
Fundamentos de Base de Datos

Ejercicios Seminario 4: Algebra Relacional

Ismael Sallami Moreno

Mayo de 2025

Índice general

1	Relación de Ejercicios	5
1.1	Ejercicio 1	5
1.2	Ejercicio 2	7



Ejercicios de álgebra relacional

1. Sobre las siguientes relaciones:

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)

Realizar las siguientes consultas expresándolas en álgebra relacional:

- Encontrar los códigos de los proveedores que suministran alguna pieza al proyecto J1.
- Encontrar los suministros cuya cantidad supere las 100 unidades.
- Encontrar los nombres de proveedores, piezas y proyectos que se encuentren en la misma ciudad.
- Encontrar los nombres de las piezas suministradas por los proveedores de Londres.
- 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.
- Encontrar los códigos de las piezas suministradas a algún proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto.
- Encontrar los códigos de los proyectos que tienen algún proveedor que no se encuentre en su misma ciudad.
- Mostrar las ciudades donde se fabrican piezas y hay proyectos.
- Mostrar las ciudades de los proveedores en las que no se fabriquen piezas.
- Encontrar los códigos de los proyectos que usan alguna pieza de las que vende S1.
- Encontrar la cantidad más pequeña enviada en algún suministro.
- Encontrar los códigos de los proyectos que no utilizan piezas rojas suministradas por proveedores de Londres.
- Encontrar los códigos de los proyectos que tienen como único proveedor a S1.
- Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.
- Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.
- Encontrar los códigos de los proyectos a los que S1 suministra todas las piezas existentes.
- Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.



2. Sobre el siguiente esquema:

LISTA_BODA (REF#, DESCRIPCION, PRECIO)
INVITACIONES (NOMBRE, DIRECCION, CIUDAD)
CONFIRMAN (NOMBRE, NUMERO)
RESERVA_REGALO (NOMBRE, REF#, FECHA)

Realizar las siguientes consultas expresándolas en álgebra relacional:

- a) Encontrar la descripción de los regalos que no han sido reservados.
- b) Encontrar la dirección de los invitados que confirman la asistencia de más de dos personas.
- c) Encontrar el nombre y la referencia del regalo más caro ya reservado.

Anotaciones sobre la notación usada:

- $\pi_{attributes}(Relation)$: Projection
- $\sigma_{condition}(Relation)$: Selection
- $R \bowtie_{condition} S$: Theta Join
- $R \bowtie S$: Natural Join (joins on common attribute names)
- $R \cup S$: Union
- $R - S$: Difference
- $R \times S$: Cartesian Product
- $\rho_{NewName}(Relation)$ or $\rho_{NewAttr \leftarrow OldAttr}(Relation)$: Rename
- $R \div S$: Division
- For min/max operations without explicit aggregate functions, we use a common pattern:
find all items, then subtract those items for which a smaller/larger one exists.

1 Relación de Ejercicios

1.1. Ejercicio 1

1. Sobre las siguientes relaciones:

- **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) **Encontrar los códigos de los proveedores que suministran alguna pieza al proyecto J1.**

$$\pi_{CODPRO}(\sigma_{CODPJ='J1'}(SPJ))$$

b) **Encontrar los suministros cuya cantidad supere las 100 unidades.**

$$\sigma_{CANTIDAD>100}(SPJ)$$

c) **Encontrar los nombres de proveedores, piezas y proyectos que se encuentren en la misma ciudad.** (Esto implica que el proveedor, la pieza y el proyecto están en la misma ciudad y existe una venta que los relaciona)

$$\pi_{S.NOMPRO,P.NOMPIE,J.NOMPJ}(\sigma_{S.CIUDAD=P.CIUDAD \wedge P.CIUDAD=J.CIUDAD}$$

$$(S \bowtie_{S.CODPRO=SPJ.CODPRO} SPJ \bowtie_{SPJ.CODPIE=P.CODPIE} P \bowtie_{SPJ.CODPJ=J.CODPJ} J))$$

d) **Encontrar los nombres de las piezas suministradas por los proveedores de Londres.**

$$\pi_{P.NOMPIE}(P \bowtie_{P.CODPIE=SPJ.CODPIE} (SPJ \bowtie_{SPJ.CODPRO=S.CODPRO} (\sigma_{S.CIUDAD='Londres'}(S))))$$

e) **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.**

$$\pi_{S.CIUDAD,J.CIUDAD}(S \bowtie_{S.CODPRO=SPJ.CODPRO} SPJ \bowtie_{SPJ.CODPJ=J.CODPJ} J)$$

f) **Encontrar los códigos de las piezas suministradas a algún proyecto por un**

proveedor que se encuentre en la misma ciudad que el proyecto.

$$\pi_{SPJ.CODPIE}((\sigma_{S.CIUDAD=J.CIUDAD}(S \bowtie J)) \bowtie_{S.CODPRO=SPJ.CODPRO \wedge J.CODPJ=SPJ.CODPJ} SPJ)$$

O más detalladamente:

$$R_1 \leftarrow S \bowtie_{S.CODPRO=SPJ.CODPRO} SPJ$$

$$R_2 \leftarrow J \bowtie_{J.CODPJ=R_1.CODPJ} R_1$$

$$\pi_{CODPIE}(\sigma_{S.CIUDAD=J.CIUDAD}(R_2))$$

- g) **Encontrar los códigos de los proyectos que tienen algún proveedor que no se encuentre en su misma ciudad.**

$$\pi_{J.CODPJ}(\sigma_{S.CIUDAD \neq J.CIUDAD}(S \bowtie_{S.CODPRO=SPJ.CODPRO} SPJ \bowtie_{SPJ.CODPJ=J.CODPJ} J))$$

- h) **Mostrar las ciudades donde se fabrican piezas y hay proyectos.**

$$(\pi_{CIUDAD}(P)) \cap (\pi_{CIUDAD}(J))$$

- i) **Mostrar las ciudades de los proveedores en las que no se fabriquen piezas.**

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

- j) **Encontrar los códigos de los proyectos que usan alguna pieza de las que vende S1.**

$$\pi_{CODPJ}(SPJ \bowtie (\pi_{CODPIE}(\sigma_{CODPRO='S1'}(SPJ))))$$

- k) **Encontrar la cantidad más pequeña enviada en algún suministro.** (Cantidades en SPJ menos aquellas cantidades para las que existe una cantidad estrictamente menor)

$$SPJ_1 \leftarrow \rho_{SPJ_1}(SPJ)$$

$$SPJ_2 \leftarrow \rho_{SPJ_2}(SPJ)$$

$$CantidadesNoMinimas \leftarrow \pi_{SPJ_1.CANTIDAD}(SPJ_1 \bowtie_{SPJ_1.CANTIDAD > SPJ_2.CANTIDAD} SPJ_2)$$

$$Resultado \leftarrow \pi_{CANTIDAD}(SPJ) - CantidadesNoMinimas$$

- l) **Encontrar los códigos de los proyectos que no utilizan piezas rojas suministradas por proveedores de Londres.**

$$ProyectosConPiezasRojasDeLondres \leftarrow \pi_{SPJ.CODPJ}((\sigma_{S.CIUDAD='Londres'}(S)))$$

$$\bowtie_{S.CODPRO=SPJ.CODPRO} SPJ \bowtie_{SPJ.CODPIE=P.CODPIE} (\sigma_{P.COLOR='Roja'}(P))$$

$$\text{Resultado} \leftarrow \pi_{\text{CODPJ}}(J) - \text{ProyectosConPiezasRojasDeLondres}$$

- m) **Encontrar los códigos de los proyectos que tienen como único proveedor a S1.**

$$\text{ProyectosSuministradosPorS1} \leftarrow \pi_{\text{CODPJ}}(\sigma_{\text{CODPRO}='S1'}(SPJ))$$

$$\text{ProyectosSuministradosPorOtros} \leftarrow \pi_{\text{CODPJ}}(\sigma_{\text{CODPRO} \neq 'S1'}(SPJ))$$

$$\text{Resultado} \leftarrow \text{ProyectosSuministradosPorS1} - \text{ProyectosSuministradosPorOtros}$$

- n) **Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.**

$$\text{ProyectosParis} \leftarrow \pi_{\text{CODPJ}}(\sigma_{\text{CIUDAD}='Paris'}(J))$$

$$\text{PiezasSuministradasAProyectos} \leftarrow \pi_{\text{CODPIE,CODPJ}}(SPJ)$$

$$\text{Resultado} \leftarrow \text{PiezasSuministradasAProyectos} \div \text{ProyectosParis}$$

- ñ) **Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.** (Proveedores 's' tales que existe una pieza 'p' que 's' suministra a todos los proyectos 'j')

$$\text{SuministrosCompletosPorPieza} \leftarrow (\pi_{\text{CODPRO,CODPIE,CODPJ}}(SPJ)) \div (\pi_{\text{CODPJ}}(J))$$

$$\text{Resultado} \leftarrow \pi_{\text{CODPRO}}(\text{SuministrosCompletosPorPieza})$$

- o) **Encontrar los códigos de los proyectos a los que S1 suministra todas las piezas existentes.**

$$\text{TodasLasPiezas} \leftarrow \pi_{\text{CODPIE}}(P)$$

$$\text{SuministrosS1} \leftarrow \pi_{\text{CODPJ,CODPIE}}(\sigma_{\text{CODPRO}='S1'}(SPJ))$$

$$\text{Resultado} \leftarrow \text{SuministrosS1} \div \text{TodasLasPiezas}$$

- p) **Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.** (Proveedores 's' que para toda pieza 'p' y para todo proyecto 'j', existe la tupla (s,p,j) en SPJ)

$$\text{TodasCombinacionesPiezaProyecto} \leftarrow \pi_{\text{CODPIE}}(P) \times \pi_{\text{CODPJ}}(J)$$

$$\text{Resultado} \leftarrow (\pi_{\text{CODPRO,CODPIE,CODPJ}}(SPJ)) \div \text{TodasCombinacionesPiezaProyecto}$$

1.2. Ejercicio 2

2. Sobre el siguiente esquema:

- **LISTA_BODA** (REF#, DESCRIPCION, PRECIO)

- **INVITACIONES** (NOMBRE, DIRECCION, CIUDAD)
- **CONFIRMAN** (NOMBRE, NUMERO)
- **RESERVA_REGALO** (NOMBRE, REF#, FECHA)

a) **Encontrar la descripción de los regalos que no han sido reservados.**

$$ReferenciasReservadas \leftarrow \pi_{REF\#}(RESERVA_{REGALO})$$

$$ReferenciasNoReservadas \leftarrow \pi_{REF\#}(LISTA_{BODA}) - ReferenciasReservadas$$

$$Resultado \leftarrow \pi_{DESCRIPCION}(LISTA_{BODA} \bowtie ReferenciasNoReservadas)$$

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

$$InvitadosConMasDeDos \leftarrow \sigma_{NUMERO > 2}(CONFIRMAN)$$

$$Resultado \leftarrow \pi_{DIRECCION}(INVITACIONES \bowtie InvitadosConMasDeDos)$$

c) **Encontrar el nombre y la referencia del regalo más caro ya reservado.** (Regalos reservados menos aquellos regalos reservados para los cuales existe otro regalo reservado estrictamente más caro)

$$RegalosReservadosConPrecio \leftarrow LISTA_{BODA} \bowtie RESERVA_{REGALO}$$

$$R1 \leftarrow \rho_{R1}(RegalosReservadosConPrecio)$$

$$R2 \leftarrow \rho_{R2}(RegalosReservadosConPrecio)$$

$$RegalosQueNoSonLosMasCaros \leftarrow \pi_{R1.NOMBRE, R1.REF\#}(R1 \bowtie_{R1.PRECIO < R2.PRECIO} R2)$$

$$TodosLosNombresYRefReservados \leftarrow \pi_{NOMBRE, REF\#}(RegalosReservadosConPrecio)$$

$$Resultado \leftarrow TodosLosNombresYRefReservados - RegalosQueNoSonLosMasCaros$$