

# 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.

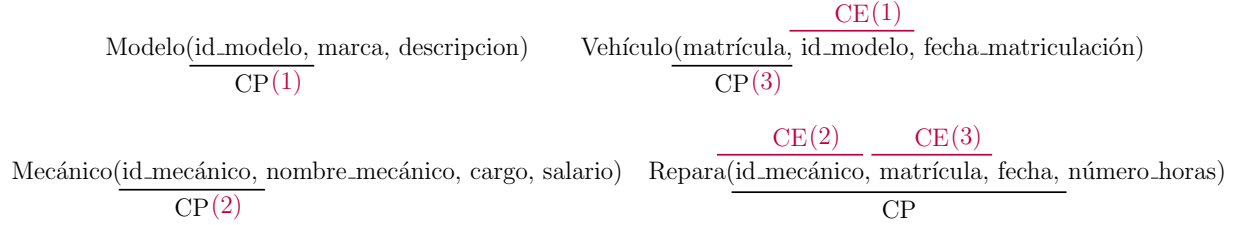


Figura 1.1: Esquema de base de datos relacional del Ejercicio 1.1.1.

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.

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

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.

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

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.

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

<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.

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

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

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

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

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

8. Marca de los vehículos que no han tenido ninguna reparación en el año 2022.
9. Código de los mecánicos que han reparado vehículos de, al menos, dos marcas distintas.
10. Vehículos que tienen una sola reparación.
11. Vehículos que han sufrido las reparaciones con la duración más alta.
12. Mecánicos que tienen el salario más bajo.
13. Mecánicos cuyo salario es uno de los dos salarios más bajos.
14. Vehículos que han sido reparados alguna vez por cada uno de los mecánicos.
15. Mecánicos que han reparado vehículos de todas las marcas.
16. Vehículos a los que el mecánico de id 123 les ha hecho todas las reparaciones.
17. Marcas para las que todos sus vehículos han sido reparados alguna vez por un empleado con un salario superior a 30000.
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.
19. Marcas para las que todos sus vehículos han sido reparados alguna vez por el mismo mecánico.
20. Mecánico más joven que ha reparado vehículos de todas las marcas.

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

|  |   |
|--|---|
| Proveedor( <u>codpro, nompro, status, ciudad</u> ) | Pieza( <u>codpie, nompie, color, peso, ciudad</u> ) |
| CP(1)  | CP(2)   |
|  | CE(1) CE(2) CE(3)                                   |
| Proyecto( <u>codpj, nompj, ciudad</u> )            | Ventas( <u>codpro, codpie, codpj, cantidad</u> )    |
| CP(3)  | CP  |

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.
2. Encontrar los códigos de las piezas suministradas a algún proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto.
3. Encontrar los códigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.
4. Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos.
5. Mostrar todas las ciudades de los proveedores en las que no se fabriquen piezas.
6. Mostrar todas las ciudades de los proveedores en las que además se fabriquen piezas.
7. Encontrar los códigos de los proyectos que usan una pieza que vende S1.
8. Encontrar la cantidad más pequeña enviada en algún suministro.
9. Encontrar los códigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres.
10. Encontrar los códigos de los proyectos que tienen como único proveedor a S1.
11. Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.
12. Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.
13. Encontrar los códigos de los proyectos a los que el proveedor S1 suministra todas las piezas existentes.
14. Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.

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