} (\sigma_{\text{codpro}=\text{S1}}(\text{Ventas}))$$

8. Encontrar la cantidad más pequeña enviada en algún suministro.

Se trata de encontrar el mínimo del atributo *cantidad* de la relación *Ventas*.

$$\begin{aligned}\rho(\text{Ventas}) &= A, B \\ \rho[\sigma_{A.\text{cantidad} < B.\text{cantidad}}(A \times B)] &= C \\ \rho(\pi_{B.\text{cantidad}}(C)) &= D\end{aligned}$$

Por tanto, la cantidad más pequeña enviada en algún suministro es:

$$\pi_{\text{cantidad}}(\text{Ventas}) - D$$

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

$$\begin{aligned}\rho[\pi_{\text{codpj}}((\text{Ventas} \bowtie \sigma_{\text{ciudad}=\text{Londres}}(\text{Proveedor})) \\ \bowtie_{\text{Ventas.codpie}=\text{Pieza.codpie}} \sigma_{\text{Pieza.color}=\text{rojo}}(\text{Pieza}))] &= A \\ \pi_{\text{codpj}}(\text{Proyecto}) - A\end{aligned}$$

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

$$\pi_{\text{codpj}}(\text{Ventas}) - \pi_{\text{codpj}}(\sigma_{\text{codpro} \neq \text{S1}}(\text{Ventas}))$$

11. Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.

Se trata de una división. El dividendo es:

$$\rho \left[ \pi_{\text{codpie}}^{\text{codpj}}(\text{Ventas}) \right] = A$$

El divisor es:

$$\rho[\pi_{\text{codpj}}(\sigma_{\text{ciudad}=\text{París}}(\text{Proyecto}))] = B$$

Por tanto, la consulta sería:

$$A \div B$$

12. Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.

El dividendo es:

$$\rho \left[ \pi_{\text{codpro}}^{\text{codpie codpj}}(\text{Ventas}) \right] = A$$

El divisor es:

$$\rho[\pi_{\text{codpj}}(\text{Proyecto})] = B$$

La consulta es:

$$\pi_{\text{codpro}}(A \div B)$$

13. Encontrar los códigos de los proyectos a los que el proveedor S1 suministra todas las piezas existentes.

EL dividendo es:

$$\rho \left[ \pi_{\substack{\text{codpj} \\ \text{codpie}}} (\sigma_{\text{codpro}=\text{S1}}(\text{Ventas})) \right] = A$$

El divisor es:

$$\rho [\pi_{\text{codpie}}(\text{Pieza})] = B$$

La consulta es:

$$\pi_{\text{codpj}}(A \div B)$$

14. Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.

Podríamos pensar en hacer dos divisiones. Veamos este proceso:

$$\begin{aligned} \rho \left[ \pi_{\substack{\text{codpro} \\ \text{codpie} \\ \text{codpj}}} (\text{Ventas}) \right] &= A \\ \rho [\pi_{\text{codpie}}(\text{Pieza})] &= B \\ \rho [\pi_{\text{codpj}}(\text{Proyecto})] &= C \end{aligned}$$

Podríamos pensar entonces que la consulta sería:

$$\pi_{\text{codpro}}((A \div B) \div C)$$

No obstante, aquí obtenemos los códigos que proporcionan todas las piezas y que proporcionan a todos los proyectos, pero no necesariamente todas las piezas a todos los proyectos. Para esto último, hemos de considerar todas las parejas posibles de piezas y proyectos, y obtener los códigos de los proveedores que suministran todas las parejas posibles.

$$\begin{aligned} \rho \left[ \pi_{\substack{\text{codpro} \\ \text{codpie} \\ \text{codpj}}} (\text{Ventas}) \right] &= A \\ \rho [\pi_{\text{codpie}}(\text{Pieza}) \times \pi_{\text{codpj}}(\text{Proyecto})] &= B \end{aligned}$$

La consulta por tanto queda:

$$A \div B$$

15. Pieza con más peso entre las que pesan menos de 100.

Partimos de las piezas que pesan menos de 100:

$$\rho [\sigma_{\text{peso}<100}(\text{Pieza})] = A, B$$

Ahora, obtenemos los que no son candidatos a máximo de entre estas piezas:

$$\begin{aligned} \rho [\sigma_{A.\text{peso}<B.\text{peso}}(A \times B)] &= C \\ \rho [\pi_{A.\text{peso}}(C)] &= D \end{aligned}$$



Por tanto, el mayor peso menor de 100 es:

$$\rho[\pi_{\text{peso}}(A) - D] = E$$

Por tanto, la pieza con más peso entre las que pesan menos de 100 es:

$$\pi_{\text{codpie}}[A \bowtie E]$$

16. Entre los proyectos de Jaén, mostrar el que ha suministrado la pieza de mayor peso (puede haber más de uno).
17. Proyectos para los que la lista de piezas que han suministrado tiene al menos dos piezas distintas.
18. Proyectos para los que la lista de piezas que han suministrado tiene exactamente dos piezas distintas.
19. Proveedores que han hecho una o dos ventas (y no más).
20. Proveedores en los que todos sus suministros son de una pieza roja o de una pieza de Granada.

$$\pi_{\text{codpro}}(\text{Ventas}) - \pi_{\text{codpro}}(\text{Ventas} \bowtie \sigma_{\text{color} \neq \text{rojo} \wedge \text{ciudad} \neq \text{Granada}}(\text{Piezas}))$$