

Práctica 3 - Álgebra Relacional

Resolver los siguientes ejercicios aplicando las operaciones del álgebra relacional vistas en la materia:

- Selección σ
- Proyección π
- Producto Cartesiano \times
- Producto Natural \bowtie
- Diferencia o Resta $-$
- Intersección \cap
- Unión \cup
- División \div
- Renombre ρ

PARTE I

1) Para cada inciso, indique la opción correcta.

A) Indique cuál/es de las siguientes operaciones son válidas:

☒ $A(a, b, c) \cup B(a, b, d) \rightarrow$ es válida, siempre y cuando el dominio de c y d sea el mismo.

☒ $(A(a, b, c) \bowtie B(a, b)) - C(a, b, c) \rightarrow$ es válida, en el producto natural me quedo con los campos a, b, c, lo que me permite hacer la diferencia (unión compatible)

☒ $(A(a, b, c) \bowtie B(a, d, e)) \cap D(a, b, c, d, e) \rightarrow$ es válida, en el producto natural me quedo con los campos a, b, c, d, e, lo que me permite hacer la intersección (unión compatible)

☐ $(A(a, b, c) \times B(a, b, d)) \cap D(a, b, c, d) \rightarrow$ no es válida, el producto cartesiano entre A y B tiene 6 atributos, lo que no me permite hacer la intersección (no son de unión compatible)

B) Para la operación de resta es necesario que los esquemas involucrados sean compatibles, es decir, deben cumplir las siguientes condiciones:

☒ Deben tener la misma cantidad de columnas

☒ Las columnas deben ser del mismo dominio

☒ El orden de las columnas debe ser el mismo

☐ Las columnas deben tener igual nombre

2) ¿Para cuáles de las siguientes operaciones es necesario que los operandos sean unión compatibles? Marque todas las opciones correctas:

☒ resta $-$

☐ división \div

☒ unión \cup

☐ producto cartesiano \times

☐ producto natural \bowtie

3) Dados los siguientes esquemas

COMPRA(#compra, fecha, monto_total)
 COMPRA_PRODUCTO(#compra, cantidad, #producto)
 PRODUCTO(#producto, nombre, precio)

Indique qué formato (conjunto de atributos) tiene el resultado de aplicar la siguiente operación.

consultar

Π #producto COMPRA_PRODUCTO % Π #producto PRODUCTO

[X] (#compra, cantidad)
 [] (#compra, cantidad, #producto)
 [] (#compra)

Para que la consulta tenga más sentido:

Π #compra,#producto COMPRA_PRODUCTO % Π #producto PRODUCTO

Entonces no me devuelve cantidad

4) Dado el siguiente esquema:

PASAJERO (#pasajero, nombre, dni, puntaje)
 PASAJERO_RESERVA (#pasajero, #reserva)
 RESERVA (#reserva, #vuelo, fecha_reserva, monto, #asiento)
 VUELO (#vuelo, aeropuerto_salida, aeropuerto_destino, fecha_vuelo)

Indicar si las siguientes consultas obtienen el resultado correcto (sin tener en cuenta la optimización).

A) Obtener los pasajeros que tengan reservas sobre vuelos del próximo año, listando #pasajero, #vuelo y #asiento.

VUELOS_PROX_AÑO $\leftarrow \sigma$ fecha_vuelo \geq 1/1/2026 AND fecha_vuelo \leq 31/12/2026 VUELO

RES $\leftarrow \Pi$ #pasajero,#vuelo,#asiento (VUELOS_PROX_AÑO [X] RESERVA [X]
 PASAJERO_RESERVA)

Consulta correcta.

B) Obtener el listado de montos de reservas realizadas para vuelos efectuados el pasado Agosto desde Buenos Aires a Córdoba.

PASAJERO (#pasajero, nombre, dni, puntaje)
 PASAJERO_RESERVA (#pasajero, #reserva)
 RESERVA (#reserva, #vuelo, fecha_reserva, monto, #asiento)
 VUELO (#vuelo, aeropuerto_salida, aeropuerto_destino, fecha_vuelo)

VUELOS_BUE_CBA $\leftarrow \sigma_{\text{ciudad_salida}=\text{"Buenos Aires"} \text{ AND } \text{ciudad_destino}=\text{"Córdoba"}} \text{VUELO}$

RESERV_AGO $\leftarrow (\sigma_{\text{fecha_reserva} \geq 1/8/2025 \text{ AND } \text{fecha_reserva} \leq 31/8/2025} \text{RESERVA}) \mid X \mid$
 VUELOS_BUE_CBA

RES $\leftarrow \pi_{\text{monto}} \text{RESERV_AGO}$

Consulta correcta.

C) Obtener el/los pasajeros que solo hayan reservado vuelos cuyo aeropuerto de salida sea el aeropuerto "Ministro Pistarini". Listar el nombre y dni de los pasajeros.

PASAJERO (#pasajero, nombre, dni, puntaje)
 PASAJERO_RESERVA (#pasajero, #reserva)
 RESERVA (#reserva, #vuelo, fecha_reserva, monto, #asiento)
 VUELO (#vuelo, aeropuerto_salida, aeropuerto_destino, fecha_vuelo)

VUELOS_PISTARINI $\leftarrow \pi_{\#vuelo} (\sigma_{\text{aeropuerto_salida}=\text{"Ministro Pistarini"}} \text{VUELO})$

RESERVA_PISTARINI $\leftarrow \pi_{\#pasajero} (\text{VUELOS_PISTARINI} \mid X \mid \text{RESERVA})$

PASAJEROS_PISTARINI $\leftarrow \pi_{\text{nombre, dni}} (\text{RESERVA_PISTARINI} \mid X \mid \text{PASAJERO})$

Consulta incorrecta, hay que hacer una diferencia entre los pasajeros que volaron desde Pistarini y los que volaron desde otros aeropuertos. Además en RESERVA_PISTARINI no se encuentra el atributo #pasajero. Consulta correcta:

VUELOS_PISTARINI $\leftarrow \sigma_{\text{aeropuerto_salida} = \text{"Ministro Pistarini"}} (\text{VUELO})$

VUELOS_OTROS $\leftarrow \sigma_{\text{aeropuerto_salida} \neq \text{"Ministro Pistarini"}} (\text{VUELO})$

PASAJ_PISTARINI $\leftarrow \pi_{\#pasajero} (\text{VUELOS_PISTARINI} \mid X \mid \text{RESERVA} \mid X \mid$
 PASAJERO_RESERVA)

PASAJ_OTROS $\leftarrow \pi_{\#pasajero} (\text{VUELOS_OTROS} \mid X \mid \text{RESERVA} \mid X \mid \text{PASAJERO_RESERVA})$

SOLO_PISTARINI $\leftarrow \text{PASAJ_PISTARINI} - \text{PASAJ_OTROS}$

RES $\leftarrow \pi_{\text{nombre, dni}} (\text{SOLO_PISTARINI} \mid X \mid \text{PASAJERO})$

D) Obtener el/los id/s de los pasajeros que hayan realizado reservas por un monto superior a \$99000

PASAJERO (#pasajero, nombre, dni, puntaje)

PASAJERO_RESERVA (#pasajero, #reserva)

RESERVA (#reserva, #vuelo, fecha_reserva, monto, #asiento)

VUELO (#vuelo, aeropuerto_salida, aeropuerto_destino, fecha_vuelo)

RESERVAS_MAS_99000 <- Π #pasajero (σ monto < 99000 RESERVA)

Consulta Incorrecta. Debo hacer un producto natural con PASAJERO RESERVA para obtener el id de pasajero. Ademas debo consultar por monto mayor a 99000. Consulta correcta:

RESERVAS_MAS_99000 <- Π #pasajero((σ monto > 99000 RESERVA) \times PASAJERO_RESERVA)

Parte II: Para cada uno de los esquemas dados, resolver las consultas pedidas:

6) Choferes

DUEÑO (id_dueño, nombre, teléfono, dirección, dni)

CHOFER (id_chofer, nombre, teléfono, dirección, fecha_licencia_desde, fecha_licencia_hasta, dni)

AUTO (patente, id_dueño, id_chofer, marca, modelo, año)

VIAJE (patente, hora_desde, hora_hasta, origen, destino, tarifa, metraje)

a) Listar el dni, nombre y teléfono de todos los dueños que NO son choferes

DUEÑOS <- Π dni (DUEÑO)

CHOFERS <- Π dni (CHOFER)

RES <- Π dni, nombre, teléfono ((DUEÑO - CHOFER) \times DUEÑO)

b) Listar la patente y el id_chofer de todos los autos a cuyos choferes les caduca la licencia el 01/01/2026

CHOFERLICENCIA <- Π id_chofer ((σ fecha_licencia_hasta = 01/01/2026 CHOFER))

RES <- Π patente, id_chofer (AUTO \times CHOFERLICENCIA)

7) Estudiantes y carreras

ESTUDIANTE (#legajo, nombreCompleto, nacionalidad, añoDelIngreso, códigoDeCarrera)

CARRERA (códigoDeCarrera, nombre)

INSCRIPCIONAMATERIA (#legajo, códigoDeMateria)

MATERIA (códigoDeMateria, nombre)

a) Obtener el nombre de los estudiantes que ingresaron en 2023.

```
RES <- Π nombreCompleto (σ añoDelIngreso = '2023' ESTUDIANTE)
```

b) Obtener el nombre de los estudiantes con nacionalidad “Argentina” que NO estén en la carrera con código “LI07”

```
RES <- Π nombreCompleto ((σ (nacionalidad = 'Argentina') & (códigoDeCarrera <> 'LI07')) ESTUDIANTE)
```

c) Obtener el legajo de los estudiantes que se hayan anotado en TODAS las materias.

```
RES <- Π #legajo (INSCRIPCIONAMATERIA % (Π códigoMateria MATERIA))
```

Puedo obviar la proyección del #legajo?

8) Cursos

```
LUGAR_TRABAJO ( #empleado, #departamento )
```

```
CURSO_EXIGIDO ( #departamento, #curso )
```

```
CURSO_REALIZADO ( #empleado, #curso )
```

a) ¿Quiénes son los empleados que han hecho todos los cursos, independientemente de qué departamento los exija?

```
RES <- Π #empleado (CURSO_REALIZADO % (Π #curso CURSO_EXIGIDO))
```

b) ¿Quiénes son los empleados que ya han realizado todos los cursos exigidos por sus departamentos?

```
CURSOS_A_HACER ← Π #empleado, #curso ( LUGAR_TRABAJO |X| CURSO_EXIGIDO )
```

```
CURSOS_PENDIENTES ← CURSOS_A_HACER - ( CURSO_REALIZADO )
```

```
RES ← Π #empleado (LUGAR_TRABAJO) - Π #empleado (CURSOS_PENDIENTES)
```

Preguntar.

```

// pares (empleado, curso) que SU departamento exige
REQ <- π #empleado, #curso ( LUGAR_TRABAJO ⋈ CURSO_EXIGIDO )

// pares exigidos que el empleado NO realizó
MISSING <- REQ - π #empleado, #curso ( CURSO_REALIZADO )

// empleados sin faltantes
RES <- π #empleado (REQ) - π #empleado (MISSING)

```

9) Fabricantes de Muebles

TIPOMUEBLE (id_tipomueble, descripción)

FABRICANTE (id_fabricante, nombrefabricante, cuit)

TIPOMADERA (id_tipomadera, nombremadera)

AMBIENTE (id_ambiente, descripcionambiente)

MUEBLE (id_mueble, id_tipomueble, id_fabricante, id_tipomadera, precio, dimensiones, descripcion)

MUEBLEAMBIENTE (id_mueble, id_ambiente)

a. Obtener los nombres de los fabricantes que fabrican muebles en todos los tipos de madera.

```
MADERAS_TODAS <- Π id_tipomadera (TIPOMADERA)
```

```
FABRICANTES_MUEBLES <- Π id_tipomadera, id_fabricante (MUEBLE)
```

```
RES <- Π nombrefabricante ( (FABRICANTE) |X| (FABRICANTES_MUEBLES % MADERAS_TODAS) )
```

b. Obtener los nombres de los fabricantes que sólo fabrican muebles en Pino.

```
PINO <- Π id_tipomadera (σ nombremadera = 'Pino' TIPOMADERA)
```

```
NO_PINO <- Π id_tipomadera (σ nombremadera <> 'Pino' TIPOMADERA)
```

```
FABRICANTES_PINO <- Π id_fabricante (PINO |X| MUEBLE)
```

```
FABRICANTES_NO_PINO <- Π id_fabricante (NO_PINO |X| MUEBLE)
```

```
RES <- Π nombrefabricante ( FABRICANTE |X| (FABRICANTES_PINO - FABRICANTES_NO_PINO) )
```

Preguntar: es necesario guardar el resultado en RES o directamente puedo proyectar?

c. Obtener los nombres de los fabricantes que fabrican muebles para todos los ambientes.

```
AMBIENTES_TODOS <- Π id_ambiente (AMBIENTE)
```

```
FABRICANTES_AMBIENTES <- Π id_ambiente, id_fabricante (MUEBLE |X| MUEBLEAMBIENTE)
```

```
RES <- Π nombrefabricante ( (FABRICANTE) |X| (FABRICANTES_AMBIENTES % AMBIENTES_TODOS) )
```

d. Obtener los nombres de los fabricantes que sólo fabrican muebles para oficina.

```
OFICINAS <- Π id_ambiente (σ descripcionambiente = 'Oficina' AMBIENTE)
```

```
NO_OFICINAS <- Π id_ambiente (σ descripcionambiente <> 'Oficina' AMBIENTE)
```

```
FABRICANTES_OFICINA <- Π id_fabricante (OFICINAS |X| MUEBLEAMBIENTE |X| MUEBLE )
```

```
FABRICANTES_NO_OFICINA <- Π id_fabricante (NO_OFICINAS |X| MUEBLEAMBIENTE |X| MUEBLE )
```

RES $\leftarrow \Pi$ nombrefabricante (FABRICANTE |X| (FABRICANTES_OFICINA - FABRICANTES_NO_OFICINA))

e. Obtener los nombres de los fabricantes que sólo fabrican muebles para baño y cocina.

BAÑO $\leftarrow \Pi$ id_ambiente (σ descripcionambiente = 'Baño' AMBIENTE)

COCINA $\leftarrow \Pi$ id_ambiente (σ descripcionambiente = 'Cocina' AMBIENTE)

OTROS $\leftarrow \Pi$ id_ambiente (σ (descripcionambiente \neq 'Baño')AND(descripcionambiente \neq 'Cocina') AMBIENTE)

FABRICANTES_BAÑO $\leftarrow \Pi$ id_fabricante (BAÑO |X| MUEBLEAMBIENTE |X| MUEBLE)

FABRICANTES_COCINA $\leftarrow \Pi$ id_fabricante (COCINA |X| MUEBLEAMBIENTE |X| MUEBLE)

FABRICANTES_BAÑO_COCINA \leftarrow (FABRICANTES_BAÑO \cap FABRICANTES_COCINA)

FABRICANTES_OTROS $\leftarrow \Pi$ id_fabricante (OTROS |X| MUEBLEAMBIENTE |X| MUEBLE)

RES $\leftarrow \Pi$ nombrefabricante (FABRICANTE |X| (FABRICANTES_BAÑO_COCINA - FABRICANTES_OTROS))

La solución de la imagen está bien? \rightarrow el producto natural de la primer consulta realmente me devuelve los id_fabricante de aquellos que fabrican muebles para baño Y cocina?

e. Obtener los nombres de los fabricantes que sólo fabrican muebles para baño y cocina.

BANCOCINA $\leftarrow \Pi_{id_fabricante}$ (MUEBLE |X| MUEBLEAMBIENTE |X|

($\Pi_{id_ambiente}$ ($\sigma_{descripcionambiente="baño" \text{ or } descripcionambiente="cocina"}$ (AMBIENTE))))

NOBANCOCINA $\leftarrow \Pi_{id_fabricante}$ (MUEBLE |X| MUEBLEAMBIENTE |X|

($\Pi_{id_ambiente}$ ($\sigma_{descripcionambiente \neq "baño" \text{ and } descripcionambiente \neq "cocina"}$ (AMBIENTE))))

$\Pi_{nombrefabricante}$ ((BANCOCINA - NOBANCOCINA) |X| FABRICANTE)

f. Obtener los nombres de los fabricantes que producen muebles de cedro y roble.

CEDRO $\leftarrow \Pi$ id_tipomadera (σ nombremadera = 'Cedro' TIPOMADERA)

ROBLE $\leftarrow \Pi$ id_tipomadera (σ nombremadera = 'Roble' TIPOMADERA)

FABRICANTES_CEDRO $\leftarrow \Pi$ id_fabricante (CEDRO |X| MUEBLE)

FABRICANTES_ROBLE $\leftarrow \Pi$ id_fabricante (ROBLE |X| MUEBLE)

FABRICANTES_AMBOS \leftarrow (FABRICANTES_CEDRO \cap FABRICANTES_ROBLE)

RES $\leftarrow \Pi$ nombrefabricante (FABRICANTE |X| (FABRICANTES_AMBOS))

g. Obtener los nombres de los fabricantes que producen muebles de melamina o MDF

```
MELAMINA_MDF <- Π id_tipomadera (σ (nombremadera = 'Melamina')OR(nombremadera = 'MDF')
TIPOMADERA)
```

```
FABRICANTES_MELAMINA_MDF <- Π id_fabricante (MELAMINA_MDF |X| MUEBLE )
```

```
RES <- Π nombrefabricante (FABRICANTE |X| (FABRICANTES_MELAMINA_MDF))
```

Consultar

OTRA OPCION MAS LARGA

```
MELAMINA <- Π id_tipomadera (σ nombremadera = 'Melamina' TIPOMADERA)
```

```
MDF <- Π id_tipomadera (σ nombremadera = 'MDF' TIPOMADERA)
```

```
FABRICANTES_MELAMINA <- Π id_fabricante (MELAMINA |X| MUEBLE )
```

```
FABRICANTES_MDF <- Π id_fabricante (MDF |X| MUEBLE )
```

```
FABRICANTES_AMBOS <- ( FABRICANTES_MELAMINA U FABRICANTES_MDF )
```

```
RES <- Π nombrefabricante (FABRICANTE |X| (FABRICANTES_AMBOS))
```